



A Framework for Modeling and Improving Agile Requirements Engineering

- PhD Thesis -

Eva-Maria Schön
September | 2017

Advisors: Dr. M^a José Escalona Cuaresma
Dr. Jörg Thomaschewski

Computer Languages and Systems Department
University of Seville

Contents

Chapter 1	Introduction.....	13
Chapter 2	Related Work	17
2.1.	Foundations Agile Methodologies, Human-Centered Design and Requirements Engineering 17	
2.1.1.	Agile Methodologies.....	17
2.1.2.	Human-Centered Design	18
2.1.3.	Requirements Engineering.....	19
2.1.4.	Agile Requirements Engineering.....	20
2.2.	Relevant Work from other Research Groups	22
2.2.1.	Salah et al.....	22
2.2.2.	Silva et al.....	22
2.2.3.	Méndez Fernández et al.....	23
2.3.	State of the Art Agile Requirements Engineering	24
2.3.1.	Summary of Related Literature Reviews.....	24
2.3.2.	Gap Analysis in Related Literature Reviews.....	25
2.3.3.	Review Method.....	26
2.3.4.	Results and Discussion – State of the Art	31
2.3.5.	Revision of State of the Art Analysis.....	43
2.4.	Implications and Gap Analysis	48
2.5.	Chapter Summary.....	49
Chapter 3	Challenges and Objectives	50
3.1.	Challenges	50
3.2.	Objectives.....	51
3.2.1.	Perform a Systematic Literature Review.....	53
3.2.2.	Propose a Metamodel for Agile RE	54
3.2.3.	Provide Domain Specific Models of the Metamodel.....	54
3.2.4.	Identify the Most Important Problems in Agile RE.....	54
3.2.5.	Create Agile RE Patterns.....	55
3.2.6.	Share the Knowledge Concerning Agile RE.....	55
3.3.	Chapter Summary.....	56
Chapter 4	Metamodel for Agile Requirements Engineering.....	57
4.1.	Summary of Related Work.....	57
4.2.	Gap Analysis of Related Work	58
4.3.	Research Approach.....	59
4.3.1.	Objectives and Research Questions.....	59

4.3.2.	Developing the Agile RE Metamodel	59
4.3.3.	Developing a Profile for the Agile RE Metamodel	60
4.4.	Results and Discussion – Agile RE Metamodel	61
4.4.1.	Description of the Agile RE Metamodel.....	63
4.4.2.	Additional Views on the Agile RE Metamodel	76
4.4.3.	Profile for Agile RE Metamodel	78
4.4.4.	Impact and Limitations.....	79
4.5.	Chapter Summary.....	80
Chapter 5	Identification of Agile RE Problems.....	81
5.1.	Summary of Related Work.....	81
5.2.	Gap Analysis of Related Work	82
5.3.	Research Method	83
5.3.1.	Objectives and Research Questions.....	84
5.3.2.	General Study Design	84
5.3.3.	Panel of Experts.....	85
5.3.4.	Round 1.....	87
5.3.5.	Round 2.....	88
5.3.6.	Round 3.....	91
5.4.	Results and Discussion – Agile RE Problems	93
5.4.1.	(RQ-5.1) What Are The Key Problems in Agile Requirements Engineering?	93
5.4.2.	(RQ-5.2) How Can We Deal With The Identified Key Problems?	102
5.4.3.	Meaning of Findings.....	104
5.4.4.	Limitations	104
5.5.	Chapter Summary.....	105
Chapter 6	Creation of Agile RE Patterns	106
6.1.	Summary of Related Work.....	106
6.2.	Gap Analysis of Related Work	107
6.3.	Research Method	107
6.3.1.	Objectives and Research Questions.....	108
6.3.2.	Pattern Mining Process.....	108
6.4.	Results and Discussion – Agile RE Patterns.....	110
6.4.1.	Overview Agile RE Patterns.....	111
6.4.2.	Examples of Agile RE Patterns	114
6.4.3.	Knowledge Sharing by Means of a Web Application	119
6.5.	Chapter Summary.....	121

Chapter 7	Application of the Framework in Industry.....	122
7.1.	General Application of the Framework	122
7.2.	Research Method	124
7.2.1.	Objectives and Research Questions.....	124
7.2.2.	Tailoring the Metamodel into Domain Specific Models	124
7.3.	Results and Discussion – Application of the Framework	125
7.3.1.	Case A: Tailoring Within a Kanban-based Environment in Germany	125
7.3.2.	Case B: Tailoring Within a Scrum-based Environment in Spain.....	128
7.3.3.	Impact and Limitations.....	131
7.4.	Chapter Summary.....	132
Chapter 8	Results, Future Work and Conclusions	133
8.1.	Results.....	133
8.2.	Future Work	136
8.3.	Relation to IWT2 research group	137
8.4.	Conclusions.....	138
References		140
Appendix I	Glossary.....	148
Appendix II	Protocol of SLR.....	150
Appendix III	Change Log Agile RE Metamodel.....	156
Appendix IV	Agile RE Problems	159
Appendix V	Publications and Experiences	210

Figures

Figure 1: Graphical abstract highlighting the contribution of this PhD thesis by means of summarizing the connections of the outcomes.	15
Figure 2: Agile Values stated by the Manifesto for Agile Software Development (cf. (Beck et al., 2001)).....	18
Figure 3: Human-Centred Design ISO 9241-210:2010 (International Organization for Standardization, 2010).....	18
Figure 4: Sequential phases approach to software development.....	19
Figure 5: In ASD people focus on outcomes and how they can fulfil human needs through the outputs they produce.....	20
Figure 6: Venn diagram showing intersection of research areas influencing agile RE.....	21
Figure 7: Related work has not investigated Agile RE from the perspective of HCD up to now.....	25
Figure 8: Phases of a SLR (cf. (Kitchenham and Charters, 2007)).....	26
Figure 9: Search process comprising six phases and snowballing. After analyzing the data, we included 27 papers into our study.....	28
Figure 10: Quality assessment.....	32
Figure 11: Revision of search process comprising six phases and snowballing. After analyzing the data, we included 5 papers into the revision of our study.....	44
Figure 12: Distribution of search results among digital libraries.....	45
Figure 13: Conceptual schema representing the objectives to be achieved with the PhD thesis.....	53
Figure 14: Mapping between vision of this PhD thesis, objectives and research questions	56
Figure 15: Phases to create the agile RE metamodel.....	59
Figure 16: Creating a profile with Enterprise Architect.....	61
Figure 17: Agile RE metamodel (v.1.8).....	62
Figure 18: Detailed view on metaclass <<User>>.....	76
Figure 19: Detailed view on metaclass <<Stakeholder>>.....	77
Figure 20: Detailed view on metaclass <<AgileTeam>>.....	77
Figure 21: Profile for agile RE metamodel.....	78
Figure 22: Phases of a Delphi study.....	83
Figure 23: General process of study.....	84
Figure 24: Type of organization for which experts work.....	86
Figure 25: Process models used by experts.....	86
Figure 26: Excerpt of questionnaire of round 1.....	87
Figure 27: Exemplary item of round 2.....	89
Figure 28: Degree of refinement of requirements.....	91
Figure 29: Categorized key problems in agile RE.....	95
Figure 30: Importance key problem C1 (N=13).....	96
Figure 31: Importance key problem C2 (N=14).....	97
Figure 32: Importance key problem C3 (N=15).....	98
Figure 33: Importance key problem C4 (N=15).....	99
Figure 34: Importance key problem C5 (N=12).....	100
Figure 35: Importance key problem C6 (N=14).....	101
Figure 36: Steps for creation of agile RE patterns.....	108
Figure 37: Pattern mining process and knowledge sharing.....	108

Figure 38: Relation between agile RE problem and agile technique.....	109
Figure 39: Overview of types of agile techniques.....	110
Figure 40: Agile RE patterns are composed of dynamic and static parts	112
Figure 41: Metaclass AgileREPattern	114
Figure 42: Example of a Kanban board.....	115
Figure 43: Example of a story map	116
Figure 44: Review Meeting in form of an exhibition (Glase, 2016)	118
Figure 45: Landing page agileRE.org.....	119
Figure 46: Example of agile RE pattern (Story Map) presented on agileRE.org	120
Figure 47: General application of the framework by means of tailoring domain specific models for agile RE	123
Figure 48: Modeling agile RE by instantiating the agile RE metamodel.....	124
Figure 49: Domain specific model for agile RE in a Kanban-based environment.....	126
Figure 50: Example of multiple Kanban boards for pattern Continuous management of requirements by means of tools.....	128
Figure 51: Domain specific model for agile RE in a Scrum-based environment	129
Figure 52: Search process.....	151
Figure 53: Item 1.1 – direct cooperation with end users	178
Figure 54: Item 1.2 – iterative involvement of stakeholders	178
Figure 55: Item 1.3 – face to face communication.....	179
Figure 56: Item 1.4 – evaluation of requirements with end users	179
Figure 57: Items 1.5 – review of implemented requirements.....	180
Figure 58: Item 1.6 – continuous communication.....	180
Figure 59: Item 2.1 – understanding of agile values.....	181
Figure 60: Item 2.2 – avoiding large requirements analysis in the beginning	181
Figure 61: Item 2.3 – decisions of the development team	182
Figure 62: Item 2.4 – handling requirements within an organization	182
Figure 63: Item 2.5 – understanding agile values in terms of fears of losing control	183
Figure 64: Item 2.6 – perception of requirement analysis.....	183
Figure 65: Item 3.1 – impact analysis of components	184
Figure 66: Item 3.2 – impact of styleguides and prototypes on user experience	185
Figure 67: Item 3.3 – restricting solution finding with finalized prototypes	185
Figure 68: Item 3.4 – continuous requirements management.....	186
Figure 69: Item 3.5 – classical methods for elicitation and evaluation.....	186
Figure 70: Item 3.6 – sharing insights with the whole development team.....	186
Figure 71: Item 3.7 – traceability of requirements	187
Figure 72: Item 3.8 – point of time for carrying out RE in agile projects	187
Figure 73: Item 3.9 – handling non functional requirements	188
Figure 74: Item 3.10 – changes in requirements documentation.....	188
Figure 75: Item 4.1 – detailing requirements for short-term iterations	189
Figure 76: Item 4.2 – short-term changes in priorities of requirements.....	189
Figure 77: Item 4.3 – outlook on upcoming iterations	190
Figure 78: Item 4.4 – avoiding effort estimation in hours.....	190
Figure 79: Item 4.5 – slicing requirements to fit into iteration	191
Figure 80: Item 4.6 – losing sight of the big picture due to complex requirements	191
Figure 81: Item 5.1 – coordination effort due to dependencies.....	192
Figure 82: Item 5.2 – analyzing requirements to avoid interdependencies	192
Figure 83: Item 5.3 – describing requirements in terms of objectives	193

Figure 84: Item 5.4 – deviating test cases from requirements	193
Figure 85: Item 5.5 – clarity of requirements.....	193
Figure 86: Item 5.6 – reasoning about benefits of a requirement	194

Tables

Table 1: Activities in agile RE	21
Table 2: Keywords used for search	27
Table 3: Search space	28
Table 4: Quality checklist.....	30
Table 5: Distribution according to research methods.....	32
Table 6: Sub-criteria RQ1.....	33
Table 7: Sub-criteria RQ2.....	36
Table 8: Sub-criteria RQ3.....	38
Table 9: Artifacts in Agile RE.....	38
Table 10: Overview of parameters for revision of search	43
Table 11: Findings of revision related to RQ1.1	45
Table 12: Findings of revision related to RQ1.2	46
Table 13: Overview of objectives of this PhD thesis.....	52
Table 14: Overview of conceptual improvements of the agile RE metamodel	60
Table 15: Color coding scheme for the agile RE metamodel	62
Table 16: Problems in agile RE reported by related work.....	82
Table 17: Overview of rounds.....	85
Table 18: Know-how of panel (N=26) in terms of ASD	86
Table 19: Exemplary item in round 1	88
Table 20: Problems in agile RE (key problems are highlighted in green, whereas problems where only 1/3 of the experts rate as problem are highlighted in yellow).....	92
Table 21: Key problems in agile RE.....	94
Table 22: Calculations of importance concerning key problem C1	96
Table 23: Calculations of importance concerning key problem C2	97
Table 24: Calculations of importance concerning key problem C3	98
Table 25: Calculations of importance concerning key problem C4	99
Table 26: Calculations of importance concerning key problem C5	100
Table 27: Calculations of importance concerning key problem C6	101
Table 28: Agile RE patterns matched to agile RE problems.....	111
Table 29: Classification of Agile RE patterns in dynamic and static.....	113
Table 30: Search space	151
Table 31: Full list of search keywords	152
Table 32: Inclusion and exclusion criteria.....	153
Table 33: Quality assessment criteria	154
Table 34: Data Collection Form	154

Acknowledgements

There are many people around me, who supported me during this PhD. I hope that I can express all my gratitude along the following paragraphs. This work would not have been possible without all of the support and effort from these people.

To my family and especially my husband for their patience, understanding and love.

To my friends around me for being there and always having an open ear.

To my PhD directors María José Escalona and Jörg Thomaschewski for their collaboration, their advice and all their valuable feedback, which guides me through this tough times.

To all my fellow researchers in the IWT2 research group (my new family in Seville) for all the valuable discussions, feedback and hospitality.

To all my fellow researchers in the Research Group for Agile Software Development and User Experience (Emden) for all discussions and feedback.

To my colleagues from CGI for backing and supporting in pretests and philosophical discussions about Agile. In addition, to my mentors Walid Farag and Lars Napret.

To the experts in metamodeling, for their feedback, which helped improving my metamodel and the profile (Manuel Mejías, Nora Koch, and the IWT2 Research Group).

To the panel of experts of my iterative expert judgement process, who provided valuable data and insights. And also to Dominique Winter, who supported the study of identifying agile RE problems.

To the experts, who helped improving my agile RE patterns. In particular, my shepherd Dina Salah and the pattern community, with whom I get in touch with during the EuroPLoP 2017.

To Stefan Stratmann, who supported the realization of agileRE.org.

To Brigit van Loggem, who helped me improving the vision statement and the elevator pitch of this PhD thesis.

To all people of the conferences, with whom I get in touch with during the last years. The discussions were very constructive and helped me refining my ideas.

To the experts, who helped improving my manuscript (Inmaculada Morilla Hidalgo, Jens Hellmers).

To all of you, thank you! This work would not have been done without your support.

Abstract

Context. Companies adopt hybrid development models consisting of an integration of agile methodologies and Human-Centered Design (HCD) with the aim to increase value delivery as well as to reduce time to market. This has an impact on how Requirements Engineering (RE) is carried out in an agile environment. To this end, people apply different kind of agile techniques like artifacts, meetings, methods, and roles. In this context, companies often struggle with improving their value chain in a systematic manner, since guidelines for choosing an appropriate set of agile techniques are missing.

Objective. The vision of this PhD thesis is to build a framework for modeling agile RE. Organizations benefit from implementing this framework by increasing their value delivery (organization external) and improving the collaboration (organizational intern).

Method. We followed an inductive research approach, where we used the learnings from several studies to create the framework. In the beginning, we carried out a Systematic Literature Review (SLR) to analyze the state of the art of agile RE with focus on user and stakeholder involvement. Subsequent, we created the agile RE metamodel, which evolved iteratively along the consecutively studies. Based on the metamodel, we defined an profile that can be used to create domain specific models according to the organizational environment. Moreover, we conducted a Delphi study in order to identify the most important problems industry has to face today in terms of agile RE. The results were used as input for a systematic pattern mining process, which was utilized in order to create agile RE patterns.

Results. The framework for modeling agile RE consists of three main components: i) *agile RE metamodel*, which can be used to analyze the organizational environment in terms of value delivery ii) *catalogue of agile RE problems*, which allows to detect recurring problems in terms of agile RE iii) *catalogue of agile RE patterns*, which allows to solve the detected problems. The agile RE metamodel comes with a profile, which can be used to deviate domain specific models. In addition, we created tool support for the framework by means of a web application (agileRE.org), which allows us to share the knowledge and best practices for agile RE. Furthermore, we proved how the framework can be applied in industry by means of case studies in Germany and in Spain.

Conclusion. The framework for modeling agile RE empowers companies to improve their organizational environments in terms of value delivery and collaboration. Companies can use the framework for improving their value chain in a systematic manner. In particular, it gives guidance for choosing appropriate agile techniques, which fit to the changing needs of the organizational environment. In addition, we can state that the framework is applicable on an international level.

Resumen

Contexto. Con el objetivo de incrementar la potencialidad de sus desarrollos y de reducir el tiempo de puesta en el mercado, las empresas adoptan modelos de desarrollo híbridos que integran metodologías ágiles y diseño centrado en el usuario (DCU). El tratamiento de los requisitos de software en entornos ágiles es algo que impacta de manera directa en la consecución de estos objetivos. Por ello, los equipos aplican diferentes técnicas de tratamiento de requisitos como los artefactos, reuniones, métodos de trabajos grupales o el tratamiento efectivo de roles. Sin embargo, las empresas a menudo se encuentran con dificultades para elegir las mejores técnicas a aplicar en su contexto y hay una carencia de guías de soporte.

Objetivo. La visión de esta tesis doctoral es construir un framework para trabajar de manera efectiva con requisitos ágiles. La idea esencial es que las organizaciones y empresas puedan usar el framework para mejorar tanto su cadena de valor (visión externa) como para mejorar sus procesos de desarrollo (visión interna).

Método. Para el desarrollo del trabajo se ha usado una metodología de investigación inductiva, usando diferentes métodos de trabajo. Inicialmente, se ha llevado a cabo un estudio sistemático de la literatura (SLR) que nos permite evaluar el estado del arte en el tratamiento de requisitos ágiles pero centrado en cómo se trabaja con la involucración de los diferentes stakeholders en el proceso. Hemos continuado aplicando la ingeniería guiada por modelos desarrollando un metamodelo para trabajar con los requisitos ágiles y un perfil que permite definir un lenguaje específico de dominio para el uso del metamodelo en entornos concretos. Este trabajo se ha enriquecido con la aplicación de un estudio usando Delphi para identificar los problemas más importantes que la industria se encuentra a la hora de trabajar con ingeniería de requisitos en entornos ágiles. Finalmente, con los resultados hemos conseguido desarrollar un conjunto de patrones para la creación de requisitos ágiles.

Resultados. El framework para modelar requisitos ágiles tiene tres componentes principales: i) Metamodelo para trabajar con requisitos ágiles que servirá para analizar el entorno de la organización. ii) un catálogo de posibles problemas que se encuentran en entornos ágiles y iii) un catálogo de patrones de requisitos ágiles que resuelven los problemas detectados. El metamodelo para el trabajo con requisitos ágiles viene acompañado de un lenguaje específico de dominio, basado en un perfil. Y, además, se ha creado una aplicación web (agileRE.org) que ayuda a poner en común el conocimiento. Por último, el framework ha sido aplicado con éxito en entornos empresariales españoles y alemanes.

Conclusión. El framework para modelar requisitos ágiles ayuda a las compañías a mejorar sus entornos organizaciones in términos de costes de desarrollo y aspectos colaborativos. Las empresas pueden usar el framework para mejorar su cadena de valor de una manera sistemática. En particular, da una guía para elegir técnicas apropiadas en el tratamiento de requisitos ágiles, pudiendo adaptarse al a realidad del entorno concreto de trabajo.

Business world is characterized by complexity, since market requirements are changing quickly. Accordingly, providers are facing the challenge to reduce time to market while delivering innovative products that customer love. *Agile software development (ASD)* promises benefits such as on-time delivery and customer satisfaction (Dingsøyr and Dyba, 2008), thus it aims to deliver business value in short iterations. Therefore, the development process is carried out incrementally and empirically, which is an advantage because direction of product development can be changed immediately. Besides, humans and interactions are at the center of such methodologies (Beck et al., 2001).

ASD is adopted by industry both to reduce time to market and to increase value delivery for customers and users. Agile methodologies like Scrum (Schwaber, 2004), Extreme Programming (XP) (Beck, 2000) or Kanban (Anderson, 2010) share an environment which is continuously improving in terms of collaboration, processes and tools by using mechanisms like retrospectives (Schwaber, 2004) or kaizen (Anderson, 2010).

In industry, new trends emerge quickly and agile techniques and tools are volatile. The continuous improvement of the environment lead to a quickly evolving as well as rapidly changing knowledge base in this field of research. To this end, industry and research must work together hand in hand in order to stay connected.

The increasing distribution of agile methodologies in industry leads to changes in value systems of organizations. Focusing on user needs and value delivery becomes an important aspect in product development due to the increasing competition in all areas. In addition, value delivery is an emergent research theme as shown by (Dingsøyr and Lassenius, 2016).

Agile methodologies have in common the agile values on which they rely (Beck et al., 2001):

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Agile values have an impact on the culture of an organization. Values like embracing change, collaboration, and value delivery become necessary assets. This implies changes in value systems of organizations from plan-driven to value-driven.

On the one hand people in plan-driven organizations often negotiate about project plans, pricing models and the amount of features they can develop with the available resources. They are emphasizing the generated outputs such as number of created features during a time period. On the other hand, people in value-driven organizations discuss visions, experiences and human values as well as the way to address them through the product. They focus on the outcomes that the delivered outputs entail.

Current survey studies reveal that often hybrid process models consisting of an integration of different agile methodologies are applied in industry (VersionOne Inc., 2017), (Komus et al., 2017). One example is NDT-Agile (Salinas Torrecilla, 2016), which is a mature Agile framework for Web projects based on an extension of Scrum, XP, and NDT (Navigational Development Techniques)

(Escalona and Aragon, 2008), with the main goal of covering all CMMI-DEV goals. Further examples of hybrid models can be found in (Schön et al., 2016b), (Pfeiffer et al., 2016).

These hybrid process models are adapted to the specific needs of the environment and have some commonalities in terms of requirements. Requirements in those environments are volatile due to changing business priorities, evolving markets, or technical conditions. Industry has the challenge to find out how to cope with those volatile conditions.

Therefore, established Requirements Engineering (RE) approaches need to be adapted to the requirements of ASD. The conditions ASD comes with have an impact on the way RE is carried out in agile environments. On the one hand, sequential approaches to RE (Sommerville and Sawyer, 1997) need to be adapted due to the iterative and incremental approach. On the other hand, additional methodologies like Human-Centered Design (HCD) (International Organization for Standardization, 2010) are integrated due to the strong focus on user and stakeholder involvement. HCD is defined by ISO 9241-210:2010¹ (International Organization for Standardization, 2010) and describes an approach to interactive product development where the ergonomics of human-system interaction plays an important role. ISO/DIS 9241-220.2 (International Organization for Standardization, 2017) details this concept by providing processes for enabling, executing and assessing human-centered design within organizations. Compared to User-Centered Design (UCD) (International Organization for Standardization, 1999), HCD covers a broader view on human needs and emphasizes the impacts on further stakeholders besides the user.

Flexibility, continuous management of requirements as well as a high level on communication and collaboration among disciplines and departments are essential. Companies need to bridge the gaps within one organization and to break up silos, so that people can work together and strive for continuous improvement in order to reach the common goal: increasing value delivery for customers and users.

The vision of this PhD thesis is to build a framework for modeling and improving agile RE process models to increase value delivery and collaboration by means of empirical research. To this end, we want to achieve the following objectives:

- Perform a systematic literature review, in order to capture the current state of the art of agile RE with focus on stakeholder and user involvement.
- Propose a conceptual approach for agile RE in order to provide an abstract solution for agile RE management by means of a metamodel and an application into domain specific models by means of a profile.
- Provide techniques to apply the conceptual approach to the enterprise environment by means of agile RE problems and agile RE patterns.
- Share the knowledge concerning agile RE, in order to empower people to become a value-driven organization.

Figure 1 highlights the contribution of this PhD thesis. In order to achieve our vision, we identified the most important problems in the context of agile RE industry has to face today by means of an iterative expert judgement process with 26 experts in the field of ASD. The identified agile RE problems were generalized over three rounds by reducing contextual information in order to find the core of the problems.

¹ In this PhD thesis we will use the international standards instead of national standards (e.g. UNE EN ISO 9241-210:2010 for Spain or DIN EN ISO 9241-210:2011-01 for Germany).

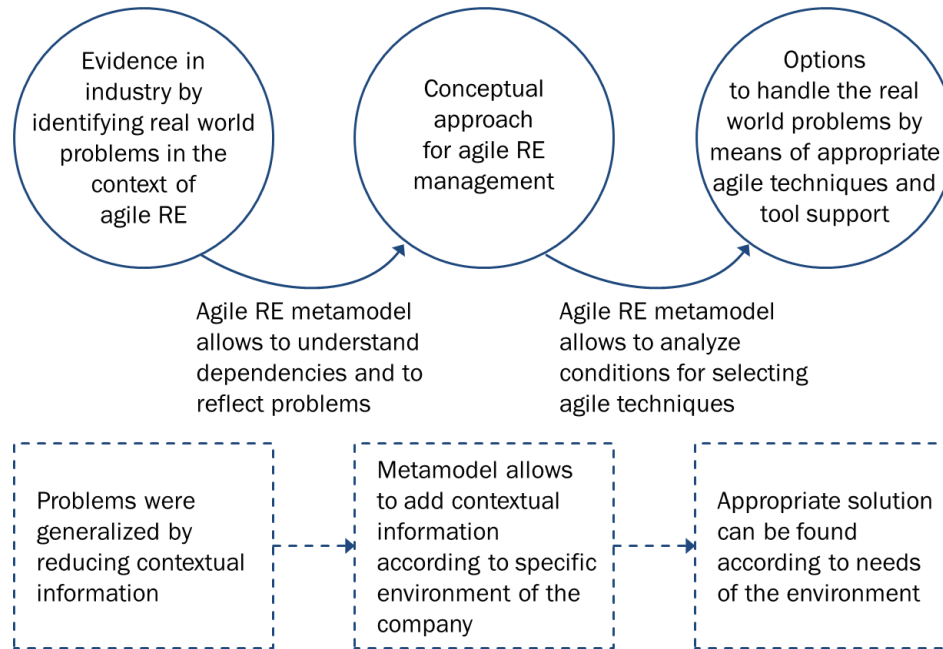


Figure 1: Graphical abstract highlighting the contribution of this PhD thesis by means of summarizing the connections of the outcomes.

Moreover, we contribute a conceptual approach for agile RE management by means of the agile RE metamodel. Agile practitioners and researcher can use the metamodel in order to model or improve their agile RE approaches. On the one hand, the metamodel can be used to understand the dependencies and enables to reflect agile RE problems. On the other hand, it allows adding contextual information according to the specific environment in which the product development is carried out.

In this context, the agile RE metamodel helps to analyze the conditions for selecting appropriate agile techniques with which agile RE problems can be solved. With regard to problem solving, we contribute options to handle the real world problems by means of appropriate agile techniques and providing tool support.

The tool consists of agile RE patterns, which are distributed among the community by means of a web application (agileRE.org) with the aim to share the knowledge about agile RE.

The framework for modeling agile RE has been built by an inductive (bottom up) research approach, where we used the learnings from our research studies in order to create the components of the framework. Hence, this PhD thesis is structured as follows:

Chapter 2 starts with a deep analysis of the state of the art of agile RE with focus on stakeholder and user involvement. We will analyze the relation among Agile, HCD, and RE. Moreover, we will carry out a gap analysis of the related work.

Then, chapter 3 details the challenges and objectives of this PhD thesis based on the results of the analysis of the related work.

Chapter 4 presents the agile RE metamodel, which is the heart of our framework. We will explain the metamodel and provide an abstract view on the complex field of agile RE. Moreover, we will discuss the benefits and limitations of the metamodel. In addition, we create a visual language for the agile RE metamodel by means of a profile. This profile can be used to build instances of the metamodel, which allows solving real problems in industry.

In chapter 5, we will present the catalogue of agile RE problems. Therefore, we identify the most important problems in agile RE industry has to cope with today. We will discuss the results of our iterative expert judgement process with 26 experts working in the field of ASD.

Subsequent, chapter 6 shows the catalogue of agile RE patterns. To this end, the chapter introduces the concept of agile RE patterns, with which the identified agile RE problems can be solved by means of examples. In addition, the web application (agileRE.org) will be explained.

Chapter 7 provides an overview of how the framework for modeling agile RE can be applied in industry. Therefore, we present domain specific models of the metamodel in order to show how the metamodel can be instantiated using the profile. One process model is based on Kanban and is used in the area of e-commerce. The other process model is based on Scrum and applied in the area of e-government

Finally, chapter 8 presents the final conclusion of this research, points out future lines of research and outlines the relation of this work to the IWT2 research group.

This work also contains five Appendixes: Appendix I lists a glossary of the different terms used throughout the work, in order to ease its understanding. Appendix II provides additional material to the state of the art analysis by means of the underlying review protocol. Then, Appendix III gives an overview of the improvements of the metamodel by means of a change log. Appendix IV includes additional material of the iterative expert judgement process and shows original statements given by the experts as well as the results reports. In conclusion, Appendix V enumerates all publications and research projects.

The previous chapter presented the context of the problem domain we want to address with this PhD thesis, which refers to the three main fields of *Agile*, *HCD* and *RE*. In this chapter, we will explore the problem domain in detail and perform a deep analysis of the state of the art related to agile RE with focus on user and stakeholder involvement.

In section 2.1, foundations concerning the three main fields (Agile, HCD, and RE) will be discussed. Based on these foundations, we will give a definition of agile RE, which is used in this PhD thesis. Section 2.2 presents the related work from other research groups. Then, section 2.3 aims to capture the current state of the art of research related to agile RE with focus on stakeholder and user involvement. Therefore, we carried out a secondary study of existing research by means of conducting a Systematic Literature Review (SLR). In particular, we investigate what approaches exist to involve stakeholder in the process, which methodologies are accepted to present the user perspective and how requirements management is been carried out. Section 2.4 discusses on the implications of the findings and outlines the gaps in existing research.

2.1. Foundations Agile Methodologies, Human-Centered Design and Requirements Engineering

In this section, we will present the foundations related to agile methodologies, HCD, and RE. In addition, we will provide a definition for agile RE, which builds the basis for this work and which we elaborate in the following chapters.

2.1.1. Agile Methodologies

In the mid-80s, Takeuchi and Nonaka (Takeuchi et al., 1986) already stated that a sequential phases approach to product development is not well suited due to the lack of flexibility. In the 90s, there have been different movements concerning lightweight process models like *Scrum* (Schwaber, 1997), (Schwaber, 2004), *Extreme Programming* (XP) (Beck, 2000), or *Feature-Driven Development* (Palmer and Felsing, 2001).

In 2001, the leaders of these different movements came together and tried to find a common ground for their work. As result, the *Agile Manifesto* (Beck et al., 2001) emerged. The agile manifesto comprises agile values (see Figure 2), which are assisted by 12 principles (see (Beck et al., 2001)).

Although the agile manifesto was created in 2001, agile values still guiding agile teams and most of the principles even play an important role in today's agile community (Williams, 2012), (Schön et al., 2015).

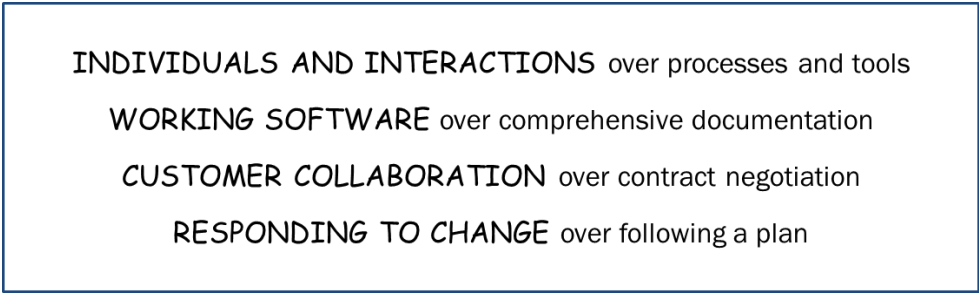


Figure 2: Agile Values stated by the Manifesto for Agile Software Development (cf. (Beck et al., 2001))

Nowadays, ASD is becoming more popular in all fields of industry due to enabling immediately changes in the direction of product development. Current surveys reveal that Scrum and hybrid models based on Scrum/XP are agile methodologies followed most closely (VersionOne, 2016). Moreover, Kanban (Anderson, 2004), (Anderson, 2010) is playing an important role among the agile community (Komus et al., 2017).

2.1.2. Human-Centered Design

During the last decades a lot of different terms evolved describing quality models from a user’s perspective as well as process models to create interactive products addressing these quality models.

HCD is defined by ISO 9241-210:2010 (International Organization for Standardization, 2010) and describes an approach to interactive product development where the ergonomics of human-system interaction plays an important role. ISO 9241-210:2010 offers generic recommendations for every step of the HCD process (see Figure 3). The document explicitly does not recommend specific methods. In order to implement HCD into an organization ISO/DIS 9241-220.2 (International Organization for Standardization, 2017) has been created. ISO/DIS 9241-220.2 provides processes for enabling, executing and assessing human-centered design within organizations.

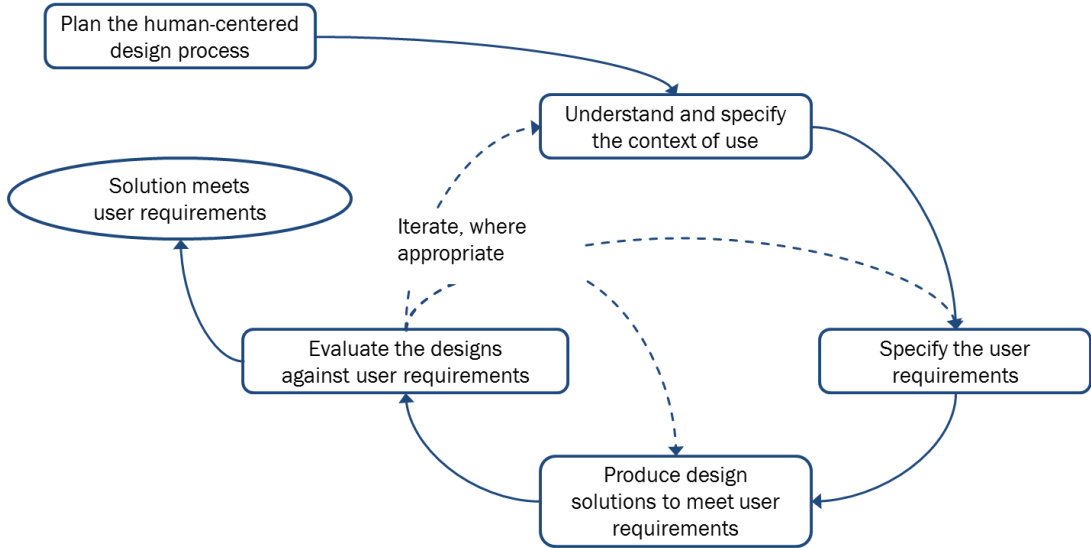


Figure 3: Human-Centred Design ISO 9241-210:2010 (International Organization for Standardization, 2010)

The concept of HCD is elaborated on *User-Centered Design (UCD)* (International Organization for Standardization, 1999) and covers a broader view on human needs. Compared to UCD, HCD emphasizes the impacts on further stakeholders besides the user.

The *User Experience (UX)* is a “*person's perceptions and responses resulting from the use and/or anticipated use of a product, system or service*” (International Organization for Standardization, 2010). Whereas the *Usability* is defined as extent “*to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use*”(International Organization for Standardization, 1998).

2.1.3. Requirements Engineering

Requirements are the base of all software products and consequently *Requirements Engineering (RE)* plays an important role in system development. The term *Requirements Engineering* was already used in the 70-s (Richter et al., 1976). Then, RE gained in popularity with the first conferences in the 90-s (RE '93 and ICRE '94²).

Traditional process models, such as the *waterfall model* (Royce, 1970) consist of sequential phases and start with an upfront design. Figure 4 shows the underlying sequential phases. In the beginning of these projects, all requirements are consolidated into a specification document. Based on this, schedule and budget are estimated. Although Royce (Royce, 1970) stated that there are iterations in the process of development of software, he conveyed that the requirements analysis is completed at some point. In industry, this leads to the assumption that requirements are seen as true and changes are made using a heavyweight change request process while using traditional process models.

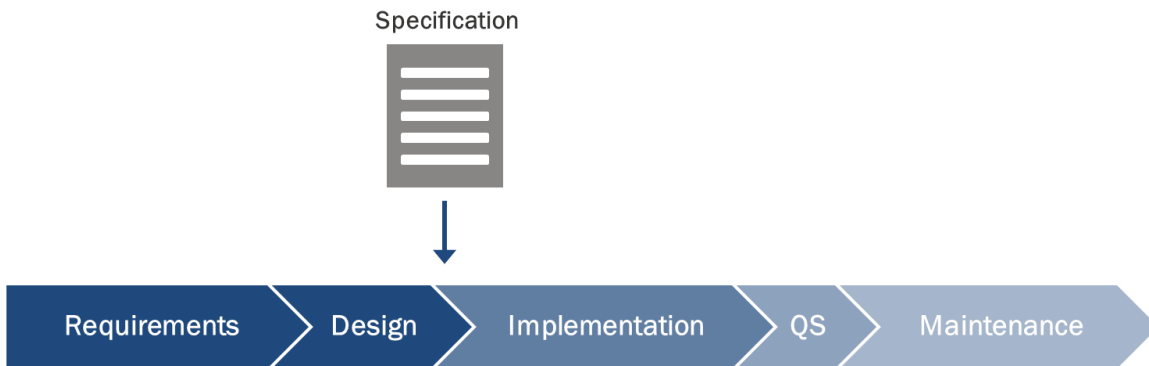


Figure 4: Sequential phases approach to software development

Sommerville and Sawyer (Sommerville and Sawyer, 1997) describe RE as combination of all activities concerning discovering, documenting, and maintaining a set of requirements. They recommend using systematic and repeatable techniques to ensure that product requirements are complete, consistent and relevant. The requirements are consolidated in a specification document, where details are added while product development proceeds.

² <http://requirements-engineering.org>

2.1.4. Agile Requirements Engineering

The established definition of RE, outlined in section 2.1.3 fails in terms of ASD. Agile methodologies are used to cope with increasing complexity in product development. The former definition of RE and the understanding of it the does not work in an agile environment.

Since the agile movement, software development has moved away from plan-driven to value-driven process models (see Figure 5). People in plan-driven environments often negotiate about pricing models, project plans and how many product features they can develop with the available resources. They are emphasizing the generated outputs (e.g. number of created features or number of releases during a time period). In contrast, people in value-driven environments discuss visions, experiences and human values as well as how they can address them through the product. They concentrate on outcomes, which means that they are focused on the difference that outputs entail. Therefore, product development with agile methodologies is mainly driven by human values.

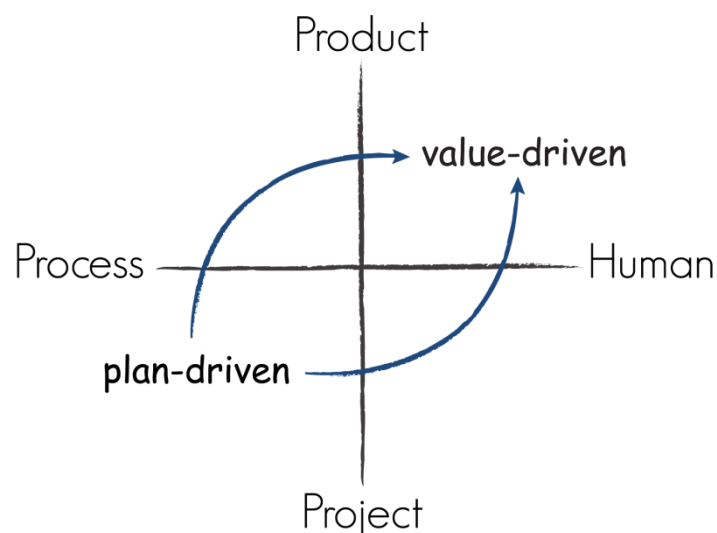


Figure 5: In ASD people focus on outcomes and how they can fulfil human needs through the outputs they produce.

In the context of agile methodologies, RE should be carried out iteratively during the whole development process instead of a closed phase in the beginning. Compared to established RE approaches ((Sommerville and Sawyer, 1997), (Pohl, 2010)), a list of prioritized requirements (*Product Backlog* (Schwaber, 2004)) is used instead of a requirements specification document. In agile RE, requirements are often treated as hypotheses that are continuously validated through frequent stakeholder and user feedback (Olsson and Bosch, 2015). In addition, requirements are regularly described from a user perspective in the form of epics and user stories (Cohn, 2004). The main agile RE activities are not clearly separated activities, since they can take place in parallel. Moreover, the wording of the activities is adapted to an agile environment: *discovery, refinement, prioritization, review, and documentation*, see (Schön et al., 2017d). The activities are repeated each iteration and only required information is elaborated before the next iteration starts. For this purpose, RE in agile environments is carried out just-in-time with a *Little Design Up Front* (Adikari et al., 2009). This leads to a kind of ad hoc nature of agile RE.

Agile techniques like *Continuous Delivery* (CD) (Humble and Farley, 2010) have an impact on the manner and the frequency of usability testing nowadays (Larusdottir et al., 2010). New information is given along the product development by the user and the system itself. This knowledge is processed during further steps and it conditions the decision-making process. Therefore, requirements are treated like assumptions, which are validated continuously.

Hybrid development models (Schön et al., 2016b), (Pfeiffer et al., 2016) with the integration of HCD, are applied with the aim to deliver competitive products with a suitable UX. Therefore, stakeholder and user involvement during RE are essential in order to establish a collaborative environment with constant feedback loops.

Agile RE is a cross-functional research area comprising areas like HCD, ASD, RE (see Figure 6). Contributing to the body of knowledge of agile RE implies considering research from all of the aforementioned areas.

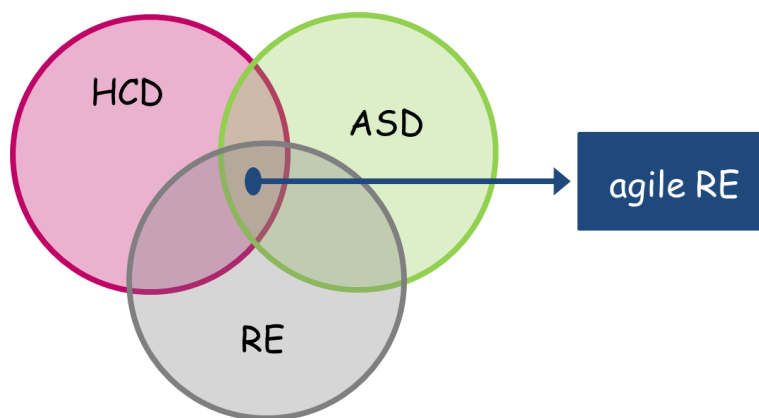


Figure 6: Venn diagram showing intersection of research areas influencing agile RE

Agile RE activities (see Table 1) are slightly different compared to the activities known from established RE (*elicitation, analysis and negotiation, and validation* (Sommerville and Sawyer, 1997)). Agile RE is carried out in an iterative manner and the activities are not separated in phases. In comparison to established RE, they can take place in parallel.

Table 1: Activities in agile RE

Agile RE activity	Description
Discovery	Eliciting new ideas/requirements
Refinement	Clarifying and analyzing new ideas/requirements
Prioritization	Measuring the value that the development will add to the system
Review	Checking if the requirement is implemented in the manner to deliver value
Documentation	Capturing discussion and decisions around the requirement

2.2. Relevant Work from other Research Groups

Besides the state of the art analysis of agile RE, this section presents the relevant work from other research groups in the field, which is related to the work of this PhD thesis. In this section, we want to highlight the most related authors by giving a short overview of their work. The relation between their work and this PhD thesis is additionally outlined in appropriate chapters of this work in the sections *Gap analysis of Related Work* (see 2.3.2, 4.2, 6.2).

2.2.1. Salah et al.

Salah et al. study the integration of UCD and Agile. Hence, Salah (Salah, 2011) contributes a framework for the integration of UCD and ASD. The framework consists of two components. On the one hand, a capability maturity model that measures the maturity of an organization in terms of the integration of UCD and ASD. On the other hand, a framework implementer, who is an individual person responsible for implementing the framework.

Salah et al. (Salah et al., 2016) elaborated the former part of the framework and presented a descriptive maturity model for integrating UCD and Agile. The maturity model is evaluated by means of a SLR (Salah et al., 2014) and evaluation forms (expert review).

The SLR conducted by Salah et al. (Salah et al., 2014) aims to identify challenges with the integration of UCD and ASD as well as propose practices, which are applied to address these challenges (see section 2.3.1).

In addition, Salah et al. (Salah et al., 2015) created patterns, which support the integration of UCD and ASD (see section 6.1).

2.2.2. Silva et al.

The research group around Silva et al. is investigating a similar research line like Salah et al. They conducted several SLRs in the field of the integration of HCD and Agile. For instance, Silva et al. (Silva da Silva et al., 2011) contribute a SLR with the aim to analyze the key aspects concerning the integration of UCD and Agile. In this SLR, they identified several artifacts, practices and needs, which are applied to integrate UCD and Agile (see section 2.3.1).

Based on this study, Silva et al. (Silva da Silva et al., 2012) propose a framework for integrating UX and agile. The framework was assessed in a field study, where they found some issues regarding the collaboration of UX designer and developers due to organizational conditions.

Furthermore, Silva contribute *Agile Usability Patterns* in collaboration with Bertholdo et al. (Bertholdo et al., 2014). These *Agile Usability Patterns* are focused on the early stages of HCD (see section 6.1).

Silva (Silva Da Silva et al., 2015) carried out multiple case studies to investigate how usability evaluation is carried out in agile environments. They identified new usability evaluation methods, which emerged with agile (e.g. peer review by pairing).

Another systematic mapping study was conducted in collaboration with Garcia, Silva and Selbach Silveira (Garcia et al., 2017). The aim of tis study was to analyze which artifacts are used to facilitate the communication in agile environments, where UCD plays an important role.

2.2.3. Méndez Fernández et al.

Méndez Fernández et al. study the field of RE process improvements. This field is very close to the one, which is investigated by this PhD work. However, their work has a slightly different viewpoint since our work is focused on RE in agile environments with strong focus on stakeholder and user involvement. Their viewpoint is more artifact-centered (*plan-driven*), whereas our viewpoint is more human-centered (*value-driven*).

Méndez Fernández et al. (Méndez Fernández et al., 2010) contribute a metamodel for artifact-based RE, which is inferred by two existing RE models in industry. The metamodel is focused on the topic of handling artifacts in RE.

Furthermore, they (Méndez Fernández et al., 2011) conducted a case study with an artefact-based RE approach, which defines two artefact types (business specification and requirements specification) as a domain specific interpretation of RE.

Méndez Fernández and Wieringa (Méndez Fernández and Wieringa, 2013) elaborated on the ideas and presented an artefact-based, problem-driven RE improvement approach. The approach starts with an empirical analysis of the improvement problem in terms of understanding the practice of modelling and documentation in a company. Based on this analysis, improvements are defined by means of an artefact model. Subsequent, a RE reference model is defined.

In addition, Méndez Fernández et al. (Méndez Fernández et al., 2016) work on the *Naming the Pain in Requirements Engineering (NaPiRE)* initiative, which study the status quo and problems in practical RE by means of an international series of surveys. The *NaPiRE* team (Méndez Fernández et al., 2017) consists of more than 50 researchers, who analyze problems in terms of RE in more than 200 companies worldwide by means of survey studies. In this context, they also publish a paper concerning RE problems in agile projects (Wagner et al., 2017).

