

Software Development for Supporting Trustworthiness Assessment in Computer-Mediated Introduction: A Requirements Engineering Approach

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Informatik und Angewandte Kognitionswissenschaft
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation von

Angela Borchert

aus

Bochum, Deutschland

1. Gutachter: Prof. Dr. Maritta Heisel
2. Gutachter: Prof. Dr. Nicole Krämer

*To the discovery of ideas, visions,
and possibilities.*



by Chiharu Shiota.

Trust the process

Abstract

Getting to know new people via social media is a challenge. Users struggle with assessing the trustworthiness of others via the software application. However, believing that other users are trustworthy and act towards one's positive expectations is often-times a prerequisite for interactions - whether to start or continue online interactions or transfer them to the offline sphere. This work investigates the research objective of how software engineers can develop social media applications that support users in their online trustworthiness assessment. This dissertation is based on previous trust research and a conducted literature review to provide a theoretical model for trustworthiness in social media. Furthermore, existing practices in requirements engineering, risk management, goal modelling, and feature modelling are considered, to introduce various practical methods for software engineers. In addition, quantitative and qualitative research is conducted to apply, test, and evaluate the methods of this work. The theoretical results of this work are the *trustworthiness framework for computer-mediated introductions* and the *overview of trustworthiness facets*. Both provide a theoretical framework for the practical implications by which psychological processes can be transferred to the online environment. Practical results are the *method for eliciting trust-related software features*, called *TrustSoFt*, the *guideline for selecting trustworthiness facets*, i* goal modelling for *TrustSoFt*, and the *method for establishing feature models for trustworthiness assessment*. All practical approaches support software engineers in specifying trust-related software features by which users can assess the trustworthiness of other users, the service provider, and the social media application. The qualitative and quantitative evaluation of *TrustSoFt* has shown that *TrustSoFt* is appreciated by users of the method. The resulting software features of *TrustSoFt* reduce the trust concerns of end users. The *TrustSoFt* use case for online dating applications resulted in innovative trust-related software features that are not used by existing applications, yet. Future work can facilitate the *TrustSoFt* application by developing a digital tool. Furthermore, additional qualitative and quantitative studies or practical applications in industrial software development projects can support the validity of the theoretical and practical findings.

Keywords: trustworthiness assessment, requirements engineering, computer-mediated introduction

Kurzfassung

Neue Menschen über soziale Medien kennenzulernen ist eine Herausforderung. Für Nutzende ist es schwierig die Vertrauenswürdigkeit anderer Nutzender mithilfe der Software-Anwendung einzuschätzen. Jedoch ist der Glaube, dass Andere vertrauenswürdig sind und sich entsprechend der eigenen positiven Erwartungen verhalten oftmals die Voraussetzungen für eine Interaktion - sei es, um Online-Interaktionen zu beginnen, fortzusetzen, oder sie in die Offline-Sphäre zu verlegen. Diese Arbeit setzt sich als Forschungsziel, Software-Ingenieure bei der Entwicklung von Social-Media-Anwendungen anzuleiten, welche Nutzende bei ihrer Beurteilung der Vertrauenswürdigkeit unterstützen.

Die Arbeit stützt sich auf die bisherige Vertrauensforschung sowie eine durchgeführte strukturierte Literaturrecherche, um ein theoretisches Modell für die Vertrauenswürdigkeit in sozialen Medien abzuleiten. Darüber hinaus werden bestehende Praktiken in den Bereichen Requirements Engineering, Risikomanagement, Zielmodellierung und Feature-Modellierung berücksichtigt, um verschiedene praktische Methodiken für Software-Ingenieure aufzuzeigen. Des Weiteren werden quantitative und qualitative Analysen durchgeführt, um die Methodiken dieser Arbeit anzuwenden, zu testen und zu evaluieren.

Die theoretischen Ergebnisse dieser Arbeit sind der *Vertrauenswürdigkeitskontext für computervermittelte Einführungen* und der *Überblick über die Facetten der Vertrauenswürdigkeit*. Beide bieten einen theoretischen Rahmen für die praktischen Ansätze, durch die psychologische Vertrauensprozesse in die Online-Umgebung übertragen werden können.

Praxisbezogene Ergebnisse sind die *Methode zur Ermittlung vertrauensbezogener Software-Features*, kurz genannt *TrustSoFt*, der *Leitfaden zur Auswahl von Vertrauenswürdigkeitsfacetten*, i* Zielmodellierung für TrustSoFt und die *Methode zur Erstellung von Feature-Modellen für die Vertrauenswürdigkeitsbewertung*. Alle methodischen Ansätze unterstützen Software-Ingenieure bei der Spezifikation vertrauensbezogener Software-Features, anhand derer Nutzende die Vertrauenswürdigkeit anderer Nutzende, des Dienstansbieters und der Social-Media-Anwendung bewerten können.

Die qualitative und quantitative Evaluierung von TrustSoFt hat gezeigt, dass TrustSoFt von Anwendern geschätzt wird und dass die resultierenden Software-Features die Vertrauensbedenken der Nutzenden verringern. Die Durchführung Trust-

SoFts für Online-Dating-Applikationen führten zu innovativen vertrauensbezogenen Software-Features, die von vorhandenen Applikationen noch nicht genutzt werden. Zukünftige Arbeiten können die Durchführung von TrustSoFt durch die Entwicklung eines digitalen Werkzeugs erleichtern. Darüber hinaus können zusätzliche qualitative und quantitative Studien sowie das Durchführen von TrustSoFt in Softwareentwicklungsprojekten der Industrie die Validität der theoretischen und praktischen Ergebnisse unterstützen.

Stichworte: Bewertung der Vertrauenswürdigkeit, Requirements Engineering, computer-vermitteltes Kennenlernen

Acknowledgments

Als die Zeit der Doktorarbeit im August 2018 begann, konnte ich es kaum fassen, dass ich wirklich diesen Schritt gehe. Voller Freude habe ich dieses Kapitel meines Lebens begonnen – und voller Freude blicke ich auf diese Zeit zurück. Es war ein Auf und Ab an Emotionen, Ideen und Textpassagen, die ich erweiterte und teilweise auch wieder löschte. Es war eine Zeit der Diskussionen, der Denkanstöße und der Finesse, Gedanken einzufangen, auszubauen und zu vollenden. Diese Zeit, das Ergebnis in Form dieser Dissertation und mein Wachstum als Person, verdanke ich zum Großteil mir.

Ich war eine Person des Selbstzweifels, die sich und ihre Ideen in Frage gestellt hat und sich sorgte, nicht gut genug zu sein. Eine Person, die sich stets bemüht hatte, andere zufriedenzustellen und für gut befunden zu werden. Ich würde lügen jetzt zu behaupten, dass diese Facette in mir nicht mehr vorhanden sei. Alte Muster besiegt man eben nicht von heute auf morgen. Aber ich habe in all der Zeit gelernt, mir selbst zu vertrauen. Den Selbstzweifel loszulassen. Denn was soll schon passieren, wenn ich anfangs auf mich zu hören und mich endlich zu verwirklichen? Nach all der Zeit bin ich wahrhaftig und zweifellos ich selbst.

Meine Dissertation ist maßgebend für diesen Prozess gewesen. Die Niederschläge und Erfolge meiner Forschung in Form von Kritik, Ablehnungen und Zusagen meiner wissenschaftlichen Arbeiten haben mein Durchhaltevermögen und meine Stressresistenz getestet. Die Kraft und den Mut stetig weiterzumachen, habe ich vor allem durch die Unterstützung von großartigen Menschen aufbringen können. Dazu gehören unter anderem meine Doktormutter Prof. Dr. Maritta Heisel und mein Arbeitskollege Dr. Nicolás Díaz Ferreyra. Bei jeglichen Zweifeln an mir und meinen Ideen waren sie an meiner Seite. Zusätzlich zu ihrer fachlichen Unterstützung hatten sie für mich ein offenes Ohr und warme Worte parat. Außerdem möchte ich auch ein großes Dankeschön an meine Herzensmenschen richten. Während einige von ihnen mich für einen gewissen Zeitraum begleitet haben, sind manche immer noch an meiner Seite und andere neu dazugestoßen. Ich danke euch für den Spiegel, den ihr mir vorgehalten habt, für eure Ehrlichkeit und eure Impulse mir neue Perspektiven aufzuzeigen. Zu meinen Herzensmenschen gehören unter anderem meine Mutter Ubonwanna Promsooth, Sarah Bludau und Sebastian Kaczynski. Meine Mutter schenkt mir stets bedingungslose Liebe und ihren Glauben, dass ich alles schaffen kann. Meine Yogini Sarah erinnert mich an meine Stärke und mein Licht,

wenn ich mal am Boden bin und die Welt etwas dunkel erscheint. Sebastian konnte mir Lösungen aufgezeigt, wenn ich mal keinen klaren Gedanken fassen konnte. Dank ihm weiß ich um meiner Würde und meiner Kraft der Akzeptanz. Zusätzlich zu den Fünfen gibt es noch so viele liebe Menschen, die mich unterstützen und an mich glauben. Ihre positive Energie spornt mich an, jeden Tag mein Bestes zu geben. Ich danke euch allen aus ganzem Herzen.

Mit dieser Dissertation habe ich mir selbst bewiesen, dass Vertrauen einem die Kraft geben kann, über sich selbst hinauszuwachsen. Ich bin dankbar für all die Erfahrungen der Vergangenheit, die mich zu der gemacht haben, die ich heute bin. Nun bin ich voller Vorfreude auf all das, was noch kommen mag.

Included publications

Partial results of this dissertation have been published in:

Paper 1 Angela Borchert, Nicolás Emilio Díaz Ferreyra, and Maritta Heisel. Building trustworthiness in computer-mediated introduction: A facet-oriented framework. In *International Conference on Social Media and Society*, pages 39–46, 2020.

Paper 2 Angela Borchert and Maritta Heisel. The role of trustworthiness facets for developing social media applications: A structured literature review. *Information*, 13(1):34, 2022.

Paper 3 Angela Borchert, Nicolás Emilio Díaz Ferreyra, and Maritta Heisel. A conceptual method for eliciting trust-related software features for computer-mediated introduction. In *ENASE*, pages 269–280, 2020.

Paper 4 Angela Borchert, Nicolas Emilio Díaz Ferreyra, and Maritta Heisel. Balancing trust and privacy in computer-mediated introduction: featuring risk as a determinant for trustworthiness requirements elicitation. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–10, 2020.

Paper 5 Angela Borchert and Maritta Heisel. Conflict identification and resolution for trust-related requirements elicitation a goal modeling approach. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 12(1):111–131, 2021.

Paper 6 Angela Borchert, Nicolás E Díaz Ferreyra, and Maritta Heisel. Meeting strangers online: Feature models for trustworthiness assessment. In *Human-Centered Software Engineering: 9th IFIP WG 13.2 International Working Conference, HCSE 2022, Eindhoven, The Netherlands, August 24–26, 2022, Proceedings*, pages 3–22. Springer, 2022.

Paper 7 Angela Borchert, Aidmar Wainakh, Nicole Krämer, Max Mühlhäuser, and Maritta Heisel. Mitigating privacy concerns by developing trust-related software features for a hybrid social media application. In *ENASE*, pages 269–280, 2021.

Paper 8 Angela Borchert, Aidmar Wainakh, Nicole Krämer, Max Mühlhäuser, and Maritta Heisel. The relevance of privacy concerns, trust, and risk for hybrid social media. In *Evaluation of Novel Approaches to Software Engineering: 16th International Conference, ENASE 2021, Virtual Event, April 26–27, 2021, Revised Selected Papers*, pages 88–111. Springer, 2022.

Paper 9 Angela Borchert, Elija Cassidy, and Maritta Heisel. Safety First? Gender Differences in Online Dating Behavior and Trust Concerns. Submitted for publication, 2023.

Contents

Abstract	v
Kurzfassung	viii
Acknowledgments	xii
Included Publications	xiii
Contents	xv
List of figures	xxi
List of tables	xxix
Abbreviation	xxxiii
1 Introduction	1
1.1 The Interplay of Users' Psychological Processes and the Social Media Application	2
1.2 Impression Management and the Relevance of Trustworthiness Assessments in Social Media	3
1.3 Computer-mediated Introductions	4
1.4 Considering Trustworthiness in Software Engineering	6
1.5 Research Outline	7
2 Theoretical Background	13
2.1 Trust and Trustworthiness	14
2.2 Software Development Life Cycle and Requirements Engineering . . .	16
2.3 Software Features and Digital Nudges	19
2.4 Method for Systematic Analysis of Trustworthiness Requirements . .	20
2.5 Risk	21
2.6 i* Goal Modelling Notation	24
2.7 Feature Models	28
3 Building a Trust Framework for CMI	31
3.1 The Framework of Trustworthiness for CMI	32

3.2	Trustworthiness Facets	35
3.2.1	The Overview of Trustworthiness Facets	37
3.2.2	The Guideline for Selecting Trustworthiness Facets	38
4	TrustSoFt - A Method for Eliciting Trust-Related Software Features	45
5	Risk as a Determinant for Prioritisation in TrustSoFt	55
5.1	Types of Conflicts	56
5.2	The TrustSoFt method extended by risk	60
6	Goal Modelling for TrustSoFt - the Model-Based Approach	73
6.1	The Adapted i^* Notation for TrustSoFt	74
6.2	The Procedure for Goal Model Creation in TrustSoFt	77
6.3	Conflict Identification and Resolution with the Adapted TrustSoFt Goal Models	82
7	Evaluation of TrustSoFt	91
7.1	Limitations of the evaluation	99
7.2	Take-home message of the TrustSoFt evaluation	100
8	The Enhanced TrustSoFt Concept	102
9	Trust-related Software Features - A Concretised Definition	109
10	Catalogue for Trust-Related Software Features	113
11	Method for Establishing Feature Models for Online Trustworthiness Assessments	120
11.1	Features Model Creation	121
11.1.1	Extended Feature Model Notation for Trustworthiness Assessments	123
11.1.2	Feature Modelling	125
11.1.3	The Facet Attribution Process	127
11.1.3.1	The Allocation Phase	127
11.1.3.2	The Propagation Phase	130
11.2	Validation of Trust-related Software Features in Feature Models	132
11.3	Configuration of Trust-Related Software Product Lines	133

11.4 Application Example for the Method for Establishing Feature Models for Trustworthiness Assessments	134
12 Applying TrustSoFt for Developing and Evaluating a Hybrid Social Media Application	141
12.1 Hybrid Social Media	142
12.2 Research Model	143
12.2.1 Understanding the HSM context	144
12.2.2 The Impact of TrustSoFt Software Features	147
12.3 Method	148
12.4 Applying TrustSoFt for HushTweet	152
12.5 Results	154
12.5.1 Population	154
12.5.2 Information Privacy Concerns regarding HSM	158
12.5.3 Results of the Research Models – Hypotheses H1-H5	159
12.5.4 The Impact of TrustSoFt software features – Hypotheses H6, H7a, and H7b	161
12.5.5 Demographic and Issue-related Differences in the Variables	162
12.6 Discussion	170
12.6.1 The Relevance of Information Privacy Concerns	170
12.6.2 Relationships of the Constructs - the Research Models	171
12.6.3 The Role of Demographic and Issue-related Variables for HSM Development	172
12.6.4 Limitations and Future Research	175
12.6.5 The Impact of TrustSoFt Software Features	177
12.6.6 Lessons learnt for TrustSoFt	178
13 Applying TrustSoFt and the Extended Feature Models for a Use Case in Online Dating	181
13.1 Trust Concerns in Online Dating	182
13.2 Deriving Trustworthiness Facets	187
13.3 Requirements Elicitation and Goal Modelling	193
13.3.1 Goal Model for Safety Concern of Female Online Dating Users	195
13.3.2 Goal Model for Misrepresentation Concern of Male Users	200
13.4 Trust-Related Software Features for Online Dating - Feature Models	206
13.4.1 Feature Model for the Safety Concern of Female Users	207

13.4.1.1	Feature Modelling of Date Check	211
13.4.1.2	Facet Allocation for Date Check	217
13.4.1.3	Facet Propagation for Date Check	220
13.4.2	Feature Model for Misrepresentation Concern of Male Users	223
13.4.2.1	Feature Modelling of Appearance Verifier	225
13.4.2.2	Facet Allocation for the Appearance Verifier	231
13.4.2.3	Facet Propagation for Appearance Verifier	234
14	Related Work	237
14.1	Users' Trust Issues in Software Development	237
14.2	From Trustworthiness Goals to Trustworthiness Requirements with i* Goal Modelling	240
14.2.1	Risk Analysis for Conflict Resolution and Requirement Pri- orisation	242
14.3	Software Features and Feature Modelling in the Context of Trust	243
15	Discussion	245
15.1	Results	245
15.2	Discussion of Results	252
15.2.1	RQ1 – Trustworthiness in the context of CMI	252
15.2.2	Ethics about Trustworthiness in the Context of CMI	254
15.2.3	RQ2 – The trustworthiness facets	255
15.2.4	RQ3 – The introduction of TrustSoFt as a conceptual method	257
15.2.5	RQ4 – Considering risk for deciding on conflicting options	259
15.2.6	RQ5 – i* goal modelling for TrustSoFt	260
15.2.7	RQ6 - Feature models for trustworthiness assessment	261
15.2.8	RQ7 - Addressed concerns, trust, and risk in hybrid social media	264
15.2.9	RQ8 – The use case: Safety and Misrepresentation Concerns	266
15.3	Theoretical and Practical Implications	269
16	Conclusion	271
	Bibliography	274
	Appendix	306
A	Overview of Trustworthiness Facets for Individuals	307

B	Overview of Trustworthiness Facets for Technology	310
C	Overview of Trustworthiness Facets for Organisations	313
D	Materials of the TrustSoFt Evaluation	316
D.1	Evaluation Sheet for TrustSoFt	316
D.2	Exemplary Interview Questions by the Developers	318
E	Materials of the HSM user study	319
E.1	Internet Users' Information Privacy Concern scale & Concern for In- formation Privacy Scale	319
E.2	Global Information Privacy Concern	321
E.3	Trusting Beliefs	321
E.4	Risk Beliefs	322
E.5	Perceived Trustworthiness of HushTweet	322
E.6	Willingness to Use HushTweet	323
F	The Catalogues for Trust-Related Software Features for the Online Dating Use Case	325
F.1	The Catalogue for the Safety Check of Users	326
F.2	The Catalogue for the Appearance Verifier	340
F.3	The Catalogue for the Appearance Verifier	349
G	Declaration of Individual Contributions	358

List of Figures

2.1	Risk matrix with the example of “financial loss by scammers” that is evaluated with by a medium risk value.	23
2.2	i* notation for actors, the actor boundary, and the INS-relationship link.	25
2.3	i* notation for the intentional elements goal, soft goal, task, resource, and belief.	26
2.4	i* notation for the dependencies.	26
2.5	i* notation intentional links.	27
2.6	Overview of the feature model notation	29
2.7	Exemplary feature model about for an online dating application. . . .	30
3.1	The framework of trustworthiness for CMI.	33
3.2	Guideline for selecting trustworthiness facets from the overview of trustworthiness facets by consulting users, experts, or literature. The yellow arrows show the guideline for identified problematic characteristics of the vulnerable state, while the green arrows present the paths for the desired characteristics of the goal state.	40
4.1	Conceptual TrustSoFt Method	46
5.1	Example of a hard conflict.	58
5.2	Example of a soft conflict.	59
5.3	Extended TrustSoFt method by the determinant risk.	61
6.1	Adapted i* Notation to TrustSoFt Elements. Inspired by [39].	74

6.2	Exemplary goal model for the catfishing problem in online dating.	82
6.3	Example of frame colour change for trustworthiness requirements and goals during conflict identification and resolution. Figure by [39].	83
6.4	Procedure for Conflict Resolution	87
6.5	Exemplary actor boundary of an online dating application after a conflict is resolved.	89
8.1	The enhanced conceptual TrustSoFt method based on the conducted TrustSoFt evaluation from Chapter 7.	103
9.1	Challenges of online trustworthiness assessments, solution approaches for software features and resulting features types	110
10.1	First part of the catalogue structure for trust-related software features. This part covers the basic information of a trust-related software feature. The figure is taken over from Paper 6 [37].	115
10.2	Second part of catalogue structure for trust-related software features. This part covers the information on an asset of a trust-related software feature. The figure is taken over from Paper 6 [37].	116
11.1	Method for establishing feature models for online trustworthiness assessments. The figure is taken over from Paper 6 [37].	122
11.2	The extended feature model notation for trustworthiness assessments.	123
11.3	Exemplary feature model for catfish protection. The figure is taken over from Paper 6 [37].	136
11.4	Asset information of the catalogue structure for the feature “green check mark”	137
12.1	Overview of the research model including the hypotheses of this study.	145
12.2	Sample of the Full-features Hushtweet mockup.	150

12.3 SEM of the HSM Concept group. ** p < .01, *** p < .001	160
12.4 HSM concept group - simple slopes for the moderations of age and ID misrepresentation.	164
12.5 Full-featured group - simple slopes for the moderation effects of age. .	166
12.6 Full-featured group - Effect of education on countered information privacy concerns on risk beliefs.	167
12.7 Full-featured group - simple slopes for the moderation effects of identification misrepresentation.	168
12.8 Full-featured group - simple slopes for the moderation effects of privacy invasion.	169
13.1 Goal model for safety concern. This model only contains the application with its intentional elements in dependence on the other actors. The full insights of the application are given in Figure 13.2	196
13.2 This is the complete application with its intentional elements, which is part of the goal model in Figure 13.1	198
13.3 Goal model for misrepresentation concern. This model only contains the application with its intentional elements in dependence on the other actors. The full insights of the application are given in Figure 13.4	202
13.4 This is the complete application with its intentional elements, which is part of the goal model in Figure 13.3	204
13.5 Basic information of the catalogue for trust-related software features for the concept feature “safety check of users”.	207
13.6 Trust-related software features for the concept feature “safety check of users”.	208
13.7 Asset information for the trust-related software feature “date check”. .	211

13.8	Feature model for the trust-related software feature “date check” for the safety concern of female online dating users.	216
13.9	List of trustworthiness facets for the concept feature “date check”.	222
13.10	Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”.	224
13.11	Asset information for the trust-related software feature “appearance verifier”.	226
13.12	Feature model for the trust-related software feature “appearance verifier” for the misrepresentation concern of male online dating users.	230
13.13	List of trustworthiness facets for the concept feature “authenticity check”.	235
F.1	Basic information of the catalogue for trust-related software features for the concept feature “safety check of users”.	326
F.2	Asset information for the feature asset “date check”.	327
F.3	Asset information for the feature asset “date terms share”.	328
F.4	Asset information for the feature asset “share button”.	328
F.5	Asset information for the feature asset “sharing algorithm”.	329
F.6	Asset information for the feature asset “date page”.	330
F.7	Asset information for the feature asset “panic button”.	331
F.8	Asset information for the feature asset “police call”.	331
F.9	Asset information for the feature asset “date request”.	332
F.10	Asset information for the feature asset “input field date terms”.	332
F.11	Asset information for the feature asset “information date terms female/male user”.	333
F.12	Asset information for the feature asset “time”.	333

F.13 Asset information for the feature asset “location”	334
F.14 Asset information for the feature asset “button “Ask for date””	334
F.15 Asset information for the feature asset “date invitation”	335
F.16 Asset information for the feature asset “button accept date”	335
F.17 Asset information for the feature asset “button decline date”	336
F.18 Asset information for the feature asset “button on match page to date page”	336
F.19 Asset information for the feature asset “feedback to the date”	337
F.20 Asset information for the feature asset “questions about date”	337
F.21 Asset information for the feature asset “input field for feedback”	338
F.22 Asset information for the feature asset “date term feedback female/- male user”	338
F.23 Asset information for the feature asset “algorithm promise fulfillment score”	339
F.24 Asset information for the feature asset “promise fulfillment score”	339
F.25 Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”	340
F.26 Asset information for the trust-related software feature “appearance verifier”	341
F.27 Asset information for “profile setting page”	341
F.28 Asset information for “profile upload algorithm”	342
F.29 Asset information for “picture upload button”	342
F.30 Asset information for “toggle switch “appearance verifier””	343
F.31 Asset information for “information icon”	343

F.32	Asset information for “information about appearance verifier”	344
F.33	Asset information for “confirmation window “appearance verifier””	344
F.34	Asset information for “approve button”	345
F.35	Asset information for “decline button”	345
F.36	Asset information for “pattern recognition algorithm”	346
F.37	Asset information for “profile picture”	346
F.38	Asset information for “actual appearance”	346
F.39	Asset information for “real-time video”	347
F.40	Asset information for “real-time photo”	347
F.41	Asset information for “authenticity information”	347
F.42	Asset information for “authenticity score”	348
F.43	Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”	349
F.44	Asset information for the trust-related software feature “appearance verifier”	350
F.45	Asset information for “profile setting page”	350
F.46	Asset information for “profile upload algorithm”	351
F.47	Asset information for “picture upload button”	351
F.48	Asset information for “toggle switch “appearance verifier””	352
F.49	Asset information for “information icon”	352
F.50	Asset information for “information about appearance verifier”	353
F.51	Asset information for “confirmation window “appearance verifier””	353
F.52	Asset information for “approve button”	354

F.53	Asset information for “decline button”	354
F.54	Asset information for “pattern recognition algorithm”	355
F.55	Asset information for “profile picture”	355
F.56	Asset information for “actual appearance”	355
F.57	Asset information for “real-time video”	356
F.58	Asset information for “real-time photo”	356
F.59	Asset information for “authenticity information”	356
F.60	Asset information for “authenticity score”	357

List of Tables

1.1	Overview of the scientific papers this work is based on and the research questions 1-6.	10
1.2	Overview of the scientific papers this work is based on and the research questions 7 and 8.	12
4.1	Exemplary TrustSoFt results for the trust concern “catfishing” and the trustworthiness goal “User Authentication”.	53
12.1	First part of the overview of trust-related software features for HushTweet from the TrustSoFt method.	155
12.2	Second part of the overview of trust-related software features for HushTweet from the TrustSoFt method.	156
12.3	Third part of the overview of trust-related software features for HushTweet from the TrustSoFt method.	157
12.4	Experimental groups and descriptive results of the populations.	158
12.5	Descriptive results of the scales.	159
13.1	Women: General Online Dating Concerns	183
13.2	Men: General Online Dating Concerns	183
13.3	Women: Concerns about Online Dating Impact on Own Person	184
13.4	Men: Concerns about Online Dating Impact on Own Person	184
13.5	Women: Trust Concerns About Other Users	185
13.6	Men: Trust Concerns About Other Users	185
13.7	Women: Trust Concerns About Service Providers	186

13.8 Men: Trust Concerns About Service Providers	186
13.9 Women: Trust Concerns About Applications	186
13.10Men: Trust Concerns About Applications	186
13.11Overview of the Identification of Trustworthiness Facets for Female Online Dating Users. The Coloured Trustworthiness Facets are Going to be Considered for Women’s Safety Concern.	191
13.12Overview of the Identification of Trustworthiness Facets for Male On- line Dating Users. The Coloured Trustworthiness Facets are Going to be Considered for Men’s Concern about Misrepresentation.	192
13.13Overview of Relevant Trustworthiness Facets for Use Case Including Definitions.	194
13.14Expressions of the safety concern of female online dating users, pos- sible user goals, and potential trustworthiness goals.	195
13.15Expressions of the misrepresentation concern of male online dating users, possible user goals, and potential trustworthiness goals	203
15.1 Overview of the results from this dissertation for Research Questions RQ1 - RQ4	247
15.2 Overview of the results from this dissertation for Research Questions RQ5 - RQ7	249
15.3 Overview of the results from this dissertation for Research Question RQ8	251

Abbreviation

ANOVA	Analyses of Variance
CFIP	Concern for Information Privacy Scale
CMI	Computer-mediated introduction
GIPC	General Information Privacy Concern Scale
HSM	Hybrid Social Media
ID	Identification
IUIPC	Internet Users' Information Privacy Concern Scale
SDLC	Software Development Life Cycle
TR-SF	Trust-related Software Feature
TrustSoFt	Method for Eliciting Trust-related Software Features

1

Introduction

Since the Internet became available to the general public as a mass medium in 1990, society and people's everyday lives have changed dramatically. With the rise of the Internet, common activities that people usually do in the offline world have been shifted to the online sphere [166]. Such activities are for example communicating with other people, asking for information or help, building or maintaining social relationships, or exchanging goods or services. As these activities usually involve at least two individuals, people rely on social media applications that connect them with other users to perform these activities. Traditional social media like Facebook, Twitter, or LinkedIn focus on connecting people for communication and maintaining relationships. Regarding the search for information, social review sites like Yelp or TripAdvisor or discussion sites like Reddit or Quora allow users to share personal experiences and help each other find answers to their problems. In terms of establishing new relationships, online dating or friendship services like Tinder or Meetup provide users with a platform to get to know new people. Furthermore, sharing economy applications enable individuals to connect with others for exchanging goods or services. Especially in terms of online dating, friendship services, and sharing economy, social media initiate online interactions among strangers that are intended to be shifted into the offline world. These so-called computer-mediated introductions (CMIs) create forms of social relationships that differ from those of offline interactions [321]. At the same time, social media that follow the principle of CMI also influence people's experiences, thoughts, and behaviour in building relationships.

The shift from offline to online activities is accompanied by many challenges for users and the development of social media. This dissertation focuses on the challenges of the interplay between social media applications and users in terms of the psychological process of users' trustworthiness assessment. The next paragraphs of this introduction discuss why trustworthiness assessments are relevant in social

media – especially in those of CMI – and how they can be related to the development of social media applications. At the end of this chapter, the research outline is presented including the structure and the research questions of this dissertation.

1.1 The Interplay of Users’ Psychological Processes and the Social Media Application

As activities have shifted from the offline to the online environment, they are now mediated by software applications. Software applications support online activities by including perceptual cues in their user interface. Perceptual cues may involve visual, auditory or haptic software features (e.g. textual notifications, notification sound, notification vibration) [48]. Thereby, software applications pave the way for communication and interactions with and between users and also trigger and mediate associated psychological processes [299]. Psychological processes cover human behaviour, cognition, and emotion [127]. For example, the sound of an application notification can entice the user to interact with an application more frequently, which may trigger excessive use including Internet usage addiction (behaviour) [197]. In addition, user profiles provide information that an individual would not know when getting to know the other person offline. Thereby, user profiles activate trust-building processes for first online interactions that are different from those offline (cognition) [45]. Visually displayed warnings, such as warning messages from an application or from phishing emails, can trigger a sense of anxiety in users (emotion) [244]. These examples give an outlook of how software applications can impact the psychological processes of their users by presenting perceptual cues.

However, the perceptual cues from software applications differ from those that users are familiar with from the offline world [20]. As a consequence, users may have difficulties interpreting online cues. The difficulties may also concern the modified psychological processes and accompanying personal consequences that the online cues trigger. In addition, perceptual cues in software applications are prone to manipulation by the ones creating them [44]. The creators of online cues are the software development team and the service provider. Thus, the psychological processes of users and the user experience are highly dependent on the design of perceptual cues, which are subject to the intentions and interests of the creators.

For example, a service provider is interested in collecting user data to display tailored advertisements from third-party organisations to users [143]. Therefore, the intention of service providers might be to nudge users into accepting data collection. Yet, the service provider is required by law to inform users about data collection and its purpose as well as to provide them control of what data is collected. As for the design of the perceptual cues, some websites have used color to highlight the button in the terms of use to confirm full data collection. In contrast, the button to restrict data collection was designed to be inconspicuous, for example, by being grayed out. Such a graphical design can manipulate users not being aware of their privacy control. It may entice users to quickly confirm full data collection [118].

1.2 Impression Management and the Relevance of Trustworthiness Assessments in Social Media

Especially in social media, the intention of those creating perceptual cues and thereby impacting user experience is crucial. It can affect the establishment and maintenance of relationships. Besides the software development team and the service provider, social media users are also creators of online perceptual cues. By user-generated content in form of profiles, posts, comments, or chat messages, users can actively impact the psychological processes of other users [171].

When creating user-generated content, an underlying intention of users is to present themselves “in a good light”. Users try to make a positive impression on others. This phenomenon is called impression management [92]. It can also be performed by the service provider via the software application [133]. Impression management is a representation of the self that can differ from the actual self with or without conscious intent. Such a deviation may concern information about personality, appearance, or experiences made. By impression management, users or service providers positively impact their perceived trustworthiness [251].

Perceived trustworthiness is a crucial factor for social interactions and relationship-building [193]. Based on the perceived trustworthiness, people decide whether to start or continue interactions with another party [96]. Perceived trustworthiness negatively correlates with perceived risk [208]. This means that the higher the trustworthiness of a party is perceived, the less likely it is believed that risks occur during

an interaction with the party. When a party is perceived as trustworthy, risks of the interaction are tolerated and relationships deepen [297]. Yet, perceived trustworthiness does not ultimately imply that a party actually is trustworthy - meaning that the party will act in accordance with one's positive expectations [174]. However, as nobody can really know how another party will act in the future, a trustworthiness assessment is most often the only indicator of whether an interaction is safe and promising for the own interest [168].

Due to the importance of the trustworthiness assessment for social interactions and relationship-building, it is in the users' interest to conduct it properly. Yet, performing a trustworthiness assessment online is challenging. As mentioned before, other parties can try to deceive users about their trustworthiness on purpose. Moreover, online trustworthiness assessments involve different cues than offline ones. As a result, online trustworthiness assessments are modified and potentially more complex online than offline ones in terms of interpretation [7]. Moreover, trustworthiness assessments are oftentimes an unconscious psychological process [33]. As a consequence, users may not perform them at all or are unaware of them. Furthermore, users might not be able to name why they believe another party to be trustworthy as the resulting trust is simply a gut feeling [33].

On these grounds, social media users need support in their trustworthiness assessment to perform it accurately and be aware of the importance of perceived trustworthiness. This work focuses on the process of trustworthiness assessments in the context of social media applications and how social media applications – more precisely the software development team – can support users in performing the trustworthiness assessment.

1.3 Computer-mediated Introductions

Trustworthiness assessments are highly relevant for all types of social networks. Assessing the trustworthiness of the three social media parties i) service provider, ii) its software application as a technical entity, and iii) other users impact whether social media users want to engage with the application in general and the users in specific [90, 142, 204]. All three social media parties affect the online activity that a user desires to perform via the application [14].

Trustworthiness assessments are discussed to be especially important when users are lacking experience with the other party [219]. This is the case for initial interactions, such as between two strangers. In such interactions, individuals perceive involved risks as particularly serious [123]. For individuals, it is a challenge to evaluate whether strangers have other intentions and interests than the ones they communicate with. They lack empirical values from previous interactions, which serve as a reference as to whether their counterpart is acting according to expectations. Therefore, the spectrum of potential unwanted incidents is perceived as large. Women especially perceive the risk of interactions with strangers as high [123]. They are concerned about violent crime or sexual violence from male strangers.

Interaction with strangers is part of the business model of social media applications that focus on so-called *computer-mediated introductions* (CMIs). CMI applications introduce strangers with compatible interests online for potential offline encounters [242]. CMI can be distinguished between private and business CMI. While private CMI focuses on social exchange, business CMI most often involves a monetary exchange for goods or services between private buyers and providers [242]. Private CMI covers online dating or friendship applications that people use to establish romantic, sexual, or platonic relationships. Business CMI covers sharing economy applications such as for interests like private lodging, ride-sharing, or shared food consumption to counter food waste.

In contrast to other types of social media, the spectrum of CMI risks and related unwanted incidents is larger. In addition to the online risks of social media use, there are risks specific to CMI that involve face-to-face interactions. Online risks are for example hacking, identity theft or cyber-bullying [5, 137]. While these risks emanate from other users, there are also risks coming from organisational structures like the service provider or the application as a form of technology. Service providers and third-party organisations like advertisers pose privacy risks like data misuse [285]. Using software applications involves security risks [125]. Regarding CMI risks from offline encounters, there are differences between private and business CMI. For online dating, risks rank from damaged self-esteem when romantic feelings are hurt to sexually transmitted diseases, ghosting, the online dating romance scam, and date rape [300, 291, 46, 249]. Ghosting describes the sudden ignoring of a matching user [291]. Online dating romance scam usually involves fake profiles with the intent that users fall in love with the profile. After a while, the scammer invents

an emergency and exerts pressure on the victim to provide financial help [46]. Date rape describes the case when perpetrators use online dating to find victims for sexual violence during offline encounters [249]. Concerning sharing economy, the risks differ depending on the sector as well as for buyers and sellers. In general, buyers risk the poor quality of services or products. This may relate to a dangerous driving style for ride-sharing, missing inventory in rented accommodations, or rotten food in the food sector [62, 325, 333]. For sellers, risks can be to be underpaid by buyers, damage to rented objects, or robbery [325].

Due to the high risks of CMI, this dissertation focuses on CMI in particular, even though the findings can be applied to social media in general. To reduce the complexity of various user roles, application examples for the following chapters focus on the context of online dating, disregarding Sharing Economy.

1.4 Considering Trustworthiness in Software Engineering

Since CMI poses high risks, the trustworthiness assessment serves as a way for users to reduce their concerns and confidently engage with the CMI application for the particular online activity [120]. However, as explained in the previous paragraph, assessing trustworthiness online is challenging and offers no guarantee that another party actually is trustworthy. Therefore, software applications should provide users with perceptual cues by which they can perform their trustworthiness assessment as best as possible. The best outcome of a trustworthiness assessment is when the perceived trustworthiness of another party converges with its actual trustworthiness. At that point, users can evaluate best whether a CMI risk is justified for an interaction. By aiding users in their trustworthiness assessment, software applications can actively contribute to the mitigation of CMI risks.

State-of-the-art software development does consider psychological processes for designing applications that people love to use. There is software development that targets habit-forming digital products [94] or considers learning processes for e-learning platforms [318]. Yet, in the context of trust, software development mainly targets the design of trustworthy software. Trustworthy software means that software runs as expected and is aligned to certain software qualities, such as confiden-

tiality or security [134]. For trustworthy software, methods and technologies have been established, such as the requirements specification method by Mohammadi et al. [229] or the blockchain technology [271]. However, to the knowledge of the author, there are no software development methods that consider the psychological process of people's trustworthiness assessment of other parties within software design. Instead of users' trustworthiness assessment, software development techniques usually focus on user needs. As an example, so-called user stories are created to determine who the user is, what the user wants, and why the user wants it [202]. While such techniques may support users' trustworthiness assessment indirectly by chance, the actual psychological processes are disregarded for software development. Therefore, this dissertation addresses the research gap in how software engineers can consider the users' trustworthiness assessment and the perceived trustworthiness of the three CMI parties in software development.

1.5 Research Outline

Social media and CMI are online environments that involve risks emanating from the three parties i) the service provider, ii) the software application, and iii) the users. Users perform trustworthiness assessments of these parties via the social media application to estimate whether the associated risks are justified for the specific interaction. The trustworthiness assessment impacts the decision of social media users on whether to start or continue interactions. Concerning CMI, the perceived trustworthiness of other users is additionally relevant for the decision of whether to meet them in the offline world.

However, performing a trustworthiness assessment online has its challenges. Reasons for this are the given online cues by the software application that are prone to manipulation by the three parties and that are unfamiliar to users compared to offline cues. Another reason is the assessment itself, which oftentimes is an unconscious process. Users are lacking awareness about the process in specific and the role of perceived trustworthiness in general.

On these grounds, social media users are in need of a supportive software application that considers their underlying psychological process of the trustworthiness assessment in its design. By providing useful trust-related software features in the

user interface, users can be supported in performing the trustworthiness assessment accurately. It is assumed that a properly conducted trustworthiness assessment, whose resulting perceived trustworthiness resembles the actual trustworthiness of other parties, mitigates the likelihood of risks occurring. Based on this assumption, software development teams can build applications that are safer for their users. In addition, software engineers can positively impact the user experience by supporting the trustworthiness assessment. Users additionally feel safer and can relax during social interactions in this regard, which impacts the success of the performed on-line activity. Thereby, users are more satisfied when using the application leading to higher engagement with the application, which again serves the service provider and its business model.

Therefore, the research objective of this dissertation is to provide software development teams with a method for developing user-centered social media applications. The method shall consider users' underlying psychological processes of the trustworthiness assessment. The resulting software application shall provide useful trust-related software features by which users are enabled and supported in their trustworthiness assessment.

The dissertation is based on nine scientific papers that are listed on page xiii in the front matter and in Tables 1.1 and 1.2. Each paper addresses a different research question that contributes to the research objective. Throughout the dissertation, the chapters refer to the single papers as a guiding structure. The structure is explained in the following, together with the research questions of this work. An overview of the research questions and the related scientific papers is given below in Tables 1.1 and 1.2. An overview of the papers and the authors is given in the preface of this work.

The first step for accomplishing the research objective is to gain knowledge about trust in the context of CMI. Therefore, Research Question RQ1 asks how trustworthiness is involved in social media and CMI systems. RQ1 is answered in Paper 1 "Building Trustworthiness in Computer-mediated Introduction: A Facet-oriented Framework". Furthermore, Paper 1 introduces the concept of trustworthiness facets. The findings are explained in Chapter 3.1.

Paper 2 is "The Role of Trustworthiness Facets for Developing Social Media Applications: A Literature Review". It answers Research Question RQ2: "What

are the trustworthiness facets of individuals (e.g., user), organisations (e.g., service provider), and technology (e.g., application)?”. Chapter 3.2 picks up Paper 2 and provides an overview of the trustworthiness facts of the three social media parties that have been identified via a literature review. The overview of trustworthiness facets provides software developers with a database that is relevant for developing social media applications that consider users’ trustworthiness facets.

After Chapters 3.1 and 3.2 have addressed the basis of trustworthiness regarding CMI use, Chapter 4 introduces the method for eliciting trust-related software features (TrustSoFt). TrustSoFt is a requirements elicitation method that has been introduced in Paper 3 “A Conceptual Method for Eliciting Trust-Related Software Features for Computer-mediated Introduction”. It provides a solution to Research Question RQ3: “How can software developers build social media systems that support their users in their trustworthiness assessment?”. Yet, TrustSoFt is a conceptual method. It gets refined in the following chapters.

The first refinement of the TrustSoFt method is presented in Chapter 5. It refers to Paper 4 “Balancing Trust and Privacy in Computer-mediated Introduction: Featuring Risk as a Determinant for Trustworthiness Requirements Elicitation”. Chapter 5 introduces risk as a deciding determinant for those requirements resulting from TrustSoFt that conflict with each other. Thereby, Chapter 5 answers Research Question RQ4: “How can software developers prioritise TrustSoFt elements and decide on conflicting ones during software development?”. For the two purposes mentioned in Research Question RQ4, TrustSoFt gets extended by steps for risk assessment and risk management.

Another refinement of TrustSoFt is made in Chapter 6. By Paper 5 “Conflict Identification and Resolution for Trust-Related Requirements Elicitation: A Goal Modeling Approach”, TrustSoFt is extended by the approach of goal modelling. Thereby, Research Question RQ5 is addressed: “How can the software development process for supporting users’ trustworthiness assessment be conducted systematically as a model-based approach?”. Goal modelling aims to support software developers in applying TrustSoFt while simultaneously documenting the whole process. In addition, this work introduces how goal modelling can be used for conflict identification and resolution between the TrustSoFt elements “goal” and “requirement”. Thereby, the answer to Research Question RQ4 is enriched by a model-based approach.

Publication	Research Question
<u>Paper 1:</u> Building Trustworthiness in Computer-mediated Introduction: A Facet-oriented Framework	<u>RQ1:</u> How is trustworthiness involved in social media and CMI systems?
<u>Paper 2:</u> The Role of Trustworthiness Facets for Developing Social Media Applications: A Literature Review	<u>RQ2:</u> What are the trustworthiness facets of individuals (e.g., user), organisations (e.g., service provider), and technology (e.g., application)?
<u>Paper 3:</u> A Conceptual Method for Eliciting Trust-Related Software Features for Computer-mediated Introduction	<u>RQ3:</u> How can software developers build social media systems that support their users in their trustworthiness assessment?
<u>Paper 4:</u> Balancing Trust and Privacy in Computer-mediated Introduction: Featuring Risk as a Determinant for Trustworthiness Requirements Elicitation	<u>RQ4:</u> How can software developers prioritise TrustSoFt elements and decide on conflicting ones during software development?
<u>Paper 5:</u> Conflict Identification and Resolution for Trust-Related Requirements Elicitation: A Goal Modeling Approach	<u>RQ4:</u> How can software developers prioritise TrustSoFt elements and decide on conflicting ones during software development? <u>RQ5:</u> How can the software development process for supporting users' trustworthiness assessment be conducted systematically as a model-based approach?
<u>Paper 6:</u> Meeting Strangers Online: Feature Models for Trustworthiness Assessment.	<u>RQ6:</u> How can trust-related software features be created, documented, configured, and validated?

Table 1.1: Overview of the scientific papers this work is based on and the research questions 1-6.

At this point in the development of TrustSoFt, the method was evaluated in student projects for developing online dating and sharing economy applications. The TrustSoFt evaluation is presented in Chapter 7. Based on the findings of the evaluation, the drawbacks of TrustSoFt were eliminated. The concept of TrustSoFt is updated in Chapter 8.

To that point, TrustSoFt is a requirements elicitation method that inspires trust-related software features. For a structured and documented specification of trust-related software features, Chapters 9, 10 and 11 introduce trust-related software features in more detail and present an extended form of feature models. The chapters are based on Paper 6 “Meeting Strangers Online: Feature Models for Trustworthiness Assessment”. They address Research Question RQ6: “How can trust-related software features be created, documented, configured, and validated?”. As pointed out in the research question, the feature models can additionally be used for the configuration and validation of software features and software product lines. The output of TrustSoFt serves as input for the establishment of feature models. The resulting software features can be included in social media or CMI applications for supporting users in their trustworthiness assessment.

Answering Research Questions 1 - 6 to that point is sufficient for accomplishing the research objective. The subsequent chapters are about the application of TrustSoFt to application examples. Chapter 12 presents how TrustSoFt has been applied for the development of a hybrid social media application. Chapter 12 relates to Paper 7 – “Developing Trust-related Software Features for a Hybrid Social Media Application” – and Paper 8: “The Relevance of Privacy Concerns, Trust, and Risk for Hybrid Social Media”. The focus is on Research Question RQ7: “How do software features resulting from software development to support users’ trustworthiness assessment impact users?” Software features that resulted from the TrustSoFt method were implemented in prototypes and tested in an online user survey. Users were analysed concerning the perceived trustworthiness of the prototypes, privacy concerns, perceived risk, and willingness to use the respective application.

Another use case for TrustSoFt is presented in Chapter 13. It addresses Research Question RQ8: “What are the trust concerns of female and male online dating users?”. Paper 9 – “Safety First? Gender Differences in Online Dating Behavior and Trust Concerns” – identifies the concerns by an interview study. The results

Publication	Research Question
<p><u>Paper 7:</u> Mitigating Privacy Concerns by Developing Trust-related Software Features for a Hybrid Social Media Application</p> <p><u>Paper 8:</u> The Relevance of Privacy Concerns, Trust, and Risk for Hybrid Social Media</p>	<p><u>RQ7:</u> How do software features resulting from software development to support users' trustworthiness assessment impact users?</p>
<p><u>Paper 9:</u> Safety First? Gender Differences in Online Dating Behavior and Trust Concerns</p>	<p><u>RQ8:</u> What are the trust concerns of female and male online dating users?</p>

Table 1.2: Overview of the scientific papers this work is based on and the research questions 7 and 8.

serve as input for TrustSoFt to specify trust-related software features that counter two major concerns of online dating users. For the use case, goal modelling and feature modelling are demonstrated.

After the application examples, related work to this dissertation is presented in Chapter 14. This is followed by the discussion of the results in Chapter 15 including limitations, future work, and theoretical and practical implications. The last Chapter of this dissertation is the conclusion in Chapter 16.

2

Theoretical Background

As this work strives to support software engineers in developing software applications that aid users in their trustworthiness assessment, this work is grounded on previous research on trust, methods for software development, and modelling notations valuable for software engineering. Former trust research provides a fundamental basis for understanding the internal processes of individuals during trust-building. These trust-building processes are related to perceived trustworthiness and the evaluation and tolerance of associated risks. Therefore, software engineers need knowledge about trust-building to enable and support trust-building processes via a software application. Section 2.1 introduces the terms *trust* and *trustworthiness*. Furthermore, it reports former trust research relevant to the development of social media applications like CMI.

A significant contribution of this dissertation is a software engineering method for the early phase of software development by which software requirements and features can be elicited. Therefore, Section 2.2 introduced the Software Development Life Cycle (SDLC) including the relevance of requirements engineers and software requirements in this context. Next, Section 2.3 introduces software features and digital nudges, which can be regarded as a special kind of software feature.

In addition, knowledge about software development procedures in the context of trust is also significant for this work. For this purpose, Section 2.4 introduces the *Method for Systematic Analysis of Trustworthiness Requirements* by Mohammadi et al.[229]. The method by Mohammad et al. serves as a basis for the method developed in this dissertation. They resemble each other in their structure which allows the transfer of users' trust issues for the development of trust-related software features.

During the creation of this work, further knowledge has been proven fundamental. The concept of risk and risk assessment is introduced in Section 2.5, as previous research identified risk as a valuable determinant for choosing among options of which some can be conflicting with each other [304]. Furthermore, modelling has been proven a beneficial tool for software engineering methods to increase clarity and domain understanding [131]. On these grounds, goal modelling and feature modelling are included in the theoretical background, because they match the content of the method introduced in this dissertation. Therefore, the i^* goal modelling notation is presented in Section 2.6 and the feature modelling notation in Section 2.7.

2.1 Trust and Trustworthiness

Trust has been analyzed in various social science disciplines such as philosophy, sociology, psychology, and economics. In social science, trust is understood as an underlying concept of relationships between one entity and another [115]. Within such relationships, entities can take the role of *trustor* and *trustee*. The trustor is the trusting party, while the trustee is the party to be trusted [213]. Since relationships are most often based on mutuality, a party usually takes the role of trustor and trustee simultaneously [97]. Depending on the application field, different kinds of parties can take the role of trustor or trustees, for example, individuals in general, role-specific individuals like employers or employees, or entities like governments and organisations. In today's digital age, trust has additionally become a relevant factor in the relationship of users with technology [174]. Oftentimes, it depends on users' trust in technology whether they make use of it [217]. Thus, trust research is an uprising topic in the discipline of computer science, as well.

Throughout the disciplines, trust is characterized by the three dimensions of uncertainty, vulnerability, and positive expectations. Uncertainty describes an infinite set of events that may occur, but whose probability of occurrence is complex to evaluate and oftentimes not certainly known [266, 203]. Uncertainty in relationships and interactions is present because a trustor cannot certainly know how a trustee acts in the future. The trustor lacks information about the true intention of the trustee [234]. Moreover, the trustee is an autonomous, uncontrollable party so that behaviour cannot be forced or guaranteed by others [234]. Therefore, relationships and interactions are risky endeavours, as they might result in undesired outcomes

for the involved parties [203].

For this reason, trust refers to a trustor's willingness to make oneself vulnerable to a trustee [16]. Vulnerability includes the tolerance of uncertainty and associated risks with an interaction [203]. Hence, trust can be regarded as a coping strategy for uncertainty [234] that reflects the confidence of a trustor for desired outcomes [203]. This confidence is rooted in the trustor's perception of cues and traits presented by the trustee. Based on the perceived cues and traits, the trustor tries to assess the trustee's trustworthiness. Due to this trustworthiness assessment, a trustor estimates whether a trustee is able and willing to behave as desired in a given situation [217]. This perspective of trust relates to the definition of Lewicki and Wiethoff, who describe trust as positive expectations the trustor holds about a trustee [193]. As positive expectations relate to a subjective assumption, Lewicki and Wiethoff use the term *trusting belief* instead of trust [193]. Along with their argumentation of trusting beliefs, the result of a trustworthiness assessment conducted by the trustor is not the actual trustworthiness of a trustee but the *perceived trustworthiness*. In the end, a trustor usually cannot know, due to the uncertainty of relationships and interactions, whether a trustee truly is trustworthy and acts accordingly to one's expectations [203].

The process of trust-building accompanies the formation of relationships [193]. With increasing trust, relationships deepen and vice versa. A relationship usually begins with *initial trust*, also called *swift trust* and evolves to *knowledge-based trust* during time [261]. The fundamental factor of initial and knowledge-based trust is the experience made with a trustee. At the first encounter, the trustor cannot assess the trustworthiness of the trustee based on previous experiences and existing knowledge base. The trustworthiness assessment relies on a first judgment of the cues presented by the trustee. The first judgment of the cues may lead to inferences about personality traits or expectations about potential future interaction outcomes, whereupon initial trust emerges [220].

Usually, initial trust relies on the following three cognitive categorization processes, which are i) reputation categorization, ii) stereotyping, and iii) unit grouping [220]. Reputation categorization respects second-hand information about attributes or behaviours of the trustee that positively impact their perceived trustworthiness [17]. Stereotyping describes the stigmatization of other parties due to perceived cues or attributes during the first interaction (e.g., appearance, gender, age, voice) or by

previously received second-hand information. Stereotyping relies on general biases or prejudices that have formed before the interaction with the specific party [154]. Concerning unit grouping, the trustor believes to be part of one group or community with the trustee. Being part of the same group leads to the conclusion of shared goals, values, and beliefs, which in turn fosters the perceived trustworthiness of a trustee.

As the number of interactions increases, the trustor experiences the past and current performances of the trustee. At that point, the trustor establishes a knowledge base about the trustee and, thus, develops knowledge-based trust. Previous trustworthiness assessments and assumptions about attributes that positively impact trustworthiness have been confirmed in previous interactions. Thereby, a trustor can better foresee the actions of the trustee in specific situations [194]. Compared to initial trust, knowledge-based trust is more stable in terms of performance lapses of circumstance changes [193].

Fostering relationships has many benefits, such as cohesion in an interpersonal context, profit in a commercial context, or user engagement in a technology context [86, 135, 270]. Therefore, it is decisive for trustees to be perceived as trustworthy and prove one's trustworthiness to the trustor in interactions. On these grounds, it is in the trustee's interest to signal cues and attributes relevant to the trustor's trustworthiness assessment. Thereby, relationships can develop from initial trust to more stable knowledge-based trust. Even if a trustor might not have initial trust in a trustee but is obliged to interact with the trustee for some reason, a trustee can prove one's trustworthiness by complying with the trustor's positive expectations and, thus, create knowledge-based trust [15]. Therefore, trustors must carefully assess the trustee's trustworthiness since the trustee may tend to act accordingly to the trustor's values and expectations even though they might not correspond to the ones of the trustee [235].

2.2 Software Development Life Cycle and Requirements Engineering

When developing software, the procedure usually follows a software development life cycle (SDLC) [267]. A SDLC is a process for software developers or designers to plan,

create, test, and deploy a software system. It is also known as systems, application, or product development life cycle. In the past, multiple SDLCs have been introduced that have been adapted to various development styles or team structures or enrich former SDLCs with more detail to single SDLC phases [267]. Yet, SDLCs can usually be traced back to the waterfall model, also known as the cascade model, which is a linear, sequential way of software development. Other SDLCs are iterative or hybrid procedures that include feedback loops between SDLC phases. Yet, SDLCs usually cover six phases that are based on the waterfall model, which are 1) planning and analysis, 2) design, 3) development, 4) testing, 5) deployment, and 6) maintenance [19, 267, 188].

In the planning and analysis phase, software developers analyse the problems which are addressed or appear in the context of the software to be developed. For that purpose, they closely work together with involved stakeholders facing the problems, such as end-users or business stakeholders. After obtaining an understanding of the relevant problems, the software engineers decide on the goals that counter the problems and that the software needs to fulfill. In the course of understanding and addressing relevant problems, the planning and analysis phase covers large parts of requirements engineering. Requirements engineering can be defined as “the process of eliciting stakeholder needs and desires and developing them into an agreed-upon set of detailed requirements that can serve as a basis for all subsequent development activities” [153]. Therefore, some researchers hold the opinion that requirements engineering is the most important area for software engineering [58].

In the planning and analysis phase, the requirements engineering activities requirements elicitation, requirements analysis, and requirements specification are conducted. Requirements elicitation describes the process of identifying the stakeholders’ needs and wants of the software to be developed. Common methods for requirements elicitation are user interviews and user surveys [331]. On this basis, software engineers specify software behaviour in form of software requirements. Software requirements are distinguished into functional and non-functional requirements [153]. Functional requirements “describe the behavioral aspects of a system” [10]. Usually, they are defined by means of use cases that describe in what way users interact with the software. In contrast, non-functional requirements are qualities that software should meet or constraints that should be avoided regarding the design and operation of the software [110]. Examples of non-functional requirements

are properties such as performance or usability.

After requirements elicitation comes requirements analysis. Software engineers examine the interrelations and conflicts between the requirements of the different stakeholders and try to resolve the conflicts. As soon as the requirements are so far settled, software engineers move to requirements specification. Requirements specification describes the activities of documenting requirements in formal language by written or graphical models [1].

Another activity of the planning and analysis phase is to check on the software's feasibility. A feasibility analysis may concern amongst others the costs of developing and maintaining the software, the revenue, or the ability of the developers to realize the specified goals and requirements.

After the planning and analysis phase follows the design phase. The design phase encompasses procedures for defining solutions to the analysed problems from the previous planning and analysis phase. The design phase includes, for example, algorithm design like determining a programming language, software architecture design such as deciding on software components to be included, and graphical user interface design like what software features to include. The next phase is the development phase, which is also known as the implementation phase. In this phase, the defined requirements and design specifications are converted into an operational application by writing and compiling programming code. After the software engineers have implemented the software, the testing phase starts. The testing phase is also called the verification or validation phase. It covers practices for checking whether the programmed software from the development phase meets the requirements and specifications from the planning and analysis phase and the design phase. In the context of requirements engineering, the testing phase includes requirements validation. Only if the requirements are validated, the requirements specification from the planning and analysis phase becomes officially valid. Furthermore, the testing phase involves identifying and resolving bugs in the code. The fifth phase is the deployment phase. It marks the process of delivering the software to the intended end-users. The delivery may encompass a whole application or parts of the application, like single software features. The last phase of SDLCs usually is the maintenance phase. In the maintenance phase, deployed software is modified by correcting errors, refining output, or enhancing performance or quality.

2.3 Software Features and Digital Nudges

The specification of software features is part of the design phase that is followed by requirements elicitation and specification from the planning and analysis phase of a software development life cycle. Software feature can be described as “a unit of functionality of a software system that satisfies a requirement, represents a design decision, and provides a potential configuration option” [11]. Software features consist of a set of core, mandatory assets, and variable, optional assets [189]. As components of a software feature, assets in combination create a feature. Thereby, software features be adapted and tailored to specific domain scenarios and are, thus, highly variable and reusable depending on their asset composition [11, 189]. As software features are “a prominent or distinctive user-visible aspect, quality, or characteristic of a software system” [162], they need to be implemented in the user interface at some point. This concerns the design, interaction, and information aspect of features in the front-end while features also hold a code aspect in the back-end of systems.

An example of a software feature is the filtering feature in online dating [303]. Its requirement is to filter users to a subset of users by characteristics selectable by the user. The benefit of the filtering feature is that it enables the user to quickly find other users that match the user’s interest. From a design perspective, the filtering feature can be realized by checkboxes or a search field for characteristics that the user can enter freely. Moreover, configuration options for the filtering feature can be for example the characteristics by which users are filtered into subsets of users, such as certain hobbies, smoking habits, or personality traits. The design and configuration options already represent feature assets, whose composition results in a tailored software feature.

In line with the definition of software features are digital nudges. Digital nudges can be regarded as a special category of software features from the domain of persuasive technologies and soft paternalistic interventions. Persuasive technologies try to change the users’ attitudes, behaviour, or both without constraining user action [101], while soft paternalistic interventions use information to guide users to safer and better choices for their good [3]. On this basis, digital nudges can be defined as user-interface design concepts that use “information and interaction elements to guide user behaviours in digital environments, without restricting the individual’s

freedom of choice” [225]. Freedom of choice can be realised by an open choice architecture, which allows users to choose options without making use of deception or coercion [238]. As a tool of soft paternalistic interventions, digital nudges aim to change user attitude, user behaviour, or both to guide choices that are benefiting the user [3, 101]. In addition, digital nudges can be used to increase user awareness in specific areas [3].

To develop digital nudges and enable behavioural change, the Fogg Behavioural Model identifies three requirements [101]. First, a digital nudge needs to be designed in a way that encourages the motivation of users. Second, a digital nudge must consider the users’ ability to perform the targeted behaviour. Third, a trigger needs to be implemented that incentivises users to show the targeted behaviour. All three requirements need to be considered simultaneously within the system. Moreover, best practices have revealed that behavioural change can be realized by digital nudges that provide certain forms of content or information, such as explanations of behaviour patterns and solutions for unfavourable behaviour [225, 298].

Nudge catalogues provide software engineers with overviews of reusable digital nudges to support software development. These are for example the model for the design of nudges (DINU model) [225] or the nudging design principles [294].

2.4 Method for Systematic Analysis of Trustworthiness Requirements

The method for systematic analysis of trustworthiness requirements by Mohammadi et al. is a requirements engineering method for developing trustworthy cyber-physical systems [229]. By trustworthy cyber-physical systems, Mohammadi et al. refer to software systems that realize software qualities, run reliably, and are trustworthy to users for these reasons. The method for systematic analysis of trustworthiness requirements serves as a basis for the method to elicit trust-related software features presented in Chapter 4. The steps of its top-down approach for specifying trustworthiness requirements are briefly described in the following.

The method for systematic analysis of trustworthiness requirements consists of four steps, which are i) obtaining trust concerns, ii) specifying trustworthiness

goals, iii) eliciting trustworthiness requirements, and iv) determining trustworthiness properties. The first step describes the starting point of the requirements elicitation method with the concerns stakeholders have regarding the system to be developed. The concerns are called trust concerns, as they reduce trust in the software. The method considers stakeholders involved in the business process of the software. Based on identified concerns, the second step is to derive trustworthiness goals. Trustworthiness goals are the objectives of the stakeholders in the context of their trust concerns. They are to be achieved as software goals by the system. The trustworthiness goals serve as a basis for deriving trustworthiness requirements. Trustworthiness requirements describe capabilities the system should meet or conditions it should enable. They relate to functional software requirements and realize trustworthiness goals. As the last step, the trustworthiness requirements are related to trustworthiness properties. Trustworthiness properties describe how the requirements can be realized in the business process of software development in form of qualities. They describe the ways in which user trust in the system can be positively influenced and relate to non-functional requirements.

The method for systematic analysis of trustworthiness requirements can be applied as a model-based approach by using goal modelling and business process modelling. Thereby, trustworthiness goals and stakeholder activities can be presented in the context of business processes for software development.

2.5 Risk

Risk is a variable that usually is considered for decision-making when the knowledge of consequences is limited [304]. Different from uncertainty, for risk, the probabilities associated with the possible outcome of the given choice options are assumed to be known. Therefore, risks can usually be weighed against each other to decide on the choice option whose risks are optimally less severe and less probable.

According to the ISO 31000 standard for risk management, risk can be expressed by i) the risk source, ii) potential events, iii) their consequences, and iv) the probability [146]. A *risk source* describes an element from which a risk may arise. In the context of this work, the risk source is CMI use. The risk source for CMI users can be concretised into other CMI users, the CMI service provider, or the CMI applica-

tion. Due to CMI use, *potential events* – positive or negative ones – can evolve. This work focuses on potential negative events, which correspond with the trust concerns of CMI users, such as interacting with a scammer on social media [272]. Those negative events might involve *consequences*, which can be called unwanted incidents [cf. 83]. An unwanted incident concerning the potential negative event of interacting with a scammer could be a financial loss. Unwanted incidents can be further analysed by their *probability* and severity [83]. Regarding the financial loss due to scammers, data about this topic is needed to evaluate its probability and severity. The Federal Trade Commission of the United States reported that in 2021 more than 95.000 Americans lost a total of \$770 million by scammers on social media, which is 18 times higher than in 2017 when 5.000 people lost about \$42 million [100]. In 2021, there were 296,48 million social media users in the United States [289]. Based on these numbers, the probability of financial loss by scammers on social media is 0.03% and, thus, classified as rare. Regarding the severity, the subjective evaluation of the author here is that the severity of financial loss depends on the amount of lost money. It may involve psychological damage or financial distress which is why the severity is estimated to range from moderate to major.

As discussed above, risk is a useful decision determinant. Based on risk, decision options can be prioritized in a ranking order [65]. Risk has been used as a deciding determinant in various fields, such as in software risk management [232], governmental safety decisions [247], or in product development [61]. To make use of risk as a decision determinant, it first needs to be evaluated by a risk assessment. According to the ISO 31000 standard, a risk assessment is a systematic, iterative, and collaborative process in which the knowledge of involved stakeholders is considered [146]. Risk assessment describes a procedure consisting of the three phases i) risk identification, ii) risk analysis, and iii) risk evaluation.

Risk identification describes the process of finding, recognizing, and describing risks. It further specifies sources of risk, their causes, emerging threats and chances, indicators that suggest risks, the value of involved assets, consequences like unwanted incidents, assumptions, beliefs, limitation of knowledge, and time-related factors. Moreover, the risks are set in relation to each other. Risk identification is important for understanding risks, their cause, and their consequences. Previous research proposes a multitude of techniques for this process, for example, brainstorming or checklists [59].

		Severity				
		<i>Insignificant</i>	<i>Minor</i>	<i>Moderate</i>	<i>Major</i>	<i>Catastrophic</i>
Probability	<i>Rare</i>				Financial loss by scammers	
	<i>Unlikely</i>					
	<i>Possible</i>					
	<i>Likely</i>					
	<i>Certain</i>					

Figure 2.1: Risk matrix with the example of “financial loss by scammers” that is evaluated with by a medium risk value.

The next phase is risk analysis. Risk analysis involves quantitative or qualitative approaches to determine the severity and probability of unwanted incidents. Probability can be reported by the absolute or relative frequency of occurrence of an unwanted incident. Based on these values, probability can be further assigned to categories such as rare, unlikely, possible, likely, or certain [83]. Severity is based on an evaluation of the problematic nature of the unwanted incidents for the involved stakeholders. Severity can be classified into the categories insignificant, minor, moderate, major, and catastrophic [83]. Based on the probability and severity evaluation, unwanted incidents can be mapped on a risk matrix, as is exemplarily presented in Figure 2.1. The probability categories rare to certain represent the lines of the matrix lines while the severity categories insignificant to catastrophic represent the columns. In the risk matrix of Figure 2.1, the unwanted incident “financial loss by scammers” is included in the field where the probability category rare and the severity category major meet. When weighing the severity categories moderate and major for this unwanted incident, the worst-case scenarios are assumed to cover the complete severity range of the unwanted incident.

The last step of the risk assessment is risk evaluation. For that step, a risk matrix is a supportive tool to visualize the defined risk acceptance level. The risk acceptance level is defined by software engineers to decide on a threshold of which risks need to be considered during software development. In the case of risk matrices, the risk acceptance level is included by colouring the matrix fields, as can be seen in Figure 2.1. Green fields mark the risk acceptance level of an unwanted incident as acceptable, yellow fields as critical, and red fields as unacceptable. By this procedure, unwanted incidents can be ranked regarding their acceptance level showing

the prioritization of what unwanted incidents software engineers should deal with first.

2.6 i* Goal Modelling Notation

The i* framework is a goal modelling language for the early phase of software development. It is used by software engineers to understand the problem domain about who does what and why [327]. As an actor-oriented approach, the i* notation allows modelling information systems in an organisational environment with heterogeneous actors. Software engineers can depict the goals of the various actors and what tasks they undertake to achieve them. Thereby, the intentionality of actors becomes apparent. In terms of software development, i* goal modelling is a valuable approach to requirements engineering. By goal modelling, it can be specified what requirements are contributing to the achievement of software goals.

The i* goal modelling notation is based on the work of Yu [327]. i* goal models consist of two parts - the Strategic Dependency (SD) model and the Strategic Rationale (SR) model. The SD model focuses on involved stakeholders, who are called actors, and the dependencies between them. The SR model depicts the intentionalities of the single actors in a given context. While the SR model shows the interdependent external world of involved actors regarding specific use cases, the SR model presents the internal intentionalities of the actors.

In the following, the elements of the i* notation are defined. The graphics are taken over from Paper 5 [39] in this chapter. The appearance of the elements is depicted next to the explanation of their functionalities. The focus in this section lies on the elements that are relevant to the CMI context and the TrustSoFt method introduced in this work (see Chapter 4). Additional elements can be looked up in the iStarWiki of the RWTH Aachen [305]. Examples of complete i* goal models are presented in Chapters 6 and 13.3.

Stakeholders aka actors. As mentioned before, i* goal models provide the possibility to model the goals and dependencies of different stakeholders, who are known as actors. In the context of CMI, actors are usually the end-user, the

application, and the service provider. To model trust concerns that relate to issues between users, usually, two end-users need to be modelled. Below, the relevant elements for the actors are described and depicted in Figure 2.2.

Actors. As active entities, actors achieve their goals by applying their knowledge and skills. Actors can be individuals, technologies, or organisations.

Actor Association Links are used to model the relationship among actors. There are six associates in the i^* goal modelling notation. The *is-part-of-association* points out components that are part of a whole. Each component is considered to be an intentional actor. The *ISA-association* describes an actor that is a specialised form of another actor. The *plays-association* shows a role an actor has. The *covers-association* emphasises a position of a role. The *occupies-association* describes that an actor occupies a position. Last but not least, the *INS-association* is used to represent an instance of a more general entity. In the context of CMI, the application can be regarded as an instance of the CMI service provider. The INS-association is represented by an arrow that is labelled with "INS" and points from the application to the service provider. It is used in the exemplary goal models in Figure 6.2 on page 82.

Actor Boundary. Actor boundaries are added to a goal model when the SR model is created. An actor boundary partly encloses the actor as depicted in Figure 2.2 and includes all intentional elements (see next paragraph) attributable to the respective actors inside. This means that the size of the actor boundary adapts to the size and number of the intentional elements of an actor.

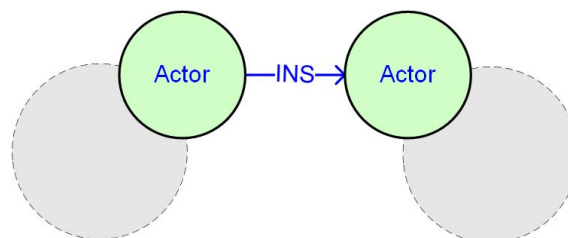


Figure 2.2: i^* notation for actors, the actor boundary, and the INS-relationship link.

Intentional Elements are used to model the intentionalities of actors. Intentionalities can be expressed by goals and soft goals, tasks, resources, and beliefs. They are depicted in Figure 2.3.

Goals are a state of the world that an actor desires to achieve. As they specify a state to be achieved, it is about the what and not about the how. Goals can be evaluated by objective criteria whether they are achieved or not.

Soft goals. Like goals, soft goals describe a state of the world desired by an actor. In contrast to goals, soft goal satisfaction depends on the subjective evaluation of the respective actor.

Tasks are behavioural procedures or activities that are carried out by actors.

Resources. Physical or informational entities are modelled as resources. Once modelled, they are considered available or existent.

Beliefs. In a strict sense, beliefs cannot be classified as intentional elements, because they do not represent an actor's intentionality but a condition an actor holds to be true. Beliefs usually describe the context of intentionalities. Therefore, they are nonetheless categorized as intentional elements.



Figure 2.3: i^* notation for the intentional elements goal, soft goal, task, resource, and belief.

Dependency links model dependencies between two actors. Depending on the direction in which the dependency is modelled, one actor is the depen-
 dee while the other is the depender. The depen-
 dee depends on the depender to
 either achieve a goal (goal dependency) or a soft goal (soft goal dependency), to
 fulfill a task (task dependency), or to receive a resource (resource dependency).
 Both actors need to cooperate to realise a dependency. The i^* notion for
 dependencies is depicted in Figure 2.4.

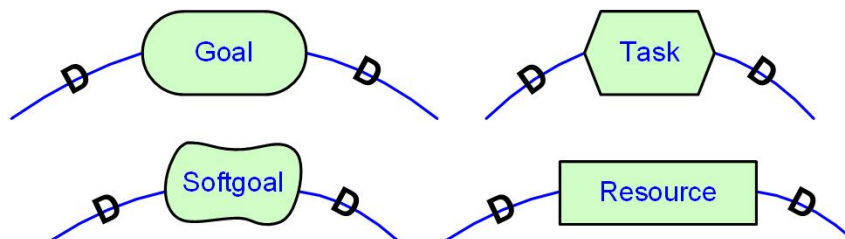


Figure 2.4: i^* notation for the dependencies.

Intentional Links put intentional elements in relation to each other by linking them. Thereby, intentional elements can be modelled in a structure in which elements depend on one another. The different types of intentional links are explained below and depicted in Figure 2.5.

Means-end links are used to demonstrate a means to achieve an end. The direction of the link is from the means to the end.

Decomposition links decompose tasks into sub-elements, such as sub-goals, sub-soft goals, sub-tasks, or resources. Decomposition links can be distinguished into AND-, OR-, or XOR-decompositions. Thereby, the logical need of a set of sub-elements is modelled. The AND-decomposition denotes that all sub-elements must be accomplished. The OR-decomposition gives the choice to determine which and how many sub-elements need to be realised. The XOR decomposition compels engineers to choose only one of the decomposed elements. The head of the decomposition link is modelled next to the parent task. Depending on the type of decomposition, the link is labelled with either "AND", "OR", or "XOR" next to its head. If a decomposition link does not have a label, it is an AND-decomposition by default.

Contribution Links connect intentional elements with soft goals. They express in what way an element contributes to a soft goal, which can be both positive and negative. Positive contributions are *make* and *help* links. Negative contributions are expressed by *break* or *hurt* links. Make and break links show that an element completely satisfies or denies a soft goal. Help and hurt links depict elements that either have a positive or negative effect on a soft goal, but are not sufficient enough to satisfy or deny it.

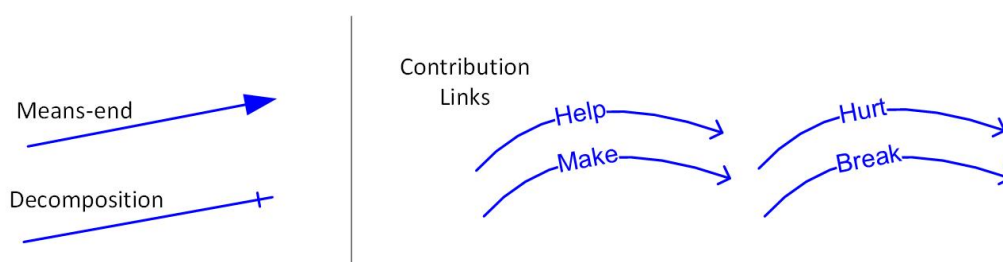


Figure 2.5: i* notation intentional links.

2.7 Feature Models

In software development, feature models are used to compactly represent software features for Software Product Line Engineering [189]. Software Product Line Engineering refers to methods, tools, and techniques for creating sets of software systems that share core features and address various market segments [250]. The principles of Software Product Line Engineering are the reusability and variability of software features so that software systems do not need to be built from scratch. Therefore, feature models serve as an appropriate technique, as they enable reusability and variability by modelling sets of features for tailoring software products to user needs or application scenarios. The resulting software is called a software product line [11].

Feature models originate from the Feature-Oriented Domain Analysis by Kang [162]. Kang introduced feature models for domain engineering, which describes processes of reusing domain knowledge (i.e. knowledge of a specific field) for software development [147]. In this context, feature models are used to consider the end users' view on implemented software requirements in form of software features.

Feature models are based on the composition of various feature assets that together form a complete software feature. Feature models are organized in the form of hierarchical tree diagrams, whose leaves represent a feature asset each. Depending on the reference, the term “feature asset” is also used synonymously with “feature” and “feature asset”. In this work, the term “feature asset” is used for the leaves underneath the root of a feature model. At the root of a feature model is the so-called *concept feature*. It represents a whole class of solutions. The concept feature has the highest degree of abstractness and is decomposed in the following layers of the tree structure. With increasing tree layers, feature assets become more and more concrete. When refining a feature asset, the refined feature asset is called the parent feature in regards to the resulting feature asset which is called the child or sub-feature of the parent feature. All feature assets are specified in natural language as keywords.

Software product lines are configured based on the relationships between parent and child features. Relationships are modelled by links, which are the tree branches, between the features. By the links, the configurator learns which feature assets must be included in a software product line and which feature assets are optional

contributing to the variability of a software product line. Figure 2.6 depicts the feature model notation presenting the links.