2.3. State of the Art Agile Requirements Engineering

In this section, we will present the current state of the art of research related to agile RE with focus on stakeholder and user involvement. Therefore, we will present the results of our Systematic Literature Review (SLR) (Schön et al., 2017a) and discuss the findings of a revision of the study.

2.3.1. Summary of Related Literature Reviews

In the literature, many reviews are conducted in order to do research on ASD. The next paragraphs briefly summarize the most related ones.

Silva et al. (Silva da Silva et al., 2011) carried out a SLR on the integration of ASD and HCD and analyzed how usability issues are addressed in agile projects. They included a comprehensive classification process based on a system covering research-related and content-related information. The authors identified the following key aspects, which play an important role for the integration: *little upfront design, prototyping, user stories, user testing, inspection evaluation and one sprint ahead*. Besides, they presented a process model for the integration of ASD and HCD that took into account their findings.

Salah et al. (Salah et al., 2014) addressed a similar area. Their review aimed to identify challenging factors for the integration of ASD and HCD. They presented the challenges in a very understandable manner with good examples. Additionally, they explored practices and success factors to face these challenges. The reported challenges are: *lack of allocated time for upfront activities, difficulty of modularization, optimizing the work between developers and HCD practitioners, performing usability testing and lack of documentation*.

Brhel et al.'s (Brhel et al., 2015) literature review, examined hybrid development models, such as ASD and HCD. Their main objective was to capture the state of the art of the integration of ASD and HCD. Compared to (Silva da Silva et al., 2011) and (Salah et al., 2014), they addressed a more holistic view of ASD. Thus, they focused on four dimensions (process, people/social, technology and practices) with a coding system and five derived principles: *1) separate product discovery and product creation, 2) iterative and incremental design and development, 3) parallel interwoven creation tracks, 4) continuous stakeholder involvement, and 5) artifact-mediated communication*. Furthermore, they contributed to a classification system for existing work in the field of user-centered ASD.

Heikkilä et al. (Heikkilä et al., 2015) conducted a mapping study of RE in ASD in order to identify gaps in the scientific knowledge. Their results indicate that the definition of agile RE is vague. Furthermore, they identified the following benefits from agile RE: *lower process overheads, improved requirements understanding, a reduced tendency to overallocate development resources, responsiveness to change, rapid delivery of value, and improved customer relationships*. The following problems were identified: *use of customer representatives, insufficiency of the user story format, prioritization of requirements, growing technical debt, tacit requirements knowledge, and imprecise effort estimation*.

The main purpose of Inayat et al.'s (Inayat et al., 2015) literature review is to make clear Agile RE challenges and practices, including a good discussion on related work. Moreover, they aimed to understand how traditional RE problems are resolved using Agile RE. In summary, they provided 17 commonly used practices and also practical challenges that agile teams had to face. The practices are: *Face-to-face communication, customer involvement and interaction, user stories, iterative*

requirements, requirement prioritization, change management, cross-functional teams, prototyping, testing before coding, requirements modeling, requirements management, review meetings and acceptance tests, code refactoring, shared conceptualization, pairing for requirements analysis, retrospectives and continuous planning.

Soares et al. (Soares et al., 2015) combined a literature review with an exploratory study. They analyzed difficulties while working with requirements in an agile environment, particularly, causes that can lead to documentation debt (e.g. missing, inadequate or incomplete requirements). They contribute with their work to an important research topic, for documentation in ASD is often treated in an inadequate manner. The authors defined 10 difficulties that occur when identifying and managing agile requirements: requirement prioritization, non-functional requirements identification, lack of information, volatility of requirements, requirements definition, dependence between requirements, to predict impacts of changes, user dependence, communication and collaboration with users, and requirements validation. Furthermore, they uncovered difficulties when using user stories instead of use cases.

2.3.2. Gap Analysis in Related Literature Reviews

To sum up, it can be said that related literature reviews cover many aspects of Agile RE. Nevertheless, analyzing the existing work, we observed some shortcomings. Silva et al. (Silva da Silva et al., 2011) and Salah et al. (Salah et al., 2014) worked on the integration of HCD and ASD. In this context, they studied the collaboration between HCD specialists and developers but they did not pay the same attention to stakeholder and user involvement. The other published reviews (Brhel et al., 2015), (Heikkilä et al., 2015), (Inayat et al., 2015) and (Soares et al., 2015) identified stakeholder and user involvement as one of the key aspects in ASD, although they only presented partially how this problem might be solved. As they treated this problem as one out of many, we consider that they only scratch the surface.

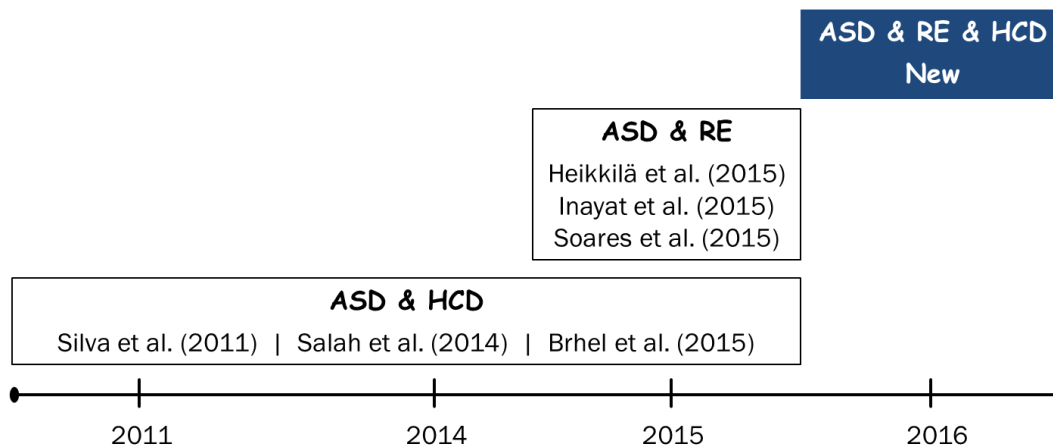


Figure 7: Related work has not investigated Agile RE from the perspective of HCD up to now

Since human beings and their values play one of the most important roles in value-driven organizations (see Figure 5), it is necessary to further investigate this aspect in the agile RE field. To this end, we conducted this SLR. However, to the best of our knowledge, no systematic review has

previously been published³ which investigates RE with focus on stakeholder and user involvement in agile environments (see Figure 7).

2.3.3. Review Method

Appropriate guidelines have been followed for conducting the systematic review, particularly the guidelines for SLRs in Software Engineering by Kitchenham and Charters (Kitchenham and Charters, 2007). According to these guidelines, our SLR consists of three main phases. Figure 8 shows the most important stages of each phase.

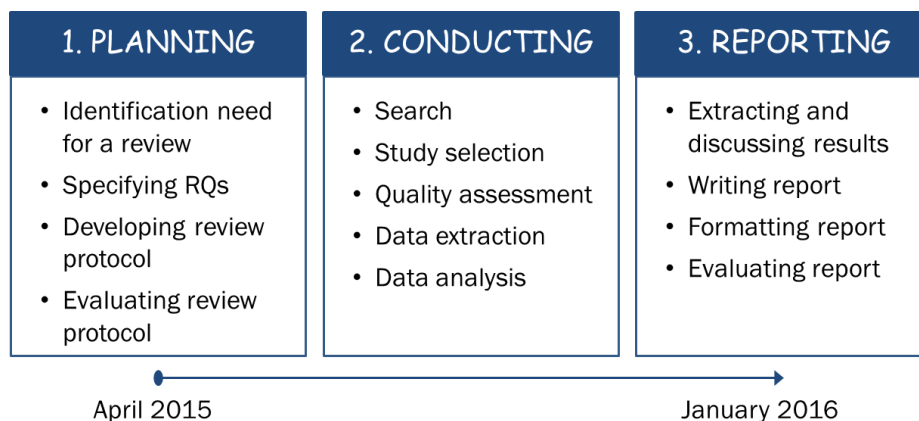


Figure 8: Phases of a SLR (cf. (Kitchenham and Charters, 2007))

Due to the high number of retrieved studies, we used the software Mendeley and excel sheets in order to manage information obtained in an efficient manner.

2.3.3.1. Objectives and Research Questions

Our goal was to gather the state of the art of the literature related to RE, by looking at stakeholder and user involvement in agile methodologies. Therefore, we created three complementary research questions (RQ), which are specified by the following sub-criteria:

RQ 1.1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

On one hand, we analyzed whether the existing approaches involve stakeholders and users directly into the development process. On the other hand, our aim was to study whether the approaches apply a process model for the involvement. In addition, we queried what kind of methods they use in order to gather data. With regard to the agility of the existing approaches, we examined whether there are iterations along the development process.

³ Date of publication: January 2017

RQ 1.2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

Concerning this RQ, we analyzed the included studies in terms of methodologies that are used to handle the user perspective within agile environments. Furthermore, we investigated how the knowledge of user requirements is shared among stakeholders.

RQ 1.3: What are the common ways for requirements management in ASD?

In terms of this RQ, our aim was to investigate what types of artifacts are used and how they are utilized. Moreover, we wanted to discover whether the documentation of requirements is understandable without further knowledge in order to be able to work in a collaborative manner. In addition, we examined the treatment of non-functional requirements.

2.3.3.2. Protocol Development

In the beginning of the planning phase we undertook an initial informal search for other SLRs concerning a similar scope of this field. The relevant ones are presented in section 2.3.1 as related works. During the informal search we found a few relevant studies, which fit our research objectives. Accompanied by the already identified studies, we used these SLRs as basis to create our RQs and to develop our review protocol (see Appendix II (Schön et al., 2016a)), which was carried out in an incremental and iterative way by two independent researchers.

2.3.3.3. Search Strategy and Data Sources

Subsequent to the definition of the research objectives and the RQs we elaborated our research strategy. Therefore, we selected keywords, created a search string and specified the search space and the search process that was used to reduce the number of papers.

In a first step, we extracted a set of keywords from the studies we had and matched it with our research objectives. Secondly, we identified alternative spellings and synonyms. Since the search process is a critical aspect, we had to optimize the keywords iteratively. Thus, we defined a set of keywords, tested them in various databases and finally, we refined them. The final list can be found in Table 2.

Table 2: Keywords used for search

Category	Keywords
Agile methodology	agile, scrum, kanban, extreme programming, lean
Human Computer Interaction	hci, hmi, ucd, usability, human, user
Requirements Engineering	requirements engineering

Afterwards we connected the keywords with Boolean operators and designed our search string as follows:

(agile OR scrum OR kanban OR “extreme programming” OR lean) AND (hci OR hmi OR ucd OR usability OR human OR user) AND (“requirements engineering”)

The search space included digital libraries, specific journals and conference proceedings. It is worth mentioning that every digital library has its own characteristics concerning its search engines.

To this purpose, we had to adapt the search string for every library. The search was documented in a separate document that included the following information for every single digital library: name, search strategy, date of search, years covered by the search and a documentation regarding the adaption of each search string to every single trial. Table 3 shows an extract from this.

Table 3: Search space

Digital library	Search strategy	Date of search
Google scholar	full text	2015-06-13
Science direct	abstract, title and keywords	2015-06-10
SpringerLink	abstract and keywords	2015-06-12
Scopus	abstract, title and keywords	2015-06-10
IEEEXplore	abstract and keywords	2015-06-12
ACM	abstract and keywords	2015-06-11

At the beginning, the search results showed a high amount of papers (42,808 findings). In order to reduce the results, we carried out the search process in different phases (see Figure 9).

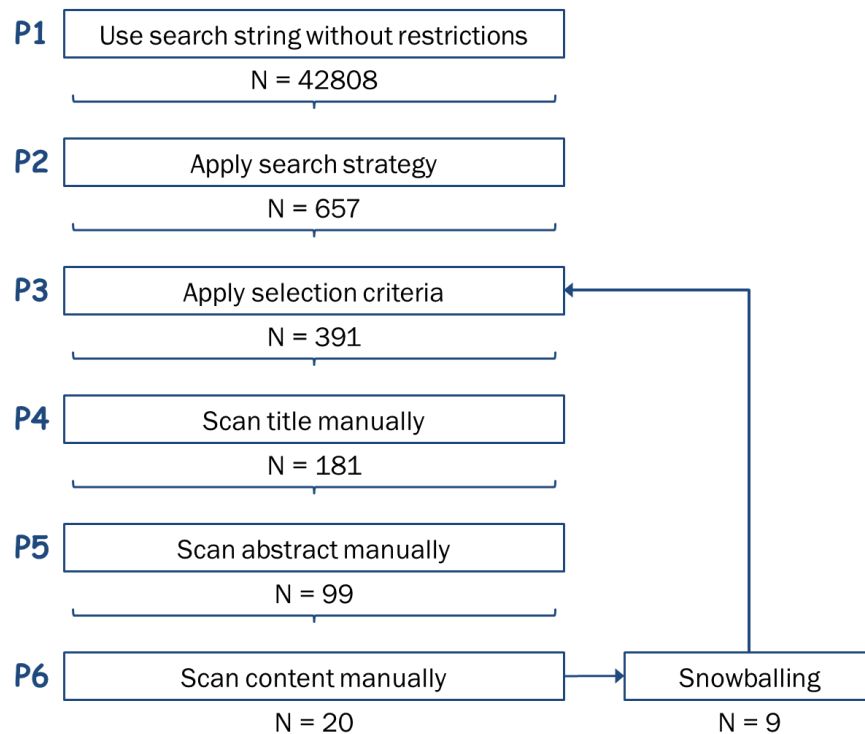


Figure 9: Search process comprising six phases and snowballing. After analyzing the data, we included 27 papers into our study.

In addition to the initial search process, we started snowballing for identified papers at P6 (see Figure 9). We applied forward snowballing (search in papers that cited the paper) and backward snowballing (search in the reference list of the paper) (Jalali and Wohlin, 2012).

Snowballing helped us to identify a total amount of 965 more papers (forward snowballing N = 355; backward snowballing N = 610). For these papers, we also used the search process and started at P3. At the end of this second search process, we identified nine papers that were taken into account for data extraction.

2.3.3.4. Study Selection

The selection criteria were divided into inclusion and exclusion criteria and were applied to P3 of the search process.

Inclusion criteria were: papers written in English; papers published in between 1995-2015; papers under peer review; papers presenting approaches to integrate user into agile development processes; papers related to Agile RE; papers associated with agile requirements documentation; and specific book chapters.

Exclusion criteria are: no full books; papers whose full text were not available; papers with results that had been already published; papers that were not focused on agile development; papers only presenting ideas, lessons learnt, recommendations or guidelines; papers introducing tools whose underlying methodology was not comprehensibly described (black box); and studies, whose primary focus moved away from agile methodologies.

Due to the high amount of findings at P3, we reduced the time period for including papers from 1995-2015 to 2007-2015. Our aim was to cope with the current state of the art and not analyzing the evolvement over time. At the end of data extraction we found 19 papers that were obviously relevant to our study and 9 where we proofed whether they included relevant information for answering our RQs. If a paper contained relevant information, we would include it in the study. In light of this, 8 papers were only useful for the study. The other one missed relevant details, so that we discarded it during the data extraction phase.

As a result of the snowballing process, we identified a few authors, who published more than one relevant publication. Therefore, we had to identify the relevant papers we aimed to include in our study. For this purpose, we contacted the authors to either include the most cited paper or the latest one dealing with the approach.

2.3.3.5. Quality Assessment

We elaborated a quality checklist to assess the individual studies. There were three available answers for every question (see Table 4). This checklist was used to evaluate the quality of the included studies.

Table 4: Quality checklist

Item	Assessment Criteria	Score	Description
QA1	Is the proposal validated?	-1	No, it is not validated
		0	Partially, it is validated in a laboratory or only parts of the proposal are validated
		1	Yes, by a case study.
QA2	Does the study present a detailed description of the approach?	-1	No, details are missing
		0	Partially, if you want to use the approach, you need to read the references
		1	Yes, the approach can be used with presented details
QA3	Does the study present a personal opinion piece or viewpoint?	-1	Yes, it does.
		0	Partially, since related work is explained and paper is set into a specific context
		1	No, the paper is based on research
QA4	Has the study been cited by other authors?	-1	No, no one cited the study
		0	Partially, between 1-5 articles cited the study
		1	Yes, more than 5 articles cited the study
QA5	Includes the paper a clear statement of the aims of the study?	-1	No, aims are not described.
		0	Partially, aims are described but unclearly
		1	Yes, aims are well described and clear

2.3.3.6. Data Extraction and Analysis

According to Kitchenham and Charters' (Kitchenham and Charters, 2007) guidelines, a form for data extraction was set up. We used Mendeley in order to mark text passages and ratings. That software also supported the data extraction in regard to defined attributes from the protocol:

- Basic information: title, authors, publication date, DOI and URL
- Publication data: journal, conference, date (of conference), publisher, volume, issue, pages, keywords and abstract

In addition to Mendeley, we set up a data collection form in Excel to take out the following data:

- Research method: e.g. experiment, quasi-experiment, lessons learnt
- case study, opinion survey or tertiary study
- Research approach: deductive, inductive or hybrid
- Agile method: e.g. Scrum, XP, Kanban or hybrid
- Method: e.g. pair-programming, stand up meeting or usability pattern
- Artifacts: e.g. user stories, Kanban board, personas, prototypes
- Short summary
- Results and contributions
- Personal assessment
- Number of included references
- Number of papers that cited the study

All identified papers were taken into consideration to carry out the data extraction process, where we found that taking out data in line with the form was not always possible because of the way studies were reported. In cases where the required information was not provided or not clearly reported we used “n.a.”, in order to fill in the form. On one hand, we extracted quantitative data (e.g. publication channel or research method) and on the other hand, we extracted qualitative data (e.g. content or short summary).

We used 3-point Likert items in order to weigh the single items (*covers the criteria, covers the criteria partially or do not cover the criteria*) with the aim to answer the RQs with its sub-criteria. We chose the option *covers the criteria partially*, in cases where one item could not clearly be answered by the study.

2.3.4. Results and Discussion – State of the Art

We included in our work 27 identified relevant studies. Firstly, we describe characteristics of the studies and show quantitative data (e.g. publication channel, research method or quality overall). Secondly, we state our findings related to the RQs.

2.3.4.1. Summary of Studies

Concerning the publication channel, the studies were published in conference proceedings or scientific journals. In comparison, 21 (78%) of the included studies were published in conference proceedings and only 6 papers (22%) appeared in scientific journals.

Table 5 presents the distribution of the studies’ underlying research method. In summary, 19 studies (70%) used case studies. For this kind of publication type, we distinguished among Case Study, Multi-Case Study and Case Study in the laboratory. Most of the studies (15 paper, 56%) were carried out as single Case Study in economic enterprises. Furthermore, two works (7%) were executed in laboratories or in a simulated context. In addition, two studies (7%) were carried out as a Multi-Case Study and in five papers (19%) the authors described an approach from a theoretical viewpoint. In those papers, they did not use further research in order to validate their approaches. However, in some cases that might be a starting point for their future research. In contrast to this, results were clearly presented in one of the papers as starting point for future research activities; in consequence the authors published it as a research perspective. Furthermore, one study used an experiment and another one used semi-structured interviews as a main research method.

In conclusion, it can be stated that RE in agile methodologies is often investigated in real life context and this research field is very close to existing work practices in companies. We are aware about the fact that the results of a single case study might not be generalized to other settings and that this may have an impact on the interpretation of our results.

Table 5: Distribution according to research methods

Research Method	Paper total	Percentage
Case Study	15	56%
Case Study in Laboratory	2	7%
Multi-Case Study	2	7%
Description of Approach	5	19%
Experiment	1	4%
Research Perspective	1	4%
Semi-Structured Interviews	1	4%

We used the quality checklist presented in Table 4 in order to evaluate each study. The overall results from the quality assessment are shown in Figure 10.

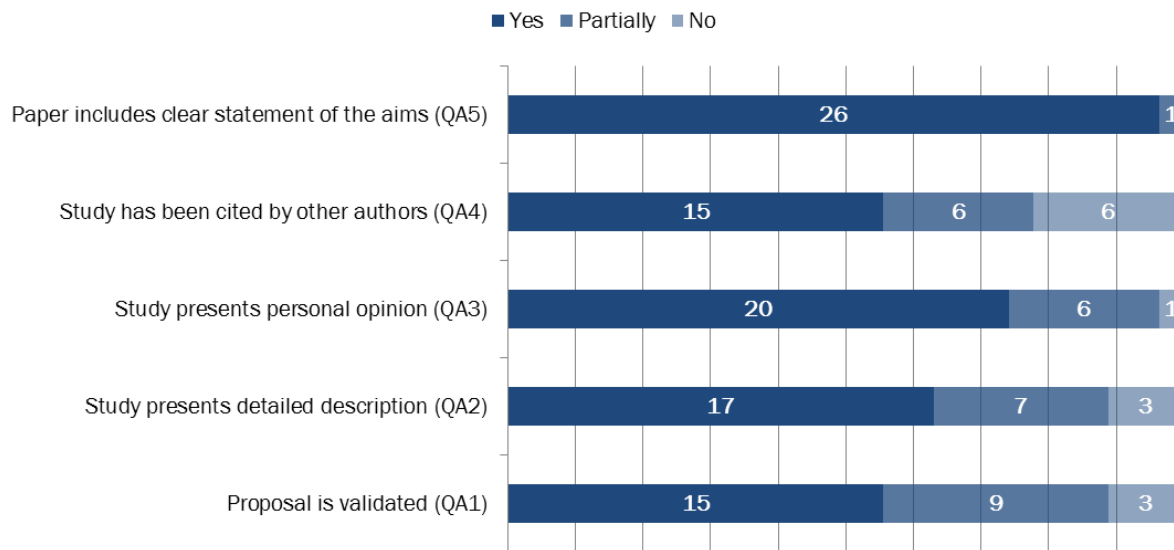


Figure 10: Quality assessment

The first criterion (QA1) examines whether the proposal is validated. For 15 papers this is true, as they used case studies in order to validate their proposal. Nine papers either validated their proposals in a laboratory or assessed only parts of the proposal. We also included three papers whose proposals were not validated at all. With QA2 we confirmed whether the study presented a detailed description of the approach. In 17 papers the approach is described with enough details, so that other researchers could use it. In comparison, when the approach of seven papers should be used again, included references have to be read. Two papers missed details. QA3 queried whether the study provided a personal opinion or viewpoint. 20 out of the 27 papers were based on a clearly defined research design. For 6 papers, the related work was explained and the paper was set into context. Nonetheless, there was also one study that did not clearly describe the research method. QA4 wondered how many times the study had been cited in other papers. For this purpose, we used the number of citation from Google scholar (Assessment date 2015-11-20). 15 studies had more than five citations in other papers. Six papers had been cited in among 1-5 articles and six papers had no citation until the assessment date. With the last criterion (QA5) we tested if the

aims of the study were included in the paper. In 26 of the works, the aims were well described and clear. Only one paper lacked presenting the aims, since they were described very unclearly.

To sum up, six papers fulfilled every quality criterion ((Cajander et al., 2013), (Losada et al., 2013), (Abdullah et al., 2011), (Kautz, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008)). We have to be aware that the results might be different at publication date of the results, due to the different number of citations at assessment date that QA4 required.

2.3.4.2. (RQ 1.1) What Approaches Exist, Which Involve Stakeholders in the Process of RE and Are Compatible With ASD?

Table 6 presents the results from the evaluation of the sub-criteria related to RQ1. Therefore, we list the studies that clearly have a positive answer, “Yes”. Additionally, the distribution according to the items is also shown.

Table 6: Sub-criteria RQ1

ID	Sub-criteria	Covered by study	Overall
C1.1	Stakeholders are involved directly.	(Bellucci et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Dragicevic et al., 2014), (Liskin et al., 2014), (Rivero et al., 2014), (Wanderley et al., 2014), (Cajander et al., 2013), (Losada et al., 2013), (Kamthan, 2013), (Maguire, 2013), (Abdullah et al., 2011), (Näkki et al., 2011), (Mahmud and Veneziano, 2011), (Bourimi et al., 2010), (Kautz, 2010), (Lee et al., 2010), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 77% Part = 19% No = 4%
C1.2	The user is involved directly.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Olsson and Bosch, 2015), (Rivero et al., 2014), (Wanderley et al., 2014), (Cajander et al., 2013), (Kamthan, 2013), (Losada et al., 2013), (Maguire, 2013), (Näkki et al., 2011), (Mahmud and Veneziano, 2011), (Kautz, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 56% Part = 33% No = 11%
C1.3	They use a process in order to involve stakeholders.	(Bellucci et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Dragicevic et al., 2014), (Liskin et al., 2014), (Nawrocki et al., 2014), (Rivero et al., 2014), (Wanderley et al., 2014), (Cajander et al., 2013), (Kamthan, 2013), (Losada et al., 2013), (Maguire, 2013), (Issa and AlAli, 2011), (Näkki et al., 2011), (Mahmud and Veneziano, 2011), (Bourimi et al., 2010), (Kautz, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 78% Part = 19% No = 4%
C1.4	There are iterations during the development process.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Dragicevic et al., 2014), (Liskin et al., 2014), (Nawrocki et al., 2014), (Rivero et al., 2014), (Wanderley et al., 2014), (Cajander et al., 2013), (Losada et al., 2013), (Kamthan, 2013), (Maguire, 2013), (Näkki et al., 2011), (Bourimi et al., 2010), (Kautz, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 70% Part = 30% No = 0%
C1.5	They use methods in order to gather data.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Dragicevic et al., 2014), (Nawrocki et al., 2014), (Adikari et al., 2013), (Cajander et al., 2013), (Maguire, 2013), (Näkki et al., 2011),	N = 27 Yes = 70% Part = 30% No = 0%

		(Issa and AlAli, 2011), (Mahmud and Veneziano, 2011), (Kautz, 2010), (Lee et al., 2010), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	
--	--	--	--

Subsequent to the quantitative analysis presented in Table 6, the following paragraphs will put forward the highlights we found in the included studies concerning sub-criteria by means of a qualitative analysis.

Stakeholder and User Involvement

Bellucci et al. (Bellucci et al., 2015) combined XP with co-design sessions in order to develop a product with strong user involvement. They explored in a field study, how users interact and work with a prototype. Based on their findings, the prototype was developed iteratively. The authors considered this approach as a tactic to quickly deploy an evolving prototype.

Harbers et al. (Harbers et al., 2015) studied the application of a *Value Story workshop* for the elicitation process of user stories. The aim of this workshop is to embed stakeholder values into the RE process. Therefore, direct and indirect stakeholders have to be identified. In a second step, the values of each stakeholder group are revealed. Then, a situation for each value is provided and stakeholder needs in this situation are analyzed.

Olsson et al. (Olsson and Bosch, 2015) derived a conceptual model from multiple case studies that emphasized the need for combining qualitative feedback in early stages of development with quantitative customer observation in later stages of development. In the *Qualitative/Quantitative Customer-driven Development (QCD) model* requirements are treated as hypotheses that are validated through continuous customer feedback.

Several studies provided additional roles, which should be applied to an agile environment in order to address communication gap or take over responsibility for particular tasks. Dragicevic et al. (Dragicevic et al., 2014) claimed that *business users* could collect necessary data and documents (e.g. templates, scans of documents or screens) and provide information to developers who model those data in UML.

Moreover, Lee et al. (Lee et al., 2010) stressed the role of an *Agile-HCD specialist (AUS)*, which has to be responsible for bridging the communication gap between developer and UX designer. They also present a *usability-pattern-based requirement-analysis method* that help AUSs to do their tasks of requirements specification.

Kautz (Kautz, 2010) wondered how customers and users participate in ASD with Participatory Design. He reports that customer and users were involved indirectly and directly through different kind of activities. Additionally, the role of an *onsite customer* was applied and they had weekly feedback loops. Kautz states that one of the important benefits of the frequent feedback loops is that misunderstandings are detected in time and changes can be applied early before they grow into larger problems. For this reason, the user generates a feeling of trust that has impact on the development.

Collaboration and shared understanding are essential to ASD (Beck et al., 2001). To this end, and attending to the results of a Multi-Case Study, Ramesh et al. (Ramesh et al., 2010) described informal and frequent communication as the core of agile RE. They stated that customers were directly involved in each iteration. Requirements were elicited, refined and validated through face-

to-face communication with the customer. They claim requirements analysis as a social–political process that depends on human interaction and is influenced by several contextual factors.

Data Gathering in Agile RE

The results of sub-criteria C1.5 (Table 6) revealed that 70% of the studies reviewed used methods in order to gather data. In addition to traditional methods known from agile methodologies (e.g. planning or reviews) reported by (Buchan, 2014) and (Ramesh et al., 2010), the reviewed studies included further methods for data gathering within Agile RE.

For instance, we found representative examples in the following studies: Bellucci et al. (Bellucci et al., 2015) applied weekly co-design sessions as a meeting point between users and designers. During these sessions, researchers collected impression and feedback concerning users' experience with the system and usage scenarios. They also gathered information about non-usage and misusing of implemented features. Similarly, Kautz (Kautz, 2010) studied *Participatory Design* activities within ASD. He reported that data gathering with customer and user involvement took place on an ongoing basis. Communication was structured through planning games, presentation of working software and acceptance tests.

Lucia et al. (Lucia and Qusef, 2010) present an overview regarding agile RE. *Interviews, Brainstorming, Ethnography* and *Use Case analysis* are the most important elicitation techniques from their point of view.

Näkki et al. (Näkki et al., 2011) describe users as a source of ideas and looked at them as decision makers throughout the design process. For this purpose, they enabled *Lead users* to participate via online co-creation tools with social media mechanisms.

Considering that analyzing the context of use is an important activity to achieve a human-centered process, several studies recommend using a structured process for this activity. They describe how to carry out a *Contextual Inquiry* (Beyer and Holtzblatt, 1998) as an appropriate manner (Obendorf and Finck, 2008), (Maguire, 2013). With regard to Human-Centered Design, Maguire (Maguire, 2013) propose different methods following activities from ISO 9241-210:2010. In addition to a Contextual Inquiry, he suggests to accomplish a *stakeholder analysis* to explore the context of use. Furthermore, he proposes *surveys, interviews, discussions, focus groups, competitor analysis* and *user journeys* for specifying user requirements. Maguire recommends gathering data with *conceptual design meetings* and *co-design workshops* during the creation of design solutions. For the evaluation, he highlights the use of *user walkthroughs* and *usability tests*.

2.3.4.3. (RQ 1.2) Which Agile Methodologies, Which Are Capable of Presenting the User Perspective to Stakeholders, can be Found?

Table 7 shows the results from the evaluation of the sub-criteria related to RQ2. Therefore, we list the studies that clearly answered the questions positively. Moreover, we display the distribution in terms of the items.

Table 7: Sub-criteria RQ2

ID	Sub-criteria	Covered by study	Overall
C2.1	The proposal used a methodology.	(Bellucci et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Dragicevic et al., 2014), (Nawrocki et al., 2014), (Rivero et al., 2014), (Wanderley et al., 2014), (Adikari et al., 2013), (Kamthan, 2013), (Losada et al., 2013), (Maguire, 2013), (Farid, 2012), (Issa and AlAli, 2011), (Näkki et al., 2011), (Bourimi et al., 2010), (Kautz, 2010), (Lee et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 25 Yes = 80 % Part = 16% No = 4%
C2.2	The knowledge about the user requirements is shared among the stakeholders.	(Bellucci et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Liskin et al., 2014), (Rivero et al., 2014), (Cajander et al., 2013), (Maguire, 2013), (Kamthan, 2013), (Abdullah et al., 2011), (Bourimi et al., 2010), (Kautz, 2010), (Lee et al., 2010), (Lucia and Qusef, 2010), (Mommel et al., 2007)	N = 27 Yes = 52% Part = 48% No = 0%

Subsequent to the quantitative analysis presented in Table 7, the following sub-sections will present the highlights we found in the included studies concerning sub-criteria by means of a qualitative analysis.

User Perspective in ASD

Cajander et al. (Cajander et al., 2013) interviewed 21 IT professionals in order to analyze the user perspective in ASD. On one hand, they found that the responsibility for the user perspective is often unclear and on the other hand, they discover that in some cases the user perspective is neither discussed nor described. Furthermore, Cajander et al. state that ad hoc nature of user involvement and design feedback exist in agile projects. Moreover, they conclude that new usability methods arose because of the agile requirements (e.g. speed, efficiency or focus on deliverables instead of documentation).

Methodologies in Agile RE

Several studies deal with methodologies that are used to extend agile methodologies like scrum and XP, with the aim to better understand the user perspective.

Human-Centered Design (HCD): Maguire (Maguire, 2013) extended the HCD framework (International Organization for Standardization, 2010) for agile development. There are four main activities in HCD, which are performed iteratively: *a) understand and specify the context of use, b) specify user requirements, c) produce design solutions and d) evaluate designs against requirements.* Maguire suggests different methods in order to perform each activity. Additionally, he recommends producing clearly defined artifacts based on the gathered information.

Design Thinking: A study by Adikari et al. (Adikari et al., 2013) propose a framework based on three methodologies: *Design Thinking, UX design* and *ASD*. A real world system context was being explored with other relevant systems using Design Thinking. As a result, a *reframed context* was build. They report that the knowledge of the reframed contexts could be used to create products, systems or services using UX design and ASD.

Contextual Inquiry: Several studies (e.g. (Obendorf and Finck, 2008) and (Maguire, 2013)) state that performing a *Contextual Inquiry* (Beyer and Holtzblatt, 1998) is useful to explore the user perspective and to gather data concerning both, users and context of use.

Participatory Design: Kautz (Kautz, 2010) studied how customers and users participated in ASD with *Participatory Design*. Kautz focused on the role of the customers and users and how they were involved through different activities in design and development. Similarly, Bellucci et al. (Bellucci et al., 2015) investigated an approach to design with and by the user. For this reason, they carried out a field study to evaluate how users interacted with a prototype. Based on their findings, the prototype was developed iteratively through constant feedback loops. In addition, Näkki et al. (Näkki et al., 2011) worked on an application of a lead-user approach in the context of ASD. The chosen lead users participated actively in the innovation process through idea generation as well as in all phases of the development process via online co-creation tools. With regard to Participatory Design, Olsson et al. (Olsson and Bosch, 2015) developed the *QCD* model based on a multi-case study. They confirm that it is important to combine qualitative and quantitative feedback techniques in order to achieve continuous customer validation. The authors treat requirements as hypotheses that are validated with customers before they are taken into account for development.

Shared Understanding

Abdullah et al. (Abdullah et al., 2011) cope with communication patterns in an agile team and in particular, how communication and collaboration supported RE activities (gathering, clarifying and evolving) in an agile environments. They built the concept of shared conceptualization, which mean that the development team shares a common understanding of the requirements, which is deeper than shared understanding. They describe that there is a link between communication and memory. Little information about a requirement was documented on a story card, but members of the agile team built a related concept in their minds that was based on discussion concerning the requirement.

A study by Buchan (Buchan, 2014) also provides insights into the concept of shared understanding. Buchan developed a theory of *shared understanding of requirements* (SUR). He states that SUR is a specialized form of a *Team Mental Model* with focus on RE. Furthermore, he defines two activities in SUR development: 1) uncovering a gap collaboratively, and 2) addressing this gap to achieve a new state of SUR. The goal is to enable team members achieve a consistent understanding of the requirements.

2.3.4.4. (RQ 1.3) What are the Common Ways for Requirements Management in ASD?

Table 8 presents the results from the evaluation of sub-criteria related to RQ3 as well as it lists the studies that clearly answered the questions positively. In addition, it also shows the distribution according the items.

Table 8: Sub-criteria RQ3

ID	Sub-criteria	Covered by study	Overall
C3.1	They are using artifacts.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Dragicevic et al., 2014), (Liskin et al., 2014), (Nawrocki et al., 2014) (Wanderley et al., 2014), (Rivero et al., 2014), (Cajander et al., 2013), (Losada et al., 2013), (Maguire, 2013), (Abdullah et al., 2011), (Issa and AlAli, 2011), (Mahmud and Veneziano, 2011), (Näkki et al., 2011), (Bourimi et al., 2010), (Lee et al., 2010), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Kautz, 2010), Farid, (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 93% Part = 7% No = 0%
C3.2	The documentation is understandable without further knowledge.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Harbers et al., 2015), (Olsson and Bosch, 2015), (Buchan, 2014), (Liskin et al., 2014), (Rivero et al., 2014), (Cajander et al., 2013), (Maguire, 2013), (Abdullah et al., 2011), (Issa and AlAli, 2011), (Näkki et al., 2011), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Kautz, 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N = 27 Yes = 63% Part = 33% No = 4%
C3.3	They distinguish between functional and non-functional requirements.	(Harbers et al., 2015), (Dragicevic et al., 2014), (Nawrocki et al., 2014), (Rivero et al., 2014), (Cajander et al., 2013), (Losada et al., 2013), (Maguire, 2013), (Farid, 2012), (Issa and AlAli, 2011), (Bourimi et al., 2010), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Mommel et al., 2007)	N = 26 Yes = 50% Part = 42% No = 8%

Subsequent to the quantitative analysis presented in Table 8, the following sub-sections will present the highlights we found in the included studies concerning sub-criteria by means of a qualitative analysis.

Artifacts in Agile RE

Artifacts are used for communication, elaboration, validation and documentation of requirements in agile environment. In sum, we identified 57 different artifacts mentioned in the included studies. 17 out of 57 are mentioned more than twice (see Table 9).

Table 9: Artifacts in Agile RE

Artifact	Description	Reference	Perc.
User Story	User story is a description of a feature written from the perspective of the person who needs this. It consists of a written text, conversation about it and acceptance criteria.	(Blomkvist et al., 2015), (Harbers et al., 2015), (Buchan, 2014), (Liskin et al., 2014), (Nawrocki et al., 2014), (Rivero et al., 2014), (Wanderley et al., 2014), (Kamthan, 2013), (Maguire, 2013), (Farid, 2012), (Näkki et al., 2011), (Kautz, 2010), (Lee et al., 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008)	N=27 56%

Prototype	Prototype is a software application model that supports the evaluation of design alternatives and communication.	(Bellucci et al., 2015), (Blomkvist et al., 2015), (Nawrocki et al., 2014), (Rivero et al., 2014), (Maguire, 2013), (Kamthan, 2013), (Losada et al., 2013), (Näkki et al., 2011), (Bourimi et al., 2010), (Lucia and Qusef, 2010), (Ramesh et al., 2010), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N=27 41%
Use Case	Use case describes an action or event steps, which are needed to achieve a goal.	(Harbers et al., 2015), (Nawrocki et al., 2014), (Losada et al., 2013), (Farid, 2012), (Issa and AlAli, 2011), (Lucia and Qusef, 2010), (Mommel et al., 2007)	N=27 26%
Scenario	Scenario is a textual representation of a problem and describes the interaction between user and system in a specific context.	(Blomkvist et al., 2015), (Harbers et al., 2015), (Nawrocki et al., 2014), (Maguire, 2013), (Obendorf and Finck, 2008), (Mommel et al., 2007)	N=27 22%
Story Card	Story Card is a physical representation for the written text and shows details from a user story.	(Buchan, 2014), (Farid, 2012), (Abdullah et al., 2011), (Kautz, 2010), (Ramesh et al., 2010)	N=27 22%
Persona	Persona describes a fictitious person that represents a larger part of the target group.	(Blomkvist et al., 2015), (Liskin et al., 2014), (Kamthan, 2013), (Maguire, 2013)	N=27 15%
Vision	Vision is an abstract description of the overarching goal that guides product development and aligns development, business people and other stakeholders.	(Buchan, 2014), (Liskin et al., 2014), (Mommel et al., 2007)	N=27 11%
UML diagram	Unified Modeling Language (UML) provides a standard to visualize the design of a system.	(Dragicevic et al., 2014), (Bourimi et al., 2010), (Lee et al., 2010)	N=27 11%
Storyboard	Storyboard presents the workflow of the user in a sequence of pictures.	(Näkki et al., 2011), (Lucia and Qusef, 2010), (Obendorf and Finck, 2008)	N=27 11%
Task	Each of the parts into which one user story is split. It describes more technical requirements.	(Liskin et al., 2014), (Obendorf and Finck, 2008)	N=27 7%
Kanban board	Kanban board visualizes the progress of a requirement through the workflow of the development team.	(Buchan, 2014), (Liskin et al., 2014)	N=27 7%
UI pattern	UI pattern describes an abstract solution for recurring design problems and give inspiration to designer.	(Lee et al., 2010), (Mommel et al., 2007)	N=27 7%
Essential use case	Essential use case describes user tasks and it represents a simplified and generalized form of use cases.	(Lee et al., 2010), (Mommel et al., 2007)	N=27 7%

Picture	Picture is a visual representation (e.g. photograph or painting)	(Bellucci et al., 2015), (Näkki et al., 2011)	N=27 7%
Video	Video consists of a sequence of images processed electronically that are seen in a recording and are displayed on a screen.	(Bellucci et al., 2015), (Näkki et al., 2011)	N=27 7%
Mind Map	Mind map is a diagram used to visualize and organize information.	(Wanderley et al., 2014), (Mahmud and Veneziano, 2011)	N=27 7%
UI specification	Written specification that describes the UI of a system. The text is enriched by mock-ups or icons, among other elements.	(Blomkvist et al., 2015), (Maguire, 2013)	N=27 7%

At this point, we would like to highlight some key artifacts (usage <20%).

User stories are the most frequent used artifact in ASD. The included studies describe how they can be created and represented. Näkki et al. (Näkki et al., 2011) use the concept of *needs-based user stories*. Therefore, they collected users' everyday needs and challenges regarding a specific domain. Users were involved during requirements elaboration, by commenting and rating features in order to allow the prioritization of features. Harbers et al. (Harbers et al., 2015) suggest using a *Value Story workshop* to embed stakeholder values in the elicitation process of requirements. The requirements resulting from this workshop are collected as *value-based user stories*. In addition to the classical format of user stories (Cohn, 2004), Wanderley et al. (Wanderley et al., 2014) present a visual language for user stories. The visual representation of the user story supports the evaluation of the requirements with users and can be utilized with a *User Story Visual Editor*.

Prototypes are categorized by the studies on different types of *fidelity* (low, mid and high fidelity). Besides, the authors use the terms *prototypes*, *mock-ups* and *wireframes*. Lucia et al. (Lucia and Qusef, 2010) recommend using *paper prototypes* to document requirements with the purpose of communication and knowledge sharing between stakeholders and agile teams. In addition, Obendorf et al. (Obendorf and Finck, 2008) state that paper prototypes and informal drawings are very useful in discussions with users. *Informal drawings (sketches)* were also used in Blomkvist et al. (Blomkvist et al., 2015). In comparison, Rivero et al. (Rivero et al., 2014) applied html mockups to start the modeling process in *Model-Driven Web Engineering*. HTML-based mockups were used, on one hand, as a foundation to specify features like content, navigation and business logic and, on the other hand, to generate platform-independent UI specifications. Furthermore, Nawrocki et al. (Nawrocki et al., 2014) propose to use mockups to elicit test cases from users to make use cases testable.

Use Cases (UC) are often used to describe the behavior of a system from a more technical viewpoint compared to user stories. Issa et al. (Issa and AlAli, 2011) built a *UC patterns catalogue* that could be used as a feature checklist and to design an initial version of a UC model. Their *UC meta-model* addresses the environmental, technical, structural, eventual and traceability dimensions of the anticipated system. Besides, Nawrocki et al. (Nawrocki et al., 2014) argued that UCs could be reused for user manual generation, and for effort estimation. Farid (Farid, 2012) divides requirements into functional and non-functional. Functional requirements are described as

Agile Use Case, whereas non-functional requirements are presented as *Agile Loose Case*. *Aspect-Oriented "pointcut" operators* link functional to non-functional requirements.

Scenarios describe how users interact with a system in a specific context. To this end, they are often combined with *personas* (Maguire, 2013), (Blomkvist et al., 2015). In Bellucci et al. (Bellucci et al., 2015), designers used a journal in order to document scenarios based on user insights, gathered with a *user diary* and *usage try-outs*. Moreover, Obendorf et al. (Obendorf and Finck, 2008) used scenarios to connect a *design vision* with the more technical tasks of programmers in a *Scenario-based usability engineering* approach.

Story Cards present additional information related to user stories. Abdullah et al. (Abdullah et al., 2011) report that story cards allow capturing plans (estimates), history (who worked on the card) and goals. Besides, Farid (Farid, 2012) defined exactly which questions had to be answered with a story card in the *W8 User Story Card Model*. The eight "W" are: who, what, why, without ignoring, while it is nice to have, within, with a priority of and which may impact. On the contrary, Ramesh et al. (Ramesh et al., 2010) used story cards to document requirements for the next iteration that were elicited through communications with the customer.

Artifacts that are mentioned one time: Wall, pin board, Event-driven Process Chain (EPC) models, business process repository, domain models, snapshots, tags, SUI model, index cards, data flow diagrams, user wish list, user journey, UX requirements, design concept, evaluation goals, test specifications, role model, task model, operational model, interaction scenarios, user performance, experience goals, hedonic quality goals, document with Functional Requirements (FR) and Non-Functional Requirements (NFR), realization concept, refined into requirements lists, working software, system model, user model, effect maps, sketches, Product Backlog, user-diaries with picture and videos, sprint goal, delivery roadmap, definition of "done" and sprint burn-down chart.

Concerning **classical agile artifacts** (e.g. Product Backlog, Sprint Goal and Sprint Backlog) we can point out that the included studies rarely mentioned them. This leads to the conclusion that not every used artifact is reported by the studies.

Documentation of Requirements

The results of the literature review show that there are also some problems regarding documentation of requirements. Blomkvist et al. (Blomkvist et al., 2015) identified that not every artifact was used by developers. They report that developers did not read *personas*, *scenarios* and *effect maps*. They describe this phenomenon as *TAGRI principle* (They Ain't Gonna Read It). For this purpose, it is important to find the right combination of artifacts that fit the context of the project and people working in it.

Furthermore, Blomkvist et al. (Blomkvist et al., 2015) recommend that HCD specialists should translate their work directly into user stories, otherwise user stories have a strong technical focus. They state that user stories provide a way to translate HCD work into a format that had been already used in ASD. In contrast, Cajander et al. (Cajander et al., 2013) report that they are not well suited to usability work as there are difficulties to describe usability aspects in such a way. To their mind, usability needs to be addressed on a higher level.

We also find another aspect concerning the treatment of user stories. Liskin et al. (Liskin et al., 2014) studied the granularity of user stories, specifically the level of functionality an artifact deals with. They verified that there were communication and planning issues for big stories (implementation < 1 week) but they were too vague, thus, they recommend splitting such stories.

Furthermore, they report that requirements artifacts could avoid miscommunication and make requirements visible.

Functional and Non-Functional Requirements

Several studies classify requirements as functional or non-functional (see C3.3, Table 8). However, we have uncovered some problems concerning the treatment of Non-Functional Requirements (NFR).

Ramesh et al. (Ramesh et al., 2010) identified, in a multi-case study, neglected non-functional requirements as a challenge of Agile RE. They report that some organizations had not been paid much attention to NFR in early development cycles and that this lack of attention has often led to redevelopment and bottlenecks.

This challenge is addressed by several studies (e.g. (Bourimi et al., 2010), (Farid, 2012), (Nawrocki et al., 2014)). Bourimi et al. (Bourimi et al., 2010) consider NFR in early stages of the development process with their *Agile Framework for integrating non-functional Requirements Engineering (AFFINE framework)*. Therefore, they introduce the role of *NRF stakeholder* into Scrum, who is responsible for managing NRFs and acts like a facilitator to all stakeholder of the project. On the contrary, Farid (Farid, 2012) developed an agile methodology for identifying, linking, and modeling NFRs with FRs through different kind of cases and aspect-oriented pointcut operators. In the *Non-functional Requirements Modeling for Agile Processes (NORMAP)* requirements are classified as functional or non-functional by taxonomy. Nawrocki et al. (Nawrocki et al., 2014) looked at the elicitation of NFRs and proposed a method called *SENoR (Structured Elicitation of Non-functional Requirements)*, which consists of three steps: 1) presentation of the business case 2) series of short brainstorming session according to ISO/IEC 25010:2011 (ISO, 2011) and 3) voting with regard to the importance of the elicit requirements.

Another problem is reported by Lucia et al. (Lucia and Qusef, 2010). They claim that there is often a lack of formal acceptance tests for NFRs. A similar observation was made by Dragicevic et al. (Dragicevic et al., 2014), who recommend specifying at least one KPI to measure each NRF.

2.3.4.5. Limitations of the Review

There may be some relevant papers we missed because of the high amount of published literature, even though we used a predefined protocol and followed a rigorous search strategy to ensure the completeness of our study. We addressed this risk through forward and backward snowballing, since it also preserves us for a bias in the selection process. The selection process was mainly performed by the first author of the paper. We applied our search strategy, due to the high amount of findings from our search. For the phases P4-P6 (scan title, abstract and content manually) the first author decided which papers were relevant to be included. This may lead to a certain degree of subjectivity while performing such a selection. However, in cases of difficult decisions, the first author consulted the others in order to reduce subjectivity.

Another possible weakness of our approach might be the chosen selection criteria. For example, we focused on papers written in English language. Therefore, there might be relevant studies written in languages other than English, which were excluded because of the applied exclusion criteria.

With regard to the limitations in data extraction, we are aware of the fact that some aspects in reviewed studies (e.g. artifacts and methods) might be poorly reported. For this purpose, our results

would have been different, if the studies had been reported more accurately. We tried to address this issue through an extensive quality assessment of the included studies.

2.3.5. Revision of State of the Art Analysis

The initial search phase of the SLR was carried out in between June 2015 and August 2015 (see section 2.3.3.3). We are aware that there may be new papers that are relevant for our state of the art analysis. To this end, we carried out a revision of the search process in June 2017, so that we do not miss important works, which have been published after our initial analysis. The revision allows us to check whether our results of the state of the art analysis are still up-to-date since.

2.3.5.1. Search Strategy and Study Selection

For carrying out the revision of the SLR, we used the same review protocol (see Appendix II) with an additional restriction to the year of publication (2015 - 2017) and an extension of conditions to include new papers. Moreover, we excluded our own papers from the search and papers that had been included in the initial evaluation. The parameters for the revision are summarized in Table 10.

Table 10: Overview of parameters for revision of search

Search string	(agile OR scrum OR kanban OR “extreme programming” OR lean) AND (“human-computer interaction” OR “human-machine interaction” OR “user-centered design” OR usability OR human* OR user*) AND (“requirements engineering”)
Search space	<ul style="list-style-type: none"> • Google scholar (full text) • Science direct (abstract, title and keywords) • SpringerLink (abstract and keywords) • Scopus (abstract, title and keywords) • IEEEExplore (abstract and keywords) • ACM (abstract and keywords)
Selection criteria	<p><i>Inclusion criteria:</i> papers written in English; papers published in between 1995-2015; papers under peer review; papers presenting approaches to integrate user into agile development processes; papers related to Agile RE; papers associated with agile requirements documentation; and specific book chapters.</p> <p><i>Exclusion criteria:</i> no full books; papers whose full text were not available; papers with results that had been already published; papers that were not focused on agile development; papers only presenting ideas, lessons learnt, recommendations or guidelines; papers introducing tools whose underlying methodology was not comprehensibly described (black box); and studies, whose primary focus moved away from agile methodologies; own contributions; papers that already had been included in initial evaluation</p>
Time period	Published in between 2015 – 2017 (gap to initial search until now)
Conditions for considering new paper	<ul style="list-style-type: none"> a) Paper presents new insights that are highly relevant in terms of our RQs and fulfill the defined sub-criteria (see section 2.3.3.1) b) Paper presents groundbreaking new work in terms of our RQs (see section 2.3.3.1) c) Paper fills the identified research gap in existing literature (see section 2.4)

Figure 11 shows the revision of our search process and the findings for each phase (P1_R-P6_R). Compared to the search results from our initial search phase in 2015 (see P1=42808, Figure 9), the findings for P1_R are much higher (see P1_R=171942, Figure 11). The high amount of papers found in P1_R may stem from changes in the search engines of the databases. The adapted search strings, which we used in 2015 did not work and we had to adapt them again. In P6_R, 7 papers were analyzed by means of a screening of the whole content. We applied snowballing (forward and backward) for those papers, but did not find anything relevant for being included.

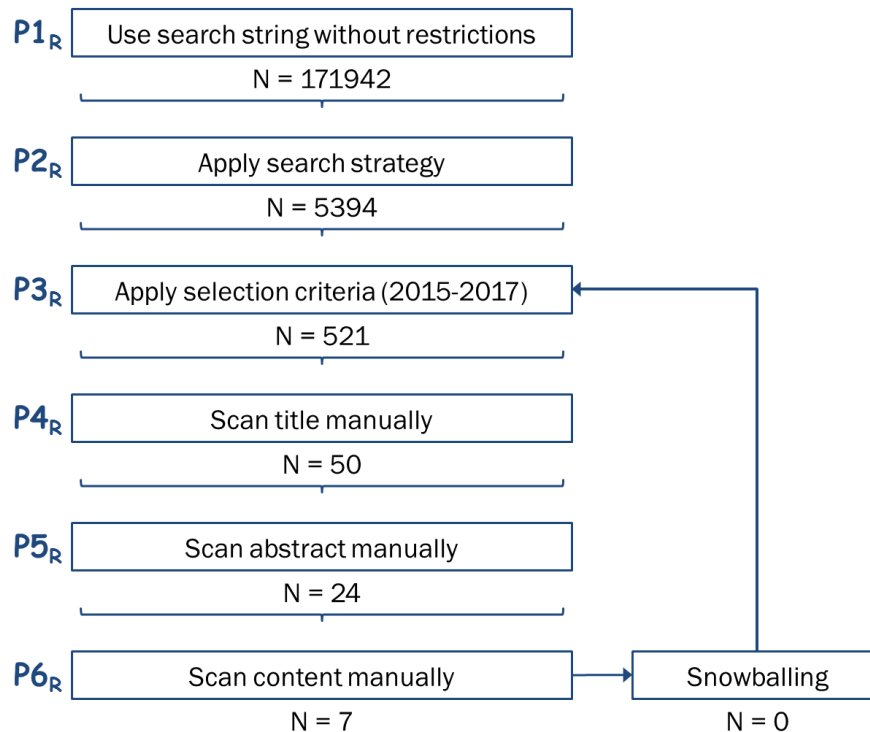


Figure 11: Revision of search process comprising six phases and snowballing. After analyzing the data, we included 5 papers into the revision of our study.

Figure 12 displays the distribution of search results among digital libraries (ACM, Science direct, Scopus, IEEEExplore, SpringerLink and other). In particular, it shows the distribution where the full texts have been retrieved. The highest amount of papers were found in ACM (63972), followed by Science direct (51967) and IEEEExplore (49047). Moreover, we found 4425 papers in the digital libraries from SpringerLink and 2530 papers in Scopus. We were able to reduce the number of findings by applying our selection criteria with the additional restriction to the year of publication (2015-2017). Finally, we identified 5 papers, which are relevant for the revision of the state of the art analysis. One paper was found via SpringerLink, two papers via IEEEExplore, and one via ACM. Moreover, we found one more paper during an informal search via an alert on google scholar. The papers, which we found in SpringerLink and ACM, also occur in the findings in Scopus.

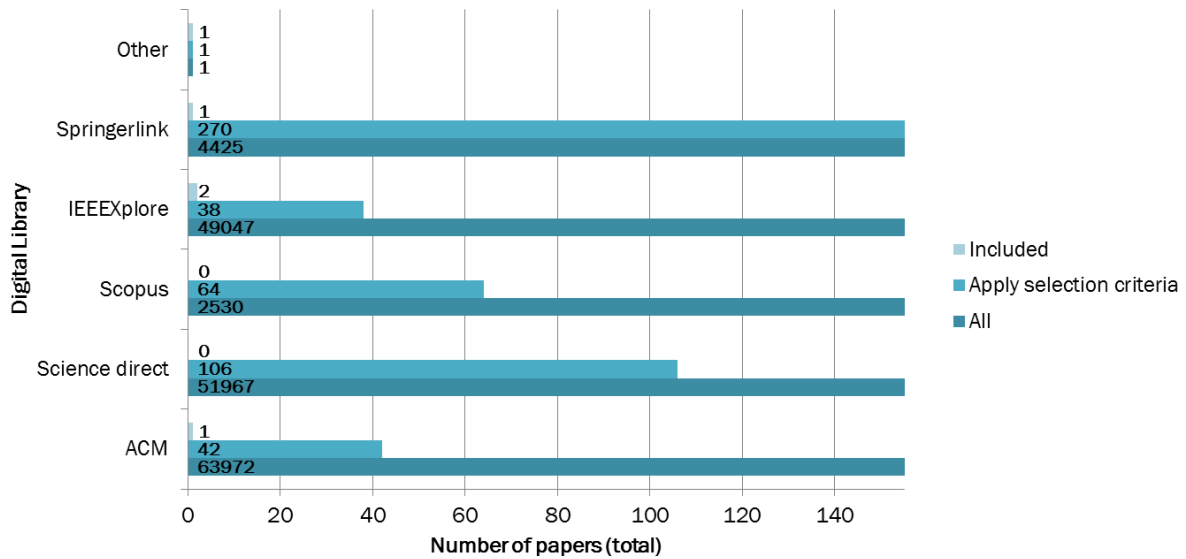


Figure 12: Distribution of search results among digital libraries

2.3.5.2. Findings and Discussion

As mentioned before, we decided to include 5 papers in the revision of our state of the art analysis. In the following, we will explain how these papers support to answer our RQs and discuss on the implications in terms of the state of the art analysis.

RQ 1.1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

Table 11 shows an overview of the findings related to RQ1.1. In sum, we found two papers that are relevant for the state of the art analysis of agile RE. Both papers are included since they present new insights that are highly relevant in terms of our RQs and fulfill the defined sub-criteria.

Table 11: Findings of revision related to RQ1.1

Paper	Stakeholders are involved directly	Users are involved directly	A process is used to involve stakeholders	Iterations are applied during the development process	Methods are used to gather data
(Kropp and Koischwitz, 2016)	x	x	x	x	x
(Lombriser et al., 2016)	x	x	x	x	x

Kropp and Koischwitz (Kropp and Koischwitz, 2016) provide a detailed description of the role and tasks of an *On-site User Experience Consultant (osUX consultant)* reported on the basis of own experiences in fixed-price projects. The *osUX consultant* is responsible for integrating HCD in agile RE and acts as an advocate for the users. S/he ensures that users and users' requirements are at the center of attention. Hence, Kropp and Koischwitz recommend HCD activities for the main phases of their projects (initiation, conceptualization, implementation, and follow-up). These

activities comprised applying methods, artifacts, and meetings and range from user research to usability testing.

Lombriser et al. (Lombriser et al., 2016) developed a novel model to link gamification and RE called *Gamified Requirements Engineering Model (GREM)*. *GREM* aims to increase stakeholder engagement by means of a gamified platform. The platform allows elicitation of agile requirements in the format of user stories. First experiments reveal that gamification has a positive effect on stakeholder engagement. Up to now, *GREM* only focuses on the activity of requirements elicitation and therefore do not provide a way to involve stakeholder into different activities of requirements management.

RQ 1.2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

Table 12 shows an overview of the findings related to RQ1.2. We found one paper that is relevant for the state of the art analysis of agile RE. The paper is included since it provides new insights that are relevant in terms of our RQs and fulfill the defined sub-criteria.

Table 12: Findings of revision related to RQ1.2

Paper	Proposal used a methodology	Knowledge about the user requirements is shared between the stakeholders
(Newman et al., 2015)	x	x

Newman et al. (Newman et al., 2015) integrate Design Thinking into the Social Software Engineering framework *Speedplay*, which is an agile and participatory framework with the aim to improve requirements elicitation. They create different prototypes with in order to explore the problem space in collaboration with users and stakeholders. The collaborative development of prototypes supports the knowledge sharing concerning the user requirements. In addition, the participants are enabled to physically engage with the identified social problem through the prototypes so that they receive a deeper understanding of the problem and find appropriate solutions.

RQ 1.3: What are the common ways for requirements management in ASD?

We found two papers, which deliver new insights in terms of our RQs. Both papers present mapping studies dealing with artifacts. In general, the evaluation of these papers in terms of our sub-criteria for RQ1.3 is not suitable, since the sub-criteria assess concrete approaches and not collections of papers. Nevertheless, both papers investigate what types of artifacts or techniques are utilized in terms of ASD. This objective correlates with aspects of our RQ1.3 and therefore, we decided to include the papers into this revision.

Garcia et al. (Garcia et al., 2017) conducted a systematic mapping study on artifacts and their role in communication in the context of integrating Agile and HCD. In particular, they analyzed which artifacts are used to increase the communication. In sum, they found 20 relevant artifacts: blueprint, card, guideline, list, map, mockup, model, persona, prototype, research results, scenario,

sketch, storyboard, storytelling, UI design, use case, user flow, user story, and wireframe. The most cited artifacts are prototypes, user stories, cards, and personas.

Magues et al. (Magues et al., 2016) performed a systematic mapping study with the aim to study the integration between agile processes and usability techniques. They state that most of the identified techniques relate to RE. They report that the following usability techniques had been used in ASD: contextual inquiry, ethnographical observation, card sorting, personas, essential use cases, task scenarios (synonym for usability user stories), scenarios and storyboards, prototyping, and cognitive walkthrough. Besides, Magues et al. analyze usability techniques for software engineering activities concerning design and evaluation.

Implications of Findings

General findings. Comparing the findings of P3 of the initial search (P3 N=391, see Figure 9) and the revision of the search (P3_R N=521, see Figure 11), we can observe an increasing number of relevant publications. To this end, we can conclude that our RQs are highly relevant in framing the available research of agile RE in terms of the state of the art analysis. We can conclude that agile RE is an emergent research topic and the body of knowledge is evolving rapidly. Moreover, we can determine that agile RE is respected as problem solving approach to ASD, since new agile techniques occur to solve identified problems.

Findings related to RQ1.1. The role of an *osUX consultant* reported by Kropp and Koischwitz (Kropp and Koischwitz, 2016) is similar to the concept of an *UCD specialist* (see section 2.3.4.2). This finding underlines the importance of a role, which is responsible for the users' needs. The gamified platform for the elicitation of user stories proposed by Lombriser et al. (Lombriser et al., 2016) seems to be a promising way to increase stakeholder engagement. Nevertheless, the platform should be extended in order to be capable to cover more activities than requirements elicitation.

Findings related to RQ1.2. With regard to the study by Newman et al. (Newman et al., 2015) we can conclude that the revision of the state of the art analysis reveals that HCD, Design Thinking, Contextual Inquiry and Participatory Design are still commonly used methodologies which are used in order to make ASD more human-centric.

Findings related to RQ1.3. We can state that the topic of artifacts is very important for the research field of agile RE. Two further research groups investigate the role of artifacts at the same time than we do⁴. The artifacts identified by (Garcia et al., 2017) are similar to the ones identified by our SLR (see section 2.3.4.4), except for blueprint and UX target. Compared to our results, they also identify user story and prototype as most frequent used artifact. Moreover, the artifacts reported by Magues et al. (Magues et al., 2016) are similar to our results, too. In terms of the most cited artifacts reported by Magues et al. (personas, contextual inquiry, prototyping) we can observe some differences. This may cause by the fact that they had a slightly different focus in their study. However, contextual inquiry is one of the methodologies, which we identified as relevant in terms of presenting the user perspective to stakeholder (RQ1.2).

Summarizing the revision of the state of the art analysis, we can conclude that our results (see section 2.3.4) are still up-to-date. With regard to the defined conditions for considering new papers (see Table 10) we can state that all papers were included in the revision because they met criteria a) *Paper presents new insights that are highly relevant in terms of our RQs and fulfill the defined*

⁴ It is worth mentioning that our paper got published before their work.

sub-criteria. Moreover, we do not find groundbreaking new work, which would change the outlined definition of the start of the art of agile RE with focus on stakeholder and user involvement.

2.4. Implications and Gap Analysis

The results of the state of the art analysis have several implications for both researchers and practitioners. Based on a qualitative analysis of the included studies, we can conclude that building a *shared understanding* of the *user perspective* is not very well established in ASD. It became obvious during the deeper analysis of the identified publications, that only a limited number of papers investigated the presence of the user perspective in ASD. These publications revealed that there were many problems concerning the direct involvement of users and stakeholders. However, we identified four methodologies (*Human-Centered Design, Design Thinking, Contextual Inquiry* and *Participatory Design*) that were integrated in ASD with the aim to increase the knowledge about user needs. Furthermore, we identified a broad range of different methods that can be used in ASD to *gather data* in terms of RE. We identified the following *key artifacts* for the *documentation* of requirements that are used in Agile RE: *User stories, prototypes, use cases, scenarios* and *story cards*. Industrial practitioners can utilize these findings as recommendations to discover the right combination of artifacts for their development process. With regards to *NFRs*, we can conclude that on one hand, there are different approaches to deal with NFRs, but on the other hand, we determine an open challenge concerning the estimation and measurement of these requirements (e.g. *UX metrics, security policies*).

General findings. The results of the SLR show that Agile RE has been studied within various research areas (e.g. software engineering, human factors or Participatory Design). We can conclude that this is an important research topic. Furthermore, the heterogeneity of the studied aspects in the reviewed studies shows that this is a complex research field with a lot of different cross-functional influences. Moreover, we can state that this research field is very close to existing work practices in companies, since most of the included studies analyze the aspects in a real life context.

Findings related to RQ1.1. Concerning our first RQ, we can conclude that continuous communication and collaboration is the most frequent used approach to involve stakeholder in the process of RE compatible with ASD. The variety of aspects and methods reported by the reviewed studies point out that there is no common process model for stakeholder involvement in an agile environment. Although a broad range of reviewed studies dealt with the development of systems including a user interface, only half of them involved users directly into the development process. However, it is necessary to engage user in order to validate assumptions during system development; otherwise requirements would be seen as a single point of truth. In light of this, we consider that further research with regard to a structured process model for stakeholder and user involvement is required.

Findings related to RQ1.2. Studies that address the user perspective in ASD provide some insights about which methodologies are useful in order to make ASD more user-centric. We find that *Human-Centered Design, Design Thinking, Contextual Inquiry* and *Participatory Design* are accepted methodologies. However, the results also show that there are problems with sharing knowledge between stakeholder concerning user requirements and the responsibility for usability/UX. To this end, we can conclude that it is a key to find appropriate methodologies that help to solve these problems.

Findings related to RQ1.3. The analysis of the reviewed studies shows that a variety of different artifacts are applied to Agile RE. We have identified *user stories*, *prototypes*, *use cases*, *scenarios* and *story cards* as the most frequent used artifacts. Furthermore, we have faced some problems regarding requirements documentation; there are difficulties to identify the right kind of artifacts, which enhance collaboration among stakeholder, developer and agile team. This is a special challenge with regard to the project setting (e.g. co-located or distributed teams). Nevertheless, it is important to create appropriate guidelines for requirements management within ASD.

2.5. Chapter Summary

This chapter presented the foundations related to agile methodologies, HCD, and RE, which build the basis for this work. Moreover, we outlined a definition for agile RE, which will be elaborated in the following chapters. Besides, we depict the relation to the work of relevant research groups in the field.

Moreover, the most relevant related work regarding the relation among Agile, HCD, and RE has been presented. To this end, we analyzed the state of the art of agile RE with focus on user and stakeholder involvement by means of a SLR.

The state of the art study helped us to achieve deep insights into the field of agile RE, in particular in terms of:

- presence of the user perspective
- stakeholder and user involvement
- integration of methodologies to make ASD more human-centered
- building a shared understanding
- data gathering
- non-functional requirements
- documentation of requirements by means of artifacts

In addition, we identified gaps in existing research related to the iterative involvement of user and stakeholder in the development process, as well as building a shared understanding of the user perspective. We found a variety of methods and artifacts, which are used in ASD for different purposes in a specific context. Guidelines for choosing the right kind of artifacts that enhance the collaboration among all people involved in the development process are missing.

Based on the knowledge we gained by analyzing the related work and background in the field of agile RE, we are enabled to define the challenges and objectives of this PhD thesis in the following chapter 3.

In an agile environment Requirements Engineering (RE) is carried out iteratively alongside the whole product development instead of a closed phase in the beginning. To this end, a just-in-time model is often used to refine high level requirements into low level tasks that can be implemented by developers. Therefore, business people, stakeholders, users and developers work together in a collaborative manner. Chapter 2 shows that there is a variety of best practices for agile RE, which are tailored to the context of the environment and often used implicitly by the people working with it.

In this chapter, we outline the challenges this PhD thesis pursues (see section 3.1) together with a detailed description of the objectives we want to achieve (see section 3.2). Therefore, we will provide an overall vision expressed by a vision statement. The vision will be split into several objectives and we will describe how we want to achieve each objective by providing an overview of the research methods, which will be used for this work.

3.1. Challenges

This section will present in detail the main challenges of this PhD thesis, based on the results of the state of the art analysis, with the aim to explain the research gaps and to justify the need of this work.

Traditional process models, such as the waterfall model (Royce, 1970), are plan driven. They consist of sequential phases and start with a big upfront design. In the beginning of these projects, all requirements are consolidated into a specification document. Based on this, schedule and budget are estimated. Requirements are seen as true and changes are made using a heavyweight change request process. In the mid-80s, Takeuchi et al. (Takeuchi et al., 1986) already stated that a sequential phases approach to product development is not well suited due to the lack of flexibility. Since then, new process models have been developed. On one hand, there are iterative process models like Rational Unified Process (Kruchten, 1998), (Kruchten, 2004). On the other hand, there are agile methodologies such as Scrum (Schwaber, 1997), (Schwaber, 2004), Extreme Programming (XP) (Beck, 2000), Feature-Driven Development (Palmer and Felsing, 2001) and Kanban (Anderson, 2004), (Anderson, 2010).

Established RE approaches (Sommerville and Sawyer, 1997), (Pohl, 2010) are created in terms of the requirements of sequential approaches to product development. The conditions ASD comes with have an impact on the way RE is carried out in agile environments. By comparison, RE in agile environments is carried out iteratively during the whole development process instead of a closed phase in the beginning. New information is given along the product development by the user and the product itself. This knowledge is processed during further steps and it conditions the decision-making process. Therefore, requirements are treated like assumptions, which are validated continuously.

Agile RE is a cross-functional research area comprising areas like Human-Centered Design (HCD), Agile Software Development (ASD), Requirements Engineering (RE) (see Figure 6). Contributing to the body of knowledge of agile RE implies considering research from all of the aforementioned areas and therefore increases the complexity. This complexity needs to be reduced in order to build a strong framework for agile RE.

Organizations facing the challenge to create a value-driven environment, where agile values build the foundation and value delivery has the highest priority. Therefore, people need to focus on outcomes and how they can fulfil human needs through the outputs they produce. The results of the state of the art analysis reveal that building a shared understanding of the user perspective is not very well established in ASD. Furthermore, we identified problems with sharing knowledge among stakeholders and agile teams concerning the user requirements. In light of this, we can conclude that organizations struggle with prioritizing the most valuable requirements.

As pointed out in chapter 2 there are related works proposing process models for agile RE (Mommel et al., 2007), (Kautz, 2010), (Maguire, 2013), (Rivero et al., 2014), (Olsson and Bosch, 2015). Analyzing the commonalities of the proposed approaches, they have in common the continuous management of requirements by involving stakeholder and user. Furthermore they are describing the way of working in the context of agile RE by means workflows, role descriptions, and agile techniques. What we can observe is that there is a huge heterogeneity among agile RE process models focusing on user and stakeholder involvement since they are all tailored to specific needs of the environment in which they are applied. Existing research does not provide a commonly accepted process model for involving users and stakeholders and it is shown that there are still open challenges. Furthermore, the related work lacks in providing generic concepts on an abstraction layer. Researchers as well as agile practitioners facing the challenge to create a collaborative environment, where value delivery for users and customers is the most important aspect.

Furthermore, the results of the SLR show that practitioners using a lot of different techniques in terms of agile RE and especially for engaging stakeholders and users. These agile techniques comprise for instance: *co-design sessions with users* (Bellucci et al., 2015), *Participatory Design* (Kautz, 2010), *Value Story Workshop* (Harbers et al., 2015), *lead user* (Näkki et al., 2011) or *Contextual Inquiry* (Obendorf and Finck, 2008), (Maguire, 2013). Each of those techniques pursues a specific context. The variety of techniques leads to difficulties choosing the right kind of agile technique, which enhances collaboration among stakeholders, users and agile teams. In terms of agile RE process models is the challenge to understand what is the impact of specific techniques and how these techniques influence the environment.

Another challenge caused by the variety of existing agile techniques is reported by Blomkvist et al. (Blomkvist et al., 2015). They found that not all artifacts are used by the agile teams and summarized this as the TAGRI (They Ain't Gonna Read It) principle. This is another indicator that shows the importance of choosing the right combination of agile techniques that fit the context. Finding an appropriate combination of agile techniques is nothing done with one trial, instead an organization must continuously reflect on their process models in order to improve them. To sum up, the overview of existing agile techniques is missing and this makes it hard to find the appropriate agile technique, which solves a specific problem.

3.2. Objectives

This section will present the vision of this PhD thesis and discuss the objectives, which we want to achieve with this work.

The vision of this PhD thesis is the following:

Building a framework for modeling agile RE. Organizations benefit from implementing this framework by increasing their value delivery (organization external) and improving the collaboration (organizational intern).

A framework can be defined as “a supporting structure around which something can be built” (Cambridge Dictionary, 2017). Following this definition, we want to provide an abstract view on the complex field of agile RE. To this end, the main part of the framework builds the agile RE metamodel, which is described in chapter 4. The framework enables researchers as well as practitioners to model new process models for agile RE, which are strongly focused on value delivery and therefore can be classified as value-driven process models. Moreover, the framework can be used to evaluate and improve existing process models for agile RE by reflecting on strengths and weaknesses of the approaches in the field.

We need to split the vision of this PhD thesis into several objectives, in order to achieve tangible and measurable results. Table 13 highlights the objectives and the corresponding results of this PhD thesis.

Table 13: Overview of objectives of this PhD thesis

ID	Objectives	Expected results
0-1	Perform a systematic literature review, in order to capture the current state of the art of the literature related to agile RE with focus on stakeholder and user involvement.	State of the art agile RE
0-2	Propose a conceptual approach for agile RE in order to provide an abstract solution for agile RE management.	Split into 0-2.1, 0-2.2
0-2.1	Propose a metamodel for agile RE, in order to provide an abstract view on the influencing parameters of agile RE.	Agile RE Metamodel
0-2.2	Provide a domain specific model in order to show how the metamodel can be applied in industry.	Instantiations of agile RE process models by means of a profile
0-3	Provide techniques to apply the conceptual approach to the enterprise environment.	Split into 0-3.1, 0-3.2
0-3.1	Identify the most important problems in agile RE industry has to face today, in order to provide solutions.	Agile RE problems
0-3.2	Create agile RE patterns, in order to solve the identified agile RE problems.	Agile RE patterns
0-4	Share the knowledge concerning agile RE, in order to empower people to become a value-driven organization.	Web application (agileRE.org)

Figure 13 displays a conceptual schema of this PhD thesis comprising the vision, the objectives and the results. We started the work in 2015 with a SLR, which builds the foundation of our research. The analysis of the state of the art helps us to understand the existing process models for agile RE in the field and enables us to build an abstract view by means of the agile RE metamodel. Subsequent to this, we iteratively elaborate the agile RE metamodel and learn how to apply it in industry. For this reason, we identify the most important challenges in agile RE industry has to face today. Together with the agile techniques known from our SLR, we can derive agile RE patterns. Moreover, we want to share the acquired knowledge about agile RE with the community by means of a web application (agileRE.org).

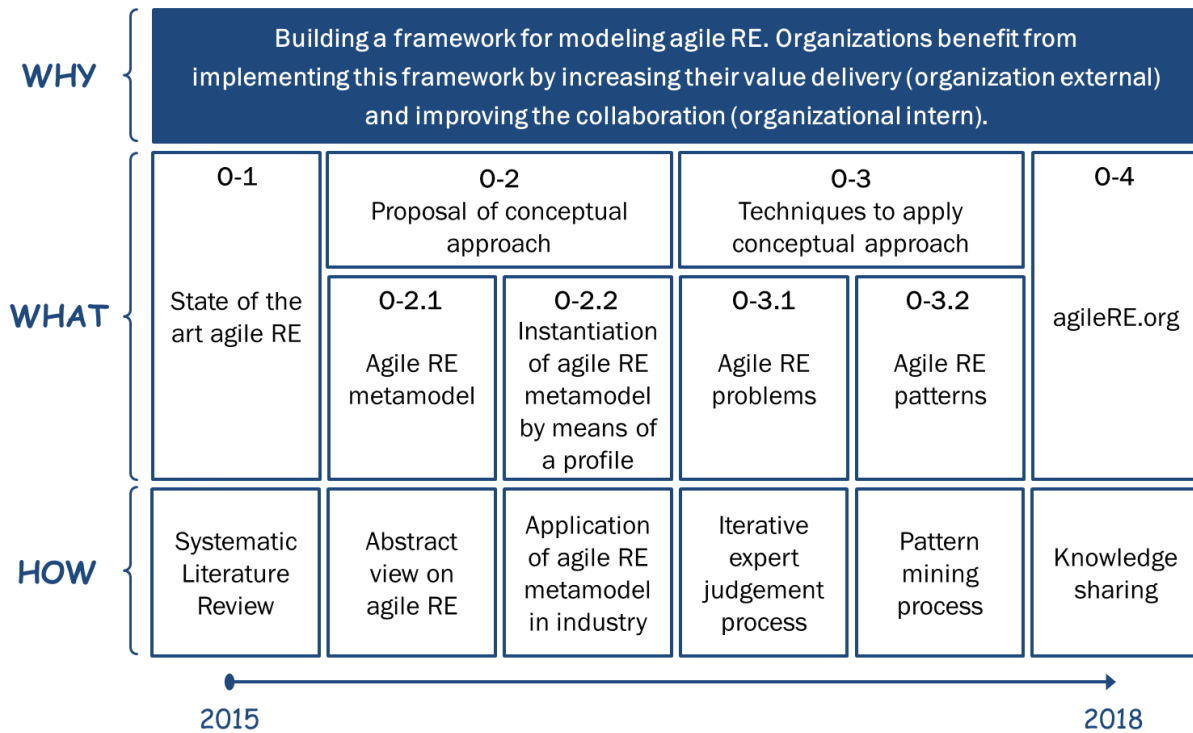


Figure 13: Conceptual schema representing the objectives to be achieved with the PhD thesis

The proposal of this PhD thesis was discussed at the Doctoral Consortium of the CAISE Conference in 2016 (Schön, 2016). We elaborated the vision and the objectives since that presentation.

In the following, we will detail the objectives presented in Table 13 and provide a description on how we want to achieve those objectives.

3.2.1. Perform a Systematic Literature Review

As mentioned before, the foundation of this PhD thesis builds a deep analysis of the state of the art of the literature related to Agile RE with focus on stakeholder and user involvement (see chapter 2). In particular, we investigate what approaches exist to involve stakeholder in the process, which methodologies are commonly used to present the user perspective and how requirements management is been carried out. This objective will be addressed by the following research questions:

- RQ-2.1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?
- RQ-2.2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?
- RQ-2.3: What are the common ways for requirements management in ASD?

In order to analyze the state of the art, we carry out a SLR with an extensive quality assessment of the relevant studies (Schön et al., 2017a). The search space comprises databases like Google scholar, Science direct, SpringerLink, Scopus, IEEEExplore, and ACM. At the beginning,

the search results showed a high amount of papers. We reduce the results while carrying out a predefined search process to identify relevant papers for further analysis.

3.2.2. Propose a Metamodel for Agile RE

The heart of our framework is the agile RE metamodel, which visualizes the big picture of agile RE [Ref.]. It provides generic concepts, which can be used for tailoring domain specific process models for agile RE according to the organizational environment, where the product is developed. On the one hand, the agile RE metamodel can be used to model new process models, on the other hand, it can be utilized to improve existing process models. Up to now there is no metamodel for agile RE, which builds an abstraction layer about the variety of existing process models. In addition, we want to create a visual language for the agile RE metamodel by means of a profile. An profile allows us to extend the metaclasses of the agile RE metamodel in order to tailor domain specific models. Therefore, this objective is guided by:

- RQ-4.1: What does a metamodel that covers agile RE approaches look like?
- RQ-4.2: What does an profile for the agile RE metamodel look like?

In order to describe agile RE on an abstract level, we will analyze existing process models for agile RE. Based on the related work and our experience working with requirements in agile environments, we can create a metamodel for agile RE (see chapter 4). This metamodel for agile RE will be optimized in several iterations by means of discussion with other researchers and practitioners in the field of agile RE as well as further studies.

3.2.3. Provide Domain Specific Models of the Metamodel

A metamodel is a model of a model and visualizes in this work an abstract view on the field of agile RE. Compared to a metamodel, a process model describes what actually happens in the field and details the way of working. A process model can be derived by creating an instance of the metamodel by means of an profile. In chapter 7, we present instances of the agile RE metamodel and provide examples how the metamodel can be applied in industry. This objective will be addressed by:

- RQ-7.1: How can the metamodel be tailored to different process models?

In order to show how the metamodel can be applied in industry, we will provide two examples of instantiations of the metamodel [Ref.]. On the one hand, we exemplify a process model for a Kanban-based process model for developing an internet-based newspaper portal. On the other hand, we derive an instantiation of the agile RE metamodel into a Scrum-based process regarding the development of web applications in the e-government areas.

3.2.4. Identify the Most Important Problems in Agile RE

For an agile transformation, organizations need to continuously improve their established approaches to RE as well as their approaches to software development. This is accompanied by some problems in terms of agile RE. The main objective of chapter 5 is to identify the most

important problems in agile RE industry has to face today. Thus, the research questions we pose are listed below:

- RQ-5.1: What are the key problems in Agile Requirements Engineering?
- RQ-5.2: How can we deal with the identified key problems?

In order to identify the most important problems in agile RE, we will conduct an iterative expert judgement process with 26 experts in the field of ASD, comprising three complementary rounds (Schön et al., 2017d). The iterative expert judgement process is rooted in a Delphi study and its main benefit is that we can use the learnings from a previous iteration for carrying out the following iterations. The questionnaire of round 1 comprises open questions, whereas the one in round 2 covers closed questions and comments. In addition, in round 3 we combine closed questions, open questions and comments.

3.2.5. Create Agile RE Patterns

Patterns are a common way for sharing knowledge concerning a specific topic. A pattern consists of a recurring problem and the core of a correlated solution. We will introduce the concept of agile RE patterns (Schön et al., 2017b) since agile RE is a complex research field, where we can find a reasonable amount of recurring problems as shown in chapter 5. The derived agile RE patterns will cause strong impact on practitioners since we gather the data among experts in the field of ASD. We address this objective by the following research questions:

- RQ-6.1: How can we create agile RE pattern?
- RQ-6.2: Which agile RE patterns do exist?

In order to create the agile RE patterns, we will carry out an iterative pattern mining process comprising three phases (see chapter 6). In the first phase, we identify agile techniques by means of an SLR. In the second phase, we identify agile RE problems and solutions to solve them. Then, in the third phase, we derive the agile RE patterns by mapping agile RE problems with agile techniques.

3.2.6. Share the Knowledge Concerning Agile RE

We want to share the acquired knowledge concerning agile RE with the community. To this end, we will provide tool support by means of a web application. The aim of the web application is to support practitioners and researchers improving their agile RE process models as well as enabling them to solve their key problems in agile RE. Our mission is to empower people to become a value-driven organization. This objective will be addressed by the following research questions:

- RQ-6.3: How can we share the knowledge concerning agile RE?

In order to share the knowledge concerning agile RE with the community, we will create a web application (agileRE.org). The web application will be developed in several iterations starting with a discussion of the ideas, followed by collaborative prototyping sessions and evaluations with potential users. The main part of the content represents the agile RE patterns.

3.3. Chapter Summary

In this chapter we presented the challenges and objectives this PhD thesis will address. The conditions of ASD have an impact on the environment where RE process models are applied. As shown previously, there is a variety of existing process models as well as best practices for agile RE considering stakeholder and user engagement. Each one is adapted to a specific context. An overall framework for improving the variety of agile RE process models is still missing.

We can use the existing knowledge and accumulate it in order to build our framework, which is the main goal of this PhD work. The framework for modeling agile RE can be used for the one thing to increase the value delivery. The definition of what value means can vary from organization or environment and needs to be defined by people who work on the product. For the other thing the framework allows to improve the collaboration among people working in the environment, where the product development takes place.

Figure 14 shows how the vision of this PhD thesis is split into four main objectives. These objectives are mapped to corresponding research questions, which will be addressed in the scope of this research work.

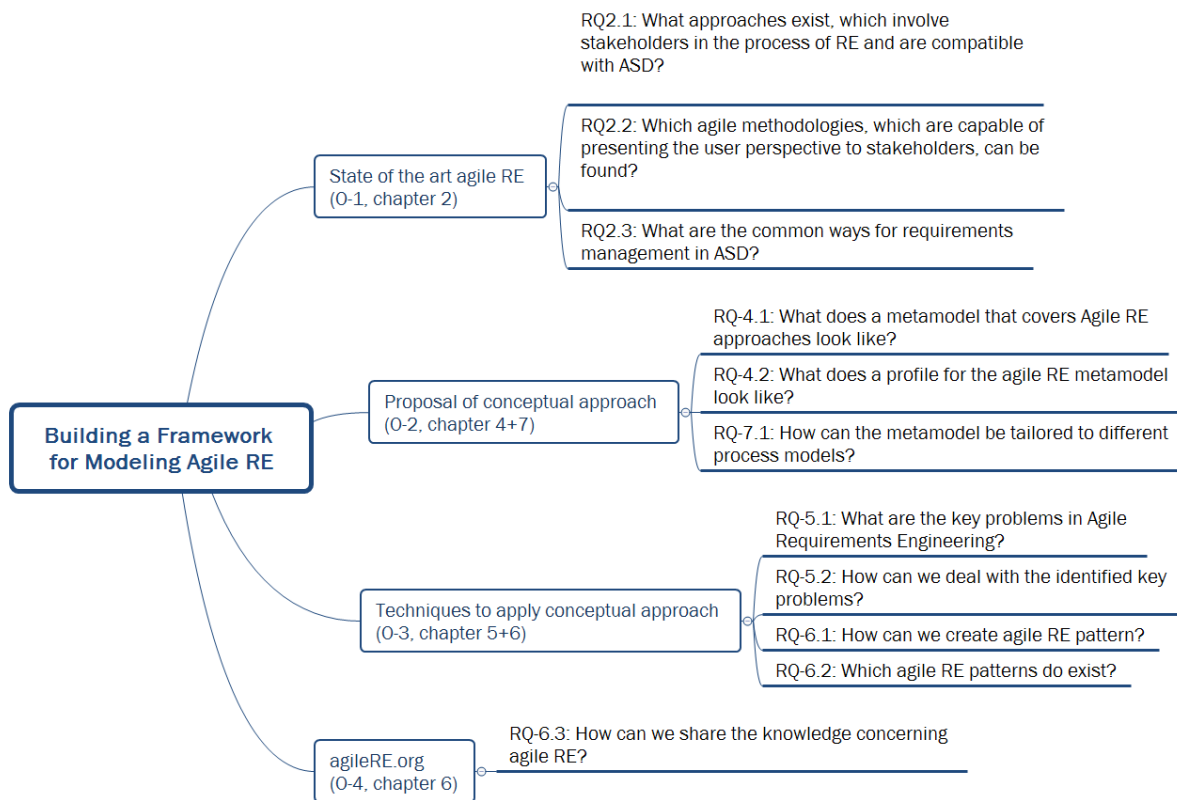


Figure 14: Mapping between vision of this PhD thesis, objectives and research questions

All the aforementioned aspects close the introduction of the PhD thesis. From now on, we will detail the contribution of this PhD thesis within the following chapters.

The previous chapters provided a state of the art analysis of agile RE with focus on stakeholder and user involvement (chapter 2) and outlined the challenges and objectives we want to address with this PhD thesis (chapter 3). It has been mentioned that agile RE is a cross-functional area where people from different domains work together in order to deliver value to their customers and users. This cross-functionality lead to a certain amount of complexity, which needs to be reduced in order to build a powerful framework for agile RE.

In industry, there can be found different process models for agile RE. These process models differ in how the way of working is described. For instance, which methods or artifacts are recommended to use for discovering, refining, prioritizing, reviewing or documenting requirements in an agile environment. In order to gain a better understanding of agile RE and to reduce the complexity of existing process models, we can build an abstract view on this field by means of a metamodel. As the environment where agile is used is complex, due to the number of diverse people working together in different roles to create one system, we need an extremely adaptive metamodel for describing agile RE on an abstract level.

In this chapter, we will contribute a conceptual approach for agile RE with the aim to provide an abstract solution for agile RE management. The main part of this conceptual approach builds the agile RE metamodel, which is visualizing the influencing parameters of agile RE and allows us to understand and to learn how to handle the aforementioned complexity [Ref]. The agile RE metamodel will be fostered by an profile, which builds the visual language of our approach.

Section 4.1 summarizes the related work, followed by a gap analysis in section 4.2. Then, section 4.3 describes the research approach and the development of the agile RE metamodel and the profile. Section 4.3.3 presents the improved agile RE metamodel, the profile and discusses on the benefits and implications.

4.1. Summary of Related Work

In literature, there can be found related studies proposing process models for agile RE. All these process models share the continuous management of requirements by involving stakeholder and user. The most related ones are highlighted below.

Memmel et al. (Memmel et al., 2007) develop a XP-based process model called *Cross-discipline User Interface and Software Engineering* (CRUISER). The process starts with an Initial Requirements Up-front Phase (IRUP) whose results are agile models that describe user needs by means of agile techniques (e.g. essential use cases, scenarios and prototypes). The gathered information is elaborated and processed during the different phases of CRUISER.

Kautz (Kautz, 2010) carried out a case study with the aim to investigate user and customer involvement in ASD. He does not claim to propose a process model but the results of the study present an implicitly applied process model. Kautz integrates Participatory Design activities in XP. The agile team can detect problems concerning misunderstanding of requirements early before they could grow into larger problems by the application of an onsite customer as well as iteratively reviews with users and customers.

Méndez Fernández et al. (Méndez Fernández et al., 2010) contribute a metamodel for artifact-based RE that covers the RE domain in general without targeting a specific development methodology. The metamodel is inferred by two existing RE models in industry. On the one hand, it provides a valuable overview of handling artifacts in RE. On the other hand, the metamodel enables companies to create artifact-based RE process models.

Maguire (Maguire, 2013) extends the HCD (International Organization for Standardization, 2010) framework for ASD. The HCD process consists of the following steps: plan the HCD process, understand and specify the context of use, describe user requirements, produce design solutions to meet such requirements and evaluate designs against them. He suggests a set of agile techniques that can be used in each step. In addition, he proposes a couple of artifacts that are generated while using agile techniques.

Rivero et al. (Rivero et al., 2014) propose the process model known as *Mockup-Driven Development* (MockupDD). Their approach supports Model-Driven Web Engineering (MDWE) and is integrated to Scrum. In the beginning of MockupDD, a quick requirements gathering stage is performed resulting in a set of user stories. Based on this, costumers and users create mockups to represent these user stories graphically. These mockups lay the foundation for the following modeling process.

Olsson et al. (Olsson and Bosch, 2015) design a process model based on a conceptual model named *Qualitative/quantitative Customer-driven Development*. It stresses the need for combining qualitative customer feedback in early stages of development with quantitative observations in later stages. Olsson et al. treat requirements as hypotheses that are validated with customers before development. Hypotheses are derived from business strategies, innovation initiatives, customer feedback and on-going validation cycles.

4.2. Gap Analysis of Related Work

Analyzing the commonalities of the proposed approaches, we can conclude that the work by Memmel et al. (Mommel et al., 2007), Kautz (Kautz, 2010), Maguire (Maguire, 2013), Rivero et al. (Rivero et al., 2014) and Olsson et al. (Olsson and Bosch, 2015) are all process models describing the way of working in the context of agile RE by means of workflows, role description and agile techniques.

The metamodel for artifact-oriented RE proposed by Méndez Fernández et al. (Méndez Fernández et al., 2010) is focused on RE in general and is not tailored to the specific requirements of ASD. In addition, they concentrate on an artifact-centered approach, whereas our work is targeting the collaboration among people and therefore is more human-centered.

However, the related work lacks in providing generic concepts on an abstraction layer. These generic concepts are important in today's business world since companies apply different types of process models for diverse teams. This leads to increasing complexity within an organization (e.g. scaled organization or other teams working with sequential approaches like waterfall model). To this end, this work proposes a metamodel for agile RE to handle this complexity. To the best of our knowledge this is the first metamodel for agile RE.

4.3. Research Approach

Figure 15 displays the phases, which were applied in order to develop the agile RE metamodel. In phase 1, we analyzed existing process models for agile RE in order to describe agile RE on an abstract level. Based on the related work and our experience working with requirements in agile environments, we have created the initial version of the agile RE metamodel (phase 2). This initial version of the metamodel was presented and discussed with the research community (Schön, 2016). The metamodel for agile RE has been optimized in several iterations by means of discussion with other researchers and practitioners in the field of agile RE as well as further empirical studies (Schön et al., 2017d).

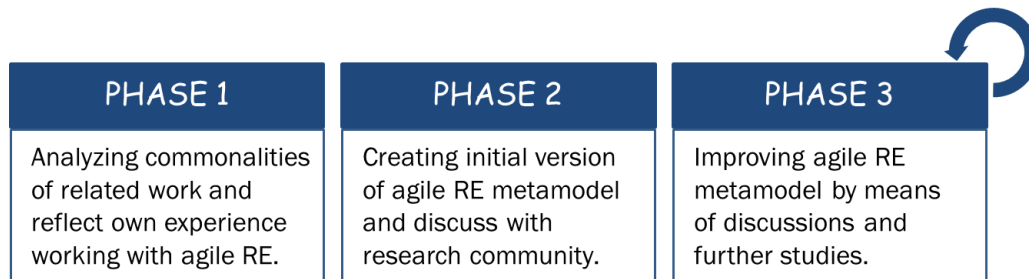


Figure 15: Phases to create the agile RE metamodel

4.3.1. Objectives and Research Questions

As mentioned before, the heart of our framework is the agile RE metamodel, which visualizes the big picture of agile RE. It provides generic concepts, which can be used for tailoring domain specific process models for agile RE according to the organizational environment, where the product is developed. For achieving this objective our research is guided by the following research questions:

- RQ-4.1: What does a metamodel that covers agile RE approaches look like?
- RQ-4.2: What does a profile for the agile RE metamodel look like?