In general, links either mark feature assets as mandatory or optional. Links can refer to single feature assets or to a set of sub-features. A *mandatory link* can either be modelled by a simple line or by a line with a filled bullet to the sub-feature. All feature assets at the end of the link must be included in a software product line. In contrast, an *optional link* for a single feature asset is a line with an empty bullet at its end. The configurator has the choice of whether to include the respective feature asset in a software product line or not. Regarding the optionality for a set of sub-features, the links emanating from a parent feature are either connected by a filled semi-circle at the top to demonstrate *OR-links* or are connected by an empty semi-circle representing *XOR-alternative-links*. OR-links denote that the configurator needs to include at least one of the sub-features in the software product line. The XOR-alternative-link means that the configurator is only allowed to select one of the sub-features for the software product line. In addition to the mandatory and optional links are links for cross-tree constraints. A dashed arrow is a *requires-link* that refers to another feature that must be added as well if the feature pointing at it is included in the software product line. In contrast, the *excludes-link* is symbolized by a double-sided dashed arrow. Feature assets that are connected by an excludes-link cannot be part of a software product line at the same time. To demonstrate how to use the feature model notation, Figure 2.7 shows an exemplary feature model that presents features of an online dating application.

The concept feature of the feature model from Figure 2.7 is the *online dating application*. For the online dating application, it is mandatory that it consists of

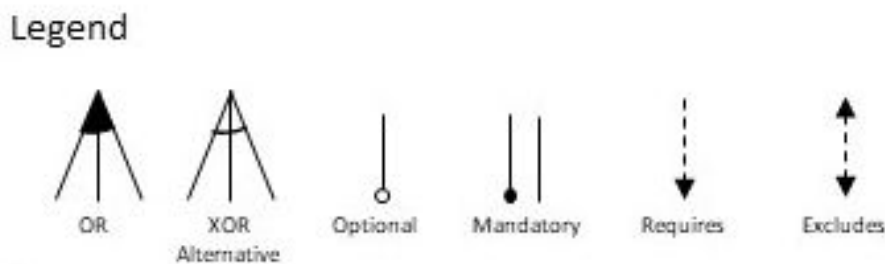


Figure 2.6: Overview of the feature model notation

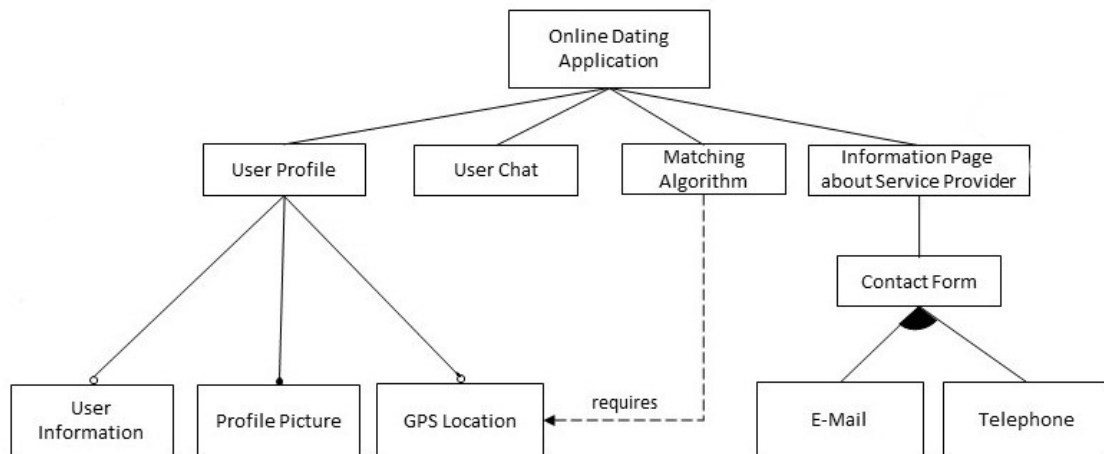


Figure 2.7: Exemplary feature model about for an online dating application.

the assets *user profile*, *user chat*, *matching algorithm*, and *information page about service provider*. The user profile is further refined into the mandatory feature assets *user information* and *GPS location*. The GPS location is mandatory because the matching algorithm requires it. For the user profile, a *profile picture* is yet optional. Furthermore, the information page about the service provider is refined into the feature asset *contact form*. The configurator can decide whether the contact form is based on e-mail, telephone, or both.

In addition to the general feature model notation, researchers have extended the notation by additional elements. An example is the cardinality-based feature model [78]. The cardinality-based feature model notation includes the multiplicity of feature assets to the models as it is known from UML-diagrams [288]. The multiplicity is used to limit the number of feature assets in a software product line. It is modelled by adding $[n,m]$ closely above a feature asset. n is the lower bound and m is the upper bound that can be replaced by numbers to specify the number of possible feature asset clones in a software product line. If m is maintained, it means that a feature asset can be included as many times in a software product line as desired by the configurator. Other extensions added extra-informational elements to the feature assets such as so-called “attributes”. Attributes are linked by a dotted line to a feature asset and can be descriptive information about the feature asset, mathematical formulas, or numerical values.

3

Building a Trust Framework for CMI

Paper 1 describes trust in the context of CMI. Trust in CMI is characterized by the three CMI parties i) user, ii) service provider, and iii) application and the user's interplay with them. In CMI, four forms of trust are involved. As CMI is used for connecting with strangers, users' development of *interpersonal trust* with other users is in the foreground. Interpersonal trust describes a trust relationship between two individuals [265]. However, in CMI, trust-building between two CMI users is mediated by the CMI application online. Therefore, this work names interpersonal trust established via a CMI application *computer-mediated interpersonal trust*. When users decide to continue their interaction offline, computer-mediated interpersonal trust shifts to interpersonal trust. The impulses from direct interaction extend the knowledge base people have from one another. The newly established interpersonal trust may confirm the existing computer-mediated interpersonal trust to expand the level of trust or oppose what has been learned about the other person online and diminish the trust level.

When interacting with a CMI application, the user additionally builds *system trust*. System trust refers to the confidence in technical systems to successfully deliver the promised service [203, 217]. To successfully deliver a promised service, it is, on the one hand, about a running, error-free system. On the other hand, it is about the quality of the service for the purpose for which the user uses the system.

Most often, the choice of a specific CMI application is encouraged by already existing *brand trust*. Brand trust describes the trust that a person has built in an organisation that sells products or services [320, 80]. Usually, the organisation distinguishes itself from other sellers by building a brand through unique selling points or organisational attributes or values [164]. In this work, brand trust denotes the trust a user has in a CMI service provider. As an example, online dating

applications like Tinder¹ or Bumble² have succeeded in creating a well-known brand around their application [145]. Brand trust is characterized by its two dimensions i) reliability and ii) intentions to i) have the (technical) competence to keep promises and satisfy customers' needs and ii) to be benevolent and supportive by the means of the customers' interests and welfare [80]. Brand trust can further be impacted through the usage of the CMI application, as the application is the technical product by which a service provider demonstrates its competence and service to the users.

In the following subsection, the trustworthiness framework of CMI explains the fundamentals of the users' trustworthiness assessment of the three parties by which computer-mediated interpersonal trust, system trust, and brand trust are established. The framework leaves out interpersonal trust that is relevant during offline encounters with CMI acquaintances since the CMI application does not directly mediate the trust-building anymore then. The second subsection of this chapter further defines trustworthiness facets, which are an essential part of the trustworthiness framework for CMI and for users' trustworthiness assessment.

3.1 The Framework of Trustworthiness for CMI

The framework of trustworthiness for CMI is introduced in Paper 1. It presents an approach to how computer-mediated interpersonal trust, system trust, and brand trust can be considered in and reflected by CMI applications. By the framework of trustworthiness, software engineers learn how to enable users in these trust-building processes via the CMI application. The framework of trustworthiness for CMI is depicted in Figure 3.1.

On the left side of Figure 3.1, the three trust types introduced in Section 3 are presented as child categories of trust. Trust is included in the framework since the CMI system needs to enable trust-building on its platform to provide users the service of introducing strangers online. As described in Chapter 2.1, trust arises when the trustor, in this context the CMI user, perceives cues and traits of the trustee. Thereby, CMI users can conclude whether the trustee is trustworthy. In the case of CMI, the role of the trustee can be taken by other CMI users, the CMI

¹www.tinder.com

²www.bumble.com

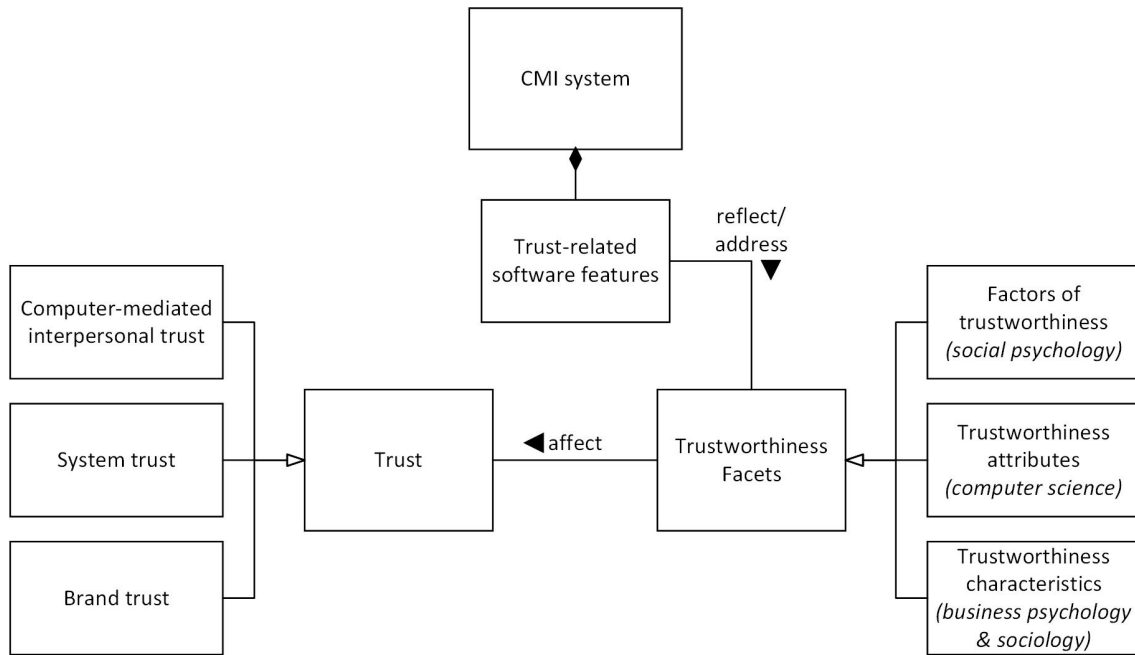


Figure 3.1: The framework of trustworthiness for CMI.

system, or the CMI service provider.

Previous research identified qualities of trustees that positively impact their (perceived) trustworthiness when assessed by the trustor (see right side of Figure 3.1). In terms of interpersonal trust, which is mediated by the CMI system during online interactions in this context, Mayer et al. introduced the *factors of trustworthiness* [213], which are ability, benevolence, and integrity. Ability denotes the competence or skill of the trustee to perform as expected by the trustor. Benevolence means the goodwill of the trustee about the trustor and their intention. Lastly, integrity describes shared principles or norms by the trustor and trustee. The factors of trustworthiness originate from the discipline of psychology and have been used or adapted for trust research by many other disciplines, as well.

Concerning system trust, Mohammadi et al. conducted a literature review for software qualities that foster the trustworthiness of information systems [230]. They defined them as *trustworthiness attributes*. Examples of trustworthiness attributes are privacy and usability. Privacy describes how far the system provides users visibility and control over their private information. Usability reflects users' ease of operation within a system and to interpret it. Trustworthiness attributes stem from the discipline of computer science.

In addition to the factors of trustworthiness for interpersonal trust and trustworthiness attributes for system trust, research has revealed further characteristics associated with the trustworthiness of organisations and brands. As an example, Delgado et al. identified [81] *fiability* and *intentionality* as trust-building. *Fiability* is a term introduced by Delgado et al. [81], which reflects an individual's belief about a brand's compliance with the given promises, such as ability or values. *Intentionality* refers to the goodwill of the brand to not misuse the consumers' vulnerability. Research has identified many more characteristics that foster (perceived) the trustworthiness of brands and organisations. As they are not covered by one term, they are included in the framework as *trustworthiness characteristics*. Trustworthiness characteristics stem from economics, business psychology, or sociology.

All three types of trustworthiness traits are relevant in CMI. They need to be considered within the CMI system to allow users trust-building with the three CMI parties via the platform. For this purpose, this work consolidates the three types of trustworthiness traits to the term *trustworthiness facets*. If trustworthiness facets are possessed by a CMI party, they positively impact its trustworthiness. If users perceive facets as available, they increase the perceived trustworthiness of the respective CMI party, even though the party may not possess the specific trustworthiness facets. Subsection 3.2 describes the trustworthiness facets in more detail.

For trust-building, the framework of trustworthiness for CMI proposes that the trustworthiness facets are addressed or reflected by software features within the CMI system. These software features are accessible concepts in the user interface [162] by which CMI users can perform a trustworthiness assessment to evaluate whether the respective CMI party possesses relevant trustworthiness facets and is thus trustworthy in specific situations. Therefore, the software features are called *trust-related software features*. A more detailed definition of trust-related software features is presented in Chapter 9. Chapter 4 explains how software engineers can elicit trust-related software features in a structured way for the development of software applications.

3.2 Trustworthiness Facets

In the previous section, trustworthiness facets are introduced as part of the framework of trustworthiness for CMI. They encompass the qualities of users, software applications, and service providers that the CMI system needs to address or reflect in its user interface to enable users' trustworthiness assessments. In this section, trustworthiness facets are defined in more detail. They are originally introduced in Papers 1 and 2.

Trustworthiness facets are desirable characteristics that encompass qualities of the CMI parties, which are positively related to their trustworthiness. By trustworthiness assessments, CMI users try to evaluate the degree to which facets seem to be present at the CMI parties. For that purpose, users assess perceivable cues by which they infer a facet and its extent. This process may involve the cognitive categorization processes explained in Chapter 2.1 for initial interactions or rely on an existing knowledge base regarding more advanced relationships. However, a trustworthiness assessment is a subjective evaluation [193]. Users cannot assess the existing trustworthiness but the *perceived trustworthiness* of another party. The accuracy of the assessed perceived trustworthiness to the actual trustworthiness depends on different factors. One factor is the users' appraisal skill, which depends on their core competencies of cue perception, understanding the cues, managing internally what has been understood, and making use of the obtained understanding for their intentions [212, 223]. Especially online, users may misinterpret cues and falsely interpret trustworthiness facets, as they are not used to an online trustworthiness assessment [44]. The online trustworthiness assessment depends on the CMI system and its software features, which may convey a distorted view of parties depending on their design and use by the parties. The use by the parties is another factor for complications in trustworthiness assessments. CMI parties may manipulate online cues in order to present themselves in a way that leaves a positive impression on other people. This process is called impression management [184]. Impression management might involve over-expression or even misrepresentation of trustworthiness facets so that established computer-mediated interpersonal trust may be rooted in false conditions.

Despite their complexity, trustworthiness assessments are an essential approach for users to check whether risks associated with CMI use are relevant to the inter-

action with a specific CMI party. Circumstances in which CMI users have concerns regarding interaction with another CMI party are introduced as an *initial state* in Paper 2. Here, it is renamed to *vulnerable state*. Vulnerable states are characterized by problematic characteristics causing a trust conflict with the involved CMI party. Problematic characteristics may relate to the specific situation or interacting parties. A problematic characteristic concerning the trustee could be for example the dishonesty of another user when the user discovers a lie about personal information, such as age. This can cause the user to be concerned about a misrepresented dating profile. Most often, vulnerable states involve a mismatch in the being, attitudes, values, or behaviour of a CMI party with the ones of the users. Usually, relevant trustworthiness facets are not available. As a consequence, CMI users feel vulnerable to the other party. For users, it seems unlikely that the involved CMI party complies with their intention or desire for an interaction outcome.

The contrary to the vulnerable state is the *goal state*. Goal states are CMI interactions that do not hold any trust concerns. Trust concerns do not exist because CMI users assess the perceivable trustworthiness facets of the other party as promising to meet their positive expectations of an interaction outcome. In that case, the concerns about potential risks associated with the interaction become irrelevant for the users. For example, online dating users depend on the skills of an online dating application to find a suitable partner. If the online dating application displays the trustworthiness facet "ability" in terms of its matching algorithm, for example by explaining the algorithm's functioning, CMI users may perceive the application as trustworthy in case the explanation is convincing. It is in the interest of the software engineer to develop systems that make users aware of vulnerable states and try to provide goal states for successfully providing a service. Goal states and vulnerable states are relevant for selecting trustworthiness facets for software development, which is explained in Chapter 3.2.2.

In the context of software development, trustworthiness facets are qualities of the three CMI parties that the software engineer needs to consider in software design. The software design should enable the three CMI parties, in the role of trustees, to (optimally) truthfully represent their trustworthiness facets in a manner that is perceptible to CMI users. Thereby, CMI users can perform their trustworthiness assessments as trustors. For the users and service providers as trustees, the software design should reflect their trustworthiness facets to the users. In terms of the

software as a trustee, the trustworthiness facets are relatable to non-functional requirements. Non-functional requirements are elicited qualities of a software system that the system should realize or address [190]. They can be distinguished into execution and evolution attributes. While execution qualities, such as usability or safety, are observable during run-time, evolution qualities, such as maintainability or reusability, describe the structure of a system [315]. This work considers evolution qualities as given since they contribute to a running, error-free system, that allows a basic trust in the system. The focus for software development here is on execution attributes that impact the user experience, which exceeds the basic functionality for providing the service. In this way, the system can prove to its users its quality in providing the service in their interest. As for users and service providers, the software design should reflect the trustworthiness facets of the system. In addition, it can be designed in a way coherent with its trustworthiness facets.

As an example, a warning message can be designed according to the trustworthiness facet “benevolence” when the wording of the warning message is formulated in a benevolent way. Another way to reflect benevolence within a system for example is to inform users in a notification about benevolent actions performed by the application, such as encryption for users’ privacy. In both cases, users can assess the benevolence of software. How to use trustworthiness facets for developing software applications is further explained in Chapter 4.

3.2.1 The Overview of Trustworthiness Facets

Paper 2 provides an overview of the trustworthiness facets for individuals, organisations, and technologies that research has already identified. The overview is created based on a structured literature review following the guideline of “Preferred reporting items for systematic review and meta-analysis” by Moher et al. [233]. The objective of the literature review was to detect papers that include variables that positively impact the trustworthiness of or trust building with at least one of the three parties. Researchers have identified such a positive relationship by either statistical calculations, theoretical derivations, or qualitative methods. Following the guideline by Moher et al. [233], the literature review passes through the phases of literature identification, screening, eligibility, and inclusion.

For the literature identification, the databases Scopus and Web of Science were

considered for keyword search. Only papers from research journals or conferences published in English were considered. The keyword search included relevant keywords, for example, “trustworthiness”, “trait”, or a CMI party. 264 papers were identified, which were reduced to 234 after removing duplicates. The papers were then screened if they fit the definition of trustworthiness facets. A set of 126 papers remained. The eligibility check considered whether the included literature considers trustee types compatible with the social media and CMI context. Trustee types were for example individuals in various roles (e.g., user, consumer), organisational structures such as institutions, companies, and service providers, and technologies like software applications, websites, or platforms. From the 126 papers, 26 were excluded, resulting in a literature review based on 100 papers.

The result of the structured literature review is an overview of trustworthiness facets that holds a total 163 facets - 68 for individuals, 55 for technology, and 40 for organisations. The overview is organised into three parts. One table contains the facets for individuals, one those for technology, and another one the facets for organisations. The overview of the trustworthiness facets for individuals, technologies, and organisations is presented in the Appendices A, B, and C.

In the overview, trustworthiness facets have been grouped regarding their similarity in their semantics. The grouping is reasonable, because some trustworthiness facets appeared multiple times in the literature review. While some trustworthiness facets have been used by the same term but divergent definitions, others share the same definition but differ in their terminology. For each group of trustworthiness facets, a definition is formulated that comprises the definition of the single facets.

3.2.2 The Guideline for Selecting Trustworthiness Facets

Trust is a highly context-dependent concept [168]. Although all trustworthiness facets contribute to the trustworthiness of a party and thereby to trust in the party, it depends on the specific situation, perceived problem, or existing concern, which trustworthiness facets are especially relevant for a trustor to be available. For user-centered software development, which oftentimes considers specific user scenarios as an initial point of development, it is thus beneficial to select trustworthiness facets that are in particular significant for a scenario [29]. For that purpose, the overview of the trustworthiness facets can serve software engineers as a database for selecting

facets that especially address users' pain points of a trust concern (see Appendices A, B, and C).

Paper 2 further provides a guideline of how relevant trustworthiness facets can be selected. The process is coherent with the techniques of requirements elicitation [331]. Practitioners may rely on user surveys, user interviews, expert opinions, or literature.

The guideline for selecting trustworthiness facets is presented in Figure 3.2 by a UML activity diagram [288]. It is based on concepts of design thinking. Design thinking implies creative procedures, which can be used for IT development to identify (digital) solutions for social needs [255]. Design thinking first analyzes the problem space to then examines creatively the solution space. The guideline for identifying trustworthiness facets relates to problem analysis. In the following, the guideline is explained step by step using an example.

The first step of the guideline for selecting trustworthiness facets is to specify the problem that shall be addressed by the software to be developed (see Figure 3.2, 1.). A problem may relate to the concerns of users that compromise the users' trust in another party. In Chapter 4, this is understood to mean a trust concern. In dependence on the specific problem or users' trust concern, relevant trustworthiness facets can be selected from the overview of trustworthiness facets. In the domain of online dating, an example of a user's trust concern is catfishing [206]. Catfishing describes the act of online dating users (also called catfish) to create an online dating profile that represents another identity for deceptive or fraudulent purposes [278].

After problem specification, software engineers need to attain a deeper understanding of the problem (2.). They can proceed in several ways, such as consulting literature or talking to experts or those affected [82]. Usually, attaining an understanding of a problem involves the definition of its context, involved stakeholders, and a cause for existence [82]. Since problems involve different aspects depending on their nature, the outcome of obtaining an understanding can vary depending on the problem. Software engineers decide which extent of knowledge is sufficient for the subsequent development process. Usually, knowledge about the problem supports engineers to identify adequate solution approaches in the later development process. In the case of catfishing, knowledge about its motivation can aid engineers to warn users. Furthermore, knowledge about the circumstance when users identify

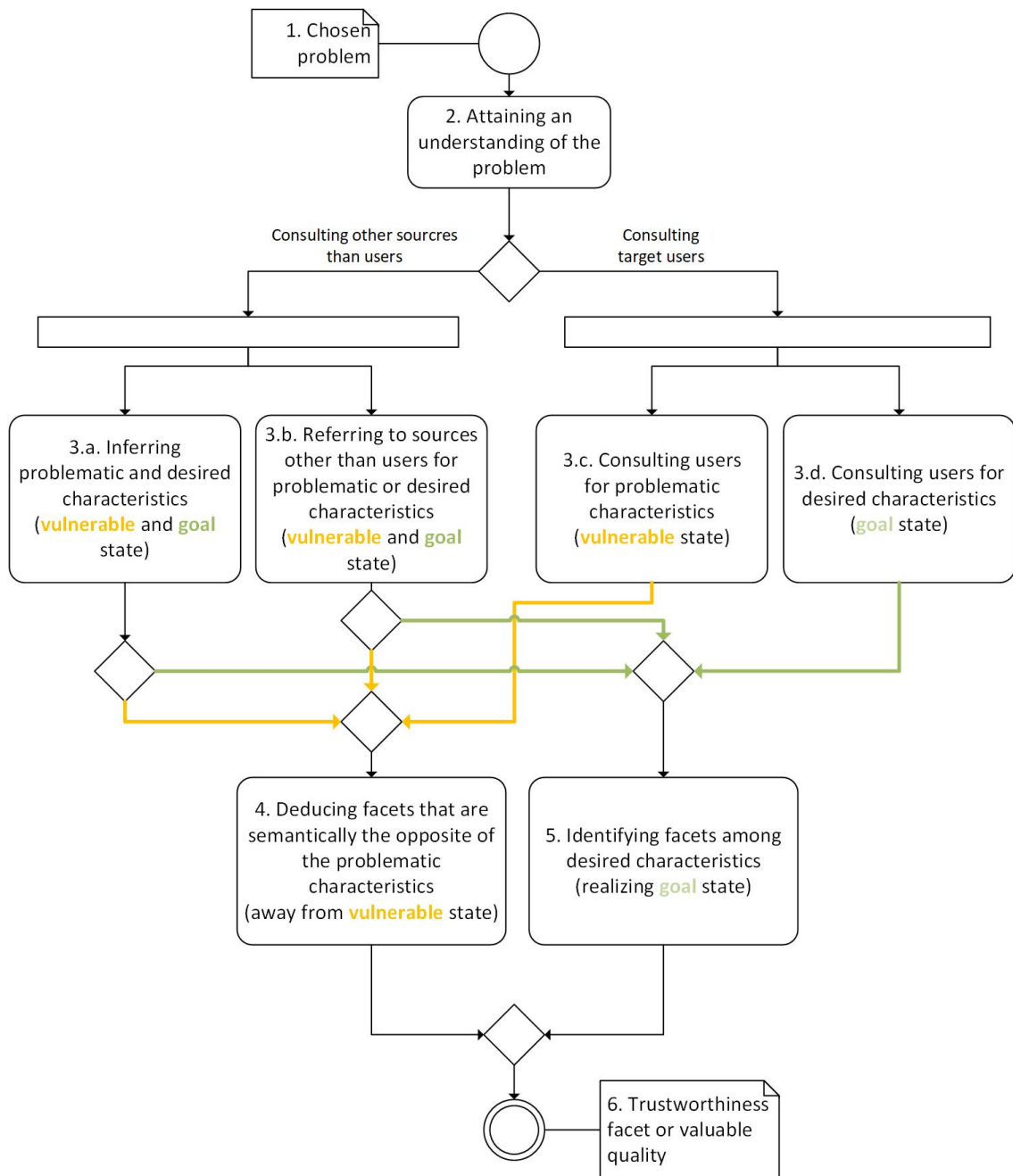


Figure 3.2: Guideline for selecting trustworthiness facets from the overview of trustworthiness facets by consulting users, experts, or literature. The yellow arrows show the guideline for identified problematic characteristics of the vulnerable state, while the green arrows present the paths for the desired characteristics of the goal state.

catfish can enrich a warning. Simmons and Lee found that catfish are motivated by boredom, monetary fraud, or to secretly check out online dating and its users [278]. Victims reported having discovered catfish during offline encounters or when catfish refuse video chats.

After engineers better understand the problem, they can either identify trustworthiness facets by consulting targeted users of the application to be developed or relying on other sources, such as literature or experts. The objective of consulting users, experts, or other sources is to identify desirable or problematic characteristics. By these, trustworthiness facets can be inferred from the overview of trustworthiness facets.

Interviewing target users (3.c. and 3.d.) is a purposeful way to elicit requirements because users can directly report their needs and pain points [275]. Both the needs and pain points might shed light on the desired and problematic characteristics. Asking general questions about the needs and pain points first is important to receive an unbiased answer from users. Concerning the catfishing example, questions could be “What do you wish for in an online dating application so that you are less concerned about catfish?” or “How would you detect a catfish?”. Afterwards, questions can target the desired and problematic characteristics more directly. An exemplary question is “With what characteristics would you describe a catfish?”. Resulting problematic and desired characteristics from user interviews can be compared with the overview of trustworthiness facets - either with the semantic opposite of problematic characteristics (4.) or directly with the desired characteristics (5.) to result in confirmed trustworthiness facets or valuable characteristics for software development (6.).

However, sometimes an engineering project lacks resources (e.g., money, time, ability), which is why it is not possible for engineers to ask target users. Thus, it is fundamentally important to take other sources into account as well - even if users have been consulted. Since user statements describe a subjective view, engineers should consider other sources to cover alternative perspectives. The subjective view of users is susceptible to leaving out relevant aspects of a problem. Omitting aspects of a problem may happen because they are not relevant for an individual, the individual is not aware of them, or the individual does not consciously want to report them due to personal reasons [275].

When consulting sources other than users, engineers may start with their own conclusions (3.a.) that they derive from their previously acquired knowledge (2.). To draw conclusions, engineers need to elaborate on the vulnerable and goal state of a problem. Discussions and brainstorming sessions in the development team support this process. In this step, the focus is on identifying desired or problematic characteristics or trustworthiness facets by imagining what characteristics users need the various parties to have or not to have. The overview of trustworthiness facets may serve as a supportive tool in this step.

In addition to drawing conclusions, additional sources can be consulted, such as experts, media reports, or scientific literature (3.b.). They may directly name trustworthiness facets, problematic characteristics, or desired characteristics. If not, additionally gained knowledge from these sources can serve as a further basis for drawing conclusions about desired or problematic characteristics.

As this work does not focus on catfishing, asking users about relevant trustworthiness facets concerning catfishing is out of the scope. Therefore, this work relies on drawing conclusions and the knowledge of other sources. For drawing conclusions, the definition of catfish is considered to identify problematic characteristics. Catfish are characterized by a divergence in the user expectations of their identity. Most often, users do not expect a catfish behind a profile, which is why a catfish is *unpredictable* for users. Unpredictability is identified as a problematic characteristic of the vulnerable state. Another problematic characteristic is proposed by Schulman, who hosted the MTV television show “Catfish” that has coined the term. Schulman characterizes catfish by *dishonesty* [274]. Since catfishing is a problematic phenomenon in online dating, most existing literature analyses problematic characteristics. Therefore, in this small example, the focus is on problematic characteristics than on identifying desired characteristics.

After problematic and desired characteristics have been extracted, the overview of trustworthiness facets can serve as a database for facet selection. In terms of the problematic characteristics (see Figure 3.2, follow the yellow arrows), engineers need to consider their semantic opposite and whether it is relatable to one of the CMI parties user, service provider, or application depending on the context of the problem. Engineers shall check if the opposite attribute is listed in the overview (4.). If so, a trustworthiness facet has been identified (6.). If not, the derived attribute is not a scientifically proven facet. Nonetheless, it may be a valuable quality for CMI

development. Concerning the desired characteristics (see Figure 3.2, follow the green arrows), software engineers can compare them directly with the trustworthiness facets in the overview (5.). Viewing the overview can also result in trustworthiness facets, or the desired characteristics are valuable qualities for the CMI system to be reflected in its design (6.). In the catfishing example, two problematic characteristics have been identified in the previous steps: unpredictability and dishonesty. The semantic opposite of the two, and thus desirable characteristics are predictability and honesty. Both are part of the overview of trustworthiness facets for individuals. Therefore, predictability and honesty are confirmed trustworthiness facets.

4

TrustSoFt - A Method for Eliciting Trust-Related Software Features

Taking into account the importance of trustworthiness for CMI systems, the question is how software can be built that supports its users in their trustworthiness assessment. As a solution approach, Paper 3 introduces the method for eliciting trust-related software features (TrustSoFt). It is a requirements elicitation method that guides software developers to structurally develop such systems. This method is suitable for software engineers, who not only aim to support users in their trustworthiness assessment but also target to reduce usage risks for safer software use. Although the context of the method is mainly CMI, it is applicable to other system developments in which the trust-building of end-users plays a crucial role. TrustSoFt can be applied for developing software applications from scratch but also to improve already existing applications.

TrustSoFt is a user-centered method that is based on the method for systematic analysis of trustworthiness requirements by Mohammadi et al., which has been introduced in Chapter 2.4 [229]. It extends the method of Mohammadi et al. by the framework of trustworthiness (see Chapter 3.1). While the method of Mohammadi et al. is for building trustworthy cyber-physical systems, TrustSoFt aims for software that enables the psychological process of trustworthiness assessments in the digital sphere. Such systems shall enable the trustworthiness assessment of the three CMI parties i) user, ii) service provider, and iii) application. TrustSoFt is a user-centered method because it addresses user concerns and aims to mitigate them. By considering user pain points like user concerns, usable software can be developed [276].

TrustSoFt is an iterative, top-down method. It consists of five overarching steps,

which are depicted in Figure 4.1. Despite its iterative approach, the individual steps of TrustSoFt can be applied repetitively as new insights are gained throughout the whole method. In this respect, several iterations are most promising to achieve a set of trust-related software features. The steps of TrustSoFt are explained in the following. They are accompanied by the catfishing example that has been already introduced in the previous section 3.2.2. Exemplary results for the catfishing concern are depicted in Table 4.1 on page 53. Further application examples are given in Chapters 12 and 13.

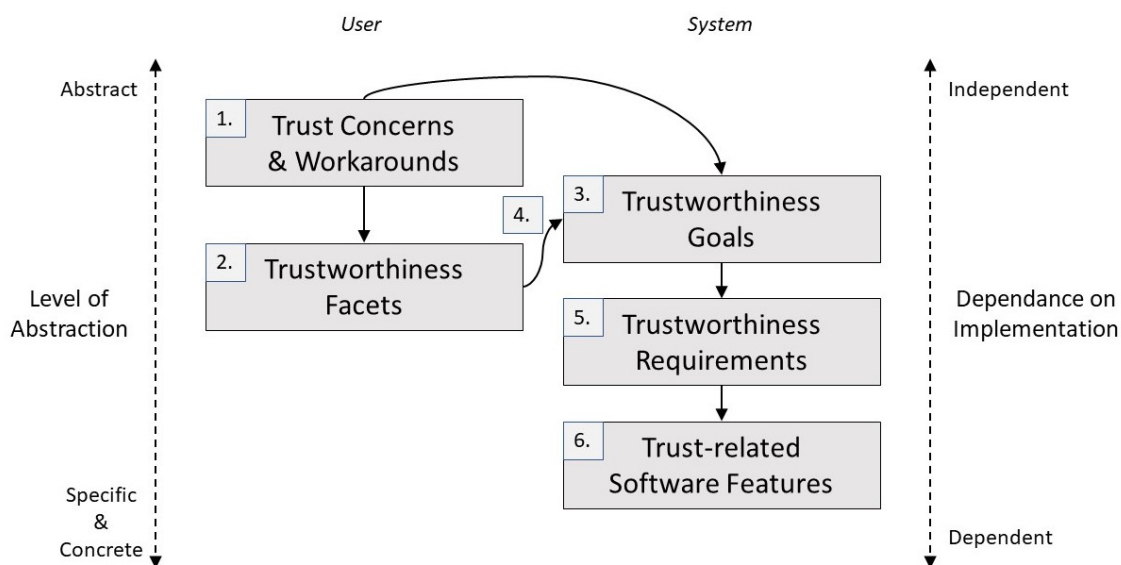


Figure 4.1: Conceptual TrustSoFt Method

Step 1: Identifying Trust Concerns and Workarounds. The first step of TrustSoFt is to identify users' trust concerns associated with software use and related workarounds. Trust concerns describe a subjective fear of individuals about undesirable scenarios to occur when using the software [231]. These scenarios usually involve interactions with other parties that are associated with one's vulnerability and negative outcomes. Although trust concerns are uncertain to happen during interactions with other parties, they impact people's trust-building to them [317]. If trust concerns are not addressed by the respective software, users apply workarounds. Workarounds are goal-driven, behavioral adaptations in form of improvisations or strategies [8]. They bypass, minimize or overcome obstacles like trust concerns that hinder individuals from achieving a personal goal. The knowledge of users' trust concerns helps software engineers to address and counter their concerns to increase

user satisfaction and safety [254, 116]. Knowledge about workarounds supports software engineers to directly be aware of usability deficiencies, which can be picked up by developing adequate software features to address system deficiencies. To identify trust concerns and workarounds, requirements elicitation techniques such as user interviews and focus groups can be conducted [331].

For TrustSoFt, trust concerns and workarounds are input from target users. They are on an abstract level and independent of software implementation (see Figure 4.1).

An example of trust concerns in online dating is catfishing, as mentioned before. To check the authenticity of online dating profiles and whether the other users are catfish, a workaround is to search other users on social network sites (e.g. Facebook, LinkedIn) [242]. When the other person has a profile on a social network site that matches what users have learnt about the person during online dating, trust concerns of catfishing are reduced.

Step 2: Determining Relevant Trustworthiness Facets. The next step is to specify trustworthiness facets that are especially relevant for each of the identified trust concerns from the previous step. Trustworthiness facets can be specified by the guideline introduced in Chapter 3.2. Most often, when conducting a user survey about trust concerns, users report trustworthiness facets explicitly or implicitly with their concerns, as they are highly related to each other. On that level of abstraction, trustworthiness facets sketch a trust concern and its resolution in more detail. Therefore, trustworthiness facets can be derived from trust concerns. Trustworthiness facets are still abstract but more specific for software development than trust concerns and workarounds. They are the link between users and software, insofar as users report facets that serve as input for software engineers to realize them in the software to be developed. Therefore, trustworthiness facets are oriented towards the middle of the independence scale for implementation.

In terms of the catfishing example, the application of the guideline for selecting trustworthiness facets has revealed *predictability* and *honesty* as relevant trustworthiness facets for CMI users when users are concerned about catfish (see Chapter 3.2.2).

Step 3: Deriving Trustworthiness Goals As the software to be developed aims to mitigate trust concerns, the next step is to define coherent trustworthiness goals. Trustworthiness goals are software goals that correspond to the objectives of users. The user objectives either contrast with the trust concerns or resolve them [228]. Software engineers can derive trustworthiness goals from trust concerns. For that purpose, natural language is used. By defining trustworthiness goals, the software to be developed is tuned to take care of its users so that they do not face their trust concerns.

Trustworthiness goals are an abstract element that is specified for the development of the system. It is independent of the implementation process.

For the catfishing concern, the user objective is to only interact with people whose profile presents their actual identity. Based on the user objective, the trustworthiness goal for the online dating software could be for example *user authentication* for all profiles on the platform. User authentication describes the software process of checking on user authenticity. User authenticity denotes that users present their true identity in their profile. In this case, true identity means the correct representation of personal information like name, age, gender, or job [191]. With the trustworthiness goal of user authentication, the online dating application tries to detect catfish. Another trustworthiness goal could be *catfish banishment*. After users have been distinguished between those with a true and fake identity, the online dating application could remove the ones with a presented fake identity from the service.

Step 4: Facet Allocation to Trustworthiness Goals. In addition to the specification of trustworthiness goals, the identified trustworthiness facets are allocated to the specified goals. The allocation ensures the reduction of trust concerns and aims at supporting trust building. Software engineers must determine which trustworthiness facets are represented or realized by which trustworthiness goal. One goal can be related to n facets. Allocating facets to goals addresses the assumption that the better the facets are considered in the software, the better the trustworthiness can be assessed so that the less relevant trust concerns will be. Through the allocation, software engineers are guided in the later process of TrustSoFt in which sense of each facet software features should be developed or designed.

Concerning the catfishing example, the facets predictability and honesty of CMI users can be realized by the trustworthiness goal “user authentication”. When users are proven to be authentic by the system, CMI users can assess that other users may behave accordingly to the expectations formed by the given online dating profile. Furthermore, users have proof of the honesty of other users concerning their identity. In regards to the trustworthiness goals “catfish detection” and “catfish banishment”, the selected trustworthiness facets for CMI users are not fitting. The two trustworthiness goals refer to behaviour of the online dating application, while the trustworthiness facets describe trustworthy users. Instead, the goals should reflect the trustworthiness facets of the application or the service provider, who represents oneself via the application. Since there are no facets identified for these two parties yet, Step 2 of trustworthiness facet selection needs to be repeated in the context of the two goals as “chosen problems” (see Figure 3.2). Software engineers need to analyze how the application and the service provider should be - meaning what trustworthiness facets they should possess - concerning catfish detection and catfish banishment so that they are trustworthy in these contexts.

Step 5: Specifying Trustworthiness Requirements. After software engineers have specified the trustworthiness goals, they can derive trustworthiness requirements. From one goal, a multitude of requirements can emerge that all contribute to achieving the trustworthiness goal. Trustworthiness requirements are functional software requirements. They describe the behaviour or capability of a system so that users benefit from it [190]. In addition, they are specified to address trust concerns and realize the trustworthiness goals in terms of the trustworthiness facets allocated to them. From all allocated trustworthiness facets of a goal, it is upon the software engineer to decide which facets are realized by which requirement. In the end, all trustworthiness facets from a trustworthiness goal need to be realized by at least one requirement.

Trustworthiness requirements are formulated in natural language. Since they describe system behaviour, it is recommended to formulate requirements as an activity the system should perform. Software engineers need to reflect on what behaviour is useful to achieve a trustworthiness goal. Furthermore, the workarounds that have been identified with the trust concerns in the first TrustSoFt step may provide indications of what requirements a system needs.

With the trustworthiness requirements, the solution approach is concretised and the dependence on the implementation increases. Trustworthiness requirements are an element to be realized in the system.

Regarding the catfish example, software engineers need to specify trustworthiness requirements for the trustworthiness goal “user authentication” which is related to the trustworthiness facets predictability and honesty. Thinking of the workaround to look up other users on social network sites, a trustworthiness requirement could be *to calculate the similarity of profile information (e.g. name, age, job) of the online dating profile and the social network profiles*. By *depicting the similarity result*, users can assess the honesty of other users for user authentication. The first requirement for calculation is indirectly involved with the trustworthiness facet honesty. It is a prerequisite for users’ trustworthiness assessment of the facet honesty, because it enables the second trustworthiness requirement of displaying the similarity results. Regarding predictability, the online dating application could *ask users that have dated the person already, whether learnt information during the date confirmed the information learnt during online interaction*. By *displaying the match of online and offline experiences other users had with the respective user*, other dating users can derive another user’s predictability. Similar to the previous two trustworthiness requirements, the requirement for asking other users about their offline and online experiences is a prerequisite for the requirement to display the results. The first requirement thus is a prerequisite for the trustworthiness facet predictability to be evaluated by the users in their trustworthiness assessment.

Step 6: Deriving Trust-Related Software Features. The last main step of TrustSoFt is about the derivation of trust-related software features from the trustworthiness requirements of the previous step. Software features are user-accessible concepts within software [140]. They realize functional and non-functional requirements. In the case of TrustSoFt, the functional requirements are the trustworthiness requirements. The non-functional requirements are the trustworthiness facets that are assigned to the trustworthiness requirements. Trust-related software features need to realize trustworthiness requirements while simultaneously considering assigned trustworthiness facets. Addressing trustworthiness facets as non-functional requirements by the software features is one reason why the software features are trust-related. The other reason is that trust-related software features are designed to support users in their trustworthiness assessment. Trust-related software features

are defined in more detail in Chapter 9. The chapter also includes information about the development and configuration of trust-related software features.

Trust-related software features are determined in natural language. They describe how the trustworthiness requirements and facets are designed in the front-end of a system. When formulating software features, software engineers can specify involved information, interaction elements, or design elements. Thereby, software engineers specify what information software features process or disclose to the user, how the interaction with the software feature is possible, and how a software feature shall look like. Additionally, software engineers can determine underlying algorithms necessary for feature implementation as well.

The level of detail in which trust-related software features are described is up to the software engineer. The higher the level of detail, the more concrete the software feature for an implementation. Nevertheless, a structure for specifying trust-related software features is proposed in Chapter 9. The structure concretizes trust-related software features for implementation as a continuation of TrustSoFt for the next phase of the Software Development Life Cycle. Within TrustSoFt, trust-related software features are at the most concrete abstract level. Their dependence on implementation is high.

For the catfishing example, there are four trustworthiness requirements that could be identified in the previous step for the trustworthiness goal “user authentication”. Two requirements are related to the trustworthiness facet honesty and two to the facet predictability. In the following, the software features for the requirements of the facet honesty are further elaborated. Table 4.1 includes the results of every TrustSoFt step for the trustworthiness goal “User Authentication”. Exemplary software features for the requirements of the trustworthiness facet predictability are also added in Table 4.1. For the trustworthiness facet honesty, the two previously elicited trustworthiness requirements are i) calculating the similarity of profile information (e.g., name, age, job) of the online dating profile and the social network profiles and ii) depicting the similarity result of the profile comparison. To realize the first requirement, a software feature can be an algorithm that checks whether the profile entries about the name, age and job of the online dating profile and social network profiles match. The output of the algorithm could be a percentage of the matching entries. However, this requirement can only be realized when the system has access to the users’ social network profiles. At that point, it becomes apparent that a

trustworthiness requirement is missing, which covers this need. Therefore, a new trustworthiness requirement has to be specified. This procedure demonstrates the repeatability of the single TrustSoFt steps if necessary. The new trustworthiness requirement can be something like *asking users for access to their social network profiles*. An option how to realize this requirement is by a software feature such as buttons in the users' profile setting. The buttons could be labelled with "connecting with social media", which is a call for action and an information element. The design of the buttons could be in form of various social media logos (design element). By clicking on them, a pop-up window could open for agreeing to link the social network sites with the online dating application by logging in (interaction element). With these two trustworthiness requirements as prerequisites, the third requirement to display the similarity result of the profiles can be realized. The requirement could be realized by a software feature that visualizes the percentage result of the similarity algorithm in the profile of online dating users. Next to the percentage could be written "profile information match with other social media profiles" (information element). The design could include a pie chart - one part coloured green for the matching information, the other part coloured red for mismatches (design element). When clicking on the pie chart, an explanation of the similarity algorithm could appear (interaction element).

After software engineers have applied TrustSoFt, they can document the results as in overview tables such as Table 4.1. The overview table should include the trust concerns, trustworthiness goals, trustworthiness requirements, together with their allocated trustworthiness facets, as well as specified trust-related software features. Thereby, software engineers receive an overview of optional trust-related software features that can realize the trustworthiness requirements and trustworthiness facets. Through this form of documentation, collections of software features are created that are tailored for individual software applications. These collections can serve software engineers as a basis for software product line engineering.

Applying TrustSoFt for software development results in a multitude of software requirements and software features. This gives software engineers room to customise software products to their liking. However, the current conceptual TrustSoFt method has a few drawbacks to software development.

Trust Concerns	Trustworthiness Goals	Trustworthiness Requirements	Trustworthiness Facets	Trust-related Software Features
Catfishing	User Authentication	calculating the similarity of profile information (e.g., name, age, job) from the online dating profile and users' social network profiles asking users to connect their social network sites with the online dating application depicting the similarity result of the profile comparison	Honesty (user)	similarity algorithm
		asking users that have dated the person already, whether the learnt information during the date confirmed the information learnt during online interaction	Honesty (user)	request button in profile settings for connection with social network sites
		displaying the match of online and offline experiences other users had with the respective user	Honesty (user)	percentage of profile similarity
			Predictability (user)	calendar button in the chat window to add the time of an arranged date pop-up window when opening the application after the date with survey questions analysis of survey questions about the match of online and offline experience (statistical calculations)
			Predictability (user)	percentage of user feedback about the match of online and offline experience in user profile

Table 4.1: Exemplary TrustSoFt results for the trust concern “catfishing” and the trustworthiness goal “User Authentication”.

1. The resulting trustworthiness requirements may restrict or be in conflict with each other. As they are specified for realising their respective trustworthiness goal, they are formulated without respecting other goals. Therefore, software engineers need to evaluate which requirements to implement in the end in the system.
2. Each element of TrustSoFt is connected with the others. During the application of TrustSoFt, there is a multitude of trustworthiness goals, facets, requirements, and software features that address one trust concern. Similar to a tree structure, it starts with one trust concern as the root, whose number of branches increases with increasing branch level - meaning that a number of trustworthiness goals is followed by many more trustworthiness requirements and so on. After applying TrustSoFt, it is difficult to keep an overview of the individual element connections.
3. The benefit of customising software by software features involves the agony of choice. The agony of choice means that it is upon the software engineer to decide which software feature options suit best for the software to be developed. Therefore, feature selection needs to be guided for an optimized software product line engineering.

To address these challenges, the TrustSoFt method needs to be extended. For the first challenge, risks are included as a deciding determinant to resolve conflicting requirements. The procedure is described in Chapter 5. For the second drawback, TrustSoFt is supported by the model-based approach of the i^* notation to visualize the connections of the single TrustSoFt elements among each other. The adjusted i^* goal models and how to create them in the TrustSoFt context are explained in Chapter 6. For the third challenge, the configuration of resulting trust-related software features is addressed by extended feature models in Chapter 11.

Last but not least, TrustSoFt is a requirements elicitation method for software engineers. The way engineers work highly impacts the resulting software product [330]. Therefore, the application of the TrustSoFt method is evaluated in Chapter 7. Based on the findings of the TrustSoFt evaluation, the TrustSoFt concept is adapted to the enhanced TrustSoFt concept in Chapter 8.

5

Risk as a Determinant for Prioritisation in TrustSoFt

By applying TrustSoFt, software engineers can elicit a multitude of trustworthiness goals and requirements. All identified goals and requirements contribute to software development in the context of trustworthiness assessment. Yet, trustworthiness requirements are elicited by considering one trustworthiness goal independently of other goals and problem contexts. Therefore, it can happen that conflicts occur between trustworthiness goals and requirements derived from different goals. Such conflicts complicate the design and implementation of software.

For example, a potential conflict usually occurs between goals and requirements focusing on the topics of privacy on the one hand and self-disclosure on the other. While privacy goals and requirements aim to provide users control over their private information, self-disclosure goals, and requirements target the retrieval of personal information. In the context of this work, software engineers should specify self-disclosure goals and requirements for the sake of users' trustworthiness assessment. While both privacy and self-disclosure goals and requirements have a beneficial purpose for the trustworthiness context, it may happen that they partly cannot be addressed in an application simultaneously or are interfering with each other due to their contradicting nature. At that point, the software engineer must decide how to manage the conflict.

For that purpose, Paper 4 enriches TrustSoFt through processes for conflict identification and conflict management. To manage conflicts, the extended TrustSoFt method relies on risk as a decision determinant (see Chapter 2.5). Software engineers can refer to the involved risk of a conflict to choose the option with the smallest risk level. Moreover, risk is highly relevant in the context of trust-building, since

trust is about tolerating risks and making oneself vulnerable to them (see Chapter 2.1). Knowledge about risks is thus relevant for software engineers to consciously address them in software development. In terms of CMI use, associated risks are for example data misuse, identity theft, harassment, sextortion, or reputation damage [74].

On these grounds, integrating the determinant risk into TrustSoFt has certain benefits. By considering risk in TrustSoFt, software engineers can ...

- ... consciously mitigate risks for the users through software development.
- ... prioritise trustworthiness goals and requirements for the implementation phase of the software development life cycle.
- ... decide on which requirement or goal to implement if a conflict prevents a simultaneous implementation in the application.

In the following, the different types of conflicts that can occur are introduced. Afterwards, the new steps of the extended TrustSoFt method are explained following Figure 5.3. For each new step of the method, an example is given.

5.1 Types of Conflicts

Conflicts describe a state of two issues that clash against each other because they are either incompatible or semantically at variance [290]. Usually, conflicts occur between elements that aim to solve different problems. In TrustSoFt, elements that tackle the same problem are derived from one another, realise the same solution approach, and are thus in line. Conflicts in TrustSoFt arise between trustworthiness goals and trustworthiness requirements that propose solution approaches for different problems and trust concerns. In TrustSoFt, conflicts can occur between i) two goals, ii) two requirements, or iii) a goal and a requirement.

There are two types of conflicts to be distinguished - hard conflicts and soft conflicts. Hard conflicts occur between goals and requirements that contradict each other in a way that they are not implementable in a system concurrently. An example of a hard conflict is depicted in Figure 5.1 on page 58. The example is

explained at the end of this section. In contrast, goals and requirements that are in soft conflict can both be implemented functionality-wise. However, they interfere with each other in that the effectiveness of their purpose is limited. An example of a soft conflict is presented in Figure 5.2 on page 59 and explained at the end of this section.

Usually, conflicts in which trustworthiness goals are involved are soft conflicts. This is because trustworthiness goals are more on an abstract level and not implementable at that point compared to trustworthiness requirements that describe concrete system behaviour (see Figure 4.1 on page 46). Concerning conflicts with trustworthiness goals, software engineers should always review the trustworthiness requirements of the goal in conflict. As requirements express specified software behaviour, the behaviour might be formulated in a way that the soft conflict is circumvented. The circumvention of conflicts will be described in more detail later in Steps 5.2 and 7 of the extended TrustSoFt method.

It is efficient to deal with conflicts directly in the planning and analysis phase of the software development life cycle. Thereby, time and costs can be saved for the subsequent phases of the Software Development Life Cycle in which conflicting solution approaches would delay the time to market. Conflict management is part of the TrustSoFt method extended by risk and is presented in Chapter 5.2.

Example hard conflict The exemplary hard conflict is in the context of Trust Concerns 1 and 2 for online dating depicted in Figure 5.1. Trust concern 1 presents the worry that other users make screenshots of one’s dating profile to pass it on to other people outside the application. Online dating users have reported being ashamed when non-users learn that they are using online dating [68]. This concern is one reason why the software application Snapchat ¹ has disabled the screenshot functionality of published content. To address Trust Concern 1, Trustworthiness Goal 1 wants the application to be transparent about which user has stored profile data on their device. The ulterior motive behind Trustworthiness Goal 1 is to not limit user action in the application by prohibiting screenshots. Yet, when users know that stored data like a screenshot is reported to the involved user, they might refrain from taking one as they want to remain anonymous. Trustworthiness Goal 1 is realised by Trustworthiness Requirement 1, which is to inform users when and who

¹www.snapchat.com

has taken a screenshot from their dating profile. Thereby, the application reflects its trustworthiness facet transparency.

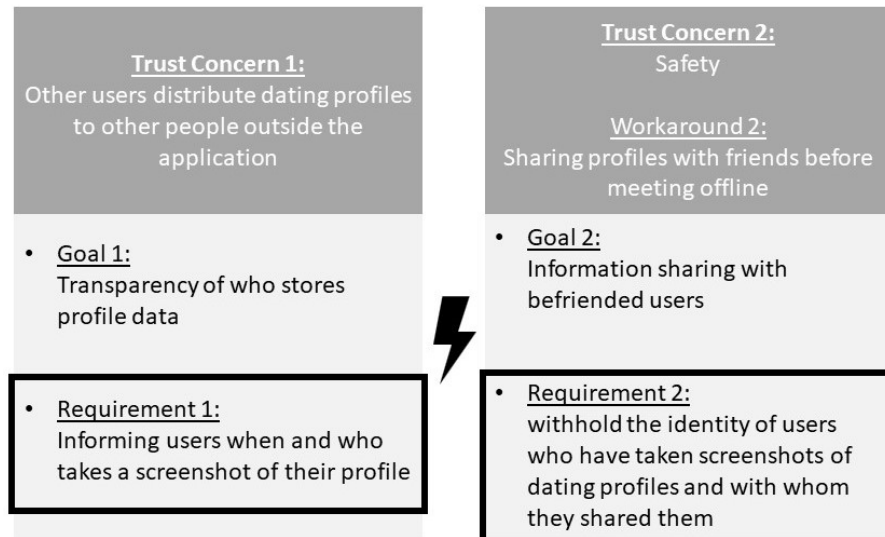


Figure 5.1: Example of a hard conflict.

In the exemplary hard conflict, Trustworthiness Requirement 2 of Trust Concern 2 is in contrast. Trust Concern 2 is about one’s safety when meeting another online dating user in the physical world for the first time. Many users take a screenshot of the dating profile to share it with friends before the date (Workaround 2) [68]. Thereby, they feel safer that in case something happens, their friends have information about the perpetrator. The application addresses the trust concern and workaround by Trustworthiness Goal 2, which is information sharing with befriended users within the application. As some users might not want the other user to know that they shared their profile with a friend and which friend has the information, Trustworthiness Requirement 2 is about keeping the identity of the respective users confidential. Thereby, the application reflects its trustworthiness facet confidentiality.

The conflict between Trustworthiness Requirements 1 and 2 is a hard conflict because informing users about the identity of who has taken a screenshot of one’s dating profile contradicts withholding the identity. Both requirements cannot be implemented in the application at the same time.

Example soft conflict The exemplary soft conflict is between Trust Concerns A and B. Trust Concern A is about catfishing. Catfishing has been introduced before in Chapter 3.2. It is the concern about profiles that represent another identity than the one using the profile [278]. Trust Concern B is about data misuse. Data misuse is the concern that the service provider, third parties, or CMI users have access to personal data and use them for their purposes [287].

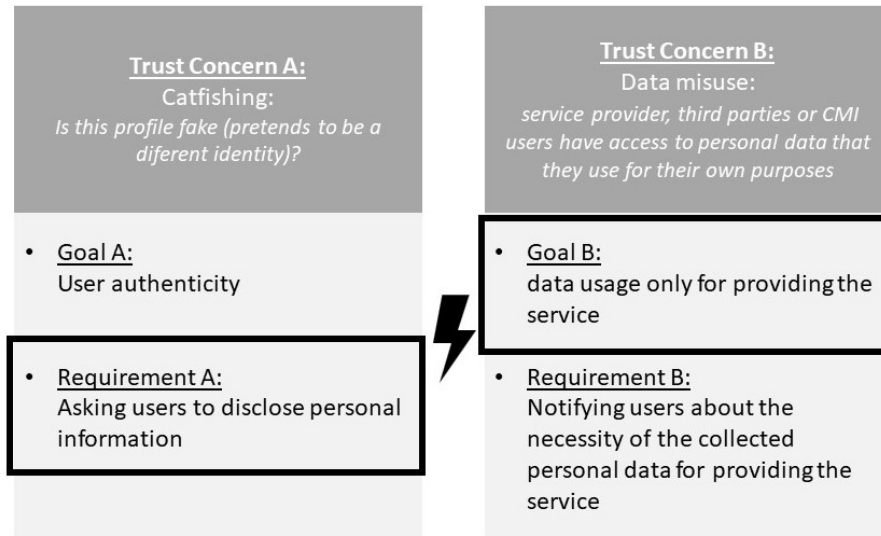


Figure 5.2: Example of a soft conflict.

To address Trust Concern A, the application aims to ensure user authenticity with Trustworthiness Goal A. User authenticity means that the user is who he/she declares to be [103]. Usually, service providers ask users for personal information to prove their authenticity by software features of user authentication [103]. Asking users to disclose personal information is thus Requirement A in this example. In the context of user authenticity, it is assumed that trustworthiness facets like user *honesty* or the application’s *ability* to prove user authenticity are involved in the trustworthiness assessment of users.

Trust Concern B, data misuse, is addressed by Trustworthiness Goal B to only use data for providing the service to which the user has agreed in the terms of service. To realise Trustworthiness Goal B, Requirement B asks to notify users about the necessity of the collected personal data for providing the service. In this context, the trustworthiness facet *transparency* of the service provider is likely to be enhanced.

In this example of a soft conflict, Trustworthiness Requirement A and Goal B are

interfering with each other. While Goal B aims to narrow data usage, Requirement A asks for more self-disclosure. Disclosing more personal information is probably not necessary in the sense of Goal B, and increases the risk of data misuse.

5.2 The TrustSoFt method extended by risk

The TrustSoFt method is extended by 1) the risk assessment of users' trust concerns to rank trustworthiness goals by their importance for the application and 2) the risk assessment of trustworthiness requirements to mitigate usage risks. Based on the prioritisation of the trustworthiness goals, software engineers can manage conflicts occurring during TrustSoFt. The extended TrustSoFt method is displayed in Figure 5.3. The original steps and procedure from Chapter 4 are depicted in white boxes and by grey arrows. The new steps are highlighted by the grey boxes and black arrows and are explained in the following.

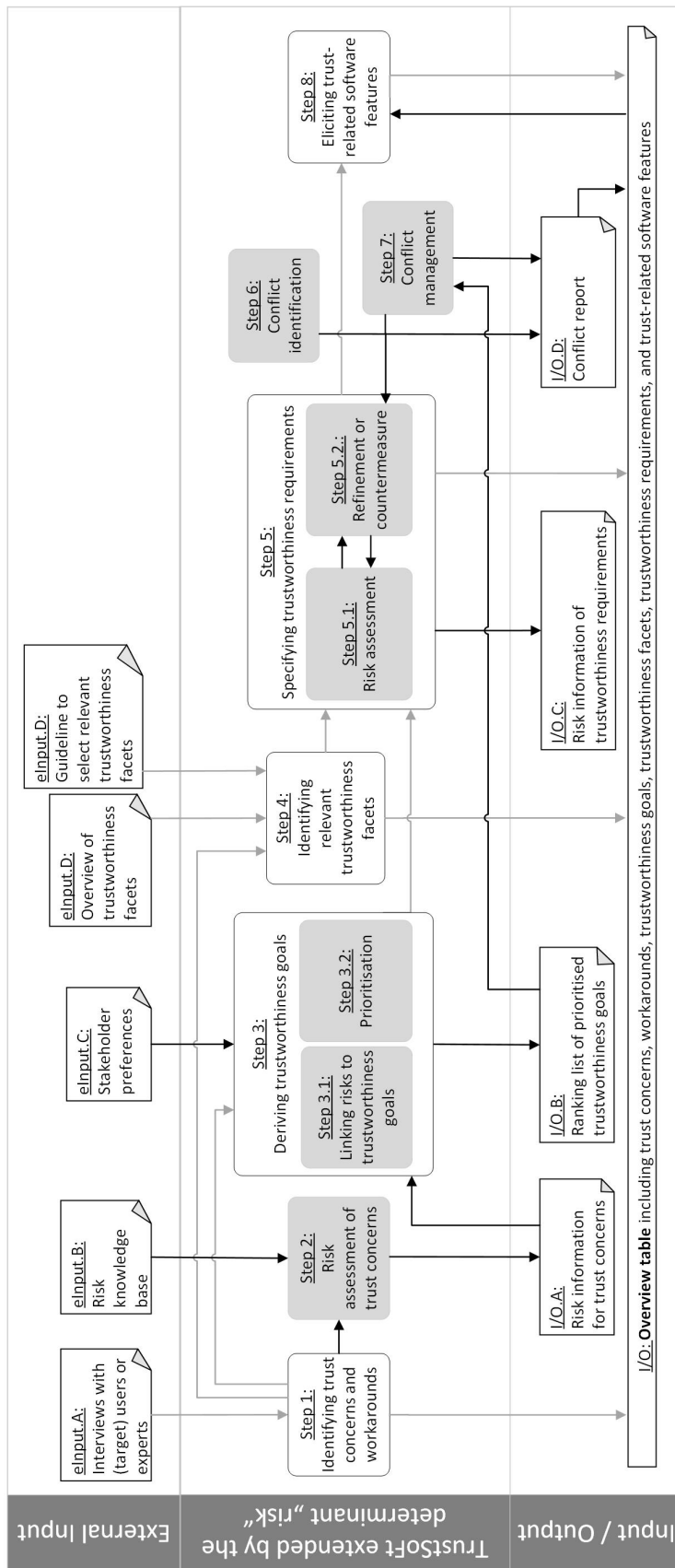


Figure 5.3: Extended TrustSoFt method by the determinant risk.

Step 2: Risk assessment of trust concerns Step 2 is the risk assessment of trust concerns, which is performed after the first step of identifying users' trust concerns and workarounds (see Figure 5.3). For each identified trust concern, software engineers shall assess how likely the concern is to happen and how severe it is. The risk assessment can be executed by having a risk knowledge base as external input (eInput.B). A risk knowledge base represents sources by which knowledge and data about risks can be obtained. Usually, risk knowledge bases can be (scientific) literature, (statistical) reports of unwanted incidents, police reports about offenses in the domain, and real data such as from social media, surveys, or interviews. Based on risk knowledge bases, software engineers can conduct risk identification and risk analysis. As a next step, the risk evaluation is conducted for determining a risk acceptance level based on the likelihood of occurrence and severity for each trust concern. The output of Step 2 is the gathered risk information (I/O.A).

For Step 2, an exemplary risk assessment is applied for the trust concerns “catfishing” (Trust Concern A) and “data misuse” (Trust Concern B) of the soft conflict presented in Figure 5.2. An unwanted incident of both catfishing and data misuse is identity theft [313, 182, 287]. In a survey by Get Safe Online ², a British internet safety website, 45% of 2,075 participants stated to have been victims of identity theft in social media in 2014 [155]. The number of recorded victims is increasing over time. Get Safe Online found an increase of 31% from 24,482 in 2014 to 32,058 in 2015 [227]. According to Javelin Research & Strategy, more than 15 million Americans had their identity stolen in 2021, where a majority of cases went unreported [52]. Together with the identities, the scammers stole up to \$52 billion [52]. Victims of identity theft suffer emotional and physical symptoms, such as depression or poor health [114]. Based on the risk analysis using these risk knowledge bases, the risk evaluation results in a catastrophic severity and possible likelihood. Therefore, the risk acceptance level for identity theft is determined as unacceptable.

Step 3.1: Linking risks to trustworthiness goals After the risk assessment of trust concerns in Step 2, trustworthiness goals are derived (Step 3) from the identified trust concerns and workarounds (Step 1) as intended by the original TrustSoFt method. The extended TrustSoFt method enriches Step 3 by the sub-steps of linking trustworthiness goals with risks (Step 3.1) and prioritising trustworthiness

²www.getsafeonline.org

goals (Step 3.2). Concerning Step 3.1, the risk information from Step 2 (I/O.A) is used to link the trustworthiness goals to those risks that the goals aim to reduce. Each risk should be related to at least one trustworthiness goal. Thereby, software engineers ensure that all risks are tackled in software development. If there are risks without an associated trustworthiness goal, engineers must derive trustworthiness goals directly from the risks in accordance with the context of the respective trust concern (Step 3).

In terms of the exemplary risk of “identity theft” from the previous step, identity theft can be linked to Goal A “User authenticity”. When an application successfully checks on user authenticity, scammers of identity theft can be identified.

Concerning Goal B, identity theft cannot be linked to it, because Goal B does not mitigate identity theft. However, as identity theft can result from data misuse, software engineers must specify a new trustworthiness goal so that for each concern all risks are covered in the software development process (Step 3). An example of a trustworthiness goal addressing the risk of identity theft and the trust concern of data misuse is “data protection”. By protecting data from being stolen, the risk of identity theft is reduced.

Step 3.2: Prioritising trustworthiness goals The prioritisation of trustworthiness goals is the second sub-step of Step 3 - deriving trustworthiness goals. The prioritisation of trustworthiness goals is relevant for the decision-making process during conflict management in Step 7. The trustworthiness goals are ranked according to their risk acceptance level of the associated risks they aim to reduce. The more unacceptable risks a goal addresses, the higher its priority is in the ranking. This ranking approach yields a preliminary prioritisation. The resulting ranking list of prioritised goals can further be refined depending on stakeholder preferences. It is up to the software engineer or the service provider to decide on the preferences in the prioritisation process. Stakeholder preferences may underlie a certain business strategy, brand image, or costs in realisation. Stakeholder preferences can further be a valuable decision factor when multiple goals share the same priority due to the same amount of associated unacceptable risks. Therefore, Step 3 has stakeholder preferences as external input (eInput.C). The output is the ranking list of prioritised trustworthiness goals (I/O.B).

In accordance with the example from the previous paragraph, Trustworthiness Goal A has a higher priority than Trustworthiness Goal B. Goal A is linked to the unacceptable risk of identity theft while Goal B is not linked to any risk in this example. In a large-scale risk assessment, the result of the prioritisation might look differently, since more risks would be identified, analysed, and evaluated than in this small application case. However, for the continuing example of the following steps, Goal A has a higher priority than Goal B.

Step 5.1: Risk assessment of trustworthiness requirements In Step 5, trustworthiness requirements are specified as intended by the original TrustSoFt method. The specified trustworthiness requirements need to realise trustworthiness goals (Step 3) while considering the realisation or reflection of trustworthiness facets (Step 4, see Chapter 4). Yet, in the extended TrustSoFt method, the specification of trustworthiness requirements is enriched by the sub-steps risk assessment (Step 5.1) and refinement or countermeasure (Step 5.2) of trustworthiness requirements. The purpose of the risk assessment of trustworthiness requirements is 1) to manage conflicts in Step 7 and 2) to mitigate risks by specifying countermeasure requirements for those requirements which are associated with risks of an unacceptable risk level (Step 5.2).

Different from the risk assessment of Step 2, which is about risks countered by trustworthiness goals, the risk assessment in this step is about risks emerging through the specified trustworthiness requirements. For each specified trustworthiness requirement, software engineers need to conduct a risk assessment. Starting with risk identification, software engineers shall identify 1 to a number of n unwanted incidents for a requirement. For risk analysis, the likelihood of occurrence and severity are specified for each unwanted incident. Thereupon, risk evaluation leads to the determination of the risk acceptance level for each unwanted incident. Based on the risk acceptance levels of all unwanted incidents that can occur through a trustworthiness requirement, engineers shall calculate a risk score. It is proposed that the risk score of a trustworthiness requirement be calculated by the mean of the risk acceptance levels of the associated risks. If the risk score of a trustworthiness requirement is unacceptable, software engineers must perform Step 5.2. Otherwise, the engineers can proceed with Step 6.

For the risk assessment, risk knowledge bases can be used as demonstrated in the

example of Step 2. However, it might be complicated to find relevant risk knowledge bases and risk data on this level of concreteness that goes along with software requirements. Therefore, the risk assessment of trustworthiness requirements is likely to rely on the expertise of the software engineers. As software engineers are experienced in requirements engineering, they are qualified to assess the drawbacks and risks of the specified trustworthiness requirements. The output of Step 5.1 is the risk information of the trustworthiness requirements (I/O.C). The specified trustworthiness requirements are documented in the overview table of TrustSoFt (I/O). Their risk scores are included in the overview table.

An example for Step 5.1 is the risk assessment of Requirement A “Asking users to disclose personal information” (see Figure 5.2). Requirement A leads users to share personal data that makes them vulnerable to data misuse [182]. Therefore, *data misuse* can be regarded as a risk of Requirement A. Another risk of personal data disclosure is the *loss of privacy*. Once users have disclosed their personal data, they cannot control anymore who has access to it [295].

Now that the risks are identified, they need to be analysed. Relying on one’s expertise, data misuse depends on additional circumstances than just asking users to disclose personal information. For example, it depends on who has access, where data is stored, and how data is protected. Therefore, the probability is rated as unlikely. Nonetheless, data misuse harms users as their data is used against their will. Thus, the severity is analysed as severe. Evaluating the risk acceptance level of data misuse for Requirement A results in an acceptable level. Regarding “loss of privacy”, this unwanted incident is rated as certain for Requirement A. Since no countermeasures are existent yet, which provide users control over their data and privacy, users cannot decide what happens to their data after the disclosure. Since the right to privacy is seen as part of human dignity and freedom [301], the severity of privacy loss is evaluated as severe. The risk acceptance level is thus unacceptable. Based on the two risk acceptance levels acceptable and unacceptable for the two risks, the risk score of Requirement A is determined as critical.

Step 5.2: Refinement or countermeasure Step 5.2 represents the refinement of trustworthiness requirements and the specification of countermeasure requirements. It is a sub-step of Step 5 and a follow-up step of Step 5.1. Software engineers perform Step 5.2 in case the risk assessment of Step 5.1 yields require-

ments whose risk score is unacceptable. For critical risks, it is up to the software engineers whether they tolerate the risk or want to reduce it.

First, software engineers should try to refine the respective unacceptable requirement. The refinement involves a re-specification of the requirement in more detail that clarifies system behaviour in such a way that the risk score is reduced. After refinement, software engineers must re-assess the risk score of the requirement by performing Step 5.1 again. The two arrows between Step 5.1 and Step 5.2 in Figure 5.3 represent that feedback loop. If a refinement is not possible or if the refinement has not changed the risk score of a trustworthiness requirement from unacceptable to acceptable, software engineers have three options.

1. The software engineers specify one or more countermeasure requirements that reduce the risk score of the problematic trustworthiness requirement. In doing so, the requirement can be implemented so that its related trust concern is covered. Yet, countermeasure requirements are mandatory to be implemented together with the respective requirement. The mandatory relationship of the requirements increases the complexity of software development. When a countermeasure requirement is specified, software engineers need to return to Step 5.1. They have to evaluate to what extent the risk score is reduced by the countermeasure requirement. If the countermeasure requirement does not reduce the risk score sufficiently, it has to be omitted from the implementation. Instead, one of the other two options can be performed.
2. Software engineers adapt the risk acceptance level. As described in Chapter 2.5, it is up to the software engineers to determine under what circumstances they tolerate risks. With an adjusted risk acceptance level, problematic requirements may be included in the application. If not, the third option should be applied.
3. The requirement with the unacceptable risk scores is not implemented in the application. Thereby, users are saved from accompanied risks. However, the trust concern that has been addressed by the omitted requirement, is consequently less well addressed by the application.

After Step 5.2 has been performed, software engineers have completed Step 5. The output of Step 5 is trustworthiness requirements and related countermeasure

requirements. Together with the risk scores, the requirements are added to the overview table of TrustSoFt (I/O).

In terms of the example in the previous step, Requirement A has a critical risk score. This means it is up to the software engineers whether they perform Step 5.2 because the risk score is not unacceptable. Yet, for this example, it is decided to reduce the risk score of Requirement A to acceptable by performing Step 5.2 on the unacceptable risk of privacy loss. As stated in the previous paragraph, Requirement A leads to loss of privacy, because no countermeasure exists yet that provides users control over their disclosed data. Therefore, a countermeasure requirement is specified for this purpose. An exemplary countermeasure requirement is for example “to let users decide for what service the disclosed data is exclusively allowed to use”. Considering the new countermeasure requirement, risk assessment is repeated (Step 5.1). The likelihood that loss of privacy occurs is now limited to the service that users allow the data usage. Moreover, it is assumed that by enabling users to control what their data is used for what service, they only agree to services where a loss of privacy is generally unlikely. Concerning the severity, nothing has changed with the countermeasure requirement. Privacy loss is still severe. Yet, with the new probability evaluation, the risk acceptance level is now acceptable. Based on this risk evaluation, the new risk score for Requirement A is acceptable.

Step 6: Conflict identification Step 6 is about identifying soft or hard conflicts. TrustSoFt elements that limit each others’ effectiveness refer to a soft conflict. TrustSoFt elements that are contradictory technical- or functional-wise refer to hard conflicts. To identify conflicts, trustworthiness goals and requirements that relate to different trust concerns must be compared pair-wise. The only exception is for trustworthiness requirements related to a countermeasure requirement. In that case, software engineers must consider countermeasure requirement(s) together with the related requirement in the conflict identification process. Countermeasure requirements can avert conflicts between the related requirement and the conflicting other TrustSoFt element.

It is upon the engineer’s expertise to evaluate the impact and drawbacks of TrustSoFt elements on other elements. Identified conflicts are documented in a conflict report (I/O.D) by jointly including the conflicting TrustSoFt elements and their status of a soft or hard conflict.

Examples of conflicts are already provided in the previous Section in Figures 5.1 and 5.2.

Step 7: Managing conflicts After hard and soft conflicts have been identified in Step 6, Step 7 discusses how to manage them. Software engineers have three options. They should be tried out in the given order. It is an attempt to include all TrustSoFt elements of a conflict.

1. Returning to Step 5.2 for refinement or countermeasure.

Step 5.2 for refinement or countermeasure is only applicable to trustworthiness requirements. Trustworthiness goals should not be adjusted, as they mark important objectives for mitigating trust concerns. Instead, for conflicts that involve trustworthiness goals, software engineers should analyse whether the trustworthiness requirements derived from the conflicting goals carry the conflict on or circumvent them. Regarding goals, requirements can be seen as a refinement themselves as they specify concrete software behaviour realising a goal. If the derived requirements do not circumvent the conflict, Step 5.2 can be performed for the requirement of the respective goal. For conflicts between a goal and a requirement, Step 5.2 shall be performed for the requirement. If a conflict is between two trustworthiness requirements, software engineers must try to either re-specify one or both requirements in dependence on each other. It is up to the engineers' preference what requirement they want to adjust.

The refinement of a requirement should be close to the original software behaviour. A refinement can either involve a re-specification of a conflicting requirement, that is a reformulation of the requirement, or including subrequirements that describe the conflicting requirement in more detail. An example of refinement can be applied to Requirement A from Figure 5.2. Requirement A can be concretised by determining for what personal information users are asked. Thereby, the conflict with Goal B can be resolved, if the personal information is necessary for providing the service. As a reformulation, Requirement A can be directly refined for example "Asking users to disclose their identity card number". Concerning a refinement in form of subrequirements, Requirement A can be enriched by Requirement A.1 "asking users about their identity card number" or Requirement A.2 "collect name

and age from connected social media platforms”. The refinement of Requirement A by Requirements A.1 and A.2 means that an application first has to ask users for agreement to disclose personal information. Afterwards, the application can directly ask users for their identity card number and then collect the information of name and age from social media platforms that the user has connected with the application.

An alternative to refinement is to add a countermeasure requirement that resolves the conflict. For example, a countermeasure requirement such as “letting users decide for what service the disclosed data is exclusively allowed to use” resolves the soft conflict in Figure 5.2. The countermeasure requirement relates Trustworthiness Requirement A to a service provided by the application. Thereby, Goal B is met.

As experience with TrustSoFt has shown, a reason why conflicts arise is often due to a vague specification of trustworthiness requirements. A refinement might be the easiest and least complex way of resolving a conflict. In contrast, countermeasures provide new, creative solution approaches. Thereby, the level of vagueness in how requirements have been specified is acceptable. Yet, countermeasures add complexity to an application by the increasing number of requirements. Software engineers must decide which conflict management option they prefer for their application.

If the engineers are successful with either refinement or countermeasure, they continue with Step 5.1 for re-assessing the risk score of the requirements. Afterwards, the engineers continue with the extended TrustSoFt method as intended from then on.

In case Step 5.2 cannot be performed, the next two options of conflict management can be applied.

- 2. Deciding on one TrustSoFt element.** If the conflict cannot be resolved by refinement or countermeasure, the software engineers must make a decision on which of the conflicting TrustSoFt elements to implement. For the decision, they have the list of prioritised goals to consult (I/O.B). If one of the involved TrustSoFt elements is a requirement, the goal from which it stems has to be considered. The goals that are involved in the conflict are compared concerning their priority. The element that is (related to) the goal with the higher priority should be implemented in the application.

Regarding the example for the extended TrustSoFt method, Step 3.2 has resulted in a priority list in which Goal A “user authenticity” has a higher priority than Goal B “data usage only for providing the service”. In the case of deciding on one TrustSoFt element of a conflict, this would mean that software engineers choose Trustworthiness Requirement A over Trustworthiness Goal B.

- 3. Considering stakeholder preferences.** Considering stakeholder preferences for choosing what element to implement is valuable for various cases. It is valuable when time and cost are limited. It is relevant when the purpose of the application follows a certain topic, such as privacy or security. In the case of conflict management, stakeholder preferences can additionally be used when consulting the list of prioritised goals leads to a stalemate situation. Then, both goals involved in the conflict have the same priority. At that point, it is up to the software engineers to consider stakeholder preferences to decide on one option for implementation.

The procedure of conflict management is documented in the conflict report (I/O.D). From the conflict report, the requirements that are going to be implemented for the application, are included in the overview table (I/O). Based on the resulting trustworthiness requirements, trust-related software features can be derived (Step 8).

Discussion The extended TrustSoFt method is a valuable guideline for software engineers to reflect on trustworthiness goals and requirements in terms of involved risk. By assessing the risks that are countered by trustworthiness goals and produced by trustworthiness requirements, software engineers can consciously address and mitigate them. In addition, the refinement of trustworthiness requirements and the specification of countermeasure requirements for risk reduction is a feedback loop that impacts both risk management and conflict management. By the refinement of trustworthiness requirements and specification of countermeasure requirements, conflicts between goals and requirements can be tackled even before the conflicts have been identified. Thereby, time and cost for conflict management are reduced.

Conflict management in TrustSoFt can be regarded as efficient software development. Inconveniences can be handled early in the planning and analysis phase of the Software Development Life Cycle before they may lead to cost-intensive problem-

fixing in the later solution-oriented phases such as the deployment phase. In addition, the extended TrustSoFt method provides three options for conflict management. The three options leave software engineers room to creatively and freely handle conflicts and develop an application to their intention.

However, the extended TrustSoFt method has some limitations. One of them relates to the freedom of action concerning the three different options for risk management. Depending on the chosen option and the interpretation of conflicts and solution approaches, software engineers come up with different results concerning trustworthiness requirements or countermeasure requirements. What is beneficial for developing individual applications, is very complex for repeating the same process with different engineers. On these grounds, the documentation of the entire TrustSoFt process in the overview tables is highly important. Only then, the extended TrustSoFt method is comprehensible for external engineers.

Another limitation is that TrustSoFt has become even more time-consuming in its two risk assessments. Risk assessments demand in-depth research about the various topics of risks. Yet, risk management is a field that is indispensable for software development [30]. For that reason, companies hire risk analysts or risk managers to perform risk assessments similar to those in the extended TrustSoFt method. For industry, ensuring users' safety and security by risk management is more profitable than damage repair [89].

6

Goal Modelling for TrustSoFt - the Model-Based Approach

For software engineering, modelling brings many benefits [245]. Benefits include, for example, the visualization of complex scenarios, which increases the engineer's understanding. Furthermore, it can serve as a basis for design decisions. Modelling also supports documentation. As TrustSoFt is insofar complex since it involves multiple interrelated elements, modelling can increase the insights and the comprehension of the TrustSoFt elements and their dependencies among each other.

On these grounds, Paper 5 introduces an adapted form of the i^* goal modelling notation as a model-based extension for TrustSoFt. i^* goal modelling and TrustSoFt are aligned to the extent that both frameworks are based on achieving software goals. By using the adapted i^* notation for TrustSoFt, software engineers can model the context of trust concerns. On this basis, software engineers can further elicit trustworthiness goals, trustworthiness requirements, and first indications for trust-related software features in relation to trustworthiness facets. Subsection 6.1 introduces the adapted i^* goal modelling notation. Afterwards, Subsection 6.2 describes how adapted i^* goal models are created. The subsections are accompanied by examples of the catfishing example introduced in Chapter 4. In addition, Subsection 6.3 introduces how the adapted i^* notation can be used for the identification and conflict resolution of conflicting trustworthiness goals and requirements. This is also illustrated with small examples.

6.1 The Adapted i^* Notation for TrustSoFt

The adapted i^* notation for TrustSoFt extends the original i^* goal modelling notation (see Chapter 2.6) by TrustSoFt elements. The TrustSoFt elements are mapped to the elements of the original i^* notation that semantically show a fit. The mapping is accompanied by frame colouring of modelled TrustSoFt elements to distinguish them from the original i^* elements. Furthermore, the new modeling technique “element reference” is added.

The TrustSoFt elements of the adapted i^* notation are depicted in Figure 6.1. They are explained in the following and demonstrated by examples from the catfishing example introduced in Chapter 4. The elements of the catfishing example are presented in a goal model on page 82.

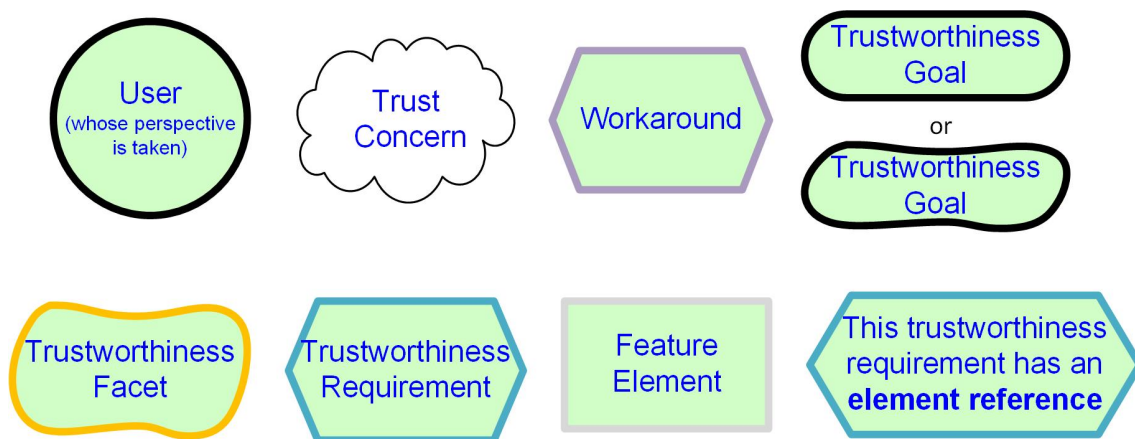


Figure 6.1: Adapted i^* Notation to TrustSoFt Elements. Inspired by [39].

Actors are modelled as in the original i^* notation. The only exception is that the end-user, whose trust concern is addressed in the model, is highlighted by a black, bold frame. Thereby, multiple end-users can be modelled and the point of view in a model is emphasized explicitly. Concerning the catfishing example, involved actors are for example “user (victim)” and “catfish”. The frame of “user (victim)” is modelled black and bold.

Trust Concerns are modelled in the form of beliefs. They are similar insofar as they describe a circumstance that the user assumes can occur. Each model contains one concern that it addresses. A trust concern is modelled within the actor boundary of a user and is connected with a user goal. If existent, the

trust concern is related to a workaround. In terms of the catfishing example, a trust concern could be formulated like “Behind online dating profiles could hide catfish”.

Workarounds are user activities and, thus, modelled as tasks. Their frame is lilac and bold. Workarounds are connected with a contribution link to trust concerns. Usually, the contribution link is a hurt contribution, because workarounds mitigate the concerns. Due to the uncertainty character of trust concerns, workarounds usually cannot completely resolve them. Additionally, workarounds can positively contribute to user goals, since it is the users’ way of achieving them. A workaround to mitigate the catfishing concern is “looking up other users’ social network profiles”.

Trustworthiness Goals are modelled in the actor boundary of the application to be developed. Depending on whether their satisfaction criteria are objectively or subjectively measurable, trustworthiness goals are modelled as goals or soft goals (see Chapter 2.6). Trustworthiness goals pick up user goals within the software and are modelled with a black and bold frame. For reasons of clarity, a goal model should only include one trustworthiness goal. Usually, multiple goals can be created for reducing the respective concern. In that case, for each goal, one model should be created. For the trust concern “Behind online dating profiles could hide catfish”, a trustworthiness goal could be “user authentication”.

Trustworthiness Facets are desired characteristics that resolve a trust concern. As a “desire to be”, their satisfaction depends on the user’s subjective opinion. Therefore, trustworthiness facets are represented by soft goals with a yellow and bold frame. Trustworthiness facets can either be modelled as dependencies between actors or within actor boundaries as qualities to be achieved. Within the actor boundary of an application, trustworthiness facets specify in what way requirements shall be realized within the application or what facet the application shall reflect for trustworthiness assessment. Depending on whose facet the application shall reflect, software engineers need to include the name of the respective CMI party in the trustworthiness facets. As an example, if the application shall reflect the trustworthiness facet “honesty” of its users, such as it would be supportive in the case of catfishing, the modelled facet is specified as *user honesty*.

Trustworthiness Requirements describe system behaviour and are, thus, modelled as tasks within the actor boundary of the application. Their frame is blue and bold. Trustworthiness requirements must either directly contribute to a trustworthiness goal or belong to a decomposition of a higher-level requirement. In addition, they shall be related to a semantically suitable trustworthiness facet. Furthermore, they can be connected to relevant feature elements. Following the principle of inheritance, trustworthiness facets or feature elements of sub-requirements are relatable to the parent requirement by a decomposition link. For the catfishing example, the trustworthiness requirement “depicting the similarity result of the profile comparison” is decomposed into the trustworthiness facet “user honesty”.

Feature Elements are resources that are needed by trustworthiness requirements for their realisation. They are highlighted by a grey, bold frame. Most often, feature elements are either pieces of information, design elements, or interaction elements. They provide the first indication of what a trust-related software feature should include. The specification of trust-related software features is explained in Chapter 9. Feature elements of goal models can serve as input for the specification. An example of a feature element could be “similarity result”. This informational feature element refers to the trustworthiness requirement “displaying the similarity result of profile comparison”. The feature element is linked to the requirement by a decomposition.

Element reference. In some cases, the specification of intentional elements involves mentioning other elements in their formulation. Such a mention highlights a relationship between two elements in a goal model, of which the mentioned element may not have been modelled at that time. If that case, the software engineer needs to include the mentioned element in the model. These textual element references within an intentional element are highlighted in bold. They serve as a reminder for software engineers to include the referred element in the model and connect it with the element that has been referred to it. An example of an element reference occurs in the trustworthiness requirement “displaying the **similarity results** of **profile** comparison”. For being realized, this requirement needs a “similarity result” and more than one “profile” for profile comparison. These words within the formulation of the trustworthiness requirement are element references. By highlighting them boldly in the requirement, the modeller knows to include them within the goal

model. In this case, the two element references are included as decomposed feature elements from the requirement via decomposition links.

6.2 The Procedure for Goal Model Creation in TrustSoFt

The procedure for goal model creation in TrustSoFt is introduced in Paper 5. For the creation of goal models, software engineers must adhere to three conditions. First, goal modeling for TrustSoFt is based on the intent to mitigate trust concerns and serve user benefit. Therefore, software engineers must model each element in the user's interest. Second, software engineers must consider the clarity of the models. In principle, a trust concern yields multiple user goals and trustworthiness goals of the application. To avoid expansive models, a goal model should map only one user goal and one trustworthiness goal at a time. This means that for one trust concern, software engineers can create a multitude of goal models involving different user goals and trustworthiness goals. Since one user goal can be addressed by multiple trustworthiness goals, the number of goal models depends on how many trustworthiness goals the software engineer identifies. As a last condition, this work aims to provide technical solutions for trust concerns. Therefore, goal models shall focus on how the application to be developed can address users' trust concerns. For this reason, elements of the two actor boundaries of the user and the application are of particular interest. While including the actor boundaries of the service provider or other users may also yield solution approaches, these goal models provide solutions from a business or social point of view.

Considering these three conditions, Paper 5 introduces modellers to proceed as follows when creating goal models. The steps are accompanied by the exemplary creation of a goal model for the catfishing example. The goal model is depicted in Figure 6.2 on page 82.

1. **Elements of the actor boundary of the user:** As a starting point, the modeller decides what trust problem or trust concern to address in the model. In this context, the actor "user" and its actor boundary are drawn of who has the trust concern. Within the actor boundary of the user, the modeller includes

the trust concern and the user goal. The user goal expresses the opposite of the trust concern. Usually, users can have multiple goals to mitigate their concerns. Again, to keep the model clear, each goal model should only address one user goal. Therefore, it is common to create multiple models to introduce a variety of solution approaches for one trust concern. If known, additional elements can be added to the actor boundary of the user that sketches the user's intentionalities about the trust concern, such as workarounds.

For the catfishing example from Chapter 4, this means that the actor "user (victim)", who is the potential victim of a catfishing attack, is added to the goal model together with the actor boundary. Within the actor boundary, the trust concern "Behind online dating profiles could hide catfish" is added. Its counterpoint is the user goal "Interacting with users whose identity is the one presented in the user profile". The user goal is modelled as a goal from the i^* notation, because it is objectively determinable, whether a profile represents a catfish or not. The user goal is linked with a break-contribution to the trust concern. The break-contribution denotes that if the user goal is achieved, the trust concern is not relevant anymore. In addition to the trust concern and user goal, the workaround "Looking up other users' social network profiles" is added to the model. It supports users in achieving the user goal, which is emphasized by a help-contribution link. A hurt-contribution link demonstrates that the workaround mitigates the trust concern.

2. **SD model with user dependencies:** Afterwards, the modeller adds further actors to the model that are relevant in the context of the trust concern. Usually, these are the actors "application" and "service provider". Since the application is a product from the service provider, it is related to the service provider with an instantiation link. Additional actors may be a representative "user" causing the addressed trust concern in the user, whose perspective is taken in the model. After the modeller has added all involved actors to the model, the dependencies among them are included. This can be realised, by reflecting on what extent the main user is dependent on the other actors to achieve the user goal. What goals, tasks, or resources does the user need from the other actors to achieve, realize, or provide her intentionalities? What trustworthiness facets does the user wish the other actors to have so that the trust concern becomes meaningless?

For the exemplary catfishing goal model, the actors "catfish", "online dating

application” and “service provider” are included in the model. The application is connected by an INS-relationship link with the service provider. As a next step, the modeller draws the dependencies of the user on the other actors. To accomplish the user goal “Interacting with users whose identity is the one presented in the user profile”, the user is dependent on the online dating application to perform user authentication. This is presented by the goal-dependency link “user authentication” that originates from the user goal. The goal-dependency demonstrates that the user needs the application to establish user authentication as a goal for itself. Since user authentication can be objectively determined whether it is performed or not, the dependency is modelled as a goal-dependency instead of a softgoal-dependency. As a side note, the user’s dependency on a user authentication by the application could alternatively be modeled as a task dependency, e.g., “verifying user authenticity”. Further dependencies are between the user and the catfish. The user wishes catfish to be honest and predictable. If catfish possessed these trustworthiness facets, they would reveal themselves and stop the deceit. This wish for trustworthiness facets concerning catfish is modelled by the two facet-dependencies “honesty” and “predictability” from the actor “user (victim)” to the actor “catfish”.

- 3. SR model of the application including necessary dependencies:** After the basic SD model is completed, the modeller continues with the SR model by adding intentional elements to the actor boundaries. As this work aims at eliciting technical solutions, the focus mostly is on the actor boundary of the application. As a starting point, the modeller specifies a trustworthiness goal for the application, which reacts to the user goal and user dependency. Afterwards, the modeller has to consider, what sub-goals or trustworthiness requirements are necessary to achieve the trustworthiness goal. As a next step, further elements can be added to the model that are required to achieve the trustworthiness goal or to complete a requirement, such as trustworthiness facets or feature elements. Usually, the facet-dependencies of the user in the model point out, which facets the application needs to reflect. Thereby, applications enable users to perform their trustworthiness assessment. During this process, dependencies of the application might become apparent. This is for example the case when the application needs input from other actors to realize trustworthiness goals or requirements.

Concerning the catfishing example, the actor boundary for the online dating application is included in the model. Considering the condition to always model in the interest of the user, the first element added is the trustworthiness goal “User authentication”. It is linked to the goal dependency “user authentication”, emanating from the actor “user (victim)”, which shows the user’s need for user authentication to be picked up by the online dating application. Now that the trustworthiness goal “user authentication” is specified, the software engineer must reflect - either creatively or backed up by literature - on how to achieve it. In the context of the exemplary goal model, an option to realize user authentication is to check whether the profile information of a user’s online dating profile matches the profile information of other social network sites. This idea originates from the user’s workaround “looking up other users’ social network profiles”. Following this idea, three trustworthiness requirements are modelled, which are: i) “asking users for access to their social network profiles”, ii) “calculating the similarity of profile information from the online dating profile and users’ social network profiles”, and iii) “depicting the similarity result of the profile comparison”. The three trustworthiness requirements are connected to the trustworthiness goal with means-end links, as they are necessary for realizing it. In addition, the specification of the trustworthiness requirements refers to elements that are necessary to realize them. These element references are “social network profiles” in the first trustworthiness requirement and “profile information”, “online dating profile”, and “social network profiles” in the second requirement. Element references in the third trustworthiness requirement are “similarity result” and “profile”. All element references are highlighted in bold in the trustworthiness requirements. As a next step, the trustworthiness requirements are analyzed for decomposition and dependencies. The trustworthiness requirement “Asking users for access to their **social network profiles**” involves an interactional feature element that is specified as an “access request”. Furthermore, the trustworthiness requirement has the element reference “social network profiles”, which is why this element is added as an informational feature element to the model. Both feature elements are decomposed from the trustworthiness requirement by a decomposition link. Concerning the feature element “social network profiles”, the online dating application depends on the user to give access to them. This is modelled by the task-dependency “agreeing on access request for social network profiles”. The task-dependency is linked to the feature element “social

network profiles” and points to the actor “catfish”. For completeness of the goal model, the element would also have to be connected to the actor “user (victim)”. However, for reasons of model clarity, the task-dependency is only connected to the actor “catfish”. Thereby, the solution approach of mitigating the trust concerns of the actor “user (victim)” is emphasized. Next, the trustworthiness requirement “calculating the similarity of **profile information** from the **online dating profile** and users’ **social network profiles**” is analyzed. First, the element references of the trustworthiness requirement are added to the model. The feature element “social network profiles” is linked to the trustworthiness requirement by a decomposition link. In addition, “social network profiles” is decomposed into the informational feature element “social network profile information”. Afterwards, the informational feature element “online dating profile” is added to the model. It is also decomposed into the informational feature element “online dating profile information”. Having these feature elements covered, their similarity needs to be calculated according to the trustworthiness requirement. For that purpose, the feature element “similarity algorithm” is also added as a decomposition to the trustworthiness requirement. The last step is the analysis of the trustworthiness requirement “depicting the **similarity result** of the **profile** comparison”. Concerning its element references, the informational feature element “similarity result” is added to the goal model as a decomposition from the feature element “similarity algorithm”. “Similarity result” is further linked by a decomposition link with the trustworthiness requirement. Regarding the element reference “profile”, the profiles of online dating and social network sites have already been added in the previous steps. Since the profile comparison is already considered in the model, there is no need to include any further element or relation to this trustworthiness requirement. In the end, the trustworthiness requirement “depicting the similarity result of the profile comparison” has been worked towards throughout the whole model. It enables users to perform their trustworthiness assessment of other users by evaluating the trustworthiness facet “user honesty”. The facet is added to the goal model as a decomposition of the trustworthiness requirement. The previous trustworthiness requirements are prerequisites for the realization of this requirement and users’ trustworthiness assessment.

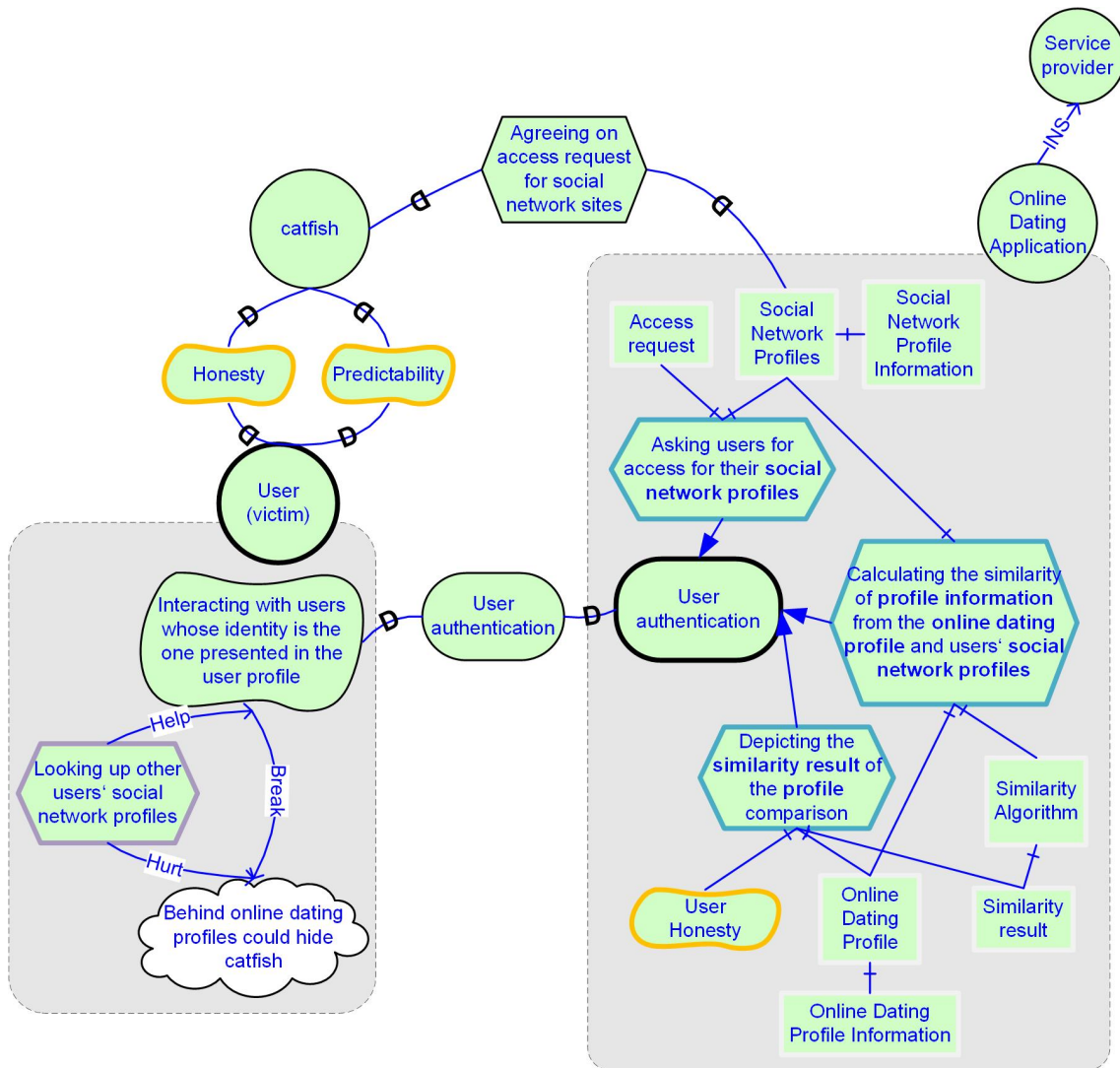


Figure 6.2: Exemplary goal model for the catfishing problem in online dating.

6.3 Conflict Identification and Resolution with the Adapted TrustSoFt Goal Models

Goal models for TrustSoFt can be additionally used to identify and resolve conflicts between requirements and goals. Using goal models for this purpose is the model-based approach that extends the concept of conflict identification and resolution of requirements and goals from Paper 4 described in Chapter 5.

As goal models target different trust concerns and user goals, conflicting trustworthiness requirements or trustworthiness goals between goal models can occur. The following procedure is an approach to resolving conflicts so that as many trustwor-

thiness goals and requirements as possible can be realised in software development.

A finding of Chapter 5 is that many conflicts stem from a lack of precision in the specification and formulation of goals and requirements. Therefore, Chapter 5 proposes a refinement of the respective trustworthiness requirements to resolve a conflict in form of a reformulation or re-specification. i^* goal modelling is a suitable approach for requirements refinement, because its notation provides this option by decomposition links. Furthermore, i^* goal modelling visualises the relations of trustworthiness requirements and goals so that learnt insights support the reformulation of the conflicting elements.

In the process of conflict identification and resolution, the frame colours of trustworthiness goals and requirements are adapted for different conflict statuses. An overview of how such elements are modelled is depicted in Figure 6.3. A red frame symbolizes that an element conflicts with another one. A green frame represents an element that resolves a conflict. Those elements can be implemented in the system to be developed. A transparent reddish frame shows that an element is potentially problematic depending on the way the element is realised. It can happen that the frame colours of an element change from red to transparent red when a conflict is only resolved under certain circumstances.

Conflict identification and resolution can also impact other intentional elements that are related to a conflicting or resolved element. Elements are either related to one another by links or when their specification has bold elements. Then, related elements need to be checked whether they are also involved in the conflict. This is upon the expertise of the software engineer to decide. If a related element is involved, its frame colour must be changed, as well.

The procedure of conflict identification and resolution is explained in the following. The explanation is accompanied by a conflict example between a hypothetical goal model with the trustworthiness goal “user privacy” and the trustworthiness



Figure 6.3: Example of frame colour change for trustworthiness requirements and goals during conflict identification and resolution. Figure by [39].

requirement “asking users that have dated the person already, whether the learnt information during the date confirmed the information learnt during the online interaction” from the catfishing example in Table 4.1 on page 53. For the example, an excerpt of the goal model showing the actor boundary of the online dating application with the resolved conflicting trustworthiness requirement is presented in Figure 6.5 on page 89.

Conflict identification. For conflict identification, goal models are compared pairwise. The single trustworthiness requirements and goals are considered on a semantic level against the ones of the other model. The advantage of a single-element evaluation instead of an element-bundle evaluation, such as whole requirement decompositions or goal contributions, is that problematic elements can be identified more easily. Single elements are easier to grasp than they are in constellations and relations to others. The semantic evaluation of single elements leads to insights into whether an element compromises or interferes with another.

An example of conflict identification is the conflict between the trustworthiness goal “user privacy” of a hypothetical goal model and the trustworthiness requirement “asking users that have dated the person already, whether the learnt information during the date confirmed the information learnt during the online interaction” from the catfishing example in Table 4.1 on page 53. Privacy is defined as an individual’s ability to limit access to one’s personal information [283]. It further describes an individual’s right to determine when, how, and to what extent information about oneself is exchanged by others [283]. According to these definitions, the trustworthiness requirement denotes that the system does not respect the privacy of its users when it asks other users about them for information. The system tries to gather new information about users without them having control over it. Therefore, the trustworthiness goal and requirement receive a red frame within their goal models. The conflict should be documented and defined outside the goal model to keep the overview. As a next step, related elements must be checked on whether they are also involved in the conflict. In Figure 6.5, the trustworthiness requirement is decomposed into the sub-requirement “asking users about the correctness of provided profile information (e.g., name, age, job)”. As this only involves information that the user agreed to provide, the requirement can be regarded as conforming to the trustworthiness goal “user privacy”. Therefore, the re-

quirement is not conflicting. Thus, its frame colour does not change. The same applies to the decomposed feature element “user answer”.

Conflict resolution. The approach for conflict resolution is to adapt system behaviour (that is requirements) and related elements in such a way that a conflict between two goal models no longer exists. Therefore, conflict resolution happens on a requirement level - even if there is a conflict between two trustworthiness goals. The premise for conflict resolution is a full understanding of the conflict and the specifications of each involved element. Software engineers can resolve conflicts by either conflict refinement or including countermeasure requirements that avert the conflict between two i^* goal models. As mentioned above, conflicts often arise due to a lack of precision in the specification of trustworthiness requirements. Therefore, software engineers should try to refine the conflicting requirements, first. If the refinement does not succeed, software engineers should consider countermeasure requirements by whose implementation the conflict is eliminated.

The named order of refinement and countermeasure is a suggestion to realise the original intention of the development. However, it is up to the developers to decide whether they want to directly find a countermeasure requirement for conflict resolution before trying to refine the conflicting requirements. In the following, the introduced guideline for conflict resolution follows the given order proposal.

If there is a conflict between two requirements, the software engineer should try to adapt one or both until the conflict is resolved. In case of a conflict between a trustworthiness goal and a requirement, either the goal definition needs to be adapted outside the model or the requirement within a model must be re-specified. However, adopting a goal definition would lead to unaddressed trust concerns, which is why software engineers should refrain from doing so. If there is a conflict between two trustworthiness goals, software engineers should analyse their requirements, and whether the specified behaviour averts the conflict. If not, the engineers should try to adapt the related requirements as described below.

It must be noted that a conflict cannot always be resolved. Sometimes, opposing intentions are pursued in the models. If a conflict cannot be resolved by refinement or countermeasures, the other two options for conflict management,

that is deciding on one conflicting element or stakeholder preference, must be applied as described in Chapter 5.

Figure 6.4 presents the procedure of conflict resolution. It involves four approaches, which are sub-requirement check, reformulation (known as “new requirement” from Paper 5), refinement/new decomposition, and countermeasure. The procedure is explained in the following.

The procedure for conflict resolution is to be applied by software engineers to goal models that contain conflicting requirements. Since usually the conflicting goal models both involve conflicting requirements or goals, the software engineer can decide with which to start first. The decision can be led by criteria such as the importance of a requirement for a system or the domain expertise of the engineer.

The starting point of the procedure for conflict resolution is the lowest conflicting requirement (recognizable by the red frame) on a decomposition structure. The reason for this is that based on how requirements are decomposed and refined, conflicts of higher-level requirements can be circumvented by specifying system behaviour in more detail. An example of how sub-requirements circumvent conflicts is described above in the paragraph about conflict identification and presented in Figure 6.5. When the lowest conflicting requirement is resolved, it represents a condition under which higher-level requirements are no longer conflicting and can be implemented. They are then marked as potentially conflicting by a reddish frame. By resolving conflicts at the lowest level of a decomposition structure, changes to the goal model are limited to the decomposed requirements, while the higher-level requirements can continue to exist. Thereby, compromises are found that resolve the conflict between the goal models while the approach of addressing the trust concerns of both models can still be realized.

As a first step, the software engineer should conduct a **sub-requirement check** (Box A, Figure 6.4). Although the conflicting requirement is the lowest conflicting one in the decomposition structure, there might be additional decomposed requirements that are not part of the conflict. In that case, the sub-requirements and their type of decomposition (AND, OR, or XOR) are checked whether they already refine the conflicting requirement in a way that the conflict is resolved. In that case, the sub-requirements receive a green frame, while

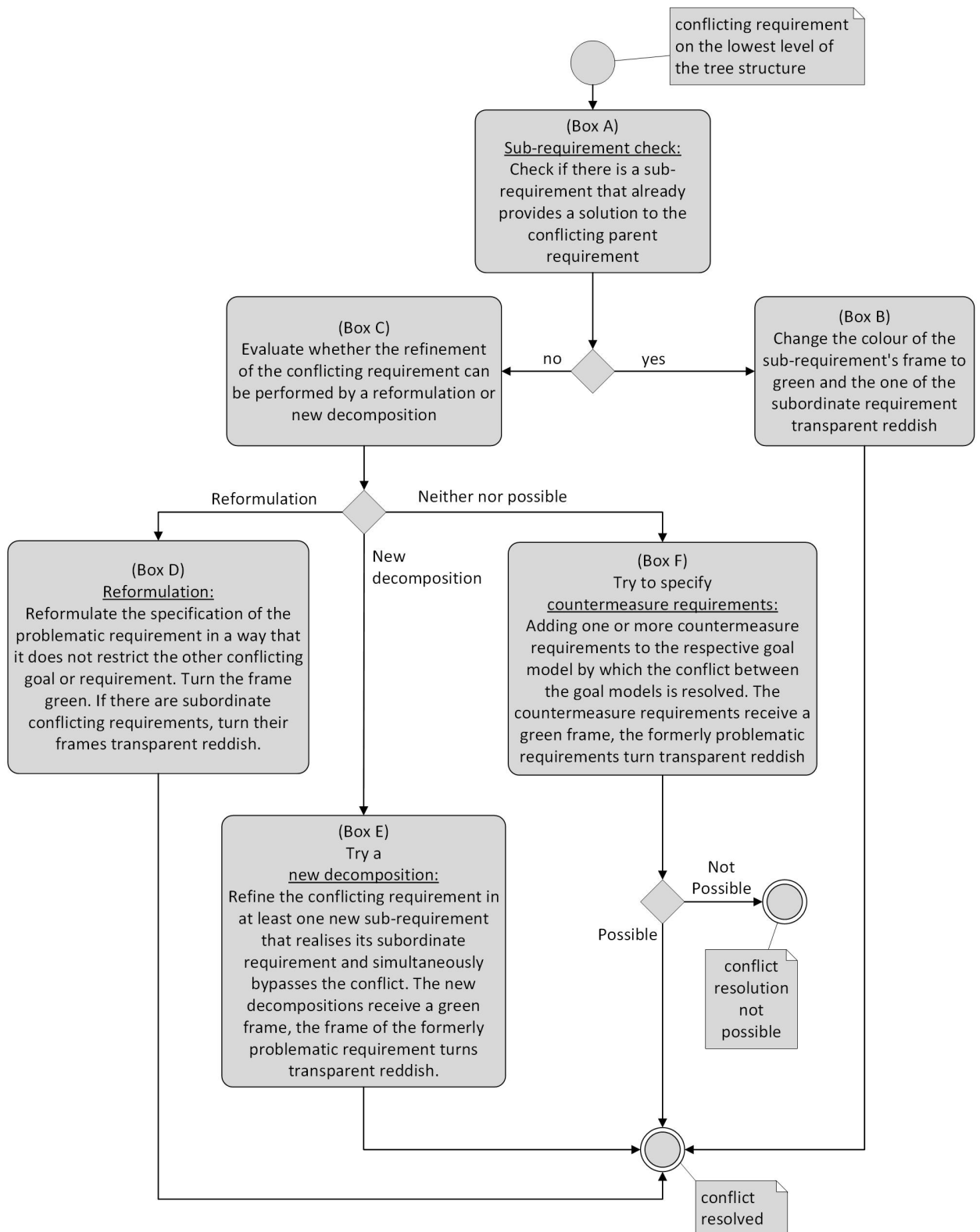


Figure 6.4: Procedure for Conflict Resolution

all subordinate requirements are marked as potentially problematic with a transparent reddish frame (Box B). An example of the sub-requirement check is depicted in Figure 6.5, which was already explained above in the paragraph about conflict identification. As the trustworthiness requirement “asking users about provided profile information (e.g., name, age, job)” is not in conflict with the trustworthiness goal “user privacy”, its frame and the one of the trustworthiness goal are coloured green. In addition, the decomposed feature element “user answer” also receives a green frame. The higher-level, formerly conflicting trustworthiness requirement receives a transparent reddish frame.

However, in the case that the conflicting requirement does not have any sub-requirements, the software engineer needs to check whether the specification of the conflicting requirement can be easily reformulated (Box C). This means that the specification can be directly changed in a way that the conflict is resolved. If this is possible, the engineer should perform the **reformulation** (Box D). To give an example, it is referred again to Figure 6.5 with the assumption that the conflict-resolving sub-requirement is not part of the goal model. In that case, the again conflicting trustworthiness requirement could be reformulated so that the conflict is resolved. A reformulation of its specification could be “Asking users that have dated the person already, whether the profile information matches what has been learnt during the date”. As explained above, from the semantics, such a requirement does not conflict with the trustworthiness goal “user privacy”. The frames of the trustworthiness requirement and the goal are turned green.

In case a reformulation is not possible, the engineer should try to refine the conflicting requirement by a **new decomposition** (Box E). The new decomposition needs to consist of at least one sub-requirement that realizes the subordinate, higher-level requirement in a way that the conflict is circumvented. The procedure of a new decomposition may lead to further new elements or decompositions. If a resolving new decomposition is possible, all new elements receive a green frame. All subordinate, higher-level conflicting elements receive a transparent reddish frame. If a new decomposition does not lead to a conflict resolution, compromises must be accepted not realising requirements or not addressing goals. To make the best decision in this case with the most benefit, the procedure of Chapter 5 can be applied. An example of a new decomposition is depicted in Figure 6.5.

For the example, it is assumed that the formerly conflicting trustworthiness requirement does not have a decomposition at the beginning of the conflict resolution procedure. The conflict that is resolved by the new decomposition is between the trustworthiness goal “User Privacy” of a hypothetical other goal model and the trustworthiness requirement “Asking users that have dated the person already, whether the learnt information during the date confirms the learnt information during online interaction” from the catfishing example from Table 4.1 on page 53.

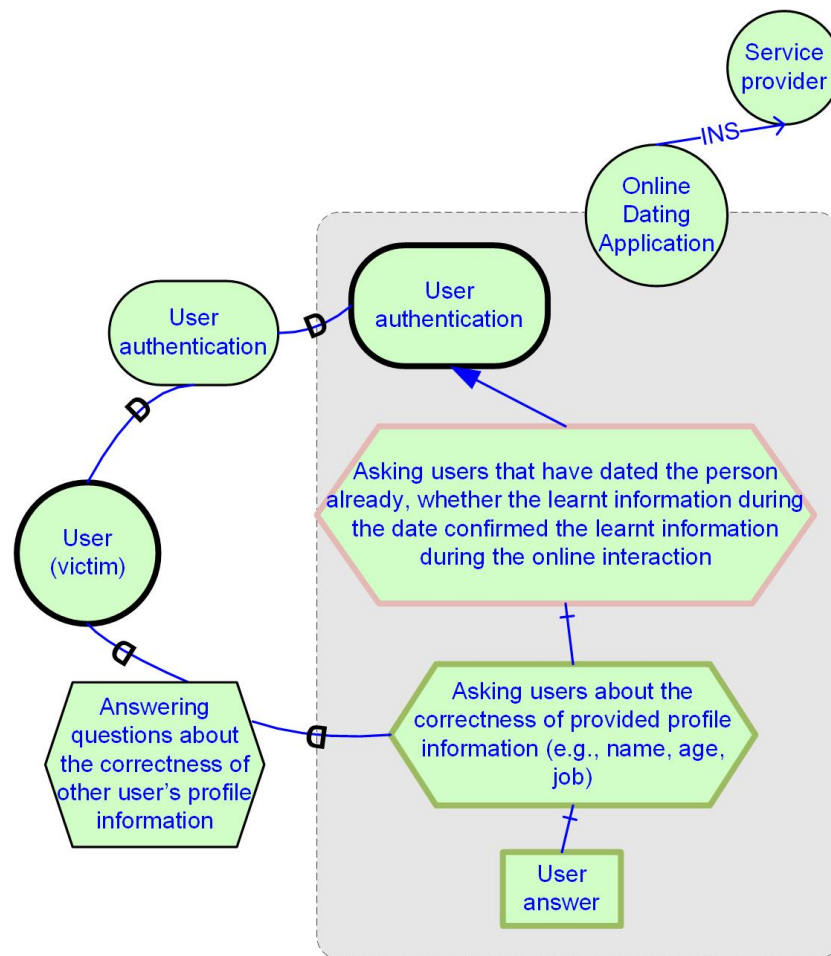


Figure 6.5: Exemplary actor boundary of an online dating application after a conflict is resolved.

Instead of reformulation or specifying new sub-requirements by new decompositions, another way of resolving conflicts is by specifying one to n countermeasure requirements. Trying to specify countermeasure requirements is proposed as the last option in the procedure for conflict resolution (Figure 6.4, Box F). Countermeasure requirements are added as trustworthiness require-

ments to the i^* goal model. They are linked to the respective trustworthiness goal by a means-end link. As countermeasure requirements resolve the conflict, their frame colour is green. The frames of the formerly conflicting TrustSoFt elements are changed into transparent reddish.

An example of a countermeasure requirement can be given concerning the conflict described earlier in the paragraph of conflict identification and Figure 6.5. To protect the privacy of users while asking users for information about other users, online dating applications may ask users for their consent in this regard. A countermeasure requirement could be “obtaining user consent to question other users about the truthfulness of personal information learnt during user interaction”. By the countermeasure requirement, users can control whether other users and the service provider share information about them so that user privacy is respected.

7

Evaluation of TrustSoFt

Newly introduced methods or tools usually need to prove their usefulness in their application for practitioners. A popular way to evaluate new methods or tools is by conducting use case studies [175]. Organisations conduct use case studies to test the application and the effect new methods or tools have in realistic contexts. Use case studies are a cost-effective way of testing that usually provide promising, practical results [175].

On this behalf, TrustSoFt has been applied and evaluated in eight product development projects in a university context. In each product development project, students in a group of five had the task to develop a new application concept for either online dating or sharing economy and to apply TrustSoFt for this purpose. This makes a total of 40 students, who applied and evaluated TrustSoFt. The students either came from the discipline of computer science or were interdisciplinary students of psychology and computer science. They are called “developers” in the following. Concerning the application concepts, examples are amongst others an online dating application for pet owners, neighbours teaching each other skills like playing the guitar, or meal rescues against food waste.

The focus of the developers’ task was on addressing three user concerns about their application to be developed that they identified through user interviews. They addressed user concerns by specifying trustworthiness goals, facets, and requirements, using the adapted i* goal modelling notation for TrustSoFt. In addition, they documented the results simultaneously in an overview table such as Table 4.1 on page 53. The resulting requirements have been implemented in form of software features in clickable front-end prototypes. After this process, the developers evaluated each TrustSoFt step in terms of benefits and drawbacks with respect to its application and the outcomes. The evaluation was guided by an evaluation sheet

that is depicted in Appendix D.1. The evaluation of the developers includes general impressions and improvement suggestions for TrustSoFt and its steps.

The evaluation of TrustSoFt is qualitatively analyzed by summarizing the thought streams of the developers about TrustSoFt and its steps. The thought streams are mainly on an abstract level. In some cases, single opinions are displayed when they are more concrete about the application of TrustSoFt. This was especially the case for improvement suggestions.

In the following, the results of the TrustSoFt evaluation are presented for each TrustSoFt step. Afterwards, the developers' overall impression of TrustSoFt is reported. This is followed by limitations of the evaluation and an overview of the most significant evaluation results. The results and improvement proposals are partly considered and discussed for an enhanced TrustSoFt concept in Chapter 8.

Step 1: Identifying Trust Concerns and Workarounds. In the first step, the product development teams had to identify the trust concerns of the target users of their application concept. For that purpose, the developer teams conducted a semi-structured, qualitative user interview study interviewing 15 target users. Each team determined interview questions in the context of their application concept that served as a guideline for asking questions. Following the concept of semi-structured interviews, the interviewers were allowed to add further questions during the interviews if they fit the communication flow. The predefined interview questions targeted users' thoughts, needs, motivation to use, and concerns in regard to the application concept. A compilation of exemplary interview questions that the various developer teams have defined is depicted in Appendix

The evaluation shows that the developers perceived this step as very beneficial. The developers believed the trust concerns were relevant to create an application that addresses the users' needs. Moreover, user interviews were convincing as a method for identifying trust concerns. Based on the user interviews, they identified new concerns they had not thought about before. New learnings especially involved users' trust concerns about them as the service provider and the application as a technology. In addition, the developers gained a deeper understanding of the target users and their needs for the application. In this respect, the interviews provided input for the first step of trust concern identification and additionally for

the subsequent steps such as software feature specification. The developers rated the user interviews as an eloquent, fast way for eliciting trust concerns. They are further convinced that other methods have higher costs concerning time and effort to get the same results. Moreover, conducting the interviews for trust concern identification also positively impacted the teams' way of working. The codes of the transcribed interviews supported the teams in structuring their future work processes. By the interview codes, the team members knew what topic needed to be addressed and organized which team member would work on it. In addition, the codes enabled all team members to work on the various topics covered in the codes without having previously studied the topic.

Despite the multitude of benefits, identifying trust concerns through user interviews had a few drawbacks. For some team members, the interviews posed a challenge due to missing experience in conducting interviews. It was difficult for them to create a flow of speech despite the prepared interview questions. Some developers further had the feeling to push the interviewees in a direction in their answers. In addition, the teams were aware that interviews cannot grasp all of the users' trust concerns as interviewees are not aware of all of them and have their subjective opinions. Another comment was that even though the interviews provided additional input as proposed software features by the target users, the proposals led to a biased perspective of the team members throughout the TrustSoFt development process. Thereby, it was a challenge to think out of the box and create new requirements and software features that target users have not stated before.

Concerning improvement suggestions for the first TrustSoFt step, some teams find it helpful if TrustSoFt guided them in formulating interview questions to identify trust concerns. Nonetheless, all teams were very pleased with the first TrustSoFt step.

Step 2: Determining Relevant Trustworthiness Facets. The second step is about identifying relevant trustworthiness facets that resolve the trust concerns as described in Chapter 4.

The developers rated trustworthiness facets as beneficial because they supported taking the perspective of the user to later identify user-centered software requirements more easily. The developers were aware that if they realize the trustworthiness

facets in their application or reflect them for the trustworthiness assessment, users' trust concerns would be mitigated. For the developers, the trustworthiness facets aid in determining what qualities the application needs to provide to make it easier for the users to use the application. Concerning facet specification, the developers rated the process as simple, since the domain knowledge had been established in the interviews. Since they additionally took the trustworthiness goals into account for identifying relevant trustworthiness facets (which is not proposed in TrustsoFt), the allocation of facets and goals proceeded automatically for them. Concerning the i* goal modelling, knowing the relevant trustworthiness facets supported the developers in specifying trustworthiness requirements. In addition, goal modelling usually led the developers to identify further relevant trustworthiness facets since modelling helped them to visualize the subject and evaluate it from new angles.

However, for the developers, considering the trustworthiness facets for product development had some drawbacks. As TrustSoFt and the trustworthiness facets were new to them, the value of the facets for the development process only became clearer in the next phase of deriving trustworthiness requirements. Thus, selecting trustworthiness facets relevant to the trust concerns was a challenge for the teams, which was accompanied by a high time investment. Therefore, a few developers questioned the significance of the trustworthiness facets compared to the time investment. They believed that following their gut feeling in addressing or realizing the trustworthiness assessment rather than exactly specifying the trustworthiness facets would lead to similar results.

As an improving suggestion, the development teams proposed to provide further information and instructions than a definition of trustworthiness facets and the overview of them from Chapter 3.2.

Step 3: Deriving Trustworthiness Goals. Step 3 was the derivation of software goals based on the identified trust concerns as it is described in Chapter 4. The trust concerns and trustworthiness goals were included in the i* goal models and the overview table (cf. Table 4.1 on page 53).

The development teams perceived Step 3 as simple and intuitive. They described the specification of trustworthiness goals as formulating the opposite of the trust concerns. It was perceived as positive the developers consciously specified trustwor-

thiness goals that counter the concerns related to the three involved stakeholders i) user, ii) service provider, and iii) application. Thereby, the teams had the feeling of covering users' trust concerns best as possible. For the teams, the trustworthiness goals additionally gave them a direction for the following development steps and how to structure their work. In addition, specifying trustworthiness goals united team members in their motivation to pull together and jointly develop the application.

No drawbacks have been mentioned for the derivation of trustworthiness goals. However, the development teams proposed improvement suggestions concerning i^* goal modelling. They advise that redundancies can be avoided when semantically similar trustworthiness goals are modelled in one i^* goal model instead of separately.

Step 4: Facet Allocation to Trustworthiness Goals. In step four, the development teams allocated the identified trustworthiness facets from the previous step to the specified trustworthiness goals.

For most of the developers, the allocation of the trustworthiness facets to the trustworthiness goals supported the specification of the trustworthiness requirements in the context of trust and users' trustworthiness assessment. However, some developers questioned the value of the allocation. They believe that the requirements specification would have been similar without the allocation so that the effort of performing the allocation could have been saved. Concerning the allocation itself, the general opinion is that it was easy because it kind of happened "automatically". Although not proposed by TrustSoFt, the developers used the trustworthiness goals to derive the trustworthiness facets. Moreover, the trustworthiness goals and facets were modelled in the i^* goal models so that the allocation was visualised even without consciously performing it.

All in all, the developers agreed that performing the allocation consciously as an own step, as it is intended by TrustSoFt, is not necessary. Due to the goal modelling, the allocation is part of the trustworthiness facet identification.

Step 5: Specifying Trustworthiness Requirements. In step 5, the development teams specified software requirements by considering the trustworthiness goals and allocated trustworthiness facets. The three TrustSoFt elements were modelled in the i^* goal model and included in the result table. The developers added

additional i^* elements, such as feature elements, to the goal models to complete them.

In total, the developers rated this step as essential for product development, especially for programming and prototyping. At this point, they recognized the shift from the abstract preliminary work to concrete behaviour descriptions. They were convinced by the benefits of trustworthiness requirements and the step to specify them insofar that it triggered group discussions to commit to a product version. Yet, the developers appreciated that the trustworthiness requirements left space in their concreteness for refinement in the subsequent steps. The teams had fun in the process of being creative. This step fostered new ideas for software features for the next step. In addition, addressing both trustworthiness goals and facets to specify trustworthiness requirements resulted in new perspectives that led to identifying new relevant trustworthiness facets. The developers had to be concrete in requirements specification and, thus, had new ideas in what way the requirements could address or realize additional trustworthiness facets than the ones identified before.

In terms of the trustworthiness requirements themselves, the developers did not see any drawbacks. However, they rated the i^* models as less useful for requirements specification. Instead, the overview tables were especially valuable and sufficient to orientate, specify and document the results. The developers added the trustworthiness requirements to the i^* goal models simply because it was prescribed by TrustSoFt.

As improvement suggestions, some developers proposed to relate the trustworthiness facets with the trustworthiness requirements instead of the trustworthiness goals. During requirements specification, the value of the trustworthiness facets became apparent. The process of requirements specification supported the facet identification, which in turn concretized the requirements specification.

Besides the benefits, drawbacks, and improvement suggestions, the developers had further comments on this step. Throughout TrustSoFt, the developers already have ideas for software features, although this is the last TrustSoFt step. Especially the first step with the user interviews inspired the developers to identify software features. Their ideas of software features impacted the requirements specification insofar that some developers derived what system behaviour is needed to realize the software features they already had in mind. Other developers stated that steps 2-5

happen rather simultaneously because they mutually depend on each other. Therefore, trustworthiness goals, facets, and requirements should be considered together. Again, other developers had the challenge of not directly eliciting software features and focusing on the requirements and facets first. Differences in their perception of the challenges of this step might be relatable to their background. Especially developers with a background in computer science rated this step as easy. Developers with a more psychological background tended to view the TrustSoFt steps holistically rather than differentiating between them.

Step 6: Deriving Trust-Related Software Features. In the last step of TrustSoFt, the developers derived trust-related software features. The derivation is based on the previously specified trustworthiness requirements while having the identified trustworthiness facets in mind. The i^* goal models and their additional elements, such as the feature elements, also contribute to the specification of trust-related software features. The features specified were added to the overview table.

For the developers, this step was very beneficial because it facilitated the prototyping. By clearly specifying trust-related software features, the developers defined feature boundaries, agreed on the scope of implementation, and, thus, saving time and effort during prototyping. It was easy for the developers to define the software features, because of their many ideas throughout the whole TrustSoFt process. Many developers appreciated this step because it helped them to concretise the ideas about the features they gained during the interviews in the first step. The developers especially perceived the overview tables as supportive. The tables and goal models ensured that every trustworthiness goal, requirement, and facet was considered by the features. Furthermore, the developers used the tables to actively elaborate on software features in the team, which empowered their teamwork and shared product vision. The developers were satisfied with the process because they believe to have given their best in addressing user concerns and needs. The elaboration in the teams enabled a structured implementation of the software features.

For the last step of TrustSoFt, the developers could not state any drawbacks. Still, they had improvement suggestions about classifying trust-related software features for the subsequent implementation. It was proposed that features can be either classified into “minimum viable product (MVP) features” or “nice to have (NTHs)”. MVPs are product versions that include just enough software features to be usable

for the end users [257]. Thereby, organisations can receive early user feedback while enhancing the product with NTHs. Another proposal for classifying features was to categorize them by their implementation type, such as “algorithm,” “design,” or “information,” to facilitate the organisation of the implementation. Furthermore, the developers proposed an improvement in the way of working together. The developers suggested that teams need to agree on a level of detail for feature specification before starting it. In their opinion, a common ground for discussions about the detail of specifications supports the effectiveness of teamwork.

Overall thoughts about TrustSoft. In total, the developers regarded TrustSoFt as a supportive method to develop software applications in a structured way. It is useful to prepare oneself for software implementation and design. They were convinced that TrustSoFt is especially valuable for achieving the overarching goal to keep track of the users’ trustworthiness assessment as a developer. Concerning teamwork, TrustSoFt guided discussions and left room for creativity to jointly identify tailored solutions. Already the first step of TrustSoFt inspired the developers to software features. Most of the developers were satisfied with structurally defining and enhancing their gained feature ideas in the subsequent TrustSoFt steps.

However, nearly a third of the developers criticized exactly this. For them, the effort of TrustSoFt exceeded its value. Some developers were already satisfied with the feature ideas gained in the interviews of the first TrustSoFt step. As they still had to apply TrustSoFt in the context of the academic project, the developers reported that having feature ideas from the very start was a bias throughout the whole TrustSoFt process. The bias expressed itself in that the developers executed TrustSoFt “backwards”. In their mind, the developers conducted the last TrustSoFt step of eliciting trust-related software features and then executed every TrustSoFt step accordingly to the features they targeted to realize. Although some new requirements and features could be determined (especially for complex problems) throughout TrustSoFt with this bias, the developers perceived the i^* goal models as very effortful and unnecessary.

In general, the developers shared the opinion that learning the i^* goal modelling notation was time-consuming and the i^* goal model creation complex. They agreed that the more complex the problem to model, the more expansive and less clear the resulting goal model. Instead, the developers preferred to document each step

in the overview result tables, which they perceived as efficient and sufficient. For them, the tables oftentimes served as a checklist to track their progress in achieving the trustworthiness goals and not forgetting any TrustSoFt elements throughout the development process. Yet, for complex problems, the developers rated the i^* goal models as valuable for increasing one's understanding and gaining new ideas.

Despite the general opinion that the i^* goal models were oftentimes too high in effort for the gained value, the developers agreed that they were beneficial insofar as to deeply enhance the domain knowledge of the problem by the graphical representations of the TrustSoFt elements. This deep knowledge is especially true for the modeller. Therefore and to react to the complexity of the goal model creation, the developers proposed to create the goal models together as a team rather than alone. Nonetheless, the i^* goal models ensured that all team members shared a common understanding of a problem and the solution approach, even if the other members took no part in the modelling process. Moreover, the developers believe that the i^* goal models support external people in comprehending what the development team strives for. In addition to the increased understanding, the developers appreciated the i^* goal models for reasons of documentation.

In terms of future work for TrustSoFt, the developers propose a validation method to prove the correctness of the applied notation. Furthermore, some developers recommended using risk matrices as a scale for prioritizing trustworthiness goals for software development. Thereby, developers can decide with what trustworthiness goal to start goal modelling.

7.1 Limitations of the evaluation

Despite the high value of the evaluation, its limitations must be considered as well. The main limitation is that the developers were students in a university context. The product development projects were academic practical projects for which the students received grades. While on the one hand, this ensured a high motivation of the students to develop a unique application and apply TrustSoFt correctly, it also means on the other hand that the students might have been biased in the evaluation of TrustSoFt. Since the lecturers of the academic practical project created TrustSoFt, students might not want to criticise them to be well-graded. To counter

the bias, it was made clear to the students that a good grade can only be achieved when the evaluation is performed critically. A critical evaluation includes both benefits and drawbacks of TrustSoFt and proposes improvement suggestions. Given the results of the evaluation, it can be said that the bias has been largely eliminated.

Another limitation is the experience of the students in product development. For most of them, it was their first experience in product development and project management. Moreover, they mostly worked equally in their task within their teams. In the free economy, product development teams usually have various positions, tasks, and abilities. This involves different roles in development teams impacting the way of working. Thus, each industrial team member might have a different perspective when it comes to a requirements elicitation method like TrustSoFt and can only afford a different effort in applying TrustSoFt. Moreover, teams in the free economy additionally have the business strategy and product vision for the next years in mind. The business strategy and product vision might have an impact on how goals and requirements are prioritized and specified. The business aspect was completely neglected in the academic projects as it was not part of the scope.

Considering these limitations, it would be valuable to know the challenges professionals would have with TrustSoFt. Furthermore, another question is to what extent TrustSoFt would be executable in the industrial work context. Due to the different professional backgrounds of industrial product development teams, their improvement suggestions would be even more valuable than the ones of the student developers.

7.2 Take-home message of the TrustSoFt evaluation

Nonetheless, the evaluation of TrustSoFt resulted in important insights. The developers perceived the TrustSoFt steps as very supportive to realize users' needs and their trustworthiness assessment for the application to be developed. Furthermore, TrustSoFt guided the developer's workflow in a structured way and led to applications that included all the goals for which they strived. Especially the user interviews in the first TrustSoFt step provided meaningful input for the whole development process. However, the i^* goal modelling was a challenge for the teams. Despite

its benefits in gaining a deeper understanding of the single scenarios, supporting a common knowledge ground within the development teams, and documentation, the developers preferred the overview result tables. In their opinion, the tables were more easily to create while providing similar value. In contrast, mastering i^* goal modelling was time-consuming and complex to apply. In terms of improvement suggestions for TrustSoFt itself, the developers proposed to use risk matrices to prioritize trustworthiness goals for the implementation. Furthermore, they proposed to relate the trustworthiness facets to the trustworthiness requirements instead of the trustworthiness goals as this would reduce the overall effort and increase the value for requirements elicitation. Last but not least, trust-related software features should be categorized in their type of feature or importance for implementation to facilitate the implementation process.

8

The Enhanced TrustSoFt Concept

The feedback from the TrustSoFt evaluation is important to enhance TrustSoFt and the accompanying workflow for practitioners. Therefore, the main findings and improvement suggestions are discussed here and considered for the enhanced TrustSoFt concept. The enhanced TrustSoFt concept is depicted in Figure 8.1. It picks up Figure 4.1 from the original TrustSoFt concept on page 46 and highlights the changes of the new version by bold arrows. In addition, the steps of the original version are rearranged in their order. Although the enhanced TrustSoFt concept is again visualised as an iterative procedure by its numbered steps, it is important to point out that the arrow between Steps 3 and 4 highlights a feedback loop. Furthermore, the arrow between Steps 1 and 5 represents the first ideas for trust-related software features based on identified trust concerns. As the evaluation has shown, these first ideas bias the developers for the whole TrustSoFt procedure. Thereby, Step 5 indirectly impacts Steps 2, 3, and 4 leading to indirect feedback loops. As TrustSoFt intends a structured elicitation of each step to create new findings, the bias and its indirect impact are not intended. Therefore, the indirect feedback loops are not visualised in Figure 8.1. Yet, as the evaluation has shown the interplay between Step 5 and Steps 2-4, TrustSoFt practitioners shall consider their ideas for each step independently of the depicted iterativeness. Instead, practitioners shall consciously include feedback loops where needed to enhance their ideas while being open to new ones. Thereby, their working flow shall be supported for developing valuable trust-related software features. Yet, the given workflow structure of the enhanced TrustSoFt concept has emerged during the TrustSoFt evaluation and has proven efficient.

In the following, the TrustSoFt concept with its new aspects is briefly explained in the following step-by-step.

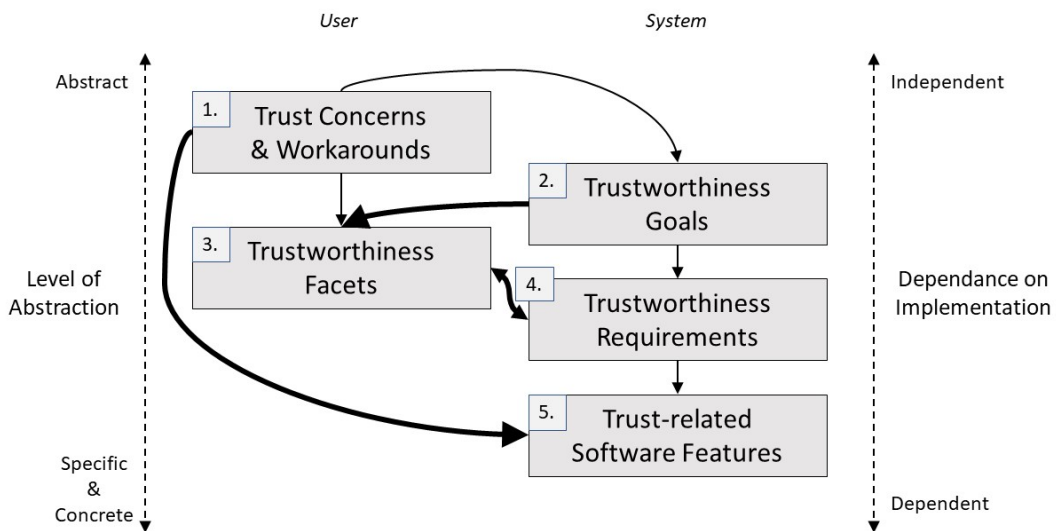


Figure 8.1: The enhanced conceptual TrustSoFt method based on the conducted TrustSoFt evaluation from Chapter 7.

Step 1: Identifying Trust Concerns & Workarounds. Like in the original TrustSoFt method, the enhanced one starts with the identification of trust concerns and workarounds. As the TrustSoFt evaluation revealed the success of user interviews for TrustSoFt, they are highly recommended for this initial step. As preparations for the user interviews, the developers asked for guidance concerning the interview questions due to their lack of experience. The interview questions should focus on trust concerns and workarounds. In addition, the interview can also consider relevant trustworthiness facets for the trust concern. Questions for target users of the application to be developed can be:

- When thinking of the application concept, what concerns would you have when using the application?
- What would you do to mitigate your concerns to continue using the application?
- Can you describe how [the other users / the service provider / the application] should be or behave so that you are not concerned about them anymore regarding [concern stated before in the interview]?

The questions can be used as a starting point to dive deeper with additional

questions that fit the conversation flow with the interviewee to learn more about the trust concerns, workarounds, or especially relevant trustworthiness facets.

The TrustSoFt evaluation has shown, that the interviewees have their own ideas for software features of which some are even trust-related to cope with their concerns. Moreover, the interviews also already inspired the developers to trust-related software features. These first ideas about software features need to be documented. In Figure 8.1, the documentation of the feature ideas is presented by the bold arrow connecting box “1. Trust Concerns & Workarounds” with box “5. Trust-related Software Features”.

Step 2: Deriving Trustworthiness Goals. As a next step, the trustworthiness goals are derived from the identified trust concerns & workarounds. The TrustSoFt evaluation has shown that developers easily determined trustworthiness goals, because the goals are semantically the opposite of the trust concerns identified. Therefore, there are no new aspects for this step in the enhanced TrustSoFt concept.

Step 3: Determining Relevant Trustworthiness Facets. After identifying trust concerns and workarounds, the relevant trustworthiness facets for the single trust concerns need to be identified. Known from the TrustSoFt evaluation, the trustworthiness facets identified in this step improve, on the one hand, developers’ domain knowledge and comprehension of the user. On the other hand, the facets provide them with fundamental knowledge for the later TrustSoFt steps of what qualities must be considered when specifying trustworthiness requirements and trust-related software features that support users’ trustworthiness assessment. The trustworthiness facets identified in this step are especially relevant, because they are in particular needed by users to assess whether the trust concern is significant in a specific user interaction or not. As in the original TrustSoFt concept, the enhanced TrustSoFt concept recommends software engineers to use the overview of trustworthiness facets and the guideline for selecting trustworthiness facets from Chapter 3.2. Referring to the answers of the user interviews of Step 1 is part of the guideline for selecting trustworthiness facets. Yet, the TrustSoFt evaluation has shown that developers valued the trustworthiness goals to derive trustworthiness facets. As the trustworthiness goals equal desired states for the application and counteract

users' trust concerns, the trustworthiness goals support engineers in identifying desired characteristics that the application should realize, check on, or display. This in turn, as explained in Chapter 3.2 in the guideline for selecting trustworthiness facets, is an approach for determining relevant trustworthiness facets. Therefore, the procedure of the developers from the TrustSoFt evaluation is added to the enhanced TrustSoFt concept. The derivation of trustworthiness facets from trustworthiness goals is visualised by the bold arrow between the boxes "Trustworthiness Goals" and "Trustworthiness Facets" in Figure 8.1.

Step 4: Specifying Trustworthiness Requirements and Allocating Trustworthiness Facets to Them For the specification of trustworthiness requirements, while considering trustworthiness facets and goals, the developers of the TrustSoFt evaluation had not stated any drawbacks. They highly appreciated this step, as it concretizes the previously specified abstract information into software behaviour. Therefore, the trustworthiness requirements specification does not involve any new aspects either.

However, there is an adaption for the trustworthiness facet allocation that is now part of Step 4. The TrustSoFt evaluation has shown that the former Step "Facet Allocation to Trustworthiness Goals" of the original TrustSoFt method did not yield a high value. Instead, the trustworthiness facets identified in Step 2 had their biggest value for the specification of trustworthiness requirements in former Step 5, now Step 4. The trustworthiness facets from Step 2 guided developers to meet the qualities required by users for their trustworthiness assessment concerning the specific trust concerns. Thereupon, the developers specified trustworthiness requirements that provide solution approaches to those very trust concerns. Yet, during requirements specification, the developers of the TrustSoFt evaluation reported having related additional trustworthiness facets than the ones determined in Step 2 to the specified trustworthiness requirements. With the additional trustworthiness facets, the developers intended to add extra value to the users' trustworthiness assessment besides mitigating the specific trust concern. In the end, all trustworthiness facets contribute to trustworthiness.

Based on this feedback, the concept of TrustSoFt is adjusted. The step "Facet Allocation to Trustworthiness Goals" of the original TrustSoFt method is omitted. Instead, the trustworthiness facets from Step 3 are directly considered for the specification of the trustworthiness requirements than indirectly via the trustworthiness

goals. Different than in the original TrustSoFt method, at least one trustworthiness requirement of a trustworthiness goal needs to realize one or more of the trustworthiness facets from Step 3. All trustworthiness facets from Step 3 must be picked up by a trustworthiness requirement of a trustworthiness goal so that users are enabled to evaluate the relevant trustworthiness facet in their trustworthiness assessment. If the trustworthiness requirements can be specified to cover additional trustworthiness facets to those from Step 3, the software engineer is welcome to integrate them into the software development process. In that case, the newly considered trustworthiness facets need to be documented, for example in the overview table or i^* goal model. This procedure is visualized in Figure 8.1 by the bold arrow connecting the boxes “Trustworthiness Requirements” and “Trustworthiness Facets”.

Step 5: Deriving Trust-Related Software Features. Concerning the derivation of trust-related software features, the evaluation has shown that developers perceive this step as very beneficial for concretely planning the application and preparing the implementation. They reported no drawbacks. However, to better prepare the implementation of trust-related software features, the developers of the TrustSoFt evaluation proposed to categorize the features either in their importance (e.g., MVP and NTH) or type of implementation (e.g., algorithm, design, information).

In terms of the TrustSoFt method, former Step 6, here Step 5, serves as a first abstract draft of how trust-related software features realize trustworthiness requirements while reflecting or addressing trustworthiness facets for users’ trustworthiness assessment. The abstract draft of trust-related software features shall guide software engineers in the early software development phase to a commitment to the following phases of the software development life cycle that are more solution-oriented.

On these grounds, the developers’ improvement proposals for trust-related software features are plausible insofar as they needed more information for the implementation of the trust-related software features in a prototype. Therefore, the definition of trust-related software features is concretised in chapter 9 by information about different types of trust-related software features. Furthermore, a catalogue structure is introduced by which features can be structured and collected in a catalogue for Software Product Line Engineering. In terms of feature structure, the scope of how software features can be designed exceeds the capabilities of goal mod-

els. Hence, a new framework is introduced that shall follow up TrustSoFt. The new framework are so-called feature models that are adapted to the context of online trustworthiness assessments and introduced in Chapter 11. By feature models, trust-related software features can be modelled with a higher level of detail than in the i^* goal models and overview tables. The improvement proposal about the feature categorization of the TrustSoFt evaluation is picked up in the concretised definition of trust-related software features (Chapter 9) and the feature models for online trustworthiness assessments (Chapter 11). Step 5 of the enhanced TrustSoFt method is regarded as input for the feature models.

The Model-Based and Table-Based Approach: For the visualisation and documentation of the TrustSoFt elements, the original TrustSoFt method uses i^* goal modelling as a model-based approach and overview tables. The overview table supports quickly noting all results. i^* goal modelling supports the specification process and may yield new TrustSoFt elements. Both approaches serve as discussion bases for software engineering teams.

In the TrustSoFt evaluation, all developers were convinced of the overview table as it serves its purpose effortlessly. However, the i^* goal models evoke ambivalent opinions in the development teams. Despite their benefits of increasing domain understanding and the intended purpose they fulfill, for many developers of the TrustSoFt evaluation, the effort of learning the notation and creating the goal models was too high compared to the gained value.

This feedback needs to be abstracted insofar as the modeling expertise, skill, and preference of engineers varies. Depending on a person's working style, some tools are more efficient than others. Therefore, for the enhanced TrustSoFt method, the overview tables are highly recommended for the TrustSoFt application. In terms of the i^* goal modelling, it can be regarded as a supportive tool. Depending on the engineer's ability, time, and costs, goal modelling can additionally be used for TrustSoFt to provide extra value and new insights.

9

Trust-related Software Features - A Concretised Definition

In course of the TrustSoFt evaluation, it became apparent that additional information about trust-related software features is valuable for their implementation. On these grounds, the definition of trust-related software features is concretised by three types in this chapter. The trust-related software features are further discussed in Chapter 10 concerning a catalogue for trust-related software features and in Chapter 11 about feature models in the context of trustworthiness assessment.

Trust-related software features are attributable to software features and digital nudges that have been introduced in Chapter 2.3. As software features, they are user-accessible, meaning that they are perceptible for users in the user interface. Furthermore, they can be designed as a digital nudge by meeting nudging criteria (see Chapter 2.3 or next Chapter 10). Trust-related software features are trust-related because they support users in assessing the trustworthiness of the three CMI parties i) user, ii) application, and iii) service provider. For that purpose, trust-related software features reflect the trustworthiness facets of these parties in the front end of systems to enable users' online trustworthiness assessment.

In addition, trust-related software features address the three challenges of trustworthiness assessment that have been introduced before in Chapter 1. The challenges are depicted in Figure 9.1. Each challenge is addressed by a solution approach, which in turn is picked up by a special type of trust-related software feature. The three types of trust-related software features have been introduced in Paper 8 and are explained in the following.

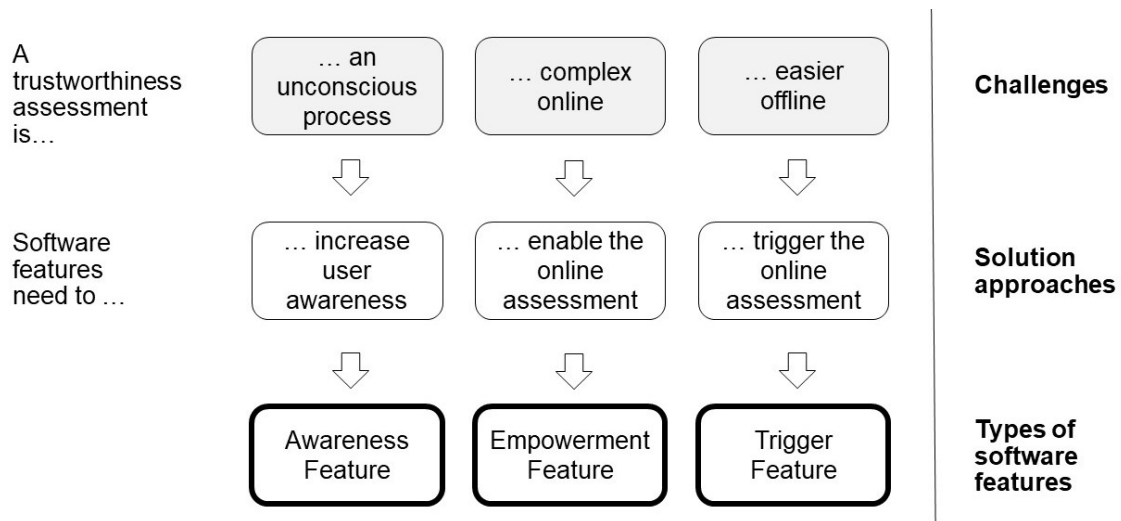


Figure 9.1: Challenges of online trustworthiness assessments, solution approaches for software features and resulting features types

Awareness Feature A characteristic of trustworthiness assessments is that they are usually conducted unconsciously by individuals [260]. Their unconscious process is insofar a challenge in the CMI context as the trustworthiness assessment is decisive for users to decide whether to meet strangers in the offline world or not [328]. Only if a user trusts another one enough, the user evaluates the risk of dangerous offline encounters as irrelevant and believes in a promising interaction. Therefore, conducting trustworthiness assessments consciously enhances the trust-building and decision-making process [28]. On these grounds, so-called **Awareness Features** shall sensitise users to trustworthiness assessments and their relevance. With the purpose to increase user awareness, Awareness Features are digital nudges that strive to provide users with information about trustworthiness assessments.

Empowerment Feature Another challenge of trustworthiness assessments is that conducting them in an online environment is different from performing them offline. Offline cues by which individuals assess the trustworthiness of other parties do not exist online (e.g., body language), are manifested in a different form than offline (e.g., appearance in reality or pictures), or are vulnerable to manipulation (e.g., profile information) [84]. Hence, users must get used to a new trustworthiness assessment process that might not come as naturally as the offline assessment and is partly not possible at all. Hence, conducting an online trustworthiness assessment is complex. Therefore, software engineers need to consider this challenge by supporting and enabling users to perform the trustworthiness assessment online.

This purpose is addressed by **Empowerment Features**. Empowerment Features present and reflect the trustworthiness facets of parties with whom users are interacting. Depending on their design, they can be a digital nudge. Empowerment Features are especially valuable when software engineers make sure that they reflect trustworthiness facets in a way that is resistant to manipulation.

Trigger Feature The last challenge of trustworthiness assessments discussed here refers to the complexity of performing the assessment online compared to offline described in the previous paragraph. Online dating users have reported skipping an extensive online introduction and the online trustworthiness assessment due to the complex process [73]. Instead, they want to meet other users in the physical world as fast as possible to get to know them in person and perform the trustworthiness assessment offline [73]. These findings are supported by the interviews conducted in Paper 9. However, conducting the trustworthiness assessment offline means agreeing on offline encounters without the measure to check on others' trustworthiness beforehand. It is assumed that this may result in higher risk then. To counteract the skipping of the online trustworthiness assessment, **Trigger Features** shall initiate users' trustworthiness assessment via the CMI application. Thereby, Trigger Features are a digital nudge. To trigger trustworthiness assessment, Trigger Features must incentivise trustworthiness assessments and encourage them by providing information about their benefits, for example.