4.3.2. Developing the Agile RE Metamodel

The initial version of the agile RE metamodel was created based on an extensive analysis of related process models for agile RE (see chapter 2) and our industry experience in terms of RE in agile environments. We presented and discussed the initial version at the Doctoral Consortium of the CAISE conference (Schön, 2016).

The discussions about the metamodel triggered first improvements. To this end, we started to elaborate the agile RE metamodel iteratively based on our learnings. This approach is similar to the concepts of ASD, where products are created in several iterations. Table 14 shows an overview of the conceptual improvements of the agile RE metamodel according to the events and activities that triggered these improvements. The current version of the metamodel is presented in Figure 17.

Table 14: Overview of conceptual improvements of the agile RE metamodel

Version	Trigger	Improvements
v1.1	Learnings from Doctoral Consortium (Schön, 2016)	Adding relations between metaclasses and extension of attributes in several metaclasses according to ISO 9241-210:2010 (International Organization for Standardization, 2010)
v1.2	Discussion with Dr. Jorge Sedeño López	Extension of attributes and changing names
v1.3	Discussion with Jutta Doetkotte in connection with her bachelor thesis	Extension of attributes and changing names, types and relations between metaclasses
v1.4	Learnings from study “Key Challenges in Agile RE” (Schön et al., 2017d)	Extension of relations, deletion of attributes, changing types of attributes, reorder attributes within metaclasses
v1.5	Learnings from study “Identifying Agile RE Pattern” (Schön et al., 2017b)	Changing attributes, extension of attributes
v1.6	Discussion with supervisors	Deletion of attributes, changing attribute types according to UML notation, adding new metaclasses
V1.7	Discussion with Dr. Nora Koch and supervisors	Extension of relations, deletion of attributes, and adding metaclass Requirement
V1.8	Discussion with Dr. Nora Koch and supervisors	Define metaclass Requirement as external class, update of relations and multiplicities, change ordering of metaclasses, delete redundant attributes

4.3.3. Developing a Profile for the Agile RE Metamodel

A profile provides mechanisms for adapting an existing metamodel with constructs that are specific for a particular domain (Object Management Group Inc. (OMG), 2015). The profile allows the instantiation of the agile RE metamodel by means of a visual language. The visual language can be used by CASE (Computer-Aided Software Engineering) tools. This approach enables us to apply the metamodel to the enterprise environment.

In order to create an profile, a metamodel must be defined as an instance of UML, since this is one level higher than the superstructure specification (Object Management Group Inc. (OMG), 2015). Therefore, the agile RE metamodel will be defined as in instance of UML using MOF (Meta Object Facility). For building our profile using an UML notation, we followed the process of creating a profile described in (García García, 2015). According to the recommendations of Garcia Garcia (García García, 2015), we used the tool Enterprise Architect (EA)⁵ with the MDG Technology Builder.

As a result, we achieved an Add-in for EA. This Add-in can be used to create visual instances of the agile RE metamodel (see section 7.3).

⁵ <http://www.sparxsystems.com/>

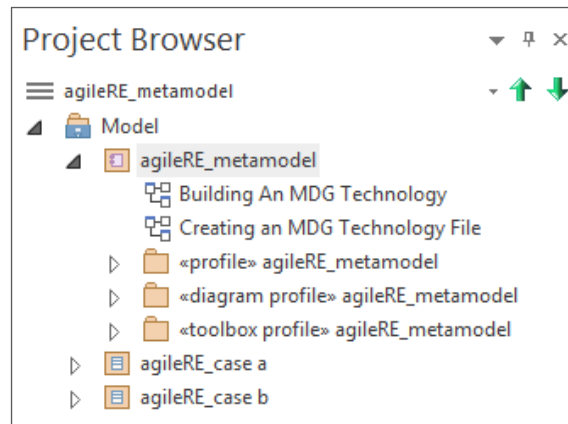


Figure 16: Creating a profile with Enterprise Architect

Figure 16 shows the three main parts of our EA project, which are explained in the following.

Package <<profile>>. The metaclasses of the agile RE metamodel are represented as stereotypes in EA. Each stereotype has a set of corresponding tagged values, which are the attributes of the metaclasses. In addition, each stereotype is linked to an appropriate UML metaclass using the *extend* relationship (see Figure 21).

Package <<diagram profile>>. The diagram profile contains all EA artifacts, which are needed to define diagrams according to the stereotypes of the profile.

Package <<toolbox profile>>. The toolbox profile contains EA artifacts, which are needed to create a customized toolbox according to the defined profile.

4.4. Results and Discussion – Agile RE Metamodel

The agile RE metamodel (Figure 17) visualizes the big picture of agile RE without providing a concrete process model. The metamodel allows analyzing the organizational environment, where the product development takes place, in terms people, relationships, methodologies, and agile techniques. For one thing, we can use the metamodel to derive a process model tailored according to the organizational environment (greenfield approach). For the other thing, we can utilize it for improving existing agile RE process models (brownfield approach).

Moreover, the agile RE metamodel represents a set of generic concepts without describing a process model itself. Therefore, it builds a framework for a process. Hence, the process model is an instantiation of the metamodel. The initial version of the agile RE metamodel is presented in (Schön, 2016), whereas Figure 17 represents the optimized version.

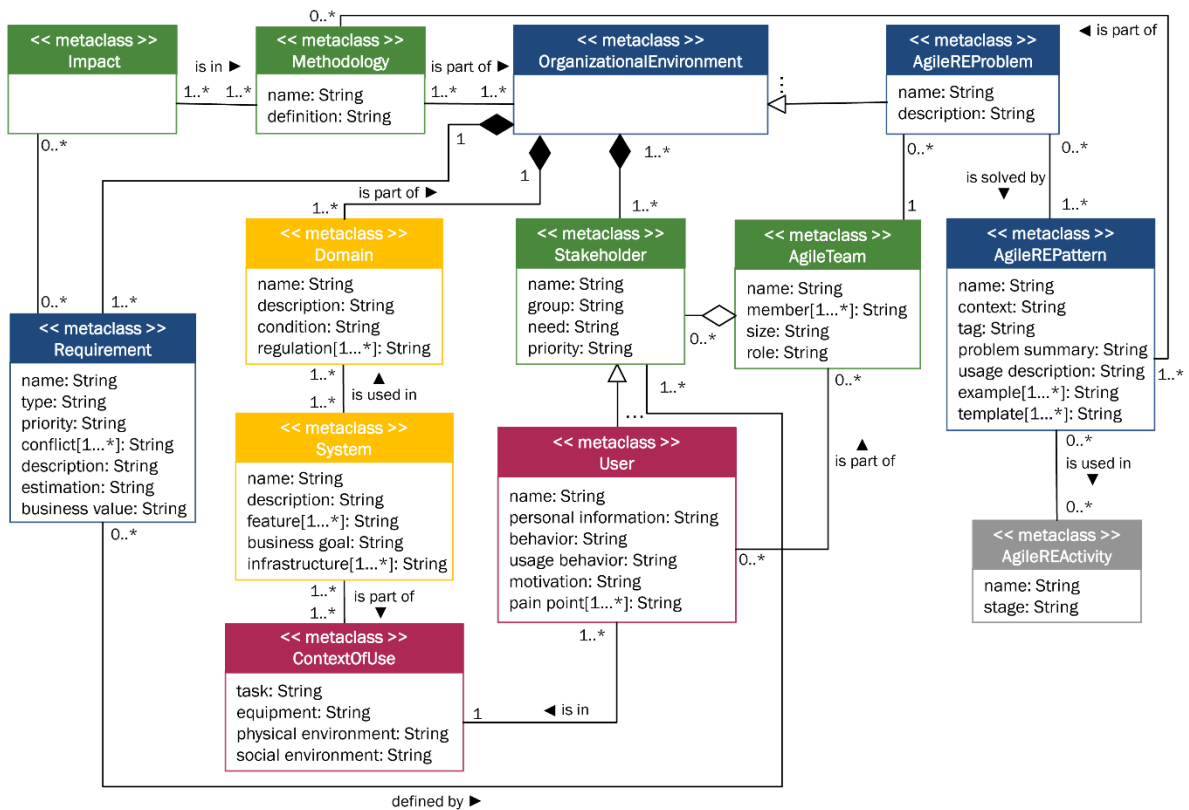


Figure 17: Agile RE metamodel (v.1.8)

The agile RE metamodel looks at the cross-functional area of agile RE. Table 15 presents the meaning of the color coding scheme of the agile RE metamodel. Moreover it explains the effects of different research fields on agile RE and outlines its cross-functionality.

Table 15: Color coding scheme for the agile RE metamodel

Color	Origin/Research Field	Metaclasses
green	ASD	Stakeholder, AgileTeam, Methodology, Impact
pink	HCD (International Organization for Standardization, 2010)	ContextOfUse, User
orange	Domain Driven Design (DDD), (Evans, 2003)	System, Domain
grey	RE	AgileREActivity
blue	Agile RE	OrganizationalEnvironment, AgileREProblem, AgileREPattern, Requirement

4.4.1. Description of the Agile RE Metamodel

In this section we will detail the concepts of the agile RE metamodel (see Figure 17) by explaining the metaclasses and relations among them.

4.4.1.1. Metaclass <<Methodology>>

Description	An important aspect in agile RE is the applied methodology (e.g. Scrum, Kanban, XP, HCD). Each methodology comes with its own requirements and has an impact on how RE is carried out. For instance, there is a difference between time-boxed approaches like Scrum or flow-driven approaches like Kanban. Requirements are managed in a different manner and therefore, the methodology needs to be considered while modeling agile RE.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise description which allows to identify the applied methodology• <i>definition: String [1]</i> description of the rules of the methodology in order to understand the impact on agile RE
Operations	no
Associations	<ul style="list-style-type: none">• <i>is part of: OrganizationalEnvironment [1..*]</i> one or many methodologies define the process model, which is used in one or many organizational environments• <i>is in: Impact [1..*]</i> one or many impacts describe what effect one or many methodologies have on the way requirements are managed. For instance in Scrum, requirements are managed by a Product Backlog, whereas in Kanban, requirements are managed by a Kanban board.• <i>is part of: AgileREPattern [1..*]</i> a methodology can have one or many agile RE Pattern
Restrictions	no

4.4.1.2. Metaclass <<Impact>>

Description	Due to its iterative and incremental character, agile methodologies have an impact on how requirements are managed in an organizational environment. The metaclass Impact is filled during runtime and describes how requirements are managed. For instance in Scrum, requirements are managed by a Product Backlog, whereas in Kanban, requirements are managed by a Kanban board.
Generalization	no
Attributes	The metaclass is filled during runtime
Operations	no
Associations	<ul style="list-style-type: none">• <i>is in: Methodology [1..*]</i> one or many impacts define the manner how requirements are managed in a methodology• <i>requirement [0..*]</i> one or many impacts describe the effect on how requirements are managed.
Restrictions	no

4.4.1.3. Metaclass <<OrganizationalEnvironment>>

Description	The organizational environment describes the surroundings or conditions in which the product development takes place. On the one hand, the organizational environment is changing because of the different people involved in the process (user, stakeholder or agile team). On the other hand, it is changing because of the domain or the agile methodology used. The metaclass OrganizationalEnvironment is also filled during runtime. Moreover, it makes the metamodel very strong and scalable in terms of future extensions. For instance, someone would like to use the agile RE metamodel in an organizational environment where sequential approaches like the waterfall model or the V-model are used. In such environments there are no agile RE problems, but other kind of problems. In this case, we do not have to change the whole metamodel, instead we can add an additional metaclass to the metaclass OrganizationalEnvironment in order to extend the framework with additional features.
Generalization	no
Attributes	Metaclass is filled during runtime
Operations	no
Associations	<ul style="list-style-type: none">• <i>is part of: Methodology [1..*]</i> one or many organizational environments are composed of methodologies, which describe the manner how the system is developed• <i>requirement [1..*] (association of composition)</i> one or many organizational environments have requirements, which are specified for the system to be developed in the organizational environment• <i>is part of: Domain [1..*] (association of composition)</i> one or many organizational environments are embedded in domains• <i>Stakeholder [1..*] (association of composition)</i> organizational environments have one or many stakeholders
Restrictions	no

4.4.1.4. Metaclass <<AgileREProblem>>

Description	The metaclass AgileREProblem describes problems, which occur in an agile environment in terms of RE. Agile RE problems occur during system development in an agile context. Examples of agile RE problems can be “ <i>continuous management of requirements</i> ” (C4) or “ <i>not to lose sight of the big picture during the implementation of complex requirements</i> ” (C3) (see Table 21).
Generalization	Metaclass <<OrganizationalEnvironment>>
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise description which allows to identify the agile RE problem• <i>description: String [1]</i> provides additional information concerning the agile RE problem to make it tangible
Operations	no
Associations	<ul style="list-style-type: none">• <i>agileTeam: String [1]</i> one agile RE problem can belong to one agile team• <i>is solved by: AgileREPattern [1..*]</i> one or many agile RE patterns can be used in order to solve an agile RE problem
Restrictions	no

4.4.1.5. Metaclass <<Domain>>

Description	The requirements for each system differ due to the diverse domains in which the system is used. The metaclass Domain becomes more important in terms of the era of Industry 4.0. As an example, it can be highlighted that there are a lot of different IoT platforms on the market that need to be customized for a specific domain (e.g. automotive, utilities or wind energy).
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise description of the area in which the system is used• <i>description: String [1]</i> provides additional information concerning the area in which the product is used• <i>condition: String [1..*]</i> description of domain specific constraints, which have an impact on the system, for instance caused by changes in market or technology• <i>regulation [1..*]: String [1..*]</i> description of laws, guidelines or standards, which have an impact on the system
Operations	no
Associations	<ul style="list-style-type: none">• <i>is part of: OrganizationalEnvironment: String [1] (association of composition)</i> one domain describes in which area the organizational environment is embedded• <i>is used in: System [1..*]</i> one or many systems are used in a domain
Restrictions	no

4.4.1.6. Metaclass <<Stakeholder>>

Description	A stakeholder is an individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations. This can be for instance management, sales, marketing, or customer. The continuous involvement of stakeholders is very important to ASD in order to develop a system, which meets the expectations and needs of customers and users.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise name of the stakeholder• <i>group: String [1..*]</i> group in which the stakeholder is part of (e.g. marketing, management)• <i>need: String [1..*]</i> related to human needs, which the stakeholder has in terms of system development• <i>priority: String [1..*]</i> describes how relevant the requests are in terms of system development
Operations	no
Associations	<ul style="list-style-type: none">• <i>organizationalEnvironment: String [1..*]</i> (association of composition) one or many stakeholders are part of an organizational environment• <i>agileTeam: String [0..*]</i> (association of aggregation) no or many stakeholders might be part of an agile team. In this case, we choose the association of aggregation since an agile team exist without having a stakeholder• <i>defined by: Requirement [0..*]</i> nor or many stakeholders define requirements
Restrictions	no

4.4.1.7. Metaclass <<AgileTeam>>

Description	The agile team groups those people who are responsible for system development including roles like developer, UX designer, tester, Agile Coach, Scrum Master or Product Owner.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise name of the agile Team• <i>member[1...*]: String [1..*]</i> describes one member of the agile team with all skills s/he has• <i>size: String [1]</i> number of team size• <i>role: String [1..*]</i> describes the roles, which the team is composed of
Operations	no
Associations	<ul style="list-style-type: none">• <i>agileREProblem: String [0..*]</i> an agile team may have agile RE problems, which they need to solve• <i>stakeholder: String [0..*] (association of aggregation)</i> nor or many agile teams can have stakeholders in it. In this case, we choose the association of aggregation since an agile team exist without having a stakeholder• <i>is part of: User [0..*]</i> nor or many agile teams might have users in it
Restrictions	no

4.4.1.8. Metaclass <<AgileREPattern>>

Description	A pattern is composed of a recurring problem and a solution description. In particular, an agile RE pattern is composed of an agile RE problem and one or more agile techniques, which support solving the problem. For instance, the agile RE problem “ <i>continuous management of requirements</i> ” can be solved by the agile RE patterns “ <i>continuous refinement meetings with stakeholders</i> ” or “ <i>sprint review meetings</i> ”.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> name of the agile RE pattern• <i>context: String [1..*]</i> preconditions that describe when to use the agile RE pattern• <i>tag: String [1..*]</i> key word that allows a categorization and findability of an agile RE pattern• <i>problem summary: String [1]</i> short description that provide details about the problem• <i>usage description: String [1..*]</i> provides additional information about how to use the agile RE pattern• <i>example[1...*]: String [1..*]</i> shows an application of the solution in order to provide guidance• <i>template[1...*]: String [1..*]</i> additional material, which eases the application of the agile RE pattern
Operations	no
Associations	<ul style="list-style-type: none">• <i>is solved by: AgileREProblem [0..*]</i> no or many agile RE patterns might be used to solve agile RE problems• <i>is part of: Methodology [0..*]</i> no or many agile RE patterns might be part of methodologies• <i>is used in: AgileREActivity [0..*]</i> no or many agile RE patterns are used in agile RE activities
Restrictions	no

4.4.1.9. Metaclass <<System>>

Description	The system is a combination of hardware, software and/or services, which describe the product.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> name of the system to be developed• <i>description: String [1..*]</i> information that describe the system and its purpose according to the readers needs• <i>feature[1...*]: String [1..*]</i> describes part of the system and provides additional information about the characteristics of it• <i>business goal: String [1..*]</i> objective, which will be addressed by the system• <i>infrastructure[1...*]: String [1..*]</i> describes the surroundings in which the system is embedded
Operations	no
Associations	<ul style="list-style-type: none">• <i>is used in: Domain [1..*]</i> one or many systems are used in a domain• <i>is part of: ContextOfUse [1..*]</i> one or many systems are part of a context of use
Restrictions	no

4.4.1.10. Metaclass <<User>>

Description	The user is a person who interacts with the system. The user is a specialized form of a stakeholder and is valued by an additional metaclass in the agile RE metamodel, since s/he is in the center of product development within a value-driven organization where HCD (International Organization for Standardization, 2010) plays an important role.
Generalization	Metaclass <<Stakeholder>>
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> name of the user• <i>personal information: String [1..*]</i> information that allows to build a picture of the user and his/her personality• <i>behavior: String [1..*]</i> information that describes typical behavior of the user in daily life• <i>usage behavior: String [1..*]</i> information that describes how s/he interacts with the system• <i>motivation: String [1..*]</i> describes the underlying reason for using the system• <i>pain point[1..*]: String [1..*]</i> information about what causes lead to the usage of the system and additional information why the system is relevant
Operations	no
Associations	<ul style="list-style-type: none">• <i>is part of: AgileTeam [0..*]</i> nor or many users might be part of an agile team• <i>is in: ContextOfUse [1]</i> one user is in a context of use while using the system
Restrictions	no

4.4.1.11. Metaclass <<AgileREActivity>>

Description	An agile RE activity is an action that is carried out in terms of RE in an agile environment. Agile RE activities (see Table 1) describe actions that take place in terms of requirements management. In the context of the agile RE metamodel, agile RE activities can be used to categorize agile RE patterns.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> name of the agile RE activity• <i>stage: String [1..*]</i> describes when the activity takes place in terms of phases of project management
Operations	no
Associations	<ul style="list-style-type: none">• <i>is used in: AgileREPattern [0..*]</i> nor or many agile RE activities can categorize agile RE patterns
Restrictions	no

4.4.1.12. Metaclass <<ContextOfUse>>

Description	The user is in a context of use during the usage of the system. The context of use is defined by (International Organization for Standardization, 2010) and comprises: users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a system is used.
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>task: String [1..*]</i> describes the work or interaction which the user carries out with the system• <i>equipment: String [1..*]</i> describes the hardware, software and materials which are used during the interaction with the system• <i>physical environment: String [1..*]</i> describes the physical surrounding in which the system is used (e.g. noise, light conditions)• <i>social environment: String [1..*]</i> describes the social surrounding in which the system is used
Operations	no
Associations	<ul style="list-style-type: none">• <i>is part of: System [1..*]</i> one or many contexts of use can detail a system• <i>is in: User [1..*]</i> one or many contexts of use can detail users
Restrictions	no

4.4.1.13. Metaclass <<Requirement>>

Description	The metaclass Requirement shows how requirements are described in agile RE. This is an external class of the agile RE metamodel (see Figure 17) since this metamodel provides generic concepts, which allow modeling how people work and collaborate in terms of agile RE. The agile RE metamodel does not aim to provide details of what a requirement look like in detail in an agile environment. Therefore, other approaches exist for instance in (Sedeño, 2017), (Sedeño et al., 2017a), or (Escalona and Aragón, 2008).
Generalization	no
Attributes	<ul style="list-style-type: none">• <i>name: String [1]</i> precise name of the requirement• <i>type: String [1]</i> describe the sort of a requirement (e.g. functional, non-functional)• <i>priority: String [1]</i> give guidance on how urgent the implementation should be done• <i>conflict[1..*]: String [1..*]</i> shows dependencies to other requirements and the needs of different stakeholder• <i>description: String [1]</i> written details to explain the requirement• <i>estimation: String [1..*]</i> describes how much effort it will cost to implement the requirement• <i>business value: String [1]</i> describes the value that will be added to the product after implementation of the requirement• <i>requester: String [1..*]</i> person or organization who ask for the implementation of the requirement
Operations	no
Associations	<ul style="list-style-type: none">• <i>impact [1..*]</i> one or many requirements can be effected by an impact• <i>organizationalEnvironment [1]</i> one requirement can be part of an organizational environment• <i>defined by: Stakeholder [1..*]</i> one or many requirements are defined by stakeholdes
Restrictions	no

Examples for each metaclasses are provided in section 7.3. These examples show how the metamodel can be applied in a real world context. In the following section, we will provide detailed views on important metaclasses of the agile RE metamodel by means of providing additional views.

4.4.2. Additional Views on the Agile RE Metamodel

As mentioned before, the human-centric view plays an important role in terms of agile RE. To this end, this section presents additional views on the agile RE metamodel in order to detail the human aspects of the agile RE metamodel.

For modeling and documentation of information concerning users, stakeholders or the agile team exposed during product development, the agile technique *personas* (Cooper, 1999) can be used. In addition, personas can be utilized for modeling user needs and stakeholder values (Sedeño et al., 2017a).

4.4.2.1. Metaclass User

The user is a person, who directly interacts with the product (International Organization for Standardization, 2010). In terms of product development, it is necessary to understand the needs of the user in order to develop a product, which is valuable to the target group. From a HCD perspective, the analysis of the target group is carried out by means of conducting user research. To this end, the user should be modeled with at least some basic information which is listed in Figure 18 as enumerations.

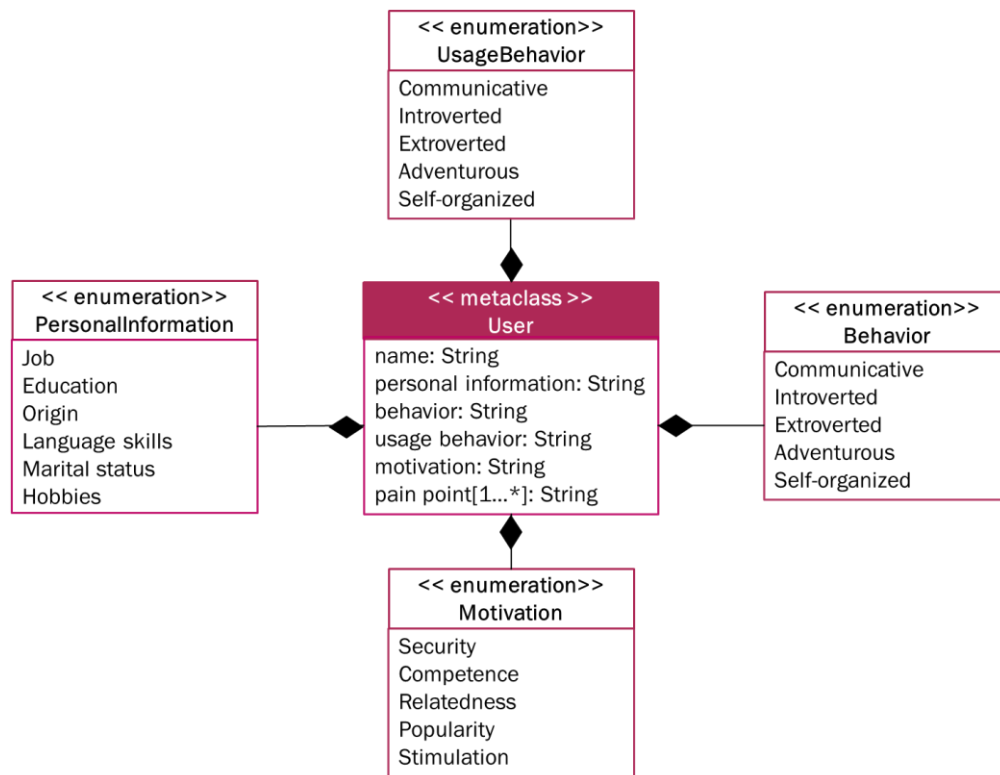


Figure 18: Detailed view on metaclass <<User>>

The personal information allows creating a picture of a concrete person with the aim to avoid self-referential design during product development. Moreover, the usage behavior helps people to understand how one specific user interacts with the product. The description of the behavior constitutes the picture of a concrete user and enables project members to generate an empathetic

focus on user needs and values. With regard to the enumeration *Motivation*, the motivational aspects relies on the concept of human values proposed by (Hassenzahl, 2010).

4.4.2.2. Metaclass Stakeholder

The stakeholder is an individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations (International Organization for Standardization, 2010). Figure 19 highlights some details that make the definition of stakeholders more clear.

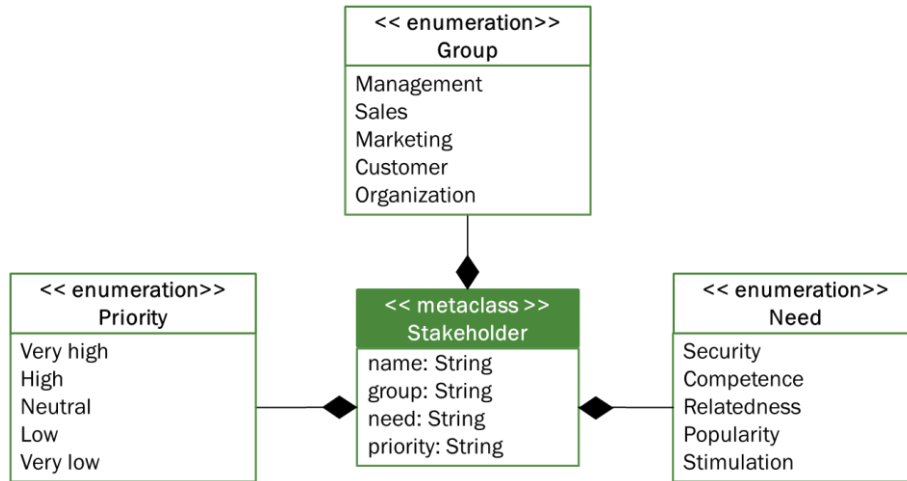


Figure 19: Detailed view on metaclass <<Stakeholder>>

A stakeholder is part of a group for instance management, sales or marketing. In addition, s/he has a priority which categories are detailed by the corresponding enumeration (see Figure 19). Similar to the motivational aspects of the user, the need is also based on the concept of human values proposed by (Hassenzahl, 2010).

4.4.2.3. Metaclass Agile Team

The metaclass *AgileTeam* details the composition of the product development team, who actually does the work of creating a product. Therefore, the enumeration *role* (see Figure 20) explains which kind of roles should be part of an agile team.

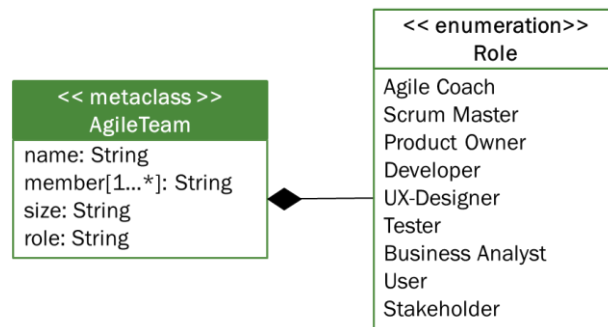


Figure 20: Detailed view on metaclass <<AgileTeam>>

4.4.3. Profile for Agile RE Metamodel

In addition to the agile RE metamodel we created a modeling language by means of a profile. As mentioned before, a profile provides mechanisms for adapting an existing metamodel with constructs that are specific for a particular domain (see section 4.3.3). To this end, a profile allows us to extend the metaclasses of the agile RE metamodel with the aim to tailor domain specific models of the metamodel by means of using a common visual language. This visual language is beneficial in terms of analyzing an organizational environment in a systematic manner, since it enables identifying dependencies among people working in the organizational environment as well as the impact, which occurs due to applied methodologies (see section 7.3).

Figure 21 presents the profile for the agile RE metamodel using an UML notation. The stereotypes are the metaclasses of the agile RE metamodel, whereas the tagged values are the attributes of the agile RE metamodel.

Each stereotype is extended from the UML metaclasses. The UML metaclass <<Enumeration>> is extended by the stereotypes *Impact* and *Methodology*. Moreover, we extend the metaclass <<Class>> by the stereotypes *OrganizationalEnvironment*, *AgileREProblem*, *AgileREPattern*, *Domain*, *System*, and *ContextOfUse*. The metaclass <<Actor>> is extended by the stereotypes *Stakeholder*, *User* and *AgileTeam*. In addition we extend the metaclass <<Activity>> by the stereotype *AgileREActivity*.

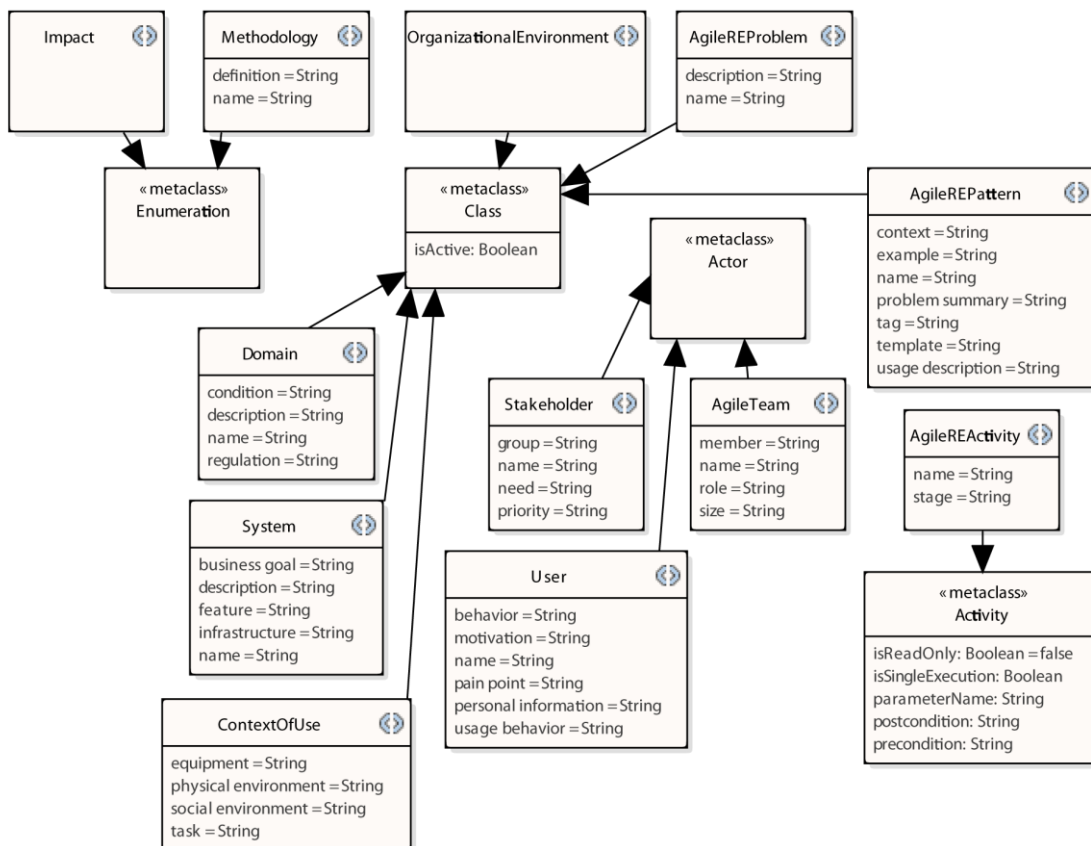


Figure 21: Profile for agile RE metamodel

We present two examples of domain specific models of the agile RE metamodel, derived by using this profile (see Figure 21) in section 7.3.

In literature, there can be found further examples of profiles addressing different objectives. For instance, Basso et al. (Basso et al., 2015) propose a profile for privacy-aware applications. Moreover, Magureanu et al. (Magureanu et al., 2012) contribute a profile, which supports the application design for Cyber Physical Systems (CPS). In addition, Escalona et al. (Escalona and Aragón, 2008), (Escalona and Koch, 2007) contribute a profile for modeling web requirements.

4.4.4. Impact and Limitations

One of the most important benefits of the proposed metamodel for agile RE is cross-functionality. Agile methodologies are based on cross-functional collaboration of people from different disciplines (e.g. programming, test or UX design). For becoming a value-driven organization, it is essential to consider all the disciplines while creating a process model for agile RE. The proposed metamodel for agile RE supports the understanding of this complexity and enables companies to become clear about the influencing parameters. In addition, researchers as well as practitioners can use the metamodel for evaluating their existing process models for agile RE aiming to improve them. In this context, the profile enables analyzing the organizational environment by means of building visual instances of the metamodel.

Practitioners can use the metamodel for instance during their retrospective meetings where obstacles with the applied processes are discussed. The purpose of a retrospective is to inspect and adapt the current organizational environment. Therefore, the agile RE metamodel allows to reflect current problems and to express them verbally. This supports the analysis in terms of what influencing factors might be changed in order to handle the problems and to increase value delivery.

Deriving a process model for agile RE based on the proposed agile RE metamodel might still be a challenge in some organizations due to the complexity of the environment (e.g. scaled organization or other teams working with sequential approaches like waterfall model). Nonetheless, the metamodel reduces this complexity by visualizing the influential parameters that can be changed in terms of the explicit view.

Someone might argue that creating a metamodel for the research field of ASD is not appropriate due to the assumed inflexibility and the effort, which occur in terms of applying the metamodel to an organizational environment. Nevertheless, we understand metamodeling as an approach for building an abstract view on the complex field of agile RE. Moreover, the agile RE metamodel allows us to analyze the organizational environment with the aim to improve it in terms of value delivery.

The idea of creating the agile RE metamodel and a profile was inspired by the work of Escalona et al. (Escalona and Aragon, 2008), (Escalona and Koch, 2007), who contribute important work in the field of Model-Driven Web Engineering (MDWE).

4.5. Chapter Summary

This chapter contributes a metamodel for agile RE (*agile RE metamodel*), aiming to build an abstraction layer about the variety of existing process models in the field. The agile RE metamodel allows us to analyze the environment of an organization in terms of value delivery and collaboration. Moreover, we contribute a visual language for the agile RE metamodel by means of a profile. The profile can be used to tailor domain specific models.

The initial version of the agile RE metamodel is inferred by an analysis of related process models for agile RE and experiences working in industry projects. We improved the metamodel iteratively by means of discussions with other researchers and further empirical studies.

The agile RE metamodel builds the heart of our framework and shapes the conceptual approach for agile RE management. It allows us to reflect the dependencies among people and product development process and supports improvement by inspection and adaptation.

The metamodel has several implications for both, researchers and practitioners. On the one hand, the metamodel can be used to model new process models for agile RE that are strongly focused on value delivery and therefore can be classified as value-driven process models. On the other hand, existing process models can be evaluated as well as improved by the metamodel.

The following chapters will propose techniques to apply the presented conceptual approach. Thus, we will identify the most important problems in agile RE industry has to face today as well as create agile RE patterns that allow to solve the identified agile RE problems.

In the previous chapters we explained our conceptual approach by both introducing the agile RE metamodel and a profile. The main part of the agile RE metamodel builds the organizational environment, which is composed of the surrounding in an agile organization. Applying agile methodologies lead from time to time to recurring problems in terms of agile RE. As explained by the metamodel agile RE problems can be solved by agile RE patterns.

For an agile transformation, organizations need to continuously improve their established approaches to RE as well as their approaches to software development. This is accompanied by some problems in terms of agile RE. The agile RE problems have different rationales. Nevertheless, often they can be abstracted to a similar core. This chapter aims to identify the most important problems in terms of agile RE industry has to face today resulting in a catalogue of agile RE problems. The identification of agile RE problems allows agile practitioners as well as researchers to improve their agile environment since the knowledge supports them to understand and to reflect their problems.

In this chapter, we will provide techniques to apply our conceptual approach to the enterprise environment. We present the results of our iterative expert judgement process, comprising three complementary rounds (Schön et al., 2017d). As mentioned before, research about agile RE is very close to existing work practices in companies. Therefore, we conducted this study with a panel consisting of 26 experts working in the field of ASD.

Section 5.1 summarizes the reported problems in agile RE by related work and presents what approaches were used in order to identify these problems. Then, section 5.2 discusses on the gaps in existing literature. In section 5.3, we will present our research design by explaining the iterative expert judgement process and the three rounds. Subsequent, in section 5.4 we discuss the results, meaning of findings and limitations of this study.

5.1. Summary of Related Work

Performing agile RE can lead to problems organizations have to deal with. In literature, there can be found some studies investigating problems in agile RE by means of different research methods. Table 16 shows an overview of the reported challenges and used research methods.

Table 16: Problems in agile RE reported by related work

Authors	Research method	Reported problems
Ramesh, Cao, Baskerville (Ramesh et al., 2010)	Multi-case study (16 companies)	Problems with cost and schedule estimation; inadequate or inappropriate architecture; neglect of non-functional requirements; customer access and participation; prioritization on a single dimension; inadequate requirements verification; minimal documentation
Bjarnason, Wnuk, Regnell (Bjarnason et al., 2011)	Case study	Planning for agility; weak requirements prioritization; weak effort estimates; quality issues; system completed late; capturing innovation; lack of documented requirements; customer-proxy role; ensuring competence (RE, VV); motivating teams for requirements work; weak requirements at start
Inayat, Salim, Marczak, Daneva, Shamshirband (Inayat et al., 2015)	Systematic Literature Review	Minimal documentation; customer availability; inappropriate architecture; budget and time estimation; neglecting non-functional requirements (NFRs); customer inability and agreement; contractual limitations; requirements change and its evaluation
Heikkila, Damian, Lassenius, Paasivaara (Heikkilä et al., 2015)	Mapping Study	Problems with client or customer representatives; insufficiency of user story format; difficulties in prioritization of requirements; growing technical debt; reliance on tacit requirements knowledge; imprecise effort estimates
Soares, Alves, Mendes, Mendonca, Spinola (Soares et al., 2015)	Systematic Literature Review	Requirement prioritization; non-functional requirements identification; lack of information; volatility of requirements; requirements definition; dependence among requirements; prediction of impacts of changes; user dependence; communication and collaboration with users; requirements validation

5.2. Gap Analysis of Related Work

Analyzing the related work, we can state that the authors use two different kinds of research approaches in general. On the one hand, Ramesh et al. (Ramesh et al., 2010) and Bjarnason et al. (Bjarnason et al., 2011) utilize case studies to investigate the problems in the field. On the other hand, Inayat et al. (Inayat et al., 2015), Heikkila et al. (Heikkilä et al., 2015) and Soares et al. (Soares et al., 2015) report problems in agile RE by analyzing primary studies with the aim to identify available evidence in existing research.

Ramesh et al. (Ramesh et al., 2010) results were published in 2010. However, as ASD is a rapidly changing research area and the body of knowledge has evolved over the last years, we need to clarify whether the reported problems are still relevant today. For instance, NFRs may not be longer a problem for industry since the concept of the Definition of Done and the usage of acceptance criteria are widely spread. Bjarnason et al. (Bjarnason et al., 2011) carry out a case study in only one company, therefore the results may not be applicable to other companies and may not be representative in general. In comparison, Inayat et al. (Inayat et al., 2015), Heikkila et al. (Heikkilä et al., 2015) and Soares et al. (Soares et al., 2015) review primary studies by analyzing existing literature, which is a good approach to get an impression of relevant aspects from a

theoretical viewpoint. Nevertheless, one could argue that this is not an appropriate approach to investigate the existing problems in practice.

To this end, the aim of this chapter is to identify the most important problems in agile RE industry has to face up today by getting insights from 26 experts in the field. To the best of our knowledge there is no existing study investigating these problems by means of a qualitative study with practicing experts in ASD working for many different companies.

5.3. Research Method

We used an iterative expert judgement process rooted in a Delphi study (Dalkey and Helmer, 1963), (Diamond et al., 2014), (Linstone and Turoff, 2002) in order to identify the most important problems industry has to struggle with in terms of agile RE. Figure 22 shows the four phases of a Delphi study and the tasks, which had been carried out.

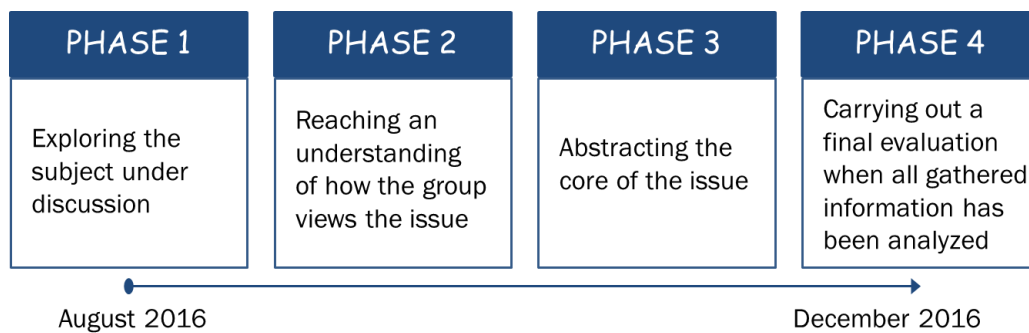


Figure 22: Phases of a Delphi study

Delphi studies are used across different domains in order to obtain the opinion of a panel of experts in an iterative structured manner by means of a series of questionnaires. They had been used since the 50s by the RAND Corporation in the military area (Dalkey and Helmer, 1963).

Linstone and Turoff (Linstone and Turoff, 2002) define the technique as follows: “*Delphi may be characterized as a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem.*”

Classically the method aims to produce consensus among the panel of experts concerning the subject under investigation. Therefore, different metrics like Fleiss’ Kappa (Fleiss, 1971) or Kendall’s coefficient of concordance (Legendre, 2005) are utilized. Recent studies reveal that the definition of what constitutes consensus is less clear (Diamond et al., 2014).

During our study, we recognized that we were able to achieve more valuable insights by adapting the questionnaire of each round to the objectives of the study. To this end, we applied a modified Delphi study where measuring consensus and stability at group level among several iterations was not the most crucial part. On the contrary, we shifted the focus to applying the valuable features of Delphi for conducting our iterative expert judgement process (Dalkey, 1969):

- Anonymity among experts to avoid influence of dominant individuals
- Iterative approach
- Controlled feedback with statistical group response

The main benefit of our modified approach was utilizing the learnings from a previous iteration for carrying out the following ones.

5.3.1. Objectives and Research Questions

The study pursues the main objective of identifying the most important problems in agile RE industry has to address today. We aim to build a shared understanding concerning these problems among voices that matter by means of experts in the field of agile RE. Thus, the research questions we pose are listed below:

- RQ-5.1: What are the key problems in Agile Requirements Engineering?
- RQ-5.2: How can we deal with the identified key problems?

5.3.2. General Study Design

The study was performed in three complementary rounds. Figure 23 gives a general overview of the process. At the beginning of each round, we started designing the questionnaire, optimized by a pretest. Once finished, the invitation was sent to the experts via email. In the second and third round, we attached the results of the previous rounds to the invitation in order to share the outcomes among the panel. The experts had two weeks to fill in the questionnaire. During the following two weeks we evaluated the results, created the report, specified the criteria for dropping items for the following round and designed the questionnaire for the next round.

We conducted the study in German since most of the experts are native speaker. Since we are aware that the term agile RE is not very accepted in the agile community and some experts understand this as a contradiction in itself, we decided not to ask for problems in agile RE directly. On the contrary, we phrased our questions differently and described the context of our study within the introduction part of each questionnaire.

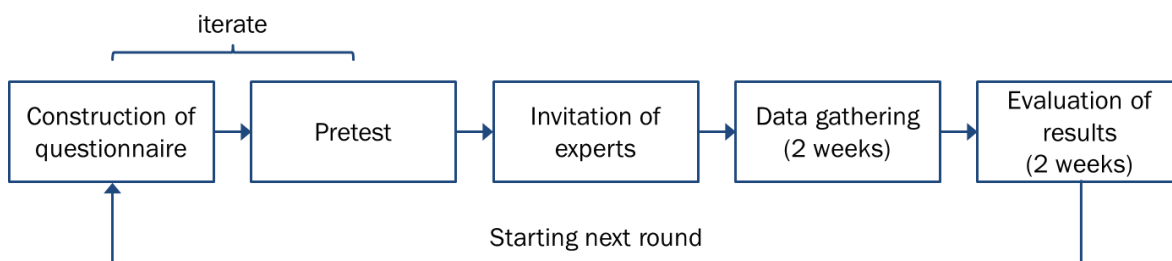


Figure 23: General process of study

We used google forms for the first and second round, whereas limesurvey was used for the third round due to the complexity of the questionnaire. In general, we decided to use 7-point Likert items since this has been proven to be the best choice in terms of avoiding interpolations within related research fields (Finstad, 2010). Besides, we adapted the quality criteria proposed by Diamond et al. (Diamond et al., 2014) so as to ensure the quality of our study.

Table 17 highlights parameters concerning the three rounds of the study. In the beginning, we sent out 54 invitations in order to recruit our panel. In the first round, we received responses from 26 experts, who will be referred as panel. The panel was quite stable among the second and third round, where we have response rates of 88.5% and 84.6%. Literature recommends to have a

minimum of 10-15 panelists, for reliable results of a Delphi study (Lilja et al., 2011). In the last round, we received responses from 22 experts which fit this requirement of reliability.

Table 17: Overview of rounds

Parameter	Round 1	Round 2	Round 3
Period of data gathering	2016-08-16 until 2016-08-29	2016-09-14 until 2016-09-28	2016-10-26 until 2016-11-11
Number of invitations	54	26	26
Responses	26	23	22
Response rate	48.1%	88.5%	84.6%

5.3.3. Panel of Experts

We selected our panelists specifically for their knowledge or position regarding the issue under study. As shown previously, the research field of agile RE is very close to existing work practices in industry (see section 2.2). To this end, we defined the reproducible criteria for selecting participants as follows:

- Many years of experience as professional in the field of ASD. With this criterion we aim to avoid bias from theoretical experts who may state a consolidated view gained from literature.
- Working experience in one or more of the following roles: Product Owner, Scrum Master, Agile Coach, Consultant for Agile Transition, Kanban Expert or Lean Startup Expert. We choose these roles due to their responsibilities they have to cope with problems in terms of Agile RE.

We recruited the panel within our network since we wanted to ensure that each expert has the required knowledge to answer our questions and to provide valuable insights into the subject under study.