10

Catalogue for Trust-Related Software Features

In software engineering, catalogues are a highly valuable tool. Catalogues are beneficial as they pose collections of entries that are searchable, identifiable, examined, and evaluated in terms of the catalogue's dimensions [56]. The creator of a catalogue determines the dimensions. They are for example the priority, functionality, relevance, or meta-information a catalogue entry contains. For software engineering, catalogues are appreciated for the purpose of reusability, documentation, and traceability [56]. They support software engineers and designers in developing Software Product Lines. There are existing catalogues that propose reusable solution approaches in form of software requirements or software features, such as the User Interface Design Pattern Library [186] or Welie.com - Patterns in Interaction Design [307].

Since software features have not yet occurred as trust-related yet, establishing a catalogue of them during their specification is valuable for developing user-centered applications where trust is central. As feature specification is performed for a specific domain or application, a catalogue of trust-related software features is tailored to the stakeholders' needs in that domain or can even convey a brand image of an application [209].

In this work, the here presented catalogue for trust-related software features is established during the creation of feature models for trustworthiness assessment (see next Chapter 11). It can be regarded as additional information given to each feature model and its leaves. When the catalogue is established, it further supports the configuration of Software Product Lines by the feature models. Therefore, this chapter introduces the catalogue structure for the catalogue of trust-related software

features and already refers to elements of the feature models introduced in Chapter 2.7.

The catalogue structure for trust-related software features has been introduced in Paper 8. It is used for including trust-related software features and their assets in a catalogue. As introduced in Chapter 2.3, Feature assets describe individual feature components that, when combined, make up a complete software feature [189]. Assets can be for example algorithms (e.g. matching algorithm in online dating), design elements (e.g. graphical symbol, colour), information necessary for or presented by a feature (e.g. user data), and interaction elements (e.g. confirmation request). Sometimes, assets can be at least one of these categories at the same time, such as a confirmation button (design and interaction element).

By including feature assets in a catalogue, the catalogue structure provides detailed additional information about each feature component. The additional information facilitates the configuration of software features and their implementation. In addition, the catalogue structure enables selective configuration of software features, which enables the variability of trust-related software features.

The catalogue structure consists of two parts. The first part is about the basic information of a trust-related software feature, which is depicted in Figure 10.1. It shows the exemplary software feature “catfish protection” that addresses the trust concern “catfishing” introduced in Chapter 4. The second part of the catalogue structure involves detailed information about a feature asset. It is presented in Figure 10.2.

Basic Information. The “Basic Information” of the catalogue structure for trust-related software features summarizes the key data of a trust-related software feature and its assets. It gives information about an abstract solution approach. Therefore, the basic information can be assigned to the concept feature of a feature model (see Chapter 2.7). It provides the informational background to the concept feature, the trust-related software features on the first layer afterwards, and their subsequent asset. The basic information is valid for the whole feature model. The basic information includes the name of a trust-related software feature as well as a description of the problem it addresses. Keywords provide an overview of the issue a feature covers. Furthermore, the software requirements that were specified in

Basic Information	
Name	<i>Catfish Protection</i>
Problem	<i>Some social media users are catfish by using fake profiles for fraudulent reasons</i>
Keywords	<i>Catfish, protection, prevention</i>
Requirements	<i>Preventing catfish attacks, protecting users from catfish, warn users about catfish, identify catfish</i>
Problematic characteristics	<i>dishonesty, unpredictability</i>
Desired characteristics	

Figure 10.1: First part of the catalogue structure for trust-related software features. This part covers the basic information of a trust-related software feature. The figure is taken over from Paper 6 [37].

TrustSoFt and which shall be realised by a software feature are part of the catalogue structure. In addition, the problematic and desired characteristics for the guideline for selecting trustworthiness facets (see Chapter 3.2) are included.

Regarding the displayed example in Figure 10.1, the trust-related software feature that addresses the trust concern of catfishing is called “catfish protection”. Catfish protection addresses the problem of social media users aka catfish that use fake profiles for fraudulent reasons. Keywords of this issue and the software feature are “catfish”, “protection”, and “prevention”. Catfish protection should realise trustworthiness requirements like “preventing catfish attacks”, “protecting users from catfish”, “warn users about catfish”, or “identify catfish”. In terms of problematic characteristics, “dishonesty” and “unpredictability” are identified (see Chapter 3.2).

Asset Information The second part of the catalogue structure is called “Asset Information”. It contains information by which assets are categorised, for example regarding design and nudging criteria. This allows software engineers to structurally configure software functions to their liking. In the context of feature models, each leaf underneath the concept feature is an asset for which the asset information is filled out. Each asset has an entry in the catalogue by means of the asset information

Asset Information	
Feature type	<input type="checkbox"/> Awareness <input type="checkbox"/> Trigger <input type="checkbox"/> Empowerment
Target group for online trustworthiness assessment	<input type="checkbox"/> Users <input type="checkbox"/> Application <input type="checkbox"/> Service Provider
User Accessibility	<input type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open choice architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for individuals	...
Trustworthiness facets for technology	...
Trustworthiness facets for service provider	...

Figure 10.2: Second part of catalogue structure for trust-related software features. This part covers the information on an asset of a trust-related software feature. The figure is taken over from Paper 6 [37].

of the catalogue structure. Thereby, a set of asset information is created that is associated with the basic information and in total creates the catalogue for trust-related software features.

The asset information of the catalogue structure contains dimensions, which in turn are defined by characteristics. Several characteristics of a dimension can be applicable at a time for an asset. The dimensions and characteristics can be checked in the catalogue structure for software product line configuration and when managing the variability of software features.

The first dimension is “feature type” which refers to the three types of trust-related software features introduced in the previous Chapter 9. In coherence with the three types of trust-related software features, the characteristics of the dimension

feature type are “Awareness”, “Empowerment”, and “Trigger”. The next dimension of the catalogue structure is called “Target group for the online trustworthiness assessment”. This dimension notes which of the three CMI parties “users”, “service provider”, or “application” is targeted by an asset for users’ trustworthiness assessment. Another dimension is “user accessibility”. It picks up the definition of software features that features are perceptible for users in the user interface. Yet, some feature assets are not user-accessible but necessary for the creation of a software feature. Therefore, the dimension “user accessibility” has the characteristics “yes” for an asset being user-accessible and “prerequisite” when an asset contributes to user accessibility while not being user-accessible. An example of a prerequisite asset is an algorithm that may be an underlying element for a user interface element presenting the algorithm’s results. To be more concrete with the type of asset, the dimension “asset category” is included in the catalogue structure. It refers to the elements as which an asset can appear within a trust-related software feature. The characteristics of the dimension “asset category” are “algorithm”, “design”, “information”, and “interaction”. The categories were recommended by the feedback of the TrustSoFt evaluation from Chapter 7. Yet, depending on the type of technology or application (e.g. wearables), further categories can be added.

The next dimension in the catalogue structure is called “nudging criteria”. The dimension nudging criteria refers to the guideline for designing nudges by Thaler and Sunstein [298] and the Fogg Behavioural Model [101] that have been introduced in Chapter 2.3. The characteristics “open choice architecture”, “guiding information”, “explaining behaviour patterns”, and “solution approaches to unfavourable behaviour” are recommendations for developing nudges [225, 298]. The characteristics “considering [the] motivational state” of users, “considering user ability”, and “presenting a behavioural trigger” belong to the Fogg Behavioural Model to persuade users for their good [101]. The characteristics of this dimension serve as a checklist if an asset contributes to a software feature being a digital nudge. If an engineer intends to develop a digital nudge, the engineer should configure assets in a way that all nudging criteria are met for the trust-related software feature. The last three dimensions of the catalogue structure involve the trustworthiness facets for individuals, technology, and the service provider. Those facets that are related to an asset are included in the respective dimension in the catalogue structure.

The most efficient way to use the catalogue for trust-related software features

is to establish the catalogue as a digital tool. Thereby, the catalogue structure can be used for Software Product Line Engineering as a search interface. Software engineers can activate those dimension characteristics in the search interface that they are looking for to retrieve suitable catalogue entries. The same is for entering trustworthiness facets by which software engineers can retrieve assets that enable online trustworthiness assessments for specific trust concerns.

As an example, software engineers may look for assets to develop a feature for catfish protection to address the catfishing problem introduced in Chapter 3.2.2. For that purpose, they may look up the catalogue for catfish protection, whose overall information is stored in the basic information catalogue structure. By selecting characteristics in the digital catalogue structure of the asset information, suitable assets can be displayed to the software engineer. An example of selecting characteristics in the catalogue structure of the asset information is depicted in Figure 11.4 on page 137, which represents the asset “green check mark” for catfish protection.

In the next chapter, feature models for online trustworthiness assessments are introduced. Feature models and the catalogue for trust-related software features complement each other for the configuration of software product lines.

11

Method for Establishing Feature Models for Online Trustworthiness Assessments

Software requirements allow software engineers great freedom of action of how to implement specified software behaviour and design applications in form of software features [329]. However, this freedom of action may complicate a concrete development as it leaves room for interpretation [329]. Concerning TrustSoFt, room for interpretation is given for the specification of trust-related software features. One and the same trustworthiness requirement can be realized in a multitude of designs, offering different kinds of interaction and providing various forms of information for users. Thereby, different types of software features can be developed, which realise the same system behaviour but can affect users in different ways and guide them to different actions. Yet, this freedom of interpretation for developing software features has the benefit of individualizing software applications for a unique user experience [210]. Therefore, the freedom of developing software features must be managed in a way that the benefits can be efficiently used while the drawback of the indefiniteness of feature options is controllable.

For that purpose, Paper 8 introduced the *method for establishing feature models for online trustworthiness assessments*, which is further explained in this chapter. The method for establishing feature models for online trustworthiness assessments is intended as a continuation of TrustSoFt. It picks up the output of the TrustSoFt method so far, which is an abstract description of trust-related software features and the feature elements of the i^* goal models. The method results in feature models for trustworthiness assessments. The method and the resulting feature models shall aid software engineers in viewing and organizing software features while having an overview of their effects on users' trust-building. The method and the resulting feature models are accompanied and supported by the catalogue of trust-related

software features. The catalogue of trust-related software features provides additional information about each feature and asset that is part of a feature model. The resulting feature models shall support software engineers in not being overwhelmed by feature options and organizing them. At the same time, the feature models shall help software engineers in overcoming the uncertainty of what feature is the best to implement for which requirement while pursuing a targeted user experience [263].

The method for establishing feature models for online trustworthiness assessments is depicted in Figure 11.1. It consists of three main steps, which are 1) feature model creation, 2) validation of software features, and 3) configuration of a software product line. In the following, each step is explained. The first step, feature model creation, relies on an extended feature model notation for trustworthiness assessments. The extended feature model notation is also introduced in the following. After the method and its three steps are introduced, an application example for catfish protection is presented.

11.1 Features Model Creation

Feature model creation is the first step of the method for establishing feature models for online trustworthiness assessments. In this step, feature models are created by using an extended feature model notation for the context of trustworthiness assessments that is introduced in the next paragraph.

Feature model creation consists of two sub-steps. These are 1.1) feature modelling and 1.2.) the facet attribution process. The facet attribution process is again split up into two sub-steps, which are 1.2.1) the allocation phase and 1.2.2) the propagation phase. All the steps are explained in the next paragraphs. Furthermore, for each step, validation conditions are introduced at the end of the respective section. Thereby, practitioners can check for the correctness of feature model creation.

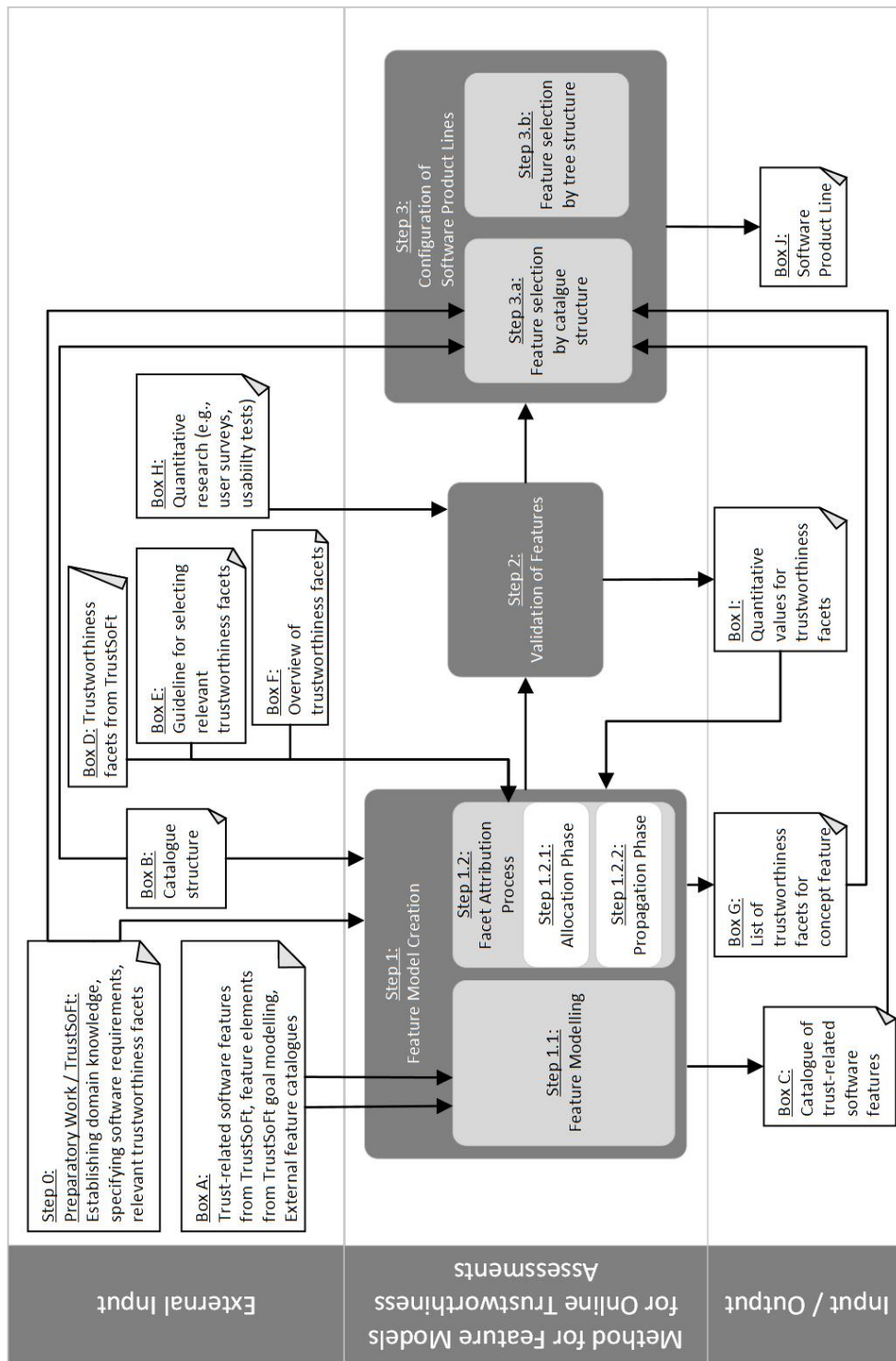


Figure 11.1: Method for establishing feature models for online trustworthiness assessments. The figure is taken over from Paper 6 [37].

11.1.1 Extended Feature Model Notation for Trustworthiness Assessments

The extended feature model notation for trustworthiness assessments is based on the original feature model notation introduced in Chapter 2.7. It has two new aspects. The first new aspect is the consideration of the three types of trust-related software features as introduced in Chapter 9 leading to new terminology in the area of the feature models. The second new aspect is the trustworthiness facet (see Chapter 3.2). The extended feature model notation is depicted in Figure 11.2.

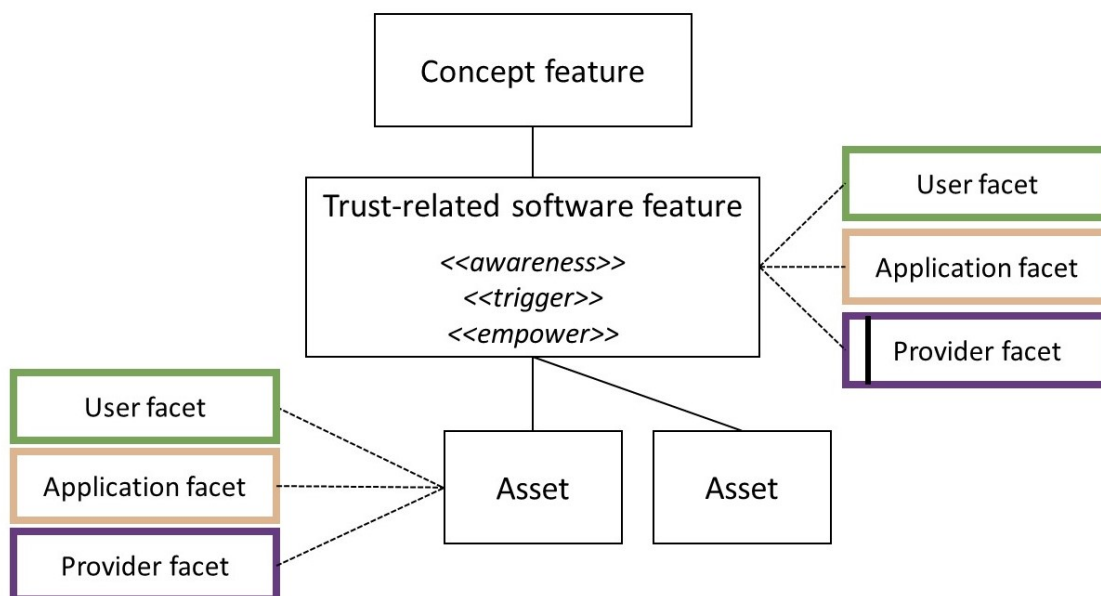


Figure 11.2: The extended feature model notation for trustworthiness assessments.

The extended feature model notation involves a new naming of the leaves in a feature model (see Figure 11.2). As intended by the original feature model notation, the root of the tree is called the concept feature. It is the most abstract solution approach. The first layer of the feature tree is the trust-related software feature. It is the most abstract asset of the concept feature proposing solution approaches for realising the concept feature. The trusted software feature is assigned a label depending on the type to which the trusted software feature can be assigned. The labels are «awareness» for an awareness feature, «trigger» for a trigger feature, and «empower» for an empowerment feature. To provide an encompassing solution approach to the concept feature, software engineers can make sure that every feature

type is proposed at least once. Thereby, software engineers can ensure a feature variety to achieve a high impact on users for their online trustworthiness assessment. After the trust-related software features in the first layer of the feature tree follow the assets. With increasing tree layers, the assets are more concrete in the description of how the solution should be implemented.

To give an example, a concept feature can be “catfish protection“. This example is a solution approach referring to the catfishing problem explained in Chapter 3.2.2. The concept feature “catfish protection“ can be realized by various software features. One can be for example “identity verification «empower»“, by which the application checks on user identity and displays the result in the graphical user interface. Thereby, users are empowered to assess whether a person is honest, which is an indicator and facet of trustworthiness. The example is also depicted in Figure 11.3 on page 136.

In addition to the three types of trust-related software features, it is relevant for the realisation of the trustworthiness assessment that software engineers know why a feature is trust-related. Since features are trust-related because they either realise or reflect a trustworthiness facet from one of the stakeholders i) user, ii) application, or iii) service provider, the trustworthiness facets are added to the feature models. When extending feature models by new elements, this usually is performed by attributes ([23], see Chapter 2.7). Therefore, the trustworthiness facets of users, the application, and the service provider are linked by dotted lines to the respective asset that realises or reflects them (see Figure 11.2). To distinguish between the three stakeholder facets, they receive different frame colours: the user facet is modelled in green, the application facet in orange, and the provider facet in purple. Thereby, software engineers know what features impact computer-mediated interpersonal trust, system trust, and brand trust (see Chapter 3). In Figure 11.2, it is exemplarily depicted that the trust-related software feature and the asset on the left are associated with trustworthiness facets. Furthermore, trustworthiness facets also underlie optionality as assets do. The principle of optionality for trustworthiness facets relates to the propagation phase of the facet attribution process and is explained in Section 11.1.3.2. For the extended feature model notation, optional trustworthiness facets are marked by a line on the left side, as is the case for the provider facet of the trust-related software feature in Figure 11.2.

Concerning the catfish protection example from above, the trustworthiness facet

“honesty“ is linked as an attribute to the feature “identity verification «empower»“. The frame of honesty is green, as the feature identity verification enables users to assess the honesty of users.

11.1.2 Feature Modelling

Feature modelling makes use of the feature modelling notation (Chapter 2.7) and the extended feature modelling notation from the previous paragraph. It is accompanied by establishing the catalogue of trust-related software features (Chapter 10), which is an output of feature modelling (Figure 11.1, Box C). For each new entry of the feature model, the respective part of the catalogue structure is filled out (Box B). Thereby, feature modellers further define the asset they are including in the feature model. Furthermore, the catalogue additionally documents each asset of a feature model.

To start feature modelling, software engineers have to do preparatory work beforehand (Figure 11.1, Step 0). The preparatory work involves gaining an understanding of the problem space, specifying software requirements, and knowing what trustworthiness facets are relevant for the problem to be mitigated. It is covered by TrustSoFt (see Chapter 8). TrustSoFt provides the first descriptions for trust-related software features and their assets. The descriptions of trust-related software features from TrustSoFt and the feature elements from the TrustSoFt goal models are input for feature modelling (Box A). They need to be considered within the feature models.

In addition to the output of TrustSoFt and the TrustSoFt goal models, software engineers can use external catalogues of software features or nudges (Box A) as inspiration for feature modelling. External catalogues are for example the User Interface Design Pattern Library [186] or the DINU model [225].

For feature modelling, practitioners need to adopt the intention that they create feature models for the trustworthiness assessment of the users. This means that they must aim for trust-related software features that reflect or address the trustworthiness facets of TrustSoFt. For that reason, practitioners need to generate assets in the feature models that either hold the respective trustworthiness facets to realise the trustworthiness assessment or contribute to the establishment of such trust-related software features. As an example, an algorithm is not user-accessible,

because it runs in the back end. Yet, it is often necessary for assets in the front end by which users can perform their trustworthiness assessment. Therefore, both must be included in the feature model. Although assets other than those relevant for assessing trustworthiness can be included in the feature models, the feature models are limited to just those assets. This places the focus of the feature models on the context of the trustworthiness assessment.

As a starting point, software engineers agree on a problem to which the feature model shall provide solution approaches. In doing so, they fill out the basic information of the catalogue structure (Figure 11.1, external input, Box B, see Figure 10.1 on page 115). Based on the problem, the concept feature, which represents an abstract solution approach to the problem, can be derived. Afterwards, software engineers need to refine the concept feature into high-level assets that can be regarded as trust-related software features that realise the concept feature. In doing so, software engineers are advised to cover all of the three feature types «awareness», «trigger», and «empowerment». Then, feature modelling follows the same procedure as described in Chapter 2.7 with the addition that software engineers consider the identified trustworthiness facets from TrustSoFt throughout the modelling process. The objective is to create features that reflect or realise relevant trustworthiness facets in the user interface. With each of the modelled layers, assets are more and more refined in terms of algorithm, information, interaction, or design. When creating feature assets in feature models, software engineers may consider the brand image in their design or can convey the brand message. Furthermore, they can perform feature modelling while respecting the business strategy of the service provider.

For each feature, software engineers have to fill out the asset information of the catalogue structure (Box B, see figure 10.2 on page 116). During feature modelling and by filling out the catalogue structure, software engineers develop a *Catalogue of Trust-related Software Features* that is tailored to the respective application and involved brand or business strategies (output, Box C).

Validation conditions for Step 1.1: Feature Modelling

- The feature model contains a concept feature that poses a high-level solution

to the trust concern of users.

- The concept feature has an entry in the basic information of the catalogue for trust-related software features.
- The feature model contains at least one awareness feature, one trigger feature, and one empowerment feature.
- Each asset in the feature model has an entry in the asset information of the catalogue of trust-related software features.
- The feature elements of the associated TrustSoFt goal models are included as assets in the feature model.
- The proposal for trust-related software features from TrustSoFt is addressed by assets in the feature model.
- Only assets are part of the feature model that either i) hold a trustworthiness facet of TrustSoFt or ii) are necessary for establishing the trust-related software feature to address or reflect the trustworthiness facet of TrustSoFt.
- One user-accessible asset is mandatory.

11.1.3 The Facet Attribution Process

The facet attribution process (Step 1.2) realises the second new aspect of the extended feature model notation, which is including the trustworthiness facets in the feature models. To attribute the trustworthiness facets to the assets of the feature model, software engineers must perform Step 1.2.1: the *allocation phase* and Step 1.2.2: the *propagation phase*. The two steps are explained in the following. Furthermore, for each sub-step, validation conditions are introduced at the end of the respective section.

11.1.3.1 The Allocation Phase

In the facet allocation phase, trustworthiness facets are related to each asset of a feature model and added as their attributes. The identified trustworthiness facets from TrustSoFt serve as input (Box D). They are the trustworthiness facets relevant

to the problem that is addressed by the respective feature model. Thus, they must be reflected by at least one user-accessible, mandatory asset to be incorporated into the trust-related software feature. That is because the trustworthiness facet must be assessable for users in the user interface so that the targeted trust concern and the associated problem can be evaluated for relevance for the interaction with a social media party. The user accessibility of assets can be looked up in the catalogue for trust-related software features.

However, there are assets that are relevant to establishing a trust-related software feature but are not designed to address the trustworthiness facets identified by TrustSoFt. Software engineers have to decide based on their expertise and logical thinking whether an asset can reflect or realise a trustworthiness facet from the TrustSoFt process. Still, an asset may address other trustworthiness facets as well and, thus, additionally supports trust-building.

An example is given by the use case from above about the empowerment feature identity verification. It is associated with the trustworthiness facet *honesty* because based on the software feature, users can derive whether other users have been honest about their identity. In addition to the trustworthiness facet *honesty*, the trust-related software feature identity verification further supports users in their trustworthiness assessment of other users. Reflecting on further effects of the feature, it becomes apparent, that users can be assessed whether they have complied with the request of the application to perform identity verification. If they have, it can be concluded that they have *integrity* with the online dating application. The interview study from Paper 9 has shown that by performing identity verification and showing integrity, the trustworthiness of other users increases. The process of facet allocation is demonstrated in more detail in the exemplary application of the method in Section 11.4. Besides logical thinking, the guideline for selecting relevant trustworthiness facets can also support the process of facet allocation (Box E, Chapter 3.2.2).

Yet, assets that are more concrete in terms of design, interaction, algorithm, or information may also convey further trustworthiness facets than the ones identified with TrustSoFt. In TrustSoFt, these facets are not identified, because the focus is on the abstract problem and not on a concrete design level. However, feature models provide much more detail about software features than TrustSoFt and the TrustSoFt goal models. Therefore, software engineers can consider on a design level

how concrete feature assets impact users in their trustworthiness assessment. As an example, a warning text message can be formulated in a benevolent or insistent manner leading users to feel benevolently supported for a decision or anxiously pushed into action. Thus, each asset may involve different and new trustworthiness facets than those of the other assets and those identified in TrustSoFt. By selecting trustworthiness facets as attributes, software engineers can determine that an asset is designed with the aim that users perceive the intended facet. For that purpose, the overview of trustworthiness facets (see Appendices A, B, and C on pages 307, 310, and 313) serves software engineers as external input (Box F). In many cases, the selection of trustworthiness facets for each feature follows the intention and expertise of the software engineer. It is up to the engineer's evaluation whether an asset can be designed in a way that it realises or reflects an intended trustworthiness facet. By selecting trustworthiness facets in that manner, brand image can be actively established.

The facet allocation phase (Step 1.2.1) begins with a trust-related software feature and continues downwards asset by asset. When continuing the facet allocation downwards the tree structure, it is likely that the trustworthiness facets of parent and child assets differ, as it is explained above. This relates to the fact that concrete design options (e.g., graphical symbols, interaction elements) are detached from the problem that the abstract feature assets in the higher tree layers (e.g., identity verification) address. Hence, additional trustworthiness facets than the relevant ones for the respective problem can be considered in software design. Trustworthiness facets related to an asset must be included in the asset's asset information in the catalogue for trust-related software features.

Validation conditions for Step 1.2.1: Allocation Phase

- Each trustworthiness facet that has been identified by TrustSoFt must be related to at least one user-accessible, mandatory asset.
 - If there is only one user-accessible, mandatory asset, it must be able to hold all trustworthiness facets identified by TrustSoFt.

- Each trustworthiness facet that is linked to an asset is documented in the asset's asset information of the catalogue for trust-related software features.

11.1.3.2 The Propagation Phase

In the propagation phase, software engineers reflect on facet differences among the feature assets and transfer trustworthiness facets among features due to the inheritance principle. In object-oriented programming, the inheritance principle describes the mechanism in which attributes are derived from one class to another based on the classes' hierarchy [205]. Child classes inherit the attributes from the parent class. Regarding the assets of the feature models, the inheritance principle is asserted differently due to the way feature configuration is explained in Chapter 2.7. With feature models, software features are configured by combining assets starting from the concept feature to the leaves of the tree depending on their optionality. By selecting assets as being included in a trust-related software feature, the trustworthiness facets of all configured assets are valid as users are exposed to the whole feature. In conclusion, an asset higher in the hierarchy inherits the trustworthiness facets of the assets lower in the hierarchy when they are configured to one software feature. The configuration of trust-related software features is explained in the next Section 11.3.

The propagation phase of the facet attribution process is an important step to realise a digital catalogue of trust-related software features. By documenting the trustworthiness facets that are allocated to complete trust-related software features, the configuration of software product lines is facilitated. Therefore, propagated trustworthiness facets are included in the asset information of the catalogue for the respective parent asset. Then, software engineers can search the catalogue for software features that impact users' trustworthiness assessment in the sense of specific trustworthiness facets. Based on the catalogue findings, they can decide which feature to implement in the application to provide users with a special user experience in terms of their trustworthiness assessments.

The propagation (Step 1.2.2) is performed from the top down to the root of the tree, branch by branch, until all leaves have been propagated. The software engineer has to check on differences among the trustworthiness facets of the leaf feature and its parent feature. All trustworthiness facets that the parent feature does not own

but the child feature, are added as attributes to the parent feature. In this process, software engineers must consider the optionality of child features. If a child feature is optional for configuration, the propagated trustworthiness facet receives a line on its left side (see Section 11.1.1, Figure 11.2). Otherwise, meaning that an asset is mandatory, the trustworthiness facets are propagated without a line. Concerning the catalogue of trust-related software features, the entry of an optional propagated trustworthiness facet receives a “(o)” behind the facet, meaning that it is optional.

The propagation ends with the trust-related software feature. The concept feature is left out of the propagation, because the number of trustworthiness facets in form of attributes would undermine the clarity of the feature model. Instead, all trustworthiness facets shall be documented in a list (Box G). Like the basic information of the catalogue structure, the list of trustworthiness facets for the concept feature provides an overview of information for the feature model. When establishing the list of trustworthiness facets, descriptive information about the frequency of occurrence for each facet can be included. This information supports software engineers in evaluating the impact a concept feature has on users’ trustworthiness assessment. It is the first step to feature model validation.

Validation conditions for Step 1.2.2: Propagation Phase

- Propagated trustworthiness facets are included in the asset information of the parent asset of the catalogue for trust-related software features.
 - Optional, propagated trustworthiness facets are marked with a “(o)” within the asset information of the catalogue for trust-related software features.
- Propagated trustworthiness facets that stem from an optional child asset have a line on the left side of their box.
- Propagated trustworthiness facets that stem from a mandatory child asset do not have a line on the left side of their box.
- The list of trustworthiness facets for the concept feature contains all trustworthiness facets within the feature model once.

11.2 Validation of Trust-related Software Features in Feature Models

The validation of trust-related software features in feature models describes the process of measuring the features' impact on user perception and the online trustworthiness assessment. For that purpose, the validation involves testing to what extent users really associate trust-related software features with the allocated trustworthiness facets. The feature model validation is performed after feature model creation (Figure 11.1, Step 2). It is based on the proposal from Arnowitz et al. [12], who suggest usability tests in which people experience single features in prototypes (Box H). Usability testing provides a testing method in which participants can rate features based on their trustworthiness facets on appropriate scales. The participants can be asked how far they perceive that a feature conveys a particular trustworthiness facet. For some trustworthiness facets, scientific scales already exist (Box H), for example for ability, benevolence, integrity, and predictability [50]. For those trustworthiness facets for which no scientific scale exists yet, software engineers may include a survey item that asks participants directly whether a feature can be related to these facets. Based on the user ratings, quantitative attribute values can be calculated by which the user perception of the various facets through the features is comparable (Box I). The resulting facet values are added within the feature models in the attributes.

For further validation, additional attributes are valuable for measuring the success rate of trust-related software features according to their feature type. For awareness features, user awareness is an appropriate attribute to measure whether a trust-related software feature impacts how aware users are about the relevance of the trustworthiness assessment. Concerning trigger features, the conversion rate for trustworthiness assessments is a meaningful attribute. The conversion rate is a value representing how many percent of the users have performed a trustworthiness assessment after interacting with the trigger feature. Thereby, it can be tested to what extent a trigger feature is successful. For empowerment features, their usefulness in assessing the trustworthiness of involved parties is an indicator of how well the system supports the online trustworthiness assessment.

11.3 Configuration of Trust-Related Software Product Lines

The last step of the method is the configuration of a software product line by using the feature models for online trustworthiness assessment (Step 3). The configuration is performed based on the original feature modelling notation explained in Chapter 2.7. The feature models provide an overview of the assets by their tree structure. They guide software engineers in the configuration by visualising the optionality of the assets for inclusion in the trust-related software features (Step 3.b). The catalogue structure is an additional tool for configuration that provides further information for targeted asset selection by the asset characteristics (Step 3.a, Box B). Especially if the catalogue structure is turned into a digital tool, it can serve software engineers as a search interface for selecting specific trustworthiness facets or other characteristics and thereby select assets. The feature models with their tree structure and the catalogue can be regarded as complementary tools for the configuration of trust-related software features.

There are two ways of approaching the configuration. The first approach is to first check on the feature tree by following the branches downwards. At each layer, the software engineers can decide what asset to implement based on the engineer's liking and the assets' optionality. The feature tree can be used as a checklist to highlight the included assets, for example by colouring the box of the asset. Every time the engineer decides to include an asset in a trust-related software feature, the engineer checks the asset information in the catalogue for asset characteristics. Thereby, the engineer keeps track of the involved trustworthiness facets and whether those, that TrustSoFt has identified, are addressed by the trust-related software feature in the end.

The second approach is by first checking on the catalogue structure for the asset characteristics that the engineer wants to be included in the respective trust-related software feature (e.g., the trustworthiness facts of TrustSoFt). Those assets, that hold the targeted characteristics, are marked in the feature tree. After the catalogue has been checked, the software engineer can view the feature model for the marked assets. Starting from the trust-related software feature, the engineer follows the branch downwards to select those assets that are marked as well as further assets that meet the engineer's liking and the optionality.

The output of the configuration is a tailored software product line (Box J).

Validation conditions for Step 3: Configuration

- At least one asset that holds a trustworthiness facet of TrustSoFt is included in the trust-related software feature.
- The trust-related software feature is user-accessible.
- Assets that are included in a trust-related software feature are highlighted in the feature model.

11.4 Application Example for the Method for Establishing Feature Models for Trustworthiness Assessments

For demonstrating the method for establishing feature models for online trustworthiness assessments, the example of catfish protection and the feature “identity verification” is picked up and presented in detail. Identity verification helps to resolve the uncertainty of whether another user has created a fake profile [39]. It is known to be an interactive tool for self-presentation, which increases users’ reputation and allows them to rate the trustworthiness of other users. Furthermore, it is combined with persistent labelling in a user profile on which basis users can derive whether the identity is verified. On these grounds, the feature “identity verification” can be categorised as an empowerment feature. The basic information about the problem “catfish” is already depicted in Figure 10.1 on page 115 and explained in Chapter 10.

In the following, the example is explained to an extent that the single steps of the feature model creation are comprehensibly mapped on the example, focusing on parts of the feature model. The exemplary feature model for catfish protection is displayed in Figure 11.3. It shows the feature model after the allocation phase of the facet attribution process. The validation and configuration are not further explained. For the validation, a quantitative study has to be conducted, which

exceeds the scope of the example. Concerning the configuration, the example is quite small. Therefore, the procedure of the configuration is briefly outlined at the end of this chapter.

Feature Model Creation For the feature model catfish protection (concept feature, root of the feature tree), the feature “identity verification“ is introduced as an empowerment feature ($\llcorner\text{empower}\lrcorner$) on the first layer of the feature tree. In the second layer of the model, identity verification is refined into the three mandatory features, which are the verification algorithm, user profile, and notification about the verification status. Verification algorithms most often try to link a user profile to identifying information. Therefore, a require-link connects the verification algorithm with the user profile. The verification algorithm has three assets that are “photo of ID card”, “phone number”, and “Facebook account”. These assets represent the identifying information that may be used for the verification of the user profile. The OR-link, which marks the three assets as optional, denotes that the algorithm has to consider at least one of the assets.

Since trust-related software features need a representation in the user interface to be user-accessible, the notification about the verification status is included in the feature model. For realising notifications, knowledge about the verification status is required. Therefore, a require-link connects the feature notification with the feature verification algorithm. The notification about the verification status considers three different statuses, which are “verified”, “not completed” and “fake” (see right subtree, layer three of the model). To express the statuses, the principle of familiarity is used for the following assets. The principle of familiarity describes increased usability and understanding of features if users have already encountered the design before [216]. On these grounds, the graphical representations of the statuses are well-known to users by their appearance and meaning. For the verified status, a graphical symbol in form of a green check mark is displayed next to the name of a user profile. Online dating applications such as Tinder ¹ already use this symbol for verified profiles. In case of uncompleted identity verification processes, an orange checkmark is presented next to a profile name. In case of fake identities, these profiles should no longer be available for matching. Interaction with these profiles is denied for all profiles including that that have had a match with the fake profile.

¹www.tinder.com

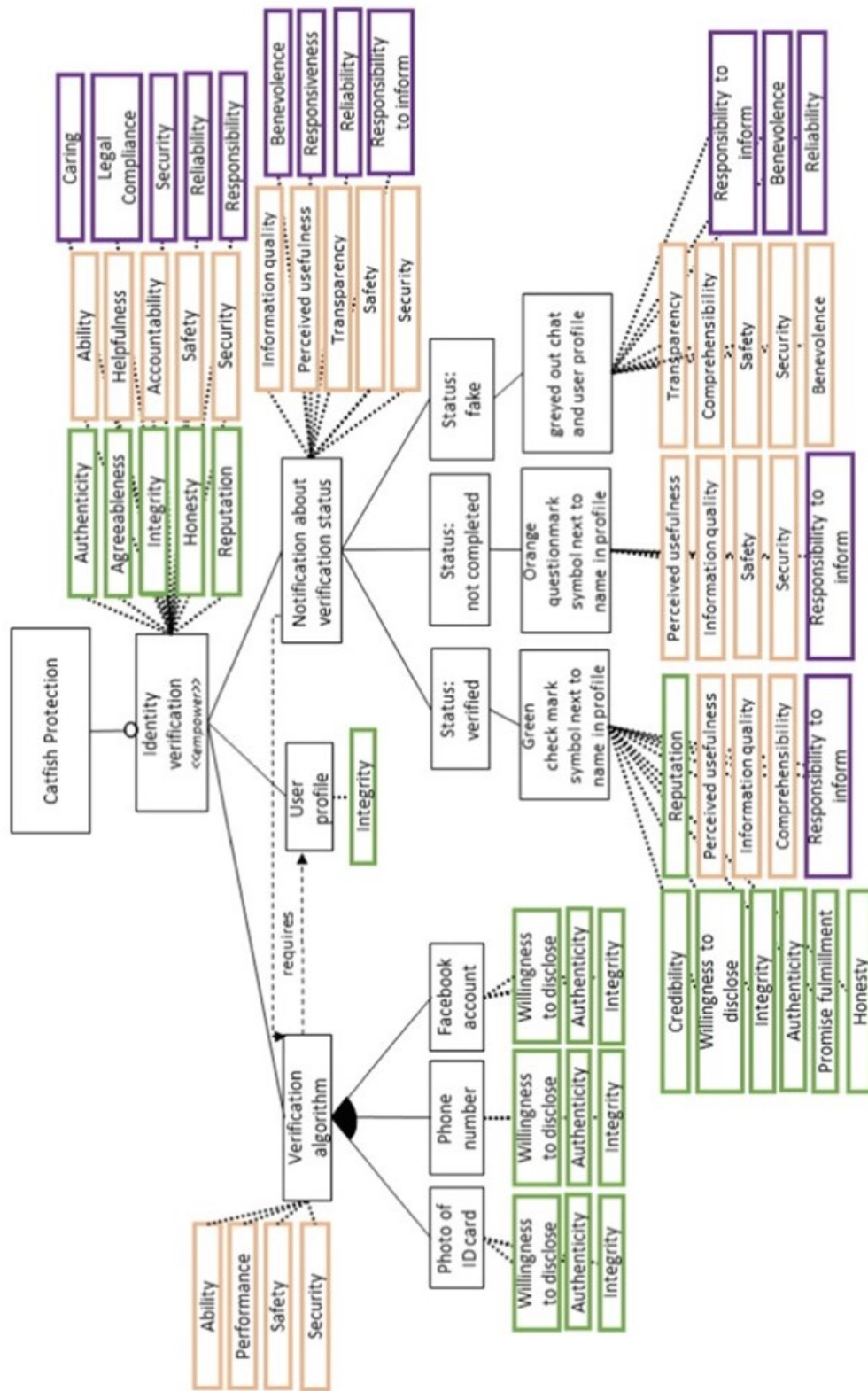


Figure 11.3: Exemplary feature model for catfish protection. The figure is taken over from Paper 6 [37].

To visualise the inactivity of a fake profile, the profile and the corresponding chat are presented greyed out for the other users.

As part of feature modeling and refinement, the asset information of the catalogue structure has to be filled out for each feature. The asset information is demonstrated for the green check mark feature in Figure 11.4.

As part of the identity verification feature, the green check mark is an empowerment feature. It is user-accessible since it is illustrated on the graphical user interface so that users can assess the trustworthiness of other users (target group). Furthermore, the green check mark is a design and information element, because it conveys the message that another user has passed the identity verification. Concerning the nudging criteria, the green check mark provides information that may guide user behaviour. Associated trustworthiness facets with the green check mark are explained in the next paragraph of the facet allocation process.

Asset Information	
Feature type	<input type="checkbox"/> Awareness <input type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for online trustworthiness assessment	<input checked="" type="checkbox"/> Users <input type="checkbox"/> Application <input type="checkbox"/> Service Provider
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open choice architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for individuals	Credibility, willingness to disclose, integrity, authenticity, promise fulfillment, honesty, reputation
Trustworthiness facets for technology	Perceived usefulness, information quality, comprehensibility
Trustworthiness facets for service provider	Responsibility to inform

Figure 11.4: Asset information of the catalogue structure for the feature “green check mark”

Facet Allocation Process After feature model creation for the catfish protection example, the trustworthiness facets are allocated to each feature. Figure 11.3 displays the feature model after the allocation phase. From the previous TrustSoFt procedure for the catfish problem, it is known that the relevant trustworthiness facets are honesty and predictability (see Chapter 3.2). For identity verification, honesty is a relevant trustworthiness facet, because users can derive the honesty from other users about their identity from the feature. Yet, predictability does not seem relevant for identity verification. Users may not derive the future behaviour of other users based on the verification status. Therefore, predictability is omitted for the feature identity verification.

In addition to the already identified trustworthiness facets from TrustSoFt, the guideline for selecting relevant trustworthiness facets is applied. According to the guideline, an understanding of the actual problem has to be acquired to which identity verification serves as a solution. In the end, identity verification shall resolve users' concerns about fake profiles by proving that an identity is true. Catfish are likely to not perform an identity verification to hide the fraud. Therefore, they would not comply with the application's norms in absolving an identity verification. Based on this problematic characteristic, we check the overview of trustworthiness facets on semantically opposite trustworthiness facets by definition (Figure 11.1, Box F). As a result, we assume that users, who perform an identity verification are associated with authenticity, agreeableness, integrity with the norms of the application, honesty, and good reputation. The trustworthiness facets are added as attributes with green bold frames (user facets) to the feature identity verification in Figure 11.3.

After the trustworthiness facets for the users have been selected, the question is how the feature identity verification impacts users' trust in the application (system trust). Following the guideline, former research is checked on that topic. Koch recommends that websites should take the responsibility for their users' safety and security concerning catfish [178]. If the feature was not implemented, users might feel insecure and not well supported. By having identity verification implemented, the online dating application presents its ability to address the problem. Furthermore, the application thereby helps its users in countering their catfish concern. In addition, it shows that it is accountable and takes care of the safety and security of its users. Having this in mind, the overview of trustworthiness facets for technology is checked on relevant facets (Box F). As a result, the trustworthiness facets ability,

helpfulness, accountability, safety, and security are added as attributes in orange bold frames (technology facets) to the feature identity verification.

As a last step for the allocation process of the feature identity verification, the trustworthiness facets of the service provider are concluded. The conclusion follows the same argumentation as in the previous paragraph for the technology facets. It is assumed that service providers express their care for users' safety and security when implementing identity verification in the application. Furthermore, catfishing has been discussed in court concerning online impersonations [178]. Therefore, service providers would demonstrate their responsibility and legal compliance. After checking the overview of trustworthiness facets for organisations for related facets, the trustworthiness facets caring, security, legal compliance, reliability, and responsibility are added to the feature model with a purple frame (organisation facets).

For the rest of the feature model, the allocation phase follows the same procedure. As a side note to Figure 11.3, for the features representing the statuses of the identity verification (layer three), no trustworthiness facets have been added as attributes. They are regarded as specifications of their parent feature and are expressed in detail by their child features. Therefore, the trustworthiness facets of the statuses equal the facets of their child features for this specific case.

After the allocation phase, the propagation phase is performed. Trustworthiness facets that are not yet allocated to parent features are now propagated and receive a line left in their attribute box. This is for example the case for "reputation" from the green check mark feature, which is added as an attribute to the notification feature.

After the feature model for catfish protection and the accompanying catalogue have been established, the trust-related software feature "identity verification" can be configured. As mentioned before, the example is very small so in fact, every asset of Figure 11.3 can be included in the feature. Yet, it is demonstrated how the feature model and the catalogue jointly support practitioners in the configuration.

Concerning catfish protection, the trustworthiness facet honesty has been pointed out by TrustSoFt as highly important for users to be evaluated. Therefore, the feature identity verification must reflect the honesty of users to the user. For that purpose, the catalogue for catfish protection can be looked up for the user trustworthiness facet honesty. As a result, the catalogue proposes the green check mark

symbol next to the user name in the user profile. Therefore, this asset is noted as a “must-have” for the configuration. As a next step, the feature model is used to check the optionality of the assets by the links. Except for the photo of the ID card, the phone number, and the Facebook account, all assets are mandatory. Since the three assets are optional by an OR-link, it is decided to include all three in the verification algorithm to ensure an encompassing check. In the end, all assets in the feature model are included in identity verification.

12

Applying TrustSoFt for Developing and Evaluating a Hybrid Social Media Application

To evaluate the resulting software features from TrustSoFt, TrustSoFt has been applied for developing the hybrid social media application “HushTweet”. The objective is to elicit software features that counter the information privacy concerns associated with HushTweet use. Due to the scope of the development project, the application of TrustSoFt is limited to its core steps and the overview tables leaving out risk assessments, goal modelling, and feature modelling. The resulting trust-related software features are implemented in prototypes for being tested in online user surveys. The results give indications that the software features successfully mitigate information privacy concerns and risk beliefs while increasing trusting beliefs in HushTweet. Findings about the TrustSoFt application show that the choice of what user concern is addressed highly impacts the extent of the features’ impact on users.

This chapter refers to Papers 7 and 8 and addresses Research Question RQ7 “How do software features resulting from software development to support users’ trustworthiness assessment impact users?”. In the following, hybrid social media are briefly introduced. This is followed up by the research model of this study. The research model introduces user variables for gaining an in-depth understanding of HushTweet representing hybrid social media and how the software features resulting from TrustSoFt impact user variables. The research model further introduces research questions for this study. Afterwards, the application of TrustSoFt for HushTweet is summarised, followed by presenting the method for the online user

surveys and the results. At the end of this chapter, the results for HushTweet and hybrid social media are discussed as well as the findings for the TrustSoFt application. All figures are taken over from Papers 7 and 8 [41, 42].

12.1 Hybrid Social Media

Hybrid social media (HSM) combine the benefits of both commercial and privacy-preserving social media [311]. Commercial social media offer users the service of connecting them with other people online without any monetary costs. Their business model is based on generating profit from user data by realising targeted advertisements on their platforms for other companies [143]. Selling user data is one reason why commercial social media are discredited for showing insufficient commitment to their users' privacy. Another reason is that the pioneers of social media Facebook and Twitter have been associated with data leakages in the past, such as Cambridge Analytica in 2018 or the Twitter leak in 2023 [144, 122]. Yet, Facebook and Twitter have very large user bases. They are regarded as an essential part of modern society due to their integration into many other web services [21].

In contrast, privacy-preserving social media have emerged as an alternative to commercial social media and formed new social networks [117]. The objective of privacy-preserving social media is to avoid the intrusion of their users' privacy by protecting user data. For that purpose, privacy-preserving social media are based on distributed technologies by which user data is stored outside the reach of a central provider [117]. In addition, privacy-preserving social media provide high transparency. The implementation is open source and the design is discussed in public. Despite its benefits, privacy-preserving social media are not well-adopted by users. This has several reasons, such as their high usage complexity, poor functionality, and low scalability [311]. In addition, people do not seem to want to give up the benefits of commercial social media to protect their privacy [158].

Therefore, HSM provide users with the benefits of both commercial and privacy-preserving social media [311]. HSM are built as a frame on top of a commercial social media platform. HSM can make use of the large user base of the commercial social media provider to enable user connectivity as well as private communication. Communication between users can be conducted beyond the knowledge of the re-

spective commercial social media provider by storing data in distributed systems. Hence, HSM have the main characteristics of privacy-preserving social media. However, to gain access to the social network of commercial social media, HSM in turn offer consolidated user data to the commercial social media partner. Instead of disclosing individual user data, data is merged into categories of target groups. HSM users still receive somewhat tailored advertising from the commercial social media platform that satisfies the business model of commercial social media. As an example, by using HSM, the information about a user called Jane Doe, who is 32 years old, works as a paramedic, and has bought hiking shoes for 120€ is available neither to HSM nor commercial social media. Instead, Jane Doe can be categorized as a woman, age 29-35, whose profession is in the medical sector, and whose interests are sports.

To this point, HSM is a rather unknown social media option. As with privacy-preserving social media, users might not easily adopt it. Therefore, the objective is to develop an HSM application that addresses users' privacy concerns, stands in contrast to the high complexity of privacy-preserving social media, and convinces users of its trustworthiness. The HSM application shall be called "HushTweet". It enables privacy-preserving Twitter use. To get a first draft about the functionality of HushTweet, TrustSoFt can be regarded as an adequate software development method for the planning phase of HushTweet in the Software Development Life Cycle. By using TrustSoFt, trustworthiness requirements can be specified and software features can be derived that counter people's privacy concerns and consider the application's perceived trustworthiness for the users.

12.2 Research Model

In the context of HSM and TrustSoFt, the research model includes the variables "information privacy concern", "trusting beliefs", "risk belief" and "willingness to use". The research model is depicted in Figure 12.1. It is based on the work of Malhotra et al. [208]. The research model is created for two research objectives: 1) gaining an understanding of the user variables and their effect on each other in the HSM context, and 2) analysing the impact of the resulting TrustSoFt software features on the user variables. First, the research model is introduced in the context of the first research objective in the next paragraph. Afterwards, the research model

is explained in the context of the second research objective.

As a side note, this study was originally intended to test the perceived trustworthiness of HushTweet, because it fits the context of TrustSoFt. Yet, to be in accordance with the work of Malhotra et al. [208], this research focuses on trusting beliefs. As Malhotra et al. introduced scientific questionnaires to the variables of the research model, this study adopted the variables for the ease of survey conduction. In addition, following consistently the work of Malhotra et al. provides proof of the validity of this study. Per definition, the variable trusting beliefs is very similar to perceived trustworthiness, which is confirmed by the results of this research. Scales for both variables are part of the user survey.

12.2.1 Understanding the HSM context

As HSM aims to support users in their information privacy, the starting point of this research model is users' concerns about their information privacy. Information privacy is defined as "the ability of the individual to personally control information about one's self" [292]. Malhotra et al. have identified the most prominent factors contributing to information privacy [208]. The six information privacy concerns are addressed later in the user survey. They are introduced in the following:

Awareness of privacy practices refers to the extent to which an individual knows or is aware of organisational practices for information privacy. The awareness of privacy practices of users refers to the transparency with which an organisation openly communicates its privacy practices. Users have shown concerns that organisations lack appropriate privacy practices [322].

Collection is about the amount and quality of personal data possessed by third parties. When people are concerned about data collection while using online services, they tend to weigh the costs of disclosing personal information against the gained benefit of the service.

Control describes the degree to which users can decide on the processing of personal data. Actions of control are for example providing approval, performing modifications, giving rejections, or opting out. Especially concerning social media, users are concerned about having little control over their personal data [53].

Errors cover organisational problems with processing user data. Errors can be accidental or intentional. Intentional errors mean that data is maliciously falsified. Concerns about errors include the fear that errors are taking place and that organisations are making too little effort to reduce or eliminate them.

Improper access is about parties accessing data without being authorised to do so. Improper access can occur due to technological gaps or organisational policies. Regulations like the General Data Protection Regulation specify that only those individuals and organisations should have access to data who really need to know about it for providing the service.

Unauthorised Secondary Use describes the use of personal data for other purposes than authorised by the respective user.

Users have stated concerns about the above issues of information privacy during social media use [180]. Therefore, this study analyses each information privacy concern for the HSM context. Figure 12.1 depicts a representative research model. “Information privacy concerns” represents the different information privacy concerns introduced above.

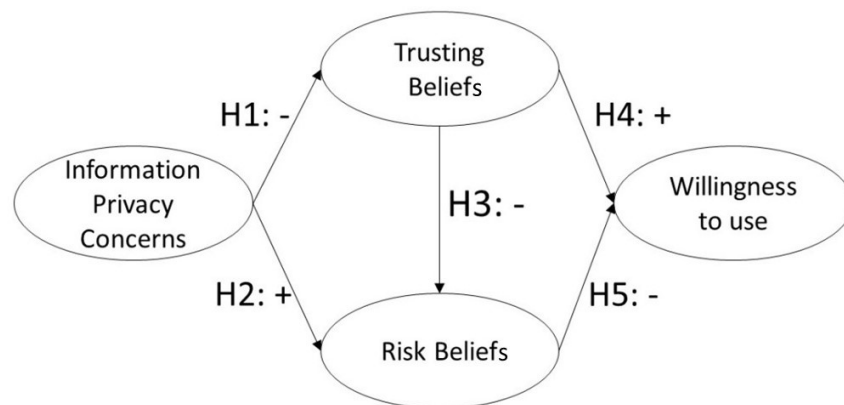


Figure 12.1: Overview of the research model including the hypotheses of this study.

The next variable in the research model is *trusting beliefs*. Trusting beliefs describe an individual’s belief whether an organisation and its software application protect its customers’ personal information and, thus, is trustworthy [208]. For this research, trusting beliefs, therefore, represent people’s trust in the application HushTweet and HSM to protect user data. Previous research has shown that privacy concerns reduce trust in other parties [195, 199]. Therefore, it is assumed that the more users are

concerned about their information privacy, the less they trust HushTweet and HSM in protecting user data. Based on this assumption, Hypothesis H1 is formulated.

Hypothesis H1: The information privacy concerns of HushTweet users have a negative effect on their trusting beliefs.

The third variable in the research model is *risk beliefs*. Risk beliefs represent the individual's expectation of a potential loss due to the disclosure of personal information to an organisation [208]. In this research, risk beliefs are related to associated unwanted incidents that may occur by disclosing personal information to HushTweet. Regarding risk beliefs, the relationships between information privacy concerns and trusting beliefs in HushTweet are tested. Former research found that privacy concerns increase people's risk beliefs while trusting beliefs decrease them [195]. Therefore, Hypotheses H2 and H3 are as follows:

Hypothesis H2: The information privacy concerns of HushTweet users have a positive effect on their risk beliefs.

Hypothesis H3: The trusting beliefs of HushTweet users have a negative effect on their risk beliefs.

The last variable of the research model is the *willingness to use* HSM by the application HushTweet. As HSM and privacy-preserving social media are not yet well adopted by social media users, this research is interested in analysing how trusting beliefs and risk beliefs may impact people's behavioral intention to use HSM. Former research found that trust in an application increases the intention to use it, while risk beliefs reduce the willingness [167, 312]. Therefore, Hypotheses H4 and H5 are as follows:

Hypothesis H4: The trusting beliefs of HushTweet users have a positive effect on their willingness to use HushTweet.

Hypothesis H5: The risk beliefs of HushTweet users have a negative effect on their willingness to use HushTweet.

12.2.2 The Impact of TrustSoFt Software Features

The second research objective is about evaluating the impact of the TrustSoFt software features. For that purpose, a new research model is created, which respects the TrustSoFt software features. Additional hypotheses are tested.

Concerning the research model, it is updated from the variable “information privacy concerns” to “addressed information privacy concerns”. This variable is about the extent to which the TrustSoFt software features have mitigated the information privacy concerns of HushTweet users. The term “addressed information privacy concerns” represents the six information privacy concerns presented in Chapter 12.2.1 being addressed (e.g., “addressed errors concern” or “addressed control concern”) as well as the overall addressed information privacy concerns that result from consolidating the six single addressed concerns. As a consequence, there are six research models for each addressed information privacy concern and one for the overall addressed information privacy concerns. The variable “addressed information privacy concerns” is elicited in the online user survey which is explained in the experimental design introduced in the next Section 12.3. Depending on its specification, it points out to what kind of TrustSoFt software features survey participants have been exposed to. As an example, the variable “addressed control concern” points out to the “Control” concern of the experimental group that has been exposed to TrustSoFt software features aiming to mitigate the control concern. It is assumed that if a concern is addressed by tailored software features the respective concern actually is mitigated. Therefore, Hypothesis H6 states the following:

Hypothesis H6: HSM applications, which have software features implemented for mitigating a particular information privacy concern, have a positive impact on the respectively addressed information privacy concern.

In a conclusion, an HSM application that addresses all information privacy concerns through software features should have the biggest impact on trusting beliefs and risk beliefs compared to an HSM application that addresses only one information privacy issue. This conclusion is addressed by Hypotheses H7a and H7b, which are:

Hypothesis H7a and H7b: HSM applications, which have software features im-

plemented for countering all information privacy concerns (Full-featured Group) lead to a) the highest trusting beliefs and b) the lowest risk beliefs compared to HSM applications addressing only one concern by software features.

In addition to the new hypotheses, the former Hypotheses H1 and H2 can be updated in accordance with the updated variable “addressed information privacy concerns”. The updated hypotheses are Hypotheses H1.1 and H2.1, which reflect the opposite effect than before because the respective variable is not a concern anymore but an addressed concern. Therefore, Hypotheses H1.1 and H2.1 are as follows:

Hypothesis H1.1: The countered information privacy concerns of HushTweet users have a positive effect on their trusting beliefs.

Hypothesis H2.1: The countered information privacy concerns of HushTweet users have a negative effect on their risk beliefs.

12.3 Method

To test the research models and the hypotheses, an extensive online survey was conducted via the crowdsourcing webpage Amazon Mechanical Turk ¹. The structure, used resources, and methodologies are explained below.

Experimental Design For the online survey, a between-group design is chosen with nine experimental groups. The experimental groups differ on the research objective and the HushTweet mockup version they are exposed to before answering the questionnaires. The HSM Concept group addresses the first research objective to understand the HSM context in terms of information privacy concerns, trusting beliefs, risk beliefs, and the willingness to use HSM. HushTweet is only introduced by a textual description representing HSM. The rest of the experimental groups are considered for the second research objective of analysing the impact of TrustSoFt software features on the research model. The Basic App group is the control group. It interacted with a HushTweet version that has no features implemented

¹www.mturk.com

for reducing information privacy concerns. In contrast, the Full-featured group interacted with a Hushtweet version including all elicited TrustSoFt software features, addressing all information privacy concerns. The rest of the experimental groups interacted with a HushTweet version including only those software features for one information privacy concern. An overview of the experimental groups is given in Table 12.4 on page 158 together with the characteristics of the population.

HushTweet Mockups In total, eight mockup versions of HushTweet have been developed with the online design tool Figma ². These are clickable front-end drafts, which offer multiple interaction paths. One mockup version is intended for each experimental group, except the HSM concept group, which does not interact with any mockup. The basic mockup version is intended for the Basic App group and does not include any software features that aim to mitigate information privacy concerns. For the other experimental groups, the basic mockup version is enriched by software features elicited with TrustSoFt (see next section). For the experimental groups whose single information privacy concern is addressed, the basic mockup version is extended by three software features for the respective concern. The Full-featured group receives a HushTweet mockup version that includes all software features from the other experimental groups. A sample of the HushTweet Full-featured mockup version is depicted in Figure 12.2. On the left, the menu bar of HushTweet is presented. The red frames point to hidden TrustSoFt software features that participants can discover. In the middle, the main page of HushTweet is depicted, on which the tweets of other users are shown. Tweets with a white background are public tweets. These are visible to everyone. Tweets with a dark grey background are private tweets. They are only visible to a chosen audience. On the right, in the upper corner, the HushTweet page for posting content is illustrated. Users receive additional information highlighted by the red frame. In the lower corner, tweet and interaction options are presented.

The included software features have been carefully selected for the different mockups from the set of specified TrustSoFt software features. They are similar in nature, addressing the same or similar trustworthiness facets while covering as many facets as possible. Thereby, the results of the experimental groups are somewhat comparable. As an example, for each mockup version, a FAQ feature is included. The

²www.figma.com

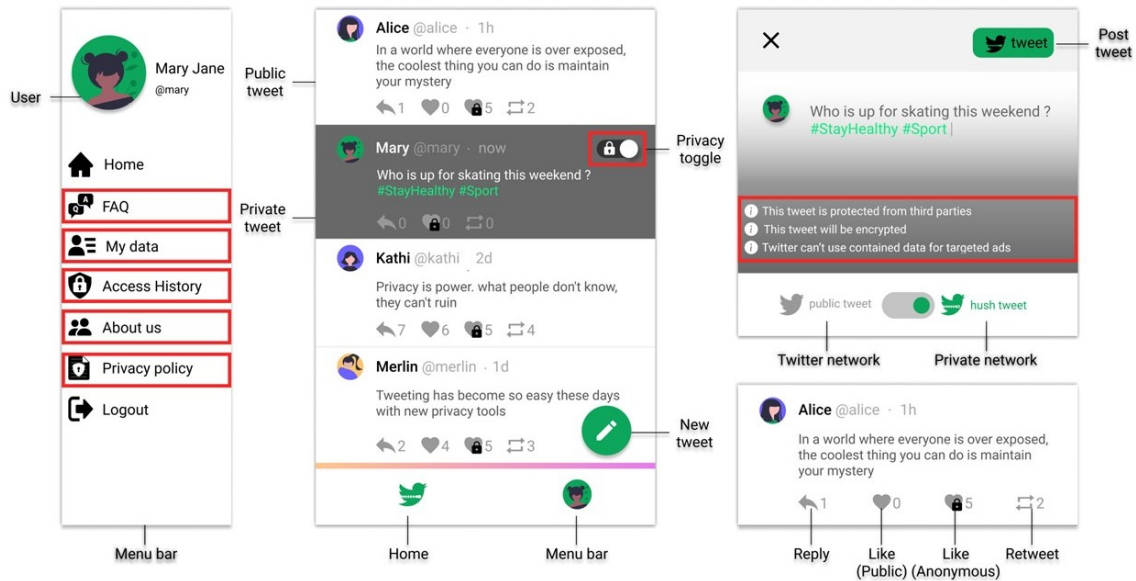


Figure 12.2: Sample of the Full-features Hushtweet mockup.

FAQ feature is an answered question on how the respective concern is treated in HushTweet. An overview of all software features included in the HushTweet mockups is depicted in the Tables 12.1, 12.2, and 12.3 on pages 155, 156 and 157. Within the mockups, the included software features are highlighted by red frames. The red frames counter the risk that the software features are considered standard. Instead, they ensure that study participants consciously perceive the software features for countering the respective information privacy concern.

Scales For testing the research models, scientific scales have been selected. The scales are mainly adopted from the work of Malhotra et al. [208]. These are the Internet Users' Information Privacy Concern scale (IUIPC) [208], the Concern for Information Privacy Scale (CFIP) [280], the General Information Privacy Concern scale (GIPC) [280], and the scales for trusting beliefs and risk beliefs [149].

Malhotra et al. have extended the CFIP by Smith et al. [280] to the context of the Internet, resulting in the IUICP [208]. Thus, the CFIP is part of the IUICP. By including the IUICP in the set of used scales for this study, all information privacy concerns are surveyed. The GIPC is included as a generic control scale to check whether its results are similar to the IUICP.

In accordance with the research of this dissertation, the scale for the perceived trustworthiness of online shops [50] is also added to the set of used scales. It serves as

a control scale for the trusting beliefs scale of Jarvenpaa et al. [149]. Furthermore, it provides insights into the trustworthiness facets ability, benevolence, integrity, and predictability. The trustworthiness facets are the sub-scales of the scale for the perceived trustworthiness of online shops. Instead of measuring the perceived trustworthiness of online shops, the wording of the scale was adapted to HushTweet measuring its perceived trustworthiness.

To measure the willingness to use HushTweet, eight questionnaire items have been self-developed. The validity of each item has been scientifically tested. Only those items with acceptable validity have been included in the analysis of the willingness to use HushTweet.

For all scales, a 7-point Likert scale has been used with 1=“strongly disagree” to 7=“strongly agree”. All experimental groups have answered the questionnaires. Yet, while the HSM Concept group was asked about their information privacy concerns by the IUICP, the other experimental groups were asked about their addressed information privacy concerns. Therefore, the wording of the IUICP has been modified. Instead of asking the participants for the extent of their information privacy concerns, they were asked to what degree HushTweet mitigates their information privacy concerns. Rewording took place for example when “online companies” and “computer databases” were replaced by “HushTweet” and “distributed databases”. All used scales are depicted in the Appendix.

In addition to the variables of the research model, demographic and issue-related variables were asked and tested. The variables again comply with the research of Malhotra et al. [208]. The demographic variables are *gender*, *age*, and *education*. The issue-related variables are *internet use* in hours per day, the amount of *media exposure* concerning how often individuals were exposed to news reports of privacy violations, the frequency of *experienced privacy violations*, and the occurrence of *misrepresentation of identification*. The latter refers to the occurrence of how often people have provided false identification information when they were asked for them by organisations. For each variable, one item was used based on the work of Malhotra et al. [208]. The items can also be found in the Appendix.

Procedure For all experimental groups, the procedure is nearly the same. First, the participants received a briefing about the context of the study to then

answer the GIPC for their general privacy concerns. Afterwards, the groups were introduced to the HSM concept, HushTweet, and its basic functionalities through a short descriptive text. The imparted knowledge was tested by means of six questions. Only those participants were included for survey analysis, who understood the concept of HushTweet by answering half the questions correctly. At that point of the study procedure, all experimental groups except the HSM concept group interacted with their respective HushTweet mockup version for at least five minutes. During the interaction, the experimental groups had to solve tasks modified to the specific HushTweet mockup version. The tasks were related to the implemented software features countering the respective information privacy concern. As a next step, all experimental groups received the remaining scales in the following order: perceived trustworthiness scale, UIIPC, CFIP, trusting beliefs scale, risk beliefs scale, and the questions concerning the willingness to use HushTweet. In the end, demographic and issue-related questions were posed. Participants were also asked for further concerns about HSM and HushTweet.

12.4 Applying TrustSoFt for HushTweet

TrustSoFt is applied for HushTweet to elicit frontend software features that counter the six information privacy concerns introduced in Section 12.2. The TrustSoFt application is briefly explained step by step in the following. Moreover, it is exemplarily illustrated by the concern “Errors”. Yet, an overview of all steps for each information privacy concern is presented in Tables 12.1, 12.2 and 12.3 on pages 155, 156 and 157.

Trust Concerns. The starting point of TrustSoFt is the information privacy concerns that have been identified by Smith et al. and Malhotra et al. [280, 208] (see Section 12.2). For each concern, TrustSoFt is applied separately. As a first step, an understanding of information privacy concerns must be gained. For that reason, the definitions of the concerns are revisited to make oneself aware of identifiable characteristics and descriptive keywords that characterise the concern. For the “Errors” concern, descriptive keywords are for example “errors in personal data”, “deliberate and accidental errors”, and “error minimisation”.

Trustworthiness Goals. Understanding the respective concern supports the specification of trustworthiness goals. The specified trustworthiness goals shall mitigate the respective concern and increase overall user satisfaction. Concerning the “Errors” concern, the characteristic keywords point out relevant trustworthiness goals. To overcome errors in HushTweet, the objective is to store personal data accurately and error-free. Therefore, an exemplary trustworthiness goal for the “Errors” concern is “data accuracy”.

Trustworthiness Facets. Based on the trust concern and the trustworthiness goal, the trustworthiness facets are derived. Depending on the concern, various stakeholders can be involved. The trustworthiness facets must be derived with regard to the associated stakeholders. For that purpose, the overview of trustworthiness facets in Appendices A, B, and C on pages 307, 310, and 313 serve as tools for selecting trustworthiness facets that can be related to the given concern and goal.

For the “Errors” concern and the trustworthiness goal “data accuracy”, HushTweet in form of the software application is identified as a relevant stakeholder. In this context, the trustworthiness facets “data integrity” and “fault tolerance” are selected as desirable for HushTweet. Data integrity describes the consistency and accuracy of data throughout its origin, transfer or reuse [187]. Fault tolerance means that despite deliberate and accidental errors the service is still delivered correctly.

Trustworthiness Requirements. As a next step, the specification of trustworthiness requirements defines what HushTweet should do to achieve the trustworthiness goals while simultaneously meeting the trustworthiness facets.

To meet “data accuracy” and “data integrity” in the context of the “Errors” concern, an exemplary trustworthiness requirement is that HushTweet verifies the correctness of user data.

Software Features. Finally, the software features are elicited and specify how the trustworthiness requirements are realised in the front end of HushTweet.

Ideas for software features that implement the requirement of verifying the correctness of user data are for example (1) an alert message when tweeting privately

that “Data is correctly and safely stored” and (2) answered questions in the FAQ section. Answered questions can be for example “How does HushTweet ensure the correctness and integrity of my data?” and “Does HushTweet modify my data?”.

12.5 Results

In the following, the results of this study are reported. First, the details on the population of the participants are presented. Then follows the descriptive results of people’s information privacy concerns, trusting beliefs in and the perceived trustworthiness of HSM, risk beliefs, and the willingness to use HSM like HushTweet. Afterwards, the relationships of the various variables are tested regarding the research models and Hypotheses H1-H5 – including Hypotheses H1.1 and H2.1. This again is followed by the results of how the TrustSoFt software features address and impact the information privacy concerns, trusting beliefs, and risk beliefs addressed by Hypotheses H6-H7.b. The reported results are partly copied from the Papers 6 and 7 [41, 42].

12.5.1 Population

In total, 2300 participants took part in the online user survey via Amazon Mechanical Turk. 300 participants constitute the HSM Concept group, while the other experimental groups consist of 250 participants each. A qualification requirement for participation was an experience of more than 1000 completed and approved surveys on Amazon Mechanical Turk. This qualification requirement should ensure that only participants took part, who are known to fill out surveys properly and without haste. In addition, the population was filtered for completed data sets and participants who had three or more answers correct on the HushTweet comprehension test (see Section 12.3). Due to these exclusion criteria, between 7% and 19% of the population for each experimental group were not permissible for the analysis. Table 12.4 presents the descriptive statistics for the final population of each experimental group concerning gender, age, and education level. With an average rate of 62,3% male and 32,8% female participants, the experimental populations resemble

Privacy Concern	Trustworthiness Goal	Trustworthiness Facet	Trustworthiness Requirement	Trust-related Software Feature
Awareness of Privacy Practices	Clarity of privacy practices	Transparency	Providing an overview of privacy practices	FAQ: “How does HushTweet protect my privacy?”
	Clarity of privacy practices	Transparency	Informing users about the privacy practices of private tweets	Alert message on tweeting (1) privately: “This tweet will be encrypted” and (2) publicly: “Twitter has access to this data”
Collection	Clarity of privacy practices	Transparency, Provider Integrity	Informing users about the legally binding commitments of HushTweet regarding privacy	“Privacy Policy” page that informs users on data collection and its purpose
	Fairness	Completeness, Transparency, Provider Integrity	Showing users the statistical information that they are part of. Clarifying their benefits from the data collection	“My data” page that contains: (1) a description of the statistical data and its purpose, (2) a list of statistical information that the user is part of
	Awareness	Transparency, Provider Integrity, Provider Predictability	Informing users about their data usage by Twitter and HushTweet and the services they receive in return	FAQ: “How do HushTweet and Twitter use my data?”, “What is my benefit from HushTweet’s service compared to Twitter’s service?”
	Awareness	Transparency, Provider Integrity	Informing users on their data usage by Twitter and HushTweet	Alert messages on tweeting (1) privately: “Twitter can’t use contained data for targeted ads.”, and (2) publicly: “Twitter might use contained data for targeted ads”

Table 12.1: First part of the overview of trust-related software features for HushTweet from the TrustSoFt method.

Privacy Concern	Trustworth. Goal	Trustworthiness Facet	Trustworthiness Requirement	Trust-related Software Feature
Control	Data control	Privacy (control)	Allowing users to decide how their data is shared	A toggle button that allows to change the user's posted tweets between private and public any time
	Data control	Privacy (control)	Allowing users to delete all their data	FAQ: "How can I delete my data?", "My data" page that contains a button for deleting all user data
	Procedure control	Privacy (control)	Allowing users to decide what data is used for the statistical information	"My data" page that contains: (1) a list of statistical information that the user is part of, and (2) a toggle button for each item of this information with which the user can opt-in/opt-out of the collection
Error	Data accuracy	Data integrity	Verifying the correctness of data	Alert message on tweeting privately: "Data is correctly and safely stored"
	Data accuracy	Data integrity	Verifying the correctness of data	FAQ: "How does HushTweet ensure the correctness and integrity of my data?" Tweet is stored locally when disconnected from network. Alert message: "Don't worry, your tweet is stored locally. You can post when you are reconnected."
Data accuracy	Fault tolerance	Maintaining data accuracy on network disconnection		

Table 12.2: Second part of the overview of trust-related software features for HushTweet from the TrustSoFt method.

Privacy Concern	Trustworth. Goal	Trustworthiness Facet	Trustworthiness Requirement	Trust-related Software Feature
Improper Access	Technical Access Control	Confidentiality	Protecting user data from unauthorised users or parties	Alert message on tweeting (1) privately: “This tweet is secured against unauthorised parties”, and (2) publicly: “This tweet will be protected by Twitter and partners.”
	Technical Access Control	Transparency, Traceability	Showing users who had access to their data	“Access History” page displays (1) time of login, (2) data access for calculating statistical information by HushTweet, and (3) profile view by other users
Unauth. Secondary Use	Organisational Access Control	Provider Integrity	Clarifying the HushTweet policy in regard to the restricted access of developers to user data	FAQ: “How does HushTweet protect my data from improper access?”
	Authorisation	Transparency	Informing users about the data usage by HushTweet. Requesting data access and usage	“Authorisation” page containing (1) a description of data access by HushTweet and (2) a toggle button to authorise HushTweet to use data for calculating statistical information
	Clarity of intent	Provider integrity, Provider benevolence	Informing users about HushTweet being a research project	“About us” page that informs about HushTweet and the research project
	Clarity of data use purpose	Transparency, Provider integrity	Informing users about how HushTweet uses the statistical information and who has access to it	FAQ: “For what purpose is my data used?”, “Is my data shared with third parties?”

Table 12.3: Third part of the overview of trust-related software features for HushTweet from the TrustSoFt method.

Group	Population (n)	Men %	Women %	Age (M)	Bachelor's degree or higher (%)
HSM Concept	245	62.4	37.6	34.1	71.8
Basic App	205	68.3	31.2	33.6	84.5
Awareness	222	63.1	36.0	33.5	67.1
Collection	223	63.2	36.3	35.6	66.9
Control	223	58.3	39.5	37.6	70.8
Errors	211	58.7	39.8	32.6	87.7
Improper Access	202	58.4	41.6	35.9	72.8
Unauthorised S. Use	216	64.8	35.2	33.8	83.8
Full-featured	233	63.9	35.2	35.6	68.3

Table 12.4: Experimental groups and descriptive results of the populations.

the gender imbalance of Twitter users worldwide in January 2021 with 68,5% men and 31,5% women [241].

12.5.2 Information Privacy Concerns regarding HSM

To understand people's information privacy concerns and the other variables in the context of HSM, a descriptive analysis of the scales from the "HSM Concept" group was conducted. The GIPC has a mean of $M=4.89$ and a standard derivation of $SD=.93$. In comparison, the mean of the IUIPC is $M=5.73$, $SD=.74$. Both scales strongly correlate ($r=.561$, $p<.001$). Having a look at the individual information privacy concerns, the participants rated that HushTweet should consider each concern in the following order (from high to low): (1) Unauthorised secondary use ($M=6.26$, $SD=.93$), (2) awareness of privacy practices ($M=6.16$, $SD=.84$), (3) improper access ($M=5.89$, $SD=1.03$), (4) control ($M=5.87$, $SD=.86$), (5) errors ($M=5.14$, $SD=1.30$), and (6) collection ($M=5.04$, $SD=1.17$). Regarding the other variables, trusting beliefs have a mean of $M=5.14$, $SD=1.08$. The perceived trustworthiness of HSM like HushTweet was rated with $M=5.24$, $SD=.97$. Concerning the trustworthiness facets, integrity was rated the highest ($M=5.42$, $SD=1.12$), followed by benevolence ($M=5.40$, $SD=1.11$), ability ($M=5.33$, $SD=1.02$), and predictability ($M=5.07$, $SD=1.07$). Risk beliefs were rated with $M=3.58$, $SD=.94$. The participants rated the willingness to use HSM like HushTweet with a mean of $M=5.36$, $SD=1.02$.

The descriptive results show that the participants were moderately concerned about their information privacy. They agreed that HushTweet should address infor-

mation privacy concerns. Yet, the participants tended to trust HSM like HushTweet and tended to disagree that it is risky. It is worth mentioning that the relatively high values of the standard derivations of all the variables show the diversity of the participants' opinions on the topic.

Scale	M	SD
GIPC	4.89	.93
IUIPC	5.73	.74
–Unauthorised Secondary Use	6.26	.93
–Awareness of Privacy Practices	6.16	.84
–Improper Access	5.89	1.03
–Control	5.87	.86
–Errors	5.14	1.30
–Collection	5.04	1.17
Trusting Beliefs	5.14	1.08
Perceived Trustworthiness	5.24	.97
–Integrity	5.42	1.12
–Benevolence	5.40	1.11
–Ability	5.33	1.02
–Predictability	5.07	1.07
Risk Beliefs	3.58	.94
Willingness to use HSM	5.36	1.02

Table 12.5: Descriptive results of the scales.

12.5.3 Results of the Research Models – Hypotheses H1-H5

To analyse Hypotheses H1-H5 – including H1.1 and H2.1 – of the research models, the statistical method of Structural Equation Modelling (SEM) was applied for each experimental group. By using SEM, research models can be tested whether the theoretically-based modelled causality of the variables is valid for the given data set. For the SEM calculation, the guideline of Anderson and Gerbing [9] was used.

For each SEM, only those questionnaire items were considered for analysis with an internal scale consistency higher than $\alpha = .70$. Items below that value do not measure the scale construct in a valid way. Another analysis criterion involves the constructs in the research model. Constructs are the sub-scales contributing to the overall scale, i.e. the single information privacy concerns of the IUICP and the trustworthiness facets of the perceived trustworthiness scale. Only constructs with factor loadings higher than .700 were considered in the analysis. Omitted constructs

do not contribute much to the overall scale. In terms of information privacy concerns, “Collection” is excluded from every SEM of the experimental groups. The privacy concern “Errors” was only relevant for the experimental groups “Control”, “Errors” and “Improper Access”. Moreover, the model fit of the SEMs was checked by confirmatory factor analysis [141]. The model fit of all calculated SEMs is at least acceptable with a comparative fit index (CFI) and Tucker-Lewis index (TLI) higher than .90, a root-mean-square error of approximation (RMSE) lower than .80 and a normed chi-square (χ^2/df) lower than 5.

Figure 12.3 presents the SEM of the HSM Concept group. Its model fit is good ($(\chi^2/df)=1.943$, $TLI=.949$, $CFI=.956$, $RMSEA=.062$). Regarding the hypotheses, the SEM cannot confirm Hypothesis H1. The relation between information privacy concerns and trusting beliefs is not significant. Thus, it cannot be said that information privacy concerns reduce trusting beliefs in HSM. Concerning Hypothesis H2, a small positive effect of information privacy concerns on risk beliefs can be observed. With increasing information privacy concerns, risk beliefs in HSM slightly increase as well. Hypothesis H3 is also supported. Trusting beliefs in HSM highly reduce risk beliefs. Last but not least, the willingness to use HSM increases with increasing trusting beliefs (H4) and slightly decreases with increasing risk beliefs (H5).

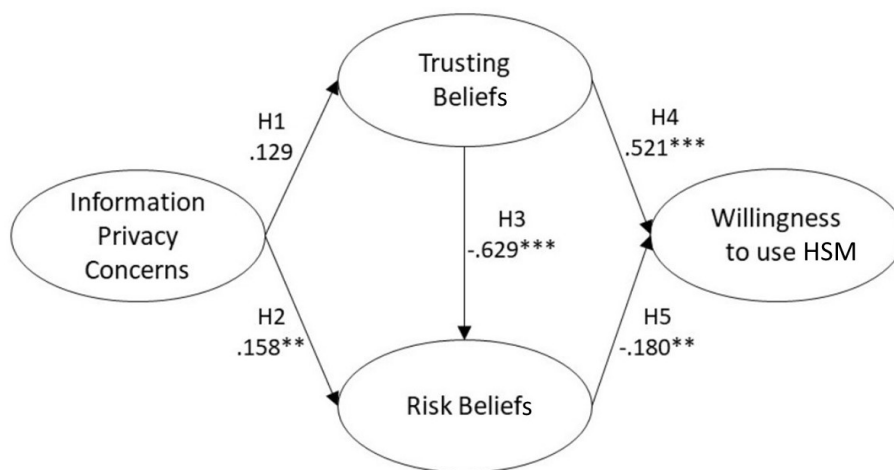


Figure 12.3: SEM of the HSM Concept group. ** $p < .01$, *** $p < .001$

For the other experimental groups whose information privacy concerns were addressed as they used the HushTweet mockups, Hypothesis H1.1 can be confirmed. Countered information privacy concerns highly affect trusting beliefs in a positive way throughout all experimental groups. Trusting beliefs, in turn, strongly influence people’s willingness to use HushTweet, confirming Hypothesis H4. Hypothesis

H2.1 about the countered information privacy concerns reducing risk beliefs cannot be confirmed. The relationship is not statistically significant. Hypothesis H5 – the negative relationship of risk beliefs on the willingness to use HushTweet – cannot be supported as well. While for some experimental groups, the relationship is not statistically significant, for the other experimental groups, risk beliefs positively impact the willingness to use HushTweet with a weak effect. This is the case for the experimental groups “Basic” ($r=.208$, $p=.001$), “Control” ($r=.178$, $p=.007$) and “Unauthorised Secondary Use” ($r=.110$, $p=.044$). Therefore, Hypothesis H5 can partly be falsified. Last but not least, Hypothesis H3 assumes a negative impact of trusting beliefs on risk beliefs. Hypothesis H3 can only be confirmed for the Full-featured group ($r=-.398$, $p=.001$).

12.5.4 The Impact of TrustSoFt software features – Hypotheses H6, H7a, and H7b

Analysing how the TrustSoFt software features impact HushTweet users, Hypotheses H6, H7a, and H7b are tested for all experimental groups that used a HushTweet mockup. This leaves the HSM Concept group unconsidered. The hypotheses are tested by two-factor analyses of variance (ANOVAs) [9]. The objective is to examine differences in the addressed information privacy concerns between the experimental groups. It is expected that the single information privacy concerns are rated the highest by the experimental group that was exposed to the corresponding HushTweet mockup with the implemented TrustSoFt software features addressing the respective concern (Hypothesis H6).

For the analysis, only those information privacy concerns are considered whose internal consistency had a Cronbach’s alpha higher than $\alpha > .70$. The concern “Collection” is excluded from analysis for any experimental group. Furthermore, the concern “Control” has an insufficient internal consistency in the experimental groups “Control”, “Collection”, and “Improper Access”.

Hypothesis H6 can only be supported for the concern and experimental group “Errors”. The “Errors” group rated the “Errors” concern the highest with $F(7,1727)=4.249$, $p=.000$, partial $\eta^2=.017$. Yet, only 1.3% of the variation of the countered “Errors” concern around the total mean value can be explained by the implemented “Errors”

software features (adjusted R-square). The effect size of the model is $f=.13$, which can be interpreted as weak. Posthoc tests with the Bonferroni correction show significant differences ($p < .05$) between the “Errors” group ($M=5.34$, $SD=.98$) and the groups “Awareness” ($M=4.95$, $SD=1.11$), “Collection” ($M=4.97$, $SD=1.05$), “Control” ($M=4.82$, $SD=1.06$), and “Unauthorised Secondary Use” ($M=4.95$, $SD=1.21$), indicating real significant results.

Another noticeable result is provided by the ANOVA for the “Control” concern that is significantly different from the experimental groups ($F(7,1727)=2.063$, $p=.044$, partial $\eta^2=.008$). Yet, contrary to what is assumed by Hypothesis H6, it is the “Awareness” group that has rated the “Control” concern to be countered the best of all information privacy concerns ($M=5.86$, $SD=.91$). The “Control” group is on the second rank of rating the “Control” concern the highest ($M=5.83$, $SD=1.01$).

Regarding Hypotheses H7a and H7b, two-factor ANOVAs are calculated for the Full-featured group. For reasons of interest, the ANOVAs are also calculated for the other experimental groups. Concerning Hypothesis H7.a, the ANOVAs for trusting beliefs and the perceived trustworthiness of HushTweet are not statistically significant for any of the experimental groups. Hypothesis H7a cannot be confirmed. The only significant ANOVA model in the context of trust is for the trustworthiness facet integrity ($F(7,1727)=2.017$, $p=.05$, partial $\eta^2=.008$). The “Awareness” group rated the trustworthiness facet integrity the highest ($M=5.89$, $SD=.93$), while the “Errors” group rated it the lowest ($M=5.60$, $SD=.97$).

Concerning Hypothesis H7b, the ANOVA shows that risk beliefs about HushTweet are rated the lowest by the “Awareness” group ($M=3.32$, $SD=.11$) instead of the Full-featured group as expected ($F(7,1727)=10.364$, $p=.000$, partial $\eta^2=.040$). On these grounds, Hypothesis H7b is rejected. As a side result, the highest rated risk beliefs are from the “Errors” group ($M=4.35$, $SD=.11$). The Basic App group was at the second highest position ($M=4.11$, $SD=.11$).

12.5.5 Demographic and Issue-related Differences in the Variables

Following along the research of Malhotra et al. [208], the demographic and issue-related variables of the population are also analysed concerning differences in the

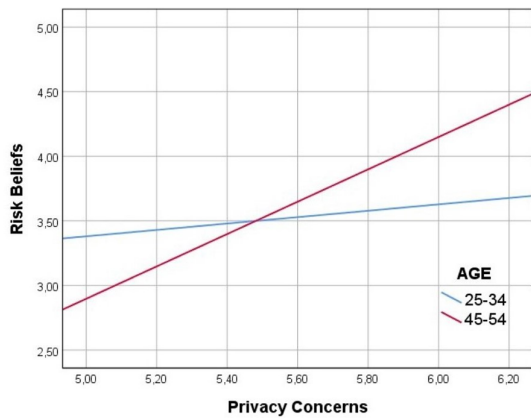
constructs (addressed) information privacy concerns, trusting beliefs, risk beliefs, and the willingness to use HushTweet. By analysing user differences, insights can be gained for the development of HSM applications.

Therefore, an exploratory moderation analysis was calculated for the two boundary groups HSM Concept and Full-featured. The analysis was employed for the demographic variables gender, age, and education. Concerning the issue-related variables, privacy invasion and identification misrepresentation are considered. Internet use and media exposure are excluded from the analysis because the results demonstrate that the scale from Malhotra et al. [208] has no relevance to current behaviours.

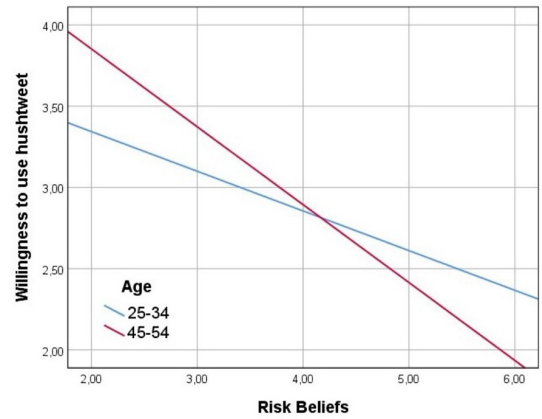
To perform the moderation analysis, the PROCESS macro in the SPSS software is used with standardised variables [128]. Dummy coding was used for the categorical variables [157] with the following reference variables: “Never” for “ID misrepresentation”, “high school” for “education”, and “25–34” for “age” in the HSM Concept group. Age is a metric variable for the Full-featured Group and, thus, does not need a reference variable. From the ordinal variables, some categories cannot be considered representative, because of the very small number of associated participants. Therefore, the following variable categories are excluded: “diverse” from “gender” and “some school, no degree” from “education” for both experimental groups; “doctoral degree” from “education” for the Full-featured group; “18-24” from “age” from the HSM Concept group. Boxplots are used for omitting extreme outliers [91]. After moderation analysis, simple slope analyses were conducted to examine and visualise interaction effects [4]. The simple slope analyses are depicted in Figures 12.4 - 12.8 and are reported per variable for the two experimental groups. For the ordinal variables, the interaction graphics show the significant categories of the variables. For the metric variables, the percentiles low, medium, and high are presented.

Moderations for the HSM Concept Group. For the HSM Concept Group, a total of three moderation effects are detected. They involve the variables age and ID misrepresentation. The interaction effects are depicted in Figure 12.4. No moderation effects are found for gender, education, and privacy invasion.

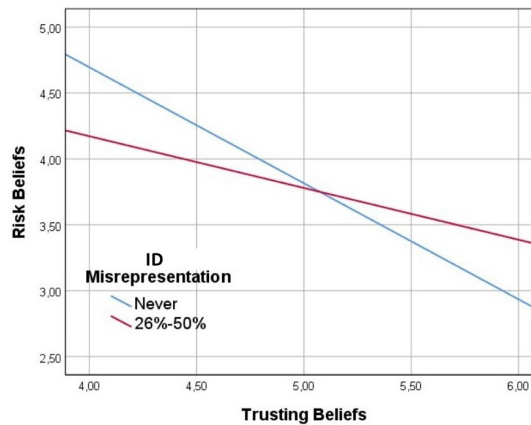
Age is a moderating variable for the effects between (a) privacy concerns and risk beliefs ($R^2 = .128$, $F(9, 234) = 3.828$, $p = .002$) and (b) risk beliefs and willingness



(a) Effect of age on privacy concerns and risk beliefs.



(b) Effect of age on risk beliefs and the willingness to use HushTweet



(c) Effect of ID misrepresentation on trusting beliefs and risk beliefs.

Figure 12.4: HSM concept group - simple slopes for the moderations of age and ID misrepresentation.

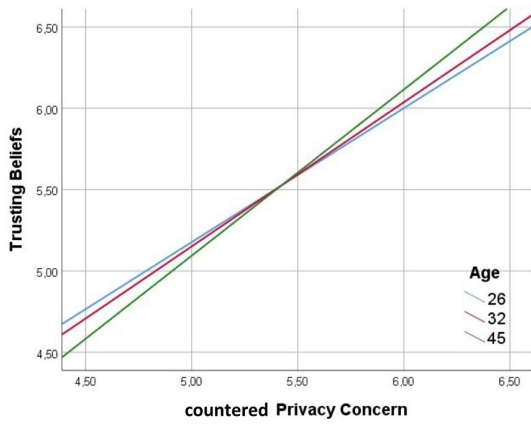
to use HSM like HushTweet ($R^2 = .254$, $F(9, 234) = 8.858$, $p < .001$). An interaction effect can be found for users with the age “45-54” for the prediction of information privacy concerns on risk beliefs ($\beta = .523$, $t(244) = 2.28$, $p = .023$, Figure 12.4a). For risk beliefs and willingness to use, there is also an interaction effect with the age group “45-54” ($\beta = .353$, $t(244) = 2.04$, $p = .043$, see Figure 12.4b). Another moderation is found for ID misrepresentation on the prediction of trusting beliefs on risk beliefs ($F(9, 234) = 14.625$, $p < .001$, predicting 36% of the variance) for the category “26%-50%” ($\beta = .367$, $t(244) = 2.11$, $p = .036$; see Figure 12.4c).

Moderations for the Full-featured Group. For the Full-featured group, a total of 11 moderation effects are found. They involve the variables age, education, ID misrepresentation, and privacy invasion. The interaction effects are depicted in Figures 12.5, 12.6, 12.7, and 12.8. No moderation effects are found for gender.

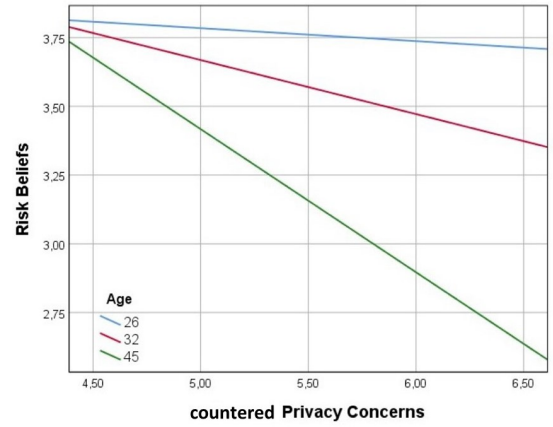
Concerning the user variable age, three moderating effects are found on the predictions of (a) countered information privacy concerns on trusting beliefs ($F(3, 228) = 133.541, p < .001$, predicting 63.73% of the variance), (b) countered information privacy concerns on risk beliefs $F(3, 226) = 6.176, p < .001$, predicting 7.58% of the variance), and (c) trusting beliefs on the willingness to use HushTweet ($F(3, 228) = 53.415, p < .001$, predicting 41.27% of the variance). The interaction effects with each dependent variable (first variable) for predicting the independent one (latter variable) are as follows (a) $\beta = .09, t(232) = 2.40, p = .017$ (Figure 12.5a), (b) $\beta = .13, t(230) = 2.26, p = .025$; (Figure 12.5b), and (c) $\beta = .13, t(232) = 2.63, p = .009$ (Figure 12.5c).

For education, a moderating effect is identified on countered privacy concerns on risk beliefs ($F(9, 218) = 3.640, p < .001$, predicting 13.06% of the variance). The interaction effect involves people having a Master's degree compared to those with a high school graduation ($\beta = .144, t(228) = 2.02, p = .044$, Figure 12.6).

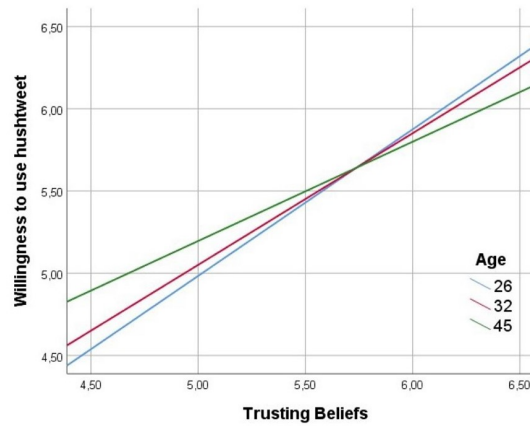
The four interaction effects of the user variable identification misrepresentation are depicted in Figure 12.7. ID misrepresentation moderated the relationships of addressed privacy concerns and trusting beliefs ($F(9, 219) = 50.684, p < .001$, predicting 67.56% of the variance). People who misrepresented identifiable information in over 75% of all cases interacted with addressed privacy concerns when predicting trusting beliefs ($\beta = 1.37, t(229) = 3.09, p = .002$; Figure 12.7a). For addressed privacy concerns and risk beliefs ($F(9, 223) = 4.338, p < .001$, predicting 15.05% of the variance), interaction effects with addressed privacy concerns are found for the categories "26%–50%" ($\beta = .396, t(233) = 2.62, p = .010$) and "51%–75%" ($\beta = .778, t(233) = 3.08, p = .002$; Figure 12.7b). For the moderation with trusting beliefs and risk beliefs ($F(9, 219) = p < .001$, predicting 19.80% of the variance), the interaction effect with trusting beliefs was significant for the categories "26%–50%" ($\beta = .386, t(229) = 2.42, p = .016$) and "51%–75%" ($\beta = .534, t(229) = 2.09, p = .038$; Figure 12.7c). The last moderation of "ID misrepresentation" is found for



(a) Effect of age on countered information privacy concerns and trusting beliefs.



(b) Effect of age on countered information privacy concerns and risk beliefs.



(c) Effect of age on trusting beliefs on the willingness to use HushTweet.

Figure 12.5: Full-featured group - simple slopes for the moderation effects of age.

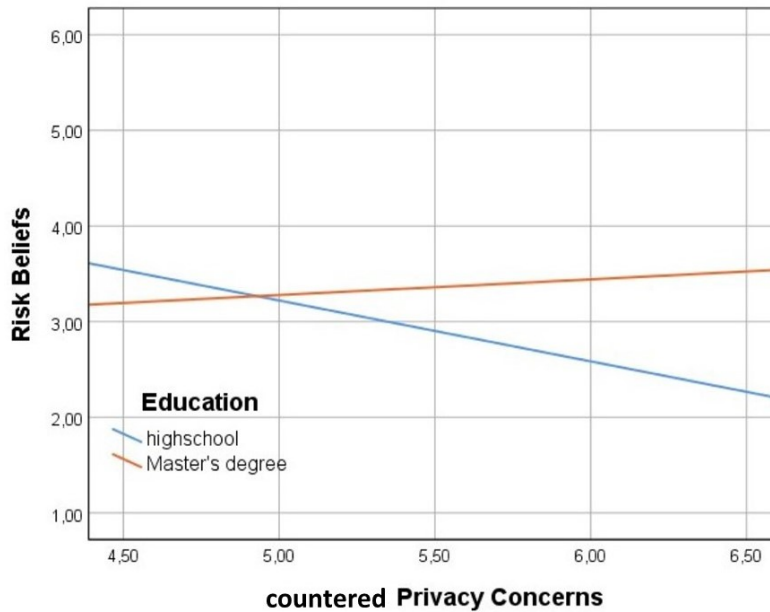
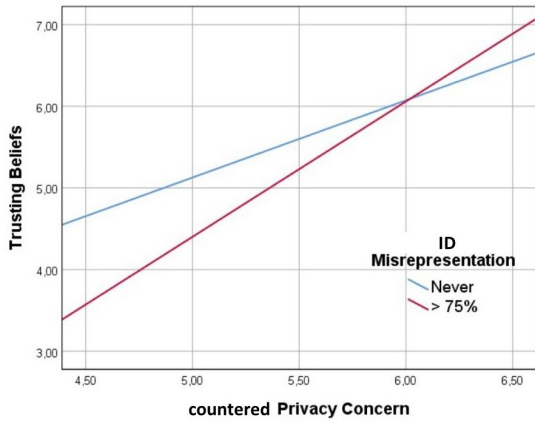


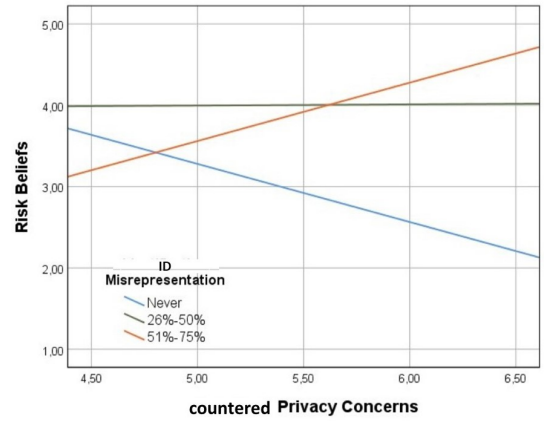
Figure 12.6: Full-featured group - Effect of education on countered information privacy concerns on risk beliefs.

trusting beliefs and the willingness to use ($F(9, 219) = 20.968, p < .001$, predicting 46.29% of the variance). The categories “26%–50%” ($\beta = .302, t(229) = 2.32, p = .021$) and “over 75%” ($\beta = .501, t(229) = 2.04, p = .042$) significantly interacted with trusting beliefs for the prediction of the willingness to use HushTweet (Figure 12.7d).

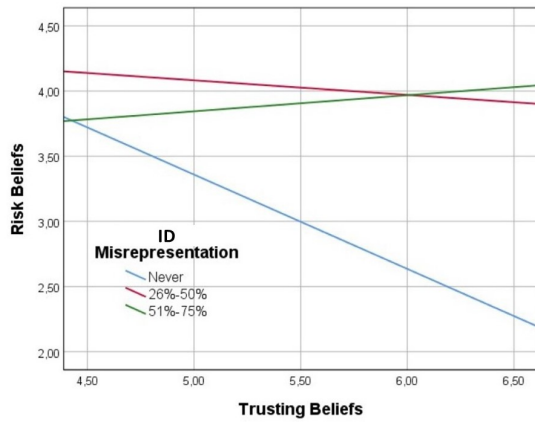
For privacy invasion, we found three moderations for (a) addressed privacy concerns and risk beliefs ($F(3, 228) = 23.812, p < .001$, predicting 23.86% of the variance), (b) trusting beliefs and risk beliefs ($F(3, 228) = 29.362, p < .001$, predicting 27.87% of the variance), and (c) risk beliefs and the willingness to use HushTweet ($F(3, 228) = 4.575, p = .004$, predicting 5.68% of the variance). They are depicted in Figure 12.8. The interaction effects are as follows: (a) $\beta = .315, t(232) = 4.83, p < .001$; see Figure 12.8a, (b) $\beta = .27, t(232) = 4.42., p < .001$; see Figure 12.8b, and (c) $\beta = .159, t(232) = 2.27, p = .024$; see Figure 12.8c.



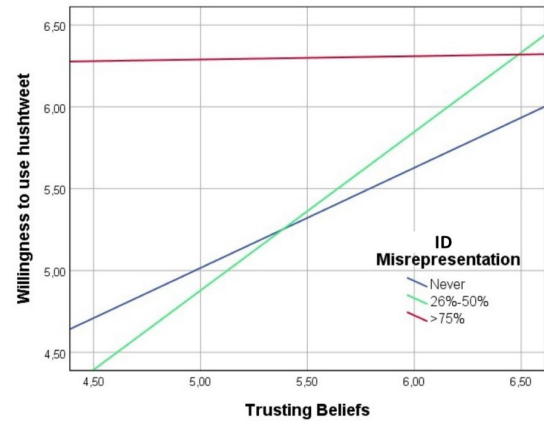
(a) Effect of ID misrepresentation on countered information privacy concerns and trusting beliefs.



(b) Effect of ID misrepresentation on countered information privacy concerns and risk beliefs.

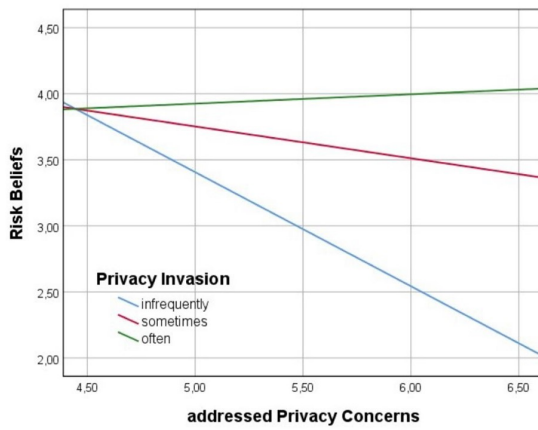


(c) Effect of ID misrepresentation on countered trusting beliefs and risk beliefs.

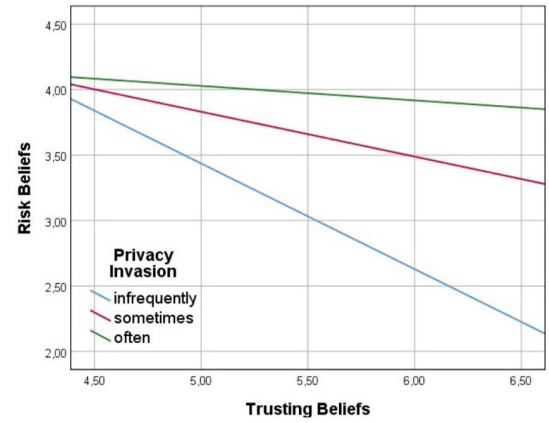


(d) Effect of ID misrepresentation on trusting beliefs and the willingness to use HushTweet.

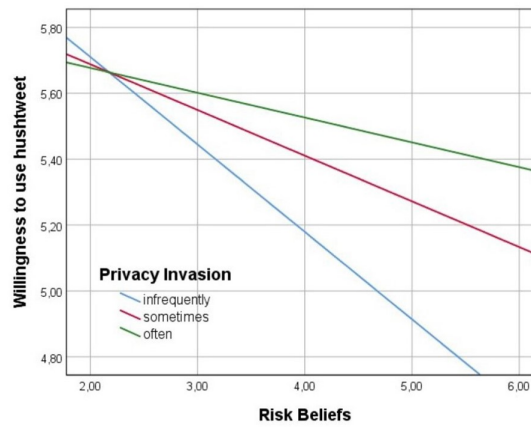
Figure 12.7: Full-featured group - simple slopes for the moderation effects of identification misrepresentation.



(a) Effect of privacy invasion on addressed information privacy concerns and risk beliefs.



(b) Effect of privacy invasion on trusting beliefs and risk beliefs.



(c) Effect of privacy invasion on risk beliefs on the willingness to use HushTweet.

Figure 12.8: Full-featured group - simple slopes for the moderation effects of privacy invasion.

12.6 Discussion

This study answers two major research objectives. Firstly, an understanding of the HSM context is established regarding information privacy concerns, trusting beliefs, risk beliefs, and the willingness to use HSM. Secondly, information privacy concerns in HSM have been addressed by applying TrustSoFt for eliciting trust-related software features. Thereby, the impact of TrustSoFt software features is analysed on people's information privacy concerns, trusting beliefs, risk beliefs, and willingness to use the HSM application HushTweet.

In this section, the results of the online user survey are discussed concerning (1) the relevance of the information privacy concerns, (2) the relationships of the research model variables, (3) the meaning of the demographic and issue-related variables for the development of HSM, (4) the impact of trust-related software features on users, and (5) lessons learnt about TrustSoFt. The last two topics also refer to Research Question RQ7 "How do software features resulting from software development to support users' trustworthiness assessment impact users?". They are picked up again in the discussion of the research questions in Chapter 15.2.8. At the end of this chapter, the limitations and future work of this study are discussed.

12.6.1 The Relevance of Information Privacy Concerns

Although the literature has introduced the six information privacy concerns as relevant in the context of the Internet, the results of this study suggest that some concerns are more relevant than others for HSM users. The results show that for HSM users "Unauthorised Secondary Use" is the most relevant concern followed by the "Awareness of Privacy Practices" and "Improper Access". In contrast, "Errors" and "Collection" are the least relevant information privacy concerns. The results are aligned with the findings of Smith et al. [280], who also found "Unauthorised Secondary Use" and "Improper Access" as higher in concern than "Errors" and "Collection".

Concerning "Collection" and "Errors", their irrelevance for users is partly supported by the weak factor loadings of the SEM, which means that they do not contribute enough to the participants' overall information privacy concern. The un-

acceptable internal consistency of the “Collection” sub-scale further is an indicator that the sub-scale may not be the correct instrument for measuring “Collection” in the HSM context. Reasons for that may be that the participants seem to believe that the actual purpose of HSM is the limitation of personal data collection. Therefore, the actual concern of “Collection” may not be applicable to HSM users as it is defined from former research before. In the case of the “Errors” concern, the HSM context can be a reason why it is weakly manifested. HSM leverages encrypted and distributed data storage, which contributes to a lower risk of malicious attacks on personal data. Therefore, people might be less concerned about errors in their data.

12.6.2 Relationships of the Constructs - the Research Models

Partly unexpected were the relationships between information privacy concerns, trusting beliefs, risk beliefs, and the willingness to use HSM. Information privacy concerns do not negatively affect trusting beliefs in HSM, which is contrary to former research about information privacy concerns [196, 208]. Therefore, this finding suggests that information privacy concerns are detached from trust in HSM. A similar finding was made by Kusyanti et al. about Indonesian teen Facebook users, whose trust in Facebook was independent of their privacy concerns [185]. Kusyanti et al. argue that users seem to trust Facebook’s promise to do their best in keeping personal data safe when it comes to Facebook’s own actions. Yet, Facebook stated that they cannot guarantee the actions of third parties. A similar argument can be mapped to the HSM context. As a category of social media, HSM may be unconsciously associated with the privacy-disrupting reputation of previous data leakages triggering information privacy concerns. Yet, the purpose of HSM is to offer users a privacy-preserving alternative, which users may trust despite general information privacy concerns. This argumentation complies with the result that the addressed information privacy concerns have a positive impact on people’s trusting beliefs in HushTweet. Based on these findings, it is highly recommended to implement trust-related software features that reduce user concerns and thereby increase users’ trust in the application.

In terms of risk beliefs, it is not surprising that information privacy concerns slightly increase risk beliefs in HSM. However, addressed information privacy con-

cerns do not reduce risk beliefs. An explanation can be that the TrustSoFt software features emphasised the existence of information privacy risks by stating to have risk counter-measurements implemented. Therefore, HSM users might be consciously aware of the risks during HSM usage so that the implemented TrustSoFt software features cannot reduce risk beliefs. Yet, the finding is noticeable that with increasing risk beliefs, people are more willing to use HushTweet. Together with the before-mentioned trusting beliefs in HushTweet, this finding can be an indicator that people believe in the purpose of HushTweet to be privacy-preserving. On these grounds, they might be more willing to make use of HSM as they might believe in its risk-reducing measurements.

12.6.3 The Role of Demographic and Issue-related Variables for HSM Development

The results show different magnitudes of the analysed construct relationships depending on the type of user, i.e., regarding the demographic and issue-related variables. The different user types emerge in age, education, identification misrepresentation, and privacy invasion. No differences could be found for gender. In addition, differences appear between those participants who only know the HSM concept and those who have used the HSM application HushTweet. According to the findings of Junco, differences between these two groups are likely due to the contextual gap between hypothetically using an application and actually using it [159]. In the following, the differences per user type for both participant groups are discussed.

Age. Differences for people of different ages can be found concerning the relationships of the analysed constructs. Differences appear for both groups of participants – those who are only familiar with the HSM concept and those who have used HushTweet.

For participants, who are only familiar with the HSM concept, risk beliefs got stronger with increasing information privacy concerns. At the same time, the riskier they believed HSM to be, the less willing they were to use HSM. These effects were stronger for people aged 45 to 54 than among those aged 25 to 34. For the participants who have used HushTweet, addressed information privacy concerns led to increasing trusting beliefs and reduced risk beliefs. The effect of these two relation-

ships was stronger for people with increasing age. Yet, it is the younger people who were more willing to use HushTweet with higher trusting beliefs compared to the older ones. Still, older people were also interested in using it.

The findings indicate that with increasing age, people are more cautious concerning the HSM concept and HushTweet usage. At the same time, they are easier to convince by privacy-preserving software features in terms of trusting beliefs and risk beliefs. Goldfarb and Tucker found similar age tendencies [112]. People higher in age had higher privacy concerns and were less willing to disclose private information. Their explanation was that older people have other privacy preferences than younger ones, have long faced information technology, and are thus more aware of privacy risks.

Education. Different effects for users regarding their education were only found for those participants who have used HushTweet. Participants with high school graduation believe HushTweet to be less risky the more their information privacy concerns are addressed by the TrustSoFt software features. This is the opposite for people with a Master's degree, whose risk beliefs slightly increase the more their information privacy concerns are addressed.

The negative effect of addressed information privacy concerns on risk beliefs for people with high school graduation complies with the findings of Malhotra et al. [208]. The positive effect for people with a Master's degree might occur due to their higher level of education. A higher level of education usually leads to higher awareness of information security and associated risks [256]. As the implemented software features sensitise users to information privacy risks, which the features aim to reduce, people with higher education might become even more aware of the risks and thus become more cautious. In contrast, people with a lower educational level might be comforted by the software features, because in the end, they believe them to mitigate the information privacy risks.

Identification Misrepresentation. The participants, who misrepresented their identification in different frequencies, showed variations in the manifestations of the analysed construct relationships. This is the case for both the participants who were only introduced to the HSM concept and who have used HushTweet.

For the participants, who were only introduced to the HSM concept, risk beliefs decreased the more the people believed HushTweet to be trustworthy. The effect is smaller for people who sometimes misrepresent their identity than for people who never have done it. Similar results were found by Malhotra et al. [208]. Regarding the participants who have used HushTweet, the same results were found. However, for them, results are also found for people who often misrepresent their personal information. In this case, people believe HushTweet to be riskier with increasing trusting beliefs. Although the positive effect is weak, it is surprising that it is not negative. The results differ from the findings of Malhotra et al. [208]. It is assumed that people, who more often misrepresent their identity online, generally have a higher risk awareness. As the implemented TrustSoFt software features emphasise information privacy risks by aiming to reduce them, people with general high-risk awareness might become more cautious even though their trusting beliefs increase.

This explanation may also be true for the prediction of addressed information privacy concerns on risk beliefs. There, a similar phenomenon can be observed for people with different frequencies in their identification misrepresentation. People, who never misrepresented their identity, believe HushTweet to be less risky the more their information privacy concerns are addressed. For those, who sometimes falsified their identity, addressed information privacy concerns do not have any effect on their risk beliefs. In contrast, people who often misrepresent their identity have increasing risk beliefs the better their information privacy concerns are addressed.

Furthermore, effects have been found for the positive prediction of addressed information privacy concerns on trusting beliefs for people who never misrepresented their identity. Their trusting beliefs are in general higher than for people who more frequently misrepresent their personal information. Therefore, it is assumed that people who never performed identification misrepresentation have a high trust propensity. Trust propensity describes the general personal predisposition to trust others [71]. It is also discussed to be the reason for the self-disclosure of personal data. Heirman et al. found that despite privacy concerns, trust propensity predicts the self-disclosure of personal information in exchange for commercial incentives [130]. The trusting beliefs of people who never falsified identifiable information are only exceeded by the trusting beliefs of those who very often misrepresent their identity — but only when their information privacy concerns were highly mitigated.

Last but not least, effects are also found for the positive prediction of trusting

beliefs on the willingness to use HSM and HushTweet. This prediction is increasingly strong for people, who never, sometimes, and very often misrepresent their identifiable information. However, trusting beliefs are less relevant for people, who very often misrepresent their identity. Their willingness to use HSM applications is exceptionally high in general. Again, it is assumed that there are people who are highly aware of their privacy and associated risks. For them, ID misrepresentation is a privacy protection strategy [152]. Therefore, HSM applications might be especially appealing to people that very frequently misrepresent their identity due to their privacy-preserving characteristics.

Privacy Invasion. Moderating effects for user differences in privacy invasion were only found for participants who interacted with HushTweet. With increasing experiences in privacy invasions, the negative predictions of (a) addressed information privacy concerns on risk beliefs, (b) trusting beliefs on risk beliefs, and (c) risk beliefs on the willingness to use HushTweet are weaker. However, there is a turnaround for people, who experienced privacy invasions often. They slightly believe HushTweet to be riskier the more their privacy concerns are addressed.

The results make sense, considering that people with increasing experiences of privacy invasion become more sensitive to privacy and social media risks [324]. People who are more aware of social media risks tend to use social media less often [324]. The findings of Yang reflect why risk beliefs are higher among users who have experienced more privacy invasions than among users who have experienced them less often. The presumed higher risk awareness may explain why risk beliefs increase for people with more frequent privacy invasion experiences despite that their information privacy concerns are addressed and their trusting beliefs in HushTweet. In the end, HushTweet still is a social media application. Yet, HushTweet's privacy-preserving functionalities seem to be especially appreciated by people who have made more experiences with privacy invasion. This may explain the effect that they are more willing to use HushTweet.

12.6.4 Limitations and Future Research

HSM is a technology that is not widely known. Moreover, it is relatively complex and not easy to understand for regular users [311]. For these reasons, all participants

of the online user survey conducted for this study got introduced to the exemplary HSM application HushTweet. Thereby, the understanding of the HSM concept should be facilitated by a tangible example. In addition, HushTweet enabled the analysis of the impact of TrustSoFt software features on the users. However, introducing HushTweet to the participants has the limitation that participants can only indirectly react to the general concept of the HSM technology. Their answers are likely to be biased based on the design and usability of the specific HSM application HushTweet. Therefore, this study is limited to the scope of HushTweet. Nonetheless, the given answers are useful for the development of other HSM applications. After all, only participants were included in the analysis who comprehended the HSM concept.

Concerning the design and usability of HushTweet, the participants of the user study provided positive feedback. Their feedback reflects a high likability of HushTweet and fun during the usage. Furthermore, the feedback indicates that HushTweet complies with the quality of situational normality – meaning that the application is perceived as proper, and originating from a serious, success-oriented service provider [218]. Providing an appealing user interface is important for positively impacting the perception and performance of users during software use [286]. On these grounds, aiming at situational normality by an appealing design was necessary to demonstrate HushTweet users the privacy-preserving advantages of HSM and thereby mitigating information privacy concerns. However, the appealing user interface may lead to biases or intervening effects on users' information privacy concerns or trusting beliefs, such as from factors like branding or marketing.

Another limitation is the focus on information privacy concerns. Participants have stated additional concerns that are not about information processing and privacy but concern economic aspects of the service provider, such as the business model of HSM service providers. Future work needs to tackle these concerns as well in order to adequately develop HSM applications. Furthermore, in accordance with the information privacy concerns, the TrustSoFt software features have been selected to counter the concerns. The selection of the features was subject to the principle that the features are similar in design and message for comparability reasons. Yet, it still cannot be ensured that the software features for the various concerns countered the concerns equally strongly. Moreover, the software features can still impact the users in different ways. Therefore, it is interesting for future work to examine each

software feature and its impact individually. Furthermore, each feature could be modified for various user types. Based on the findings, tailored features for age groups or for people with various experiences in privacy invasion might be useful to better address their trusting and risk beliefs.

With the analysis of user differences concerning the examined constructs, it became apparent that additional variables should be considered for future work. Knowledge about variables like risk and privacy awareness or trust propensity would shed light concerning certain interplay of (addressed) information privacy concerns, trusting and risk beliefs, and the willingness to use HSM for the various user types. Thereby, HSM applications can be better tailored to their users.

12.6.5 The Impact of TrustSoFt Software Features

Implementing TrustSoFt software features for the information privacy concerns has a different effect than expected. Only the software features targeting to mitigate the “Errors” concern actually addressed the “Errors” concern the most compared to the other information privacy concerns. The other TrustSoFt software features scored higher in addressing other information privacy concerns than the one they were intended for.

A reason for this phenomenon might be that the “Errors” concern and the respective software features differed from the features of the other concerns. While other information privacy concerns are unobservable when they happen, errors are often-times observable because they can hinder a system from functioning. Yet, error-free, functioning systems are presupposed by users [95]. However, the prototype of the “Errors” group involved an error so that a counter-measure feature could be demonstrated. Thereby, the users’ “Errors” concern is confirmed by HushTweet before it got addressed by the application. In contrast, the other information privacy concerns are not directly confirmed in the prototypes. Instead, the concerns are not directly tangible but indirectly present through the implemented, countering software features. Based on this assumption, the participants are highly aware of the “Errors” concern and the counter-measure software features. In contrast, the other concerns are less present in the prototypes than the “Errors” concern. The awareness of errors in HushTweet may also be the reason, that the trustworthiness facet “integrity” is rated the lowest in this group. By the presented error, HushTweet

demonstrates to act less trustworthy.

Moreover, some of the information privacy concerns might be relatable to the other concerns. “Awareness of privacy practices” is one of the highest-rated concerns. When being addressed by TrustSoFt software features, the “Control” concern is remarkably mitigated as well. Therefore, it can be concluded that increasing users’ awareness of privacy practices simultaneously increases their feeling of being in control. Kani et al. support this finding. They found that software features, which create privacy awareness, support users in controlling their privacy concerns [163].

12.6.6 Lessons learnt for TrustSoFt

The objective of the TrustSoFt application in this study was to elicit software features that mitigate information privacy concerns. Yet, the information privacy concerns were identified by previous literature as relevant to the Internet context. For the TrustSoFt application, it was assumed that each concern was equally relevant for HSM users. However, the results indicate the opposite. While on the one hand, the concerns were rated differently in their relevance for HSM, on the other hand, some concerns seem to simultaneously address further concerns. An example of both phenomena is the “Awareness for Privacy Practices”. People were comparably more concerned about the applied privacy practices of HushTweet than about other information privacy concerns. In addition, as discussed in the previous section 12.6.5, software features for the “Awareness for Privacy Practices” seem to cover additional concerns, such as the “Control” concern. For the TrustSoFt application, this means that concerns need to be chosen wisely. By addressing those concerns that also relate to other concerns, software features can be elicited, whose impact is highly efficient in addressing more than one area.

In light of this observation, it is highly recommended for those applying TrustSoFt to gain deep insights into the user concerns, first. Qualitative approaches, like user interviews about their concerns, may yield knowledge about the relevance and potential trans-concern impacts of users’ needs and pain points.

Another lesson learnt is concerning the targeted trustworthiness assessment. Due to the scope of the user survey, only the basic trustworthiness facets “ability”, “benevolence”, “integrity”, and “predictability” have been measured. In the case

of the “Awareness” features, the trustworthiness facet “integrity” was targeted to reflect the integrity of the HSM application with the desire for user privacy. The results show that the participants effectively perceived the integrity of the HSM application HushTweet which in turn led to increased trustworthiness of the application.

13

Applying TrustSoFt and the Extended Feature Models for a Use Case in Online Dating

In this chapter, TrustSoFt and the extended feature models are applied to an online dating use case. The objective is to identify trust-related software features that address the trust concerns of female and male online dating users. The resulting software features can be included in online dating applications.

The purpose of this chapter is to exemplify the application of the two methods TrustSoFt and the one for establishing feature models for trustworthiness assessment. For this purpose, the application of the two methods is limited to two trust concerns of online dating users due to work scope, since their application usually leads to overarching outcomes by which whole applications can be developed. To still ensure a diversity of the resulting software features, the most relevant concerns of female and male online dating users are regarded in the following.

In the following, the application of TrustSoFt and the extended feature models are presented in four sections. The first Section 13.1 refers to Paper 9. The trust concerns of female and male online dating users are identified by semi-structured interviews. In this section, the trust concerns for the following steps of TrustSoFt are determined. The second Section 13.2 is about the identification of relevant trustworthiness facets. For that purpose, the guideline for facet identification from Paper2, Chapter 3.2.2 is applied. Section 13.3 introduces two goal models for the concerns of online dating users. By the two goal models, trustworthiness goals and requirements are specified as described in Papers 3 and 5. As the last step, Section 13.4 proposes trust-related software features for the specified trustworthiness

requirements by extended feature models according to Paper 8.

13.1 Trust Concerns in Online Dating

Over the past decade, online dating has evolved into a more and more socially accepted way of getting to know new people and mating [279]. Especially in the Covid-19 pandemic, people used online dating to flee from social isolation and overcome the physical boundaries that do not exist when interacting online [314]. By using online dating, people hope to find love, friends, travel companions, sex, entertainment, or a boost for their self-esteem [253]. Despite the beneficial potential of online dating, people have since expressed concerns about its use. User concerns are for example related to emotional vulnerability, sexually transmitted infections, or deceit [74]. Women especially emphasized safety concerns [74].

However, current research indicates not only a reduced online dating stigma but also a change in the online dating behaviour of men and women over the past decade [85]. Therefore, the assumption is that user concerns may have changed as well. Knowledge of current user needs is significant to develop software that is user-centered and up-to-date [275].

Against this motivation, Paper 9 explores the trust concerns of male and female online dating users. A special focus for users' trust concerns lies on the CMI trustees i) users, ii) service provider, and iii) online dating application as a technology. In addition, Paper 9 identifies additional concerns related to general online dating use and the impact online dating might have on the users' behavior or cognition. As a result, five different types of concerns are analyzed.

A total of 32 semi-structured in-depth interviews were conducted - 16 of them with current or former male online dating users, and 16 with females. Of the 16 men, three were homosexual while the rest was heterosexual. Regarding the female participants, one woman was homosexual, one bi-sexual, and the rest heterosexual. We aimed for an international population to prevent national dating biases. The recruiting was conducted in international social media groups on the social network Facebook, via snowballing by already interviewed participants and through word-of-mouth and personal contact. Participants received monetary compensation. The

ethics committee of the Department of Computer Science and Applied Cognitive Science of the Faculty of Engineering of the University of Duisburg-Essen approved the interview study.

The interview was designed for one hour. Ten prepared questions asked for people’s opinions on trust and online dating, their online dating concerns and workarounds, appreciated trustworthiness facets, as well as useful software features. After the interviews were conducted, they were first transcribed using the AI-based software as a service Amberscript ¹ and then manually proofread and edited by two transcribers. Afterwards, we performed a qualitative content analysis in which we combined a deductive top-down and inductive data-driven bottom-up analysis [273]. The interviews were coded into a category system.

General Online Dating Concerns of female Users	n
Being recognized offline as an online dating user	5
Being judged for online dating use	5
Being replaceable against other online dating users	2

Table 13.1: Women: General Online Dating Concerns

General Online Dating Concerns of male Users	n
Superficial dating process	4
Unauthentic communication	4
Not suitable to get to know people properly	2
Time investment	2
Dating option overflow	1
Losing anonymity	1
Racism	1

Table 13.2: Men: General Online Dating Concerns

Concerning general online dating concerns, women reported three and men seven. They are presented in Table 13.1 and Table 13.2. Women are worried that people in the physical world know that they are online dating users (n=5). Furthermore, they are afraid of being judged by other individuals for using online dating (n=5). In terms of online interaction, women are concerned about being exchangeable by other online dating users as there are many other dating options that might be a better fit (n=2). The concerns of men are that online dating is superficial because the outer appearance of people is in the foreground (n=4) and that online communication is unauthentic as chat passages can be reused for different users (n=4). Men are worried that online dating is not suitable to get to know people (n=2) while the time investment is high (n=2). Other concerns relate to the overflow of user options

¹www.amberscript.com

for dating (n=1), anonymity loss (n=1), and racism (n=1). Based on the results, we interpret that women fear negative consequences for their reputation and value as a person. In contrast, the concern of men is process-based, focusing on personal costs and sense of purpose.

Women's concerns about the impact of online dating use on their person	n
Revealing too much about their person	3
Being superficial	2

Table 13.3: Women: Concerns about Online Dating Impact on Own Person

Men's concerns about the impact of online dating use on their person	n
Dehumanizing women as a product	5
Insecurity about self-presentation	2

Table 13.4: Men: Concerns about Online Dating Impact on Own Person

Female and male users are additionally concerned about the impact online dating use has on their thought processes and behaviour. Both genders reported two concerns in specific. The concerns are listed in Table 13.3 and Table 13.4. Women are afraid of negative consequences and of making themselves vulnerable when providing personal information (n=3). Furthermore, they are concerned about becoming superficial when using online dating (n=2), as they associate online dating with judging other users based on limited, trivial information. Men are concerned about dehumanizing female online dating users as a product (n=5). They tend to evaluate them by attributes on a mental checklist men have to define suitable dating partners. Thereby, female online dating users become more artificial and less human for male users. Furthermore, men consider women to assess potential dating partners similarly. With this background, men show insecurity about their self-presentation as they fear being negatively perceived by female users when they compare them to other users (n=2). Although both female and male users are concerned about how information processing in online dating may impact themselves as a person, the underlying nature of their concerns is different. Female users relate to their feeling of vulnerability and how their character may change. Male users elaborate on a process level and how their person is involved in it.

The trust concerns of female and male online dating users that relate to other users are depicted in Table 13.5 and Table 13.6. Women are concerned about fake

Women's Trust Concerns About Other Users	n
Fake profiles	13
Safety	11
Divergent intentions	9
Dishonesty	7
Personality different on dates than while chatting	4
Receiving sexual text messages	4
Ghosting	3
Personal information misused to cause harm	2
Stalking	2

Table 13.5: Women: Trust Concerns About Other Users

Men's Concerns About Other Users	n
Fake profiles	10
Misrepresentation	8
Divergent intentions	8
Scam	6
Social bots	5
Personality different on dates than while chatting	4
little response from (female) users	3
Missing spark at offline date	3

Table 13.6: Men: Trust Concerns About Other Users

profiles (n=13) and their physical safety including sexual abuse (n=11). Moreover, they consider the intentions of other users that differ from their own (n=9). Women are further concerned about people's dishonesty in terms of conscious lies or concealing information (n=7) and personalities that seem different during chatting than they actually are offline (n=4). Additional concerns of female users are receiving sexually offensive text messages including nude pictures (n=4), abruptly ending contact (ghosting) (n=3), the misuse of personal information to harm them (n=2), and stalking in the physical world (n=2). Male users share the concerns of female users regarding fake profiles (n=10), divergent intentions (n=8), and different personalities online than offline (n=4). In contrast, men are worried about misrepresented information in dating profiles including pictures modified to positive (n=8), scams (n=8), social bots (n=5), lacking or little response from other online dating users (n=3), and missing chemistry when meeting offline (n=3).

When comparing the differences between female and male concerns, it becomes apparent that women struggle with the feeling of vulnerability and safety in a physical or sexual sense. The concerns of male users are congruent with the fear of deception or unmet expectations.

Table 13.7 and Table 13.8 present the trust concerns of female and male users about service providers. Female users are worried about how their data is processed (n=3). Another concern is that service providers control who gets to know whom

Women's Trust Concerns About Service Providers	n
Data usage	3
Pre-selection of dating options	1

Table 13.7: Women: Trust Concerns About Service Providers

Men's Trust Concerns About Service Providers	n
Profit-orientation	7
Data usage	5
Creation of fake profiles	3

Table 13.8: Men: Trust Concerns About Service Providers

since the provider presents users with a pre-selection of other users (n=1). Male users are well aware of the profit orientation of service providers. Considering this, they doubt the providers' benevolence to truly support users in finding a partner. They elaborate that this would mean that service providers accept a loss of users and thereby an omission of profit. An additional concern, according to the women interviewed, is the data use of service providers (n=5). Moreover, male users are concerned that service providers create fake profiles and use social bots in this context to drive user engagement (n=3). In terms of trust concerns regarding the service provider, female users do not have many. Male users are a bit more concerned. Their concerns are related to the underlying online dating procedures and the economic intentions of the service provider.

Women's Trust Concerns About Applications	n
Lacking ability to detect fake profiles	2
Providing application with data	2

Table 13.9: Women: Trust Concerns About Applications

Men's Trust Concerns About Applications	n
Lacking ability to detect fake profiles	2
Inability to convey offline cues	2
Providing application with data	1

Table 13.10: Men: Trust Concerns About Applications

The trust concerns of female and male users about online dating applications are depicted in Tables 13.9 and 13.10. Both female and male users are equally concerned about applications' lacking ability to detect fake profiles (n=2, n=2) and to provide applications with personal data (n=2, n=1). Male users additionally are worried that online dating applications cannot convey offline cues that male users perceive as relevant for the general dating process.

Based on the results of the interview study, Paper 9 concludes that female users

are primarily feeling vulnerable in terms of their safety, which impacts most of their single concerns. In contrast, male users are more aware of the underlying processes of online dating and are more critical of them. Their concerns relate to experiences that diverge from either their expectations of user interactions in specific or online dating processes in general. Female users are less concerned about service providers than male users. Trust concerns about online dating applications are less prominent for both genders.

For the exemplary use case scenario, one trust concern of the female and one of the male users will be addressed in the subsequent sections. The choice falls on the most frequently mentioned concern of the genders, which are not the same. For female users, this is their safety concern. For male users, it is their concern about misrepresentation in dating profiles.

13.2 Deriving Trustworthiness Facets

In this section, relevant trustworthiness facets are identified for the safety concern of female online dating users and the misrepresentation concern of male online dating users. Identifying relevant trustworthiness facets for these concerns provides input for the following TrustSoFt steps in the next sections. Thereby, digital solutions can be developed that mitigate these concerns and support female and male users to assess whether their concerns are relevant during specific interactions with the other gender. The focus lies on heterosexual interaction.

To identify relevant trustworthiness facets, the guideline from Paper 2 explained in Chapter 3.2 is applied. After the problems to be addressed have been chosen (see Figure 3.2 on page 40, 1. step), an understanding of the problems is attained (2. step). For that reason, the interviews of Paper 9 and additional literature are considered. The details of the two problems inherent in the concerns are as follows:

Concern of female online dating users: Safety. In the interviews, female users reported safety concerns. They feel more vulnerable to physical or sexual assault while heterosexual male users have not mentioned safety concerns at all. In terms of physical assaults, female users stated concerns about physical aggression against them, kidnapping, or murder. Furthermore, women fear

that other users may find out via online dating applications or through other online channels where they live or hang out in the offline world. They are afraid that this knowledge may be used to stalk or overpower them. For these reasons, women regard the feature of the online dating application to display the distance to other users using GPS technology as critical. Regarding sexual assaults, female users are afraid of being pushed to do sexually more than they want to do, up to and including rape. Milder forms of unwanted sexual approaches are online sexual harassment in form of sexual text messages. Women reported having received messages that call for sex or include male genital pictures (also called dick pics). In the interviews, women agreed that their safety concerns are related to the fear of losing control and the freedom to do what they wish to do. The safety concern of female online dating users is confirmed by other works, such as Couch and Liamputtong [73], Gillett [109] or Pruchniewska [252]. Kalra and Bhugra argue that women's fear of losing control relates to real or perceived unequal power relations that involve the biological, social, or cultural inferiority of women compared to men [161]. To cope with their safety concerns, female online dating users apply various safety strategies [109]. These are, among others, to only meet other users in public where people are around, dressing conservatively to not give "a wrong idea" that could be understood as a sexual invitation, or sending friends the live-location during offline dates via smartphone by GPS.

Concern of male online dating users: Misrepresentation in online dating profiles.

Male online dating users reported in the interviews their concern about misrepresented online dating profiles. Misrepresentation is a distortion of the truth, where personal information is consciously and intentionally adjusted without pretending to be someone else [92]. Misrepresentation differentiates from fake profiles concerning the degree of deviation in the self-presentation. Fake profiles represent an identity different from the one that has created the profile [183]. Hall et al. identified seven categories of misrepresentation in online dating, which concern personal assets, relationship goals, personal interests, personal attributes, past relationships, weight, and age [124]. While men are more likely to misrepresent personal assets, relationship goals, personal interests, and personal attributes, women tend to misrepresent weight [124]. In the interviews from Paper 9, male users reported being concerned about misrepresented profile pictures, in which women use filters or wear heavy make-up.

Male users assume that users misrepresent their profiles to brag or to present their exaggerated positive or best version of themselves for receiving positive feedback online (e.g. a match). This assumption is confirmed by Ellison et al.[92], who discovered that online dating users try to find a balance between the pressure of impression management and presenting an authentic self. Impression management describes procedures by which individuals try to make a positive impression during the initial stages of getting to know new people, like in dating being perceived as attractive [184]. Ellison et al. found out that users often solve their inner conflict by presenting their "ideal self", which is a version they want to become in the future [92]. In the interviews of Paper 9, male users acknowledge that it is hard to recognize misrepresentation online. Most often, they discover misrepresentation on the first date offline.

After the knowledge about the concerns is attained, the next step is to specify the problematic and desired characteristics of the actual problem. According to the guideline for identifying trustworthiness facets, specifying problematic and desired characteristics can be realized by drawing own conclusions (see Figure 3.2 on page 40, Step 3.a.), refer to literature or experts (Step 3.b.), or ask users directly (Step 3.c. & 3.d.).

Since online dating users were asked in the interview study of Paper 9 about the trustworthiness facets they would like to see in other online dating users, the identified facets are considered here as initial input for the use case. However, the interviewees were asked for relevant trustworthiness facets in general and not for those related to the two concerns of this use case in specific. Therefore, the trustworthiness facets from the interviews are reported in total to be then evaluated regarding the relevance of the two trust concerns. Those that are considered relevant will be addressed in the following use case. In addition to them, further trustworthiness facets are sourced in additional literature. By this procedure, the guideline for trustworthiness facet identification is applied by consulting users, drawing own conclusions for specific concerns, and checking on literature.

In the interviews, online dating users were asked what characteristics other users should have to be trustworthy. This question aims for desired characteristics (Step 3.c.). Another question was what reduces trust in others. As a reply, some participants mentioned problematic characteristics (Step 3.d.). Female online dating users reported that male users need to be authentic, communicative, friendly, hon-

est, humorous, likable, open, open-minded, respectful, or social they perceive them as more trustworthy. In contrast, female users believe that users who seem arrogant, exaggerated in their way or frequency of chatting, disrespectful, unfriendly, self-centered, or spontaneous planners are less trustworthy. Male online dating users reported that attentiveness, authenticity, emotional stability, friendliness, honesty, humor, integrity, likability, open-mindedness, openness, and reliability are desired characteristics for trustworthiness. On the contrary, problematic characteristics are arrogance, incommunicability, insecurity, narcissism, negativity, or superficiality.

After the desired characteristics have been determined, they are compared to the overview of trustworthiness facets (see Appendices A, B, and C) to identify whether facets are among them (Step 5). Moreover, the antonym dictionary *thesaurus*² is used to specify the semantic opposites of the problematic characteristics (Step 4). The antonyms, which are desired characteristics in terms of their meaning, are then also checked for a match using the overview of trustworthiness facets.

The identified trustworthiness facets for female users are depicted in Table 13.11 and for male users in Table 13.12. Most of the facets are identical to what the users have said in the interviews. As an example, female and male users have stated authenticity as an important characteristic to trust other online dating users. Enli and Rosenberg have identified perceived authenticity to positively impact perceived trustworthiness, which is thus a facet [93]. Some of the desired characteristics from the interviews are relatable to the trustworthiness facets of the overview. The characteristic communicativeness relates to the facets willingness to disclose and openness, while sociability is involved with social desirability from what the interview participants have explained in more detail. Therefore, these facets have been added to Tables 13.11 and 13.12. Concerning the problematic characteristics, the antonym webpage thesaurus provided desired characteristics that are also related to trustworthiness facets in the overview (e.g. modesty is related to humbleness).

Now that the trustworthiness facets from the interview study in Paper 9 have been identified, they are reviewed for their relevance to the safety concern of female online dating users and the misrepresentation concern of male users. The review involves checking for each facet if it will necessarily be violated if the concern were to arise. Those for which this is true are relevant trustworthiness facets for the concern in question. By being able to assess these facets through the software, users

²www.thesaurus.com

Problematic characteristics of female users	Desired characteristics of female users	Trustworthiness Facets for Female Users
	Authenticity	Authenticity
	Communicativeness	Willingness to disclose Openness
	Friendliness	-
	Honesty	Honesty
	Humor	-
	Likability	Likability
	Open-mindedness	Openness
	Openness	Openness
	Respectfulness	Respectfulness
	Sociability	Social desirability
Arrogance	Modesty	Humbleness
Exaggeration	Truth	Truthfulness
Disrespectfulness	Respectfulness	Respectfulness
Unfriendliness	Friendliness, goodwill	Goodwill
Self-centeredness	Humility	Humbleness
Spontaneity	-	Predictability Promise Fulfillment Empathy

Table 13.11: Overview of the Identification of Trustworthiness Facets for Female Online Dating Users. The Coloured Trustworthiness Facets are Going to be Considered for Women’s Safety Concern.

can better evaluate whether their concerns are relevant to the individual user.

For example, someone who violates another person’s safety may be generally honest or likable. Therefore, these two facets are irrelevant to safety. However, hurting other people is incompatible with respectfulness or goodwill towards the person. Moreover, such behaviour deviates from the usual online dating behaviour and from the agreed usage terms of the online dating application to behave appropriately. Those online dating users are not predictable and do not keep promises. Based on these reflections, the trustworthiness facets respectfulness, goodwill, predictability, and promise fulfillment are relevant to the safety concern of female online dating users. They are marked grey in Table 13.11. In terms of misrepresentation, the trustworthiness facets authenticity, honesty, integrity, willingness to disclose, and openness would be hurt. They are considered relevant to the misrepresentation concern of male online dating users and marked grey in Table 13.12.

Now that users have been interviewed and conclusions have been drawn, litera-

Problematic Characteristics for Male Users	Desired Characteristics for Male Users	Trustworthiness Facets for Male Users
	Attentiveness	Attentiveness
	Authenticity	Authenticity
	Emotional stability	Emotional stability
	Friendliness	-
	Honesty	Honesty
	Humor	-
	Integrity	Integrity
	Likability	Likability
	Open-mindedness	Openness
	Openness	Openness
	Reliability	Reliability
Arrogance	Modesty	Humbleness
Incommunicability	Communicativeness	Willingness to disclose Openness
Insecurity	-	-
Narcissism	Humility	Humbleness
Negativity	Positivity	-
Superficiality	Seriousness	-
		Agreeableness

Table 13.12: Overview of the Identification of Trustworthiness Facets for Male Online Dating Users. The Coloured Trustworthiness Facets are Going to be Considered for Men's Concern about Misrepresentation.

ture is additionally consulted. Concerning the evaluation of whether someone might endanger one's safety, it is relevant to know what characteristics constitute perpetrators. Research has identified some characteristics that are significantly relatable to abusers. These are for example dominance [51] or a lack of empathy [111]. Checking the overview for related characteristics to these two, empathy has been identified as a trustworthiness facet in research before [316]. It is added to the facets to be considered in the following method steps and marked grey in Table 13.11.

The same procedure is now carried out for the misrepresentation concern. Literature has shown that the characteristics honesty [92], openness, and agreeableness [124] are negatively related to misrepresentation. Honesty and openness have been identified as relevant trustworthiness facets for misrepresentation before. Agreeableness has been identified as positively impacting perceived trustworthiness as well [22]. It is added to the relevant trustworthiness facets for the use case.

13.3 Requirements Elicitation and Goal Modelling

Now that the trust concerns and trustworthiness facets have been identified, the next step is to specify trustworthiness goals and requirements. For that purpose, i* goal models are created as described in Chapter 4. As this is an exemplary use case, for each concern one goal model is created to demonstrate TrustSoFt and exemplify resulting software features for online dating applications. The start is made by the goal model for the safety concerns of female users, followed by the goal model for the misrepresentation concern of male users.

Applying TrustSoFt on both use cases demonstrates how goal models are created and used for requirements and feature elicitation. Yet, the two exemplary goal models do not have any conflicts. For an example of how to proceed for conflict identification and resolution, reference is made to Chapter 6.3.

Trustworthiness Facet	Definition	Reference
Agreeableness	A personality characteristic that describes the tendency to be cooperative, compassionate, and trusting. It is the opposite of being inconsiderate, suspicious, and pessimistic.	[22]
Authenticity	The true expression of an individual's values and beliefs leading to sincere choices about actions that are true to an individual's self instead of being scripted by social norms.	[239, 57]
Empathy	An emotional response of an individual based on the emotions of another individual. The emotional response resembles the emotion of the other person, while the recognition is given that the source of the emotion is not one's own.	[77]
Goodwill of others.	An individual's intention to attend to the interests [139]	
Honesty	Any behaviour that is in line with the truth.	[26]
Integrity	A trustee complies with the trustor's accepted principles and has the reputation for being honest and truthful.	[213, 139]
Openness	"mental accessibility, or the willingness to share ideas and information freely with others"	[139]
Predictability	An individual's ability to forecast the actions of another party.	[248]
Respectfulness	The trustee regards "others and their perspective as valuable"	[316]
Willingness to disclose	"An individual's willingness to reveal personal information [...] online"	[237]

Table 13.13: Overview of Relevant Trustworthiness Facets for Use Case Including Definitions.

Trust Concern	User Goal	Trustworthiness Goal
Male users can physically or sexually assault me on a date	Safety check of users online	Enabling safety check
Male users can physically or sexually assault me on a date	Safe dates	Safe date support
Male users collect data about me online to harm me offline	Cautious information disclosure	Sensible information control

Table 13.14: Expressions of the safety concern of female online dating users, possible user goals, and potential trustworthiness goals.

13.3.1 Goal Model for Safety Concern of Female Online Dating Users

The goal model for the safety concern of female online dating users can be seen in Figures 13.1 on page 196 and 13.2 on page 198. Figure 13.1 depicts the SD model and the SR model of female online dating users and the application. Yet, the SR model of the application only contains the intentional elements that relate the application to the other actors. Due to its large size, the complete SR model of the application is presented in Figure 13.2.

The goal model creation starts with the SD model. For the safety concern, the actors *female online dating users*, *male online dating users* and the *online dating application* as an instantiation (connected by INS-link) of the *online dating service provider* are involved. The perspective of female online dating users is taken because their safety concern is addressed. Therefore, their actor element receives a black bold frame.

As a next step, the trust concern of female users is added inside their actor boundary. The safety concern implies different expressions of fear. One is that *male users can physically or sexually assault [women] on a date*. This concern is addressed in the goal model. Another expression of the safety concern is that male users collect data about female users online which enables them to harm women offline. To counter the safety concern, possible user goals and potential trustworthiness goals based thereon are depicted in Table 13.14. The table can be extended.

For the concern to be addressed, the user goal *safety check of users online* is added to the goal model. Female users have stated in the interviews from Paper 9 that

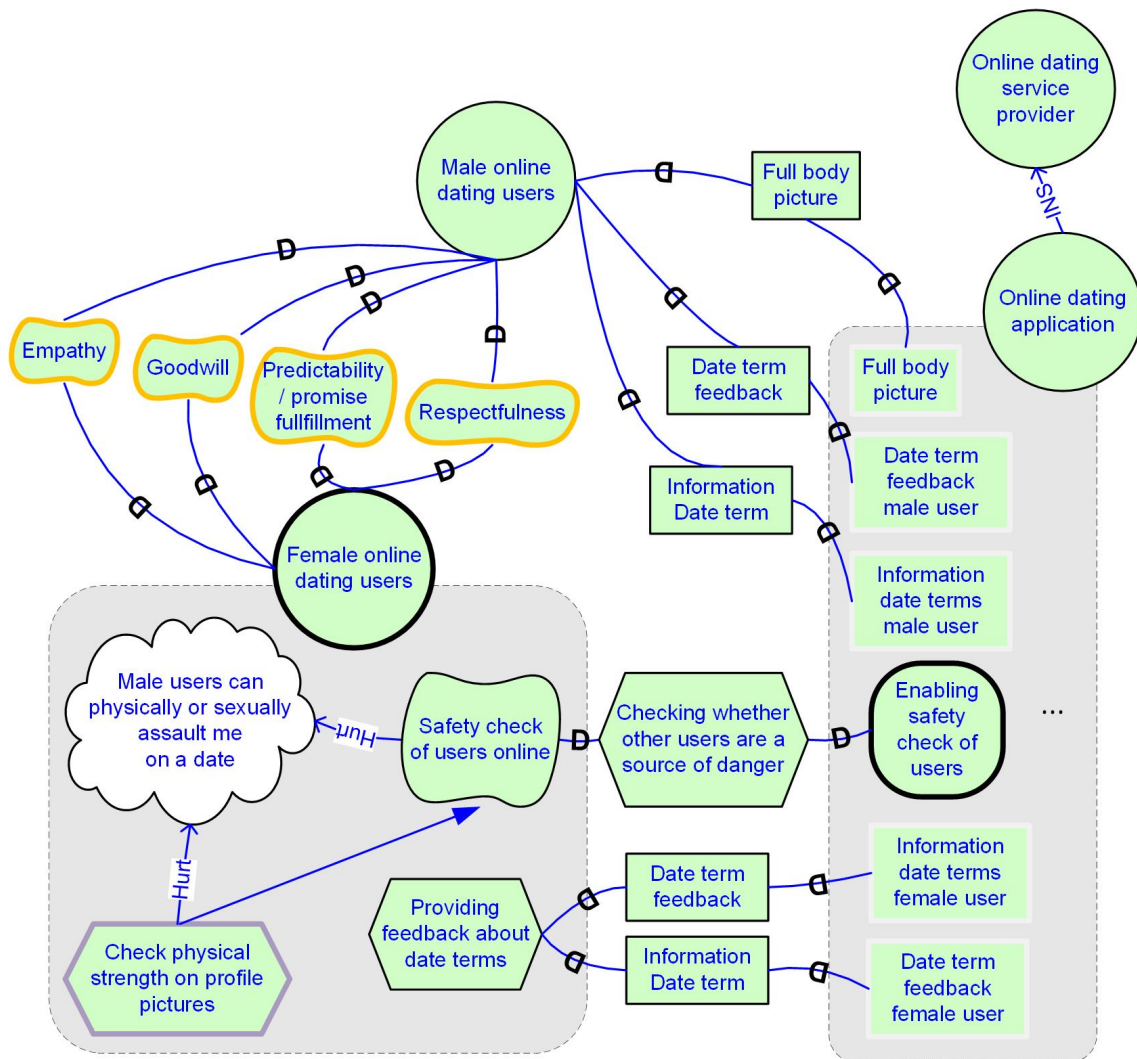


Figure 13.1: Goal model for safety concern. This model only contains the application with its intentional elements in dependence on the other actors. The full insights of the application are given in Figure 13.2

they try to assess the online dating profile of male users concerning the danger male users may pose to their safety. As this check is based on a subjective evaluation, the user goal is modelled as a soft goal that reduces the trust concern a bit (hurt contribution link). In the interviews, female users reported that one approach for the safety check is to *check [the] physical strength* of male users based on uploaded pictures. Although this is not a workaround in the strict sense, because female users use the online dating app for the evaluation, it is a counter strategy that is used without an explicit software feature. Therefore, it is included in the model in form of a workaround for potentially being picked up later as a requirement or feature. The workaround is related to the user goal by a means-end link. Furthermore, a hurt contribution link connects the workaround with the trust concern, because it reduces the trust concern of female users but does not eliminate it.

After including the safety concern to the boundary of female online dating users, the dependencies of the involved actors are modelled for gaining an understanding of the safety concern context. For that reason, the trustworthiness facets that have been identified in the previous chapter are included as soft goal dependencies from the female users to the male users. Female users wish male users to show *empathy, goodwill, predictability/promise fulfillment* and *respectfulness* to trust them that they will not endanger their safety. For modellers, these user facets need to be picked up by the trustworthiness requirements of the application later in the modelling process. Thereby, software features can be designed that reflect these facets for users' trustworthiness assessment.

As a next step, the dependency of female users on the online dating application is modelled. The application needs to realise the user goal "safety check of users online". Therefore, a task dependency is created from the female users to the application. Via the system, female users want to complete the task of *checking whether other users are a source of danger*. For that reason, the trustworthiness goal of the application is to *[enable the] safety check of users*.

Now, that the trustworthiness goal has been specified, the intentional elements of the application are modelled. They are depicted in Figure 13.2. To achieve the trustworthiness goal, two trustworthiness requirements are specified. The application needs to *present cues for [the] safety evaluation of each user*. To do so, it needs to *collect data about **safety criteria** of [each] user*. *Safety criteria* is depicted in bold because it is a term that needs further elaboration and is a relevant element

that has to be added to the goal model. By safety criteria, characteristics for the safety check are meant. These are amongst others the trustworthiness facets that have been identified for male users. In addition, a safety criterion is the *physical strength* of male users that female users have stated to evaluate as their workaround. To collect data about safety criteria, sub-requirements have to be specified. Unfortunately, only sub-requirements for the criteria predictability/promise fulfillment, user respectfulness, and physical strength could be decomposed. Reflecting users' empathy and goodwill through software requirements still is a challenge that needs to be realized in future work.