The panel consisted of 26 experts who are working in 19 different companies located in Germany and Switzerland. The companies differ in terms of their employees from small to large organizations and spread over different sectors (e.g. consulting, e-commerce, GIS provider, cyber security, publishing industry, or fiscal). The experts referred the following job titles: Consultant (7x), Scrum Master / Agile Coach (3x), Head of Digital Media (2x), Head of Project Management (2x), Product Owner (2x), Project Manager (2x), Software Engineer (2x), Director Consulting Services (1x), Director of a company (1x), Head of UX (1x), UX Enthusiast (1x), Team Coach (1x), and Solution Architect (1x).

Figure 24 displays the type of organization the different panelists are working for. It can be observed that most of the experts (20 out of 26) working for companies in the private sector. No expert is working in for an university or a research institution.

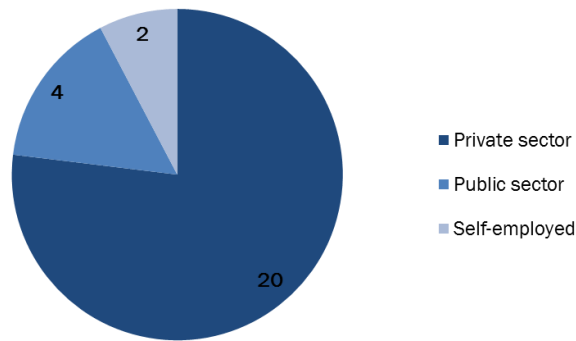


Figure 24: Type of organization for which experts work

In general, the experts had 2-10 years of experience working in ASD (average = 6.14 years). In comparison, experts have about 0-16 years of experience with RE (average = 6.65 years). Even though one expert stated that he had no experience with RE at all, we decided to include his answers into the study, since he has long experience in ASD and in general there do not exist a specific role of a requirements engineer.

Figure 25 shows the kind of process models experts have been working with. It is worth mentioning that most of the experts have experience both with sequential approaches and with agile approaches.

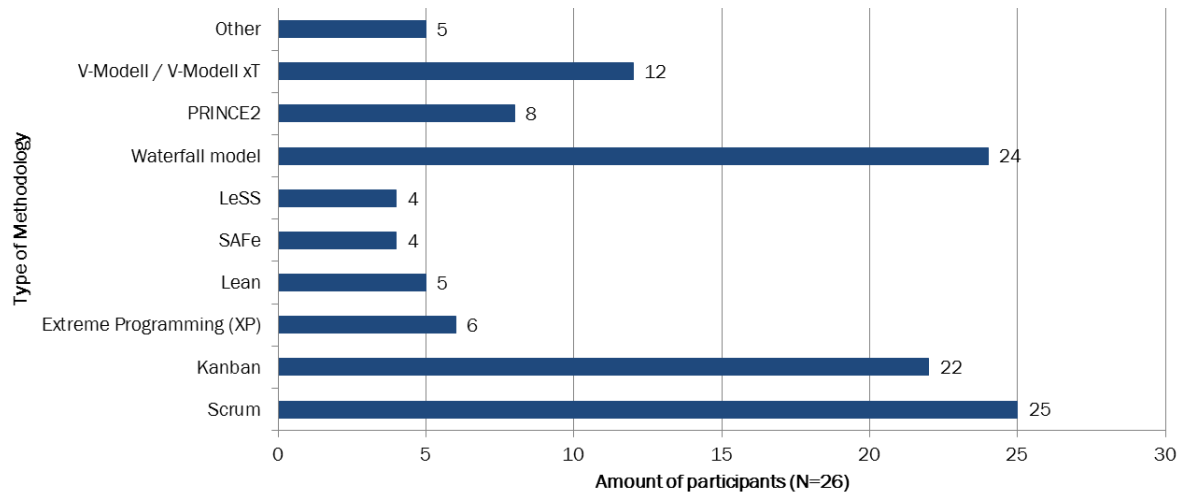


Figure 25: Process models used by experts

In addition, Table 18 displays the know-how level in terms of ASD rated by experts themselves. In sum, 84.6% of the experts rate their know-how either as high or very high. None of them assess his/her knowledge as poor.

Table 18: Know-how of panel (N=26) in terms of ASD

know-how	1	2	3	4	5	know-how
very poor	0.0%	0.0%	15.4%	69.2%	15.4%	very high

5.3.4. Round 1

Questionnaire Design

The questionnaire of the first round comprised two sections. The first section queried personal information for the characterization of each expert (summarized in section 5.3.3). The second section contained two open questions, repeated 15 times (see Figure 26). The first question queried the most important challenges with requirements in terms of ASD. The second question asked for a statement for each challenge to clarify why the expert considered this challenge as important. The minimum number of required answers was 3, whereas the maximum was 15. In sum, we received 107 answers (items) from 26 experts.

Herausforderungen im Umgang mit Anforderungen in der Agilen Softwareentwicklung

*Required

Was sind aus Ihrer Sicht die größten Herausforderungen im Umgang mit Anforderungen im Kontext der Agilen Softwareentwicklung (Beschreiben Sie mindestens 3, jedoch maximal 15)?

Geben Sie ein kurzes Kommentar zu jeder Herausforderung, wieso Sie diese als wichtig erachten.

1. Herausforderung

1.a) Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung? *

Your answer

1.b) Wieso erachten Sie Herausforderung 1a als wichtig? *

Your answer

2. Herausforderung

2.a) Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung? *

Your answer

2.b) Wieso erachten Sie Herausforderung 2a als wichtig? *

Your answer

Figure 26: Excerpt of questionnaire of round 1

Data Analysis

With respect to data analysis, each challenge was categorized by the authors (see (Schön et al., 2017d)) during a workshop. Those items, which could not be categorized easily, were discussed within the group of authors. We used the following categories: stakeholder and user involvement, collaboration within the team, vision and big picture, iteration planning and estimation, granularity of requirements, dependencies of requirements, understanding agile and agile values, continuous delivery of value, roles and responsibilities, need for security, requirement validation, RE methods, format of requirements, clarity of requirements, prioritization, refinement, discovery and transparency. Additionally, the reported challenges were categorized according to their agile RE activity (see Table 1).

Results

As mentioned before, we carried out the study in German. The results of round 1 are the categorized items stated by the experts, which are mostly written in German. Table 19 shows an example of an item consisting of a challenge and a statement concerning importance.

Table 19: Exemplary item in round 1

Question round 1	Answer given by expert
What challenge do you perceive with requirements in terms of Agile Software Development?	Stakeholders affected by requirements or changing the system are not involved.
Why do you consider this challenge as important?	In one of my projects, representatives of end users did not really know the pain of end users. Even the early UI prototypes were tested by incorrect stakeholders, which led to risks of conflicts and failure.

For avoiding bias, we did not translate all the statements. Nevertheless, the categorized items can be found in Appendix IV. In addition, the result report of round one can be found in (Schön et al., 2017e).

5.3.5. Round 2

Questionnaire Design

We checked each item of round one critically, whether or not it was appropriate for answering our RQs and being queried in the next round. Thus, items of round 1 were consolidated or excluded. Subsequent, we reduced the contextual information given by the expert and merged related items, in order to find the core of the stated issues.

In the end, we identified 34 items as relevant for assessing them in round 2. Based on those items, we created the questionnaire for the second round. The resulting questionnaire assessed 34 items related to the following topics:

- stakeholder and user involvement (6 items)
- understanding agile and agile values (6 items)
- RE methods (10 items)
- iteration planning and estimation (6 items)
- format of requirements (6 items)

The items were grouped into sections by the aforementioned topics. After each section, the experts had the opportunity to give additional comments in free text form. They rated each item using 7-point Likert items (see example in Figure 27). Moreover, they could choose giving no statement.

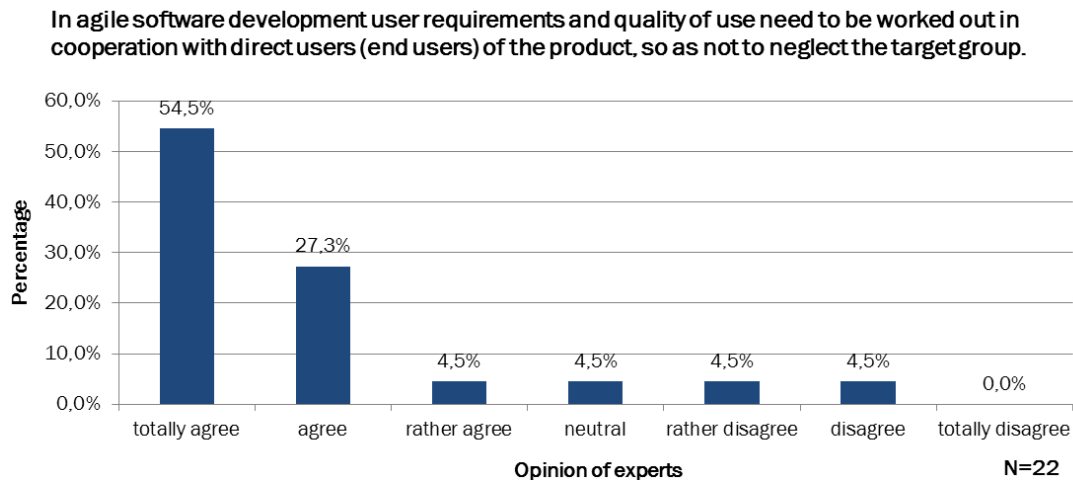


Figure 27: Exemplary item of round 2

Data Analysis

To sum up, we received responses from 23 experts. For each item we calculated mean, variance and standard deviation. Additionally, we created a diagram showing the distribution of experts' opinion (see example in Figure 27) and discussed on the meaning of findings.

Results

Calculation of mean, variance, standard deviation and the diagram of experts' opinion can be found in Appendix IV. The result report of round two can be found in (Schön et al., 2017f).

Stakeholder and User Involvement. Experts agree that it is necessary to have a direct involvement of user and stakeholder so that product development will succeed and user's needs can be fulfilled (Item 1.1, Item 1.2). In this context, there is a high correlation to both the type of product and the reachability of the target group (see comment expert 9). In addition, lightweight methods need to be applied in order to involve users and stakeholder (see comment expert 15). Furthermore, the continuous communication is an important aspect in this context (Item 1.6). The experts agree that face-to-face communication is an appropriate technique not to miss important expectations from different stakeholders (Item 1.3). Moreover, we can state that the experts differ whether the requirements need to be validated with direct end users before development can start (see Item 1.4). This may correlate with the culture of a company in terms of HCD (International Organization for Standardization, 2010). Moreover, the experts have a different opinion concerning the moment of receiving feedback from the stakeholder (Item 1.5).

Understanding Agile and Agile Values. The experts agree that in ASD a lot of decisions are in responsibility of the development team. Therefore, stakeholders need to understand that they cannot take part in every decision (Item 2.3). However, decisions concerning the change of scope and features of the product should be discussed with stakeholders (see comment expert 23). Moreover, the panel agrees that stakeholder do not comprehend that requirements should be negotiable with the development team (Item 2.1). In terms of this, expert 7 stated that the development team should decide about technical realization, whereas the Product Owner together with the stakeholders should take the decision concerning functional questions. In ASD there is often a kind of misinterpretation and Agile is equalized to chaos of requirements or missing documentation. This leads to resistance due to loss of control and fear of increasing risk (Item 2.5). In terms of this, there is a strong correlation to the phase of agile transition of an organization (see comment expert 24). In general, the panel differs concerning a pre-performed requirement analysis (Item 2.2). However, expert 23 state that it is useful to consolidate business requirements in the beginning, without being too detailed.

Requirements Engineering Methods. The panel strongly agrees that in ASD the continuous management of requirements is important since not all of them are fixed in the beginning and they may change over the course of time (Item 3.4). Additionally, experts agree upon that established techniques for elicitation and evaluation of requirements are often too slow (Item 3.5). Techniques need to be applied, which support the sharing of knowledge among the whole development team (Item 3.6). Furthermore, we can observe strong agreement among the panel that in ASD requirement documents need to be adapted to changing conditions with low effort (Item 3.10). The experts refuse that in ASD do not exist a technique to define NFRs (Item 3.9). This may lead to the assumption that there is no longer a problem concerning the handling of NFRs in ASD. Expert 16 recommends the following best practices for handling NFRs: Definition of Done, smoke test, performance test and penetration test. Moreover, the experts agree that missing styleguides and prototypes can lead to the problem that developers implement their own assumptions of UX, which can result in a bad UX for users (Item 3.2). In summary, we can conclude that the established understanding of RE and known techniques need to be adapted to the context of ASD.

Iteration Planning and Estimation. Experts agree that it is important to focus on the refinement of requirements for the short-term iterations in order to react flexible on new information (Item 4.1). Moreover, an outlook on the upcoming iterations is important for coordinating related projects. Nevertheless, this outlook should not be a binding one (Item 4.3). The panel agrees that requirements should be split in such a manner that they can be implemented in one iteration by adding value to the product (Item 4.5). Furthermore, the sight of the big picture should not be loosed due to the complexity of requirements (Item 4.6). The results have impact on the modeling of agile RE process models and how requirements are refined. Figure 28 displays the degree of refinement of requirements over the course of time.

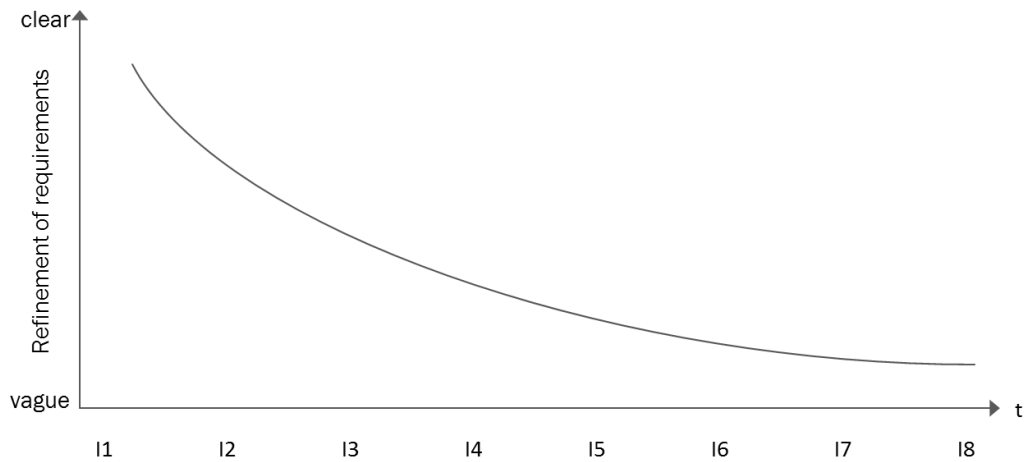


Figure 28: Degree of refinement of requirements

Format of Requirements. The panel agrees that requirements must be formulated as objectives that describe the problem area so that the creativity in solution finding is not restricted (Item 5.3). In addition, the benefits of a requirement must be justified in order to make added value of the implementation clear as well as decisions for a specific requirement comprehensible (Item 5.6). Experts agree that requirements must be captured in such a way that detailed test cases can be derived from them for quality assurance (Item 5.4). Requirements must be formulate in a clear manner in order to avoid uncertainty in terms of implementation (Item 5.5). Moreover, the experts agree that in ASD functional or technical dependencies lead to a high coordination effort (Item 5.1). The additional comments stated by the experts reveal that the items related to the category format of requirements are also applicable in project using non-agile approaches (see comment expert 11 and expert 12).

5.3.6. Round 3

Questionnaire Design

Analyzing the results of round 2, we were not able to clearly answer our predefined RQs (see section 5.3.1). We had some assumptions about the problems in agile RE that need to be verified. To this end, we created a modified questionnaire for the third round.

We reduced the number of items when designing the questionnaire. Considering items from round 2, we assessed each item according to a) its relevance in terms of our RQs, b) the importance in terms of the attributes of agile RE, c) the opinion of the experts and the comprehensibility of the items. Therefore, each author evaluated the items of the second round by means of these criteria. Afterwards, we discussed the evaluation and decided which items to include in round 3. In addition, we optimized the items.

The final questionnaire comprised two parts. The first part queried in sum 20 potential key problems of agile RE related to the following topics: stakeholder and user (3 items), requirements management (7 items), methods and artifacts (5 items) and format of requirements (5 items). The experts were asked to rate each item, whether or not it is a problem in agile RE. Moreover, they had the option to choose giving no statement. Then, the second part evaluated those items that experts identified as problem in terms of importance, following 7-point Likert items

(totally important, important, rather important, neutral, rather unimportant, unimportant, totally unimportant, no statement). In addition, experts optionally had the chance to provide a solution for solving the problems.

Data Analysis

In sum, 22 experts filled in the questionnaire. We classified each of the 20 items as problem in Agile RE since we derived all items from the results of the previous rounds. Besides, we calculated the number of experts who rated each item as a problem. Then, we defined problems as key in those cases where 2/3 of the experts' answers were: "Yes, it is a challenge". Finally, we calculated the importance for those items.

Results

Summarizing the results of the three complementary rounds, we derived 20 problems that companies have to cope with in terms of agile RE. The result report of round three can be found in (Schön et al., 2017g). Table 20 highlights the agile RE problems (translated items). The six key problems are highlighted in green, whereas the ones where less than 1/3 of the experts rate as problem are highlighted in yellow. The recommended techniques to cope with those problems can be found in Appendix IV. We did not translate the whole responses due to of avoiding bias.

Table 20: Problems in agile RE (key problems are highlighted in green, whereas problems where only 1/3 of the experts rate as problem are highlighted in yellow)

ID	Problems in agile RE	N	Yes	No
C1	In agile software development functional or technical dependencies with other teams are a challenge because a considerable coordination effort is required.	17	14 (82.4%)	3 (17.6%)
C2	In agile software development it is a challenge that stakeholders understand that the development team can make independent (detailed) decisions.	20	15 (75.0%)	5 (25.0%)
C3	In agile software development it is a challenge not to lose sight of the big picture during the implementation of complex requirements.	20	15 (75.0%)	5 (25.0%)
C4	In agile software development continuous management of requirements is a challenge since not all of them are fixed at the beginning and they may change over the course of the project.	22	16 (72.7%)	6 (27.3%)
C5	In agile software development it is a challenge to work out user requirements and quality of use in cooperation with direct users (end users) of the product.	18	13 (72.2%)	5 (27.8%)
C6	In agile software development it is a challenge to involve stakeholders throughout the whole development process in regular iterations so that product development will succeed.	20	14 (70.0%)	6 (30.0%)
C7	In agile software development it is a challenge that the requirements to be implemented are clearly defined from the development start since the priorities often change in the short term.	21	13 (61.9%)	8 (38.1%)
C8	In agile software development it is a challenge to analyze requirements with regard to the past development in order to avoid side effects.	15	9 (60.0%)	6 (40.0%)

C9	In agile software development it is a challenge to formulate requirements as objectives that describe the problem area so that the creativity in solution finding is not restricted.	22	13 (59.0%)	9 (41.0%)
C10	In agile software development it is a challenge to slice requirements in such a way that they offer added value for the product.	20	11 (55.0%)	9 (45.0%)
C11	In agile software development it is a challenge to justify the benefits of the requirements in order to make the added value of the implementation clear as well as decisions for a specific requirement comprehensible.	21	11 (52.4%)	10 (47.6%)
C12	In agile software development it is a challenge to document changes to the requirements comprehensibly.	18	9 (50.0%)	9 (50.0%)
C13	In agile software development it is a challenge to establish non-functional requirements.	19	9 (47.4%)	10 (52.6%)
C14	In agile software development it is a challenge to focus only on the refinement of the requirements for the short-term iterations.	22	10 (45.5%)	12 (54.5%)
C15	In agile software development it is a challenge to develop an outlook on the next iterations without making it a binding one.	21	9 (42.9%)	12 (57.1%)
C16	In agile software development it is a challenge to design requirement documents in such a way that they can be adapted to changing surrounding factors at reasonable effort.	21	9 (42.9%)	12 (57.1%)
C17	In agile software development it is a challenge to use methods for elicitation and evaluation of requirements in which the findings are shared with the development team.	20	8 (40.0%)	12 (60.0%)
C18	In agile software development it is a challenge to capture requirements in such a way that detailed test cases can be derived from them for quality assurance.	21	8 (38.1%)	13 (61.9%)
C19	In agile software development it is a challenge to formulate clear and comprehensible requirements in order to avoid uncertainties in the development.	22	7 (31.8%)	15 (68.2%)
C20	In agile software development it is a challenge that elicitation and evaluation of requirements are not fast enough in the project context.	17	5 (29.4%)	12 (70.6%)

5.4. Results and Discussion – Agile RE Problems

In the following sections we will answer our RQs by presenting the main results of our iterative expert judgement process. On the one hand, we present the key problems in agile RE and discuss the importance rated by the experts (RQ-5.1). On the other hand, we present options to deal with the identified key problems (RQ-5.2). Moreover, we clarify the meaning of findings and present the limitations of this study.

5.4.1. (RQ-5.1) What Are The Key Problems in Agile Requirements Engineering?

We identified six key problems industry has to face today in terms of agile RE (see Table 21). In general experts weighted the identified problems as important and none of them rated one of the six key problems as unimportant.

All problems related to the category *stakeholder and user* are classified as key problems (C2, C5, C6). Therefore, we can conclude that organizations still struggle to the agile transition. Evolving an agile mindset within a whole organization even in parts that are not close to development is still a problem companies have to address.

Typically, agile transformation starts in development-oriented parts of an organization. Transforming an organization to become more agile implies a change within the whole organization. The results show that there is a gap between knowledge and understanding agile values (Beck et al., 2001) within organizations. Development-oriented techniques evolve rapidly. In comparison, there are still problems involving stakeholders and users into the agile processes (C2, C5, C6).

Two problems (C1, C4), related to category *requirements management*, are key in agile RE. On the one hand, companies have an issue with the continuous management of requirements. On the other hand, they have a problem with technical or functional dependencies due to raising effort in coordination. Besides, one problem of methods and artifacts (C3) is a key problem.

ASD is commonly used in environments where people have to solve complex adaptive problems (Schwaber and Sutherland, 2016). Concerning C1, C3, and C4 we can state that there are still problems to be solved, due to the complexity of problems, which are not addressed by agile techniques properly. To this end, existing techniques and methods must be adapted or new techniques need to be found.

Table 21: Key problems in agile RE

ID	Key problems in agile RE	N	Yes	No
C1	In agile software development functional or technical dependencies with other teams are a challenge because a considerable coordination effort is required.	17	14 (82.4%)	3 (17.6%)
C2	In agile software development it is a challenge that stakeholders understand that the development team can make independent (detailed) decisions.	20	15 (75.0%)	5 (25.0%)
C3	In agile software development it is a challenge not to lose sight of the big picture during the implementation of complex requirements.	20	15 (75.0%)	5 (25.0%)
C4	In agile software development continuous management of requirements is a challenge since not all of them are fixed at the beginning and they may change over the course of the project.	22	16 (72.7%)	6 (27.3%)
C5	In agile software development it is a challenge to work out user requirements and quality of use in cooperation with direct users (end users) of the product.	18	13 (72.2%)	5 (27.8%)
C6	In agile software development it is a challenge to involve stakeholders throughout the whole development process in regular iterations, so that product development will succeed.	20	14 (70.0%)	6 (30.0%)

Figure 29 offers an overview of the categorized key problems.

STAKEHOLDER AND USER	<ul style="list-style-type: none"> • understanding of agile values of the stakeholders (C2) • refine requirements in collaboration with users (C5) • involve stakeholder iteratively (C6)
REQUIREMENTS MANAGEMENT	<ul style="list-style-type: none"> • technical or functional dependencies to other teams (C1) • continuous requirements management (C4)
METHODS AND ARTIFACTS	<ul style="list-style-type: none"> • staying focused on the big picture (C3)

Figure 29: Categorized key problems in agile RE

At this point, we would like to present how experts evaluate the importance of the six key problems. Therefore, the items are translated from German. The percentage of *Important* is expressing the sum of *totally important* and *important*; the percentage of *Unimportant* is showing the sum of *totally unimportant* and *unimportant*.

The number of experts (N), who assess item C1_i-C6_i in terms of the importance results from the number of experts who valued the corresponding item as a problem (C1-C6, see Table 21), minus the number of experts who choose *no statement*.

C1_i How important is the coordination, which evolves due to functional or technical dependencies with other teams?

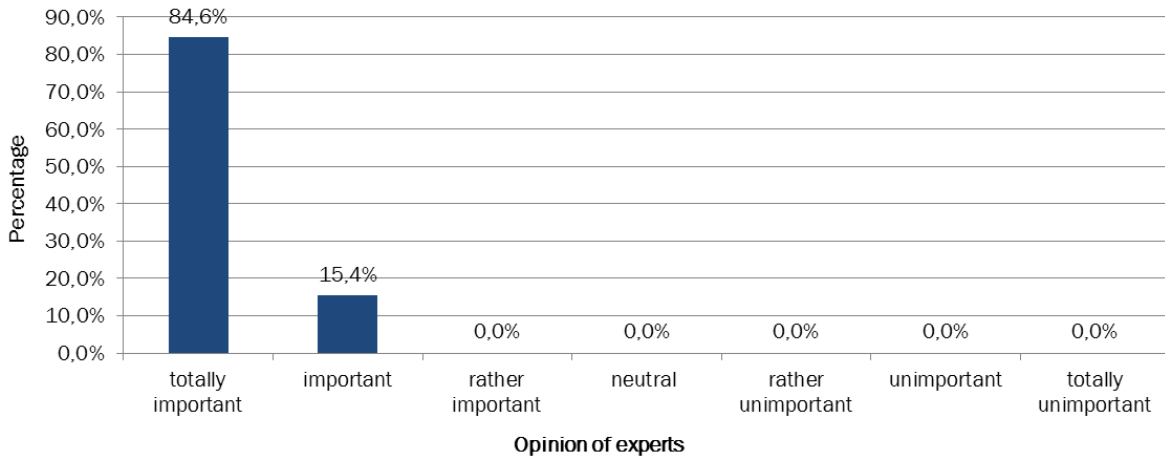


Figure 30: Importance key problem C1 (N=13)

13 experts assessed the importance of functional or technical dependencies to other teams (C1_i), although C1 is the agile RE problem most expert stated that it is a problem (see Table 21). Compared to other items (C2_i-C6_i), we can observe a very low standard deviation (0.38), since 84,6% of the experts weigh this item as *totally important* and 15.4% weigh it as *important*. In sum, the panel rates this key problem as important. No expert weighs this item as unimportant.

Table 22: Calculations of importance concerning key problem C1

ID	N	Mean	Standard deviation	Important	Unimportant
C1 _i	13	1.15	0.38	100%	0%

C2_i How important is that stakeholders understand, that the development team can make independent (detailed) decisions?

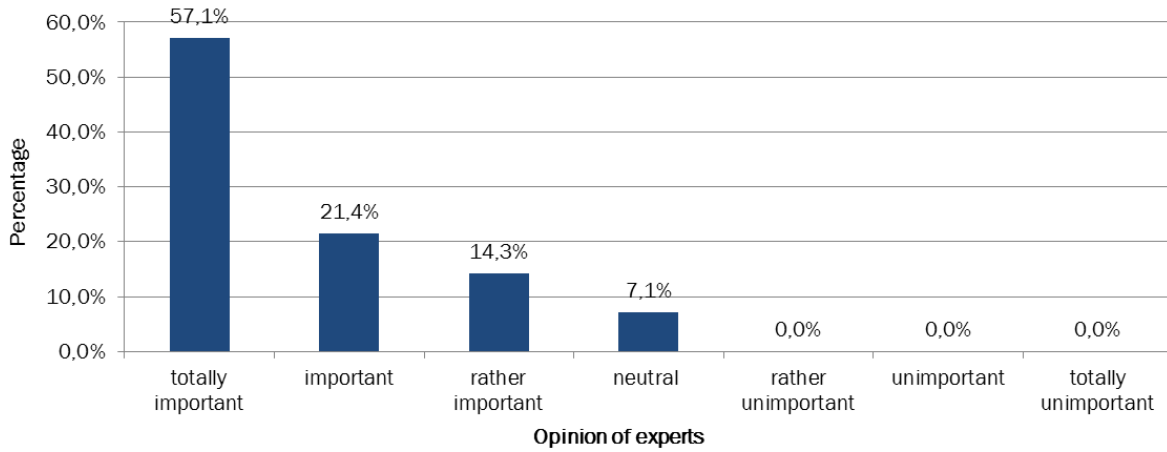


Figure 31: Importance key problem C2 (N=14)

With C2_i we queried the panel how important they perceive that stakeholders understand that the development team can make independent (detailed) decisions. This has direct impact on the understanding of agile values within an organization. 14 experts assessed this item and the answers range from *totally important* (57.1%), over *important* (21.4%), and *rather important* (14.3%), to *neutral* (7.1%). Compared to other items, this item has the highest standard deviation (0.99). In sum, the panel rates this key problem as important. No expert weighs this item as unimportant. One expert weighs this problem as neutral.

Table 23: Calculations of importance concerning key problem C2

ID	N	Mean	Standard deviation	Important	Unimportant
C2 _i	14	1.71	0.99	92.9%	0%

C3_i How important is not to lose sight of the big picture during the implementation of complex requirements?

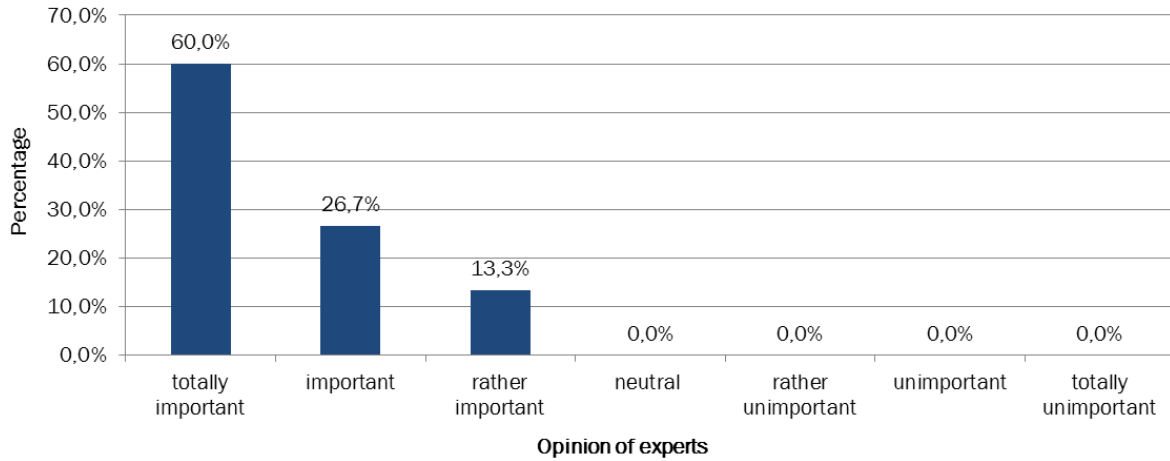


Figure 32: Importance key problem C3 (N=15)

15 experts of the panel assessed the importance of not losing sight of the big picture during the implementation of complex requirements. 60% of those experts rate C3_i as *totally important*, 26.7% weigh the item as *important*, and 13.3% as *rather important*. The standard deviation is 0.74. In sum, the panel rates this key problem as important to totally important. No expert weighs this item as unimportant.

Table 24: Calculations of importance concerning key problem C3

ID	N	Mean	Standard deviation	Important	Unimportant
C3 _i	15	1.53	0.74	100%	0%

C4_i How important is the continuous management of requirements?

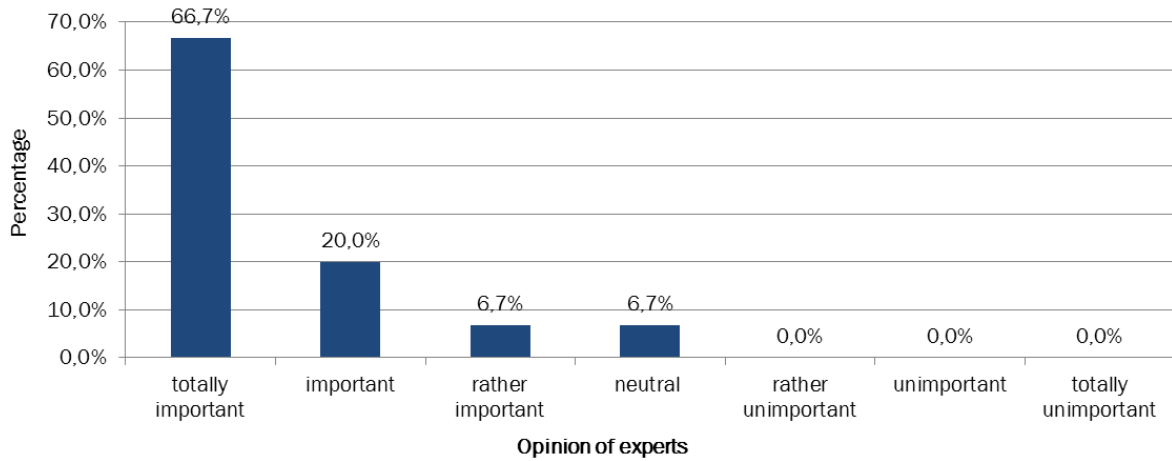


Figure 33: Importance key problem C4 (N=15)

The sample for measuring the importance of the continuous management of the requirements consists of 15 experts. The opinion of experts distributes among *totally important* (66.7%), *important* (20%), *rather important* (6.7%), and *neutral* (6.7%). Compared to other items, this one has also a relatively high standard deviation (0.92). In sum, the panel rates this key problem as important to totally important. No expert weighs this item as unimportant.

Table 25: Calculations of importance concerning key problem C4

ID	N	Mean	Standard deviation	Important	Unimportant
C4 _i	15	1.53	0.92	93.3%	0%

C5_i How important is it to work out user requirements and quality of use in cooperation with direct users (end users) of the product?

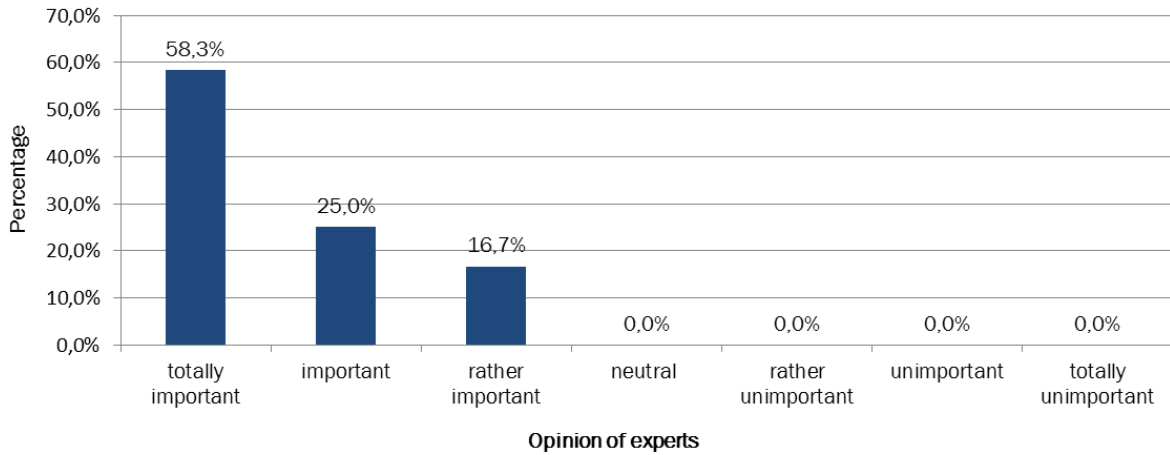


Figure 34: Importance key problem C5 (N=12)

This item (C5_i) was rated by the fewest number of experts (N=12). We measured how important is to work out user requirements and quality of use in cooperation with direct users of the product. 58.3% of the experts weigh this item as *totally important*, 25% weigh it as *important*, and 16.7% as *rather important*. The standard deviation is 0.79. In sum, the panel rates this key problem as important to totally important. No expert weighs this item as unimportant.

Table 26: Calculations of importance concerning key problem C5

ID	N	Mean	Standard deviation	Important	Unimportant
C5 _i	12	1.58	0.79	100%	0%

C6_i How important is it to involve stakeholders throughout the whole development process in regular iterations?

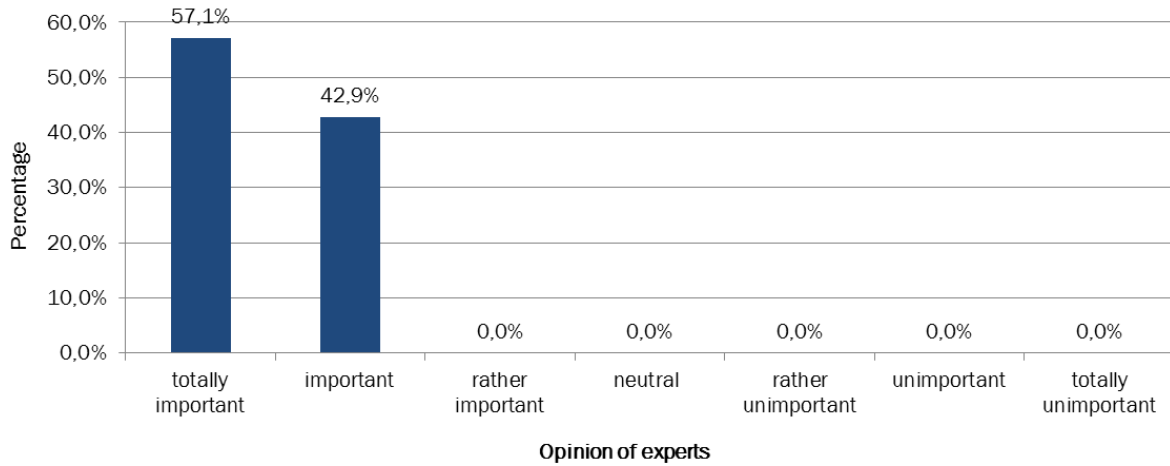


Figure 35: Importance key problem C6 (N=14)

Item C6_i queried the importance of the involvement of stakeholders throughout the whole development process in regular iterations. The opinion of the experts divides into *totally important* (57.1%) and *important* (42.9%). For item C6_i, we observe with 0.26 the lowest standard deviation. In sum, the panel rates this key problem as important to totally important. No expert weighs this item as unimportant.

Table 27: Calculations of importance concerning key problem C6

ID	N	Mean	Standard deviation	Important	Unimportant
C6 _i	14	1.43	0.26	100%	0%

5.4.2. (RQ-5.2) How Can We Deal With The Identified Key Problems?

Experts recommend techniques, methods and tools in order to deal with the problems in agile RE. Below, we will list the techniques and methods proposed by the panel for each key problem.

C1 In agile software development functional or technical dependencies with other teams are a challenge because a considerable coordination effort is required.

More than three experts recommended using scaled frameworks such as LeSS (The LeSS Company B.V., 2017), SAFe (Scaled Agile Inc., 2017) or Scrum of Scrum (Sutherland, 2001). Moreover, they proposed the use of the following techniques: creating a common understanding among all, enhancing continuous communication and collaboration, training the ability to solve dependencies, holding weekly coordination meetings, organizing teams in matrix management, building communities of practices for transcending topics, release planning (SAFe), team-transcending availability of product and sprint backlogs, involving temporary representatives in other teams, enforcing continuous integration, improving API-driven development and microservices.

C2 In agile software development it is a challenge that stakeholders understand that the development team can make independent (detailed) decisions.

The following techniques were suggested: continuous coordination and presenting possible solutions to stakeholder, providing transparency about rationales of the decisions, strengthening product owner with competency in decision making and helping stakeholders become aware of the consequences of interfering into detailed decisions.

More than three experts recommended providing alternative solutions for one requirement. In addition, it is useful to demonstrate that the recommended solution of a stakeholder is an alternative out of many. In previous rounds, more than one expert stated that product owner and stakeholder altogether decide what to be developed. In contrast, the development team decides how the requirement should be developed.

C3 In agile software development it is a challenge not to lose sight of the big picture during the implementation of complex requirements.

The following techniques were recommended: creating a shared understanding regarding the meaning of the big picture by means of a product vision, defining epics or sub goals in the beginning, managing the big picture as a responsibility of the product owner, providing transparency concerning changes among all, understanding connections among user stories by means of story mapping, visualizing customer journey in the beginning, involving users continuously in order to focus on the problem to be solved and identifying central contact person for related topics to enable rapid coordination. Moreover, the experts advised to use visualization by means of roadmaps, sketches of the system and processes, and value streams.

C4 In agile software development continuous management of requirements is a challenge since not all of them are fixed at the beginning and they may change over the course of the project.

The experts proposed the following techniques, methods and tools: collaborating closely with the requesting stakeholder, communicating regularly within the team, refining and prioritizing continuously the Product Backlog, grooming on demand (Kanban), describing in detail the requirements in the sprint backlog, reviewing the results regularly, discussing the maturity level of a requirement with the team, grouping user stories to epics, using Kano analysis, screening and scoring the theme, weighting relatively, utilizing spike stories to evaluate uncertainty in requirements and using ticketing tools (e.g. JIRA).

C5 In agile software development it is a challenge to work out user requirements and quality of use in cooperation with direct users (end users) of the product.

The experts recommended utilizing the following techniques: prototypes, interviews, observing users by the think aloud method, A/B testing, UX labs, analyzing usage behavior, friendly user tests, alpha/beta/silent launches, improving continuously a released version, utilizing a UX-board for play back user insights and testing hypotheses with real users. In addition, one expert suggested adapting user research to ASD by reducing the methods to the minimal, evaluate within the team without report creation, reducing financial restrictions for user involvement as well as problems of accessing real user by means of panels or a prior recruitment.

C6 In agile software development it is a challenge to involve stakeholders throughout the whole development process in regular iterations so that product development will succeed.

The following techniques were proposed: defining stakeholders and their involvement in regular iterations, proposing goals instead of prescribing solutions, involving all possible stakeholders in the beginning and reducing the amount of people over time.

More than eight experts suggested involving stakeholders by regular planning and review meetings to gather feedback and useful information. In light of this, they recommended clarifying the purpose of the meetings and the importance of the outcomes to be discussed beforehand.

5.4.3. Meaning of Findings

Comparing our findings to the identified problems of the related work (see Table 16), we can conclude that 16 out of our 20 problems are not reported by the related studies.

Our key problem C5 (user involvement) is reported by all related studies. In addition, three studies (Inayat et al., 2015), (Soares et al., 2015), (Ramesh et al., 2010) report issues with non-functional requirements, which is comparable to our problem C13. There is also a relation between the key problem C4 (continuous requirements management) and the problem “requirements change and its evaluation” reported by (Inayat et al., 2015). Moreover, the key problem C1 (technical or functional dependencies to other teams) is reported by (Soares et al., 2015) in a slightly different manner since they phrase it like “dependence between requirements”.

Moreover, the results show that the identified problems are often not limited to ASD, but they rather refer to software development in general. Therefore, we can conclude that organizations still struggle with agile transition and understanding agile values, in particular, in terms of stakeholder and user involvement.

The identified agile RE problems allow agile practitioners as well as researchers to improve their agile environment since the knowledge supports them to understand and to reflect their problems. The high level description of the key problems can be enriched with contextual information, so that they are adapted to the specific environments. For instance agile practitioners can ask themselves during a retrospective whether they struggle with one of the identified key problems and think about the variation of the problem.

5.4.4. Limitations

We are aware that the design of a questionnaire is important for the process of data gathering. To this end, we made several pretests of each questionnaire we used with participants matching our criteria of expert selection. Nevertheless, we observed two experts struggling with the UX of the questionnaire tool (Google Forms) used in round 1. Therefore, we decided to use another tool (LimeSurvey) for the questionnaire in round 3, which was more complex than the previous two.

To carry out the study, the group of authors created summaries of the results and made decisions concerning the kind of items they had to query in the following rounds. That may lead to bias in the opinion building process of the panel. We tried to prevent this point by being very accurate in terms of data analysis and by creating the reports. In addition, we selected items for the following rounds through the selection criteria defined earlier.

Our panel consists of experts from Germany and Switzerland. Therefore, one could argue that there is a geographical limitation of the applicability of the results. In terms of this, we have to say that most of the experts are working in international environments. Nevertheless, we decided not to tackle this aspect within the scope of this study in order to keep the complexity of this study as low as possible. We started to set up new questionnaires with the aim to assess the identified agile RE problems (see Table 20) by means of an international sample.

It could be argued that there are some limitations in terms of the general application of the results due to the usage of an expert validation. However, the experts work for different companies and therefore they have different levels of experience in the agile transformation of an organization. Additionally, the identified key problems (Table 21) only give guidance for the optimization of existing process models for agile RE in industry and do not claim to be fully complete.

5.5. Chapter Summary

This chapter has addressed the identification of the most important problems in agile RE industry has to face up today. Moreover, we examined how to deal with those problems. For that purpose, we carried out an iterative expert judgement process comprising three complementary rounds. During these rounds we find the core of a specific problem by abstracting contextual information of the problems. The learnings from previous iterations were used for carrying out the following ones. Our panel consisted of 26 experts in the field of ASD working for 19 different companies.

We have identified 20 problems industry has to address at present in terms of agile RE. Six of these problems have been defined as key problems:

- technical or functional dependencies to other teams (C1)
- understanding of agile values of the stakeholders (C2)
- staying focused on the big picture (C3)
- continuous requirements management (C4)
- refine requirements in collaboration with users (C5)
- involve stakeholder iteratively (C6)

In addition, we have analyzed options to deal with those key problems by means of agile techniques recommended by the panel. The results show that the identified challenges are often not limited to ASD, but they rather refer to software development in general. Therefore, we can conclude that organizations still struggle with agile transition and understanding agile values, in particular, in terms of stakeholder and user involvement.

The agile RE problems and agile techniques allow us to create agile RE patterns. The concept of agile RE patterns will be introduced in the following chapter.

In the former chapter we have started to provide techniques for applying our conceptual approach to the enterprise environment. To this end, we identified agile RE problems in industry by means of empirical research (presented in chapter 5). Moreover, we identified agile techniques that allow us to handle these problems. Combined with the introduction of agile RE patterns within this chapter, we provide a full framework for agile RE that enables us to apply our conceptual approach to the enterprise environment.

This chapter aims to introduce the concept of agile RE patterns (Schön et al., 2017b). Moreover, it shows how agile RE patterns can be created by means of a systematic pattern mining process. The aim of agile RE patterns is to share the knowledge about best practices in terms of agile RE. Agile RE patterns allow agile practitioners as well as researchers to model and to improve their agile RE process models by enabling inspection and adaptation.

In ASD, the continuous improvement happens on two dimensions. On the one hand, the product quality is improved by agile techniques like continuous integration (Humble and Farley, 2010) or iterative review meetings (Schwaber, 2004). On the other hand, the quality of the environment is continuously improving in terms of collaboration, processes and tools by using mechanisms like retrospectives (Schwaber, 2004) or kaizen (Anderson, 2010). Agile RE Patterns support the improvement of the latter dimension of quality.

This chapter is structured as follows: section 6.1 gives a brief overview of patterns and related work in this field. Then, section 6.2 discusses on the gaps in related work. Section 6.3 presents our research method and the objectives. Finally, section 6.4 discusses the results by means of an overview of identified agile RE patterns and a presentation of the web application agileRE.org.