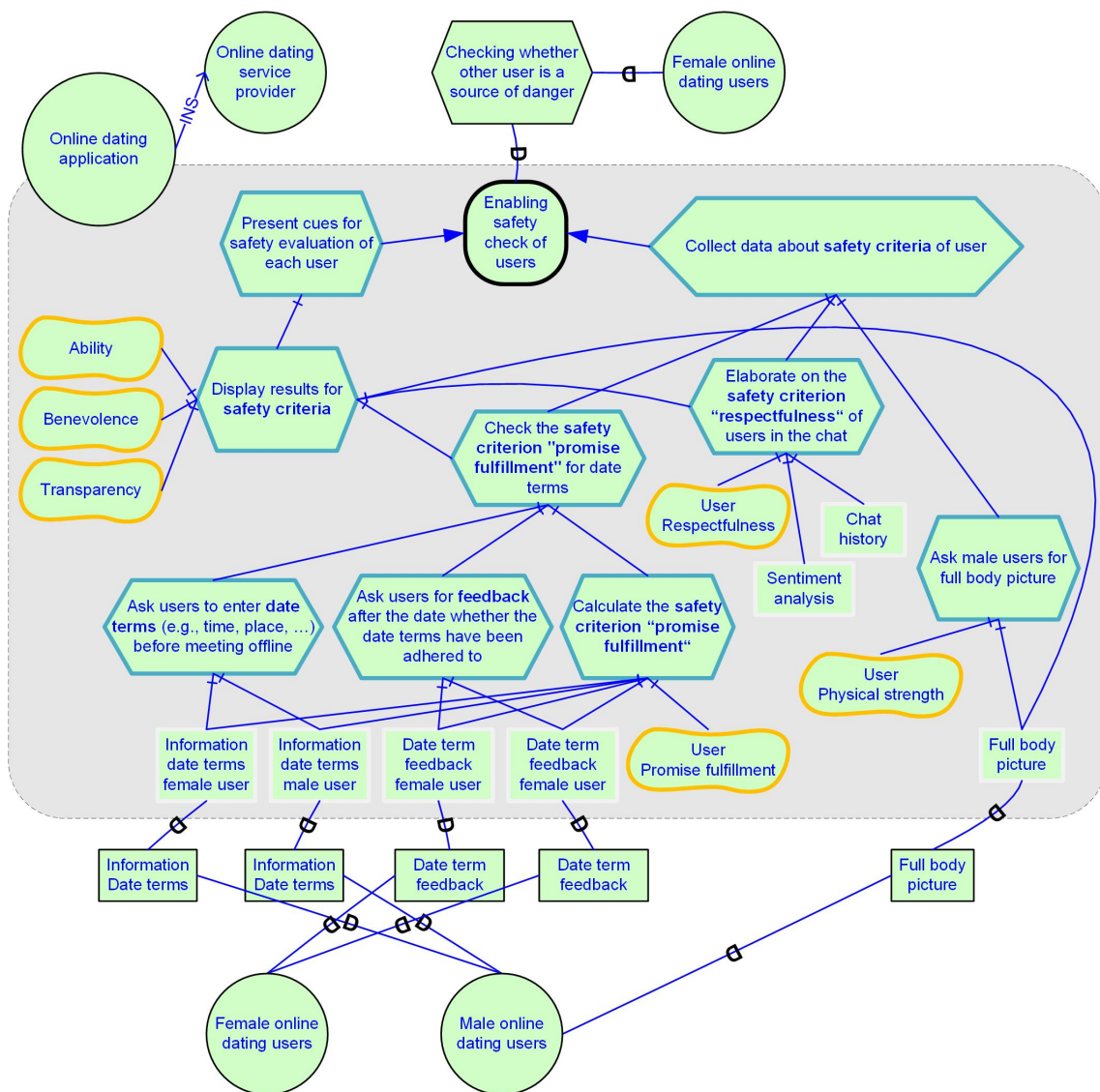


Figure 13.2: This is the complete application with its intentional elements, which is part of the goal model in Figure 13.1

One of the sub-requirements is to *check the **safety criterion “promise fulfillment” for date terms***. The term “safety criterion promise fulfillment” is still marked bold, because it is not yet an element within the model itself. The requirement is further refined in three sub-requirements. The first is to *ask users to enter **date terms** (e.g., time, place, ...) before meeting offline*. This requirement is further decomposed into the two feature elements *information date terms female user* and *information date terms male user*. This set of requirements demands the system to ask a female and a male user to enter key data, such as time and location, about an offline encounter before it happens. The application needs the resource *information [about the] date terms* from each user, which is depicted as a resource dependency link from the application to each of the users. The task *providing feedback about date terms* is included in the actor boundary of the female online dating users to illustrate the dependency correctly. The second sub-requirement for the safety criterion promise fulfillment is to *ask users for feedback after the date whether the date terms have been adhered to*. “Feedback” is written boldly because it needs to be included in the application as an own element. This is done by a decomposition link that relates the feature elements *date term feedback female user* and *date term feedback male user* to the sub-requirement. As this feedback is needed by the female and male users, two resource dependencies are modelled from the application to the two end-users with the dependuums *date term feedback*. For female online dating users, the dependency is connected to their task of *providing feedback about date terms*. The third sub-requirement is to *calculate the **safety criterion “promise fulfillment”***. It is decomposed by the trustworthiness facet promise fulfillment of the user because it results from the calculation. To calculate users’ promise fulfillment, the sub-requirement makes use of the feature elements (decomposition links) that have been decomposed by the other two sub-requirements before.

The second sub-requirement of the trustworthiness requirement for collecting data about the safety criteria of users is to *elaborate the **safety criterion “respectfulness” of users in the chat***. The sub-requirement is decomposed into the trustworthiness facet *user respectfulness* and the feature elements *sentiment analysis* and *chat history*. This set of elements describes that the users’ respectfulness is elaborated by the application based on the chat history using sentiment analysis. Sentiment analysis is based on algorithms that detect people’s opinions, attitudes, and emotions by analysing natural language [222]. It is an approach by which the application can estimate to what extent a user is respectful to others.

The third sub-requirement for collecting data about the safety criteria is to *ask male users for [a] full body picture*. This requirement picks up the workaround of female users to estimate the physical strength of male users by the uploaded pictures. Therefore, the sub-requirement is decomposed to the trustworthiness facet *user physical strength* and the feature element *full body picture*. The system is dependent on the male users to provide the picture, which is presented by the resource dependency *Full body picture* from the application to male online dating users.

After the application has collected data about the safety criteria of users, it can realize the trustworthiness requirement to *present cues for [the] safety evaluation of each user*. The requirement is refined into its sub-requirement *display results for safety criteria*. By displaying the results for safety criteria to enable users' mutual trustworthiness assessment, the application demonstrates its trustworthiness facets *ability, benevolence* and *transparency*. This requirement may positively impact users' trust in the online dating application if they appreciate the trustworthiness requirements. To display safety criteria, the sub-requirement needs to have access to the data. Therefore, it is further decomposed to the feature element "full body picture" and the two sub-requirements presented before, which are "check the safety criterion promise fulfillment for date terms" and "elaborate the safety criterion respectfulness of users in chat".

13.3.2 Goal Model for Misrepresentation Concern of Male Users

The misrepresentation concern of male online dating users is modelled as goal models in Figures 13.3 and 13.4. Figure 13.3 depicts the SD model and the SR model of male online dating users and the online dating application. However, the SR model for the application only contains the intentional elements that are dependent on the other actors. For reasons of size and clarity, the complete SR model of the application is illustrated in Figure 13.4 on page 204.

Goal modelling starts with the SD model to represent the context of the misrepresentation concern. The concern involves the actors *male online dating users, female online dating users* and the *online dating application* as an instantiation (INS-link)

of the *online dating service provider*. As the application aims to mitigate the concern of male users, they are the main actors and receive a bold frame.

Taking the perspective of male online dating users, the first action is to model the misrepresentation concern. Again, the trust concern can be expressed in multiple ways. A few are depicted as examples in Table 13.15 together with possible user goals and potential trustworthiness goals. In general, male users are concerned that users do not take the truth too seriously, which is why they aim to recognise fibs. The application can support them by aiming for a misrepresentation check. To be more specific with misrepresentation, former research has identified that female users fib about their weight. This expresses another concern of male users. They aim to date women who actually have the weight they stated online. The application needs to check whether women might have lied about their weight, which is called weight check.

Another expression of the misrepresentation trust concern is that male users do not want to be perceived as a worse dating option than other male users. Different from the other misrepresentation concerns, male users reflect on themselves whether they resemble the misrepresenting women they are concerned about. Since male users understand the reason for the misrepresentation but refrain from it themselves, their user goal is to present themselves in the best way possible while still being authentic online. The application may aim for an authenticity check to realise the goal of male users.

For the goal model presented in this work, the trust concern that *female users look different in their profile pictures than in reality* is modelled. This concern has been stated in the interviews of Paper 9. To counter the trust concern, the user goal of male users is to *[date] women who look like in their profile pictures*. For that reason, some male users want to *video chat before meeting offline* or *check social media channels for additional pictures*. These are workarounds, which are added to the model. The first workaround can eliminate the trust concern in single cases (break contribution link). The latter workaround reduces the concern (hurt contribution link). Both workarounds are means for users to achieve their user goals on their own (means-end links).

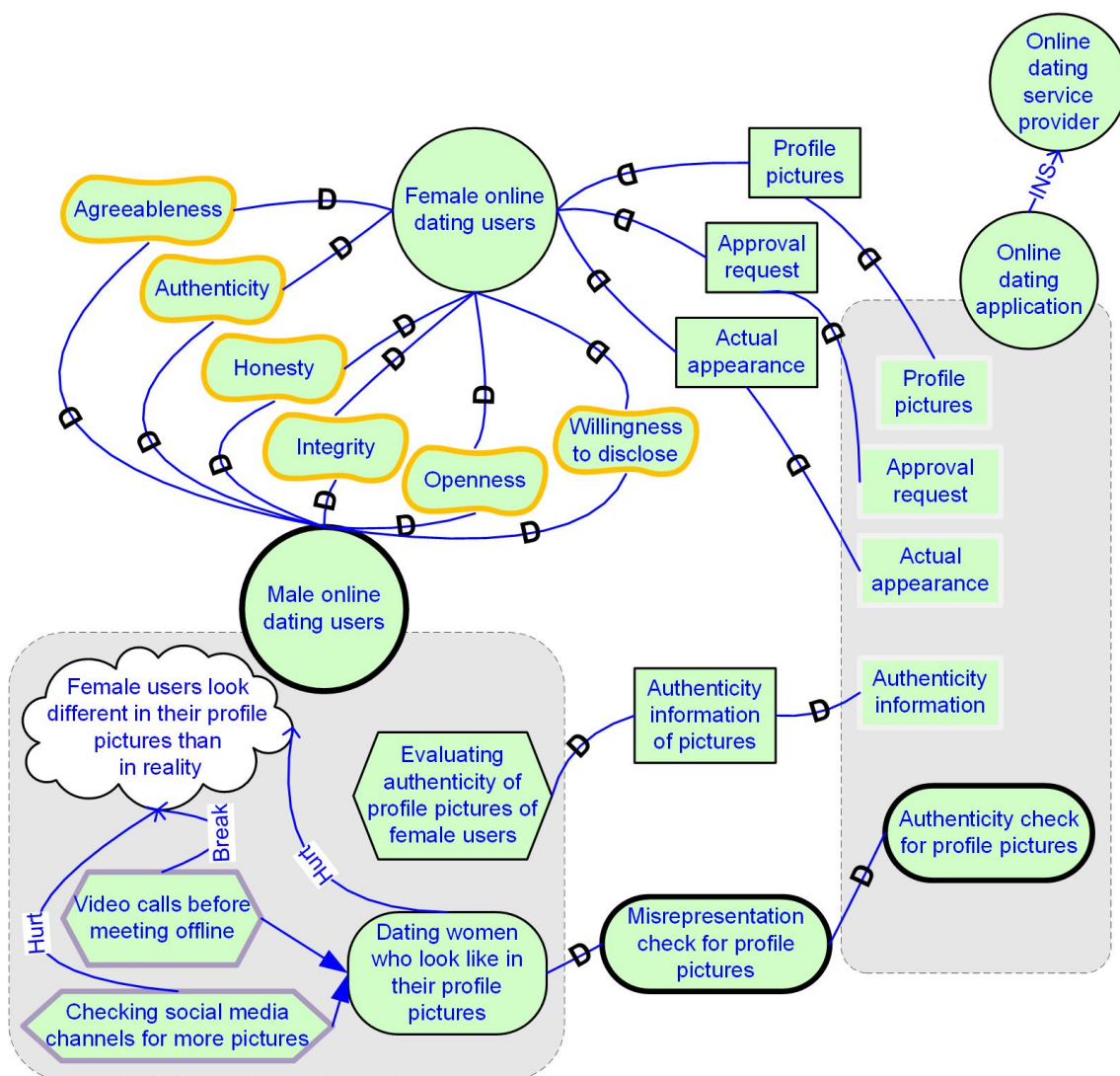


Figure 13.3: Goal model for misrepresentation concern. This model only contains the application with its intentional elements in dependence on the other actors. The full insights of the application are given in Figure 13.4

After modelling the concern and related elements in the actor boundary of the male users, the dependencies towards female online dating users are modelled. Male users are dependent on female users to possess the trustworthiness facets identified in the previous chapter. Then, the trust concern is unlikely to happen. On these grounds, dependencies from the male to the female users are drawn with the trustworthiness facets *agreeableness*, *authenticity*, *honesty*, *integrity*, *openness* and *willingness to disclose*. To assess these facets, male users need the application to reflect them through software features.

Trust Concern	User Goal	Trustworthiness Goal
Users do not take the truth too seriously	Recognising fib	Misrepresentation check
Female users fib about their weight	Dating women who have the weight they stated online	Weight check
Female users look different in their profile pictures than in reality	Dating women who look like in their profile pictures	Misrepresentation check for profile pictures
As a dating option, I do not want to be perceived as worse than other male users	Presenting myself the best way possible while still being myself	Authenticity check

Table 13.15: Expressions of the misrepresentation concern of male online dating users, possible user goals, and potential trustworthiness goals

Focusing now on the online dating application, male users are dependent on the application to achieve their user goals. This is modelled by the goal dependency *misrepresentation check for profile pictures*. It is a dependency from male users to the application that results in the application’s trustworthiness goal *Authenticity check for profile pictures*. By checking whether individuals look authentic in their pictures to their actual appearance, users can check whether other users have misrepresented their pictures.

Figure 13.4 illustrates the complete actor boundary of the application on page 204. The trustworthiness goal can be realized by the three trustworthiness requirements *compare users’ **actual appearance** with **profile pictures***, *display authenticity check*, and *inform users about the procedure of the algorithm check*. Actual appearance and profile pictures are presented in bold because these are necessary elements that need to be included in the model. The trustworthiness requirement for comparing the actual appearance with profile pictures is decomposed into the sub-requirements *ask users for profile pictures* and *ask users for authenticity check*. Both requirements are decomposed into the user trustworthiness facets *agreeableness*, *integrity*, *openness*, and *willingness to disclose*. If users comply with the sub-requirements, these are the facets they show to the system and via the system to the other users.

The sub-requirement that asks users for profile pictures is further decomposed into the feature element *profile pictures*. The application is depended on the users to receive the profile pictures, which are modelled by the resource dependency *profile pictures*. Although the authenticity check is useful and applicable for both male and

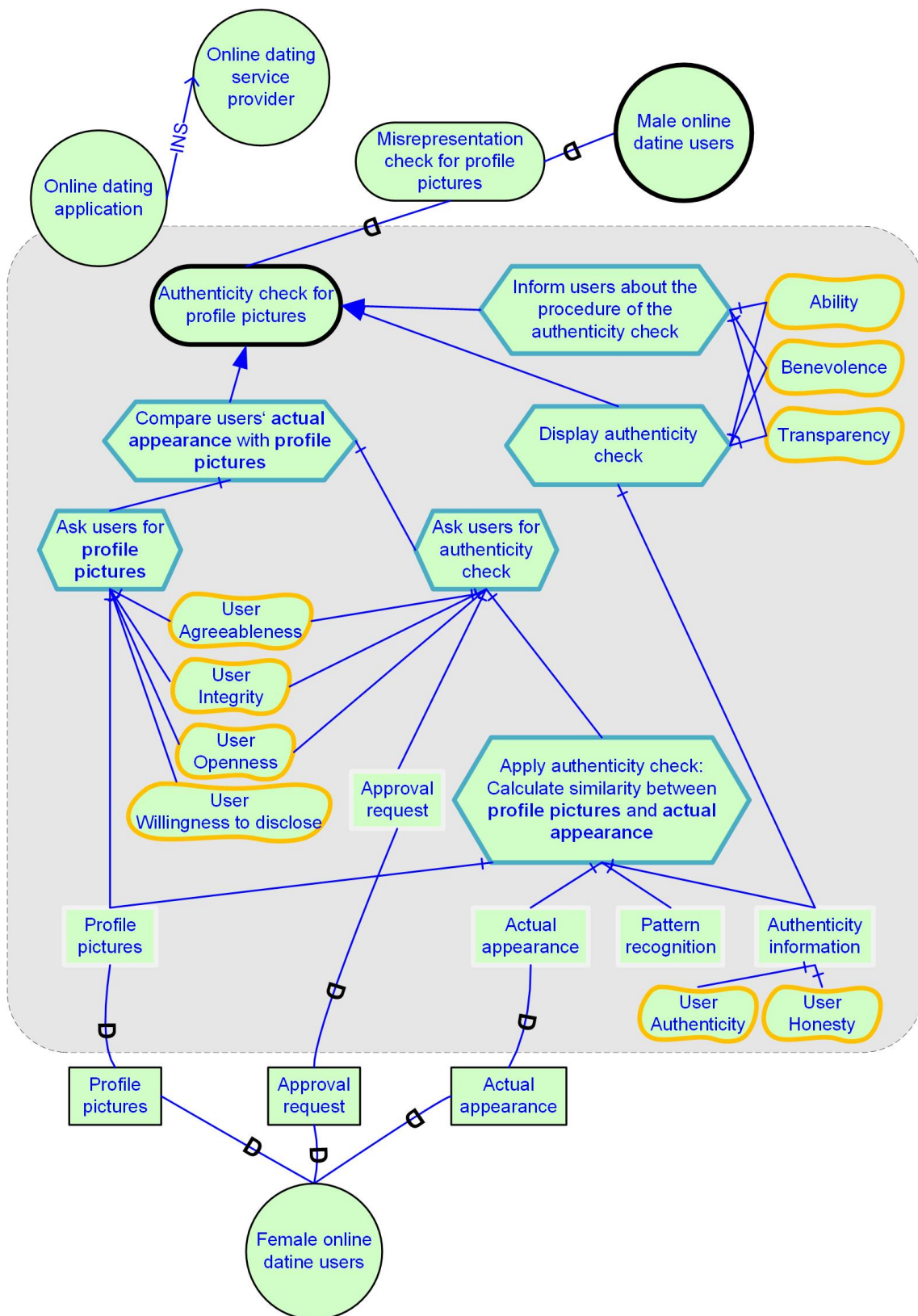


Figure 13.4: This is the complete application with its intentional elements, which is part of the goal model in Figure 13.3

female users, the goal model presents the dependency link only to female users to emphasize the misrepresentation concern of male users and keep the goal model readable.

The other sub-requirement for asking users to conduct the authenticity check is in addition to the facets decomposed into the feature element *approval request* and the sub-requirement *apply authenticity check*. The approval request is a feature element by which users are asked whether they want to participate in the authenticity check. Therefore, the application is dependent on an answer from the user, which is modelled by the resources dependency *approval request* from the application to the female online dating users. The sub-requirement for applying the authenticity check means to *calculate [the] similarity between **profile pictures** and [the] **actual appearance***. For that reason, the sub-requirement is decomposed into the feature elements *profile pictures*, *actual appearance*, *pattern recognition*, and *authenticity information*. The profile pictures and the actual appearance of the users are needed for the similarity calculation. The model has not yet defined the software feature for eliciting the actual appearance. This will be specified in Chapter 13.4. For similarity calculation, pattern recognition shall be used. Pattern recognition is realised by algorithms that can be related to data analysis, information retrieval, image analysis, computer graphics, or machine learning [67]. It automatically recognized patterns in data. It can be used on images for authentication purposes [67]. The pattern recognition of the profile pictures and the actual appearance of users shall result in authenticity information. Based on the authenticity information, the user trustworthiness facets *authenticity* and *honesty* can be assessed (decomposition links).

The feature element authenticity information is additionally a decomposition of the trustworthiness requirement *display authenticity check*. By displaying the authenticity information of profile pictures, male users can check whether a misrepresentation is present.

The third trustworthiness requirement is to *inform users about [the] authenticity check*. This means that users shall receive information about the purpose and procedure of the authenticity check. This requirement, in addition to the one for displaying the authenticity check, reflects the trustworthiness facets *ability*, *benevolence*, and *transparency* of the online dating application. Users, who appreciate the application for this requirement, may perceive the application as increasingly

trustworthy.

13.4 Trust-Related Software Features for Online Dating - Feature Models

This chapter presents trust-related software features as a solution approach for the safety concern of female online dating users and the misrepresentation concern of male online dating users. The previously performed TrustSoFt method and i^* goal models from Chapters 13.1, 13.2, and 13.3 serve as input for specifying tailored trust-related software features. As a methodological approach, the first step of the method for establishing feature models for trustworthiness assessment is applied. The first step involves feature model creation including feature modelling and the facet allocation process consisting of the allocation and propagation phase. The other steps of the method for establishing feature models for trustworthiness assessment - validation and configuration - are disregarded for this example. For validation, a user study must be conducted, which goes beyond the scope of this example. Concerning configuration, feature models must present multiple solution approaches as is the case when developing an application. In that case, software product lines can be configured. However, this example is limited to one trust-related software feature to demonstrate feature modelling. Instead, Chapter 11 provides a small configuration example.

In the following, the presented feature models introduce trust-related software features to the trustworthiness goal and requirements visualised in the i^* goal models that address the safety concern of female online dating users and the misrepresentation concern of male online dating users. For female users, the i^* goal model aims for a safety check of users. Regarding male online dating users, the trustworthiness goal is an authenticity check for profile pictures. First, the creation of the feature model for the female safety concern is presented, then the feature model for the misrepresentation concern of the male users. The feature model for the female safety concern is depicted in Figure 13.8 on page 216. The feature model for the male misrepresentation concern is illustrated in Figure 13.12 on page 230.

13.4.1 Feature Model for the Safety Concern of Female Users

Following the method for establishing feature models for trustworthiness assessment, the first step for specifying trust-related software features is preparatory work for understanding the context of the features (Step 0). For that purpose, the goal model in Figures 13.1 and 13.2 on the pages 196 and 198 is considered.

The trust-related software features shall address women’s concern that *male online dating users pose a risk for the female users’ safety due to their physical superiority* (problem). Unwanted incidents may be *physical violence* or *sexual assault* on offline dates (keywords). Therefore, female online dating users would like to have a safety check of users online. The application addresses this user goal by the trustworthiness goal of enabling a safety check of users. Based on the trustworthiness goal, the concept feature *safety check of users* is created (name). For the safety check of users, the basic information of the catalogue for trust-related software features is filled out. It is depicted in Figure 13.5 on page 207.

Basic Information	
Name	Safety check of users
Problem	Men pose a risk to women’s safety due to their physical superiority
Keywords	Safety, physical violence, sexual assault, gender issue
Requirements	Check the safety criterion “promise fulfillment” for date terms, Elaborate on the safety criterion “respectfulness” of users in the chat, Ask male users for full body picture, Display results for safety criteria, ...
Problematic characteristics	Physical strength
Desired characteristics	Empathy, goodwill, respectfulness, promise fulfillment

Figure 13.5: Basic information of the catalogue for trust-related software features for the concept feature “safety check of users”.

According to the goal model in Figure 13.2 on page 198, the safety check for users needs to realise two main trustworthiness requirements: i) *present cues for [the] safety evaluation of each user* and II) *collect data about safety criteria of user[s]*. Their four sub-requirements provide more details for deriving trust-related software features. They are added exemplary to the basic information in Figure 13.5 on page 207 in the Appendix to emphasize their relevance for the derivation of the trust-related software features. For reasons of completeness, usually, all trustworthiness requirements must be added to the basic information. Concerning the problematic characteristics, female users have mentioned the *physical strength* as concerning. Regarding the desired characteristics, it is concluded that female users wish for *empathy, goodwill, respectfulness* and *promise fulfillment*. As mentioned in Chapter 13.2, these characteristics are relevant for users to assess the trustworthiness of other users in the context of safety. Therefore, they are trustworthiness facets, even though physical strength is not included in the overview of trustworthiness facets (see Appendices A on page 307). For this example, the focus lies on the trustworthiness facets physical strength, respectfulness, and promise fulfillment, because they have been simultaneously identified as safety criteria in the goal model.

Based on the four trustworthiness requirements depicted in Figure 13.5, the trust-related software features can be derived that reflect the goal model in Figures 13.1 and 13.2 on the pages 196 and 198. An overview of the trust-related software features for the concept feature “safety check of users” is presented in Figure 13.6.

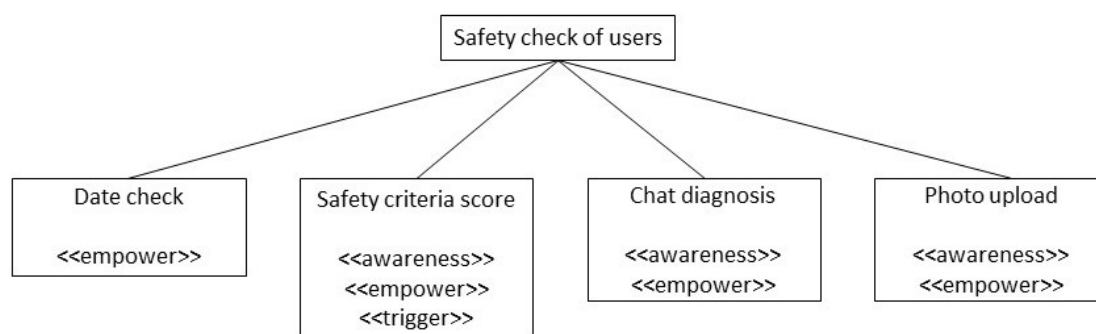


Figure 13.6: Trust-related software features for the concept feature “safety check of users”.

The requirement *check the safety criterion "promise fulfillment" for date terms* leads to the trust-related software feature *date check*. “Date check” aims to support users in setting a date via the online dating application. Users can agree on date

terms such as time and location online. After the date, users are asked for feedback, on whether the other user appeared on the date as promised. Based on a set of user feedback, a “promise fulfillment score” can be calculated. In addition, the feature “date check” can be used for further features in the context of meeting offline. Online dating users have mentioned in the interviews of Paper 9 that their workaround is to tell their friends when and where to meet other online dating users. The application could allow users to share the date terms with friends outside the online dating application. Furthermore, a “panic button” could be included in the trust-related software feature. The “panic button” could be activated by users to call the police to the date location if they need help. Since the feature “date check” leads to the “promise fulfillment score”, which is assessed by users for the trustworthiness of other users, the feature “date check” is determined as an empowerment feature.

For the trustworthiness requirement “elaborate on the safety criterion ‘respectfulness’ of users in the chat”, the trust-related software feature *chat diagnosis* is introduced. Chat diagnosis is a feature that analyses the chat communication of online dating users for respectfulness. It aims to detect rude, insulting, and sexually aggressive text messages. For that purpose, natural language processing can be used for sentiment analysis [169]. The feature shall also transparently inform users about the chat diagnosis and its purpose. Before users send an improper text message, they shall be informed about the bad impression other users might receive from this message. It is assumed that if people are made aware that their chat is evaluated, unwanted incidents can be avoided. Unwanted incidents can be for example sexting, which is also relatable to women’s cyber safety [281]. By chat diagnosis, disrespectful communication can be detected, which can be used to provide a respectfulness score for user profiles or to develop countermeasures to avoid such online behaviour or protect users. In the case of the respectfulness score, users are supported in their online trustworthiness assessment. Furthermore, they are made aware of how they might be perceived by other users and the effects on their perceived trustworthiness. Therefore, the feature chat diagnosis is determined as an empowerment and awareness feature.

Regarding the safety criteria promise fulfillment and respectfulness, their scores can be calculated by the trust-related software features “date check” and chat diagnosis. However, for the trustworthiness assessment, the scores must be presented to the user in the user interface. Therefore, the trust-related software feature *safety*

criteria score is proposed. It addresses the trustworthiness requirement “display results for safety criteria” from the goal model in Figure 13.2. In the user profiles, the calculated score for promise fulfillment and respectfulness shall be included. A possibility for the calculation of the scores is the calculation of the mean value of all occurrences. For both scores, a percentage value could be presented. For promise fulfillment, the percentage value could illustrate how often users kept to the date terms. For respectfulness, the percentage value could show how many cases users interacted respectfully. By rude text messages, the percentage could be reduced. Presenting a safety criteria score for promise fulfillment and respectfulness makes users aware that they are assessed for their trustworthiness. Furthermore, the scores also trigger a trustworthiness assessment. In addition, they empower users in their assessment. Therefore, the trust-related software feature safety criteria score is determined as an awareness, empowerment, and trigger feature.

In terms of the trustworthiness requirement “ask male users for full body pictures”, it is aimed that female users can derive the physical strength of their dating option. Therefore, the trust-related software feature *photo upload* is established. With the option to upload pictures, male users receive the note that female users feel safer if they can estimate the physical strength of the person to meet offline. The feature photo upload enables on the one hand the trustworthiness assessment for female users and on the other hand, makes male users aware that their trustworthiness is being assessed. Therefore, photo upload is determined as an awareness and empowerment feature.

This chapter aims to demonstrate feature models for trustworthiness assessment by a small example. To not go beyond the scope of a small example, feature model creation is limited to the trust-related software feature “date check” for the safety concern of female online dating users. In the following, feature modelling for “date check” is explained. For each feature asset, the asset information of the catalogue for trust-related software features is completed. Not every asset information is explained in depth. Yet, the asset information for each asset can be looked up in Appendix D of this work. After feature modelling, the facet allocation and propagation phase for “date check” are explained. For each step, the validation conditions are checked.

13.4.1.1 Feature Modelling of Date Check

The feature model for the trust-related software feature “date check” is depicted in Figure 13.8 on page 216. The root of the feature model is the concept feature “safety check of users”. It is realised by the trust-related software feature *date check*, which is labelled with «empower» to mark it as an empowerment feature. With the inclusion in the feature model, “date check” receives an entry in the catalogue for asset information. It is depicted in Figure 13.7 on page 211.

Asset Information – Date Check	
Feature Type	<input type="checkbox"/> Awareness <input type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Allocated:</u> Approachability, availability, openness, dynamism <u>Propagated:</u> Promise fulfillment
Trustworthiness facets for application	<u>Allocated:</u> Non-repudiation, (perceived) usefulness <u>Propagated:</u> (perceived) usefulness, confidentiality, ability <u>Optional:</u> Ability, functionality, (perceived) usefulness, safety, social presence
Trustworthiness facets for service provider	<u>Allocated:</u> Concern <u>Optional:</u> Reputation, caring

Figure 13.7: Asset information for the trust-related software feature “date check”.

As mentioned before, the feature type of “date check” is empowerment. “Date

check” is created to support users in the trustworthiness assessment of other users (target group for trustworthiness assessment). Although it enables the trustworthiness assessment in the long run, it is not user accessible yet. The purpose of “date check” is to generate the “promise fulfillment score”. Therefore, “date check” is a prerequisite concerning user accessibility. The “promise fulfillment score” gets user-accessible through the trust-related software feature “safety criteria score” (see Figure 13.6. Concerning the asset category, “date check” aims to gather information for enabling trustworthiness assessment. In addition, it is a feature for initiating a date and getting to know each other better in the offline world. Therefore, “date check” belongs to the asset categories information and interaction. In terms of the nudging criteria, “date check” provides an open choice architecture for setting up dates and approving or declining “date invitations”. Furthermore, it considers the motivational state of users to meet offline and to share date information with friends. The trustworthiness facets of “date check” are included in the facet allocation and facet propagation phase.

“Date check” can be further refined in three parts according to the sub-sub-requirements from the goal model in Figure 13.2. The three requirements are 1) to ask users about the date terms, 2) to ask for feedback after the date whether the date terms have been adhered to, and 3) to calculate the safety criterion “promise fulfillment” for meeting offline. To address the first and second requirements, the asset *date page* is introduced. It is a mandatory asset for the trust-related software feature “date check”. “Date page” provides a user interface, where users can ask each other for an offline encounter by proposing a “time” and “location”. The “date page” has the same asset information as “date check”, with the exception that it is user-accessible. Its asset information is illustrated in Appendix D on page 330. Users can get to the “date page” by a button on the match page - that is the page where two users communicate with each other. The button is a mandatory asset. In the feature model, it could be refined by an asset representing the label of the button, such as “Ask your match for a date”. For the clarity of the feature model, the button is not further specified in detail. The button is a concrete interaction element (asset category). Therefore, it is not categorised in terms of feature type and target group in the asset information. The button involves four nudging criteria, which are 1) the open choice architecture, 2) guiding information considering the label of the button telling users what to do, 3) considering users’ motivational state of setting up a date, and 4) considering user ability to perceive and click the button

on the match page.

To realise the first trustworthiness requirement “ask users about the date terms”, the mandatory feature assets *date request* and *date invitation* are included in the feature model. “Date request” describes that users can ask each other to meet in person. This asset invites interaction and the exchange of information (asset category). By being included in the online dating application, date request proposes to set up dates on the platform instead of other channels. Therefore, “date request” is a behavioural trigger (nudging criteria).

“Date request” has two assets, which are 1) an *input field [for] date terms*, 2) the *button “Ask for [a] date”*. Users can type in the date terms into the input field. Therefore, the asset is further refined into *information date terms female/male user*. This asset picks up the feature elements “information date terms female user” and “information date terms male user” from the goal model in Figure 13.2. It links the information about the date terms, that is *time* and *location*, to the female or male user, depending on who has posed the “date request”. Information about the date terms, like “time” and “location”, are prerequisites for calculating the “promise fulfillment score” (user accessibility). Linking the information to the single users is relevant for the asset *panic button*, which is explained later below.

“Date request” is required for the feature asset *date invitation*. As soon as a user has asked another user for a date by “date request”, the other user receives a notification - the “date invitation”. The “date invitation” includes the *time* and *location* that the other user has proposed in the “date request” (require-links). The “date invitation” further includes a *button [to] accept [the] date* and a *button [to] decline [the] date*. When the user accepts the date, the “time” and “location” of the date are related to the information about the date terms for that user (require-links). The accept button is a user-accessible prerequisite for calculating the displaying of the “promise fulfillment score”. In contrast, the “decline button” is user-accessible but not a prerequisite.

The second trustworthiness requirement, “ask for feedback after the date whether the date terms have been adhered to”, is addressed by the feature asset *feedback to the date*. Users are able to provide feedback on the date on the “date page” after the given time that has been stated in the “date request”. To receive feedback from users, *questions about the date* are posed on the “date page”. This asset can further

be refined by the specific questions asked, which is not done here to keep the example small. Examples of feedback questions could be “Did the other user appear on the date” or “Have the profile information of your date been correct?”. For creating questions, the user’s ability to answer them must be considered. By posing the questions and implementing *input field[s] for [the] answer[s]*, behavioural triggers are used. The answers are the *date term feedback [from the] female user* and the *date term feedback [from the] male user*. These two assets pick up the homonymous feature elements from the goal model in Figure 13.2.

The last feature asset of the “date page” is the optional *panic button*. The “panic button” is linked to the information about the date terms of the users. Users can press the “panic button” on the “date page” during the date if they feel in danger. The “panic button” activates a *police call*. The police receive the “time” and “location” of the date.

For the third trustworthiness requirement “calculate the safety criterion ‘promise fulfillment’”, the feature asset *algorithm promise fulfillment score* is included in the feature model. Its child asset is the *promise fulfillment score*. The algorithm uses the “date term feedback” (require-links). The “promise fulfillment score” is a prerequisite for users’ trustworthiness assessment.

Last but not least, the software feature “date check” has an optional feature asset called *date term share*. As female users have stated the workaround to share the date terms with their friends before meeting other online dating users, this asset addresses exactly the workaround. Date term share is activated by clicking the *share button*. By clicking the button, the *sharing algorithm* uses the *contact information [of the user’s] friends* to send them the information about the date terms (require-links). As the “share button” is representing the date term share algorithm, it has many characteristics of the asset information. It is user-accessible and refers to an algorithm, provides information, is an interaction element, and relies on design principles. Furthermore, it has an open-choice architecture, because users can decide whether to use it or not. Furthermore, it provides a solution approach for the unfavourable situation of meeting strangers without the knowledge of friends. In addition, the “share button” considers people’s motivational state of performing the workaround and provides a behavioural trigger by providing users the option to click it.

Now that the feature model has been established, it is checked with the validation criteria. The validation conditions are depicted below with either a check mark or a cross. In some cases, the validation conditions are discussed below the respective bullet point.

Check of the Validation conditions for Feature Modelling

- ✓ The feature model contains a concept feature that poses a high-level solution to the trust concern of users.
- ✓ The concept feature has an entry in the basic information of the catalogue for trust-related software features.
- ✓ The feature model contains at least one awareness feature, one trigger feature, and one empowerment feature.

However, this is a small example. Even though the trust-related software feature “safety criteria score” is all three feature types, further features should be added that focus on either one of the feature types.

- ✓ Each asset in the feature model has an entry in the asset information of the catalogue of trust-related software features.
- ✓ The feature elements of the associated TrustSoFt goal models are included as assets in the feature model.
- ✓ The proposal for trust-related software features from TrustSoFt is addressed by assets in the feature model.

The trustworthiness requirements from the goal model in Figure 13.2 are addressed.

- × Only assets are part of the feature model that either i) hold a trustworthiness facet of TrustSoFt or ii) are necessary for establishing the trust-related software feature to address or reflect the trustworthiness facet of TrustSoFt.

The assets “date terms share” and “panic button” address the safety concern of female online dating users but are not related to the trustworthiness assessment. Yet, they are valuable assets that enrich the feature “date check”. Although this validation check is important to keep the focus of the feature

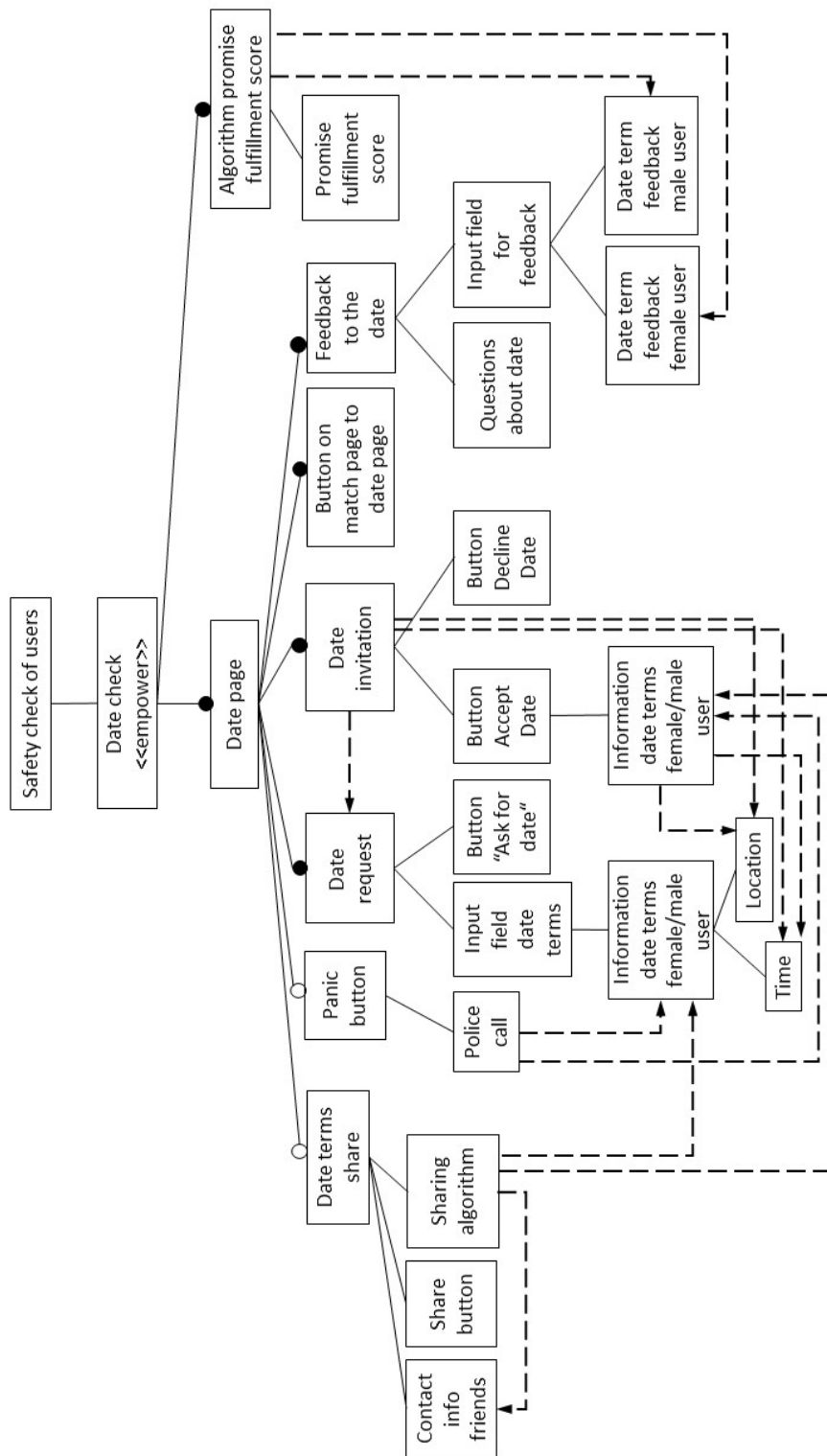


Figure 13.8: Feature model for the trust-related software feature “date check” for the safety concern of female online dating users.

models on the trustworthiness assessment, it should be treated as a soft validation condition. This means that even though it is not true for this case, the feature model is still correctly established. To be correct for the future, the asset information could contain the characteristic that an asset is irrelevant to the trustworthiness assessment of the targeted group. However, such assets as date terms share or the “panic button” can positively contribute to the trustworthiness of the application. It depends on the way they are constructed. They can be trust-related when they provide additional beneficial value for users, exceeding the basic functionality, that users associate with the trustworthiness facets of the software application.

- ✓ One user-accessible asset is mandatory.

13.4.1.2 Facet Allocation for Date Check

In accordance with the method for establishing feature models for trustworthiness assessment, the facet allocation phase begins with the trust-related software feature and continues down the tree involving each asset. For the facet allocation phase, each asset is compared and related to the overview of trustworthiness facets in the Appendices A, B, and C. When a facet is identified for an asset, the facet is included in the feature model and the asset information of the catalogue for the safety check of users. Concerning the feature model, including the trustworthiness facet in the model would exceed the DIN A4 format of this dissertation. Therefore, in the following, the trustworthiness facets of each asset are briefly stated and explained. The documentation is carried out via the asset information of the catalogue in Appendix D.

For the empowerment feature “date check”, the following trustworthiness facets for users can be identified: *approachability*, *availability*, *openness* and *dynamism*. By using “date check”, a user approaches another person and makes oneself vulnerable by asking for a date. Furthermore, the person shows that he/she is available. Furthermore, the person is open to further interaction and shows dynamism in actively approaching another user for a social event. Concerning the trustworthiness facts of technology, “date check” aims to demonstrate the (*perceived*) *usefulness* of the online dating application. In terms of the service provider, “date check” might

demonstrate the provider’s *concern* that people meet in the offline world.

The next asset in the feature model is the “date terms share” asset. As it enables the communication between a user and friends, it is not relatable to the trustworthiness facets of the other user. The same is true for its refined assets “contact information friends”, “share button”, and “sharing algorithm”. Concerning the online dating application, “date terms share” targets the trustworthiness facets *ability*, *functionality*, *(perceived) usefulness*, *safety* and *social presence*. For the refined child assets, the “sharing algorithm” may be associated with the ability of the application. Last but not least, the service provider seems unrelatable concerning their trustworthiness towards the asset “date terms share” and its child assets.

The trust-related software feature “date check” is performed on the “date page”. “Date page” is neutral concerning how other users are perceived. For the online dating application and the service provider, their perception of the users depends on the design of the “date page”. Software engineers could now choose trustworthiness facets to declare their intent of how the application or service provider should be perceived via the “date page”. As this is not the focus of this work, there are no trustworthiness facets linked to “date page” for this example.

The “panic button” is also again detached from the perception of other users. Instead, it can be perceived as an expression of the application to take care of people’s *safety*. In addition, depending on other factors outside the application, such as marketing, the “panic button” could impact the *reputation* of the application to support people’s safety. The impact on the *reputation* is also present for the service provider. The service provider could also be perceived as increasingly trustworthy by the “panic button”, because people believe the provider to be *caring*. Concerning the child asset “police call”, users are highly dependent on the algorithm to send the information of the date terms to the police. Therefore, its *performance* of the online dating application is highly important.

For “date request”, the trustworthiness facets are the same as for “date check” with the same explanations. For the child assets “input field date terms” and “button ‘ask for [a] date’”, no trustworthiness facets of the three parties are relatable. Concerning the technology facets, a given functioning of the application regarding its functionality is considered standard [240] so that the facets are irrelevant for such concrete design and interaction elements. The same is true for the actual

“information date terms female/male user” that is “time” and “location”.

For “date invitation”, the trustworthiness facets are the same as for “date request”. Unlike the user facets from the “date request”, where users prove the facets by creating a request, “date invitation” reflects the facets for users’ trustworthiness assessment. When users use the accept button for accepting the date, they demonstrate the user facets *availability* and *openness*. Concerning the “decline button” and the information about the date terms, no trustworthiness facets are relatable. These assets are at such a concrete design level that no trustworthiness can be derived from them. The same is true for the “button on [the] match page to [the] date page”. Regarding facets for the service provider, these assets do not refer to organisational or reputational characteristics, hence no organisational facets can be attributed.

The next asset is “feedback to the date”. This asset cannot be used to assess the trustworthiness of the other user, nor does it check the trustworthiness of the user providing feedback. Therefore, it is not relatable to user trustworthiness facets. Regarding the technology trustworthiness facets, the online dating application promises *confidentiality* for the provided data when using “feedback to the date”. The other user will not receive the feedback itself but only the resulting “promise fulfillment score”. Concerning the service provider, this asset does not show any of its trustworthiness facets, because it is not linked to any organisational processes. The child assets of “feedback to the date” are again on a concrete design level. Therefore, they are not relatable to any trustworthiness facets.

The last child asset of the trust-related software feature is the “algorithm [for the] promise fulfillment score”. As it runs in the background, it does not impact the trustworthiness assessment of the users directly. However, when being an underlying element of a user-accessible feature, the algorithm is related to the *ability* of the online dating application. Concerning its child asset “promise fulfillment score”, if the asset is presented in the user interface, it is associated with the user trustworthiness facet *promise fulfillment*.

Now that the trustworthiness facets have been allocated to each asset of the feature model, the validation conditions are checked.

Check of the Validation conditions for the Allocation Phase

- × Each trustworthiness facet that has been identified by TrustSoFt must be related to at least one user-accessible, mandatory asset.

In this feature model excerpt, the trustworthiness facet to be addressed is “promise fulfillment”. It belongs to the asset “promise fulfillment score” that is not user-accessible. However, as this is only an excerpt, the “promise fulfillment score” would be required by the trust-related software feature “safety criteria score” from Figure 13.6. This trust-related software feature would make the “promise fulfillment score” user-accessible in form of a percentage value in the user profile. In this case, the validation condition is satisfied.

- If there is only one user-accessible, mandatory asset, it must be able to hold all trustworthiness facets identified by TrustSoFt.

This validation condition is not applicable to this feature model.

- ✓ Each trustworthiness facet that is linked to an asset is documented in the asset’s asset information of the catalogue for trust-related software features.

Every trustworthiness facet has been included in the asset information presented in Appendix D.

13.4.1.3 Facet Propagation for Date Check

After the trustworthiness facets have been allocated, they are now propagated in accordance with the inheritance principle. The propagated trustworthiness facets are included in each respective asset information of the catalogue of “safety check” for users in Appendix D.

The propagation starts with the child assets of the asset “date terms share” on the left of the feature model except in Figure 13.8. “Contact info friends” and “share button” do not have any trustworthiness facets. “Sharing algorithm” is associated with the “ability” of the online dating application. The parent asset “date terms share” is also associated with the “ability” of the online dating application. Therefore, for this asset set, no propagation can be performed. The propagation is now performed at a higher level. “Date terms share” passes the facets “ability”, “functionality”, “(perceived) usefulness”, “safety”, and “social presence” of the online dating application onto the asset “date page”. As “date terms share” is an

optional asset, the propagated trustworthiness facets are highlighted as optional. “Date page” also passes the optional facets on to “date check”.

The next leaf of the feature model is “police call”. It is associated with the “performance” of the online dating application. It is propagated to the “panic button”. “Panic button” is an optional asset. Therefore, it passes the facets from “police call” on to “date page” as optional facets. In addition, it also propagates its own facets as optional to “date page”, which are “safety” and “reputation” as technological facets and “reputation” and “caring” as organisational facets. “Date page” passes the propagated facets onto “date check” as optional facets.

The next assets for propagation are “time” and “location”. They do not have any facets so the propagation continues with “information date terms female/male user”. This asset neither has trustworthiness facets. The next higher asset level involves “input field date terms” and “button ‘Ask for date’”. Again, there are no allocated facets. Continuing with “date request”, this asset is related to the user facets “approachability”, “availability”, “openness” and “dynamism” and the technological facet “(perceived) usefulness”. They are all passed on to “date page”. “Date page” would propagate them to “date check”, but “date check” already has them all allocated.

The next leaf for propagation is “information date terms female/male users”. Like before, no trustworthiness facets are associated with this asset. The same is for the “button decline date”. For the “button accept date”, the user trustworthiness facets are “availability” and “openness”. Yet, “date invitation” is already associated with these facets. In addition, it also is related to the user facets “approachability”, “dynamism” and the technological “(perceived) usefulness”. “Date page” already has these facets propagated as it is consequently the case for “date check”.

For the “button on [the] match page to [the] date page”, no related trustworthiness facets are documented. Therefore, this asset does not propagate any facets.

The next assets for propagation are “date term feedback female user” and “date term feedback male user”. Both are not related to any trustworthiness facets. Their parent asset “input field for feedback” and the asset “questions about [the] date” do neither convey any trustworthiness facets. Yet, their parent asset “feedback to the date” is associated with the “confidentiality” of the online dating application. “Confidentiality” is propagated to “date page” and “date check”.

Last but not least comes the asset “promise fulfillment score”. Its use facet “promise fulfillment” is propagated to the parent asset “algorithm promise fulfillment score”. From the algorithm, “promise fulfillment” and the technological facet “ability” are propagated to “date check”.

In the end, the list of trustworthiness facets for the concept feature is created. It is depicted in Figure 13.9. For this example, the trustworthiness facets comply with the ones of the trust-related software feature “date check”, because the example only considers the feature model excerpt.

List of trustworthiness facet for the concept feature “Date Check”

Trustworthiness facets – user

- Approachability
- Availability
- Dynamism
- Openness
- Promise fulfillment

Trustworthiness facets – technology

- Ability
- Confidentiality
- Functionality
- Non-repudiation
- (Perceived) usefulness
- Safety
- Social presence

Trustworthiness facets – service provider

- Caring
- Concern
- Reputation

Figure 13.9: List of trustworthiness facets for the concept feature “date check”.

The last step of the propagation phase is to check the validation conditions.

Validation conditions for the Propagation Phase

- ✓ Propagated trustworthiness facets are included in the asset information of the parent asset of the catalogue for trust-related software features.
 - ✓ Optional, propagated trustworthiness facets are emphasised accordingly within the asset information of the catalogue for trust-related software features.
- Propagated trustworthiness facets that stem from a mandatory child asset do not have a line on the left side of their box.

As this example does not illustrate the feature model after the propagation phase, this validation condition cannot be checked.

- ✓ The list of trustworthiness facets for the concept feature contains all trustworthiness facets within the feature model once.

13.4.2 Feature Model for Misrepresentation Concern of Male Users

To specify trust-related software features for the misrepresentation concern of male online dating users, preparatory work for understanding the context is performed first (Step 0, the method for establishing feature models for trustworthiness assessment, Figure 11.1 page 122). For that purpose, the goal model in Figures 13.3 and 13.4 is on pages 202 and 204 is considered.

The trust-related software features to be specified shall address the concern of male online dating users that *female users look different in their profile pictures than in reality* (problem). When discovering the discrepancy between the pictures with the actual appearance, male online dating users associate this circumstance with either willful *deception* or tolerable *impression management* that still oftentimes leads to disappointment (keywords in basic information). On this basis, the user goal of male online dating users is to date women who look like in their profile pictures. Thus, the trustworthiness goal of the application is an *authenticity check for profile pictures*. To realise the trustworthiness goal, the concept feature in the feature models is the *authenticity check of users*. For the authenticity check of users, the basic information of the catalogue for trust-related software features is filled out. It is depicted in Figure 13.10.

According to the goal model 13.4, the authenticity check for users needs to realise three main trustworthiness requirements: i) *inform users about the procedure of the authenticity check*, ii) *compare users' actual appearance with [the] profile pictures*, and iii) *display [the] authenticity check*. The second requirement further has three sub-requirements that are a) *ask users for profile pictures*, b) *ask users for authenticity check*, and b.a) *apply authenticity check: calculate [the] similarity between [the] profile pictures and [the] actual appearance*. For the misrepresentation concern, a few trustworthiness facets have been identified in the goal model in Figure 13.4. These are *agreeableness*, *integrity*, *openness*, the *willingness to disclose* personal information, *authenticity*, and *honesty* for users. For the application, the trustworthiness

Basic Information	
Name	Authenticity check
Problem	Female users look dfferent on their online dating profile pictures than in reality
Keywords	Misrepresentation, deception, impression management, authenticity
Requirements	Inform users about the procedure of the authenticity check, Compare users' actual appearance with profile pictures, Ask users for profile pictures, Ask users for authentictiy check, Apply authenticity check: calculate similarity between profile pictures and actual appearance, Display authenticity check
Problematic characteristics	-
Desired characteristics	User: Agreeableness, Integrity, Openness, Willingness to Disclose, Authenticity, Honesty Application: Ability, Benevolence, Transparency

Figure 13.10: Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”.

facets *ability*, *benevolence* and *transparency* have been identified. The trustworthiness facets are documented for the desired characteristics in the basic information of the catalogue of the authenticity check of users in Figure 13.10.

Based on the six trustworthiness requirements depicted in Figure 13.10, software features can be derived that reflect the goal model in Figures 13.3 and 13.3. The six trustworthiness requirements can be realised by one trust-related software feature that is called *appearance verifier*.

The appearance verifier aims to calculate the similarity of users' appearance and the uploaded online dating profiles and presents the resulting authenticity information to other users. Thereby, it triggers and empowers users' trustworthiness assessment of other users. Moreover, it informs users about the purpose and procedure of the authenticity check and increases awareness of the misrepresentation of one's appearance. On these grounds, the appearance verifier is an empowerment, trigger, and awareness feature.

The goal models in Figures 13.3 and 13.4 provide a small, exemplary solution

approach that is picked up in this chapter for demonstrating the feature models. There are more solution approaches possible in the form of trust-related software features. As an example, videos could be integrated into profiles instead of profile pictures as another source that might be harder to misrepresent for the common user. Yet, for the subsequent chapter, the focus is on the exemplary trust-related software feature “appearance verifier”.

13.4.2.1 Feature Modelling of Appearance Verifier

The feature model for the concept feature “authenticity check of users” and the trust-related software feature “appearance verifier” is depicted in Figure 13.12 on page 230. As the trust-related software feature “appearance verifier” is an empowerment, trigger, and awareness feature, it is labelled with «empower», «trigger», and «awareness» in the feature model. With the inclusion in the feature model, the “appearance verifier” receives an entry in the catalogue for asset information. It is depicted in Figure 13.11.

As mentioned before, the feature type of “appearance verifier” is *empowerment*, *trigger*, and *awareness*. “Appearance verifier” is created to support users in the trustworthiness assessment of other *users* (target group for trustworthiness assessment). It provides users with information for the trustworthiness assessment and is, thus, user-accessible. Concerning the asset category, “appearance verifier” is based on an *algorithm* and provides *information* about users’ authenticity in regards to their presented appearance online. Regarding the nudging criteria, “appearance verifier” shows *guiding information* for the trustworthiness assessment and the decision to interact with other users. It is a *behavioural trigger* for performing the trustworthiness assessment. The trustworthiness facets of the appearance verifier are determined in the facet allocation and facet propagation phase.

“Appearance verifier” is refined into three assets based on the three trustworthiness requirements from the goal model in Figure 13.4. The three requirements are 1) to compare users’ actual appearance with the profile pictures, 2) to display the authenticity check, and 3) to inform users about the procedure of the authenticity check.

The first trustworthiness requirement is again divided into sub-requirements. The

Asset Information – Appearance verifier	
Feature Type	<input checked="" type="checkbox"/> Awareness <input checked="" type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	
Trustworthiness facets for application	
Trustworthiness facets for service provider	

Figure 13.11: Asset information for the trust-related software feature “appearance verifier”.

first sub-requirement is to ask users for profile pictures. As known from already existing online dating applications like Tinder or Bumble, users can upload profile pictures on the *profile setting page*. On the “profile setting page”, users can find a picture upload functionality consisting of a *picture upload algorithm* and *picture upload button*.

The asset information for the “profile setting page” is as follows. The “profile setting page” is *user-accessible* and a *prerequisite* for supporting users in their trustworthiness assessment. Regarding its asset category, it involves *algorithms*, such as the “picture upload algorithm”, it provides *information* to the profile settings, such as the optional appearance verifier, it enables users *interaction* with the application, and it involves *design*. Furthermore, it provides an *open choice architecture* by which users can create their online dating profile at their own will. The “profile setting page” includes guiding information for the appearance verifier. Moreover, it considers users’ *motivational state* and *ability* for creating an online representation of themselves and includes behavioural triggers in the form of interaction elements

so that users can create an online dating profile.

The “picture upload algorithm” is a *prerequisite* for the online trustworthiness assessment. Its asset category is an *algorithm*. For the “picture upload button”, user accessibility is given and it is a *prerequisite* for the actual assessment of user’s appearance authenticity. Its asset category is *interaction*. With a button label, such as “upload picture”, it also provides *information* about its functionality. Concerning the nudging criteria, the “picture upload button” is a *behavioural trigger* that considers *user ability* and their *motivational state* for uploading pictures.

The second sub-requirement is to ask users for confirmation to perform the authenticity check. This could be done again on the “profile setting page” underneath a visualisation of the uploaded profile pictures. By a *toggle switch* called “authenticity verifier”, users can decide whether to allow and include the verifier to their profile by turning it on or off. The “toggle switch” is an *interaction* element that is *user-accessible* and a *prerequisite* for the online trustworthiness assessment. Another option is that every time, a user uploads a picture, a *confirmation window* pops up, asking whether the profile picture can be used for the authenticity verifier. Users can answer by a *approve button* or a *decline button*. The “confirmation window” is an *information*, *interaction*, and *design* element. It is *user-accessible* and a *prerequisite* for the online trustworthiness assessment. Either the “toggle switch” or the “confirmation window” must be included in the application. This optionality is visualised by a XOR-alternative link between these two feature assets.

The third sub-requirement is the realisation of the authenticity check, which is done by a *pattern recognition algorithm* to calculate the similarity between the uploaded *profile pictures* and the *actual appearance*. As the “actual appearance” is an asset outside the online sphere, its associated asset information is so far empty. Ways to collect the “actual appearance” for the application are for example *real-time video* recording or *real-time photo* within the application that does not allow any filters (OR-link). Both are *prerequisites* so that the “pattern recognition algorithm” can determine the *authenticity information* of users. By including the assets in the feature model, the feature elements of the goal model in Figure 13.4 are respected.

The “authenticity information” is relevant for targeting *users* concerning the trustworthiness assessment. It is a *prerequisite* to enable the online trustworthiness assessment and can be classified as *information*. The “authenticity information”

is *required* to realise the trustworthiness requirement “display authenticity check”, which is realised by an *authenticity score*. The “authenticity score” is the percentage of the similarity between the “profile pictures” and the “actual appearance” of users. Different from the “authenticity information”, the “authenticity score” is user-accessible and enables and triggers the trustworthiness assessment. Moreover, it makes users aware of the trustworthiness of other users and potentially of the trustworthiness assessment itself.

Last but not least, the trustworthiness requirement for informing users about the procedure of the authenticity check is addressed in the feature model. Informing users about the appearance verifier is relevant when they are confronted with it. This is the case for the “toggle switch” and the “confirmation window”, when confirming to participate in the appearance verifier, and for the “authenticity score”. In these cases, *information about [the] appearance verifier* is displayed that explains the reason and procedure of the feature. For the “toggle switch” and “authenticity score”, an *information icon* could be depicted to their right. By clicking on the “information icon”, the information appears. For reasons of clarity of the feature model, the “information about [the] appearance verifier” and the “information icon” are modelled twice in the feature model.

The complete sets of asset information of all feature assets are depicted in Appendix D.

Now that the feature model has been established, it is checked with the validation criteria. The validation conditions are depicted below with either a check mark or a cross. In some cases, the validation conditions are discussed below the respective bullet point.

Check of the Validation conditions for Feature Modelling

- ✓ The feature model contains a concept feature that poses a high-level solution to the trust concern of users.
- ✓ The concept feature has an entry in the basic information of the catalogue for trust-related software features.
- ✓ The feature model contains at least one awareness feature, one trigger feature, and one empowerment feature.

The “appearance verifier” is only one example of the authenticity check of users. Even though it covers all three feature types, more software features can be added that specialise in one of the feature types.

- ✓ Each asset in the feature model has an entry in the asset information of the catalogue of trust-related software features.
- ✓ The feature elements of the associated TrustSoFt goal models are included as assets in the feature model.
- ✓ The proposal for trust-related software features from TrustSoFt is addressed by assets in the feature model.

The trustworthiness requirements from the goal model in Figure 13.4 are addressed.

- ✓ Only assets are part of the feature model that either i) hold a trustworthiness facet of TrustSoFt or ii) are necessary for establishing the trust-related software feature to address or reflect the trustworthiness facet of TrustSoFt.
- ✓ One user-accessible asset is mandatory.

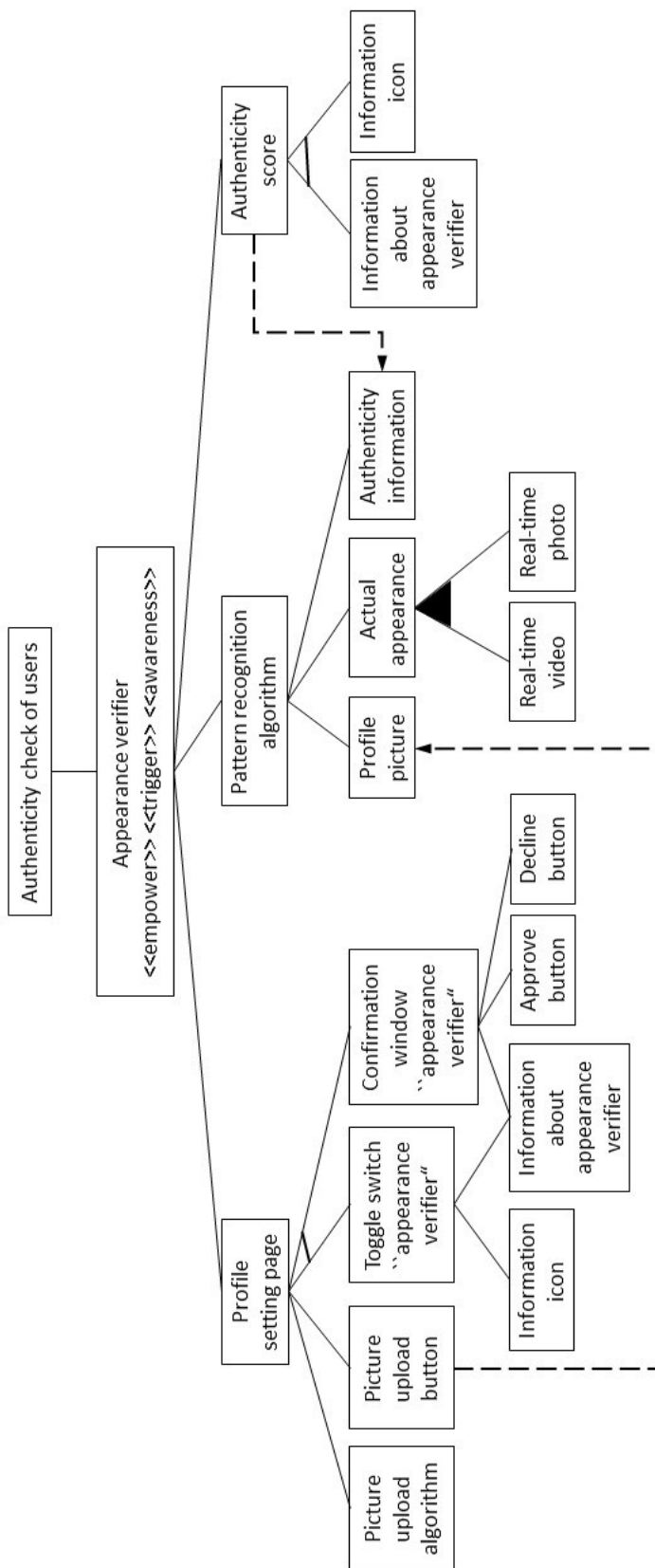


Figure 13.12: Feature model for the trust-related software feature “appearance verifier” for the misrepresentation concern of male online dating users.

13.4.2.2 Facet Allocation for the Appearance Verifier

In accordance with the method for establishing feature models for trustworthiness assessment, the facet allocation phase begins with the trust-related software feature and continues down the tree. Each asset is compared and related to the overview of trustworthiness facets in the Appendices A, B, and C. When a facet is identified as relevant for an asset, the facet is included in the feature model and the asset information of the catalogue for the authenticity verifier of users (see Appendix D). Concerning the feature model, including the trustworthiness facets in the model would exceed the DIN A4 format of this dissertation. Therefore, in the following, the trustworthiness facets of each asset are briefly stated and documented in the asset information of the catalogue in the Appendix.

For the trust-related software feature “appearance verifier”, the following trustworthiness facets for users can be identified: *attractiveness*, *honesty*, *credibility*, *truthfulness*, *authenticity*, *willingness to disclose*, *integrity*, *ethicality*, and *promise fulfillment*. Male users have stated being disappointed if female users look better in their profile picture than in reality. Therefore, the appearance verifier would help male users assess the attractiveness of female users. Moreover, the authenticity verifier depicts whether the user has been honest, credible, truthful, and authentic with the uploaded pictures. With a high “authenticity score”, users can derive that the other user was willing to disclose his/her true self in terms of appearance. In addition, authentic profile pictures show that the user has integrity in himself/herself. The same is true for his/her ethicality. Last but not least, by uploading authentic profile pictures, a user can reflect his/her promise fulfillment to be the same person when meeting other users offline. Regarding the trustworthiness facets of the application, users can assess by the appearance verifier, that the application has *ability*, provides *information quality*, and is *useful*. If the “appearance verifier” is a unique software feature, it could positively impact the *reputation* of the service provider. However, as the feature itself is detached from any organisational structures, trustworthiness facets for organisations are not considered for this example.

For the “profile setting page”, no trustworthiness facets for users are relatable, because the “profile setting page” does not reflect other users. The trustworthiness facets of the application that can be related to the “profile setting page” are highly dependent on what is depicted on the page. Therefore, the facet attribution phase

is skipped. Instead, it waits for the facet propagation phase to link relevant trustworthiness facets to the “profile setting page” depending on its integrated assets.

Regarding the “picture upload algorithm”, it can be related to the *ability* of the online dating application. In addition, the algorithm can be associated with *situational normality* for online dating applications. It is common in online dating to be able to upload profile pictures.

The “picture upload button”, “toggle switch”, and the “approve button” from the “confirmation window” are the interaction elements by which users can demonstrate their *integrity* with the features and norms of the online dating application, their *agreeableness* to participate in the “appearance verifier”, their *openness* about their person, and the *willingness to disclose* personal information. In terms of the trustworthiness facets of the online dating application, the three feature assets as well as the “information icon” of the “toggle switch” and the “decline button” of the “confirmation window”, are elementary interaction elements that are not directly linked with the trustworthiness of the online dating application. The only exception is if the interaction elements do not function as expected so that a negative effect on the trustworthiness of the application occurs. However, regarding the trustworthiness facets. Neither those for the application nor the service provider are related to the five interaction elements.

Concerning the “confirmation window” for the “appearance verifier” and the associated “information about [the] appearance verifier”, the wording and content of the message may be linked to the trustworthiness facets of the application or the service provider. At that point, the developers must determine what facet they want to target for the design of the two assets. As this is a subjective decision, no facets are related to these assets for this example.

The next asset for facet allocation is the “pattern recognition algorithm”. Like most algorithms, it reflects the *ability* of the application. Moreover, it is a technology that aims to show its *functionality* to its users. For “profile picture” and “actual appearance”, a large set of trustworthiness facets for users can be derived depending on the picture and person. As this is highly person-related, the asset information for these two assets receives the entry “...” for the trustworthiness facets of users. For the “real-time video” and “real-time photo”, users may perceive the necessity and, thus, the *usefulness* of the online dating application for that feature asset. However,

the feature assets may have additional effects on users that may indirectly impact the trustworthiness of the application. As users need to invest time and may be concerned about their privacy when interacting with the real-time video and photo, they may be annoyed by this feature asset. Being annoyed by software features reduces the trustworthiness of technology [309].

The “authenticity information” is a resulting value from the “pattern recognition algorithm” that is invisible to the user. Therefore, it is not associated with any kind of trustworthiness facet. However, it is transformed into a user-accessible “authenticity score”. The “authenticity score” reflects a user’s *attractiveness, honesty, credibility, truthfulness, authenticity, willingness to disclose, integrity, ethicality, and promise fulfillment*. The “authenticity score” is the core of the “appearance verifier” and shares the same reasons why the trustworthiness facets for the users are relevant. Furthermore, the “authenticity score” also shares the same trustworthiness facets of the application as the “authenticity verifier”. These are the *ability, information quality, and usefulness* of the online dating application.

Now that the trustworthiness facets have been allocated to each asset of the feature model, the validation conditions are checked.

Check of the Validation conditions for the Allocation Phase

- ✓ Each trustworthiness facet that has been identified by TrustSoFt must be related to at least one user-accessible, mandatory asset.
 - If there is only one user-accessible, mandatory asset, it must be able to hold all trustworthiness facets identified by TrustSoFt.

This validation condition is not applicable to this feature model.

- ✓ Each trustworthiness facet that is linked to an asset is documented in the asset’s asset information of the catalogue for trust-related software features.

Every trustworthiness facet has been included in the asset information presented in Appendix D.

13.4.2.3 Facet Propagation for Appearance Verifier

After the trustworthiness facets have been allocated, they are now propagated in accordance with the inheritance principle. The propagated trustworthiness facets are included in each respective asset information of the catalogue of the authenticity check of users in Appendix D.

The propagation starts with the child assets of the asset “profile setting page” on the left of the feature model in Figure 13.12 on page 230. The “profile upload algorithm” propagates the trustworthiness facets *ability* and *situational normality* of the online dating application to the “profile setting page”. The “picture upload button” passes the trustworthiness facets *agreeableness*, *integrity*, *openness*, and *willingness to disclose* of users on the “profile setting page”. The asset “information icon” does not hold any facet and, thus, cannot propagate any to the “toggle switch”. The same is the case for the “information about [the] appearance verifier” and the “decline button”, which as well cannot forward any facets to the “confirmation window”. Yet, the “approve button” passes the user facets *agreeableness*, *integrity*, *openness*, and *willingness to disclose* on to the “confirmation window”. These user trustworthiness facets would be propagated from the “toggle switch” and the “confirmation window” to the “profile setting page” as being optional. However, the “profile setting page” is already associated with these facets, which is why this propagation step is skipped. Last but not least, the “profile setting page” propagates those facets that the “appearance verifier” is not yet related to, which are *agreeableness* as a user facet and *situational normality* as a technology facet.

The next propagation step starts with the assets “real-time video” and “real-time photo”, which forward the technology trustworthiness facet *usefulness* as optional to the “actual appearance”. The uploaded “profile picture” and the “actual appearance” are not associated with any clear trustworthiness facets, which is why they do not pass on any to the “pattern recognition algorithm”. “Authenticity information” is neither associated with any facets. Therefore, the “pattern recognition algorithm” only receives the trustworthiness facet *usefulness* for the online dating application. As the “appearance verifier” is already associated with *ability* and *usefulness*, the “pattern recognition algorithm” solely propagates the technology facet *functionality*.

With that, the propagation phase ends. The “information icon” and “information about [the] appearance verifier” are not related to any trustworthiness facets. The

“authenticity score” holds the same facets as the “appearance verifier”.

In the end, the list of trustworthiness facets for the concept feature is created. It is depicted in Figure 13.13. For this example, it only contains the trustworthiness facets from the trust-related software feature “appearance verifier”.

List of trustworthiness facet for the concept feature “Authenticity Check”

Trustworthiness facets – user

- Agreeableness
- Credibility
- Integrity
- Reliability
- Attractiveness
- Ethicality
- Openness
- Truthfulness
- Authenticity
- Honesty
- Promise fulfillment
- Willingness to disclose

Trustworthiness facets – technology

- Ability
- Situational normality
- Functionality
- Usefulness
- Information quality

Figure 13.13: List of trustworthiness facets for the concept feature “authenticity check”.

The last step of the propagation phase is to check the validation conditions.

Validation conditions for the Propagation Phase

✓ Propagated trustworthiness facets are included in the asset information of the parent asset of the catalogue for trust-related software features.

✓ Optional propagated trustworthiness facets are emphasised accordingly within the asset information of the catalogue for trust-related software features.

● Propagated trustworthiness facets that stem from an optional child asset have a line on the left side of their box.

As this example not illustrated the feature model after the propagation phase, this validation condition cannot be checked.

● Propagated trustworthiness facets that stem from a mandatory child asset do not have a line on the left side of their box.

As this example not illustrated the feature model after the propagation phase, this validation condition cannot be checked.

- ✓ The list of trustworthiness facets for the concept feature contains all trustworthiness facets within the feature model once.

14

Related Work

TrustSoFt is a requirements engineering method for the planning and analysis phase of the Software Development Life Cycle. It is created for the development of social media applications, such as CMI. To consider the psychological process of trust building in software development, TrustSoFt covers various disciplines, such as trust research, requirements specification, risk analysis, and the front-end design of software applications. The multidisciplinary background of TrustSoFt is a main factor for its constitution of various method components. These components rank from qualitative and quantitative methodologies to different modelling notations that consider users' trustworthiness assessment. To the author's knowledge, such an alignment to trustworthiness as well as such a composition of methodologies and disciplines do not exist for the development of social media applications to this date. However, leaving the trust context and the composition of the TrustSoFt components aside, there are indeed methodologies for software development in research and industry related to the individual ones used in TrustSoFt. Therefore, this chapter will frame the single components of TrustSoFt in the context of related work.

14.1 Users' Trust Issues in Software Development

TrustSoFt is focused on user trust during its complete process. Especially trust concerns and trustworthiness facets provide a basis on which consideration software seeks to support users in their trustworthiness assessment. In the following, it is analysed to what extent user trust is addressed in related work about software development that resembles the concept of trust concerns and trustworthiness facets.

Regarding trust concerns, previous work mainly uses the term, if the application field is sensitively related to trust, such as online banking [170] or blockchain [64], or when it is about technology to run reliably [229]. In itself, however, the term trust concern is not widely used. In related work, trust concerns are rather problems in the respective application field, which reduces trust in the used technology if the problem occurs. Consequently, related work focuses on the actual trustworthiness of technology that is objectively assessable. Trust concerns are addressed by ensuring countermeasures and guaranteeing failure tolerance for technical problems that maintain the trustworthiness of the technology. Usually, related research identifies trust concerns by literature search that has identified risks or problems in the respective field.

To sum it up, while related work considers trust concerns to enable trustworthy running software, this dissertation considers trust concerns for supporting the establishment of people's trust relationships via social media through software. In doing so, this work considers how users perceive the trustworthiness of other users, the service provider, and technology, instead of solely focusing on the trustworthiness of technology.

To the author's knowledge, there are no existing approaches that consider the subjective trust processes of users in software development for this purpose. When it is about the user perspective as an input for the planning and analysis phase, state-of-the-art favours the analysis of user needs and pain points [177]. User needs and pain points are considered to increase user satisfaction and user experience [181]. For that purpose, software development teams in the industry use methodologies such as user stories to determine target users, their needs, and motivated behaviours [69]. User stories are a common approach in agile software development that is popular for increasing the speed of software deployment while continuously addressing user needs. Usually, user stories are a basis for specifying software requirements and deriving software features for the front-end design [202]. They provide input for the subsequent technical steps of coding and how users may experience the front-end design [69]. With the trend for user-centered software development by user stories, it is assumed that service providers use them as a starting point when aiming to support users' trustworthiness assessment online.

Even though user stories accompany software development throughout the whole process, they are not designed for considering underlying elements of psychologi-

cal processes like the trustworthiness assessment. For that reason, this work has introduced the trustworthiness facets to set a benchmark in the sense of the trustworthiness assessment that is considered throughout the complete planning and analysis phase. Thereby, requirements engineering is aligned with the trust context. As mentioned in chapter 3.2, previous research has identified scattered trustworthiness facets for individuals, organisations, and technology often independent of the software development context. In the software development context, characteristics related to trustworthiness such as the trustworthiness facets, are rarely found. Mohammadi et al. have provided an overview of software qualities contributing to the trustworthiness of technology and software [230]. The characteristics of Mohammadi et al. have partly been included in the overview of trustworthiness facets in Chapter 3.2.

Instead of focusing on trust-related characteristics like trustworthiness facets, other research broadly analysed general trust elements that foster user trust. Friedman et al. identified ten so-called “engineering conditions for cultivating trust online” [104]. These are for example the reliability and security of the technology, obtaining informed consent about the harm and benefits of participating in the online interaction, and providing visual cues for the trustworthiness assessment and the cues’ saliency within online environments. Latter can be achieved by presenting status cues that increase user confidence in the source and quality of information. Another engineering condition for cultivating trust online is “knowing what people online tend to do”. Thereby, Friedman et al. recommend applying risk analysis to gain knowledge in what environment software developers try to foster user trust [104]. There are many parallels between the work of Friedman et al. and this dissertation. In fact, Friedman et al. point out some of the trustworthiness facets identified in this work, recommend design elements and propose development methodologies that are familiar to TrustSoft. While Friedman et al. briefly sum up relevant factors for software development to increase user trust, this dissertation provides detailed hands-on tools for software engineers to support users in their trustworthiness assessment of other parties. Although the given engineering conditions from Friedman et al. and the trustworthiness facets from this work differ in their purpose for software development, the findings of Friedman et al. confirm the relevance of TrustSoft components like the facets and the risk analysis. Simultaneously, their findings can complement the proposals for trust-related software features given in this dissertation.

Similar to the work of Friedman et al. is the one of Shneiderman, who aims to design trust into online experiences [277]. Shneiderman provides principles and guidelines to facilitate cooperative user behaviour to thereby gain user loyalty. He recommends service providers give assurances, references, certifications from third parties, and guarantees of privacy and security. In contrast to this dissertation, Shneiderman is already in the solution space of what features are needed to convince users of the provider's and application's trustworthiness. His work provides a shortcut for developers on what to implement to generate user trust – even if the provider or application is not trustworthy. In contrast, this work considers what trustworthiness facets are relevant for users to derive tailored software features for assessing whether to trust or not to trust.

By reviewing related work, the issue of generating user trust, no matter whether the trust in the service provider or application is justified, becomes more and more apparent. The question of ethics in this regard is later discussed in Chapter 15.2.

14.2 From Trustworthiness Goals to Trustworthiness Requirements with i* Goal Modelling

TrustSoFt is a method for requirements elicitation and improvement. While requirements elicitation is about finding new requirements, requirements improvement is the check of the initial requirements for errors, inconsistencies, or critical properties [138]. For that purpose, it is inspired by Mohammadi et al. to consider software goals and to make use of goal modelling in the context of trustworthiness [229, 231]. By determining software goals, a concrete direction is given for the elicitation of software requirements that reduces ambiguity in the process [160]. i* goal modelling supports the process by providing a way of modelling related elements for requirements elicitation and improvement [327].

In the context of software goals, Kim et al. introduce trust-aware goal modelling for the use case of cooperative self-adaptive systems[173]. Their objective is to support systems in interacting with other trustworthy systems. In their work, Kim et al. address trust-required situations through goal trees that are based on pre-defined software requirements. Trust-required situations occur in three conditions, that are i) informative, ii) interactive, or iii) irrelevant. Informative trust-required situations

mean that trust is needed if a situation requires additional information. Interactive trust-required situations denote that trust is needed during the interaction of two systems to obtain additional information. Irrelevant trust-required situations describe the necessity of trust when a system interacts with unknown and irrelevant other systems. By determining goals for a trust-required situation, system design can be better tailored to the respective situation.

The work of Kim et al. differs from this dissertation regarding the context, goal modelling notation, and way of realisation. Therefore, their work can only partly be related to this one. In contrast, i^* goal modelling allows mapping all TrustSoFt elements to a model and further refining each element in more detail. Yet, the classification of goals in trust-required situations by Kim et al. is inspiring regarding users' trust concerns. Trust concerns could be classified similarly in trust-required situations as in the work of Kim et al. but in the context of social media and CMI. Such a classification could be linked to a proposed set of trustworthiness requirements and trust-related software features that meet the challenge of the trust-required situation. Thereby, TrustSoFt practitioners could be guided in the process of requirements and feature selection.

A related goal modelling approach is the one of Gans et al. [106]. They use i^* goal modelling for considering multi-perspectives for team-oriented business process analysis in social media. In particular, Gans et al. use their approach to relate trust issues of social media stakeholders to their monitoring of social network rules and requirements. For that purpose, Gans et al. consider the viewpoint of various stakeholders while taking the trust in individuals, confidence in the network, and distrust into account. Furthermore, they include the vulnerability of stakeholders in dependence on others and the temporal sequence of their actions in the models. Therefore, Gans et al. introduced new model elements for displaying pre- and post-conditions of tasks that are related to the expectations of stakeholders. Similar to this work, Gans et al. adopted i^* goal modelling to their context. However, their context differs from the purpose of TrustSoFt to elicit requirements for social media software to support users in their trustworthiness assessment. Therefore, the adoptions of both works are hardly transferable to the other context.

Independent of trust is the Annotated Goal-Oriented Requirements Analysis (AGORA) [138]. Like TrustSoFt, AGORA also considers conflicts among goals, their resolution, and analyses associated requirements changes [160]. For these purposes, Kaiya

et al. included contribution values and preference matrices to AGORA. Contribution values represent the degree a sub-goal contributes to the main goal. The preference matrices represent how much the stakeholders relevant to the problem being modeled prefer a goal. AGORA and TrustSoFt differ insofar that AGORA is software-centred and illustrates a “state of the world”. Software states, actions, and conditions as a form of goals are modelled. In contrast, TrustSoFt is user-centred and agent-oriented by providing insights into involved actors, like the user and the software. Intentional elements of the actors are modelled. Consequently, AGORA and TrustSoFt represent different perspectives in requirements engineering. AGORA inspires this work for new goal modeling elements by which TrustSoFt can be extended. Modified contribution values could be used as an evaluation criterion for requirements for the later validation phase.

14.2.1 Risk Analysis for Conflict Resolution and Requirement Prioritisation

TrustSoFt uses risk analysis for the resolution of conflicting TrustSoFt elements, the re-specification of trustworthiness requirements, and their prioritisation. In the past, related work has also used risk for software development and requirements engineering.

ProCOR is a model-based software development process [319] that builds on the model-based risk management method CORAS [83]. CORAS is based on the ISO 31000 standard (see Chapter 2.5). It provides documentation guidelines and a modelling language for assessing risks and identifying treatments. ProCOR is based on the documentation guidelines and the modelling language of CORAS to elicit security requirements. In doing so, ProCOR uses risk similarly to TrustSoFt in the form of risk management. It derives security requirements from functional requirements, whereas TrustSoFt relates trustworthiness requirements to trustworthiness facets that resemble non-functional requirements. The main difference between TrustSoFt is that CORAS and ProCOR aim to specify treatments for risk reduction. In contrast, TrustSoFt uses risk management to choose among conflicting requirements to implement the one which reduces risk the most.

Similar to that is the work of Yoon et al. [326]. They relate risks and requirements

to prioritise test cases. Yoon et al. evaluate the failure likelihood of requirements as well as the severity of negative consequences. As a technique, they use the Analytic Hierarchy Process [268], which weighs relevant factors that are then used for the prioritisation of requirements in terms of risk. Based on the prioritisation, adequate test cases are chosen. This procedure resembles the goal prioritisation of TrustSoFt. Future work needs to investigate how far the findings of Yoon et al. can be adopted in TrustSoFt.

Concerning conflicting software requirements, Mairiza et al. conducted a literature review for identifying approaches to conflict identification, analysis, and resolution of non-functional requirements [207]. Of the reviewed works, some resemble TrustSoFt insofar that they also consider stakeholder preferences in conflicts [121], the level of importance concerning conflicts [308], and how to facilitate the negotiation for conflict resolution [31]. The literature review of Mairiza et al. [207] shows that TrustSoFt is in line with existing approaches and demonstrates its relevance in the research field. Yet, different from the review work, TrustSoFt is backed up by risk as a guiding scale to identify, analyse, and handle conflicting software requirements.

Last but not least, Horkoff and Yu also introduce a goal modelling approach that combines risk analysis and conflict resolution [138]. They extended Tropos goal modeling [43] by modelling risks to elicit countermeasures based on the work of Asnar and Giogini [13]. The objective of the approach by Horkoff and Yu is to satisfy stakeholder goals on an organisational level for goals whose risks and costs are acceptable. Thereby, their work differs from TrustSoFt in that it focuses on organisational goals, which in turn involve risks and costs of an economic background. Yet, its procedure is similar to TrustSoFt in a way that it may inspire to identify countermeasures to reduce risks related to the psychological process of users' trustworthiness assessment.

14.3 Software Features and Feature Modelling in the Context of Trust

TrustSoFt aims at the elicitation of software features that reflect the trustworthiness facets of other users, the service provider, or the technology. Thereby software

features shall enable users' trustworthiness assessment. Feature models are further used for the configuration and validation of trust-related software features.

As trust-related software features can also be defined as digital nudges, the model for the design of nudges (DINU model) is a related work [225]. First, the DINU model provides a catalogue of existing nudges that can be used as input for feature models of trustworthiness. Second, it further guides practitioners in the analysis, design, and evaluation of nudges. Yet, the nudges presented in the DINU model are not trust-related to that date. In addition, the DINU model lacks the model-based approach of the feature models. It is also not directly suitable for providing suggestions for online trustworthiness assessment. However, the DINU model is a tool that offers practitioners initial ideas on how nudges can be designed. Therefore, the DINU model can serve as input for the method for feature models for trustworthiness. By modelling the nudges in feature models for trustworthiness, they can be turned trust-related and be tailored for users' trustworthiness assessment.

In terms of feature models, Martinet et al. invented a tool for selecting appropriate features based on their attributes [211]. For that purpose, they created algorithms on the basis of petri nets. Similar digital tools for the configuration of software product lines by feature models are the FeAture Model Anayser (FAMA) [24] or Requiline - a requirements engineering tool for software product lines [310]. For the interdisciplinary trust background, such digital tools are missing. Yet, the ones mentioned here provide an excellent basis to complement the feature models for trustworthiness with a tailored digital tool that facilitates the configuration of trust-related software product lines.

15

Discussion

This dissertation has examined how software engineers can proceed in the planning and analysis phase to develop social media applications that support users in their online trustworthiness assessment. In this chapter, the results are discussed.

For that purpose, Section 15.1 takes up the research questions from the introduction and provides an overview of the respective findings. Afterwards, Section 15.2 discusses each finding and refers to their limitations and future work. Last but not least, Section 15.3 emphasises the theoretical and practical implications of this dissertation.

15.1 Results

The research objective of this dissertation is to provide solutions for how software engineers can support social media users in their online trustworthiness assessment by developing adequate software applications. To provide solutions to the research objective, this dissertation is divided into different parts following the associated scientific papers.

The first part of the dissertation theoretically analyses the **context of trust** for software development and social media in Chapter 3. This part corresponds to Papers 1 and 2. As a next step, the dissertation introduces **TrustSoFt as a method** for software engineers to develop social media applications that address users' trustworthiness assessment. What first is a conceptual method is enriched by additional steps, guidelines, and modelling notations. This part of the dissertation complies with Papers 3-6 and is presented in Chapters 4-6, and Chapters 9-11. The third part of the dissertation is about **the application of TrustSoFt**. It has been

applied in academic projects for the development of online dating and Sharing Economy applications. Furthermore, it has been used to develop the hybrid social media application “HushTweet”. Regarding these TrustSoFt applications, TrustSoFt has been evaluated by the ease of application for development teams and its resulting software features’ impact on users. In addition, TrustSoFt has been applied as an application example to counter the trust concerns of male and female online dating users. The third part of the dissertation is covered by Papers 7-9 and is presented in Chapters 7, 8, 12, and 13. Since the findings from the TrustSoFt application have provided new insights for the development of the TrustSoFt method, the second and third parts of the dissertation are presented through interlocking chapters.

An overview of the findings of this dissertation is given in the following by presenting the results for each research question. An overview of the research questions, results, applied methods, and associated papers are presented in Tables 15.1-15.3.

The first Research Question RQ1 is about how trustworthiness is involved in social media and CMI systems. By literature research, three different types of trust are identified as relevant, which are i) computer-mediated interpersonal trust, ii) brand trust, and iii) system trust. With these different types of trust, three parties are associated with social media and CMI use, namely users, organisations like the service provider, and technology such as the social media application. Furthermore, the literature search revealed that for each party, different traits are associated with their trustworthiness. Based on these findings, trustworthiness facets have been introduced. Previous research further has shown that trustworthiness facets are assessed by perceivable cues. In conclusion, perceivable cues in social media applications are software features in the user interface. The relations between the various types of trust, trustworthiness facets, and trust-related software features are modelled in a conceptual framework.

With the introduction of the trustworthiness facets, Research Question 2 asks for the facets of individuals, organisations, and technology that have been identified already by previous research. A total of 163 trustworthiness facets have been identified of which 68 are for individuals, 40 for organisations, and 55 for technology. The overview of trustworthiness facets serves as an external input by which software engineers can select relevant facets for the special use cases for which they develop software. This process is supported by the guideline for selecting trustworthiness

Chapters & Publication	Research Question	Results	Applied Methods
Chapter 3.1 Paper 1: Building Trustworthiness in CMI: A Facet-oriented Framework	RQ1: How is trustworthiness involved in social media and CMI systems?	R1: The trustworthiness of i) users, ii) the service provider, and iii) the application is conveyed by software features in the application.	Literature research, conceptual framework
Chapter 3.2 Paper 2: The Role of Trustworthiness Facets for Developing Social Media Applications: A Literature Review	RQ2: What are the trust- worthiness facets of i) individuals (e.g. users), ii) organisations (e.g. service providers), and iii) technology (e.g. software applications)?	R2: Collected trustworthiness facets from more than 100 references. They can be used for the development of social media. A guideline for software engineers supports them in identifying relevant trustworthiness facets for each use case.	Structured literature review, design thinking principles, UML activity diagram
Chapter 4 Paper 3: A Conceptual Method for Eliciting Trust-related Software Features for CMI	RQ3: How can software developers build social media system that support users in their trust- worthiness assessment?	R3: TrustSoFt - A conceptual method that provides software engineers with essential steps to develop social media systems concerning the online trustworthiness assessment	Conceptual framework
Chapter 5 Paper 4: Balancing Trust and Privacy in CMI: Featuring Risk as a Determinant for Trustworthiness Requirements Elicitation	RQ4: How can software engineers decide on conflicting trust- worthiness goals and requirements during social media development?	R4: TrustSoFt is extended by further steps involving risk assessments for the prioritisation of trustworthiness goals and requirements.	Conceptual framework

Table 15.1: Overview of the results from this dissertation for Research Questions RQ1 - RQ4

facets. For the guideline, design thinking principles need to be used. The guideline is presented as a UML activity diagram.

After the theoretical basis has been established, Research Question RQ3 asks how software developers can build social media systems that support users in their trustworthiness assessment. The solution approach presented by the dissertation is the method for eliciting trust-related software features - TrustSoFt. It proposes to identify users' trust concerns and workarounds as a starting point. From these, trustworthiness facets and trustworthiness goals are derived, followed by trustworthiness requirements. These in turn are realised by trust-related software features. Throughout the process, the trustworthiness facets are taken into account for enabling users to perform their trustworthiness assessment.

As a next step, the issue of trustworthiness goals and requirements that are in conflict with each other is addressed. For that reason, Research Question RQ4 asks how software engineers can decide on conflicting trustworthiness goals and requirements during social media development. As a solution approach, this work proposes to include risk assessments in TrustSoFt. Risk serves as a determinant for deciding what alternative is implemented. As trustworthiness goals aim for the good of users, those goals of a conflict should be addressed that reduce risks the most. In terms of the trustworthiness requirements, they may pose side effects that are associated with risks. In a conflict, those trustworthiness requirements shall be implemented that are associated with the least risk. In addition to the new element of risk assessments, the dissertation proposes ways to resolve conflicts so that both conflicting alternatives can somehow be implemented. Approaches for conflict resolution are the re-specification and refinement of trustworthiness requirements.

Due to its complexity, TrustSoFt needs to be aligned for structured application. Research Question RQ5, therefore, asks how TrustSoFt can be conducted systematically with a model-based approach. This work proposes to apply the i^* goal modelling for TrustSoFt. TrustSoFt and the i^* goal modelling notation resemble each other in their structure and elements. Therefore, the i^* goal modelling notation is adapted and partly enriched by TrustSoFt elements. By applying the adapted i^* goal modelling notation, software engineers are supported when using TrustSoFt. Results are documented and provide a basis for the subsequent steps of the software development life cycle.

Chapters & Publication	Research Question	Results	Applied Methods
<p>Chapter 6</p> <p>Paper 5: Conflict Identification and Resolution for Trust-related Requirements Elicitation: A Goal Modeling Approach</p>	<p>RQ4: How can software engineers decide on conflicting trustworthiness goals and requirements during social media development?</p> <p>RQ5: How can the software development process for supporting users' trustworthiness assessment be conducted systematically as a model-based approach?</p>	<p>R4: TrustSoFt is extended by the i* goal modelling notation for a structured, model-based application. The i* goal modelling notation got extended by TrustSoFt elements. The structure of the i* goal models supports the identification and resolution of conflicting trustworthiness goals and requirements.</p>	<p>i* Goal Modelling</p>
<p>Chapters 9 - 11</p> <p>Paper 6: Meeting Strangers Online: Feature Models for Trustworthiness Assessment</p>	<p>RQ6: How can trust-related software features be created, documented, configured, and validated?</p>	<p>R6: The method for establishing feature models for online trustworthiness assessments is introduced</p>	<p>Feature modelling</p>
<p>Chapter 12</p> <p>Paper 7: Mitigating Privacy Concerns by Developing Trust-related Software Features for Hybrid Social Media</p> <p>Paper 8: The Relevance of Privacy Concerns, Trust, and Risk for Hybrid Social Media</p>	<p>RQ7: How do software features resulting from software development to support users' trustworthiness assessment impact users?</p>	<p>R7: Resulting software features from TrustSoFt do positively impact the trusting beliefs of users in the application and reduce trust concerns. Yet, the choice of trust concerns to be addressed by software features is relevant for the effectiveness of the impact the software features have on users.</p>	<p>Online user survey, quantitative methods (e.g., ANOVA, SEM), prototyping</p>

Table 15.2: Overview of the results from this dissertation for Research Questions RQ5 - RQ7

To this point, TrustSoFt focuses on the specification of trustworthiness requirements. These in turn are significant for deriving trust-related software features that address or reflect the trustworthiness facets in the user interface of the application to be developed. However, the specification of trust-related software features is not supported so far. Therefore, Research Question RQ6 asks how trust-related software features can be created, documented, configured, and validated. As a solution, this work proposes feature modelling. The notation of feature models is adapted to the trustworthiness facets. Furthermore, a method for developing feature models for online trustworthiness assessment is introduced. It is presented by a UML activity diagram. The method includes feature model creation, the relation of features and trustworthiness facets, feature validation, and the configuration of software product lines.

Each of the TrustSoFt steps is now addressed by supportive information, approaches, and methodologies for software engineers. As a next step, TrustSoFt has been evaluated in two ways. First, the ease of use has been evaluated by academic product development teams for the development of online dating and sharing economy applications. Their feedback was positive about the results for each TrustSoFt step. Especially the identification of trust concerns through user interviews was rated as highly useful. Concerning the trustworthiness facets, the feedback was that they should be foremost related to the trustworthiness requirements rather than to the trustworthiness goals as originally proposed. The main drawback in their opinion was i^* goal modelling, which was perceived as time-consuming. Instead, the development teams preferred the result tables as more efficient for the documentation of the TrustSoFt application. The results of the evaluation were addressed by updating TrustSoFt in Chapter 8.

Besides the ease of use of the TrustSoFt method, Research Question RQ7 asks for the impact of the resulting software features on users and trustworthiness. To answer this question, TrustSoFt has been applied for the development of the HSM application “HushTweet”. Beforehand, a research model was established for information privacy concerns, trusting beliefs and risk beliefs of users in HushTweet, and users’ willingness to use HushTweet. For the six information privacy concerns, software features have been designed in prototypes that addressed each concern. Through an online user survey, users’ trusting beliefs, risk beliefs, and their willingness to use HushTweet were additionally tested when they used HushTweet which had those

Publication	Research Question	Results	Applied Methods
Chapter 13.1 Paper 9: Safety First? Gender Differences in Online Dating Behavior and Trust Concerns	RQ8: What are the trust concerns of female and male online dating users?	R8: Women are especially concerned about their safety. Men are worried about deviating outcomes of what they have expected about online dating use. Men are additionally concerned about the profit orientation of the service provider. Both female and male users are concerned about fake profiles.	Qualitative research: interviews

Table 15.3: Overview of the results from this dissertation for Research Question RQ8

software features implemented. The results have shown that the trusting beliefs increased when the concerns were addressed by the software features. Moreover, the results have shown that depending on which information privacy concern features have been developed, the features addressed even more than one information privacy concern. This is an indicator that for the first step of TrustSoFt, the choice of addressed trust concern is highly relevant for the effectiveness of resulting software features.

The last part of the dissertation (without considering the conclusion) is the TrustSoFt application for the application example of addressing trust concerns of male and female online dating users. For that purpose, an interview study has been conducted to answer Research Question RQ8: “What are the trust concerns of female and male online dating users?”. With the application example, the objective was to give recommendations for online dating applications about software features that address gender needs. The results have shown that women are especially concerned about their safety. Men were especially concerned about deviations from their expectations regarding online dating use including the dating partners. Different from women, men were concerned about the profit orientation of the service provider. Male and female online dating users share concerns about fake profiles. After conducting i* goal modelling and feature modelling, this dissertation proposes the software feature “date check” for the safety concern of female users, and the feature “authenticity verifier” for male users. With date check, online dating users can agree on a time and location for an offline encounter. The feature authenticity

verifier supports online dating users in checking on the authenticity of online dating pictures.

In the next chapter, the results will be discussed.

15.2 Discussion of Results

The discussion addresses the results of each research question (see Chapter 15.1 and Tables 15.1, 15.2, and 15.3) and refers to their thematic and methodological context. In addition, limitations and future work are pointed out and additionally summarised in bullet points for each research question in the respective paragraph. After the discussion of each research question and its results, the theoretical and practical implications of this dissertation are emphasised. Concerning the whole approach of supporting trustworthiness assessments in social media by software development, the ethics are discussed in Chapter 15.2.2 which is related to trustworthiness in the context of CMI.

15.2.1 RQ1 – Trustworthiness in the context of CMI

RQ1: How is trustworthiness involved in social media and CMI?

The trustworthiness framework for CMI puts trustworthiness in the context of social media and CMI. It shows the relationships between the three trust types “computer-mediated interpersonal trust”, “brand trust”, and “system trust”, the trustworthiness facets, and CMI systems with their software features. The trustworthiness framework increases the understanding of trust and trustworthiness in the context of social media and CMI for the development of software applications. Furthermore, it provides a theoretical model for this context. Yet, at the time of the framework’s establishment, the framework has not been evaluated for correctness. It is solely based on literature research. As a limitation, the literature of the literature search does not completely focus on CMI and social media.

This dissertation especially considers the relationship between software features and the trustworthiness facets. The solution approach to the research objective is

to enable users' trustworthiness assessment by picking up the trustworthiness facets of the CMI parties user, service provider, and application by the software features. Chapter 12 has partly evaluated the trustworthiness framework by analysing how trust-related software features that are associated with trustworthiness facets impact people's trust in the HSM application "HushTweet". The study considered the trustworthiness facets ability, benevolence, integrity, and predictability. Effects have been confirmed for the relation of trust-related software features to these trustworthiness facets and to the perceived trustworthiness of HushTweet (system trust). Therefore, the validity of the trustworthiness framework for CMI is conditionally supported. This result is an indication that the trustworthiness framework can serve future research as a theoretical model. Yet, the trustworthiness framework for CMI may be analysed in more depth for each of its branches to be completely validated. For that purpose, it is relevant to examine the relation of software features to their associated trustworthiness facets and the features' impact on computer-mediated interpersonal trust and brand trust. An approach to realise this sort of study is for example by using the trustworthiness framework as a theoretical model for the validation phase of the method for establishing feature models for the trustworthiness assessment (see Chapter 11). By the validation of the relevance of the trustworthiness facets for the associated software features of the feature models, the trustworthiness framework for CMI can further be evaluated.

Another remark for the trustworthiness framework for CMI is the high dependence on how trust-related software features have been developed and designed. The quantitative study about the HSM application "HushTweet" in Chapter 12 has shown, that the impact of HushTweet's perceived trustworthiness for users differs depending on the trust-related software feature. This finding reflects that the trustworthiness framework only illustrates that trustworthiness facets can be picked up by software features. Yet, it does not consider the extent to which a software feature realises or reflects a facet and, thus, impacts perceived trustworthiness. Therefore, the trustworthiness framework for CMI can be regarded as a guiding model for software engineers on how they can pursue trustworthiness assessment support in their software development projects.

Limitations of this work

- The trustworthiness framework is based on literature whose focus is not only on social media or CMI
- The validity of the trustworthiness framework has only been tested for a few trustworthiness facets.

Future Work

- The trustworthiness framework should be supported by more validation.
- The trustworthiness framework can serve future work as a theoretical framework or guidance for software development.

15.2.2 Ethics about Trustworthiness in the Context of CMI

However, the provided knowledge of the trustworthiness framework for CMI can also be misused for unethical purposes. Software developers may purposely reflect or realise trustworthiness facets in the user interface that are not given to increase the trustworthiness of the application or the service provider. To distance from this intention, this dissertation always highlights the objective to support users in their online trustworthiness assessment by reflecting truly given trustworthiness facets relevant to scenarios of uncertainty such as users' trust concerns. Unfortunately, there is little that can be done to ensure that the trustworthiness framework for CMI is only used in accordance with ethics. In the end, it is the practitioners' responsibility for which purpose they use the trustworthiness framework. Yet, some possibilities can be considered so that unethical use is demotivated. Cross argues that when a party lacks trust in another party whether it will behave appropriately, legal regulations can create trustworthy behaviour [76]. As an example, in German criminal law, fraudulent misrepresentation is considered a criminal offense. A fraudulent misrepresentation is an intentional deception committed by deliberate misrepresentation or concealment of true facts, although there is a duty to inform [332]. However, in regards to falsely presenting an application or a service provider as trustworthy by software features, it can be argued that fraudulent misrepresentation does not apply. By software features, a misrepresentation of trustworthiness does not need to be directly stated. Instead, users may misinterpret the trustworthiness of an application or a service provider, for which an application or service provider cannot be held directly responsible. In addition, the service provider does

not have the duty per contract to inform users about its trustworthiness. Yet, fraudulent misrepresentation shows a way how to deal with the ethical problem of falsely presenting oneself as trustworthy online. It can be argued whether legal regulations should address the misrepresentation of trustworthiness.

Another possibility to deal with the ethical question of how to protect users from applications that have misrepresented their trustworthiness is by using trust badges [201]. Trust badges are software features that are clickable icons displaying that a service provider is trustworthy. Trust badges are either the promise of a service provider to stick to a certain code of conduct [201] or are certificates from external companies that have checked the service provider and their processes [129]. An example of a trust badge is the securedshop¹ seal of approval for e-commerce.

Future Work

- Future work needs to consider what measurements can be undertaken to prevent the misuse of the trustworthiness framework in an ethical sense.

15.2.3 RQ2 – The trustworthiness facets

RQ2: What are the trustworthiness facets of i) individuals (e.g. users), ii) organisations (e.g. service providers), and iii) technology (e.g. software applications)?

The overview of trustworthiness facets is a collection of facets that can be attributed to individuals, organisations, and technology. The guideline for selecting relevant trustworthiness facets supports software engineers in choosing facets from the overview for the given problem to be addressed by the software to be developed.

The overview of trustworthiness facets is based on a literature review by which each facet is scientifically proven to positively impact both trustworthiness and perceived trustworthiness. Thereby, a large collection of trustworthiness facets could be gathered by which software engineers can address users' needs for the trustworthiness assessment precisely by the software application. To enable such a large

¹www.securedshop.de

overview of trustworthiness facets, the literature review, on which the overview is based, includes facets from research that stems from different contexts. Software engineers must always consider, whether the facets from the overview match the context they want to address. In addition, trustworthiness facets are in general highly context-dependent. Depending on the context, attributes might be relevant for trusting another party that usually are irrelevant for the context of trust. An example is observable in the use case of Chapter 13 for the safety concern of female online dating users. In the safety context, the physical strength of male online dating users has been identified as a trustworthiness facet, by which female users derive whether to trust male users or not. Physical strength is not part of the overview of trustworthiness facets. It can be considered a niche facet. Taking this into account, the overview of trustworthiness facets provides facets that are usually broadly applicable to trust in many contexts. Yet, it may lack trustworthiness facets that are relevant for special scenarios. Therefore, it is always recommended to first increase one's understanding of a specific context, talk to stakeholders, consult the overview of trustworthiness facets, and consider all gained insights and information sources when selecting trustworthiness facets. Therefore, the guideline for selecting trustworthiness facets does not solely rely on the overview but advises an analysis of the problem space in terms of the trustworthiness facets. Research has shown that the availability and accessibility of additional sources highly impact the results [200]. On these grounds, it is recommendable to invest time and costs to conduct a user survey or interview experts for trustworthiness facets in addition to relying on the overview of trustworthiness facets. By such user surveys, the overview of trustworthiness facets could be extended by facets for specific contexts.

Another reason to consider multiple sources for selecting appropriate trustworthiness facets for a scenario is the limitation that practitioners underlie their subjectivity in the selection process. Practitioners select trustworthiness facets according to their understanding of which characteristics are necessary to assess the trustworthiness of the involved parties. In addition to the understanding of the context, this relevance evaluation requires not only an understanding of the problem but also the practitioners' ability to empathise and take the user's perspective. As empathy and perspective-taking are personal skills [258], the trustworthiness facets that are identified as relevant might differ depending on the practitioner. Another way to counteract the limitation of subjectivity, despite considering multiple sources, is by conducting the guideline for selecting relevant trustworthiness facets in a group of

practitioners. Moløkken and Jørgensen analysed group discussions as a way to reduce subjectivity and individual biases [236]. Future work can test the guideline for selecting trustworthiness facets in group discussions. By validating the process and the resulting facets from individuals and groups for differences, the guideline can be enhanced to support group discussions and avoid individual biases.

Limitations of this work

- The trustworthiness facets in the overview stem from research that applies to a variety of contexts. Software engineers need to check whether the facets are applicable to the context they are dealing with.

Future Work

- User studies for the identification of trustworthiness facets for specific scenarios may lead to the extension of the overview of trustworthiness facets.
- Validation and improvement of the guideline for selecting trustworthiness facets regarding group discussions and the individual bias of practitioners

15.2.4 RQ3 – The introduction of TrustSoFt as a conceptual method

RQ3: How can software developers build social media systems that support users in their trustworthiness assessment?

TrustSoFt guides software engineers to address users' trust concerns by considering workarounds, identifying relevant trustworthiness facets, and specifying trustworthiness goals, requirements, and trust-related software features. Thereby, software developers can support users in their trustworthiness assessment through tailored software applications. Chapter 4 already points out the challenges of the conceptual TrustSoFt method that are addressed within this dissertation. Challenges have been for example conflicting goals and requirements resulting from TrustSoFt that are addressed by considering risk within TrustSoFt, or the lacking overview of how the multitude of resulting TrustSoFt elements relate to each other that is addressed by i^* goal modelling. While the solution approaches these challenges

specify the conceptual TrustSoFt method in more detail, the general applicability of TrustSoFt to reduce users' trust concerns is now discussed.

The results of the user study about the HSM application "HushTweet" confirm that software features resulting from TrustSoFt mitigate user concerns and support users' assessment of whether HushTweet is trustable regarding their concerns. The findings confirm that TrustSoFt really provides a solution for how software engineers can address trust concerns and support the online trustworthiness assessment. In the case of HushTweet, users' privacy concerns can be considered to be the same throughout the use of the application. It is questionable whether, for CMI and direct online interpersonal interactions, user concerns are stable throughout the experience. Obada-Obieh and Somayaji discussed that online dating applications should consider different trust mechanisms and software requirements for the different phases of online dating usage to better address the varying user needs [243]. In contrast to broadcasting social media like Twitter or Facebook, where one user publishes content to a whole audience, CMI has three different stages of interpersonal interaction. The stages are i) *before* an online interaction, 2) *during* an online interaction, and 3) *after* and online interaction [38]. In the before-stage, users look for other users who might match their intentions. In the during-stage, users communicate via the CMI application. In the after-stage, people have either shifted the interaction to another environment, for example offline, or have ended the interaction completely. Reflecting on the safety and misrepresentation concerns of online dating users from the use case in Chapter 13.1, it is conceivable that the concerns are differently intense in the different stages and may involve different trustworthiness facets for the trustworthiness assessments. Female users might be more concerned when meeting men offline, where their physical strength has a different effect on them than when considering male users in the before-stage. Male users may be more focused on the honesty of female users in the during-stage to identify misrepresentation while chatting with them than when inspecting online dating profiles in the before-stage. Therefore, it can be concluded that for each CMI stage, different facets should be considered and different trustworthiness requirements need to be specified to completely meet users' trust concerns. For future TrustSoFt applications, software engineers need to be aware of the different CMI stages when analysing the problem. Thereby, they can specify trustworthiness requirements and trust-related software features in a more targeted way.

Another point for discussion about applying TrustSoFt refers to the same argument of subjectivity as for the discussion of Research Question RQ2. The evaluation of TrustSoFt from Chapter 7 shows that the development teams perceived the process as highly dependent on the single practitioner. Practitioners had different ideas for trustworthiness requirements. A variety of ideas for trust-related software features occurred. While it can be regarded as a benefit that practitioners can work creatively with TrustSoFt to find a variety of different results, it is a drawback when speaking of a structured method that leads to the same results. The development teams of the TrustSoFt evaluation regarded it as both an advantage and a disadvantage. They proposed to apply TrustSoFt in the form of group discussions for a more unified solution approach or to let the practitioner present her thoughts to gain a unified understanding and jointly enhance the ideas.

Limitations of this work

- TrustSoFt is a method, whose results highly depend on the user applying TrustSoFt. The application of TrustSoFt might not be repeatable in a way that results in the same findings.

Future Work

- Future work should consider different stages of CMI usage when applying TrustSoFt to develop more targeted software features for the users' needs.

15.2.5 RQ4 – Considering risk for deciding on conflicting options

RQ4: How can software engineers decide on conflicting trustworthiness goals and requirements during social media development?

To handle conflicting TrustSoFt elements, “risk” is included in the method as a determinant. Risk assessments are conducted to evaluate how conflicting TrustSoFt elements mitigate or contribute to CMI usage risks. Conflicting TrustSoFt elements are either adjusted until the conflict no longer exists or practitioners decide on one option that either has a higher impact on risk reduction or less contributes to CMI risks.

Including risk to TrustSoFt is strategically important. It not only enables practitioners to manage conflicts but also plays a part in the trust context. As explained

in Chapter 2.1, trust is relevant for individuals in contexts of uncertainty in which they face risks [203]. In these contexts, individuals assess the trustworthiness of other parties to check whether they can build trust as a coping strategy to tolerate risks. Concerning TrustSoFt, implementing those TrustSoFt elements that keep CMI risks low leads to software applications that are increasingly trustworthy in fact. By those TrustSoFt elements, usage risks are reduced so that the application increasingly meets the users' expectations of successful usage. Software features should be designed in a way that users perceive the circumstance of reduced risks when they evaluate the resulting trust-related software features in the application. Such a trustworthiness assessment of the software application leads to an increased **perceived** trustworthiness of the software application.

Yet, conducting risk assessments for TrustSoFt elements is a cost-intensive process. Besides the skill of applying TrustSoFt, it needs time, access to databases that contain the relevant data for the given scenarios, and experts, who can estimate the risks. To counter these limitations, future work could develop a digital tool to conduct risk assessments as part of TrustSoFt. The digital tool could include risk matrices for risk assessment [259] that link the determined risks to the respective TrustSoFt elements. Furthermore, it could include risk databases for the given application fields, such as online dating or car sharing. The risk database could be a collection of current data on the frequency of occurrence and the impact of the various risks.

Future Work

- The development of an integrated tool in TrustSoFt that supports software engineers in the risk assessment of trustworthiness goals and requirements.

15.2.6 RQ5 – i* goal modelling for TrustSoFt

RQ5: How can the software development process for supporting users' trustworthiness assessment be conducted systematically with a model-based approach?

To apply TrustSoFt as a structured, model-based approach, an adapted form of

i* goal modelling notation is introduced. By i* goal modelling, each TrustSoFt step can be modelled. In addition, the goal models support conflict identification and resolution of conflicting TrustSoFt elements.

As the TrustSoFt evaluation in Chapter 7 has shown, the opinion about the i* goal models diverged. On the one hand, it was highly appreciated that creating the models supports a structured TrustSoFt application and cooperation with other development team members. On the other hand, i* goal models were criticised for becoming too large to be clear and too time-consuming in their creation. As discussed in Chapter 8, i* goal modelling for TrustSoFt seems to be a matter of skill and favour of each practitioner. Still, to address the drawbacks of goal modelling, Boness et al. suggest using automatic tools for drawing [32]. Thereby, the focus of the practitioners is on gaining new knowledge. Another suggestion for time-constrained projects is the iterative and interactive cooperation by a group of practitioners, as is the case for agile software development [32]. The findings of Boness et al. show that the discussion within groups leads to more findings than creating goal models alone.

Limitation of this work

- i* goal modelling for TrustSoFt depends on the skills and favour of the TrustSoFt users.
- i* goal models can grow big which can reduce the clarity of the whole model.

15.2.7 RQ6 - Feature models for trustworthiness assessment

RQ6: How can trust-related software features be created, documented, configured, and validated?

In order to design trust-related software features in more systematically, feature models for trustworthiness assessment and a method for establishing them are introduced. The feature models for trustworthiness assessment extend the feature modelling notation by the trustworthiness facets as attributes for each feature asset. Feature assets can be validated by their facets. The trustworthiness facets can

further be used as a basis for the targeted configuration of trust-related software features for supporting users' trustworthiness assessment. In addition to the feature models, a catalogue structure for trust-related software features is introduced for cataloguing each feature asset.

The use case in Chapter 13 has demonstrated that feature models for trustworthiness assessment enable practitioners to model their ideas for software features from TrustSoF in more detail. Establishing the feature models serves as planning for feature implementation. By adding the trustworthiness facets, the (intended) effect on the users can be documented, which is later validated in the validation phase. Besides linking the facets to the feature assets to support users' trustworthiness assessment, practitioners can also add facets to the feature assets that they intend for users to perceive. This possibility is critical in the ethical context that is discussed in Chapter 15.2.2. By modelling what practitioners want users to perceive – rather than just what users should be able to assess based on a feature asset – practitioners can manipulate perceived trustworthiness to their will. As for planning software development, feature models are a good tool. Yet, it is problematic to protect users from malicious intentions of software developers and service providers from the outside.

The trustworthiness facets are one example of the dependence on how a model is created based on the wishes of practitioners. Another example is the degree of detail that a model illustrates. Practitioners can decide in how much detail they want to create a software feature. The more details, the clearer it is for the development team what should be implemented in the software application. Feature models are a tool that practitioners can use to freely define features in terms of their algorithm, design, information, and interaction elements. However, similar to the *i** goal models, a drawback of the feature models is their increasing size. The size of the models gets even bigger when the trustworthiness facets are allocated to each feature asset. A way to counter this problem is by creating and using feature models in a digital tool. Within the digital tool, feature models could be depicted with their major feature assets that are expandable and made visible by clicking on them. Sub-trees or trustworthiness facets could be hidden to ensure the clarity of the models. Expanding relevant parts of the feature model while the rest is hidden leads to a focused view of what is currently relevant for the software development teams. Current digital tools for feature models are for example the FeAture Model Analyser

(FAMA) [24] and Requiline - a requirements engineering tool for software product lines [310]. Yet, these tools are not tailored for the feature models of trustworthiness assessment. They do not consider the catalogue for trust-related software features. By extending a digital tool by the catalogue structure, practitioners could search for feature assets by asset characteristics. Moreover, the tool could be used for the configuration of trust-related software features.

The last discussion point for the feature models refers again to the trustworthiness facets of the allocation and propagation phase. During these phases for the use cases in Chapter 13, it became apparent that trustworthiness facets are not the only relevant attributes for trustworthiness. Some feature assets had a negative effect on trustworthiness. This negative effect can be described by the problematic characteristics that have been introduced in Chapter 3.2. They are the opposite of trustworthiness facets. An example is dishonesty as the counterpart of honesty. These problematic characteristics foster distrust. Lewicki et al. argue to respect distrust as an own dimension [192]. They differentiate distrust from the low spectrum of trust and define it as an own research area. In this regard, future work for supporting users' trustworthiness assessment by software needs to further consider distrust during software development. The problematic characteristics from the guideline for selecting trustworthiness facets may serve as a first input for the distrust approach. For the feature models for trustworthiness assessment, problematic characteristics can be included as attributes. Thereby, the impact on perceived trustworthiness can be calculated anew for each trust-related software feature.

Limitations of this work

- The feature models for trustworthiness assessment can be misused for malicious intentions by determining trustworthiness facets that are actually not given.
- Feature models can grow big which can reduce the clarity of the whole model.

Future Work

- The development of a digital tool for the establishment of feature models for trustworthiness assessment. Such a tool can increase the clarity of models, can support the establishment of models and may guide the configuration of trust-related software features.
- Problematic characteristics of distrust should be considered in future work when creating feature models of trustworthiness assessment and designing trust-related software features.

15.2.8 RQ7 - Addressed concerns, trust, and risk in hybrid social media

RQ7: How do software features resulting from software development to support users' trustworthiness assessment impact users?

In Chapter 12, trust-related software features have been specified with TrustSoFt for the hybrid social media application “HushTweet”. Their impact was tested regarding whether they decrease the information privacy concerns of users for which they have been created. Furthermore, their impact was examined on people’s trusting beliefs in HushTweet. This chapter picks up the discussion about this research from Chapter 12. As this study applies to HushTweet, it must be said beforehand that the results cannot be generalised to other application fields. Yet, the results point out the impact trust-related software features have on HushTweet users. It can be assumed that the impact of features on users is similarly feasible for other contexts, such as CMI.

The results of this study have shown that all trust-related software features decrease users' information privacy concerns. However, contrary to expectations, it was mostly the case that the features reduced other concerns more than the concern they were intended for. A reason for this divergent impact can be that the concerns have a different level of significance for the users. Even though all concerns have been identified as relevant by former research, single concerns can be differently prominent in the given context [284] or concerns are differently important for different individuals [126]. Therefore, it is highly important for developers to directly ask (targeted) users of the software application about their concerns regarding usage. At the same time, some concerns might be related to each other so that they can be addressed by one and the same software feature. Again, it is important to analyse relations among concerns before specifying features. This allows the development to be carried out in a target-oriented manner.

Moreover, the results also have shown that trust-related software features positively impact people's belief that HushTweet can be trusted. In addition, the software features that targeted the trustworthiness facet *integrity* were successful in doing so. Yet, it was a challenge to test whether the targeted trustworthiness facets of features were addressed, as well. The size of the study did not allow extensive testing of each trustworthiness facet. With the increasing duration of the online survey, negative side effects could have taken place, such as the lapsing concentration of participants resulting in errors in answering the survey [99]. Furthermore, the single trustworthiness facets lack scientific questionnaires for testing them. Scientific questionnaires are mostly available for the "basic" trustworthiness facets "ability", "benevolence", "integrity", and "predictability", which are known to be the factors of trustworthiness [213]. Future work needs to develop more scientific questionnaires for them. This would also facilitate the validation of trustworthiness facets for the feature models of trustworthiness assessment. Furthermore, future work could conduct user experience testing for single trust-related software features to directly analyse their relationship with the trustworthiness facets.

Limitations of this work

- The size of the user study is too small for deriving conclusions about each trustworthiness facet.
- There are no scientific questionnaires for each trustworthiness facet.

Future Work

- Future work needs to analyse the relations of various concerns among each other when specifying trust concerns. Thereby, software features can be elicited that address more than one trust concern.
- Future work can conduct user experiences studies to analyse each trustworthiness facet that is addressed by a trust-related software feature.
- Future work can establish scientific questionnaires for the trustworthiness facets and their validation of being addressed by trust-related software features.

15.2.9 RQ8 – The use case: Safety and Misrepresentation Concerns

RQ8: What are the trust concerns of female and male online dating users?

In the context of this dissertation, Research Question RQ8 has been posed for obtaining a starting point for the TrustSoFt use case. On the one hand, female and male online dating users share trust concerns, while on the other hand, gender-specific trust concerns have been identified. Shared trust concerns are for example fake profiles concerning other users, the application's lacking ability to identify fake profiles, and the data usage of service providers. Differences between the trust concerns of female and male online dating users are amongst others women's safety

issues about male users and men's concerns that female users misrepresent themselves online. Concerning the service provider, male users were worried about their profit orientation, while female users have not considered the service provider in their concerns at all. In terms of the application, both genders have not stated any more, differentiating, concerns. As the safety and misrepresentation concerns have been addressed in the use case in Chapter 13, these two concerns are the focus of this discussion. It is discussed why the safety and misrepresentation concerns are the most prominent for female and male online dating users. Furthermore, it is discussed how the concerns are currently addressed by online dating applications.

The safety concern of female online dating users may relate to women's vulnerability in the physical and sexual sense compared to men. As men are stronger from their physics [132], they are generally able to physically overpower women. Therefore, women are concerned that men may physically or sexually act against their will [79]. The research found, that in the online dating context, women's safety concern is justified. Women are endangered by gendered violence, online sexual aggression, and online abuse [109, 215].

For male online dating users, a major concern is that female users look differently on their online dating profile pictures than offline. The misrepresentation concern of males may be related to the "sex roles" in romantic dating. Eaton and Rose found that the sex roles in heterosexual relationships maintained stable over the past years [88]. According to them, cultural scripts specify dating preferences and dating rules. Men are looking for physical attractiveness in female dating partners. Women are taught to take care of their appearance when being successful in dating. In contrast, women value financially successful male dating partners. Men have learned from the cultural scripts that being successful in their jobs increases their attractiveness. The findings of Eaton and Rose are supported by the ones of Abramova et al. for online dating [2]. They found that female online dating users are more tolerant towards the appearance of male dating users, while male dating users have exact body type preferences.

With increasing cases of date rape in the past, the safety of women in online dating has been discussed broadly in the news [47]. Online dating applications like Tinder, Hinge, and Bumble have heard the call for safety features. In their terms of service, they request users to affirm that they have not been convicted of or waived a felony or violent crime, including sex crimes. Users can report other users if they

violate the terms of service. Furthermore, online dating applications provide a photo verification process, which reassures that users look alike in their profile pictures. In addition, online dating applications have partly included a “Safety and Policy Center”. These blog entries or sections within the application teach users how to behave in situations that may endanger their safety. These features can be evaluated as a basic foundation for informing users and trusting in their honesty to act accordingly. Yet, most often software features are missing that actively support users when they are in a questionable situation, such as the panic button (see Chapter 13.4).

In terms of the misrepresentation concern of male online dating users, some online dating applications like chat&yamo² have included blog entries to inform about and explain the intention behind misrepresentations [323]. However, other software features are yet missing. The photo verification process mentioned before does not check on misrepresentations but verifies the physical identity when a user signs up. The “appearance verifier” proposed in Chapter 13.4 would be a complementary software feature that actively supports users in their misrepresentation concerns during online dating usage.

In conclusion, online dating applications can still better address the trust concerns of female and male users. TrustSoFt may serve as a software development method to do so. Future work may analyse whether current software features really reduce the safety and misrepresentation concerns of female and male online dating users and what software features may pose better solution approaches. However, it seems as if service providers are not dependent on investing in applications that mitigate users’ safety and misrepresentation concerns to guarantee business success. People use online dating even if their concerns are not addressed. For service providers, addressing these user concerns might involve adaptations to the business strategy or their service. For these adaptations, it might be necessary to involve organisations in the offline sphere, such as the police. It is conceivable, that from a business point of view, the financial investment is not profitable for online dating service providers. They may not increase their user base by the safety or misrepresentation features, nor might they increase their profit.

Future Work

²www.chat-yamo.com

- Future work may analyse how far current software features really address the safety and misrepresentation concerns of female and male online dating users. TrustSoFt can be applied to elicit software features that propose solution approaches for the users' concerns.

15.3 Theoretical and Practical Implications

This dissertation provides both theoretical and practical implications. The theoretical implications are the trustworthiness framework for CMI and the overview of trustworthiness facets. The trustworthiness framework for CMI points out how social media applications and CMI applications can reflect the trustworthiness of users, service providers, and the application so that users can build trust in these parties. As mentioned in Chapter 15.2, the trustworthiness framework for CMI can be considered a theoretical model for software development. It can first serve software engineers as guidance for software development. Second, it can be used as a theoretical model for the validation of trust-related software features. With the framework, the relation of trust-related software features with trustworthiness facets and with trust in users, the service provider, and the application can be validated.

Another theoretical implication is the overview of trustworthiness facets. It provides knowledge of what characteristics users, service providers, and software applications can have that positively impact their trustworthiness. Furthermore, software engineers can use the overview as a database for selecting appropriate trustworthiness facets for software development. Thereby, software engineers can develop software by which users are supported in their trustworthiness assessment. As a critical point for ethics, software engineers can misuse the overview of trustworthiness facts to let users, service providers, or applications be perceived as trustworthy although this might not be the case.

Both the trustworthiness framework for CMI and the overview of trustworthiness facets serve as a theoretical basis for the method of eliciting trust-related software features - TrustSoFt. TrustSoFt is a practical implication of this dissertation. With TrustSoFt, software engineers are guided step by step to develop social media applications that support users in their trustworthiness assessment of involved parties. Furthermore, TrustSoFt provides practical implications on how i* goal modeling can

be used for specifying software goals, requirements, and features. Moreover, TrustSoFt and the adapted i* goal modelling notation implicate how conflicting goals and requirements can be identified and managed.

In the process of developing software applications that support users in their trustworthiness assessment, this dissertation provides two additional practical implications. These are the guideline for selecting trustworthiness facets and the method for establishing feature models for trustworthiness assessment. The guideline for selecting trustworthiness facets supports software engineers to decide what trustworthiness facet a user, service provider, or application needs to reflect in a given scenario. Thereby, users can assess whether the other parties are trustworthy or not. The guideline is a tool by which software engineers can actively enable trust-building processes of users via the software application. It can be regarded as an interdisciplinary guideline that bridges offline psychological processes to the online sphere.

The method for establishing feature models for trustworthiness assessment picks up TrustSoFt to specify trust-related software features in more detail. It enables software engineers to plan software features in regard to their impact on the perceived trustworthiness of users, the service provider, and the application. By the method for establishing feature models for trustworthiness assessment, software engineers can establish and validate feature models. Furthermore, they can configure software product lines that provide users with a unique trust-building user experience. The method for establishing feature models for trustworthiness assessment enables software engineers to elaborate on the effect features have on users and their online relationships with other parties. The method embeds psychological processes into software development.

16

Conclusion

This dissertation aims to support social media users in their online trustworthiness assessment of other users, the service provider, and the software application. To achieve this research objective, this work introduces the TrustSoFt method by which software engineers specify trust-related software features that consider users' underlying psychological processes of trust building. TrustSoFt has been applied to the context of hybrid social media and online dating. The applications indicate that resulting TrustSoFt features mitigate users' trust concerns, enable the assessment of the trustworthiness of other parties, and can provide innovative solution approaches for addressed concerns.

To elaborate on these insights, multiple research steps have been carried out. First, a theoretical basis was established by the trustworthiness framework for CMI. Through literature research, trust research was placed in the context of social media and CMI. The framework illustrates that trust in involved parties is impacted when users evaluate software features that reflect trustworthiness facets. The trustworthiness framework of CMI is theoretically underpinned by the overview of trustworthiness facets. A literature review revealed scientifically-based trustworthiness facets for individuals, organisations, and technology. Together with the guideline of selecting appropriate trustworthiness facets, the overview of trustworthiness facets serves software engineers as a tool for specifying trust-related software features.

Both the guideline and the overview can be used during applying TrustSoFt. TrustSoFt has been introduced as a conceptual model, first. It is established based on existing requirements elicitation methods. Yet, it contains the psychological trust elements to transfer the trustworthiness assessment to the online sphere. In the following parts of the dissertation, TrustSoFt was enriched by further elements to facilitate and enhance its application. Risk has been included in TrustSoFt as a

determinant to manage conflicting TrustSoFt elements that inhibit the simultaneous implementation of software features. Furthermore, the i^* goal modelling notation has been adapted to TrustSoFt to enable a model-based approach to obtain a better visualisation and discussion base of TrustSoFt within software development teams.

TrustSoFt provides first ideas for trust-related software features that address the research objective of supporting users in their online trustworthiness assessment. To specify trust-related software features in more detail, the feature modelling notation has been adapted to the trust context. Feature modelling is essential for the method of establishing feature models for trustworthiness assessment. The method guides software engineers in specifying trust-related software features by feature assets. Moreover, established feature models can document the validation of features in terms of related trustworthiness facets, which in turn supports the configuration of trust-related software features and software product lines. In line with the feature models for trustworthiness assessment is the catalogue for trust-related software features. Each feature asset receives an entry in the catalogue with its asset information. The catalogue plays a major role when practitioners may search for feature assets with certain characteristics, such as a specific trustworthiness facet. Thereby, practitioners can configure specific trust-related product lines.

Besides the theoretical basis and the practical approaches for software engineers, this dissertation contains evaluations and use cases. Academic software development teams have applied TrustSoFt for planning the development of sharing economy and online dating applications. Their qualitative feedback was used to enhance TrustSoFt. Moreover, TrustSoFt was applied for developing a hybrid social media application. The resulting trust-related software features have been quantitatively tested for their impact on users. Indeed, the resulting features reduced the information privacy concerns of users and increased their trusting beliefs in the application. A main finding was that the choice of addressed trust concern in TrustSoFt is highly relevant to the extent of the features' impact on users. Last but not least, TrustSoFt was applied to the qualitatively identified trust concerns of female and male online dating users. The resulting trust-related software features demonstrate that for each trust concern, an innovative solution can be specified.

This dissertation provides methodological solutions for software developers to develop applications that support users in their online trustworthiness assessment. Yet, it is recommended that future work conducts further studies on the relationship

between single trust-related software features and targeted trustworthiness facets. Such studies would strengthen the trustworthiness framework for CMI in its validity. Thereby, the framework could become an accepted theoretical model for trust-related software development. Another recommendation for future work is the development of a digital tool for TrustSoFt and the feature models for trustworthiness assessment. By a digital tool, the application of the feature models would be facilitated for practitioners. Practitioners could be supported in automatically creating TrustSoFt goal models. Moreover, a digital tool for feature models could implement the catalogue of trust-related software features to support the configuration of software product lines with the models.

In total, this dissertation provides theoretical implications with the trustworthiness framework for CMI and the overview of trustworthiness facets. They support software development that considers the psychological processes of trust building. For software development itself, this work contributes practical implications by various methods. With TrustSoFt, the guideline for selecting appropriate trustworthiness facets, i* goal modelling, and the method for establishing feature models for trustworthiness assessment, software engineers can specify trust-related software features that mitigate users' trust concerns and support them in their online trustworthiness assessment.

Bibliography

- [1] Ieee guide for software requirements specifications. *IEEE Std 830-1984*, pages 1–26, 1984.
- [2] Olga Abramova, Annika Baumann, Hanna Krasnova, and Peter Buxmann. Gender differences in online dating: What do we know so far? a systematic literature review. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 3858–3867, 2016.
- [3] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Manya Sleeper, et al. Nudges for privacy and security: Understanding and assisting users’ choices online. *ACM Computing Surveys (CSUR)*, 50(3):1–41, 2017.
- [4] Leona S Aiken, Stephen G West, and Raymond R Reno. *Multiple regression: Testing and interpreting interactions*. sage, 1991.
- [5] Esma Aïmeur and David Schönfeld. The ultimate invasion of privacy: Identity theft. In *2011 Ninth Annual International Conference on Privacy, Security and Trust*, pages 24–31, 2011.
- [6] Anna Akhmedova, Neus Vila-Brunet, and Marta Mas-Machuca. Building trust in sharing economy platforms: trust antecedents and their configurations. *Internet Research*, 31(4):1463–1490, 2021.
- [7] Majed Alkhamees, Saleh Alsaleem, Muhammad Al-Qurishi, Majed Al-Rubaian, and Amir Hussain. User trustworthiness in online social networks: A systematic review. *Applied Soft Computing*, 103:107159, 2021.
- [8] Steven Alter. A workaround design system for anticipating, designing, and/or preventing workarounds. In *Enterprise, business-process and information systems modeling*, pages 489–498. Springer, 2015.
- [9] James C Anderson and David W Gerbing. Structural equation modeling in practice: A review and recommended two-step approach. *Psychological bulletin*, 103(3):411, 1988.

- [10] Ana I Anton. *Goal identification and refinement in the specification of software-based information systems*. Georgia Institute of Technology, 1997.
- [11] Sven Apel and Christian Kästner. An overview of feature-oriented software development. *J. Object Technol.*, 8(5):49–84, 2009.
- [12] Jonathan Arnowitz, Michael Arent, and Nevin Berger. *Effective prototyping for software makers*. Elsevier, 2010.
- [13] Yudistira Asnar and Paolo Giorgini. Modelling risk and identifying countermeasure in organisations. In *Critical Information Infrastructures Security: First International Workshop, CRITIS 2006, Samos, Greece, August 31-September 1, 2006. Revised Papers 1*, pages 55–66, 2006.
- [14] Simon Attfield, Gabriella Kazai, Mounia Lalmas, and Benjamin Piwowarski. Towards a science of user engagement (position paper). In *WSDM workshop on user modelling for Web applications*, volume 1, 2011.
- [15] Robert Axelrod. Die evolution der Kooperation. In *Die Evolution der Kooperation*. Oldenbourg Wissenschaftsverlag, 2014.
- [16] Annette Baier. Trust and antitrust. *ethics*, 96(2):231–260, 1986.
- [17] K Suzanne Barber and Joonoo Kim. Belief revision process based on trust: Agents evaluating reputation of information sources. In *Trust in Cyber-societies*, pages 73–82. Springer, 2001.
- [18] Yakov Bart, Venkatesh Shankar, Fareena Sultan, and Glen L Urban. Are the drivers and role of online trust the same for all web sites and consumers? a large-scale exploratory empirical study. *Journal of marketing*, 69(4):133–152, 2005.
- [19] Youssef Bassil. A simulation model for the waterfall software development life cycle. *International Journal of Engineering and Technology*, 2(5), 2012.
- [20] Nancy K Baym. *Personal connections in the digital age*. John Wiley & Sons, 2015.
- [21] Dr David Beer. Social network (ing) sites... revisiting the story so far: A response to danah boyd & nicole ellison. *Journal of computer-mediated communication*, 13(2):516–529, 2008.

- [22] Avner Ben-Ner and Freyr Halldorsson. Trusting and trustworthiness: What are they, how to measure them, and what affects them. *Journal of Economic Psychology*, 31(1):64–79, 2010.
- [23] David Benavides, Sergio Segura, and Antonio Ruiz-Cortés. Automated analysis of feature models 20 years later: A literature review. *Information systems*, 35(6):615–636, 2010.
- [24] David Benavides, Sergio Segura, Pablo Trinidad, and Antonio Ruiz Cortés. Fama: Tooling a framework for the automated analysis of feature models. *VaMoS*, 2007:01, 2007.
- [25] Izak Benbasat and Weiquan Wang. Trust in and adoption of online recommendation agents. *Journal of the association for information systems*, 6(3):4, 2005.
- [26] Yoella Bereby-Meyer and Shaul Shalvi. Deliberate honesty. *Current Opinion in Psychology*, 6:195–198, 2015.
- [27] Neville F Bews and Gedeon J Rossouw. A role for business ethics in facilitating trustworthiness. *Journal of Business Ethics*, 39:377–390, 2002.
- [28] Keith J Blois. Trust in business to business relationships: An evaluation of its status. *Journal of management studies*, 36(2):197–215, 1999.
- [29] Susanne Bodker. Scenarios in user-centred design-setting the stage for reflection and action. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*, pages 11–pp, 1999.
- [30] Barry Boehm. Software risk management. In *ESEC’89: 2nd European Software Engineering Conference University of Warwick, Coventry, UK September 11–15, 1989 Proceedings*, pages 1–19, 2005.
- [31] Barry W. Boehm and Hoh Peter In. Identifying quality-requirement conflicts. *Proceedings of the Second International Conference on Requirements Engineering*, pages 218–, 1996.
- [32] Kenneth Duncan Boness, Marc Bartsch, Stephen Cook, and Rachel Harrison. A practical approach to goal modelling for time-constrained projects. *Ist International Workshop on Requirements Engineering of Information Systems*

- for the Digital Economy (REISDE), 2nd ICETE International Conference on e-Business and Telecommunications, 2005.*
- [33] Jean-François Bonnefon, Astrid Hopfensitz, and Wim De Neys. The modular nature of trustworthiness detection. *Journal of Experimental Psychology: General*, 142(1):143, 2013.
- [34] Angela Borchert, Elija Cassidy, and Maritta Heisel. Safety First? Gender Differences in Online Dating Behavior and Trust Concerns. Submitted for publication, 2023.
- [35] Angela Borchert, Nicolás Emilio Díaz Ferreyra, and Maritta Heisel. Balancing trust and privacy in computer-mediated introduction: featuring risk as a determinant for trustworthiness requirements elicitation. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–10, 2020.
- [36] Angela Borchert, Nicolás Emilio Díaz Ferreyra, and Maritta Heisel. Building trustworthiness in computer-mediated introduction: A facet-oriented framework. In *International Conference on Social Media and Society*, pages 39–46, 2020.
- [37] Angela Borchert, Nicolás E Díaz Ferreyra, and Maritta Heisel. Meeting strangers online: Feature models for trustworthiness assessment. In *Human-Centered Software Engineering: 9th IFIP WG 13.2 International Working Conference, HCSE 2022, Eindhoven, The Netherlands, August 24–26, 2022, Proceedings*, pages 3–22. Springer, 2022.
- [38] Angela Borchert, Nicolás Emilio Díaz Ferreyra, and Maritta Heisel. A conceptual method for eliciting trust-related software features for computer-mediated introduction. In *ENASE*, pages 269–280, 2020.
- [39] Angela Borchert and Maritta Heisel. Conflict identification and resolution for trust-related requirements elicitation a goal modeling approach. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 12(1):111–131, 2021.
- [40] Angela Borchert and Maritta Heisel. The role of trustworthiness facets for developing social media applications: A structured literature review. *Information*, 13(1):34, 2022.

- [41] Angela Borchert, Aidmar Wainakh, Nicole Krämer, Max Mühlhäuser, and Maritta Heisel. Mitigating privacy concerns by developing trust-related software features for a hybrid social media application. In *ENASE*, pages 269–280, 2021.
- [42] Angela Borchert, Aidmar Wainakh, Nicole Krämer, Max Mühlhäuser, and Maritta Heisel. The relevance of privacy concerns, trust, and risk for hybrid social media. In *Evaluation of Novel Approaches to Software Engineering: 16th International Conference, ENASE 2021, Virtual Event, April 26-27, 2021, Revised Selected Papers*, pages 88–111. Springer, 2022.
- [43] Paolo Bresciani, Anna Perini, Paolo Giorgini, Fausto Giunchiglia, and John Mylopoulos. Tropos: An agent-oriented software development methodology. *Autonomous Agents and Multi-Agent Systems*, 8:203–236, 2004.
- [44] Erica J Briscoe, D Scott Appling, and Heather Hayes. Cues to deception in social media communications. In *2014 47th Hawaii international conference on system sciences*, pages 1435–1443, 2014.
- [45] Jo Bryce and James Fraser. The role of disclosure of personal information in the evaluation of risk and trust in young peoples’ online interactions. *Computers in Human Behavior*, 30:299–306, 2014.
- [46] Tom Buchanan and Monica T Whitty. The online dating romance scam: causes and consequences of victimhood. *Psychology, Crime & Law*, 20(3):261–283, 2014.
- [47] S Burga. If you’re looking for love online, here’s what to know about dating app safety, 2023.
- [48] Jennifer L Burke, Matthew S Prewett, Ashley A Gray, Liuquin Yang, Frederick RB Stilson, Michael D Coovert, Linda R Elliot, and Elizabeth Redden. Comparing the effects of visual-auditory and visual-tactile feedback on user performance: a meta-analysis. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 108–117, 2006.
- [49] John K Butler Jr. Toward understanding and measuring conditions of trust: Evolution of a conditions of trust inventory. *Journal of management*, 17(3):643–663, 1991.

- [50] Oliver B Büttner and Anja S Göritz. Perceived trustworthiness of online shops. *Journal of Consumer Behaviour: An International Research Review*, 7(1):35–50, 2008.
- [51] Radka Bužgová and Kateřina Ivanová. Violation of ethical principles in institutional care for older people. *Nursing ethics*, 18(1):64–78, 2011.
- [52] John Buzzard. 2022 identity fraud study: The virtual battleground, 2022.
- [53] Jason Anthony Cain and Iveta Imre. Everybody wants some: Collection and control of personal information, privacy concerns, and social media use. *New Media & Society*, 24(12):2705–2724, 2022.
- [54] Cam Caldwell and Stephen E Clapham. organisational trustworthiness: An international perspective. *Journal of business ethics*, 47:349–364, 2003.
- [55] David G Carnevale. *Trustworthy government: Leadership and management strategies for building trust and high performance*. Jossey-Bass, 1995.
- [56] Juan Manuel Carrillo de Gea, Joaquín Nicolás, José L Fernández-Alemán, and Ambrosio Toval. Automated support for reuse-based requirements engineering in global software engineering. *Journal of Software: Evolution and Process*, 29(8):e1873, 2017.
- [57] Glenn R Carroll and Dennis Ray Wheaton. The organisational construction of authenticity: An examination of contemporary food and dining in the us. *Research in organisational behavior*, 29:255–282, 2009.
- [58] Abhijit Chakraborty, Mrinal Kanti Baowaly, Ashraful Arefin, and Ali Newaz Bahar. The role of requirement engineering in software development life cycle. *Journal of emerging trends in computing and information sciences*, 3(5), 2012.
- [59] Robert J Chapman. The effectiveness of working group risk identification and assessment techniques. *International Journal of Project Management*, 16(6):333–343, 1998.
- [60] Prakash K Chathoth, Brenda Mak, Janet Sim, Vinnie Jauhari, and Kamal Manaktola. Assessing dimensions of organisational trust across cultures: A comparative analysis of us and indian full service hotels. *International Journal of hospitality management*, 30(2):233–242, 2011.

- [61] Atanu Chaudhuri, Bhaba Krishna Mohanty, and Kashi Naresh Singh. Supply chain risk assessment during new product development: a group decision making approach using numeric and linguistic data. *International Journal of Production Research*, 51(10):2790–2804, 2013.
- [62] Aihui Chen and Yaobin Lu. Protective behavior in ride-sharing through the lens of protection motivation theory and usage situation theory. *International Journal of Information Management*, 61:102402, 2021.
- [63] Changfeng Chen. Identifying significant factors influencing consumer trust in an online travel site. *Information Technology & Tourism*, 8(3-4):197–214, 2006.
- [64] Jiachi Chen, Xin Xia, David Lo, John Grundy, and Xiaohu Yang. Maintenance-related concerns for post-deployed ethereum smart contract development: issues, techniques, and future challenges. *Empirical Software Engineering*, 26(6):117, 2021.
- [65] Shyi-Ming Chen and Jim-Ho Chen. Fuzzy risk analysis based on ranking generalized fuzzy numbers with different heights and different spreads. *Expert systems with applications*, 36(3):6833–6842, 2009.
- [66] Dan S Chiaburu and Audrey S Lim. Manager trustworthiness or interactional justice? predicting organisational citizenship behaviors. *Journal of business ethics*, 83:453–467, 2008.
- [67] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546, 2005.
- [68] Camille Cobb and Tadayoshi Kohno. How public is my private life? privacy in online dating. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1231–1240, 2017.
- [69] Mike Cohn. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [70] Jason A Colquitt, Jerald Greenberg, and Cindy P Zapata-Phelan. What is organisational justice? a historical overview. 2005.

- [71] Jason A Colquitt, Brent A Scott, and Jeffery A LePine. Trust, trustworthiness, and trust propensity: a meta-analytic test of their unique relationships with risk taking and job performance. *Journal of applied psychology*, 92(4):909, 2007.
- [72] Brian J Corbitt, Theerasak Thanasankit, and Han Yi. Trust and e-commerce: a study of consumer perceptions. *Electronic commerce research and applications*, 2(3):203–215, 2003.
- [73] Danielle Couch and Pranee Liamputtong. Online dating and mating: Perceptions of risk and health among online users. *Health, Risk & Society*, 9(3):275–294, 2007.
- [74] Danielle Couch, Pranee Liamputtong, and Marian Pitts. What are the real and perceived risks and dangers of online dating? perspectives from online daters: Health risks in the media. *Health, Risk & Society*, 14(7-8):697–714, 2012.
- [75] Lawrence A Crosby, Kenneth R Evans, and Deborah Cowles. Relationship quality in services selling: an interpersonal influence perspective. *Journal of marketing*, 54(3):68–81, 1990.
- [76] Frank B Cross. Law and trust. *Geo. LJ*, 93:1457, 2004.
- [77] Benjamin MP Cuff, Sarah J Brown, Laura Taylor, and Douglas J Howat. Empathy: A review of the concept. *Emotion review*, 8(2):144–153, 2016.
- [78] Krzysztof Czarnecki, Simon Helsen, and Ulrich Eisenecker. Staged configuration using feature models. In *International conference on software product lines*, pages 266–283, 2004.
- [79] Debra J Davidson and Wiluam R Freudenburg. Gender and environmental risk concerns: A review and analysis of available research. *Environment and behavior*, 28(3):302–339, 1996.
- [80] Elena Delgado-Ballester and José Luis Munuera-Alemán. Does brand trust matter to brand equity? *Journal of product & brand management*, 2005.
- [81] Elena Delgado-Ballester, Jose Luis Munuera-Aleman, and Maria Jesus Yague-Guillen. Development and validation of a brand trust scale. *International journal of market research*, 45(1):35–54, 2003.

- [82] George Demiris, Debra Parker Oliver, and Karla T Washington. Defining and analyzing the problem. *Behavioral intervention research in hospice and palliative care: Building an evidence base*, pages 27–39, 2019.
- [83] Folker Den Braber, Ida Hogganvik, M Soldal Lund, Ketik Stølen, and Fredrik Vraalsen. Model-based security analysis in seven steps—a guided tour to the coras method. *BT Technology Journal*, 25(1):101, 2007.
- [84] Shuai Ding, Shan-Lin Yang, and Chao Fu. A novel evidential reasoning based method for software trustworthiness evaluation under the uncertain and unreliable environment. *Expert Systems with Applications*, 39(3):2700–2709, 2012.
- [85] Rachel Dinh, Patrick Gildersleve, Chris Blex, and Taha Yasseri. Computational courtship understanding the evolution of online dating through large-scale data analysis. *Journal of Computational Social Science*, 5(1):401–426, 2022.
- [86] Kurt T Dirks. The effects of interpersonal trust on work group performance. *Journal of applied psychology*, 84(3):445, 1999.
- [87] Patricia M Doney and Joseph P Cannon. An examination of the nature of trust in buyer–seller relationships. *Journal of marketing*, 61(2):35–51, 1997.
- [88] Asia Anna Eaton and Suzanna Rose. Has dating become more egalitarian? a 35 year review using sex roles. *Sex roles*, 64:843–862, 2011.
- [89] Leslie Edwards and Leslie J Edwards. *Practical risk management in the construction industry*. Thomas Telford, 1995.
- [90] Florian N Egger et al. Affective design of e-commerce user interfaces: How to maximise perceived trustworthiness. In *Proc. Intl. Conf. Affective Human Factors Design*, pages 317–324, 2001.
- [91] Alan C Elliott and Wayne A Woodward. *Statistical analysis quick reference guidebook: With SPSS examples*. Sage, 2007.
- [92] Nicole Ellison, Rebecca Heino, and Jennifer Gibbs. Managing impressions online: Self-presentation processes in the online dating environment. *Journal of computer-mediated communication*, 11(2):415–441, 2006.

- [93] Gunn Enli and Linda Therese Rosenberg. Trust in the age of social media: Populist politicians seem more authentic. *Social media+ society*, 4(1):2056305118764430, 2018.
- [94] Nir Eyal. *Hooked: How to build habit-forming products*. Penguin, 2014.
- [95] Nina Ferreri and Christopher B Mayhorn. Examining frustration and performance when priming user expectations and providing a technology malfunction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 64, pages 1846–1850, 2020.
- [96] Donald L Ferrin, Michelle C Bligh, and Jeffrey C Kohles. It takes two to tango: An interdependence analysis of the spiraling of perceived trustworthiness and cooperation in interpersonal and intergroup relationships. *organisational Behavior and Human Decision Processes*, 107(2):161–178, 2008.
- [97] Donald L Ferrin, Kurt T Dirks, and Pri P Shah. Direct and indirect effects of third-party relationships on interpersonal trust. *Journal of applied psychology*, 91(4):870, 2006.
- [98] Gerald R Ferris, Fred R Blass, Ceasar Douglas, Robert W Kolodinsky, and Darren C Treadway. Personal reputation in organisations. 2003.
- [99] Andy Field and Graham Hole. *How to design and report experiments*. Sage, 2002.
- [100] Emma Fletcher. Social media is a gold mine for scammers in 2021, 2022.
- [101] Brian J Fogg. A behavior model for persuasive design. In *Proceedings of the 4th international Conference on Persuasive Technology*, pages 1–7, 2009.
- [102] Brian J Fogg and Hsiang Tseng. The elements of computer credibility. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 80–87, 1999.
- [103] David Freeman, Sakshi Jain, Markus Dürmuth, Battista Biggio, and Giorgio Giacinto. Who are you? a statistical approach to measuring user authenticity. In *NDSS*, volume 16, pages 21–24, 2016.
- [104] Batya Friedman, Peter H Khan Jr, and Daniel C Howe. Trust online. *Communications of the ACM*, 43(12):34–40, 2000.