6.1. Summary of Related Work

The concept of pattern relies on the work by Alexander et al. (Alexander et al., 1977). They created a pattern language concerning the architecture of towns and buildings. They define a pattern as follows: *“Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.”* (Alexander et al., 1977).

A pattern is a reusable solution to a commonly recurring problem. The pattern does not state a concrete solution it rather provides a description of best practices of how to solve a problem. Therefore, the solution is described on a high level and need to be adapted to the environment where the pattern is applied.

One of the earliest adoptions of the concept of patterns in terms of software engineering can be found in the community of object-oriented software designers (Sowizral, 1987). Nowadays, patterns are common practice in different domains like user interface design, programming languages or cyber security (Henninger and Corrêa, 2007).

In literature, there can be found some related works that present patterns focused on the optimization of the product development process in agile environments.

Bertholdo et al. (Bertholdo et al., 2014) created agile usability patterns that aim to share best practices concerning the integration of ASD and UCD. They conducted a literature review in order to

find usability practices used by the agile community. On the one hand, they pose patterns for the UCD early stages:

- Identify need for human-centered design: *sprint zero, one sprint ahead, UX specialists as Product Owners, users time is valuable, parallel tracks, UX specialists as full-time member of the agile team*
- Specify context of use: *little design upfront, contact plan for users*
- Specify requirements: *user stories, more collaboration less documents, prototypes as specification*

On the other hand, Bertholdo et al. (Bertholdo et al., 2016) identify agile usability patterns for UCD final stages:

- Create Design Solutions: *low fidelity prototyping, high fidelity prototyping, design studio, collaborative and participative design*
- Evaluate designs: *tests with users, evaluation by inspection, RITE method, acceptance tests*

Salah et al. (Salah et al., 2015) also propose patterns for integrating ASD and UCD. For the identification of their patterns they conducted a SLR that investigated challenges in terms of the integration of ASD and UCD. Moreover, they carried out interviews to study challenges and practices applied in this context. Salah et al. recommend the following patterns: *less is more, usability testing sessions alongside agile development tests, do it RITE, developers as UCD practitioners, websites as lightweight documentation*

6.2. Gap Analysis of Related Work

Analyzing the related work, we can state that Bertholdo et al. and Salah et al. both identified very relevant patterns for the integration of ASD and UCD. These patterns are related to the agile RE patterns, presented in this PhD thesis. Nevertheless, their patterns only focus on the integration of ASD and UCD and investigate a very specific problem domain.

The proposed agile RE patterns coping a broader field, since they are dealing with problems related to agile RE. The agile RE problems (see chapter 5) are related to the problems stated by Bertholdo et al. and Salah et al. but go beyond their work.

We can utilize the related patterns for our pattern mining process (see section 6.3.2) in order to receive a full coverage of our problem domain. In addition, agile usability patterns can support agile RE patterns in terms of being related patterns.

6.3. Research Method

We applied a systematic pattern mining process in order to create agile RE patterns. Figure 36 displays the steps, which were carried out in order to create our agile RE patterns. In step 1, we evaluated the results of our Delphi study (see chapter 5) and reduced the contextual information given by the panel, in order to find the core of the problems in agile RE. In a second step, we enrich the findings from this study with further evidence from related work (see chapter 2). This allows us to create new results in step 3 by means of agile RE patterns.

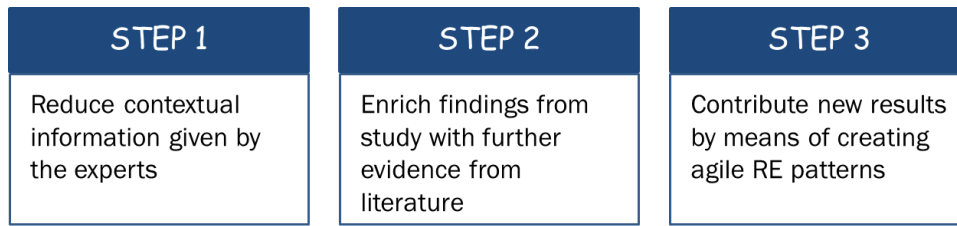


Figure 36: Steps for creation of agile RE patterns

This systematic pattern mining process comprised three phases. In the following, we will present our research objectives and the research method that was applied in order to answer the proposed research questions.

6.3.1. Objectives and Research Questions

This chapter aims to introduce the concept of agile RE patterns. Therefore, we will describe our pattern mining process based on empirical research in literature and industry. We will discuss our results and provide two examples of agile RE patterns. In sum, the pattern mining process identifies 41 agile RE patterns. The accumulated knowledge will be shared by means of a web application.

We address these objectives by the following research questions:

- RQ-6.1: How can we create agile RE pattern?
- RQ-6.2: Which agile RE patterns do exist?
- RQ-6.3: How can we share the knowledge concerning agile RE?

6.3.2. Pattern Mining Process

We carried out a systematic pattern mining process comprising three phases (see Fig. 2) for identifying the agile RE patterns. In the following subsections, we will discuss each of the three phases and the utilized research method.

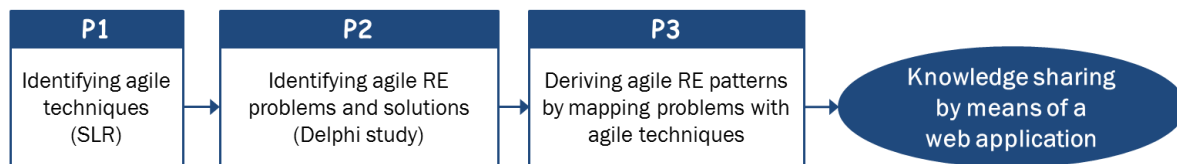


Figure 37: Pattern mining process and knowledge sharing

6.3.2.1. Phase 1: Identifying Agile Techniques

As presented previously, we conducted a SLR with the aim to capture the current state of the art related to agile RE with focus on stakeholder and user involvement (see chapter 2). A SLR is a means of applying evidence-based research in various domains. Therefore, all available research relevant to particular research questions or a specific topic area is evaluated and interpreted. A SLR can be classified as secondary study (Kitchenham and Charters, 2007).

With regard to the results of the SLR (see section 2.3.4), we identified agile techniques for stakeholder and user involvement, data gathering, integrating HCD and ASD, building shared

understanding, requirements management, documentation of requirements or non-functional requirements (NFR).

In particular, we identified 20 artifacts, which were used by more than two of the included papers (see Table 9). The artifacts are: user story, prototype, use case, scenario, story card, persona, vision, UML diagram, storyboard, task, Kanban board, UI pattern, essential use case, picture, video, mind map and UI specification. We found that HCD, Design Thinking, Contextual Inquiry and Participatory Design are commonly used methodologies that are useful in order to make ASD more human-centric.

The agile techniques can be classified as methods (e.g. mind mapping or participatory design), artifacts (e.g. user story, prototype) or roles (e.g. Agile-UCD specialist or NRF stakeholder). We can point out that the included studies rarely mentioned classical agile artifacts (e.g. Product Backlog or Sprint Goal) or meetings (e.g. daily standup or retrospective).

6.3.2.2. Phase 2: Identifying Agile RE Problems and Solutions

Subsequent to the identification of agile techniques, we carried out an empirical evaluation with the aim to identify the most important problems in agile RE that the industry has to face up today (see chapter 5). For that purpose, we conducted an iterative expert judgement process rooted in a Delphi study performed in three complementary rounds.

Once the last round was completed, we identified in sum 20 problems where six out of them are defined as key problems of agile RE, as Table 21 shows. Based on the results of our study, we have provided solutions for dealing with those key problems by means of agile techniques and tools recommended by the panel of experts (see section 5.4.2).

6.3.2.3. Phase 3: Deriving Agile RE Patterns

Following the approach by Wellhausen and Fießler (Wellhausen and Fiesser, 2011), we distinguish between problem domain and solution domain. In this chapter, the problem domain is explored by an empirical study (Schön et al., 2017d) where we identified six key problems for agile RE (Table 21). As part of this study, we also analyze the solution domain and we identify agile techniques that can be applied in order to solve the identified problems. Together with the agile techniques known from our SLR, we can derive agile RE patterns. Figure 38 shows the relation between agile RE problems and agile techniques. One agile RE problem can be solved by one or more agile techniques. On the contrary, one agile technique can solve one or more agile RE problems.

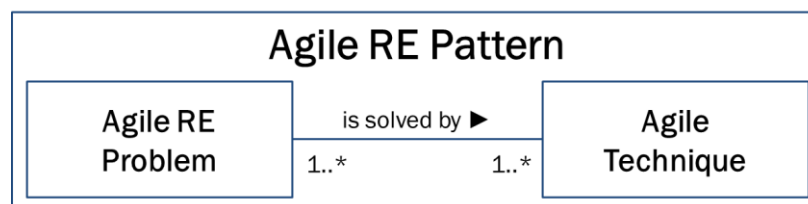


Figure 38: Relation between agile RE problem and agile technique

Figure 39 shows an overview of the different types of agile techniques.

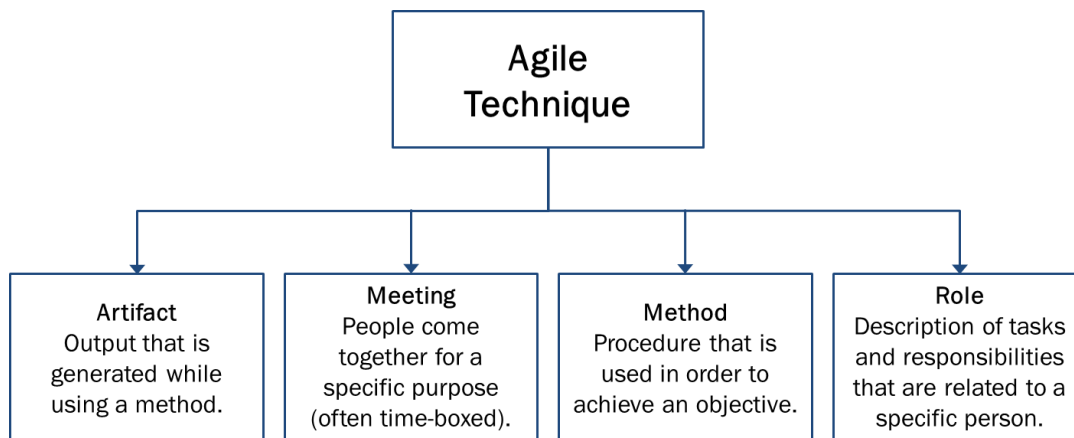


Figure 39: Overview of types of agile techniques

The mapping process between problem domain and solution domain is iterative and still ongoing. It comprised three steps:

- Experts carried out the first mapping between problem domain and solution domain during the empirical study (see section 5.4.2).
- Moreover, we mapped the identified agile techniques known from the SLR to the agile RE problems.
- Afterwards, we made a cross check with agile techniques proposed by agile methodologies (e.g. Scrum, XP, Kanban) and related patterns (see section 6.1).

As mentioned before, the experts participating in our Delphi study (see chapter 5) recommended solutions to cope with agile RE problems. One of these solutions is description of tasks that are carried out by specific roles proposed by agile methodologies. We mapped such tasks to the responsibilities of the roles in order to derive the agile RE patterns. For instance, the experts recommended the solution *continuous coordination and presenting possible solutions to stakeholder*. We accumulated this solution to the pattern *Product Owner* since s/he is responsible for engaging stakeholders. Another recommended solution was *strengthening product owner with competency in decision making*. We added this task to the pattern *Agile Coach/Scrum Master* since s/he is in charge of helping people understand the role of Product Owner.

6.4. Results and Discussion – Agile RE Patterns

Our first research question *RQ-6.1: How can we create agile RE pattern?* is already answered by the description of our pattern mining process in section 6.3.2.

In the following we will answer the second research question *RQ-6.2: Which agile RE patterns do exist?* by presenting an overview of the identified agile RE patterns and showing some examples (*Evaluation and Testing, Story Map, Product Owner and Review Meeting*).

The third research question *RQ-6.3: How can we share the knowledge concerning agile RE?* will be answered by offering an overview of the web application agileRE.org (Schön et al., 2017c).

6.4.1. Overview Agile RE Patterns

Table 28 shows the results of the mapping process described in section 6.3.2. The agile RE patterns are classified according to their agile technique referred to as pattern type (artifacts, meetings, methods or roles). There are some patterns that can either be classified as methods or artifacts. Thus, we decided to classify them in terms of the resulting artifact, which is created while using the method. For instance, story mapping would be classified as method, whereas the result (story map) is classified as artifact.

In sum, we discovered 41 agile RE patterns: 14 patterns classified as artifacts, 5 patterns classified as meetings, 17 patterns classified as methods and 5 patterns classified as roles.

In addition, the table shows what agile RE problem (Table 21) is solved by which agile RE pattern:

- C1: Technical or functional dependencies to other teams
- C2: Understanding of agile values of the stakeholders
- C3: Staying focused on the big picture
- C4: Continuous requirements management
- C5: Refine requirements in collaboration with users
- C6: Involve stakeholder iteratively

Table 28: Agile RE patterns matched to agile RE problems

Agile RE pattern name	Pattern type	C1	C2	C3	C4	C5	C6
Minimum Viable Product (MVP)	Artifacts		x	x		x	
Kanban board	Artifacts	x		x	x	x	
Prototypes	Artifacts	x		x		x	x
Definition of ready and definition of done	Artifacts				x		
User stories	Artifacts			x	x		
Product Backlog	Artifacts	x		x	x		
Roadmap	Artifacts	x		x	x		
System models	Artifacts			x			
Story map	Artifacts			x	x		
Process models	Artifacts			x			
Value stream	Artifacts			x			
Customer journey map	Artifacts			x	x		
Product vision	Artifacts			x	x		
Impact map	Artifacts			x	x		x
Refinement meeting	Meetings		x	x	x		x
Planning meeting	Meetings	x		x			x
Review meeting	Meetings		x	x	x	x	x
Daily standup meeting	Meetings	x			x		
Retrospective	Meetings	x		x			
Evaluation and testing	Methods			x	x	x	

Launch of product features	Methods					x	x
Lean user research	Methods				x	x	
Users time is valuable	Methods					x	
Co-design	Methods				x	x	x
Define agile RE process model	Methods			x	x		x
Transparency of decisions	Methods		x		x		x
Weighing up various solution proposals	Methods		x		x		
Coaching	Methods		x		x		
Lifecycle managements by means of tools	Methods	x			x		
Shared understanding	Methods	x	x	x	x		
Scaling agile	Methods	x					
Continuous integration	Methods	x					
API-driven Development	Methods	x					
Micro services	Methods	x					
Community of practices	Methods	x					
Pairing	Methods	x					
Product Owner	Roles		x	x	x		x
Agile Coach/Scrum Master	Roles	x	x	x			
Development team	Roles	x	x	x			
Expert	Roles	x	x	x	x	x	
Stakeholder	Roles		x				x

An agile RE pattern is composed of dynamic and static parts (see Figure 40). On the one hand, the agile technique *Scaling Agile* is an activity and therefore represents the dynamic part. On the other hand, the agile technique *user story* is an artifact and represents the static part of an agile RE pattern.

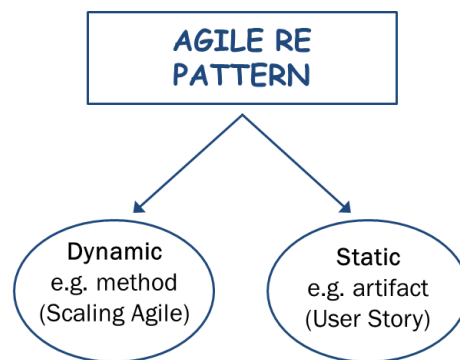


Figure 40: Agile RE patterns are composed of dynamic and static parts

Table 29 shows the classification of agile RE patterns according to their dynamic or static parts. Agile RE patterns, which agile technique is an activity are classified as dynamic agile RE pattern, whereas patterns with a static agile technique are classified as static agile RE patterns. Roles do not fit into this classification in a classical sense, since they could be handled as actors in

UML. However, in terms of classification of agile RE patterns we classify them as static due to the descriptive nature of a role description.

Table 29: Classification of Agile RE patterns in dynamic and static

Pattern type	Dynamic agile RE patterns	Static agile RE patterns
Artifacts		Minimum Viable Product (MVP), Kanban board, Prototypes, Definition of ready and definition of done, User stories, Product Backlog, Roadmap, System models, Story map, Process models, Value stream, Customer journey map, Product vision, Impact map
Meetings	Refinement meeting, Planning meeting, Review meeting, Daily standup meeting, Retrospective	
Methods	Evaluation and testing, Launch of product features, Lean user research, Users time is valuable, Co-design, Define agile RE process model, Transparency of decisions, Weighing up various solution proposals, Coaching, Lifecycle managements by means of tools, Shared understanding, Scaling agile, Continuous integration, API-driven Development, Micro services, Community of practices, Pairing	
Roles		Product Owner, Agile Coach/Scrum Master, Development team, Expert, Stakeholder

6.4.2. Examples of Agile RE Patterns

Each agile methodology (e.g. Scrum, Kanban, XP) comes with its own requirements that have impact on how RE is carried out (see Figure 17, metaclass Impact). For instance, there is a difference between flow-driven approaches like Kanban or time-boxed approaches like Scrum. To this end, agile RE problems are combined with different agile techniques, resulting in a set of related agile RE pattern.

Below, we will present four examples of Agile RE patterns:

- a) *Evaluation and Testing*
- b) *Story Map*
- c) *Product Owner*
- d) *Review Meeting*

As noticed, we used appropriate guidelines for pattern writing provided by (Wellhausen and Fiesser, 2011) and (Kohls, 2012). We created our own pattern template based on existing templates. The elements of our pattern template are the attributes of the metaclass AgileREPattern (see Figure 41), which is defined by the agile RE metamodel (see Figure 17).

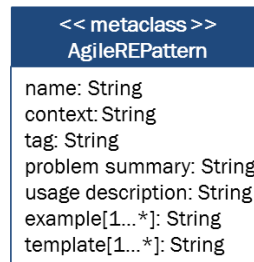


Figure 41: Metaclass AgileREPattern

The *name* is a unique identifier for the agile RE pattern, whereas the *context* set the stage, where the pattern can be applied. The *tag* is used for classification of the agile RE pattern. The *problem summary* combines a description of the problem and forces, which explain why the problem is difficult to solve. Moreover, the *usage description* entails the solution and the consequences that would follow, if the pattern is applied. The section *example* provide an excerpt of know uses and the section *template* provides templates, which are used in industry as best practices for applying the agile technique.

The agile RE pattern *Evaluation and Testing* is classified as method, *Story Map* is classified as artifact, *Product Owner* is classified as role and *Review Meeting* is a meeting.

We applied the agile RE pattern *Evaluation and Testing* in one of our case studies related to the integration of HCD and Kanban (see section 7.3.1 and (Schön et al., 2016b)). The project was carried out in a medium-sized IT company, located in Germany, specialized in e-commerce, mobile apps and SAAS tools. We aim to relaunch an internet-based newspaper portal in a period of six months along 2013/2014.

Pattern name	Evaluation and testing
Context	Working in a Kanban system make people focus on small tasks. This can cause the problem of leaving out sight of the big picture during the implementation of complex requirements. Hence, it is hard to design a positive User Experience (UX) for the user.
Tag	Discovery, refinement, review, methods
Problem summary	<p>In ASD it is a challenge not to lose sight of the big picture during the implementation of complex requirements (see C3, Table 21).</p> <ul style="list-style-type: none"> • providing a positive UX to the user. • carrying out a release evaluation continuously. • not interrupting the workflow due to scheduling testing activities and organization. • reducing costs for long-term UX testing.
Usage description	Carrying out a regular release evaluation (Schön et al., 2016b) by means of Usability and UX testing (Hartson and Pyla, 2012), (Schrepp et al., 2014). Therefore, a work in progress (WIP) limit to the last column (“Done”, see Figure 42) of the Kanban board should be introduced. The release evaluation should start, once the WIP limit is reached. The regular release evaluation helps you stay focused on the big picture. In addition, it enables improving the overall UX of the product. Introducing a WIP limit to the last column of the Kanban board reduces the complexity of finding the starting point for a release evaluation. Moreover, it allows carrying out UX testing continuously.

Examples

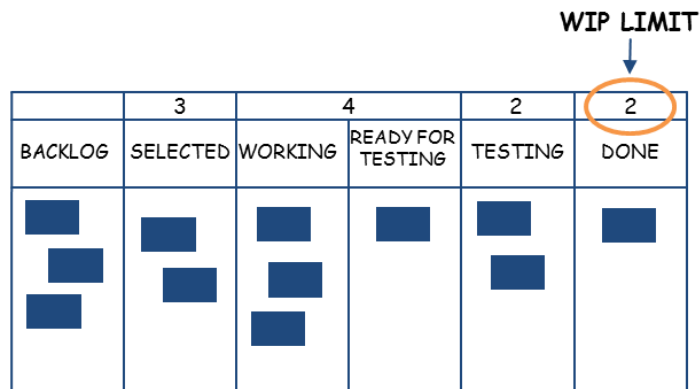


Figure 42: Example of a Kanban board

The agile RE pattern *evaluation and testing* is related to the pattern *Sprint Zero* by Bertholdo et al. (Bertholdo et al., 2014), since both patterns address the same problem in different stages. *Sprint Zero* is utilized as a stage before starting the project implementation, whereas *evaluation and testing* is a recurring task during the development of a product. Besides, the solution of the pattern *evaluation and testing* is related to the pattern *Usability Testing Sessions Alongside Agile Development Tests* by Salah et al. (Salah et al., 2015) since it may also solve the problem of scheduling Usability and UX testing in an agile environment.

Template See Figure 42

Pattern name	Story map
Context	User stories let people focus on developing small increments. This leads to the problem of losing sight of the big picture. In addition, prioritizing user stories is difficult due to the open question of what the user really needs.
Tag	Discovery, refinement, prioritization, documentation, artifacts
Problem summary	<p>In ASD it is a challenge not to lose sight of the big picture during the implementation of complex requirements (see C3, Table 21).</p> <ul style="list-style-type: none"> • it is hard to find out what the user needs and what product to build. • agile teams and stakeholder struggle with prioritizing requirements for different releases. • it is tough to identify the scope of the MVP, so it is uncertain when a first version of a product should be released.
Usage description	<p>Managing user stories by means of a story map. Story Mapping (Patton, 2014) is an agile technique that can be used to manage user stories. A story map tells the story about the product and its usage from a user’s perspective. Dependencies among user stories can be identified easily and prioritizing requirements becomes simple.</p> <p>Story Mapping keep people focused on users and their experiences with the product. In addition, the conversation among agile team members will be more vivid as well as effective due to the storytelling approach. Continuous management of requirements will be a natural activity because of the visual presentation (see Figure 43) of the prioritized user stories.</p> <p>The agile technique Story mapping is introduced by Patton (Patton, 2004), (Patton, 2014). In 2004, Patton (Patton, 2004) presented the idea but he did not call the technique Story Mapping at that time. In the following years, he noticed that other people applied a similar approach to solve problems concerning prioritization of requirements and losing sight of the big picture. Then, Patton (Patton, 2014) started realizing that he discovered a pattern.</p>

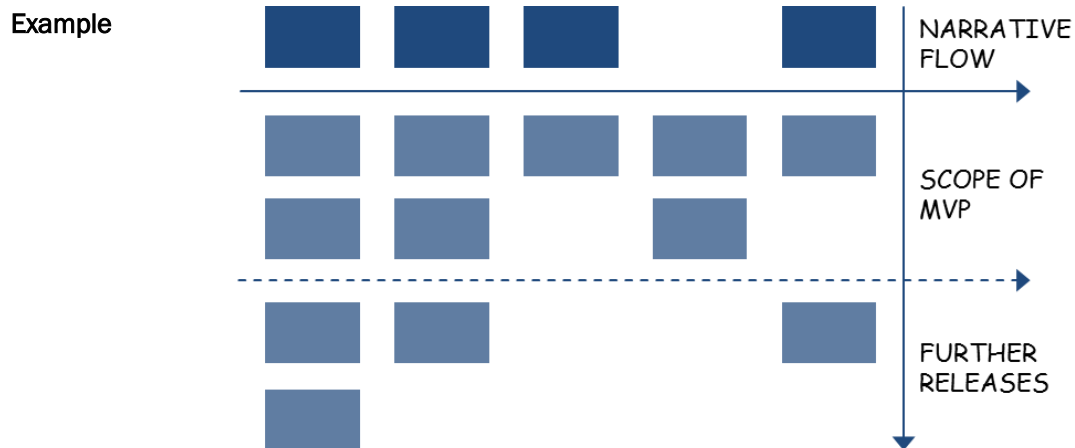


Figure 43: Example of a story map

Template See Figure 43

Pattern name	Product Owner
Context	Organizations want to maximize the return on invest (ROI) in order to achieve a continuous grow. With regard to product development, stakeholders have different desires in terms of product features or look and feel. These different desires need to be balanced in order to increase the value of a product.
Tag	Discovery, refinement, prioritization, review, roles
Problem summary	<p>In ASD it is a challenge to involve stakeholders throughout the whole development process in regular iterations, so that product development will succeed (see C6, Table 21).</p> <ul style="list-style-type: none"> • managing different interests of stakeholders is a challenge. • objectives of development and objectives of organization are sometimes disconnected. • receiving feedback is necessary for a good expectation management.
Usage description	<p>Engaging stakeholder into product development iteratively by means of a Product Owner (Schwaber and Sutherland, 2016). The Product Owner is responsible for maximizing the value of the product. Therefore, he manages the Product Backlog by prioritizing requirements according to needs of the users and customers as well as regulatory and market requirements. He collaborates with the development team, stakeholders, users and management.</p> <p>Having an empowered Product Owner in places, facilitates the decision making process in terms of strategic product management. In addition, there is one contact person, who is responsible for managing the different expectations and desires.</p> <p>The agile technique Product Owner has its origin in Scrum (Schwaber, 2004). The idea behind the role is to have one single person, who is in charge of maximizing the value of the product and to be responsible for the overall quality of the product. Compared to the role of an Agile Coach or Scrum Master, who is the person in charge in terms of process quality.</p>
Example	n.a.
Template	n.a.

Pattern name	Review Meeting
Context	Receiving feedback from stakeholders and users in an iterative manner is very valuable for product development to succeed.
Tag	Discovery, refinement, prioritization, review, meetings
Problem summary	<p>In ASD it is a challenge to work out user requirements and quality of use in cooperation with direct users (end users) of the product (see C5, Table 21).</p> <ul style="list-style-type: none"> • receiving feedback from direct users of the product. • adapt direction of product development according to users´ needs. • refine requirements according to evolving needs and wishes of users.
Usage description	<p>Holding regular Review Meetings supports receiving feedback from direct users of the product. Review Meetings (Schwaber and Sutherland, 2016) aim to inspect the product and to adapt the direction of product development to new learnings and insights. Attendees are affected people like developments team, stakeholders, users and management who collaborate in order to direct future product development.</p> <p>The Review Meeting can be organized like an exhibition, in order to receive feedback from a large group (Kraus, 2013). Attendees can try out new product features and receive additional information regarding the development progress. For this, every development team presents their objectives on a booth, where user stories with additional material like graphics, flow charts or prototypes are detailed (see example in Figure 44).</p> <p>Regular Review Meetings improve the feedback culture of an organization. The direction of product development can be adapted according to users´ and customers´ needs. Review Meetings provide space to involve users´ feedback into product development and requirements can be refined in cooperation with direct users. The concept of Review Meetings has its origin in Scrum, where the Sprint Review Meeting (Schwaber, 2004) is a necessary feature of the Scrum framework. The intention is to showcase the work of the previous Sprint and to inspect the delivered product increment as well as to adapt Product Backlog according to learnings and feedback.</p>

Example

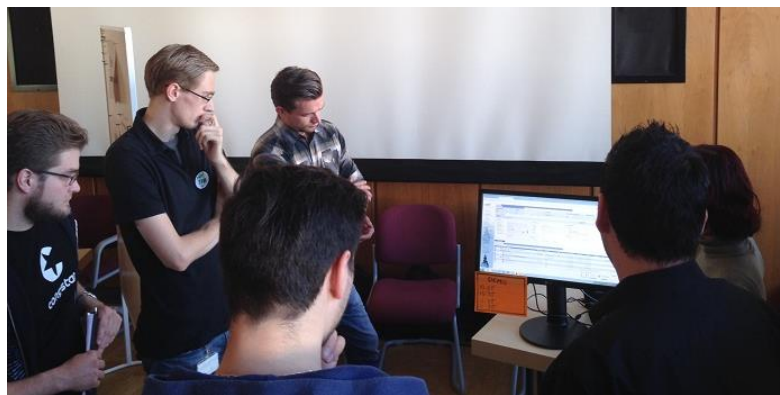


Figure 44: Review Meeting in form of an exhibition (Glase, 2016)

Template n.a.

6.4.3. Knowledge Sharing by Means of a Web Application

The knowledge concerning agile RE presented by agile RE patterns will be distributed by means of a web application. Typically, agile practitioners do not have time to read full books due to their daily business. The knowledge needs to be presented in chunks in order to be communicated in an effective manner. To this end, we decided to share the knowledge regarding agile RE by means of agile RE patterns. The written agile RE patterns will be distributed by means of a web application (agileRE.org) (Schön et al., 2017c). The aim of agileRE.org is to support practitioners and researchers improving their agile RE process models as well as to support them in solving the key problems in agile RE by means of tool support.

Figure 45 presents the landing page of agileRE.org. On the one hand, the user can browse through agile RE problems. On the other hand, the user can browse through agile RE patterns that are classified by their pattern type (artifacts, meetings, methods or roles, see Table 28).

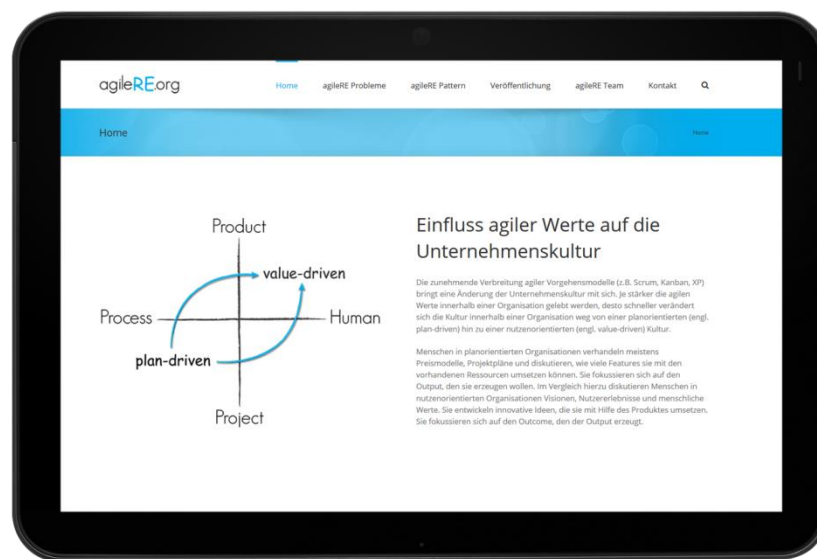


Figure 45: Landing page agileRE.org

The web application is providing tool support for our framework by means of visualizing parts of the agile RE metamodel (see chapter 4) in order to provide techniques to apply the conceptual approach to the enterprise environment. To this end, it bridges the way to playback knowledge from theory to practice.

We developed the web application in collaboration with a bachelor student. First, we had several workshops where we discussed the vision and build a shared understanding concerning the objectives of agileRE.org. Then, we analyzed existing patterns and pattern templates in order to get to know best practices in the field. The results were used as input for our collaborative prototyping workshops. Therefore, we used the prototyping tool balsamiq⁶ for creating wireframes iteratively.

Moreover, we created a template for agile RE patterns in order to present them in the same fashion. This has a positive effect on the readability from a user's perspective. The analysis of existing pattern templates on different websites allows us to understand what expectations users

⁶ <https://balsamiq.com>

have and give guidance for designing the pattern template. Figure 46 shows how our agile RE patterns are presented on agileRE.org by means of the example *Story Map*. Compared to the presentation of the pattern *Story Map* in section 6.4.2, we enriched the patterns with additional information (e.g. section for downloads, relation to other patterns, agile RE activity). This enrichment of metadata is useful in terms of linking patterns with each other on a website.

agileRE.org Home agileRE Probleme agileRE Pattern Veröffentlichung agileRE Team Kontakt Q

Story Map Home / Artefacts, Blick auf das große Ganze, Kontinuierliche Pflege der Anforderungen, / Story Map

Story Map

Beim Einsatz von User Stories fokussieren sich die Produkthersteller auf die Lieferung von kleinen Inkrementen. Dies führt dazu, dass schnell der Blick für das große Ganze verloren geht. Zudem stellt sich die Priorisierung von User Stories als schwierige Aufgabe dar, da häufiger nicht geklärt ist, was der Nutzer wirklich braucht.

Problembeschreibung

In der agilen Softwareentwicklung ist es eine Herausforderung bei der Umsetzung komplexer Anforderungen den Blick auf das große Ganze nicht zu vernachlässigen.

Wie wird das Pattern angewendet?

Story Maps können verwendet werden, um User Stories zu organisieren [1]. Mit einer Story Map können User Stories in eine sinnvolle Reihenfolge gebracht werden, sodass sie den Workflow oder Business-Prozess des Nutzers widerspiegeln und somit die Geschichte aus Nutzersicht erzählen. Mit Hilfe einer Story Map können Anforderungen priorisiert werden. Darüber hinaus können Abhängigkeiten zwischen User Stories aufgrund der visuellen Darstellung identifiziert werden.

Beispiele

Diagram illustrating a Story Map structure:

- Activities (Aktivität) are shown in blue boxes at the top.
- Tasks (Aufgabe) are shown in green boxes below activities.
- Sub-tasks (Detail/Teilaufgabe) are shown in yellow boxes below tasks.
- The map is organized into a grid with a dashed line separating the top section (Umfang des Releases für das MVP) from the bottom section (Umfang weiterer Releases).
- Labels on the right indicate 'narrativer Handlungsfluss' and 'Verdichten Details'.

Downloads

Steht in Verbindung zu

User Story

Kommt zum Einsatz bei den folgenden Aktivitäten:

Discovery	Refinement	Prioritization	Review	Documentation
X	X	X	X	X

Löst welches Problem?

Blick auf das große Ganze
Kontinuierliche Pflege der Anforderungen

Quellen

[1] J. Patton, *User Story Mapping: Discover the Whole Story, Build the Right Product*, First edit. O' Reilly, 2014.

Figure 46: Example of agile RE pattern (Story Map) presented on agileRE.org

The visual design (e.g. colours, typo, icons, and logo) is adapted to requirements of the agile community. We decided to implement the web application on the basis of WordPress⁷ since this guarantees a sustainable solution in terms of maintainability and extensibility. We have applied several methods in order to improve the web application continuously (e.g. UX testing or web analytic tools). In addition, our aim is to improve the agile RE patterns continuously. The content is currently written in German, translations will follow.

6.5. Chapter Summary

This chapter presents the concept of agile RE patterns. For this purpose, we carried out a pattern mining process comprising three phases by means of empirical research in literature and industry. In the first phase, we identified agile techniques by means of a SLR. Then, we conducted an iterative expert judgement process with 26 experts in the field of ASD to identify the most important problems in agile RE. After that, we derived in sum 41 agile RE patterns. Therefore, we mapped the agile techniques to the agile RE problems.

We showed an overview of the 41 identified agile RE patterns and presented examples of agile RE patterns: a) *Evaluation and Testing*, b) *Story Map*, c) *Product Owner* and d) *Review Meeting*.

Moreover, we presented the web application agileRE.org in order to show how the knowledge concerning agile RE will be shared among the agile community.

We can conclude that our agile RE patterns are highly relevant for the industry as well as the research community, since we gathered the data from experts in the field of ASD. Agile RE patterns enable practitioners and researchers to implement their agile RE process models and will therefore cause strong impact. With agile RE problems and agile RE patterns we provide techniques to apply our conceptual approach to the enterprise environment.

⁷ <https://en.wordpress.com>

In the previous chapters, we introduced the components of the framework for modeling agile RE. In chapter 4, we have introduced the agile RE metamodel, which describes the influencing parameters of agile RE on an abstract level. Then, in chapter 5 we presented the catalogue of agile RE problems. Subsequent, a catalogue of corresponding agile RE patterns are contributed in chapter 6.

In the following chapter, we want to demonstrate the application of the framework in industry by means of providing domain specific models of the metamodel [Ref.]. Moreover, we want to show how the components of the framework are playing together. To this end, instantiations of the metamodel will be provided for two cases. The resulting process models will detail what happens in the field while applying the framework.

We present two practical cases for the instantiation of the agile RE metamodel. On the one hand, we derive an instantiation of the agile RE metamodel into a Kanban-based process model for developing an internet-based newspaper portal within the e-commerce area (case a). The corresponding case study was carried out in Germany. On the other hand, we exemplify a process model for a Scrum-based process regarding the development of web applications in the e-government area (case a). This case study was carried out in Spain.

This chapter is structured as follows: section 7.1 outlines the general application of the conceptual approach and discusses how the framework can be applied in industry. Then, section 7.2 describes the applied method for creating and instantiating the metamodel by means of a profile. In addition, the section discusses on the instantiation into process models. Section 7.3 analyzes the results and limitations of this research. Therefore, we analyze the benefits and difficulties we observed while applying the framework.

7.1. General Application of the Framework

A user can use a product in several different manners. During the usage of a product, new ways of how to use the product are discovered and the usage is tailored to the context of use. The same metaphor is applicable for the framework presented in this PhD thesis. Organizations are complex systems and each environment is influenced by several factors as outlined by the agile RE metamodel (see Figure 17). The organizational environment in which the product development is carried out is volatile due to the different parameters, which affects it. To this end, there do not exist one fits all solution for agile RE. The proposed framework of this PhD thesis allows tailoring domain specific models for agile RE, which can be applied to specific environments.

Organizations can use the framework for modeling agile RE. On the one hand, they can use it to increase the value delivery. Value delivery is an organization external parameter since it is related to the outcome of the product according to users' and customers' needs. The definition of what value means can vary from organization or environment and needs to be defined by people who work on the product. On the other hand, organizations can utilize the framework to improve the collaboration among people working in the environment, where the product development takes

place. This is an organization internal parameter, which has a direct impact on how people feel during their work as well as on the effectiveness of product development.

The framework for modeling agile RE provides a collection of different techniques and tools, which needs to be tailored to a specific organizational environment for being applicable. Figure 47 outlines the components of the framework and describes how the framework can be used in industry.

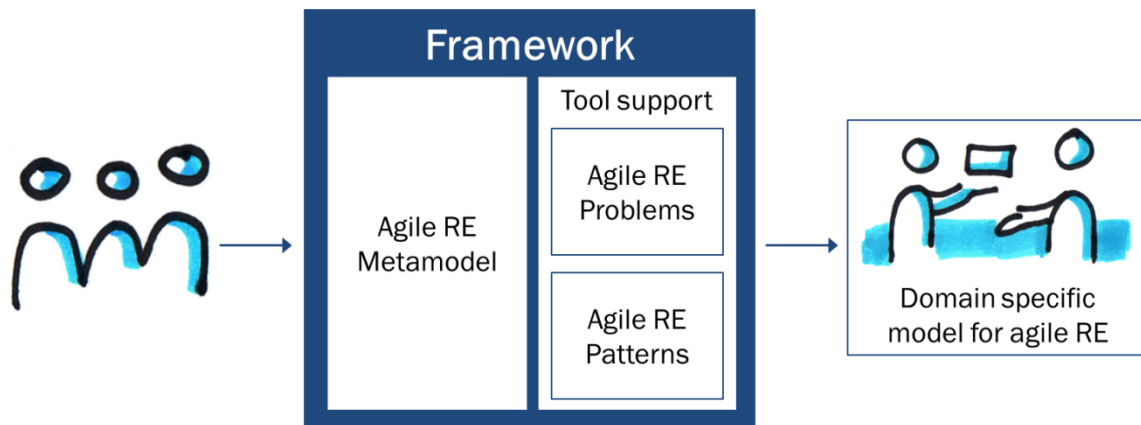


Figure 47: General application of the framework by means of tailoring domain specific models for agile RE

The framework for modeling agile RE consists of:

- The agile RE metamodel, which can be used to analyze the environment of an organization.
- A catalogue of agile RE problems, which support identifying recurring problems in terms of agile RE within the organizational environment.
- A catalogue of agile RE patterns, which allows solving the identified agile RE problems.

In terms of striving for continuous improvement of value delivery and collaboration, people can use the framework for analyzing the environment of an organization. In this context, the agile RE metamodel allows to analyze the problem domain by means of changing viewpoints in terms of people, relationships, process, and tools. The tool support of the framework (agileRE.org) supports detecting agile RE problems and proposes appropriate agile RE patterns, with which the agile RE problems can be solved. As a result of applying the framework, we receive a domain specific model for agile RE, which is tailored to the specific needs of the environment.

In general, the target group of the framework comprises practitioners as well as researchers, who are developing a product within an (agile) environment. However, the framework is beneficial for complex environments, where different people are involved in the process of product development. This is in alignment with the recommendations for the application of agile methodologies, which are appropriate for solving complex problems.

In the following sections, we present two examples of how to use the framework in industry by means of providing instantiations of the agile RE metamodel into domain specific models.

7.2. Research Method

In the following sections we will detail the objectives of this chapter and present the approach, which was followed in order to achieve the objectives.

7.2.1. Objectives and Research Questions

This chapter aims to show how the framework can be applied in industry. We will demonstrate how the agile RE metamodel can be instantiated to domain specific models. These domain specific models describe what actually happens in the field and detail the way of working. This objective will be addressed by the following research question:

- RQ-7.1: How can the metamodel be tailored to different process models?

7.2.2. Tailoring the Metamodel into Domain Specific Models

In a first step, the agile RE metamodel (Figure 17) needs to be instantiated in terms of conditions of the specific organizational environment. This step is related to the approach by Rolland (Rolland, 1993) who presents a technique for process modeling by means of building abstraction levels (process meta-level, process model and development runs). Following this approach, we are able to define how to apply the agile RE metamodel in practice. Therefore, we create instances of the metamodel (L2) to derive domain specific models (L1) that could be applied in industry (see Figure 48). We adapted the approach by Rolland to become more systematic. Hence, we recommend using a profile for building the instances of the agile RE metamodel. In our framework, the profile can be used as Add-in for EA (see section 4.3.3, 4.4.3).

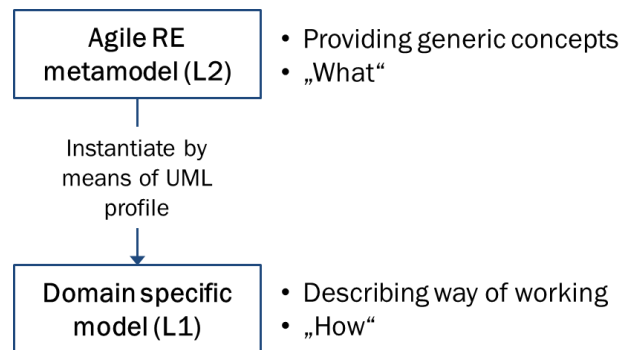


Figure 48: Modeling agile RE by instantiating the agile RE metamodel

In a next step, the catalogue of agile RE patterns can be used in order to detect specific agile RE problems. If one agile RE problem within the organizational environment is detected, the metamodel allows us to enrich this agile RE problem with contextual information.

These contextual information allows us, to choose appropriate agile RE patterns out of the pattern catalogue, which support solving the detected agile RE problem.

The next section shows how we used the framework for modeling agile RE in two different cases in industry.

7.3. Results and Discussion – Application of the Framework

The framework for modeling agile RE allows us to analyze the organizational environment, where the product development takes place. To this end, the agile RE metamodel can be tailored to process models in order to be applicable to industry. In the following subsections, we will present two cases in which we have built instances of the metamodel in order to create domain specific models.

The first process model (case a) describes an environment that uses Kanban for the relaunch of an internet-based newspaper portal. In contrast, the second process model (case b) is applied to an environment that uses Scrum for the development of a web application for e-government, with the aim to solve the problem of discovering services (e-government services).

The first application of the framework took place in a medium-sized company located in Germany. After the successful improvement of the organizational environment in that case, we decided to explore the usage of the framework in another country in order to demonstrate the international applicability. For this purpose, we decided to apply the framework in the regional government of the Spanish region of Andalusia. In the following, the two cases are explained in detail.

7.3.1. Case A: Tailoring Within a Kanban-based Environment in Germany

Kanban (Anderson, 2010) is an agile methodology whose main attribute focuses on continuous improvement. The application of Kanban starts with making a workflow visible and proceeds with continuous improvement of the existing process model. Compared to the time-box approach in Scrum, Kanban is based on a continuous flow model. In the following case, we will present an instantiation of the agile RE metamodel into a Kanban-based process model for developing an internet-based newspaper portal. The full description of the case study can be found in Schön et al. (Schön et al., 2016b).

7.3.1.1. Background of the Project

The project was carried out in a medium-sized IT company, located in Germany (Schön et al., 2016b), specialized in e-commerce, mobile apps and SAAS tools. The project team consisted of twelve members (one team leader, one project manager, two visual designers, two UX experts and six developers). The aim was the relaunch of an internet-based newspaper portal within six months in 2013/2014.

7.3.1.2. Instantiation into Kanban-based Process Model

In the following paragraphs, we will provide an instantiation of the metamodel (Figure 17) and we will discuss the way of working within the scope of case a. Figure 49 displays the domain specific model for agile RE in a Kanban-based environment for case a. For creating the instance, we used the profile in Enterprise Architect (see section 4.4.3).