- [105] John J Gabarro. The development of trust, influence and expectations. *Interpersonal behavior: Communication and understanding in relationships*, pages 290–303, 1978.
- [106] Günter Gans, Matthias Jarke, Stefanie Kethers, and Gerhard Lakemeyer. Continuous requirements management for organisation networks: a (dis) trust-based approach. *Requirements Engineering*, 8:4–22, 2003.
- [107] David Gefen. E-commerce: the role of familiarity and trust. *Omega*, 28(6):725–737, 2000.
- [108] Kim Giffin. The contribution of studies of source credibility to a theory of interpersonal trust in the communication process. *Psychological bulletin*, 68(2):104, 1967.
- [109] Rosalie Gillett. “this is not a nice safe space”: investigating women’s safety work on tinder. *Feminist Media Studies*, pages 1–17, 2021.
- [110] Martin Glinz. On non-functional requirements. In *15th IEEE international requirements engineering conference (RE 2007)*, pages 21–26, 2007.
- [111] Thomas Goergen. Stress, conflict, elder abuse and neglect in german nursing homes: A pilot study among professional caregivers. *Journal of Elder Abuse & Neglect*, 13(1):1–26, 2001.
- [112] Avi Goldfarb and Catherine Tucker. Shifts in privacy concerns. *American Economic Review*, 102(3):349–353, 2012.
- [113] Banu Golesorkhi. Gender differences and similarities in judgments of trustworthiness. *Women in Management Review*, 21(3):195–210, 2006.
- [114] Katelyn Golladay and Kristy Holtfreter. The consequences of identity theft victimization: An examination of emotional and physical health outcomes. *Victims & Offenders*, 12(5):741–760, 2017.
- [115] David Good. Individuals, interpersonal relations, and trust. *Trust: Making and breaking cooperative relations*, pages 31–48, 2000.
- [116] Jason Good and Ann Blandford. Incorporating human factors concerns into the design and safety engineering of complex control systems. In *1999 International Conference on Human Interfaces in Control Rooms, Cockpits and Command Centres*, pages 51–56, 1999.

- [117] Kalman Graffi, Sergey Podrajanski, Patrick Mukherjee, Aleksandra Kovacevic, and Ralf Steinmetz. A distributed platform for multimedia communities. In *2008 Tenth IEEE International Symposium on Multimedia*, pages 208–213, 2008.
- [118] Colin M Gray, Cristiana Santos, Nataliia Bielova, Michael Toth, and Damian Clifford. Dark patterns and the legal requirements of consent banners: An interaction criticism perspective. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2021.
- [119] Stephan G Grimmelikhuijsen and Albert J Meijer. Effects of transparency on the perceived trustworthiness of a government organisation: Evidence from an online experiment. *Journal of Public Administration Research and Theory*, 24(1):137–157, 2014.
- [120] Huimin Gu, Tingting Zhang, Can Lu, and Xiaoxiao Song. Assessing trust and risk perceptions in the sharing economy: An empirical study. *Journal of Management Studies*, 58(4):1002–1032, 2021.
- [121] Ying Guan and Aditya K Ghose. Use constraint hierarchy for non-functional requirements analysis. In *Web Engineering: 5th International Conference, ICWE 2005, Sydney, Australia, July 27-29, 2005. Proceedings 5*, pages 104–109, 2005.
- [122] The Guardian. Hackers reportedly leak email addresses of more than 200 million twitter users, 2023.
- [123] Per E Gustafsson. Gender differences in risk perception: Theoretical and methodological perspectives. *Risk analysis*, 18(6):805–811, 1998.
- [124] Jeffrey A Hall, Namkee Park, Hayeon Song, and Michael J Cody. Strategic misrepresentation in online dating: The effects of gender, self-monitoring, and personality traits. *Journal of Social and Personal Relationships*, 27(1):117–135, 2010.
- [125] Lei Hang and Do-Hyeun Kim. Sla-based sharing economy service with smart contract for resource integrity in the internet of things. *Applied Sciences*, 9(17):3602, 2019.

- [126] Gabriella M Harari and Samuel D Gosling. Concerns about facebook among users and abstainers: Relationships with individual differences and facebook use. *Translational Issues in Psychological Science*, 2(3):261, 2016.
- [127] Allison G Harvey, Edward Watkins, and Warren Mansell. *Cognitive behavioural processes across psychological disorders: A transdiagnostic approach to research and treatment*. Oxford University Press, USA, 2004.
- [128] Andrew F Hayes. *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. Guilford publications, 2017.
- [129] Milena M Head and Khaled Hassanein. Trust in e-commerce: Evaluating the impact of third-party seals. *Quarterly Journal of electronic commerce*, 3:307–326, 2002.
- [130] Wannes Heirman, Michel Walrave, and Koen Ponnet. Predicting adolescents’ disclosure of personal information in exchange for commercial incentives: An application of an extended theory of planned behavior. *Cyberpsychology, Behavior, and Social Networking*, 16(2):81–87, 2013.
- [131] Kaitlin Henderson and Alejandro Salado. Value and benefits of model-based systems engineering (mbse): Evidence from the literature. *Systems Engineering*, 24(1):51–66, 2021.
- [132] Vivian H Heyward, Sandra M Johannes-Ellis, and Jacki F Romer. Gender differences in strength. *Research quarterly for exercise and sport*, 57(2):154–159, 1986.
- [133] Scott Highhouse, Margaret E Brooks, and Gary Gregarus. An organisational impression management perspective on the formation of corporate reputations. *Journal of management*, 35(6):1481–1493, 2009.
- [134] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, Vinay Phegade, and Juan Del Cuvillo. Using innovative instructions to create trustworthy software solutions. *HASP@ ISCA*, 11(10.1145):2487726–2488370, 2013.
- [135] Donna L Hoffman, Thomas P Novak, and Marcos Peralta. Building consumer trust online. *Communications of the ACM*, 42(4):80–85, 1999.
- [136] Brian C Holtz. Trust primacy: A model of the reciprocal relations between trust and perceived justice. *Journal of Management*, 39(7):1891–1923, 2013.

- [137] Michelle Hood and Amanda L Duffy. Understanding the relationship between cyber-victimisation and cyber-bullying on social network sites: The role of moderating factors. *Personality and Individual Differences*, 133:103–108, 2018.
- [138] Jennifer Horkoff and Eric Yu. Analyzing goal models: different approaches and how to choose among them. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 675–682, 2011.
- [139] Larue Tone Hosmer. Trust: The connecting link between organisational theory and philosophical ethics. *Academy of management Review*, 20(2):379–403, 1995.
- [140] Idris Hsi and Colin Potts. Studying the evolution and enhancement of software features. In *icsm*, page 143, 2000.
- [141] Li-tze Hu and Peter M Bentler. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural equation modeling: a multidisciplinary journal*, 6(1):1–55, 1999.
- [142] Youngjin Hur, Yong Jae Ko, and Cathryn L Claussen. Acceptance of sports websites: A conceptual model. *International Journal of Sports Marketing and Sponsorship*, 2011.
- [143] Severina Iankova, Iain Davies, Chris Archer-Brown, Ben Marder, and Amy Yau. A comparison of social media marketing between b2b, b2c and mixed business models. *Industrial Marketing Management*, 81:169–179, 2019.
- [144] Jim Isaak and Mina J Hanna. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, 51(8):56–59, 2018.
- [145] Anil Isisag, Craig Thompson, Delphine Dion, Markus Giesler, Ashlee Humphreys, Gregory Carpenter, Nicholas Pendarvis, and Marius Luedicke. Contemporary investigations into the relational understanding of branding. *ACR North American Advances*, 2021.
- [146] Risk management - Principles and guidelines. Standard, International organisation for Standardization, 2018.
- [147] Software and systems engineering — Reference model for product line engineering and management. Standard, International organisation for Standardization, December 2015.

- [148] Sandy Jap, Diana C Robertson, and Ryan Hamilton. The dark side of rapport: Agent misbehavior face-to-face and online. *Management Science*, 57(9):1610–1622, 2011.
- [149] Sirkka L Jarvenpaa, Noam Tractinsky, and Lauri Saarinen. Consumer trust in an internet store: A cross-cultural validation. *Journal of Computer-Mediated Communication*, 5(2):JCMC526, 1999.
- [150] Eugene E Jennings et al. Routes to the executive suite. 1971.
- [151] Crystal X Jiang, Roy YJ Chua, Masaaki Kotabe, and Janet Y Murray. Effects of cultural ethnicity, firm size, and firm age on senior executives’ trust in their overseas business partners: Evidence from china. *Journal of International Business Studies*, 42:1150–1173, 2011.
- [152] Zhenhui Jiang, Cheng Suang Heng, and Ben CF Choi. Research note—privacy concerns and privacy-protective behavior in synchronous online social interactions. *Information Systems Research*, 24(3):579–595, 2013.
- [153] Zhi Jin. *Environment modeling-based requirements engineering for software intensive systems*. Morgan Kaufmann, 2017.
- [154] David W Johnson and Roger T Johnson. *Cooperation and competition: Theory and research*. Interaction Book Company, 1989.
- [155] Rupert Jones. Cybercrime now becoming a serious problem for many britons, 2014.
- [156] Stephen L Jones and Priti Pradhan Shah. Diagnosing the locus of trust: A temporal perspective for trustor, trustee, and dyadic influences on perceived trustworthiness. *Journal of Applied Psychology*, 101(3):392, 2016.
- [157] Paul E Jose. *Doing statistical mediation and moderation*. Guilford Press, 2013.
- [158] Mohsen Jozani, Emmanuel Ayaburi, Myung Ko, and Kim-Kwang Raymond Choo. Privacy concerns and benefits of engagement with social media-enabled apps: A privacy calculus perspective. *Computers in Human Behavior*, 107:106260, 2020.
- [159] Reynol Junco. Comparing actual and self-reported measures of facebook use. *Computers in Human Behavior*, 29(3):626–631, 2013.

- [160] Haruhiko Kaiya, Hisayuki Horai, and Motoshi Saeki. Agora: Attributed goal-oriented requirements analysis method. In *Proceedings IEEE joint international conference on requirements engineering*, pages 13–22, 2002.
- [161] Gurvinder Kalra and Dinesh Bhugra. Sexual violence against women: Understanding cross-cultural intersections. *Indian journal of psychiatry*, 55(3):244–249, 2013.
- [162] Kyo C Kang, Sholom G Cohen, James A Hess, William E Novak, and A Spencer Peterson. Feature-oriented domain analysis (foda) feasibility study. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1990.
- [163] Elahe Kani-Zabihi and Martin Helmhout. Increasing service users’ privacy awareness by introducing on-line interactive privacy features. In *Information Security Technology for Applications: 16th Nordic Conference on Secure IT Systems, NordSec 2011, Tallinn, Estonia, October 26-28, 2011, Revised Selected Papers 16*, pages 131–148, 2012.
- [164] Jean-Noel Kapferer. *The new strategic brand management: Creating and sustaining brand equity long term*. Kogan Page Publishers, 2008.
- [165] Angel Wong An Kee and Rashad Yazdanifard. The review of the ugly truth and negative aspects of online dating. *Global Journal of Management and Business Research*, 15(E4):31–36, 2015.
- [166] Aharon Kellerman. *The internet as second action space*. Routledge, 2014.
- [167] Ankit Kesharwani and Shailendra Singh Bisht. The impact of trust and perceived risk on internet banking adoption in india: An extension of technology acceptance model. *International journal of bank marketing*, 30(4):303–322, 2012.
- [168] Esther Keymolén. *Trust on the line: a philosophical exploration of trust in the networked era*. 2016.
- [169] Muhammad Taimoor Khan, Mehr Durrani, Armughan Ali, Irum Inayat, Shehzad Khalid, and Kamran Habib Khan. Sentiment analysis and the complex natural language. *Complex Adaptive Systems Modeling*, 4:1–19, 2016.

- [170] Anil Khurana and Jyoti Mehra. Trust concern in electronic banking: A literature review. *International Journal of Management, IT and Engineering*, 6(1):103–118, 2016.
- [171] Angella J Kim and Kim KP Johnson. Power of consumers using social media: Examining the influences of brand-related user-generated content on facebook. *Computers in human behavior*, 58:98–108, 2016.
- [172] Kyung Kyu Kim and Bipin Prabhakar. Initial trust and the adoption of b2c e-commerce: The case of internet banking. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 35(2):50–64, 2004.
- [173] Min-Ju Kim, Mohamed Shehab, Hyo-Cheol Lee, and Seok-Won Lee. Trust-aware goal modeling from use case for cooperative self-adaptive systems. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 4405–4410, 2018.
- [174] David Kipnis. Trust and technology. *Trust in organisations: Frontiers of theory and research*, 39:50, 1996.
- [175] Barbara Kitchenham, Lesley Pickard, and Shari Lawrence Pfleeger. Case studies for method and tool evaluation. *IEEE software*, 12(4):52–62, 1995.
- [176] Anthony C Klotz, Serge P da Motta Veiga, M Ronald Buckley, and Mark B Gavin. The role of trustworthiness in recruitment and selection: A review and guide for future research. *Journal of organisational Behavior*, 34(S1):S104–S119, 2013.
- [177] Amy J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3):1–44, 2011.
- [178] Colleen M Koch. To catch a catfish: a statutory solution for victims of online impersonation. *U. Colo. L. Rev.*, 88:233, 2017.
- [179] Marios Koufaris and William Hampton-Sosa. The development of initial trust in an online company by new customers. *Information & management*, 41(3):377–397, 2004.

- [180] Iga Kozłowska. Facebook and data privacy in the age of cambridge analytica. *The Henry M. Jackson School of International Studies*, page 1, 2018.
- [181] Christian Kraft. *User experience innovation: User centered design that works*. Apress, 2012.
- [182] Tobias Kroll and Stefan Stieglitz. Digital nudging and privacy: improving decisions about self-disclosure in social networks. *Behaviour & Information Technology*, 40(1):1–19, 2021.
- [183] Katharina Krombholz, Dieter Merkl, and Edgar Weippl. Fake identities in social media: A case study on the sustainability of the facebook business model. *Journal of Service Science Research*, 4(2):175–212, 2012.
- [184] Adrienne D Kunkel, Steven R Wilson, James Olufowote, and Scott Robson. Identity implications of influence goals: Initiating, intensifying, and ending romantic relationships. *Western Journal of Communication (includes Communication Reports)*, 67(4):382–412, 2003.
- [185] Ari Kusyanti, Dita Rahma Puspitasari, Harin Puspa Ayu Catherina, and Yustiyana April Lia Sari. Information privacy concerns on teens as facebook users in indonesia. *Procedia Computer Science*, 124:632–638, 2017.
- [186] Sari A. Laakso. *User interface design patterns*, 2003.
- [187] Carl Lagoze. Big data, data integrity, and the fracturing of the control zone. *Big Data & Society*, 1(2):2053951714558281, 2014.
- [188] Yu Beng Leau, Wooi Khong Loo, Wai Yip Tham, and Soo Fun Tan. Software development life cycle agile vs traditional approaches. In *International Conference on Information and Network Technology*, volume 37, pages 162–167, 2012.
- [189] Kwanwoo Lee, Kyo C Kang, and Jaejoon Lee. Concepts and guidelines of feature modeling for product line software engineering. In *International Conference on Software Reuse*, pages 62–77, 2002.
- [190] Dean Leffingwell and Don Widrig. *Managing software requirements: a unified approach*. Addison-Wesley Professional, 2000.

- [191] Sirpa Leppänen, Janus Spindler Møller, Thomas Rørbeck Nørreby, Andreas Stæhr, and Samu Kytölä. Authenticity, normativity and social media. *Discourse, Context and Media*, 8(June), 2015.
- [192] Roy J Lewicki, Daniel J McAllister, and Robert J Bies. Trust and distrust: New relationships and realities. *Academy of management Review*, 23(3):438–458, 1998.
- [193] Roy J Lewicki and Carolyn Wiethoff. Trust, trust development, and trust repair. *The handbook of conflict resolution: Theory and practice*, 1(1):86–107, 2000.
- [194] J David Lewis and Andrew Weigert. Trust as a social reality. *Social forces*, 63(4):967–985, 1985.
- [195] Yuan Li. Empirical studies on online information privacy concerns: Literature review and an integrative framework. *Communications of the Association for Information Systems*, 28(1):28, 2011.
- [196] Yuan Li. A multi-level model of individual information privacy beliefs. *Electronic Commerce Research and Applications*, 13(1):32–44, 2014.
- [197] Mengqi Liao and S Shyam Sundar. Sound of silence: Does muting notifications reduce phone use? *Computers in Human Behavior*, 134:107338, 2022.
- [198] Christian Licoppe. Liquidity and attachment in the mobile hookup culture. a comparative study of contrasted interactional patterns in the main uses of grindr and tinder. *Journal of Cultural Economy*, 13(1):73–90, 2020.
- [199] Shi-Woei Lin and Yu-Cheng Liu. The effects of motivations, trust, and privacy concern in social networking. *Service Business*, 6:411–424, 2012.
- [200] Xiaodan Liu, Chunhui Yuan, Muhammad Hafeez, and Ch Muhammad Nadeem Faisal. Digital trust mediated by the platform in the sharing economy from a consumer perspective. In *Proceedings of the Fourteenth International Conference on Management Science and Engineering Management: Volume 1*, pages 670–684, 2020.
- [201] David López Jiménez, Eduardo Carlos Dittmar, and Jenny Patricia Vargas Portillo. New directions in corporate social responsibility and ethics: codes

- of conduct in the digital environment. *Journal of Business Ethics*, pages 1–11, 2021.
- [202] Garm Lucassen, Fabiano Dalpiaz, Jan Martijn EM van der Werf, and Sjaak Brinkkemper. The use and effectiveness of user stories in practice. In *Requirements Engineering: Foundation for Software Quality: 22nd International Working Conference, REFSQ 2016, Gothenburg, Sweden, March 14-17, 2016, Proceedings 22*, pages 205–222, 2016.
- [203] Niklas Luhmann. *Trust and power*. John Wiley & Sons, 2018.
- [204] Xiao Ma, Jeffrey T Hancock, Kenneth Lim Mingjie, and Mor Naaman. Self-disclosure and perceived trustworthiness of airbnb host profiles. In *Proceedings of the 2017 ACM conference on computer supported cooperative work and social computing*, pages 2397–2409, 2017.
- [205] Ole Lehrmann Madsen and Birger Moller-Pedersen. Virtual classes: A powerful mechanism in object-oriented programming. In *Conference proceedings on Object-oriented programming systems, languages and applications*, pages 397–406, 1989.
- [206] Walid Magdy, Yehia Elkhatib, Gareth Tyson, Sagar Joglekar, and Nishanth Sastry. Fake it till you make it: Fishing for catfishes. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 497–504, 2017.
- [207] Dewi Mairiza, Didar Zowghi, and Nurie Nurmuliani. Managing conflicts among non-functional requirements. In *Australian Workshop on Requirements Engineering*, 2009.
- [208] Naresh K Malhotra, Sung S Kim, and James Agarwal. Internet users’ information privacy concerns (iuipe): The construct, the scale, and a causal model. *Information systems research*, 15(4):336–355, 2004.
- [209] Mike Mannion and Juha Savolainen. Aligning business and technical strategies for software product lines. In *International Conference on Software Product Lines*, pages 406–419, 2010.
- [210] Christian Märtin, Bärbel Christine Bissinger, and Pietro Asta. Optimizing the digital customer journey—improving user experience by exploiting emotions,

- personas and situations for individualized user interface adaptations. *Journal of Consumer Behaviour*, 2021.
- [211] Cristian Martinez, Nicolás Díaz, Silvio Gonnet, and Horacio Leone. A petri net variability model for software product lines. *Electronic Journal of SADIO (EJS)*, 13:35–53, 2014.
- [212] John D Mayer and Peter Salovey. The intelligence of emotional intelligence, 1993.
- [213] Roger C Mayer, James H Davis, and F David Schoorman. An integrative model of organisational trust. *Academy of management review*, 20(3):709–734, 1995.
- [214] Daniel J McAllister. Affect-and cognition-based trust as foundations for interpersonal cooperation in organisations. *Academy of management journal*, 38(1):24–59, 1995.
- [215] Jill McCartney and Susan Hellier. Match, chat, mate: A narrative analysis of online dating and sexual experiences among women. *The Journal for Nurse Practitioners*, 17(4):394–398, 2021.
- [216] Siné JP Mcdougall, Martin B Curry, and Oscar De Bruijn. Measuring symbol and icon characteristics: Norms for concreteness, complexity, meaningfulness, familiarity, and semantic distance for 239 symbols. *Behavior Research Methods, Instruments, & Computers*, 31(3):487–519, 1999.
- [217] D Harrison McKnight, Michelle Carter, Jason Bennett Thatcher, and Paul F Clay. Trust in a specific technology: An investigation of its components and measures. *ACM Transactions on management information systems (TMIS)*, 2(2):1–25, 2011.
- [218] D Harrison McKnight and Norman L Chervany. What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology. *International journal of electronic commerce*, 6(2):35–59, 2001.
- [219] D Harrison McKnight and Norman L Chervany. Reflections on an initial trust-building model. *Handbook of trust research*, 29, 2006.
- [220] D Harrison McKnight, Larry L Cummings, and Norman L Chervany. Initial trust formation in new organisational relationships. *Academy of Management review*, 23(3):473–490, 1998.

- [221] Nádia Medeiros, Naghmeh Ramezani Ivaki, Pedro Nunes Da Costa, and Marco Paulo Amorim Vieira. Towards an approach for trustworthiness assessment of software as a service. In *2017 IEEE International Conference on Edge Computing (EDGE)*, pages 220–223, 2017.
- [222] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113, 2014.
- [223] Jacqueline Meredith. *Perceived emotional competence and emotion appraisal skills in middle childhood in typically developing and behaviourally challenged children*. PhD thesis, Middlesex University, 2009.
- [224] Thomas V Merluzzi and Cheryl S Brischetto. Breach of confidentiality and perceived trustworthiness of counselors. *Journal of Counseling Psychology*, 30(2):245, 1983.
- [225] Christian Meske and Tobias Potthoff. The dinu-model—a process model for the design of nudges. 2017.
- [226] AK Mishra. organisational responses to crisis. trust in organisations. *Frontiers of theory and research*, 3(5):261–287, 1996.
- [227] Tim Mitchel. Identity fraud up by 27% in first quarter of 2015, 2015.
- [228] Nazila Gol Mohammadi, Torsten Bandyszak, Sachar Paulus, Per Håkon Meland, Thorsten Weyer, and Klaus Pohl. Extending software development methodologies to support trustworthiness-by-design. In *CAiSE Forum*, pages 213–220, 2015.
- [229] Nazila Gol Mohammadi and Maritta Heisel. A framework for systematic analysis and modeling of trustworthiness requirements using i* and bpmn. In *International Conference on Trust and Privacy in Digital Business*, pages 3–18, 2016.
- [230] Nazila Gol Mohammadi, Sachar Paulus, Mohamed Bishr, Andreas Metzger, Holger Koennecke, Sandro Hartenstein, and Klaus Pohl. An analysis of software quality attributes and their contribution to trustworthiness. In *CLOSER*, pages 542–552, 2013.

- [231] Nazila Gol Mohammadi, Nelufar Ulfat-Bunyadi, and Maritta Heisel. Problem-based derivation of trustworthiness requirements from users' trust concerns. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–10, 2018.
- [232] Abdullahi Mohamud Sharif and Shuib Basri. Software risk assessment: a review on small and medium software projects. In *International Conference on Software Engineering and Computer Systems*, pages 214–224, 2011.
- [233] David Moher, Alessandro Liberati, Jennifer Tetzlaff, Douglas G Altman, and PRISMA Group*. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *Annals of internal medicine*, 151(4):264–269, 2009.
- [234] Guido Møllering. *Trust: Reason, routine, reflexivity*. Emerald Group Publishing, 2006.
- [235] Guido Møllering. Inviting or avoiding deception through trust? conceptual exploration of an ambivalent relationship. 2008.
- [236] Kjetil Moløkken and Magne Jørgensen. Software effort estimation: unstructured group discussion as a method to reduce individual bias. In *PPIG*, page 4, 2003.
- [237] David L Mothersbaugh, William K Foxx, Sharon E Beatty, and Sijun Wang. Disclosure antecedents in an online service context: The role of sensitivity of information. *Journal of service research*, 15(1):76–98, 2012.
- [238] Robert Münscher, Max Vetter, and Thomas Scheuerle. A review and taxonomy of choice architecture techniques. *Journal of Behavioral Decision Making*, 29(5):511–524, 2016.
- [239] George E Newman and Rosanna K Smith. Kinds of authenticity. *Philosophy Compass*, 11(10):609–618, 2016.
- [240] Philip J Nickel. Trust in technological systems. *Norms in technology*, pages 223–237, 2013.
- [241] D Noyes. Distribution of twitter users worldwide as of january 2021, by gender, 2021.

- [242] Borke Obada-Obieh, Sonia Chiasson, and Anil Somayaji. “don’t break my heart!”: User security strategies for online dating. In *Workshop on Usable Security (USEC)*, 2017.
- [243] Borke Obada-Obieh and Anil Somayaji. Can i believe you? establishing trust in computer mediated introductions. In *Proceedings of the 2017 New Security Paradigms Workshop*, pages 94–106, 2017.
- [244] Dustin Ormond and Jordan Barlow. Security warning messages research: Past and future. *MWAIS 2022 Proceedings*, 7, 2022.
- [245] Mert Ozkaya. Do the informal & formal software modeling notations satisfy practitioners for software architecture modeling? *Information and Software Technology*, 95:15–33, 2018.
- [246] Narasimha Paravastu. Dimensions of technology trustworthiness and technology trust modes. *Encyclopedia of Information Science and Technology, Third Edition*, pages 4301–4309, 2015.
- [247] Elisabeth Paté-Cornell. Risk and uncertainty analysis in government safety decisions. *Risk analysis*, 22(3):633–646, 2002.
- [248] LeeAnn Perkins, Janet E Miller, Ali Hashemi, and Gary Burns. Designing for human-centered systems: Situational risk as a factor of trust in automation. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 54, pages 2130–2134, 2010.
- [249] Sonja Peteranderl, Julia Jaroschewski, and Thomas Oberfranz. Date rape - schützen tinder co. frauen bei sexuellen Übergriffen?, 2022.
- [250] Klaus Pohl, Günter Böckle, and Frank Van Der Linden. *Software product line engineering*, volume 10. Springer, 2005.
- [251] Jeffrey G Proudfoot, David Wilson, Joseph S Valacich, and Michael D Byrd. Saving face on facebook: Privacy concerns, social benefits, and impression management. *Behaviour & Information Technology*, 37(1):16–37, 2018.
- [252] Urszula Pruchniewska. “i like that it’s my choice a couple different times”: Gender, affordances, and user experience on bumble dating. *International Journal of Communication*, 14:18, 2020.

- [253] Giulia Ranzini and Christoph Lutz. Love at first swipe? explaining tinder self-presentation and motives. *Mobile Media & Communication*, 5(1):80–101, 2017.
- [254] Rupak Rauniar, Greg Rawski, Ben Johnson, and Jie Yang. Social media user satisfaction—theory development and research findings. *Journal of Internet Commerce*, 12(2):195–224, 2013.
- [255] Rim Razzouk and Valerie Shute. What is design thinking and why is it important? *Review of educational research*, 82(3):330–348, 2012.
- [256] Yacine Rezgui and Adam Marks. Information security awareness in higher education: An exploratory study. *Computers & security*, 27(7-8):241–253, 2008.
- [257] Eric Ries. Minimum viable product: a guide. *Startup lessons learned*, 3:1, 2009.
- [258] Ronald E Riggio, Joan Tucker, and David Coffaro. Social skills and empathy. *Personality and individual differences*, 10(1):93–99, 1989.
- [259] Dejan Ristić. A tool for risk assessment. *safety Engineering*, 3:121–127, 2013.
- [260] Blaine G Robbins. What is trust? a multidisciplinary review, critique, and synthesis. *Sociology compass*, 10(10):972–986, 2016.
- [261] Lionel P Robert, Alan R Denis, and Yu-Ting Caisy Hung. Individual swift trust and knowledge-based trust in face-to-face and virtual team members. *Journal of management information systems*, 26(2):241–279, 2009.
- [262] Karlene H Roberts and Charles A O’Reilly. Measuring organisational communication. *Journal of applied psychology*, 59(3):321, 1974.
- [263] Dieter Rombach. Integrated software process and product lines. In *Software Process Workshop*, pages 83–90, 2005.
- [264] Julian B Rotter. Generalized expectancies for interpersonal trust. *American psychologist*, 26(5):443, 1971.
- [265] Julian B Rotter. Interpersonal trust, trustworthiness, and gullibility. *American psychologist*, 35(1):1, 1980.

- [266] William D Rowe. Understanding uncertainty. *Risk analysis*, 14(5):743–750, 1994.
- [267] Nayan B Ruparelia. Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*, 35(3):8–13, 2010.
- [268] Thomas L Saaty. How to make a decision: the analytic hierarchy process. *European journal of operational research*, 48(1):9–26, 1990.
- [269] Sangeeta Sahney, Koustab Ghosh, and Archana Shrivastava. Conceptualizing consumer “trust” in online buying behaviour: An empirical inquiry and model development in indian context. *Journal of Asia Business Studies*, 2013.
- [270] Khalid Samhale. The impact of trust in the internet of things for health on user engagement. *Digital Business*, 2(1):100021, 2022.
- [271] Nicolás Sánchez-Gómez, Jesus Torres-Valderrama, Julián Alberto García-García, Javier J Gutiérrez, and MJ Escalona. Model-based software design and testing in blockchain smart contracts: A systematic literature review. *IEEE Access*, 8:164556–164569, 2020.
- [272] George Saridakis, Vladlena Benson, Jean-Noel Ezingard, and Hemamali Tennakoon. Individual information security, user behaviour and cyber victimisation: An empirical study of social networking users. *Technological Forecasting and Social Change*, 102:320–330, 2016.
- [273] Margrit Schreier et al. Varianten qualitativer inhaltsanalyse: ein wegweiser im dickicht der begrifflichkeiten. In *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, volume 15, 2014.
- [274] Nev Schulman. *In real life: Love, lies & identity in the digital age*. Hachette UK, 2014.
- [275] Robert Schumacher. *The handbook of global user research*. Morgan Kaufmann, 2009.
- [276] Ahmed Seffah, Jan Gulliksen, and Michel C Desmarais. An introduction to human-centered software engineering. In *Human-centered software engineering—integrating usability in the software development lifecycle*, pages 3–14. Springer, 2005.

- [277] Ben Shneiderman. Designing trust into online experiences. *Communications of the ACM*, 43(12):57–59, 2000.
- [278] Mariah Simmons and Joon Suk Lee. Catfishing: A look into online dating and impersonation. In *International Conference on Human-Computer Interaction*, pages 349–358, 2020.
- [279] Aaron Smith and Monica Anderson. 5 facts about online dating. *Fact Tank*, 29, 2016.
- [280] H Jeff Smith, Sandra J Milberg, and Sandra J Burke. Information privacy: Measuring individuals’ concerns about organisational practices. *MIS quarterly*, pages 167–196, 1996.
- [281] Peter K Smith, Fran Thompson, and Julia Davidson. Cyber safety for adolescent girls: Bullying, harassment, sexting, pornography, and solicitation. *Current opinion in obstetrics and gynecology*, 26(5):360–365, 2014.
- [282] Carmel Sofer, Ron Dotsch, Daniel HJ Wigboldus, and Alexander Todorov. What is typical is good: The influence of face typicality on perceived trustworthiness. *Psychological Science*, 26(1):39–47, 2015.
- [283] Daniel J Solove. Understanding privacy. 2008.
- [284] Reka Solymosi, Kate Bowers, and Taku Fujiyama. Mapping fear of crime as a context-dependent everyday experience that varies in space and time. *Legal and Criminological Psychology*, 20(2):193–211, 2015.
- [285] Jai-Yeol Son and Sung S Kim. Internet users’ information privacy-protective responses: A taxonomy and a nomological model. *MIS quarterly*, pages 503–529, 2008.
- [286] Andreas Sonderegger and Juergen Sauer. The influence of design aesthetics in usability testing: Effects on user performance and perceived usability. *Applied ergonomics*, 41(3):403–410, 2010.
- [287] Tariq Soussan and Marcello Trovati. Social media data misuse. In *Advances in Intelligent Networking and Collaborative Systems: The 13th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2021) 13*, pages 183–189, 2022.

- [288] OMG Available Specification. Omg unified modeling language (omg uml), superstructure, v2. 1.2. *Object Management Group*, 70, 2007.
- [289] Statista. Numbers of social network users in the united states from 2018 to 2027, 2022.
- [290] Angus Stevenson. *Oxford Dictionary of English* -. OUP Oxford, New York, London, 2010.
- [291] Maria Stoicescu. Social impact of online dating platforms. a case study on tinder. In *2020 19th RoEduNet conference: Networking in education and research (RoEduNet)*, pages 1–6, 2020.
- [292] Eugene F Stone, Hal G Gueutal, Donald G Gardner, and Stephen McClure. A field experiment comparing information-privacy values, beliefs, and attitudes across several types of organisations. *Journal of applied psychology*, 68(3):459, 1983.
- [293] Norazah Mohd Suki. A structural model of customer satisfaction and trust in vendors involved in mobile commerce. *International Journal of Business Science & Applied Management (IJBSAM)*, 6(2):18–30, 2011.
- [294] Cass R Sunstein. Nudging: a very short guide. *Journal of Consumer Policy*, 37(4):583–588, 2014.
- [295] Monika Taddicken and Cornelia Jers. The uses of privacy online: trading a loss of privacy for social web gratifications? In *Privacy online*, pages 143–156. Springer, 2011.
- [296] Chee Wee Tan, Izak Benbasat, and Ronald T Cenfetelli. Building citizen trust towards e-government services: do high quality websites matter? In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pages 217–217, 2008.
- [297] Teun Terpstra. Emotions, trust, and perceived risk: Affective and cognitive routes to flood preparedness behavior. *Risk Analysis: An International Journal*, 31(10):1658–1675, 2011.
- [298] Richard H Thaler and Cass R Sunstein. *Nudge: Wie man kluge Entscheidungen anstößt*. Ullstein eBooks, 2009.

- [299] Melina A Throuvala, Mark D Griffiths, Mike Rennoldson, and Daria J Kuss. A ‘control model’ of social media engagement in adolescence: A grounded theory analysis. *International journal of environmental research and public health*, 16(23):4696, 2019.
- [300] Stephanie Tom Tong and Joseph B Walther. Just say “no thanks”: Romantic rejection in computer-mediated communication. *Journal of Social and Personal Relationships*, 28(4):488–506, 2011.
- [301] Sabine Trepte, Leonard Reinecke, Nicole B Ellison, Oliver Quiring, Mike Z Yao, and Marc Ziegele. A cross-cultural perspective on the privacy calculus. *Social Media+ Society*, 3(1):2056305116688035, 2017.
- [302] Shawn Tseng and BJ Fogg. Credibility and computing technology. *Communications of the ACM*, 42(5):39–44, 1999.
- [303] Glenn T Tsunokai, Allison R McGrath, and Jillian K Kavanagh. Online dating preferences of asian americans. *Journal of Social and Personal Relationships*, 31(6):796–814, 2014.
- [304] Amos Tversky and Craig R Fox. Weighing risk and uncertainty. *Psychological review*, 102(2):269, 1995.
- [305] RHTW Aachen University. i star wiki, 2011.
- [306] Craig Van Slyke, Christie L Comunale, and France Belanger. Gender differences in perceptions of web-based shopping. *Communications of the ACM*, 45(8):82–86, 2002.
- [307] Martijn van Welie. User interface design patterns, 2008.
- [308] Thiago Viana, Andrea Zisman, and Arosha K. Bandara. Identifying conflicting requirements in systems of systems. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, pages 436–441, 2017.
- [309] Christian Voigt, Stephan Schlögl, and Aleksander Groth. Dark patterns in online shopping: Of sneaky tricks, perceived annoyance and respective brand trust. In *HCI in Business, Government and organisations: 8th International Conference, HCIBGO 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings*, pages 143–155, 2021.

- [310] Thomas von der Maßen and Horst Lichter. Requiline: A requirements engineering tool for software product lines. In *Software Product-Family Engineering: 5th International Workshop, PFE 2003, Siena, Italy, November 4-6, 2003. Revised Papers 5*, pages 168–180, 2004.
- [311] Aidmar Wainakh, Tim Grube, Jorg Daubert, Carsten Porth, and Max Muhlhauser. Tweet beyond the cage: a hybrid solution for the privacy dilemma in online social networks. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2019.
- [312] Edward Shih-Tse Wang and Ruenn-Lien Lin. Perceived quality factors of location-based apps on trust, perceived privacy risk, and continuous usage intention. *Behaviour & Information Technology*, 36(1):2–10, 2017.
- [313] Monica T Whitty and Tom Buchanan. The online romance scam: A serious cybercrime. *CyberPsychology, Behavior, and Social Networking*, 15(3):181–183, 2012.
- [314] Brenda K Wiederhold. How covid has changed online dating—and what lies ahead, 2021.
- [315] Karl Wiegers and Joy Beatty. *Software requirements*. Pearson Education, 2013.
- [316] Consuelo H Wilkins. Effective engagement requires trust and being trustworthy. *Medical Care*, 56(10 Suppl 1):S6, 2018.
- [317] Michele Williams. Building genuine trust through interpersonal emotion management: A threat regulation model of trust and collaboration across boundaries. *Academy of management Review*, 32(2):595–621, 2007.
- [318] Philip H Winne. Students’ calibration of knowledge and learning processes: Implications for designing powerful software learning environments. *International Journal of Educational Research*, 41(6):466–488, 2004.
- [319] Roman Wirtz, Maritta Heisel, Angela Borchert, Rene Meis, Aida Omerovic, and Ketil Stølen. Risk-based elicitation of security requirements according to the iso 27005 standard. In *Evaluation of Novel Approaches to Software Engineering: 13th International Conference, ENASE 2018, Funchal, Madeira, Portugal, March 23–24, 2018, Revised Selected Papers 8*, pages 71–97, 2019.

- [320] Lisa Wood. Brands and brand equity: definition and management. *Management decision*, 38(9):662–669, 2000.
- [321] Bo Xie. Using the internet for offline relationship formation. *Social Science Computer Review*, 25(3):396–404, 2007.
- [322] Heng Xu, Tamara Dinev, H Jeff Smith, and Paul Hart. Examining the formation of individual’s privacy concerns: Toward an integrative view. *ICIS 2008 Proceedings*, Paper 6, 2008.
- [323] Yamo. Why do some people misrepresent themselves on dating apps?, 2022.
- [324] Hongwei Chris Yang. Young american consumers’ prior negative experience of online disclosure, online privacy concerns, and privacy protection behavioral intent. *The Journal of Consumer Satisfaction, Dissatisfaction and Complaining Behavior*, 25:179–202, 2012.
- [325] Jisu Yi, Gao Yuan, and Changsok Yoo. The effect of the perceived risk on the adoption of the sharing economy in the tourism industry: The case of airbnb. *Information Processing & Management*, 57(1):102108, 2020.
- [326] Miso Yoon, Eunyoung Lee, Mikyoung Song, Byoungju Choi, et al. A test case prioritization through correlation of requirement and risk. *Journal of Software Engineering and Applications*, 5(10):823, 2012.
- [327] Eric Yu. Modeling strategic relationships for process reengineering. *Social Modeling for Requirements Engineering*, 11(2011):66–87, 2011.
- [328] Panayiotis Zaphiris and Chee Siang Ang. From online familiarity to offline trust: How a virtual community creates familiarity and trust between strangers. In *Social computing and virtual communities*, pages 195–220. Chapman and Hall/CRC, 2009.
- [329] Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. *ACM transactions on Software Engineering and Methodology (TOSEM)*, 6(1):1–30, 1997.
- [330] Xihui Zhang, Tao Hu, Hua Dai, and Xiang Li. Software development methodologies, trends, and implications. *Information Technology Journal*, 9(8):1747–1753, 2010.

- [331] Zheyang Zhang. Effective requirements development—a comparison of requirements elicitation techniques. *Software Quality Management XV: Software Quality in the Knowledge Society*, E. Berki, J. Nummenmaa, I. Sunley, M. Ross and G. Staples (Ed.) British Computer Society, pages 225–240, 2007.
- [332] Qi Zhou. A deterrence perspective on damages for fraudulent misrepresentation. *Journal of Interdisciplinary Economics*, 19(1):83–96, 2007.
- [333] Karolina Zurek. Food sharing in europe: Between regulating risks and the risks of regulating. *European Journal of Risk Regulation*, 7(4):675–687, 2016.

Appendix



Overview of Trustworthiness Facets for Individuals

Appendix A. Overview of Trustworthiness Facets for Individuals

Trustworthiness facets	Definition	References
Ability, competence, expertise, knowledge, skill, wisdom, business sense, influence, power	Skills or characteristics that enable to fulfill obligations or to have impact in a specific domain	[213, 113, 71] [156, 49, 105] [262, 87, 75] [108, 27, 226]
Accessibility, approachability, attentiveness, availability, openness, receptivity	Being physically present when needed, mentally open and receptive, easy to talk to and a careful listener	[218, 316, 49] [105, 87, 27] [226, 150]
attractiveness	Being appealing to others	[218, 108, 306]
Benevolence, availability, candor, care, loyalty, openness, receptivity, agreeableness, selflessness, honesty, altruism, goodwill	Having concerns about others, wanting something good for others and acting in their interest without an egocentric motive.	[213, 218, 230] [316, 105, 87] [108, 27, 226, 150] [176, 264]
Confidentiality, discreetness	Entrusted knowledge is kept in confidence	[49, 105, 282]
Emotional stability	"[B]eing calm, enthusiastic, free from anxiety, depression and insecurity" [49]	[108]
Empathy	The ability to comprehend feelings of others	[316]
Extraversion, dynamism	Talkativeness, sociability, friendliness	[108, 27]
Honesty, credibility, truthfulness, authenticity, openness, accuracy, willingness to disclose	Correctness of information and freely sharing information and ideas	[218, 316, 71] [49, 261, 87] [75, 150]
Humbleness	The notion to not take oneself more important than others	[96]
Integrity, fairness, consistency, reliability, discreetness, morality, ethicality, credibility, honesty	The trustee complies to the trustor's accepted principles (e.g., moral. ethical) that are predictable and reliable leading to equity	[325, 316, 113] [156, 49, 105] [75, 108, 27] [226, 176, 136]
Justice, fairness	The trustee morally respects the trustor's interests and the trustor herself - especially concerning provided information and interactions.	[49, 66, 98]

Trustworthiness facets	Definition	References
Likability, rapport	Friendliness, high sympathy and a person with whom the trustor wants to spend time together and cooperate	[87, 148, 72]
Predictability, consistency, reliability, good judgment, promise fulfillment, dependability, conscientiousness, performance	A stability in one's actions that is based on recurring behaviour, the ability to make good decisions, being productive and carrying out responsibilities reliably	[49, 105, 108] [27, 226, 150] [264, 214]
Popularity, social desirability	Social or cultural approval, socially desirable	[27, 264]
Reputation	The perceived identity of a trustee which reflects personality traits, behaviour or presented images that is based on the trustor's own observations over a period of time or on secondary sources.	[220, 224]
Respectfulness	The trustee regards "others and their perspective as valuable" [316]	[316]
Similarity, shared understanding, share of values	Perception of shared interests, values, appearance, lifestyle, status, or culture	[218, 113, 87] [75, 72, 151]

B

Overview of Trustworthiness Facets for Technology

Trustworthiness facets	Definition	References
Ability, competence, expertise, credibility	The system is believed to have the skills and expertise to perform and act effectively in specific domains and to fulfill its promised services and responsibilities. Based on that, the user accepts its advice and believes its output.	[217, 246, 200] [296, 72, 302] [25, 102, 90]
Benevolence, helpfulness goodwill	The system acts in the user's interests, cares for him/her, is well-intentioned and provides help or guidance when needed.	[217, 246, 200] [296, 72, 302]
Information quality, content quality data-related quality (consists of data integrity, data reliability data timeliness, data validity) , usefulness	The system provides sufficient information that is accurate, understandable, useful, complete, relevant and timely updated so that the user is able to evaluate the context (e.g., product, service, seller)	[217, 230, 206] [72, 293, 6]
Integrity, compliance, compatibility	The system complies with standards (e.g., industry specific standards) or regulations, adheres to the user's accepted ethical or moral codes and is compatible with his/her beliefs or values.	[217, 246, 230] [200, 296, 306] [25]
Non-Repudiation	Ability to prove to sender that data has been delivered and to prove to receiver the sender identity for an unambiguous data transmission.	[230]
Openness, transparency	The system provides how it works and complies with standards and regulations.	[230, 293]
Performance, reliability, predictability, dependability, functionality, accuracy, availability, fault tolerance, accountability, responsiveness, result demonstrability, correctness	The system executes correctly to accomplish the service that it promises. It is predictable despite potential failures and delivers proper outputs.	[217, 84, 246] [230, 200, 306] [72, 172, 269] [63, 120, 221]

Trustworthiness facets	Definition	References
Privacy, confidentiality	Privacy refers to the provision of information and the risk of its exposure to unintended parties. Systems, which respect their users' privacy, limit the access of the users' data to only authorized agents and enable users to take control of its usage.	[230, 206, 72] [90, 269, 120] [221]
Reputation, image, brand strength, visibility	On the one hand, the technology's recognition and how much it might enhance the user's social status. On the other hand, an "easy identification of the [associated] company and its activity sector" [120]	[306, 90, 63] [120, 119]
Safety	The system operates in a way that keeps its users' life and property safe and does not risk any harm or injuries.	[217, 230, 206]
Security, confidentiality	The system knows its users' vulnerabilities and protects them and their resources against attacks, misuses and unauthorized access	[84, 230, 206] [72, 90, 172] [269, 221, 18] [179]
Situational normality, social presence	The perception that the system is "normal, proper, or suited to a successful venture" as well as "personal, sociable, and [has] sensitive human elements, creating a feeling of human touch" [198].	[218, 246, 206]
Usability, comprehensibility, effectiveness, ease-of-use, efficiency	A system designed in a way that enables users to effortlessly use it with easy access to understandable information that supports users in the usage.	[246, 230, 296] [90, 293, 6] [269, 63, 179]
Website quality, completeness, perceived usefulness, web site design, interface design, likeability	On the one hand, the extent to which the implemented set of software features meet the needs of its users. On the other hand, an attractive graphical design in terms of structure, navigation, and content.	[217, 246, 230] [206, 72, 6] [269, 63, 179]

C

Overview of Trustworthiness Facets for Organisations

Trustworthiness facets	Definition	References
Ability, competence, financial balance, quality assurance	Knowledge and skills to provide the service or product promised by the organisation (while being both effective and efficient in regard to expended costs)	[220, 218, 54] [55, 226, 90]
Benevolence, concern, goodness, morality, caring, interactional courtesy, responsibility to inform	Respecting and showing respect to the interests of the consumers and not taking advantage of their vulnerability.	[220, 218, 54] [226]
Familiarity, similarity	Perception of the same values or interests	[49, 176, 72] [107]
Integrity, (procedural) fairness, justice, legal compliance, structural assurance	The existence of principles, values, standards or regulations (e.g., law, organisational policies, organisational procedures, contracts) to which an organisation corresponds as promised. This most often relates to a high quality of treatment and equity.	[220, 218, 54] [55, 49, 136] [98, 172, 60] [70]
Openness, honesty, transparency, confidential information sharing, responsibility to inform, comprehensibility	The availability, simplicity or clarity of information disclosed by an organisation that allows individuals to comprehend the performance or internal workings of that organisation.	[220, 54, 87] [27, 226, 165]
Performance	Current actions for providing a service or product, which may involve the delivery, relative costs and the performance of the service/product itself.	[87]
Reliability, credibility, consistency, dependability, responsibility, predictability	The organisation complies by its actions with its promises and offers guidance and support in times of crisis.	[220, 55, 226] [179, 60]

Trustworthiness facets	Definition	References
Reputation, prototypical organisational identity, brand image	Perception of an organisation's culture, attributes, beliefs, values, or prestige based on customer's own experience or hearsay from secondary sources.	[218, 87, 27] [176, 98, 293] [269, 63]
Responsiveness, interactivity	Being responsive to the customers' requests and providing rapid feedback	[54, 72, 293] [120]
Security	The organisation provides a comfortable, assured and safe feeling	[218]
Situational normality	The individual's belief of an organisation's success based on the perception how customary a situation with the organisation seems to be	[220, 179]
Size	The larger a company overall size and its market share position, the more experience it seems to have leading to a higher perception of trustworthiness.	[87, 269, 63] [179]
Willingness to customize, service customization	Specialized equipment or adaptation of production processes or services to meet the customer's needs.	[87, 293, 269] [63, 120, 179]

D

Materials of the TrustSoFt Evaluation

For the TrustSoFt evaluation, developer teams have first applied TrustSoFt and then evaluated it (see Chapter 7). In the following, the evaluation sheet is presented by which each developer was guided to state her/his opinion about each TrustSoFt step. Afterwards, exemplary interview questions are listed that the developers have generated for the semi-structured user interviews in order to identify trust concerns.

D.1 Evaluation Sheet for TrustSoFt

The product development projects were completed in German. Therefore, the evaluation sheet is also in German.

Feedback zur Methode „TrustSoFt“

In diesem Praxisprojekt wurde die Methode „TrustSoFt“ angewandt, um Anforderungen für eine Online Dating oder Sharing Economy Applikation zu ermitteln. Die einzelnen Schritte der Methode sind in Abbildung 1 abgebildet.

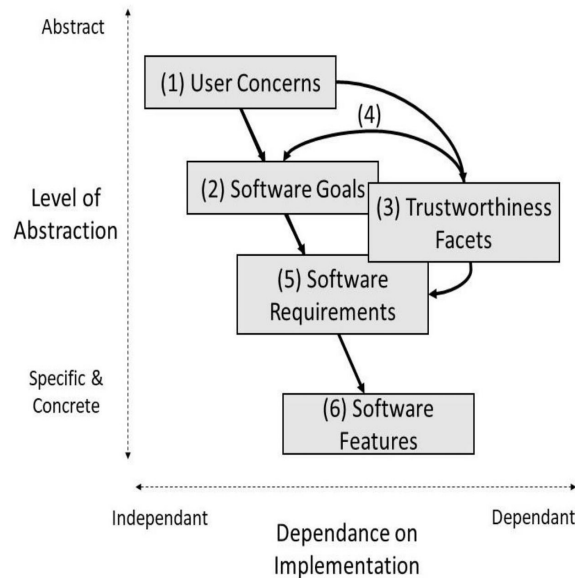


Abbildung 1: Methode TrustSoFt

In diesem Dokument soll zu den einzelnen Schritten der Methode und zur Methode selbst Feedback gegeben werden. Das Feedback wird für die Weiterentwicklung und Verbesserung der Methode verwendet. Bei dem Feedback soll es sich um eine **subjektive und kritische Stellungnahme** auf Basis der eigenen Erfahrungen handeln. Gerne kann das Feedback im Kontext und anhand von Beispielen der während des Projekts zu entwickelnden App getätigt werden. Das Feedback soll für jeden Methodenschritt und für die komplette Methode folgende Aspekte berücksichtigen:

- Was sind die **Vorteile** des Schritts/der Methode? Inwieweit war dieser Schritt/die Methode **hilfreich**? Was hat es euch gebracht?
- Welche **Nachteile** sind bei dem Schritt/der Methode aufgefallen? War etwas irrelevant oder zu vernachlässigen?
- Welche **Schwierigkeiten** gab es **bei der Umsetzung**?
- Inwiefern kann die Umsetzung weiter **unterstützt** werden, sodass sie leichter fällt oder strukturierter ist? Was für eine Art von Input oder Anleitung wird zusätzlich benötigt?
- Könnt ihr euch **alternative Umsetzungsmöglichkeiten** vorstellen, die euch bei der Umsetzung der Methode besser unterstützen können?
- Kann die Methode um weitere Schritte oder Aspekte **erweitert** werden? Welche weiteren Aspekte wären hilfreich für die Umsetzung gewesen?
- Habt ihr weitere Ideen oder Anmerkungen?

D.2 Exemplary Interview Questions by the Developers

The semi-structured interviews have been conducted in German. They are translated into English here.

- What is important to you when looking for a suitable online dating match?
- What are your concerns about this app?
- For which purpose would you use the app?
- How do other people in the social media context / online context give you the feeling of trust?
- What information are you willing to disclose in your profile that you would also like to see in the other profiles?
- What features should our app provide to ensure your personal safety?

E

Materials of the HSM user study

E.1 Internet Users' Information Privacy Concern scale & Concern for Information Privacy Scale

The scales are based on the work of Malhotra et al. and Smith [208, 280].

Here are some statements about personal information concerning HushTweet. With personal information, we refer to the information contained in tweets and likes published via HushTweet. From the standpoint of personal privacy, please indicate the extent to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. Privacy is really a matter of the right of HushTweet users to exercise control and autonomy over decisions about how their information is collected, used, and shared.
2. HushTweet discloses the way data are collected, processed and used.
3. HushTweet asks me for personal information.
4. All the personal information in the distributed databases used by HushTweet are double-checked for accuracy - no matter how much this costs.
5. HushTweet does not use personal information for any purpose unless it has been authorized by the individuals who provided information.
6. HushTweet devotes time and effort to preventing unauthorized access to the personal information.

7. In HushTweet, user control of personal information lies at the heart of user privacy.
8. HushTweet's privacy policy has a clear and conspicuous disclosure.
9. When HushTweet asks me for personal information, I sometimes think twice before providing it.
10. HushTweet makes sure that the personal information in their files is accurate.
11. When people give personal information to HushTweet for some reason, HushTweet does not use the information for any other reason.
12. Distributed databases that contain personal information are protected from unauthorized access by HushTweet - no matter what it costs.
13. My online privacy is invaded when control is lost or unwillingly reduced as a result of sharing personal information with HushTweet.
14. I am aware and knowledgeable about how my personal information is used by HushTweet.
15. I would give personal information to HushTweet.
16. HushTweet has procedures to correct errors in personal information.
17. HushTweet does not sell the personal information in the distributed databases to other companies.
18. HushTweet makes sure that unauthorized people cannot access personal information in the distributed databases.
19. HushTweet is collecting too much personal information about me.
20. HushTweet devotes time and effort to verifying the accuracy of the personal information in the distributed databases.
21. HushTweet does not share personal information with other companies unless it has been authorized by the individuals who provided the information.

E.2 Global Information Privacy Concern

The scale is introduced by Smith et al. [280].

Here are some statements about online privacy. Please indicate the extent to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. All things considered, the Internet would cause serious privacy problems.
2. Compared to others, I am more sensitive about the way online companies handle my personal information.
3. To me, it is the most important thing to keep my privacy intact from online companies.
4. I believe other people are too much concerned with online privacy issues.
5. Compared with other subjects on my mind, personal privacy is very important.
6. I am concerned about threats to my personal privacy today.

E.3 Trusting Beliefs

The scale is based on the work of Jarvenpaa et al. [149].

Here are some statements about HushTweet. Please indicate the extent to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. HushTweet is trustworthy in handling personal information.
2. HushTweet tells the truth and fulfill promises related to personal information provided by me.
3. I trust that HushTweet keeps my best interests in mind when dealing with personal information.

4. HushTweet is in general predictable and consistent regarding the usage of personal information.
5. HushTweet is always honest with customers when it comes to using personal information that I provide.

E.4 Risk Beliefs

The scale is based on the work of Jarvenpaa et al. [149].

Here are some statements about HushTweet. Please indicate the extent to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. In general, it is risky to give personal information to HushTweet.
2. There is high potential for loss associated with giving personal information to HushTweet.
3. There is too much uncertainty associated with giving personal information to HushTweet.
4. Providing HushTweet with personal information involves many unexpected problems.
5. I feel safe giving personal information to HushTweet.

E.5 Perceived Trustworthiness of HushTweet

The questionnaire is based on the scale of perceived trustworthiness for online shops [50].

Here are some statements about HushTweet. Please indicate the extend to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. HushTweet is very competent.
2. HushTweet is genuinely interested in its users' welfare.
3. I am happy with the standards by which HushTweet is operating.
4. HushTweet's methods of operation are unclear.
5. HushTweet is able to fully satisfy its users.
6. HushTweet puts users' interests first.
7. HushTweet operates scrupulously.
8. HushTweet keeps its promises.
9. One can expect good advice from HushTweet.
10. If problems arise, one can expect to be treated fairly by HushTweet.
11. You can believe the statements of HushTweet.
12. I would rely on advice from HushTweet.

E.6 Willingness to Use HushTweet

These statements concern your willingness to use HushTweet and Twitter. Please indicate the extent to which you, as an individual, agree or disagree with each statement.

Please tick your answer in the scale.

1. I am interested in using HushTweet.
2. I am willing to use HushTweet's private tweeting functionality.
3. I would rather use HushTweet's anonymous like than liking publicly on Twitter.
4. I prefer HushTweet over Twitter.
5. I would download HushTweet.

6. I am willing to use HushTweet anonymous liking functionality.
7. I would tell my friends about HushTweet.
8. I would rather use HushTweet's private tweet than tweeting publicly on Twitter.

F

The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Two excerpts of catalogues for trust-related software features are presented here. The first excerpt is for the safety check of users, which addressed the safety concern of female online dating users. The second excerpt is for the appearance verifier of users for the misrepresentation concern of male online dating users. The excerpts include the basic information and asset information that have been discussed in the feature model creation in Chapter 13.4 of the feature models in Figures 13.8 and 13.12 on pages 216 and 230.

F.1 The Catalogue for the Safety Check of Users

Basic Information	
Name	Safety check of users
Problem	Men pose a risk to women's safety due to their physical superiority
Keywords	Safety, physical violence, sexual assault, gender issue
Requirements	Check the safety criterion "promise fulfillment" for date terms, Elaborate on the safety criterion "respectfulness" of users in the chat, Ask male users for full body picture, Display results for safety criteria, ...
Problematic characteristics	Physical strength
Desired characteristics	Empathy, goodwill, respectfulness, promise fulfillment

Figure F.1: Basic information of the catalogue for trust-related software features for the concept feature "safety check of users".

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Date Check	
Feature Type	<input type="checkbox"/> Awareness <input type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Allocated:</u> Approachability, availability, openness, dynamism <u>Propagated:</u> Promise fulfillment
Trustworthiness facets for application	<u>Allocated:</u> Non-repudiation, (perceived) usefulness <u>Propagated:</u> (perceived) usefulness, confidentiality, ability <u>Optional:</u> Ability, functionality, (perceived) usefulness, safety, social presence
Trustworthiness facets for service provider	<u>Allocated:</u> Concern <u>Optional:</u> Reputation, caring

Figure F.2: Asset information for the feature asset “date check”.

Asset Information – Date terms share	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input checked="" type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	Ability, functionality, (perceived) usefulness, safety, social presence
Trustworthiness facets for service provider	-

Figure F.3: Asset information for the feature asset “date terms share”.

Asset Information – Share button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input checked="" type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.4: Asset information for the feature asset “share button”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Sharing algorithm	
User Accessibility	<input type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input checked="" type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	ability
Trustworthiness facets for service provider	-

Figure F.5: Asset information for the feature asset “sharing algorithm”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Date Page	
Feature Type	<input type="checkbox"/> Awareness <input type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Propagated:</u> Approachability, availability, openness, dynamism
Trustworthiness facets for application	<u>Propagated:</u> (perceived) usefulness, confidentiality <u>Optional:</u> Ability, functionality, (perceived) usefulness, safety, social presence, performance, safety, reputation
Trustworthiness facets for service provider	<u>Optional:</u> Reputation, caring

Figure F.6: Asset information for the feature asset “date page”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Panic button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input checked="" type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	<u>Allocated:</u> Safety, reputation <u>Propagated:</u> Performance
Trustworthiness facets for service provider	Reputation, caring

Figure F.7: Asset information for the feature asset “panic button”.

Asset Information – Police call	
User Accessibility	<input type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input checked="" type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	Performance
Trustworthiness facets for service provider	-

Figure F.8: Asset information for the feature asset “police call”.

Asset Information – Date request	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Approachability, availability, openness, dynamism
Trustworthiness facets for application	(perceived) usefulness
Trustworthiness facets for service provider	

Figure F.9: Asset information for the feature asset “date request”.

Asset Information – Input field date terms	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.10: Asset information for the feature asset “input field date terms”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Information date terms female/male user	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.11: Asset information for the feature asset “information date terms female/male user”.

Asset Information – Time	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.12: Asset information for the feature asset “time”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Location	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.13: Asset information for the feature asset “location”.

Asset Information – Button “Ask for date”	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.14: Asset information for the feature asset “button “Ask for date””.

Asset Information – Date invitation	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Approachability, availability, openness, dynamism
Trustworthiness facets for application	(perceived) usefulness
Trustworthiness facets for service provider	-

Figure F.15: Asset information for the feature asset “date invitation”.

Asset Information – Button Accept Date	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Availability, openness
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.16: Asset information for the feature asset “button accept date”.

Asset Information – Button Decline Date	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.17: Asset information for the feature asset “button decline date”.

Asset Information – Button on match page to date page	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.18: Asset information for the feature asset “button on match page to date page”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Feedback to the date	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	Confidentiality
Trustworthiness facets for service provider	-

Figure F.19: Asset information for the feature asset “feedback to the date”.

Asset Information – Questions about date	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.20: Asset information for the feature asset “questions about date”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Input field for feedback	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.21: Asset information for the feature asset “input field for feedback”.

Asset Information – Date terms feedback female/male user	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.22: Asset information for the feature asset “date term feedback female/-male user”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Algorithm promise fulfillment score	
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	Propagated: Promise fulfillment
Trustworthiness facets for application	Ability
Trustworthiness facets for service provider	-

Figure F.23: Asset information for the feature asset “algorithm promise fulfillment score”.

Asset Information – Promise Fulfillment Score	
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	Promise fulfillment
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.24: Asset information for the feature asset “promise fulfillment score”.

F.2 The Catalogue for the Appearance Verifier

Basic Information	
Name	Authenticity check
Problem	Female users look different on their online dating profile pictures than in reality
Keywords	Misrepresentation, deception, impression management, authenticity
Requirements	Inform users about the procedure of the authenticity check, Compare users' actual appearance with profile pictures, Ask users for profile pictures, Ask users for authenticity check, Apply authenticity check: calculate similarity between profile pictures and actual appearance, Display authenticity check
Problematic characteristics	-
Desired characteristics	User: Agreeableness, Integrity, Openness, Willingness to Disclose, Authenticity, Honesty Application: Ability, Benevolence, Transparency

Figure F.25: Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Appearance verifier	
Feature Type	<input checked="" type="checkbox"/> Awareness <input checked="" type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	
Trustworthiness facets for application	
Trustworthiness facets for service provider	

Figure F.26: Asset information for the trust-related software feature “appearance verifier”.

Asset Information – Profile setting page	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Propagated:</u> Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	<u>Propagated:</u> Ability, situational normality
Trustworthiness facets for service provider	-

Figure F.27: Asset information for “profile setting page”.

Asset Information – Picture upload algorithm	
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	Ability, situational normality
Trustworthiness facets for service provider	-

Figure F.28: Asset information for “profile upload algorithm”.

Asset Information – Picture upload button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.29: Asset information for “picture upload button”.

Asset Information – Toggle switch “appearance verifier”	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.30: Asset information for “toggle switch “appearance verifier””.

Asset Information – Information icon	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.31: Asset information for “information icon”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Information about appearance verifier	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.32: Asset information for “information about appearance verifier”.

Asset Information – Confirmation window “appearance verifier”	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Propagated:</u> Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.33: Asset information for “confirmation window “appearance verifier””.

Asset Information – Approve button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.34: Asset information for “approve button”.

Asset Information – Decline button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.35: Asset information for “decline button”.

Asset Information – Pattern recognition algorithm	
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	Ability, functionality <u>Propagated:</u> usefulness
Trustworthiness facets for service provider	-

Figure F.36: Asset information for “pattern recognition algorithm”.

Asset Information – Profile picture	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	...
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.37: Asset information for “profile picture”.

Asset Information – Actual appearance	
Trustworthiness facets for users	...
Trustworthiness facets for application	<u>Optional:</u> usefulness
Trustworthiness facets for service provider	-

Figure F.38: Asset information for “actual appearance”.

Asset Information – Real-time photo		
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	Usefulness	
Trustworthiness facets for service provider	-	

Figure F.39: Asset information for “real-time video”.

Asset Information – Real-time photo		
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	Usefulness	
Trustworthiness facets for service provider	-	

Figure F.40: Asset information for “real-time photo”.

Asset Information – Authenticity information		
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Application	<input type="checkbox"/> Service Provider
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	-	
Trustworthiness facets for service provider	-	

Figure F.41: Asset information for “authenticity information”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information - Authenticity score	
Feature Type	<input checked="" type="checkbox"/> Awareness <input checked="" type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Attractiveness, honesty, credibility, truthfulness, authenticity, openness, willingness to disclose, integrity, ethicality, reliability, promise fulfillment
Trustworthiness facets for application	Ability, information quality, usefulness
Trustworthiness facets for service provider	-

Figure F.42: Asset information for “authenticity score”.

F.3 The Catalogue for the Appearance Verifier

Basic Information	
Name	Authenticity check
Problem	Female users look different on their online dating profile pictures than in reality
Keywords	Misrepresentation, deception, impression management, authenticity
Requirements	Inform users about the procedure of the authenticity check, Compare users' actual appearance with profile pictures, Ask users for profile pictures, Ask users for authenticity check, Apply authenticity check: calculate similarity between profile pictures and actual appearance, Display authenticity check
Problematic characteristics	-
Desired characteristics	User: Agreeableness, Integrity, Openness, Willingness to Disclose, Authenticity, Honesty Application: Ability, Benevolence, Transparency

Figure F.43: Basic information of the catalogue for trust-related software features for the concept feature “authenticity check of users”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Appearance verifier	
Feature Type	<input checked="" type="checkbox"/> Awareness <input checked="" type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	
Trustworthiness facets for application	
Trustworthiness facets for service provider	

Figure F.44: Asset information for the trust-related software feature “appearance verifier”.

Asset Information – Profile setting page	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Propagated:</u> Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	<u>Propagated:</u> Ability, situational normality
Trustworthiness facets for service provider	-

Figure F.45: Asset information for “profile setting page”.

Asset Information – Picture upload algorithm	
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	Ability, situational normality
Trustworthiness facets for service provider	-

Figure F.46: Asset information for “profile upload algorithm”.

Asset Information – Picture upload button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.47: Asset information for “picture upload button”.

Asset Information – Toggle switch “appearance verifier”		
User Accessibility	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input type="checkbox"/> Information <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger	
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose	
Trustworthiness facets for application	-	
Trustworthiness facets for service provider	-	

Figure F.48: Asset information for “toggle switch “appearance verifier””.

Asset Information – Information icon		
User Accessibility	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input checked="" type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	-	
Trustworthiness facets for service provider	-	

Figure F.49: Asset information for “information icon”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information – Information about appearance verifier	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.50: Asset information for “information about appearance verifier”.

Asset Information – Confirmation window “appearance verifier”	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input checked="" type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	<u>Propagated:</u> Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.51: Asset information for “confirmation window “appearance verifier””.

Asset Information – Approve button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Agreeableness, integrity, openness, willingness to disclose
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.52: Asset information for “approve button”.

Asset Information – Decline button	
User Accessibility	<input checked="" type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input checked="" type="checkbox"/> Interaction
Nudging Criteria	<input checked="" type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input checked="" type="checkbox"/> Considering motivational state <input checked="" type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	-
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.53: Asset information for “decline button”.

Asset Information – Pattern recognition algorithm	
User Accessibility	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> Prerequisite
Asset Category	<input checked="" type="checkbox"/> Algorithm <input type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	-
Trustworthiness facets for application	Ability, functionality <u>Propagated:</u> usefulness
Trustworthiness facets for service provider	-

Figure F.54: Asset information for “pattern recognition algorithm”.

Asset Information – Profile picture	
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Trustworthiness facets for users	...
Trustworthiness facets for application	-
Trustworthiness facets for service provider	-

Figure F.55: Asset information for “profile picture”.

Asset Information – Actual appearance	
Trustworthiness facets for users	...
Trustworthiness facets for application	<u>Optional:</u> usefulness
Trustworthiness facets for service provider	-

Figure F.56: Asset information for “actual appearance”.

Asset Information – Real-time photo		
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	Usefulness	
Trustworthiness facets for service provider	-	

Figure F.57: Asset information for “real-time video”.

Asset Information – Real-time photo		
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	Usefulness	
Trustworthiness facets for service provider	-	

Figure F.58: Asset information for “real-time photo”.

Asset Information – Authenticity information		
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Application	<input type="checkbox"/> Service Provider
User Accessibility	<input type="checkbox"/> Yes	<input checked="" type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input type="checkbox"/> Design	<input checked="" type="checkbox"/> Information <input type="checkbox"/> Interaction
Trustworthiness facets for users	-	
Trustworthiness facets for application	-	
Trustworthiness facets for service provider	-	

Figure F.59: Asset information for “authenticity information”.

Appendix F. The Catalogues for Trust-Related Software Features for the Online Dating Use Case

Asset Information - Authenticity score	
Feature Type	<input checked="" type="checkbox"/> Awareness <input checked="" type="checkbox"/> Trigger <input checked="" type="checkbox"/> Empowerment
Target group for trustworthiness assessment	<input checked="" type="checkbox"/> User <input type="checkbox"/> Service Provider <input type="checkbox"/> Application
User Accessibility	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> Prerequisite
Asset Category	<input type="checkbox"/> Algorithm <input checked="" type="checkbox"/> Information <input type="checkbox"/> Design <input type="checkbox"/> Interaction
Nudging Criteria	<input type="checkbox"/> Open Choice Architecture <input type="checkbox"/> Guiding information <input type="checkbox"/> Explaining behaviour patterns <input type="checkbox"/> Solution approaches to unfavourable behaviour <hr/> <input type="checkbox"/> Considering motivational state <input type="checkbox"/> Considering user ability <input checked="" type="checkbox"/> Presenting a behavioural trigger
Trustworthiness facets for users	Attractiveness, honesty, credibility, truthfulness, authenticity, openness, willingness to disclose, integrity, ethicality, reliability, promise fulfillment
Trustworthiness facets for application	Ability, information quality, usefulness
Trustworthiness facets for service provider	-

Figure F.60: Asset information for “authenticity score”.

G

Declaration of Individual Contributions

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

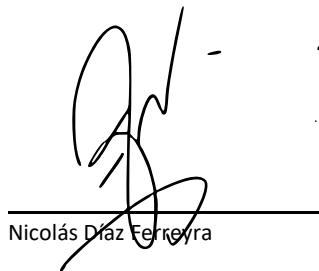
Publication title: Building trustworthiness in computer-mediated introduction: A facet-oriented framework.

Reference item: Borchert, A., Díaz Ferreyra, N. E., & Heisel, M. (2020). Building trustworthiness in computer-mediated introduction: A facet-oriented framework. In International Conference on Social Media and Society, 39-46.

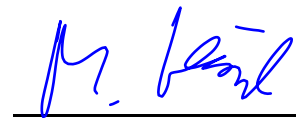
Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Draft of the manuscript.	85 %
Nicolás E. Díaz Ferreyra	<ul style="list-style-type: none">○ Discussion of the approach.○ Supervision and advice.	10 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Nicolás Díaz Ferreyra



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: The Role of Trustworthiness Facets for Developing Social Media Applications: A Structured Literature Review

Reference item: Borchert, A. & Heisel, M. (2022). The Role of Trustworthiness Facets for Developing Social Media Applications: A Structured Literature Review. *Information*,13, 34

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Literature review.○ Draft of the manuscript.	95 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %


Angela Borchert



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS


Publication title: A Conceptual Method for Eliciting Trust-Related Software Features for Computer-Mediated Introduction

Reference item: Borchert, A., Díaz Ferreyra, N. E., & Heisel, M. (2020). A Conceptual Method for Eliciting Trust-Related Software Features for Computer-Mediated Introduction. In *ENASE*, 269-280.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Draft of the manuscript.	85 %
Nicolás E. Díaz Ferreyra	<ul style="list-style-type: none">○ Discussion of the approach.○ Supervision and advice.	10 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Nicolás Díaz Ferreyra



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

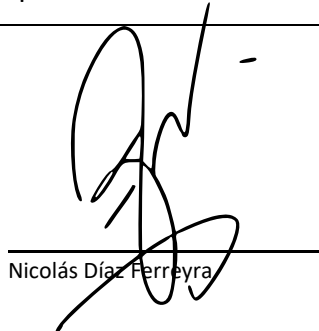
Publication title: Balancing Trust and Privacy in Computer-Mediated Introduction – Featuring Risk as a Determinant for Trustworthiness Requirements Elicitation

Reference item: Borchert, A., Díaz Ferreyra, N. E., & Heisel, M. (2020). Balancing Trust and Privacy in Computer-Mediated Introduction – Featuring Risk as a Determinant for Trustworthiness Requirements Elicitation. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 1-10.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Draft of the manuscript.	85 %
Nicolás E. Díaz Ferreyra	<ul style="list-style-type: none">○ Discussion of the approach.○ Supervision and advice.	10 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Nicolás Díaz Ferreyra



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: Conflict Identification and Resolution for Trust-Related Requirements Elicitation: A Goal Modeling Approach

Reference item: Borchert, A. & Heisel, M. (2021). Conflict Identification and Resolution for Trust-Related Requirements Elicitation: A Goal Modeling Approach. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 12(1), 111-131.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Draft of the manuscript.	95 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: Mitigating Privacy Concerns by Developing Trust-related Software Features for a Hybrid Social Media Application

Reference item: Borchert, A., Wainakh, A., Krämer, N., Mühlhäuser, M., & Heisel, M. (2021, April). Mitigating Privacy Concerns by Developing Trust-related Software Features for a Hybrid Social Media Application. In ENASE (pp. 269-280).

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the study.○ Application of software development method.○ Conduction of the study, data analysis, and study interpretation.○ Draft of the manuscript.	42,5 %
Aidmar Wainakh	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the study.○ Application of software development method.○ Design of prototype.○ Draft of manuscript.	42,5 %
Max Mühlhäuser	<ul style="list-style-type: none">○ Supervision and advice.	5 %
Nicole Krämer	<ul style="list-style-type: none">○ Advice for quantitative analysis.	5 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



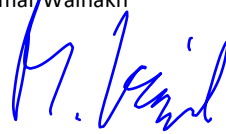
Aidmar Wainakh



Max Mühlhäuser



Nicole Krämer



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: The Relevance of Privacy Concerns, Trust, and Risk for Hybrid Social Media

Reference item: Borchert, A., Wainakh, A., Krämer, N., Mühlhäuser, M., & Heisel, M. (2021, April). The Relevance of Privacy Concerns, Trust, and Risk for Hybrid Social Media. In International Conference on Evaluation of Novel Approaches to Software Engineering (pp. 88-111). Springer, Cham.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the study.○ Application of software development method.○ Conduction of the study, data analysis, and study interpretation.○ Draft of the manuscript.	42,5 %
Aidmar Wainakh	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the study.○ Application of software development method.○ Design of prototype.○ Draft of manuscript.	42,5 %
Max Mühlhäuser	<ul style="list-style-type: none">○ Supervision and advice.	5 %
Nicole Krämer	<ul style="list-style-type: none">○ Advice for quantitative analysis.	5 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



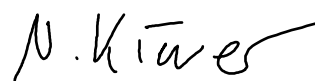
Angela Borchert



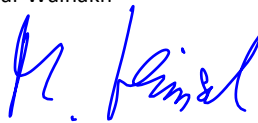
Aidmar Wainakh



Max Mühlhäuser



Nicole Krämer



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: Meeting Strangers Online: Feature Models for Trustworthiness Assessment

Reference item: Borchert, A., Díaz Ferreyra, N. E., & Heisel, M. (2022). Meeting Strangers Online: Feature Models for Trustworthiness Assessment. In *Human-Centered Software Engineering: 9th IFIP WG 13.2 International Working Conference, HCSE 2002, Eindhoven, The Netherlands, August 24-26, 2022, Proceedings (pp. 3-22)*. Cham: Springer International Publishing.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification of the work.○ Draft of the manuscript.	85 %
Nicolás E. Díaz Ferreyra	<ul style="list-style-type: none">○ Discussion of the approach.○ Supervision and advice.	10 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Nicolás Díaz Ferreyra



Maritta Heisel

DECLARATION OF INDIVIDUAL CONTRIBUTIONS

Publication title: Safety First? Gender Differences in Online Dating Behavior and Trust Concerns

Reference item: Borchert, A., Cassidy, E., & Heisel, M. (2022). Safety First? Gender Differences in Online Dating Behavior and Trust Concerns, Submitted for publication.

Author	Contribution	%
Angela Borchert	<ul style="list-style-type: none">○ Conceptualisation of the approach.○ Planification, conduction, and analysis of the interview study.○ Recruitment of study participants.○ Draft of the manuscript.	75 %
Elija Cassidy	<ul style="list-style-type: none">○ Planification of the interview study.○ Draft of the manuscript.○ Supervision and advice.	20 %
Maritta Heisel	<ul style="list-style-type: none">○ Supervision and advice.	5 %



Angela Borchert



Elija Cassidy



Maritta Heisel

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

Diese Dissertation wird via DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt und liegt auch als Print-Version vor.

DOI: 10.17185/duepublico/81410

URN: urn:nbn:de:hbz:465-20240201-144018-0

Alle Rechte vorbehalten.