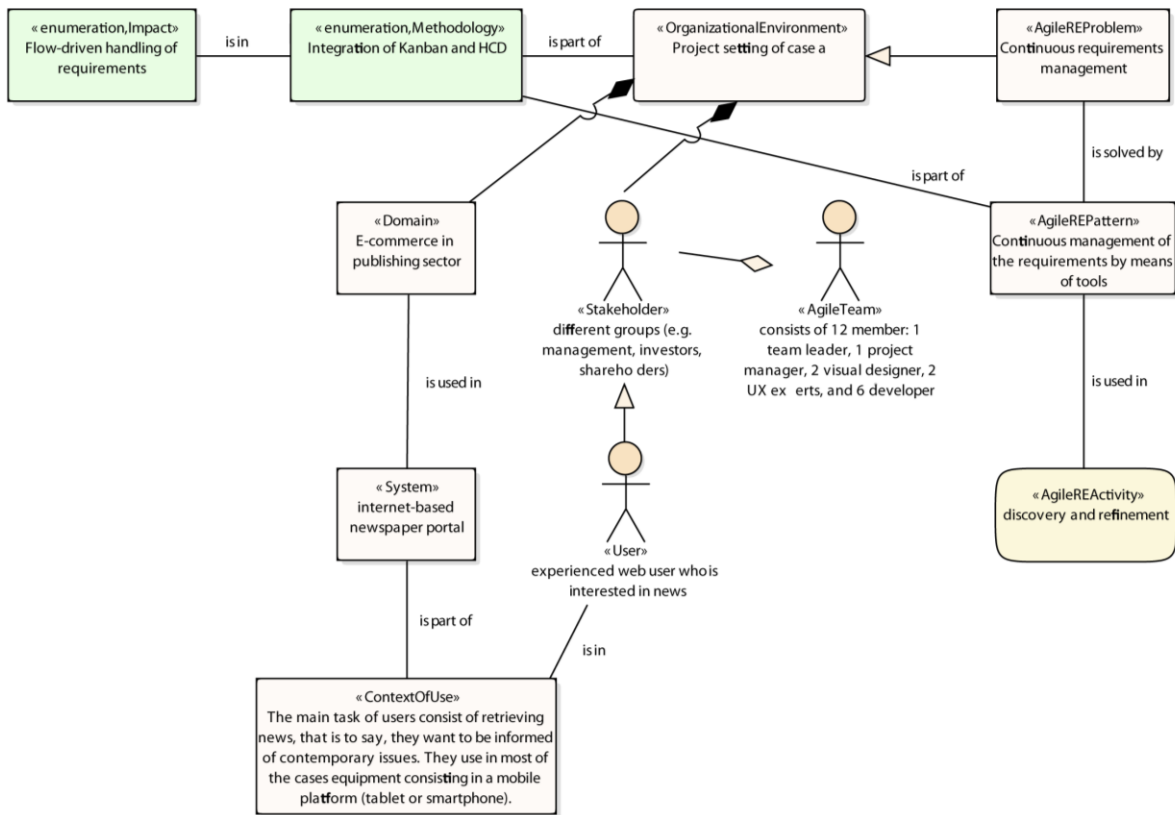


Figure 49: Domain specific model for agile RE in a Kanban-based environment

As it is shown in Figure 49, the methodology in case a is an integration of Kanban with HCD. The flow-driven approach of Kanban has an impact on how the requirements are handled. The domain where the system is used can be classified as e-commerce in the publishing sector. In light of the system, we can state that the system to be developed is an internet-based newspaper portal. In terms of the analysis of the organizational environment, we observed a lot of different stakeholders, who are involved in the development process of the system. The stakeholder groups are for instance management, investors or shareholders. In the scope of the agile RE metamodel, the user is a special type of a stakeholder. In case a, the user of the system is an experienced web user who is interested in news. Printed media is too slow for her/him and s/he appreciates consuming videos and additional interactive content. With regard to the context of use, in which the user is in, we observed that the main task of users consist of retrieving news, that is to say, they want to be informed of contemporary issues. Therefore, the first touch point of users is the home page where they can browse through the teasers and select the interesting ones for further reading. They use in most of the cases equipment consisting in a mobile platform (tablet or smartphone). Both the physical environment and the social environment of users depend on the situation in which they use the system. For instance, they can use the system in public transport that may be loud and crowded, or they can use it in their gardens where the sun is shining on the display. As mentioned before, the agile team consists of twelve members: one team leader, one project manager, two visual designers, two UX experts and six developers. All the aforementioned information describe the organizational environment in terms of the project setting for case a.

Analyzing the organizational environment, we detected the agile RE problem *continuous requirements management*. Because of the context, we decided to solve this problem by means of

using the agile RE pattern *continuous management of the requirements by means of tools*. This pattern can be categorized by the agile RE activities discovery and refinement, since both activities take place in terms of requirements management while applying the pattern.

In the following, we will present the agile RE pattern *continuous management of the requirements by means of tools*, which was applied in case a.

Pattern name	Continuous management of requirements by means of tools
Context	Established RE approaches fit into sequential approaches to software development. On the contrary, ASD is used to enhance the ability to deal with changing requirements over the course of time. Agile techniques for continuous requirements management need to be implemented in order to ensure traceability of requirements.
Tag	Discovery, refinement, methods
Problem summary	<p>In ASD, continuous management of requirements is a problem since not all of them are fixed at the beginning and they may change over the course of the project (see C4, Table 21).</p> <ul style="list-style-type: none">• providing a positive UX to the user.• carrying out a release evaluation continuously.• not interrupting the workflow due to scheduling testing activities and organization.• reducing costs for long-term UX testing.
Usage description	The workflow of the system development is visualized by means of Kanban boards for different types of tasks (e.g. UX design, development or operation). One requirement can be tracked along the workflow and its evolution is managed through the whole development process (see Figure 50). Organizing the management of requirements by means of Kanban boards implies a continuous management and tracking of changes.
Example	Figure 50 presents the interaction of two Kanban boards in case b. We used a third Kanban board for operation that was placed on the right hand side of the development board. The Kanban boards represent the workflow from design through development of the internet-based newspaper portal. One task from the design board might be split to more than one task on the development or the operation board. The aim of this procedure is to obtain a continuous flow within the board and among the boards. For working with multiple Kanban boards the project team used tool support by means of JIRA from Atlassian, displayed on several screens on the wall.
Template	See Figure 50

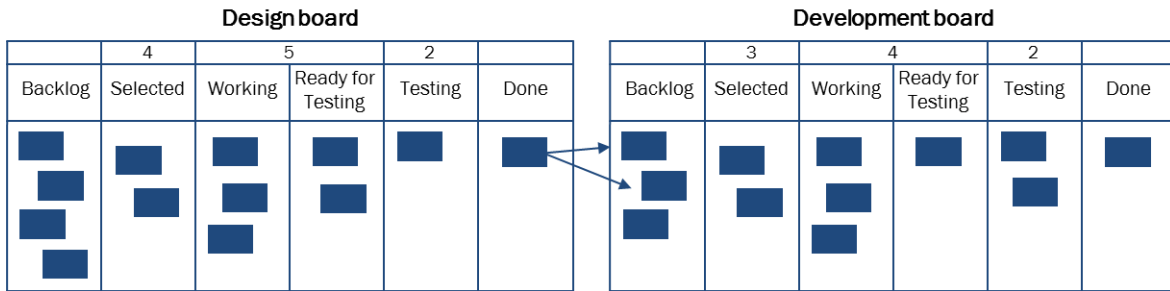


Figure 50: Example of multiple Kanban boards for pattern Continuous management of requirements by means of tools

It is worth mentioning that more than one of the presented agile RE problems occurred along the development of project case b. For instance, it was a problem not to lose sight of the big picture during the implementation of complex requirements since the development with Kanban is focused on small tasks. Therefore, the integration of HCD activities by means of a release evaluation was applied (Schön et al., 2016b).

7.3.2. Case B: Tailoring Within a Scrum-based Environment in Spain

Scrum (Beck, 2000) is an agile methodology that can be classified as an time-box approach comprising a set of roles, meetings and artifacts. Since Scrum is described as a framework where people can address complex problems, we need to adapt the framework to the specific context of a project. This case presents an instantiation of the agile RE metamodel in a project using NDT-Agile (based on Scrum and Earned Value Management) to create a web application for e-government. The full description of the case study can be found in Sedeño (Sedeño, 2017).

7.3.2.1. Background of the Project

The project was carried out at Junta de Andalucía, the regional government of the Spanish region of Andalusia. The Information and Communication Technologies (ICT) Department of the Ministry of Culture has led the project in liaison with other departments belonging to the regional government.

In between 2008-2012, the Ministry of Culture transformed itself into an organization to be able to operate under SOA paradigm (Sedeño et al., 2014). In 2012, they started to adopt agile methodologies (Torrecilla-Salinas et al., 2013), (Torrecilla-Salinas et al., 2015).

7.3.2.2. Instantiation into Scrum-based Process Model

In the following paragraphs, we will provide an instantiation of the metamodel (Figure 17). Moreover, we will discuss the way of working within the scope of case b. Figure 51 displays the domain specific model for agile RE in a Scrum-based environment for case b. For the creation of the instance, we used the profile in Enterprise Architect (see section 4.4.3).

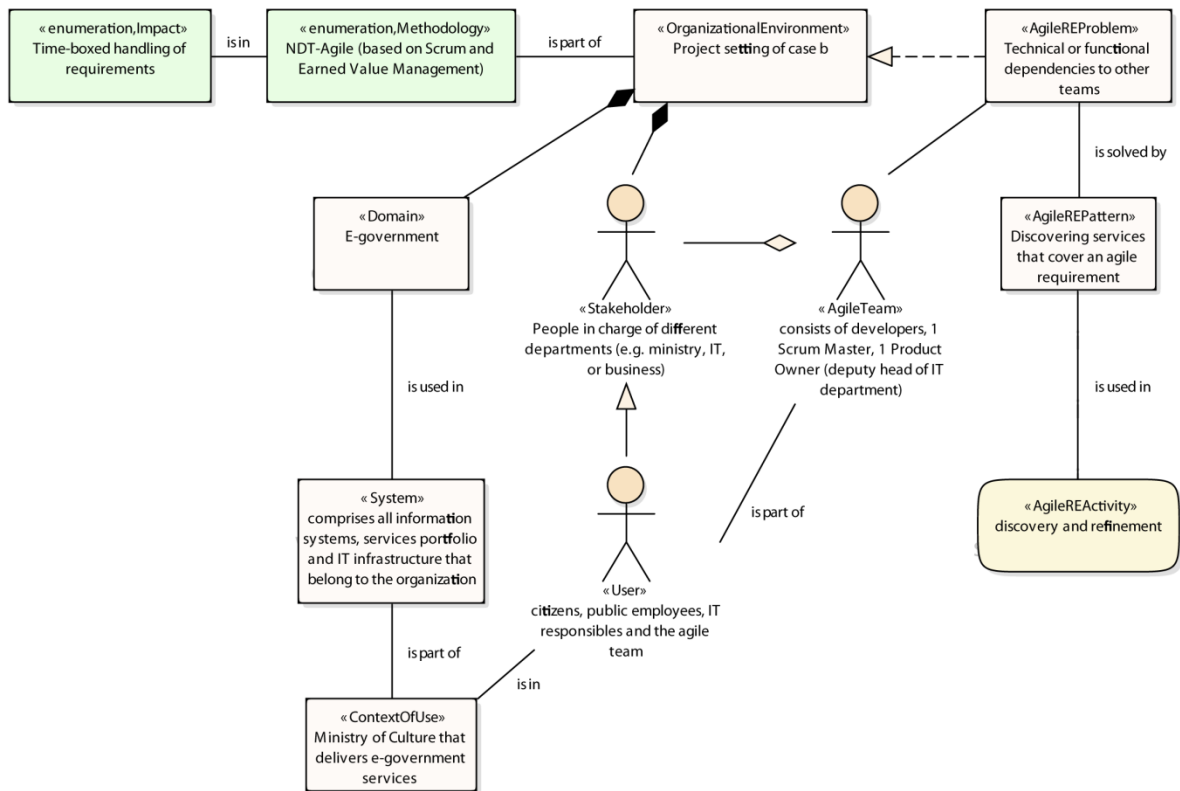


Figure 51: Domain specific model for agile RE in a Scrum-based environment

The methodology, which had been used in case b is NDT-Agile (based on Scrum and Earned Value Management). Scrum as well as NDT have an impact on how requirements are managed in case b. Requirements are managed by a Product Backlog and described in the form of user stories or using NDT. The domain of the organizational environment is related to the public sector and in particular, the area of e-government. The system in case b comprises all information systems involved in these web applications together with the services portfolio that belong to the organization. Moreover, the IT infrastructure is key part of the system. With regard to the stakeholder, we can describe them as people in charge of the different departments (e.g. ministry, IT or business). The users are citizens and public employees as well as the IT responsible or even the agile team. In light of this, the context of use is the Ministry of Culture that delivers e-government services by means of web applications using agile software developments. An agile team belongs to the IT service and consists of one Scrum Master, agile developers, and one Product Owner, who is the deputy head of IT department. All the aforementioned information describes the organizational environment in terms of the project setting for case a.

With regard to this analysis, we detected the agile RE problem *technical or functional dependencies to other teams*. In particular, the problem is related to the discovery of what services exist in the services portfolio, which are covering the user story proposed for a new web application. Moreover, this problem is related to software reusability in early stages of the software development lifecycle. The detected agile RE problem was solved by applying the agile RE pattern *discovering services that covers an agile requirement*. This pattern is categorized by the agile RE activities *discovery and refinement*. In the following, we will detail the pattern *discovering services that cover an agile requirement* which was applied in case b.

Pattern name	Discovering services that cover an agile requirement
Context	Detecting existing services, which can be found in the service portfolio of the organization, that cover user stories functionality identified by the agile team. This pattern is used in early stages of a sprint, directly after the Sprint Planning and it is necessary for the software reutilization. The identification of existing services is a very difficult task since the services area is not connected with the agile techniques that are used in the requirements elicitation phase.
Tag	Discovery, refinement, methods
Problem summary	In ASD functional or technical dependencies with other teams are a challenge because a considerable coordination effort is required (see C1, Table 21). <ul style="list-style-type: none"> • discovery of services in early stages of ASD. • discovering what services exist in the services portfolio, which cover the user story proposed for a new web application. • reusability of software in early stages of the software development lifecycle.
Usage description	For applying the solution, the process named DS4aRE (<i>Discovering services for agile requirements</i>) by (Sedeño, 2017) is used: <ol style="list-style-type: none"> 1) Indexing the Services Portfolio. 2) Formalizing the agile requirement by means of the value-based agile requirements metamodel (Sedeño, 2017). 3) Transforming this formalization into a query. 4) Launching this query against the index. 5) Preparing the result of the query. 6) Obtaining the Candidate Services (i.e. a Service that is inside the Services Portfolio and would cover an agile requirement).
Example	The Ministry needs to develop a new web application to comply with Law 39/2015 that obliges to make any notification by telematics means. The Ministry needs to know how many services of the existing e-government services portfolio could be used in this application to cover the user stories identified by the agile team.
Template	DS4aRE process (Sedeño, 2017)

7.3.3. Impact and Limitations

The application of the agile RE metamodel allows us to reflect the agile RE problems, which occurred in the presented cases (see section 7.3.1 and section 7.3.2) as well as to find appropriate solutions for handling these problems.

Analyzing the two cases, we can observe some differences in terms of the elaboration of information related to specific stereotypes (see Figure 49, Figure 51). For instance, the context of use is deeper explored and more elaborated in case a (see Figure 49). This effect occurs because of the integration of HCD. In HCD, the context of use is very important in terms of user research. Taking this as an example, we can discuss whether it makes sense to put some effort in the analysis of the context of use in case b and discuss what impact this analysis might have on the value delivery.

Further differences between the two domain specific models can be found in the manner of how people are related to each other. For instance in case b (Figure 51), the user is part of the agile team, since some member of the agile team use the system as citizens or employers of the Ministry of Culture.

Another difference between both domain specific models can be observed in the relation between the methodology and the agile RE pattern. In case a, the agile RE pattern is part of the methodology, since the agile technique *Kanban board*, is rooted in the methodology *Kanban*. In comparison, the agile technique in case b (DS4aRE process), is a new technique, which is not part of a methodology.

The profile is capable of presenting more details in the resulting domain specific models as displayed in Figure 49 and Figure 51. For instance, all attributes of the metaclasses from the agile RE metamodel (see Figure 17) can be displayed with their corresponding values. For both presented cases in this PhD thesis, we decided to hide this feature, because it is not necessary to show the details for understanding the application of the framework.

In general, the longer the framework is used, the more mature becomes the organizational environment and more information in terms of details of the domain specific model will be clarified.

With regard to the presented cases in this work, there are some shortcomings in terms of the analysis of the organizational environment. We have to state that we were not able to gather all the data the metamodel is capable of due to the time, when the case studies were carried out. The final version of the metamodel and the profile were created in July 2017, whereas the case studies were conducted before this time. This fact causes us some difficulties in applying the whole framework accurately. Nevertheless, we started new case studies, where we are going to apply the full framework in an international environment where the product development takes place with people working in different countries.

However, what we have learned from both cases is that the domain specific models allow us to receive a detailed view on complex organizational environments. This detailed view enables us to identify the problems and impediments that occur in terms of people, processes, or tools. In addition, the detailed view supports us to compare specific excerpts of different organizational environment with the aim to reflect what went well and what should be shared.

In light of this, the combination of agile RE problem and corresponding solution can be inferred to new agile RE patterns. These agile RE patterns facilitate sharing the experience with the

community. The gathered knowledge can help other people to solve their own agile RE problems, which may be similar to the presented ones.

Moreover, we were able to demonstrate that the framework for modeling agile RE is applicable on an international level and brings benefits for the industry.

7.4. Chapter Summary

In this chapter we showed how the framework for modeling agile RE can be applied in industry and how the different components of the framework collude with each other. Moreover, we explained the research method we followed to receive domain specific models for two cases.

The first domain specific model for agile RE (case a) presents a Kanban-based process model for developing an internet-based newspaper portal in the area of e-commerce. The case study was conducted in Germany. In this case, we detected the agile RE problem *Continuous requirements management*. This problem was solved by applying the agile RE pattern *Continuous management of requirements by means of tools*.

The second domain specific model for agile RE (case b) highlights a process model for a Scrum-based process regarding the development of web applications in the e-government area. This case study was located in Spain. In case b, we detected the agile RE problem *Technical or functional dependencies to other teams*, which was solved by using the agile RE pattern *Discovering services that cover an agile requirement*.

Moreover, we discussed the benefits and difficulties, which occur while applying the framework for modeling agile RE.

What we can conclude is that the framework for modeling agile RE can be applied on an international level. In addition it brings benefits for the industry since it allows to analyze the organizational environment of a company and helps to improve it in terms of value delivery and collaboration.

After presenting how the framework for modeling agile RE is applied in industry, the following chapter will summarize the results of this PhD thesis together with the conclusion.

With this PhD thesis, we contribute a framework for modeling and improving value delivery in agile environments. The framework allows improving the agile environment within organizations by means of a conceptual approach as well as techniques to apply the conceptual approach. The framework for modeling agile RE has been built by an inductive (bottom up) research approach, where we used the learnings from our research studies in order to create the components of the framework.

In particular, we analyzed the state of the art of agile RE with focus on stakeholder and user involvement. Based on this knowledge, we contribute the agile RE metamodel, which allows us to build a high level view on the complex field of agile RE. In addition, we created a profile. Then, we identified the most important problems agile practitioners have to face in industry in terms of agile RE. With regard to our conceptual approach, we analyzed solutions for handling these agile RE problems. Moreover, we created agile RE patterns, which are distributed by means of a web application (agileRE.org) in order to make the framework accessible for the community and to share the knowledge about agile RE.

In this chapter we will summarize the results of the PhD thesis. In addition, we will outline the identified future lines of research and the relations to the work of our research group (IWT2). Finally, the main conclusions of this work will be presented.

8.1. Results

Agility is increasing throughout organizations. Agile methodologies and agile techniques are used for product development across almost all industries. People in agile environments strive for continuous improvement, which leads to a rapidly evolving knowledgebase in the field of ASD. Agile techniques like continuous delivery (Humble and Farley, 2010) or DevOps (Hüttermann, 2012) are changing the way software is developed and delivered from a technical point of view.

However, requirements are often managed in a plan-driven manner although agile techniques like story mapping (Patton, 2014) or impact mapping (Adzic, 2012) are in place. Organizations still struggle with improving their processes systematically in order to increase value delivery and collaboration.

The main contribution of this PhD thesis is a framework for modeling agile RE. The framework consists of three main components:

- The agile RE metamodel, which allows analyzing the environment of an organization, where the product development takes place.
- A catalogue of agile RE problems, which supports detecting common problems that occur in terms of agile RE.
- A catalogue of agile RE patterns, which can be used to improve the environment and support solving the identified agile RE problems.

The framework provides several benefits for organizations. On the one hand, organizations can use the framework for increasing their value delivery. Value delivery is an organization external parameter, which is related to the outcome of a product in terms of users' and customers' needs. The people working in an organization need to define what value means and need to find appropriate KPIs for measuring the value. In this context, they need to leave room for iterating on the definition of value. On the other hand, organizations can utilize the framework to improve the collaboration among people working in the environment. Collaboration is an organization internal parameter, which is related to how people feel during their work as well as on the effectiveness of product development.

The presented framework enables agile practitioners as well as researchers to improve their environments in a systematic manner. In particular, the framework is suitable for complex environments, where a lot of different people with different roles are involved or hybrid models for product development are applied as shown in the case studies.

With regard to the vision of this PhD, which was decomposed into four main objectives (see section 3.2), the following results have been achieved:

- An analysis of the state of the art of agile RE with focus on stakeholder and user involvement.
- A conceptual approach for agile RE management by means of the agile RE metamodel and two examples of domain specific models, which show how the metamodel can be applied in industry. The domain specific models are created by means of a profile for the agile RE metamodel.
- Techniques to apply the conceptual approach to the enterprise environment by means of agile RE problems and agile RE patterns.
- Tool support by means of a web application (agileRE.org)

We analyzed the state of the art of agile RE with focus on user and stakeholder involvement by means of a SLR (Schön et al., 2017a). Therefore, the most relevant related work regarding the relation among Agile, HCD, and RE has been presented. The state of the art study helped us to achieve deep insights into the field of agile RE, in particular in terms of:

- presence of the user perspective
- stakeholder and user involvement
- integration of methodologies to make ASD more human-centered
- building a shared understanding
- data gathering
- non-functional requirements
- documentation of requirements by means of artifacts

Based on this study, we were able to identify gaps in existing research, which helped us to target our research. The available literature shows that there are still challenges regarding the iterative involvement of user and stakeholder in the development process as well as building a shared understanding of the user perspective. We found a variety of methods and artifacts, which are used in ASD for different purposes in a specific context. However, guidelines for choosing the right kind of artifacts that enhance collaboration among all people involved in the development process are missing.

In a next step, we created the agile RE metamodel, aiming to build an abstraction layer about the variety of existing process models for agile RE in the field. The initial version of the agile RE

metamodel is inferred by an analysis of related process models for agile RE and experiences working in industry projects (Schön, 2016). We improved the metamodel iteratively by means of discussions with other researchers and further empirical studies [Ref].

The agile RE metamodel builds the heart of our framework and shapes the conceptual approach for agile RE management. It allows us to reflect the dependencies among people and product development process and supports improvement by inspection and adaptation.

The metamodel has several implications for both, researchers and practitioners. On the one hand, the metamodel can be used to model new process models for agile RE that are strongly focused on value delivery and therefore can be classified as value-driven process models. On the other hand, existing process models can be evaluated as well as improved by the metamodel.

Moreover, we contribute a visual language for the agile RE metamodel by means of a profile. The profile allows us to extend the metaclasses of the agile RE metamodel with the aim to tailor domain specific models.

Subsequent to the definition of our conceptual approach, we contribute techniques (a catalogue of agile RE problems and a catalogue of agile RE patterns) that allow us to apply the conceptual approach to the enterprise environment.

For creating the catalogue of agile RE problems, we carried out an iterative expert judgement process comprising three complementary rounds (Schön et al., 2017d). Our panel consisted of 26 experts in the field of ASD working for 19 different companies. During these rounds we find the core of a specific problem by abstracting contextual information of the problems. The learnings from previous iterations were used for carrying out the following ones. In sum, we identified 20 problems industry has to address at present in terms of agile RE. Six of these problems have been defined as key problems:

- technical or functional dependencies to other teams (C1)
- understanding of agile values of the stakeholders (C2)
- staying focused on the big picture (C3)
- continuous requirements management (C4)
- refine requirements in collaboration with users (C5)
- involve stakeholder iteratively (C6)

A further result of this study is the analysis of options to deal with those key problems by means of agile techniques, which are recommended by the panel. These results supported us in creating the catalogue of agile RE patterns.

The agile RE problems and agile techniques allow us to create agile RE patterns. As part of the results of this PhD thesis, we introduced the concept of agile RE patterns (Schön et al., 2017b). For creating the agile RE patterns, we carried out a systematic pattern mining process comprising three phases. In the first phase, we collected the agile techniques identified by our SLR. Then, we combined these results with the ones from our iterative expert judgement process. Finally, we mapped the results in order to create agile RE patterns. An agile RE pattern is composed of an agile RE problem and appropriate agile techniques, which can be utilized to solve the problem. With this mapping, we derived in sum 41 agile RE patterns. In this work, we showed an overview of the 41 identified agile RE patterns and presented examples: a) *Evaluation and Testing*, b) *Story Map*, c) *Product Owner* and d) *Review Meeting*.

The identified agile RE patterns and the knowledge concerning agile RE will be shared among the community by means of a web application (agileRE.org). agileRE.org aims to provide tool support for our framework.

For a better understanding of our conceptual approach, we showed how the framework for modeling agile RE can be applied in industry by means of domain specific models. To this end, we provided two examples of instantiations of the metamodel. Firstly, we exemplify a process model for a Kanban-based process model for developing an internet-based newspaper portal within the e-commerce area. This case detected the agile RE problem *Continuous requirements management*. This problem is solved by applying the agile RE pattern *Continuous management of requirements by means of tools*. Secondly, we derive an instantiation of the agile RE metamodel into a Scrum-based process regarding the development of web applications in the e-government area. This case detected the agile RE problem *Technical or functional dependencies to other teams*, which is solved by using the agile RE pattern *Discovering services that cover an agile requirement*.

Summarizing the implications of this PhD thesis, we can state that we identified evidence in industry by means of identifying real world problems in the context of agile RE. The problems were generalized by reducing the contextual information of the environment. With regard to our conceptual approach for agile RE management we can learn to understand the dependencies and to reflect problems in a specific environment. In this context, the agile RE metamodel allows to analyze the environment within an organization and supports adding contextual information according to the specific needs. This is a necessary task in order to understand the conditions for selecting an appropriate agile technique, with which the detected problem can be solved. Besides, agile RE patterns help to share the knowledge and experiences with problem solving in the context of agile RE among the community.

8.2. Future Work

This PhD work has opened new lines of future research, which will be outlined in the next paragraphs.

In the scope of this PhD thesis, we showed how our framework can be utilized for improving domain specific agile RE process models in the area of e-government and e-commerce (brownfield approach). In the next years, we want to apply the framework on a greenfield approach, where we can use it to model new process models for agile RE according to needs of an organization. In industry, it is not so easy to find an appropriate case, since most of the companies already have a process model in place and it is hard to find an environment where we can start from scratch. Furthermore, we are going to apply the framework to other domains since this will allow us to learn more about the boundaries of the framework in a real world context. In a next step, we want to create a marketable version of the framework so that the value of the framework becomes more handy to industry. First companies indicate that they are highly interested in cooperation and applying the framework.

Moreover, future research may specifically focus on integrating further tools that support the semi-automatic analysis of requirements in an agile environment, similar to NDT (Escalona and Aragon, 2008) which is used for automatic analysis of requirements in sequential approaches to RE.

Another thing to highlight is that we want to enrich the identified agile RE patterns by means of further empirical research in industry. Some of the patterns had been applied in industry, whereas

others are composed during the work of this thesis and therefore are not fully evaluated. Although our agile RE patterns are a combination of best practices known from agile methodologies, we want to evaluate most of them. Those patterns, which have not been evaluated in case studies, can be classified as *proto patterns*. Our aim is to gain more experience with the agile RE patterns and to learn how they can support organizations in terms of becoming a value-driven culture. Moreover, we want to create a pattern language based on our learnings from the application of the framework for modeling agile RE.

We started to carry out further studies in order to learn more about the identified agile RE problems. First workshops among UX experts showed, that the discussions about the agile RE problems enable collaboration in solution finding among cross-functional teams (Schön and Thomaschewski, 2017).

The results of our iterative expert judgement process reveal that the identified agile RE problems are not limited to ASD. To this end, another important future line of research is to investigate in what degree the agile RE problems are applicable to RE in general. These results may offer new opportunities to extend our framework. In this context, a cooperation with the research group of the NaPiRE surveys (Méndez Fernández et al., 2017) is planned.

Moreover, we learned that patterns are a good way to playback results from research to industry. In future work, we want to investigate what more options do exist to bridge the gap between research and industry and figure out new ways how research can be applied in companies.

8.3. Relation to IWT2 research group

At this point, we want to outline the relation of this work to the research work of the IWT2 research group. In general, RE and ASD have been lines of research of the IWT2 group for several years.

The PhD work of Dr. D. Jorge Sedeño López (Sedeño, 2017), supervised by Dr. María José Escalona Cuaresma and Dr. Manuel Mejías Risoto, is focused on discovering services in early stages of ASD and contributes a process for the formalization of requirements in an agile environment. The formalization of requirements is carried out by agile techniques like a Product Backlog or user stories within the NDT-Agile framework. The formalized requirement can be mapped to a catalog of existing services in order to evaluate, which services already exist that cover the requirement. To this end, a metamodel is defined, which allows formalizing a requirement. The activity of mapping requirements to an existing catalog of services is an important activity in terms of the reusability of software components.

The PhD work of Dr. Carlos J. Torrecilla Salinas (Salinas Torrecilla, 2016), supervised by Dr. María José Escalona Cuaresma and Dr. Manuel Mejías Risoto, contributes a framework named NDT-Agile. The aim of NDT-Agile is to provide a mature agile framework for web development projects. NDT-Agile combines the following components: a lifecycle, a set of complementary techniques, and a governance model. The lifecycle is based on Scrum and additional elements to be compliant to CMMI-DEV. The usage of CMMI-DEV increases process improvement and leads to a mature agile development process. The complementary techniques are for instance project subcontracting or project reporting. The governance model provides information regarding continuous improvement or quantitative-managed project management.

After presenting the results of this PhD work, highlighting the future lines of research and outlining the relations to the IWT2 research group, the following section will state the main conclusions.

8.4. Conclusions

This PhD thesis offers several contributions in the context of agile RE. It has identified an existing gap linked to the relationship among ASD, HCD and RE. We filled this gap by a framework for modeling agile RE, which allow agile practitioners and researchers to model and improve agile RE process models with the aim to increase value delivery and collaboration within their organizations.

Based on our state of the art analysis of agile RE, we can conclude that the research field of agile RE is very close to existing work practices in companies. Besides, agile RE is a complex research field with cross-functional influences from other domains like HCD, ASD, and RE. Moreover, we can state that agile environments are volatile and existing agile RE process models in organizations differ in terms of agile techniques like artifacts, meetings, methods, and roles.

Organizations want to adopt agile methodologies in order to accelerate product delivery. Nevertheless, most of them do not know which agile methodology or agile techniques best fits into their organization. In addition, they struggle with prioritizing requirements according to needs of their users and customers.

With regard to our framework for modeling agile RE, organization can use it to improve agile RE in a systematic manner. For one thing it can be used in terms of value delivery. For the other thing, it enables improvements in terms of collaboration. In this context, the framework gives guidance for choosing appropriate agile techniques, which fit to the needs of the people working in an organizational environment. Moreover, it allows improving hybrid process models for agile RE in a systematic manner without being restricted to one specific agile methodology.

We can conclude that the framework for modeling agile RE is applicable on international level, since we provided practical cases in industry for a project in Germany and one for a project in Spain. Moreover, we demonstrated what benefits the framework brings for the industry.

What we can learn from metamodeling is that a metamodel can help to visualize the dependencies and influencing parameters of agile RE. To this end, we can conclude that the agile RE metamodel allows to reduce the complexity due to providing overview and clearness about the influencing parameters in an agile environment. In this context, the profile enables us to build visual instances of the agile RE metamodel and eases the creation of domain specific models. Domain specific models show how the agile RE metamodel can be applied in industry as well as how agile RE process models can be optimized by means of a problem solving approach.

Our iterative expert judgement process with agile practitioners reveals that organizations still struggle with agile transition and understanding agile values, in particular in terms of stakeholder and user involvement. In light of this, the presented framework of this PhD thesis provides options to deal with these problems. Feedback concerning the study from both, the panel of experts and the reviewers of the paper expose that our agile RE problems are often not limited to ASD, but they would rather refer to software development in general. We can conclude that the discussion about the identified agile RE problems enable collaboration in solution finding among cross-functional teams by means of inspection and adaptation.

Another conclusion can be drawn concerning our agile RE patterns. The created agile RE patterns are highly relevant for the industry as well as the research community since we identified the agile RE problems by means of empirical research in industry. In addition, the agile techniques are collected by means of empirical research in existing literature. History shows that patterns are an effective way to communicate knowledge, since the knowledge is presented in chunks. To this

end, we can state that agile RE patterns are a good technique to playback our results from research to industry.

As final conclusion, it must be highlighted that the presented framework empowers organization to move from a plan-driven to a value-driven culture since the proposed techniques and tools allow increasing value delivery and collaboration.

As a final statement, it is worth mentioning that most of our contributions have already been published in several international journals and conferences, as detailed in Appendix V.

References

- Abdullah, N.N.B., Honiden, S., Sharp, H., Nuseibeh, B., Notkin, D., 2011. Communication patterns of agile requirements engineering, in: Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11. ACM Press, New York, New York, USA, pp. 1-4. doi:10.1145/2068783.2068784
- Adikari, S., McDonald, C., Campbell, J., 2013. Reframed contexts: Design thinking for agile user experience design, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). doi:10.1007/978-3-642-39229-0_1
- Adikari, S., McDonald, C., Campbell, J., 2009. Little design up-front: A design science approach to integrating usability into agile requirements engineering, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). doi:10.1007/978-3-642-02574-7_62
- Adzic, G., 2012. Impact Mapping: Making a big impact with software products and projects. Provoking Thoughts, p. 84.
- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., Angel, S., 1977. A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure). Oxford University Press.
- Anderson, D.J., 2010. Kanban - Successful Evolutionary Change for your Technology Business. Blue Hole Press.
- Anderson, D.J., 2004. Making the Business Case for Agile Management - Simplifying the Complex System of Software Engineering, in: Motorola S3 Symposium. pp. 1-13.
- Basso, T., Montecchi, L., Moraes, R., Jino, M., Bondavalli, A., 2015. Towards a UML profile for privacy-aware applications. Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Se 371-378. doi:10.1109/CIT/IUCC/DASC/PICOM.2015.53
- Beck, K., 2000. Extreme Programming Explained: Embrace Change. Addison-Wesley.
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D., 2001. Manifesto for Agile Software Development [WWW Document]. URL <http://www.agilemanifesto.org/> (accessed 5.2.17).
- Bellucci, A., Jacucci, G., Kotkavuori, V., Serim, B., Ahmed, I., Ylirisku, S., 2015. Extreme Co-design: Prototyping with and by the User for Appropriation of Web-connected Tags. pp. 109-124. doi:10.1007/978-3-319-18425-8_8
- Bertholdo, A.P.O., Kon, F., Gerosa, M.A., 2016. Agile Usability Patterns for User-Centered Design Final Stages. The Handbook of Communication Science 23-33. doi:10.4135/9781412982818.n29
- Bertholdo, A.P.O., Silva Da Silva, T., De O. Melo, C., Kon, F., Silveira, M.S., 2014. Agile usability patterns for UCD early stages. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8517 LNCS, 33-44. doi:10.1007/978-3-319-07668-3_4
- Beyer, H., Holtzblatt, K., 1998. Contextual design defining customer-centered systems. Morgan Kaufmann, San Francisco, Calif. :
- Bjarnason, E., Wnuk, K., Regnell, B., 2011. A case study on benefits and side-effects of agile practices in large-scale requirements engineering, in: Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW '11. ACM Press, New York, New York, USA, pp. 1-5. doi:10.1145/2068783.2068786
- Blomkvist, J.K., Persson, J., Åberg, J., 2015. Communication through Boundary Objects in Distributed Agile Teams, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15. ACM Press, New York, New York, USA, pp. 1875-1884.

doi:10.1145/2702123.2702366

- Bourimi, M., Barth, T., Haake, J.M., Ueberschär, B., Kesdogan, D., 2010. AFFINE for enforcing earlier consideration of NFRs and human factors when building socio-technical systems following agile methodologies. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6409 LNCS, 182–189. doi:10.1007/978-3-642-16488-0_15
- Brhel, M., Meth, H., Maedche, A., Werder, K., 2015. Exploring principles of user-centered agile software development: A literature review. *Information and Software Technology* 61, 163–181. doi:10.1016/j.infsof.2015.01.004
- Buchan, J., 2014. An Empirical Cognitive Model of the Development of Shared Understanding of Requirements, in: *Communications in Computer and Information Science*. pp. 165–179. doi:10.1007/978-3-662-43610-3_13
- Cajander, Å., Larusdottir, M., Gulliksen, J., 2013. Existing but Not Explicit - The User Perspective in Scrum Projects in Practice, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 762–779. doi:10.1007/978-3-642-40477-1_52
- Cambridge Dictionary, 2017. Meaning in the Cambridge English Dictionary [WWW Document]. URL <http://dictionary.cambridge.org/dictionary/english/framework> (accessed 5.4.17).
- Cohn, M., 2004. *User Stories Applied: For Agile Software Development*.
- Cooper, A., 1999. *The Inmates Are Running the Asylum*, 1st ed. Macmillan Publishing Co., Inc., Indianapolis, IN, USA.
- Dalkey, N., 1969. An experimental study of group opinion. *Futures* 1, 408–426. doi:10.1016/S0016-3287(69)80025-X
- Dalkey, N., Helmer, O., 1963. An Experimental Application of the DELPHI Method to the Use of Experts. *Management Science* 9, 458–467. doi:10.1287/mnsc.9.3.458
- Diamond, I.R., Grant, R.C., Feldman, B.M., Pencharz, P.B., Ling, S.C., Moore, A.M., Wales, P.W., 2014. Defining consensus: A systematic review recommends methodologic criteria for reporting of Delphi studies. *Journal of Clinical Epidemiology* 67, 401–409. doi:10.1016/j.jclinepi.2013.12.002
- Dingsøyr, T., Dyba, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology* 50, 833–859. doi:10.1016/j.infsof.2008.01.006
- Dingsøyr, T., Lassenius, C., 2016. Emerging themes in agile software development: Introduction to the special section on continuous value delivery. *Information and Software Technology* 77, 56–60. doi:10.1016/j.infsof.2016.04.018
- Dragicevic, S., Celar, S., Novak, L., 2014. Use of Method for Elicitation, Documentation, and Validation of Software User Requirements (MEDoV) in Agile Software Development Projects, in: *2014 Sixth International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE, pp. 65–70. doi:10.1109/CICSyN.2014.27
- Escalona, M.J., Aragon, G., 2008. NDT. A Model-Driven Approach for Web Requirements. *IEEE Transactions on Software Engineering* 34, 377–390. doi:10.1109/TSE.2008.27
- Escalona, M.J., Aragón, G., 2008. NDT. A model-driven approach for web requirements. *IEEE Transactions on Software Engineering* 34, 377–394. doi:10.1109/TSE.2008.27
- Escalona, M.J., Koch, N., 2007. Metamodeling the Requirements of Web Systems, in: Filipe, J., Cordeiro, J., Pedrosa, V. (Eds.), *Web Information Systems and Technologies - Lecture Notes in Business Information Processing*. Springer, Berlin, Heidelberg, pp. 267–280. doi:10.1007/978-3-540-74063-6_21
- Evans, E.J., 2003. *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison Wesley.
- Farid, W.M., 2012. The NORMAP Methodology: Lightweight Engineering of Non-functional Requirements for Agile Processes, in: *2012 19th Asia-Pacific Software Engineering Conference*. IEEE, pp. 322–325. doi:10.1109/APSEC.2012.23

- Finstad, K., 2010. Response Interpolation and Scale Sensitivity: Evidence Against 5-Point Scales. *Journal of Usability Studies* 5, 104–110.
- Fleiss, J.L., 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin* 76, 378–382.
- Garcia, A., Silva da Silva, T., Selbach Silveira, M., 2017. Artifacts for Agile User-Centered Design: A Systematic Mapping, in: *Proceedings of the Annual Hawaii International Conference on System Sciences*. pp. 5859–5868. doi:10.24251/HICSS.2017.706
- García García, J.A., 2015. Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocio. University of Seville.
- Glase, S., 2016. Skaliertes Scrum und die Sprint Review Messe [WWW Document]. URL <http://www.stefanglase.de/2016/04/24/skaliertes-scrum-und-die-sprint-review-messe/> (accessed 6.4.17).
- Gothelf, J., Seiden, J., 2012. *Lean UX: Applying Lean Principles to Improve User Experience*, 1st Edition. ed. O'Reilly and Associates.
- Harbers, M., Detweiler, C., Neerincx, M.A., 2015. Requirements Engineering: Foundation for Software Quality, International Workshop on Requirements Engineering: Foundation for Software Quality, REFSQ 2015, Lecture Notes in Computer Science. Springer International Publishing, Cham. doi:10.1007/978-3-319-16101-3
- Hartson, R., Pyla, P.S., 2012. *The UX Book: Process and Guidelines for Ensuring a Quality User Experience*. Morgan Kaufman Publ Inc, Amsterdam, and Boston.
- Hassenzahl, M., 2010. *Experience design: Technology for All the Right Reasons*. Morgan & Claypool.
- Heikkilä, V.T., Damian, D., Lassenius, C., Paasivaara, M., 2015. A Mapping Study on Requirements Engineering in Agile Software Development. 2015 41st Euromicro Conference on Software Engineering and Advanced Applications 199–207. doi:10.1109/SEAA.2015.70
- Henninger, S., Corrêa, V., 2007. Software pattern communities: current practices and challenges. *Proceedings of the 14th Conference on Pattern Languages of Programs* 1–19. doi:10.1145/1772070.1772087
- Humble, J., Farley, D., 2010. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, Continuous delivery. Pearson Education, Inc, Boston, MA.
- Hüttermann, M., 2012. *DevOps for Developers*, 1st ed. ed. Apress.
- Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S., 2015. A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior* 51, 915–929. doi:10.1016/j.chb.2014.10.046
- International Organization for Standardization, 2017. ISO/DIS 9241-220.2 - Ergonomics of human-system interaction - Part 220: Processes for enabling, executing and assessing human-centred design within organizations.
- International Organization for Standardization, 2010. ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems.
- International Organization for Standardization, 1999. ISO 13407:1999 - Human-centred design processes for interactive systems.
- International Organization for Standardization, 1998. ISO 9241-11:1998 - Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability.
- ISO, 2011. ISO/IEC 25010:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models.
- Issa, A.A., AlAli, A.I., 2011. Automated requirements engineering: use case patterns-driven approach. *IET Software* 5, 287. doi:10.1049/iet-sen.2010.0014
- Jalali, S., Wohlin, C., 2012. Systematic literature studies: database searches vs. backward snowballing, in: *Proceedings of the 2012 6th ACM/IEEE International Symposium on Empirical Software*

- Engineering and Measurement (ESEM). pp. 29–38. doi:10.1145/2372251.2372257
- Kamthan, P., 2013. On the interaction between socio-technical dimensions in the next generation of requirements engineering. *International Journal of Software Engineering and its Applications* 7, 183–196.
- Kautz, K., 2010. Participatory Design Activities and Agile Software Development, in: IFIP WG 8.2/8.6 International Working Conference. pp. 303–316. doi:10.1007/978-3-642-12113-5_18
- Kitchenham, B., Charters, S., 2007. Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering* 2, 1051. doi:10.1145/1134285.1134500
- Kohls, C., 2012. The path to patterns, in: *Proceedings of the 17th European Conference on Pattern Languages of Programs - EuroPLoP '12*. ACM Press, New York, New York, USA, pp. 1–16. doi:10.1145/2602928.2603085
- Komus, A., Kuberg, M., Arifaj, A., Bach, C., Berg, C., Czeicke, D., Frizler, A., Hamziu, B., Jäck, S., Janowski, C., Kengji, P., Schäfer, J., Schulz, M., Setiabudi, J., Sonntag, P., Sprenger, S., Wongwat, A., 2017. *Study Status Quo Agile 2016/2017*.
- Kraus, S., 2013. *Scaling the Sprint Review: The Fair - Scrum Alliance [WWW Document]*. URL <https://www.scrumalliance.org/community/articles/2013/january/scaling-the-sprint-review-the-fair> (accessed 6.4.17).
- Kropp, E., Koischwitz, K., 2016. Experiences with User-Centered Design and Agile Requirements Engineering in Fixed-Price Projects, in: Ebert, A., Humayoun, R.S., Seyff, N., Perini, A., Barbosa, S.D.J. (Eds.), *UsARE 2012/2014*, LNCS 9312. pp. 47–61. doi:10.1007/978-3-319-45916-5_4
- Kruchten, P., 2004. *The Rational Unified Process: An Introduction*, 3rd ed. ed. Addison-Wesley.
- Kruchten, P., 1998. *The Rational Unified Process*, 1st Editio. ed. Addison-Wesley.
- Larusdottir, M.K., Bjarnadottir, E.R., Gulliksen, J., 2010. The Focus on Usability in Testing Practices in Industry, in: *Proceedings of Second IFIP TC 13 Symposium, HCIS 2010, Held as Part of WCC 2010*. pp. 98–109. doi:10.1007/978-3-642-15231-3_11
- Lee, S.-H., Ko, I.-Y., Kang, S., Lee, D.-H., 2010. A Usability-Pattern-Based Requirements-Analysis Method to Bridge the Gap between User Tasks and Application Features, in: *2010 IEEE 34th Annual Computer Software and Applications Conference*. IEEE, pp. 317–326. doi:10.1109/COMPSAC.2010.39
- Legendre, P., 2005. Species associations: the Kendall coefficient of concordance revisited. *Journal of Agricultural, Biological, and Environmental Statistics* 10, 226–245. doi:10.1198/108571105X46642
- Lilja, K.K., Laakso, K., Palomäki, J., 2011. Using the Delphi Method. *Technology Management in the Energy Smart World (PICMET)* 1–10.
- Linstone, H.A., Turoff, M., 2002. *The Delphi Method - Techniques and applications*, *The delphi method - Techniques and applications*. doi:10.2307/1268751
- Liskin, O., Schneider, K., Fagerholm, F., Münch, J., 2014. Understanding the role of requirements artifacts in kanban, in: *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE 2014*. ACM Press, New York, New York, USA, pp. 56–63. doi:10.1145/2593702.2593707
- Lombriser, P., Dalpiaz, F., Lucassen, G., Brinkkemper, S., 2016. Gamified Requirements Engineering: Model and Experimentation, in: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 171–187. doi:10.1007/978-3-319-30282-9_12
- Losada, B., Urretavizcaya, M., Fernández-Castro, I., 2013. A guide to agile development of interactive software with a “User Objectives”-driven methodology. *Science of Computer Programming* 78, 2268–2281. doi:10.1016/j.scico.2012.07.022
- Lucia, A. De, Qusef, A., 2010. Requirements Engineering in Agile Software Development. *Journal of Emerging Technologies in Web Intelligence* 2, 212–220. doi:10.4304/jetwi.2.3.212-220

- Magues, D.A., Castro, J.W., Acuna, S.T., 2016. HCI usability techniques in agile development, in: 2016 IEEE International Conference on Automatica (ICA-ACCA). IEEE, pp. 1–7. doi:10.1109/ICA-ACCA.2016.7778513
- Maguire, M., 2013. Using human factors standards to support user experience and agile design, in: Proceedings - International Conference, UAHCI 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA. pp. 185–194. doi:10.1007/978-3-642-39188-0_20
- Magureanu, G., Gavrilescu, M., Pescaru, D., Jian, I., 2012. UML profile for Cyber-Physical System wireless communication specification, in: 2012 7th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, pp. 383–388. doi:10.1109/SACI.2012.6250034
- Mahmud, I., Veneziano, V., 2011. Mind-mapping: An effective technique to facilitate requirements engineering in agile software development, in: 14th International Conference on Computer and Information Technology (ICCI 2011). IEEE, pp. 157–162. doi:10.1109/ICCI.2011.6164775
- Mammel, T., Gundelsweiler, F., Reiterer, H., 2007. Agile Human-Centered Software Engineering. Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 1 167–175. doi:10.1177/004057368303900411
- Méndez Fernández, D., Lochmann, K., Penzenstadler, B., Wagner, S., 2011. A Case Study on the Application of an Artefact-Based Requirements Engineering Approach. 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE) 104–113. doi:10.1049/ic.2011.0013
- Méndez Fernández, D., Penzenstadler, B., Kuhmann, M., Broy, M., 2010. A meta model for artefact-orientation: Fundamentals and lessons learned in requirements engineering. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6395 LNCS, 183–197. doi:10.1007/978-3-642-16129-2_14
- Méndez Fernández, D., Wagner, S., Felderer, M., Kalinowski, M., 2017. Naming the pain in requirements engineering (NaPiRE) [WWW Document]. URL <http://www.mendezfe.org/projects/napire/> (accessed 6.8.17).
- Méndez Fernández, D., Wagner, S., Kalinowski, M., Felderer, M., Mafra, P., Vetr?, A., Conte, T., Christiansson, M.-T., Greer, D., Lassenius, C., M?nnist?, T., Nayabi, M., Oivo, M., Penzenstadler, B., Pfahl, D., Prikladnicki, R., Ruhe, G., Schekelmann, A., Sen, S., Spinola, R., Tuzcu, A., de la Vara, J.L., Wieringa, R., 2016. Naming the pain in requirements engineering. Empirical Software Engineering 1–41. doi:10.1007/s10664-016-9451-7
- Méndez Fernández, D., Wieringa, R., 2013. Improving Requirements Engineering by Artefact Orientation, in: Heidrich J., Oivo M., Jedlitschka A., B.M.T. (Ed.), Product-Focused Software Process Improvement. PROFES 2013. Lecture Notes in Computer Science, Vol 7983. pp. 108–122. doi:10.1007/978-3-642-39259-7_11
- Näkki, P., Koskela, K., Pikkarainen, M., 2011. Practical model for user-driven innovation in agile software development, in: Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011). IEEE, pp. 1–8.
- Nawrocki, J., Ochodek, M., Jurkiewicz, J., Kopczyńska, S., Alchimowicz, B., 2014. Agile requirements engineering: A research perspective, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 40–51. doi:10.1007/978-3-319-04298-5-5
- Newman, P., Ferrario, M.A., Simm, W., Forshaw, S., Friday, A., Whittle, J., 2015. The Role of Design Thinking and Physical Prototyping in Social Software Engineering, in: 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering. IEEE, pp. 487–496. doi:10.1109/ICSE.2015.181
- Obendorf, H., Finck, M., 2008. Scenario-based usability engineering techniques in agile development processes, in: Proceeding of the Twenty-Sixth Annual CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI '08. ACM Press, New York, New York, USA, p. 2159. doi:10.1145/1358628.1358649
- Object Management Group Inc. (OMG), 2015. OMG Unified Modeling Language (v2.5).

- Olsson, H.H., Bosch, J., 2015. Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation. LNBIP 210, 154–166. doi:10.1007/978-3-319-19593-3_13
- Palmer, S.R., Felsing, M., 2001. A Practical Guide to Feature-Driven Development. Pearson Education.
- Patton, J., 2014. User Story Mapping: Discover the Whole Story, Build the Right Product, First edit. ed. O'Reilly.
- Patton, J., 2004. Interaction Design Meets Agility: Practicing Usage Centered Design on Agile Development Projects, in: Zannier, C., Erdogmus, H., Lindstrom, L. (Eds.), Lecture Notes in Computer Science. Berlin, p. 226.
- Pfeiffer, T., Hellmers, J., Schön, E.-M., Thomaschewski, J., 2016. Empowering User Interfaces for Industrie 4.0. Proceedings of the IEEE 104, 986–996. doi:10.1109/JPROC.2015.2508640
- Pohl, K., 2010. Requirements Engineering: Fundamentals, Principles, and Techniques, 1st ed. Springer Publishing Company, Incorporated.
- Ramesh, B., Cao, L., Baskerville, R., 2010. Agile requirements engineering practices and challenges: an empirical study. Information Systems Journal 20, 449–480. doi:10.1111/j.1365-2575.2007.00259.x
- Richter, M.D., Mason, J.D., Alford, M.W., Burns, I.F., Helton, H.A., 1976. Software Requirements Engineering Methodology.
- Rivero, J.M., Grigera, J., Rossi, G., Robles Luna, E., Montero, F., Gaedke, M., 2014. Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. Information and Software Technology 56, 670–687. doi:10.1016/j.infsof.2014.01.011
- Rolland, C., 1993. Modeling the requirements engineering process, in: 3rd European-Japanese Seminar on Information Modelling and Knowledge Bases Budapest, Hungary. doi:10.1.1.29.8738
- Royce, W.W., 1970. Managing the development of large software systems, in: IEEE WESCON. pp. 1–9. doi:10.1016/0378-4754(91)90107-E
- Salah, D., 2011. A framework for the integration of user centered design and agile software development processes. Proceeding of the 33rd international conference on Software engineering - ICSE '11 1132. doi:10.1145/1985793.1986017
- Salah, D., Paige, R., Cairns, P., 2016. A Maturity Model for Integrating Agile Processes and User Centred Design, in: Communications in Computer and Information Science. pp. 109–122. doi:10.1007/978-3-319-38980-6_9
- Salah, D., Paige, R., Cairns, P., 2015. Patterns for integrating agile development processes and user centred design. EuroPLOP '15 Proceedings of the 20th European Conference on Pattern Languages of Programs 1–10. doi:10.1145/2855321.2855341
- Salah, D., Paige, R.F., Cairns, P., 2014. A Systematic Literature Review for Agile Development Processes and User Centred Design Integration, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. p. 5:1–5:10. doi:10.1145/2601248.2601276
- Salinas Torrecilla, C.J., 2016. A mature Agile approach in Web Engineering contexts. University of Seville.
- Scaled Agile Inc., 2017. Scaled Agile Framework – SAFe for Lean Software and System Engineering [WWW Document]. URL <http://www.scaledagileframework.com/> (accessed 5.15.17).
- Schön, E.-M., 2016. A Methodology for Agile Requirements Engineering based on a Pattern Approach, in: Proceedings of CAISE 2016 Doctoral Consortium. Ljubljana, Slovenia, pp. 1–8.
- Schön, E.-M., Thomaschewski, J., 2017. Werkstattgespräch zum agilen Produktentwicklungsprozess. Hamburg. doi:10.13140/RG.2.2.23627.18728
- Schön, E.-M., Thomaschewski, J., Escalona, M.J., 2017a. Agile Requirements Engineering: A Systematic Literature Review. Computer Standards & Interfaces 49, 79–91. doi:http://dx.doi.org/10.1016/j.csi.2016.08.011

- Schön, E.-M., Thomaschewski, J., Escalona, M.J., 2017b. Identifying Agile Requirements Engineering Patterns in Industry by Means of Empirical Research, in: 22nd European Conference on Pattern Languages of Programs. EuroPLoP.
- Schön, E.-M., Thomaschewski, J., Escalona, M.J., 2017c. agileRE.org [WWW Document]. URL www.agileRE.org
- Schön, E.-M., Thomaschewski, J., Escalona, M.J., 2016a. Agile Requirements Engineering - Protocol for a Systematic Literature Review [WWW Document]. doi:10.13140/RG.2.2.16615.85923
- Schön, E.-M., Winter, D., Escalona, M.J., Thomaschewski, J., 2017d. Key Challenges in Agile Requirements Engineering, in: Baumeister, H., Lichter, H., Riebisch, M. (Eds.), XP 2017, LNBP 283. pp. 37–51. doi:10.1007/978-3-319-57633-6_3
- Schön, E.-M., Winter, D., Thomaschewski, J., Escalona, M.J., 2017e. Results of “Challenges in Agile Requirements Engineering” (Round 1). doi:10.13140/RG.2.2.34571.28961
- Schön, E.-M., Winter, D., Thomaschewski, J., Escalona, M.J., 2017f. Results of “Challenges in Agile Requirements Engineering” (Round 2). doi:10.13140/RG.2.2.32893.56802
- Schön, E.-M., Winter, D., Thomaschewski, J., Escalona, M.J., 2017g. Results of “Challenges in Agile Requirements Engineering” (Round 3). doi:10.13140/RG.2.2.16116.35201
- Schön, E.-M., Winter, D., Uhlenbrok, J., Escalona, M.J., Thomaschewski, J., 2016b. Enterprise Experience into the Integration of Human-Centered Design and Kanban, in: Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016). Lisbon, Portugal, pp. 133–140.
- Schön, E.-M., Escalona, M., Thomaschewski, J., 2015. Agile Values and Their Implementation in Practice. International Journal of Interactive Multimedia and Artificial Intelligence 3, 61. doi:10.9781/ijimai.2015.358
- Schrepp, M., Hinderks, A., Thomaschewski, J., 2014. Applying the User Experience Questionnaire (UEQ) in Different Evaluation Scenarios, in: Springer International Publishing (Lecture Notes in Computer Science). pp. 383–392. doi:10.1007/978-3-319-07668-3_37
- Schwaber, K., 2004. Agile Project Management with Scrum. Microsoft.
- Schwaber, K., 1997. SCRUM Development Process, in: Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G. (Eds.), Business Object Design and Implementation. London. doi:10.1007/978-1-4471-0947-1_11
- Schwaber, K., Sutherland, J., 2016. Scrum Guide [WWW Document]. URL <http://www.scrumguides.org/scrum-guide.html> (accessed 11.22.16).
- Sedeño, J., 2017. Una propuesta para el descubrimiento sistemático de servicios en fases tempranas de desarrollos ágiles. University of Seville.
- Sedeño, J., Schön, E.-M., Torrecilla-Salinas, C., Thomaschewski, J., Escalona, M.J., Mejias, M., 2017a. Modeling Agile Requirements using Context-based Persona Stories, in: Proceedings of WEBIST the 13th International Conference on Web Information Systems and Technologies. SCITEPRESS - Science and Technology Publications. pp. 196–203.
- Sedeño, J., Schön, E.-M., Torrecilla-Salinas, C., Thomaschewski, J., Escalona, M.J., Mejias, M., 2017b. Modeling Agile Requirements using Context-based Persona Stories, in: 13th International Conference on Web Information Systems and Technologies (WEBIST). pp. 196–203. doi:10.5220/0006220301960203
- Sedeño, J., Torrecilla-Salinas, C.J., Escalona, M.J., Mejías, M., 2014. An approach to transform public administration into SOA-based organizations. WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies 1, 135–142. doi:10.5220/0004794701350142
- Silva da Silva, T., Martin, A., Maurer, F., Silveira, M., 2011. User-Centered Design and Agile Methods: A Systematic Review, in: 2011 AGILE Conference. IEEE, pp. 77–86. doi:10.1109/AGILE.2011.24
- Silva da Silva, T., Selbach Silveira, M., Maurer, F., Hellmann, T., 2012. User Experience Design and Agile

- Development: From Theory to Practice. *Journal of Software Engineering and Applications* 5, 743–751. doi:10.4236/jsea.2012.510087
- Silva Da Silva, T., Silveira, M.S., Maurer, F., 2015. Usability evaluation practices within agile development. *Proceedings of the Annual Hawaii International Conference on System Sciences 2015–March*, 5133–5142. doi:10.1109/HICSS.2015.607
- Soares, H.F., Alves, N.S.R., Mendes, T.S., Mendonca, M., Spinola, R.O., 2015. Investigating the Link between User Stories and Documentation Debt on Software Projects, in: *2015 12th International Conference on Information Technology - New Generations*. IEEE, pp. 385–390. doi:10.1109/ITNG.2015.68
- Sommerville, I., Sawyer, P., 1997. *Requirements Engineering: A Good Practice Guide*, 1st ed. John Wiley & Sons, Inc., New York, USA.
- Sowizral, H.A., 1987. Design Methodology for Object-Oriented Programming, in: *OOPSLA 87 Panel Session*. pp. 91–95.
- Sutherland, J., 2001. Agile Can Scale: Inventing and Reinventing SCRUM in Five Companies. *Cutter IT Journal* 14, 5–11.
- Takeuchi, H., Nonaka, I., Takeuchi, H., 1986. The new new product development game. *Harvard Business Review* 64, 137–146. doi:10.1016/0737-6782(86)90053-6
- The LeSS Company B.V., 2017. LeSS Framework - Large Scale Scrum (LeSS) [WWW Document]. URL <https://less.works/less/framework/index.html> (accessed 5.15.17).
- Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., 2015. Estimating, planning and managing Agile Web development projects under a value-based perspective. *Information and Software Technology* 61, 124–144. doi:10.1016/j.infsof.2015.01.006
- Torrecilla-Salinas, C.J., Sedeño, J., Escalona, M.J., Mejías, M., 2013. Using an Agile framework to deliver e - Government services in public administrations, in: *21nd Annual Software Quality Management (London, 2-5, Sep, 2013)*. SQM2013.
- VersionOne, 2016. *10th Annual State of Agile Report*. doi:10.1093/jicru/ndl025
- VersionOne Inc., 2017. *11th Annual State of Agile Report*.
- Wagner, S., Méndez Fernández, D., Felderer, M., Kalinowski, M., 2017. Requirements Engineering Practice and Problems in Agile Projects: Results from an International Survey. *CibSE - 20 Ibero-American Conference on Software Engineering*.
- Wanderley, F., Silva, A., Araujo, J., Silveira, D.S., 2014. SnapMind: A framework to support consistency and validation of model-based requirements in agile development, in: *2014 IEEE 4th International Model-Driven Requirements Engineering Workshop (MoDRE)*. IEEE, pp. 47–56. doi:10.1109/MoDRE.2014.6890825
- Wellhausen, T., Fiesser, A., 2011. How to write a pattern?: a rough guide for first-time pattern authors. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*, New York, NY, USA: ACM (p. 5:1-5:9). doi:10.1145/2396716.2396721
- Williams, L., 2012. What agile teams think of agile principles. *Communications of the ACM* 55, 71–76. doi:10.1145/2133806.2133823

Word	Abbreviation	Description
Agile	-	Umbrella term for a mindset which is based on agile values, agile methodologies and agile techniques and tools
Agile RE activity	-	Activity that is carried out in terms of RE in an agile environment (see Table 1)
Agile RE pattern	-	Combination of agile RE problem and agile techniques, with which the problem can be solved
Agile RE problem	-	Problem that occurs in an agile environment in terms of RE
Agile team	-	Product development team including roles like developer, UX designer, tester, Agile Coach, Scrum Master or Product Owner
Agile Requirements Engineering	Agile RE	Requirements Engineering tailored to an agile environment
Artifact	-	Output that is generated while using a method.
Context of use	-	Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used. (International Organization for Standardization, 2010)
Domain	-	Specific area or subject that the product is being developed for.
Functional requirement	FR	Describing features of an application or a component, which support the user in carrying out tasks.
Human-Centered Design	HCD	Refers to ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems (International Organization for Standardization, 2010)
Impact	-	Effect or influence resulting from a cause
Meeting	-	People come together for a specific purpose (often time-boxed)
Method	-	Procedure that is used in order to achieve an objective
Methodology	-	A system of methods used in a particular area of study or activity
Navigational Development Techniques	NDT	Model-driven approach to deal with requirements in Web systems (Escalona and Aragon, 2008)
Non-Functional requirement	NFR	Describing constraints of a system, which affect different functional requirements, for instance security, performance, reliability.
Organizational environment	-	Surroundings or conditions in which the product development takes place.
Requirements Engineering	RE	Combination of all activities concerning discovering, documenting, and maintaining a set of requirements
Role	-	Description of tasks and responsibilities that are related to a specific person

Stakeholder	-	Individual or organization having a right, share, claim or interest in a system or in its possession of characteristics that meet their needs and expectations (International Organization for Standardization, 2010), for instance management, sales, marketing, or customer
System	-	Combination of hardware, software and/or services
User	-	Person who interacts with the product. (International Organization for Standardization, 2010)
User-Centered Design	UCD	Refers to ISO 13407:1999 - Human-centred design processes for interactive systems (International Organization for Standardization, 1999)

Objectives

Agile Software Development (ASD) is used to cope with increasing complexity in system development. Hybrid development models, with the integration of User-Centered Design (UCD), are applied with the aim to deliver competitive products with a suitable User Experience (UX). Therefore, stakeholder and user involvement during Requirements Engineering (RE) are essential in order to establish a collaborative environment with constant feedback loops. The aim of this Systematic Literature Review (SLR) is to investigate what approaches exist to involve stakeholder in the process, which methodologies are accepted to present the user perspective and how requirements management is been carried out.

Our main goal is to clarify the state of the art in the field of Agile Requirements Engineering (RE) in order to identify the existing basis for the research of the PhD thesis. In addition, we want to find out where the research fits into the current body of knowledge. To this end, we are going to conduct an SLR following appropriate guideline by Kitchenham and Charters (Kitchenham and Charters, 2007).

Research Questions

To define the aims and the scope of this SLR, we created three research questions (RQ) with the following sub-criteria:

RQ1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

On one hand, we analyzed whether the existing approaches involve stakeholders and users directly into the development process. On the other hand, our aim was to study whether the approaches apply a process model for the involvement. In addition, we queried what kind of methods they use in order to gather data. With regard to the agility of the existing approaches, we examined whether there are iterations along the development process.

RQ2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

Concerning this RQ, we analyzed the included studies in terms of methodologies that are used to handle the user perspective within agile environments. Furthermore, we investigated how the knowledge of user requirements is shared among stakeholders.

RQ3: What are the common ways for requirements management in ASD?

In terms of this RQ, our aim was to investigate what types of artifacts are used and how they are utilized. Moreover, we wanted to discover whether the documentation of requirements is understandable without further knowledge in order to be able to work in a collaborative manner. In addition, we examined the treatment of non-functional requirements.

Search Process

- Manual search process with search string
- Adapt search string for every digital library

- Use reference search (Snowballing)
 - Forward snowballing (search in papers that cited the paper)
 - Backward snowballing (search in the reference list of the paper)
- Search process is divided into the following phases:

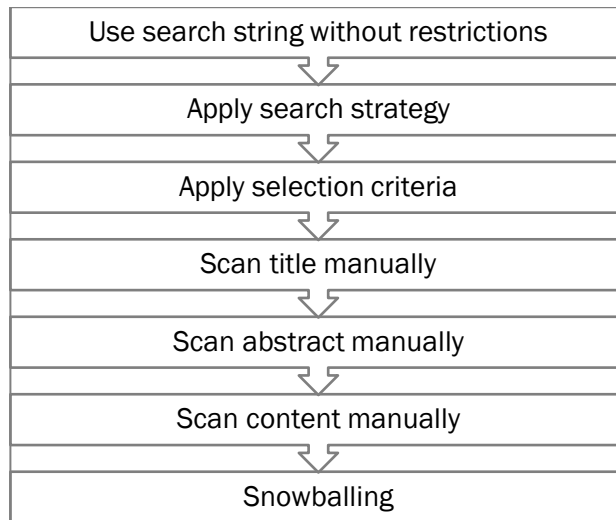


Figure 52: Search process

Search Space

The following digital libraries are taken into account:

Table 30: Search space

Digital library	Search strategy
Google scholar	full text
Science direct	abstract, title and keywords
SpringerLink	abstract and keywords
Scopus	abstract, title and keywords
IEEEXplore	abstract and keywords
ACM	abstract and keywords

Search Criteria

- Define keywords for search
- Add keywords from any relevant papers we already have
- Identify alternative spellings and synonyms
- Since the search process is a critical aspect, we optimized the search keywords iteratively (define → test → optimize)
- Full list is shown in Table 31

Table 31: Full list of search keywords

Topic	Search Keywords
Agile	agile, agile development, agile software development, agile method*, agile practice, agile approach, agile project, agile lifecycle, agile software engineering,
	Scrum, Kanban, extreme programming, lean startup, lean development, feature driven development, feature-driven development, agile unified process, rational unified process, Crystal Clear
Human Computer Interaction	human computer interaction, hci, human-computer interaction, computer-human interaction, hmi, human-machine interaction, machine-human interaction, human system interaction, human interaction
	user centered design, user-centered design, ucd, human centered design, human centered software engineering, usage-centered design, service design, customer experience, user centered development, experience design, interaction design, goal-directed design, design thinking
	Usability, usability evaluation, usability engineering, usability method, user experience, joy of use
	User, end user, end-user, consumer, costumer, participation, user involvement, user interface, participatory design, innovation design
	human factor
Requirements Engineering	requirements engineering, requirement, user-centered requirements, documentation, requirements documentation, agile documentation

Search String

Since every digital library has its own characteristics concerning its search engines, we have to adapt the search string for every library. To this end, we create a short and a long form of the search string.

ST = [Agile] + [Human Computer Interaction] + [Requirements Engineering]

Search String (short form)

(agile OR scrum OR kanban OR “extreme programming” OR lean) AND (hci OR hmi OR ucd OR usability OR human OR user) AND (“requirements engineering”)

Search String (long form)

((agile OR “agile development” OR “agile software development” OR “agile method*” OR “agile practice” OR “agile approach” OR “agile project” OR “agile lifecycle” OR “agile software engineering” OR scrum OR kanban OR “extreme programming” OR “lean startup” OR “lean development” OR “feature-driven development” OR “agile unified process” OR “rational unified process” OR “Crystal Clear”) AND (“human-computer interaction” OR hci OR “computer-human interaction” OR hmi OR “human-machine interaction” OR “machine-human interaction” OR “human system interaction” OR “human interaction” OR “user-centered design” OR ucd OR “human centered design” OR “human centered software engineering” OR “usage-centered design” OR “service design” OR “customer experience” OR “user centered development” OR “experience design” OR “interaction design” OR “goal-directed design” OR “design thinking” OR usability OR “usability evaluation” OR “usability engineering” OR “usability method” OR “user experience” OR “joy of use” OR user OR “end-user” OR

consumer OR costumer OR participation OR “user involvement” OR “user interface” OR “participatory design” OR “innovation design” OR “human factor”) AND (“requirements engineering” OR requirement* OR “user-centered requirements” OR documentation OR “requirements documentation” OR “agile documentation”))

Study Selection Criteria

Table 32: Inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
papers written in English	no full books
papers published in between 1995-2015;	papers whose full text were not available
papers under peer review	papers with results that had been already published
papers presenting approaches to integrate user into agile development processes	papers that were not focused on agile development
papers related to Agile RE	papers only presenting ideas, lessons learnt, recommendations or guidelines
papers associated with agile requirements documentation.	papers introducing tools whose underlying methodology was not comprehensibly described (black box)
specific book chapters	studies, whose primary focus moved away from agile methodologies

Study Selection Process

In the following it will be described how the selection criteria will be applied:

- Study should be included if the first 3 inclusion criteria can be applied (Yes)
- Study should be excluded if any exclusion criteria can be applied (No)
- If there are doubts with available information then
 - Read full text
 - Ask other researcher for opinion

Quality Assessment

We develop a quality checklist to assess the individual studies using a 3 point Likert scale (see Table 33).

Table 33: Quality assessment criteria

Item	Assessment Criteria	Score	Description
QA1	Is the proposal validated?	-1	No, it is not validated
		0	Partially, it is validated in a laboratory or only parts of the proposal are validated
		1	Yes, by a case study.
QA2	Does the study present a detailed description of the approach?	-1	No, details are missing
		0	Partially, if you want to use the approach, you need to read the references
		1	Yes, the approach can be used with presented details
QA3	Does the study present a personal opinion piece or viewpoint?	-1	Yes, it does.
		0	Partially, since related work is explained and paper is set into a specific context
		1	No, the paper is based on research
QA4	Has the study been cited by other authors?	-1	No, no one cited the study
		0	Partially, between 1-5 articles cited the study
		1	Yes, more than 5 articles cited the study
QA5	Includes the paper a clear statement of the aims of the study?	-1	No, aims are not described.
		0	Partially, aims are described but unclearly
		1	Yes, aims are well described and clear

Data Collection

A data collection form will be set up in an Excel sheet and used for the data extraction (see Table 34).

Table 34: Data Collection Form

Key Aspect	Attributes
Basic information	Title, Authors, Publication date, DOI, Paper ID
Publication data	Journal, conference, date (of conference) publisher, volume, issue, pages, keywords, abstract
Publication type	Research method (Experiment, quasi-experiment, lessons learnt, case study, opinion survey, tertiary study, other?)
Agile method	e.g. Scrum, XP, Kaban
HCI method	advantages, disadvantages
Short summary	contribution, ideas, concepts
Assigned to RQ	e.g. RQ1, RQ3
Research approach	Deductive or inductive
Artifacts	e.g. User Stories, Personas, prototypes
Short summary	Summary of the paper
Results/Contributions	Summary of the contribution
Personal Assessment	Comments from the author, related to the selection status (included, excluded)

Data Analysis

In the following, we describe how the research questions (see 0) will be answered.

RQ1: What approaches exist, which involve stakeholders in the process of RE and are compatible with ASD?

We will present a table of existing approaches used for stakeholder and user involvement in agile projects. The table will include a number of papers in which the concept is proposed, advantages and disadvantages. In addition, a discussion about the approaches will be provided.

RQ2: Which agile methodologies, which are capable of presenting the user perspective to stakeholders, can be found?

We will present a table of existing approaches, methods, artifacts used to declare the requirements in agile projects. The table will include a number of papers in which the concept is proposed, advantages and disadvantages. In addition, we will provide a list of methodologies, which are used in Agile to present the user perspective.

RQ3: What are the common ways for requirements management in ASD?

We will present a table of existing approaches, methods, artifacts used to declare the requirements in agile projects. The table will include a number of papers in which the concept is proposed, advantages and disadvantages. In addition, we will discuss how requirements are documented.

Results Report Round 1 – Original Comments Stated by Experts

We will present a table of existing approaches, methods, artifacts used to declare the requirements in agile projects. The table will include a number of papers in which the concept is proposed, advantages and disadvantages. In addition, we will discuss how requirements are documented.

Version	Trigger	Summary of changes	Reason for changes
1.0	Initial version	Initial version	Initial version
1.1	Learnings from Doctoral Consortium @CAISE	Add relation between System and ContextOfUse	Completeness in terms of ISO 9241-210:2010
		Extend attributes in metaclass Stakeholder	Completeness
		Extend attributes in metaclass Methodology	Completeness
		Extend attributes in metaclass System	Completeness
1.2	Discussion with JS	Metaclass Domain: add attribute "regulations"	There are differences concerning the legal or regulatory requirements within each domain
		Metaclass REActivity: change name to AgileREActivity	Defined activities for agile RE are different compared to activities known from established RE approaches
		Metaclass System: add attribute infrastructure	A system is embedded into an infrastructure
1.3	Discussion with JD	Metaclass User: change attribute type pain point from string to array	Complexity of pain point
		Metaclass System: delete attribute user	User is part of metaclass ContextOfUse and metaclass System is part of the metaclass ContextOfUse
		Add relation between metaclass Stakeholder and metaclass AgileTeam (0..*)	Stakeholder can be part of the agile team, see Lean UX (Gothelf and Seiden, 2012)
		Metaclass AgileTeam change attribute member from string to object	Member is a complex attribute like "user"
1.4	Internal review	Add relation between metaclass Methodology and metaclass AgileREPattern (0..*)	agile RE pattern can be part of a methodology (e.g. Sprint Planning, Product Backlog, Kanban board)

	Learnings from study “Key Challenges in Agile RE”	Metaclass AgileREProblem delete attribute tag	No categorization of problems is necessary
		Metaclass AgileREPattern change attribute category from string to object	Category is an object from the metaclass agileReProblem
		Metaclass AgileREPattern change order of attributes according to template for agile RE pattern on agileRE.org	Conformity to template in web application agileRE.org
1.5	Learnings from study “Identifying Agile RE Pattern”	Metaclass AgileREPattern change attribute category to problem (object)	Conformity to template in web application agileRE.org and agile RE Pattern
		Metaclass AgileREPattern: add attribute context (string)	Conformity to template in web application agileRE.org and agile RE Pattern
1.6	Discussion with MJ, JT	Metaclass ContextOfUse: delete attribute user	It is implicitly included due to the association 1...*
		Start name of all metaclasses with upper case	Conformity to UML notation
		Metaclass Domain: change type of regulation from array to complex type	Conformity to UML notation
		Metaclass System: change type from feature and infrastructure to complex type	Conformity to UML notation, objects are not displayed in metamodels
		Metaclass Environment: delete objects from other classes	It is implicitly included due to the association 1...*
		Add metaclass Impact and delete corresponding object from metaclass Methodology	Impact is a complex type, which is filled during runtime
		Metaclass AgileREPattern: change type from example and template from object to complex type	Conformity to UML notation, objects are not displayed in metamodels
		Metaclass User: change attribute pain point from string to complex type	Conformity to UML notation, arrays are not data type for metamodels
1.7	Discussion with NK, MJ, JT	Add generalization between metaclass User and metaclass Stakeholder, so that user is a specialized form of stakeholder	In RE, user is a special type of stakeholder
		Add metaclass Requirement as composition to metaclass Environment	Requirement is an essential concept in agile RE and therefore needs a metaclass
		Metaclass Stakeholder: delete attribute requirement	Is added as additional metaclass
		Metaclass AgileTeam: add role	Member of an agile team have different roles

V1.8	Discussion with NK, MJ, JT	Define metaclass Requirement as external class	The detailed description of how requirements are handled and described are out of scope of this PhD thesis since it is part of other works (e.g. NDT)
		Update of relations and multiplicities,	
		Change ordering of metaclasses,	
		Metaclass Requirement: delete attribute requester	Due to the changes of relations some information was redundant

MJ	Dr. María José Escalona
JT	Dr. Jörg Thomaschewski
JD	Jutta Doetkotte
JS	Dr. Jorge Sedeño López
NK	Dr. Nora Koch

Results Report Round 1 – Original Comments Stated by Experts

- Involvieren von Stakeholder und Nutzer -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
5	Zu Beginn fehlende Style-Guides, Prototypen, etc. sind oftmals Ursache für eine schlechte UX, da sich in der agilen Softwareentwicklung Entwicklungsstränge meist unabhängig voneinander laufen und jeder Entwickler seine eigene (oder gar keine) Vorstellung von guter UX hat.	Eine schlechte/nicht einheitliche UX führt zu Unzufriedenheit beim Kunden und den End-Nutzern des Produktes.
7	Ständige Rückkoppelung mit Stakeholdern bei Änderungen	In klassischen Organisationen wurden Anforderungen in vorgelagerten Phasen erhoben und abgestimmt und dann häufig nicht mehr angepasst. Die daraus resultierenden Denkweisen im Umgang mit Anforderungen herrscht in vielen Organisationen noch vor. Das führt dazu, dass ein PO einerseits häufig unter Zeitdruck Anforderungen erheben und abstimmen muss und zudem erheblichen Aufwand hat, um seine Stakeholder im agilen Prozess nicht abzuhängen.
13	Integration der Stakeholder	Selten hat man den Fall, dass es nur einen Stakeholder gibt. Stakeholder haben verschiedenste Interessen die sich im Zeitverlauf auch schon mal ändern. Ein guter Weg zwischen den Verschiedenen Bedürfnisse und dem was den Erfolg des Produktes/Projekts ausmacht ist nicht einfach zu finden. Regelmäßiges und qualitatives Feedback hilft, bei der effektiven und effizienten Entwicklung.
14	Stakeholdermanagement	Agiles Arbeiten ist nur das besser und erfolgreich, wenn enger und intensiver mit den Stakeholdern zusammengearbeitet wird, damit beispielsweise Feedbackzyklen wertvoll werden.
15	Anwenderorientierte Anforderungserhebung ist in agilen Prozessen genauso schwierig wie bei Wasserfall. Auch bei einer agilen Vorgehensweise muss bei der Entwicklung von B2B-Software für eine strikte Orientierung an den Bedürfnissen der Kunden und den Einbezug dieser geworben werden.	Fehlende Kundenorientierung wirkt sich negativ auf die Produktqualität aus.

16	Das Ausliefern von Zwischenlösungen	Häufig sträuben sich Beteiligte MVP oder Zwischenschritte zu akzeptieren. Dies kann zu Spannungen bzw. zu viel Erklärungsbedarf führen.
17	Anforderungen spiegeln gemeinhin einen nicht sehr kreativen Umgang mit Problemen wieder.	Gute Produkte entstehen nur unter Einbeziehung von Benutzern, nicht nur Systemen. Benutzeranforderungen und Nutzungsqualitäten herauszuarbeiten ist entscheidend für gute Produkte und ist etwas, was in der Anforderungsanalyse gerne hintenangestellt wird.
19	Kunden miteinbeziehen und Vorteile herausarbeiten.	Kunden sind oft zunächst skeptisch, wenn sie keine umfangreichen Pflichtenhefte und exakte Projektpläne mit definierten Meilensteinen bekommen, die letztlich ohnehin nicht der Realität entsprechen. Hier sollte wenn möglich der Kunde in die Prozesse des Requirement-Engineering und innerhalb von Scrum in die Reviews wenn möglich eng eingebunden werden.
22	Die kontinuierliche Einbeziehung der Stakeholder in den Entwicklungsprozess anstelle der Übergabe eines "fertigen" Konzepts an die IT, die das Ergebnis dann zum Stichtag liefern muss.	Ohne regelmäßiges Feedback von Stakeholdern, Kunden, Märkten, gibt es keine passende und kleine Lösung.
24	Distance to business people	The purpose of agile is to produce business value asap with minimum feature list. When business objectives are not clear or considered, it leads to a kind of projects like agile fixed price/scope. Only mechanics of agile requirement engineering will exist: methods, tools..etc. So, Waterfall/Scrum models result. I do have not very positive experience with such projects.
24	Stakeholders affected by requirements or changing the system are not involved	In one of my projects representatives of end users did not really knew the pain of end users. Even the early UI prototypes are tested by incorrect stakeholders, which lead to risks of conflicts and failure.
24	Lack of skill/knowledge/experience with agile way of specifying and managing requirements.	This leads to lower quality of documenting requirements though much effort is invested. Then after requirements are finally documented, the changes are not documented properly. Also, tracking changes in case of fixed price projects is not easily possible, because tools are either not known or not mature enough. Communicating requirements depends mostly on documents and less on face to face conversation. This increases risks of overseeing important expectations or challenges.
26	Die agilen Methoden insbesondere Scrum geben mit dem Wert „Working software over comprehensive documentation“ den source code Vorrang zu abgestimmten Anforderungen.	Dies bedeutet, dass insbesondere in agilen Projekte neue Kommunikationsstrukturen implementiert werden müssen und diese auch von alle Beteiligten befolgt werden müssen.

- Zusammenarbeit im Team -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
7	Verständnis der Verhandelbarkeit von Anforderungen	Seitens der Stakeholder und auch seitens eines PO fehlt häufig das Verständnis dafür, dass Anforderungen mit dem Team verhandelbar sein müssen. Zudem fehlt auch unreifen Dev-Teams die Erfahrung im Verhandeln der Anforderungen. Im Ergebnis werden teilweise wieder nicht-optimale Lösungen implementiert.
7	Koordination mit anderen Teams	Bei fachlichen oder technischen Abhängigkeiten zu anderen Teams kann erheblicher Koordinationsaufwand entstehen, um diese aufzulösen. Das betrifft sowohl die Anforderungen als auch die technische Lösung, die implementiert wird.
9	Die richtigen Entwickler finden	Gerade bei einem agilen Ansatz steigt und fällt die Software-Qualität mit dem Kompetenz und Motivation der Entwickler
18	Die beteiligten Teams sollten einer möglichst geringen Fluktuation unterliegen.	Mit steigender Teamreife wird das Team erst effizient (siehe Modell von Tuckman - forming, storming, norming, und performing)
18	Erfolge gemeinsam feiern und ein gesundes positives Arbeitsklima schaffen, um die Teambildung und Zusammenarbeit über die Teamgrenzen hinaus zu fördern.	Kommunikation ist der wichtigste Bestandteil in der Zusammenarbeit. Nur mit einer guten und offenen Kommunikation werden Konflikte unterbunden oder frühzeitig sichtbar. Die gesparte Energie kommt der Feature-Entwicklung zugute.

- Vision und Big Picture -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
14	Kontinuierliche Veränderung benötigt eine gute Produkt- und Zielvision damit keine sprichwörtlichen Flickenteppiche entstehen	siehe oben.
18	Eine gute verständliche Unternehmens-Vision, idealerweise mit passender Mission.	Langfristige Motivation erfolgt durch die Identifizierung jedes einzelnen mit dem geschaffenen Produkt. Das kann aber nur erfolgen, wenn jeder beteiligte die "Marschrichtung" (Big-Picture) verstanden hat.

- Iterationsplanung und Aufwandschätzung -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
1	Die eine Referenzstory zu finden, um diese zu Schätzen und sie als Masseinheit für die Storypunkte im Projekt zu nutzen.	Sehr wichtiger Bestandteil des Agilen Prozesses, das oft vernachlässigt wird. Ohne Referenzstory misst man meist in Stunden/Tagen Aufwand, was sich von der Scrum-Theorie weiter weg bewegt.
1	Schätzung einer Story in Storypoints und nicht in Stunden.	Bei Schätzungen in Stunden gerät man leicht in Versuchung verfügbare Ressourcen eines Teams direkt mit der Summe geschätzter Stunden zu vergleichen, sodass die Chancen die Velocity eines Teams von Sprint zu Sprint erheblich zu steigern, sehr gering sind.
6	Auftraggeber muss akzeptieren, dass der Zeitraum bis zu einer ersten lauffähigen Version und bis zu einer Marktreife nicht bei Projektbeginn zu beziffern ist.	Der Glaube vieler Entscheider, Software-Entwicklung lasse sich nach dem "über-den-Daumen-Prinzip" schätzen, ist sehr ausgeprägt.
7	Erhebung von Anforderungen abgestimmt auf Sprint-basierte Entwicklung	Die Sprint-basierte Entwicklung führt dazu, dass ein PO zum Sprintbeginn (Planning) ausreichend Stories im Status "ready" haben muss, um den Sprint sauber planen zu können. Mit der parallelen Betreuung des laufenden Sprints kann dadurch erheblicher Zeitdruck entstehen.
11	Anforderungen sollten rollierend geplant werden; d.h. die nächsten anstehenden Sprints müssen sehr konkret im Backlog beschrieben sein, die Anforderungen der nachfolgenden Sprints können grob bleiben. .	Nur so kann gewährleistet werden, dass man auch agil auf weiteren Erkenntnisgewinn reagieren kann. Die Herausforderung dabei ist, das Team auf diese Art und Weise auch kontinuierlich mit Anforderungen zu versorgen.
12	Fokussierung nur auf die Anforderungen der nächsten Iteration	Oftmals werden viele Ideen genannt, die jedoch für die unmittelbar folgende Iteration nicht von belang sind und somit die Aufmerksamkeit ablenken von den gerade wichtigen Anforderungen
12	Ausblick auf die nächsten 3-5 Iterationen darf nicht als verbindlich angesehen werden	Ein Ausblick auf die nächsten Iterationen ist für die Abstimmung mit angrenzenden Projekten / Gewerken wichtig, jedoch muss allen Beteiligten bewusst sein, dass man im Sinne der Agilität die tatsächlichen Aufgaben der nächsten Iterationen den dann vorliegenden Gegebenheiten noch anpassen muss.
14	Richtige Priorisierung nach diversen Gesichtspunkten wie Dringlichkeit, Geschäftswert, technischer Abhängigkeit...	Es müssen verschiedenste auf die Priorisierung einwirkende Faktoren bewertet und berücksichtigt werden.
16	Zeitmanagement für Weiterentwicklung gegenüber Bugfixing	Häufig kommt es bei Scrum - Teams zu Problemen beim schützen des Sprints wenn die Bugfixes zu häufig vorkommen. Lösungen wie Developer of the Week o.Ä. werden dann geschaffen um dieses Problem zu lösen. Das benötigt aber ein kleines Team, oder eine gute Verteilung des Wissens.

21	Nicht immer sind die umzusetzenden Anforderungen zum Entwicklungsstart klar definiert, da sie im Vorfeld nur kurz dokumentiert wurden, und die Prioritäten gerade nicht auf diesen Anforderungen lagen. Dies liegt in der fehlenden Planung/Vorbereitung der Ziele der nächsten Sprints und Releases. Oft verändern sich da die Prioritäten, und unklare Anforderungen müssen schnell umgesetzt werden.	In der Praxis sind Prio-Verschiebungen normales Tagesgeschäft.
----	---	--

- Größe und Granularität von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
1	Alle Stories zu einem Projekt aus einer Hand (dem Product Owner) zu bekommen.	Meist sind Projekte doch komplexer als dass sie von einem PO insgesamt erfasst werden können. Stories zu schreiben und diese zu priorisieren kann ein PO meist nicht wirklich objektiv, wenn diese nicht von einer Person erfasst worden sind.
2	Bei Scrum: Anforderungen müssen "klein" genug sein, um von einer Person in einem Sprint erledigt werden können.	Wenn das nicht gegeben ist und die Anforderung "geschnitten" wird, ergeben sich viele Abhängigkeiten :(
4	Anforderungen in der Agilen Softwareentwicklung müssen so geschnitten sein, dass sie in einem Sprint (< 4 Wochen) umsetzbar sind	Zeiträume von über einem Monat wären nicht mehr agil
9	Bei kleinen Teams und kurzen Sprints User Stories so klein zu schneiden, dass sie in einen Sprint passen ohne technisch zu werden	User Stories sollten fachlichen Mehrwert liefern und innerhalb eines Sprints umgesetzt werden können
13	Das Finden eines passenden Detailgrades in der Anforderungen (allgemein und auf der Zeitachse betrachtet)	Zu viele Details können die Beteiligten verwirren, demotivieren und erschweren den Blick für das Wesentliche. Zu wenige Details können ebenfalls zu Defokussierung führen. Das Finden eines für
21	Nicht immer sind die umzusetzenden Anforderungen zum Entwicklungsstart klar definiert, da sie im Vorfeld nur kurz dokumentiert wurden, und die Prioritäten gerade nicht auf diesen Anforderungen lagen. Dies liegt in der fehlenden Planung/Vorbereitung der Ziele der nächsten Sprints und Releases. Oft verändern sich da die Prioritäten, und unklare Anforderungen müssen schnell umgesetzt werden.	In der Praxis sind Prio-Verschiebungen normales Tagesgeschäft.
21	Schnelllebigkeit und schnelle Umsetzung lassen nur sehr selten Zeit, gezielt	Oberflächliche Anforderungen lassen viel Spielraum für die Umsetzung offen.

	Anforderungen zu definieren. Oft bleiben diese oberflächlich.	
21	Anforderungen teilweise so komplex und nicht weiter herunterbrechbar, dass sie sich nicht innerhalb eines Sprints umsetzen lassen.	Erwartung des Managements, dass nach Sprints an Bausteine "Haken gesetzt" werden können
23	Vollständigkeit der Anforderungen Part 1	Anforderungen an eine komplexere Funktion sind häufig über mehrere Sprints verteilt. Es ist wichtig das Gesamtbild zu kennen und den Überblick zu behalten und diesen dann alle die Entwickler weiterzugeben. Ansonsten werden evtl. Funktionen vergessen, oder widersprechen sich sogar. Es gibt kein Pflichtenheft aus man den Gesamtkontext aller Anforderungen und damit auch ein Gesamtbild ableiten kann.

- Abhängigkeiten von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
4	Anforderungen in der Agilen Softwareentwicklung müssen möglichst wenige externe Abhängigkeiten enthalten	Je mehr Abhängigkeiten desto größer das Risiko, dass das Team die Anforderung nicht aus eigener Kraft umsetzen kann
7	Koordination mit anderen Teams	Bei fachlichen oder technischen Abhängigkeiten zu anderen Teams kann erheblicher Koordinationsaufwand entstehen, um diese aufzulösen. Das betrifft sowohl die Anforderungen als auch die technische Lösung, die implementiert wird.
9	Die Unabhängigkeit der Anforderungen (z. B. User Stories) gewährleisten	Abhängigkeitsfreie Anforderungen erlauben ein unkompliziertes Umsortieren der Anforderungen ohne großen Kommunikationsaufwand im Team
14	Richtige Priorisierung nach diversen Gesichtspunkten wie Dringlichkeit, Geschäftswert, technischer Abhängigkeit...	Es müssen verschiedenste auf die Priorisierung einwirkende Faktoren bewertet und berücksichtigt werden.
16	Koordination von parallelen Strängen, bzw. Abhängigkeitsmanagement.	Teilweise gibt es sogar bei kleineren Projekten Abhängigkeiten, die dafür sorgen, dass man sich dem Wasserfall annähert. Hier gilt es dann eine Balance zu finden, die das Prototypisieren, Entwickeln, Testen und Ausliefern ohne Unterbrechungen bzw. Leerlauf erlaubt.
21	Fehlende Zeit für die Prüfung der Anforderungen im Bezug auf Wechselwirkung mit anderen bereits implementierten Anforderungen.	Im Security-Umfeld bedingen sich durchaus einige Anforderungen, schließen sich aus oder die eine hebt die andere auf. Um einen Ausschluss vermeiden zu können, muss ausreichend Zeit zum Prüfen da sein.

23	Vollständigkeit der Anforderungen Part 1	Anforderungen an eine komplexere Funktion sind häufig über mehrere Sprints verteilt. Es ist wichtig das Gesamtbild zu kennen und den Überblick zu behalten und diesen dann alle die Entwickler weiterzugeben. Ansonsten werden evtl. Funktionen vergessen, oder widersprechen sich sogar. Es gibt kein Pflichtenheft aus man den Gesamtkontext aller Anforderungen und damit auch ein Gesamtbild ableiten kann.
----	--	---

- Verständnis von Agile und Agilen Werten -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
6	Das Team muss die Freiheit besitzen, Aufgaben in einem festgelegten Zeitraum abzuarbeiten.	Sofern das Team neben der Entwicklungsarbeit im Rahmen von Projekte noch Regelaufgaben hat, ist diese Vorgabe mitunter nicht einzuhalten.
7	Verständnis der Verhandelbarkeit von Anforderungen	Seitens der Stakeholder und auch seitens eines PO fehlt häufig das Verständnis dafür, dass Anforderungen mit dem Team verhandelbar sein müssen. Zudem fehlt auch unreifen Dev-Teams die Erfahrung im Verhandeln der Anforderungen. Im Ergebnis werden teilweise wieder nicht-optimale Lösungen implementiert.
16	Abbauen von technischen Schulden	Zu häufig werden gerade in agilen Prozessen auch Prototypen lange im Betrieb gehalten weil diese zuerst einmal ihr soll erfüllen. Meistens führt das später zu komplexen Auflösungsversuchen, die dann die Softwareentwicklung noch länger aufhalten als diese frühzeitig zu lösen.
17	Die klassische Arbeit mit Anforderungen (große Anforderungsanalyse am Anfang, Auflistung, Schätzung, Priorisierung, Testbarkeit) führt manchmal zu unagilem wasserfallartigen Vorgehen auch wenn man eigentlich vor hat agil vorzugehen.	Agil arbeiten heißt iterativ vorgehen. Es sind also gerade nicht alle Anforderungen am Anfang bekannt, sondern es sollte immer möglich sein, Dinge ändern bzw. optimieren zu können.
18	Das Unternehmen (Fachbereiche + Management) muss die agile Entwicklung verstanden haben und im Alltag auch "leben".	Grundsätzlich hängt davon der Erfolg bzw. die Effizienz des gesamten Vorgehensmodells ab.
19	Bestehende Konzernstrukturen überzeugen.	Ohne Mitarbeit der Konzernstruktur werden agile Vorgehensmodelle korrumpiert und/oder als Kontrollinstrumente missbraucht.
19	Mitarbeiter überzeugen.	Es gibt immer noch sehr viele Vorbehalte gegenüber agilen Vorgehensmodellen. Nicht nur in der Führung, sondern auch innerhalb eines agilen Teams. Es werden von beiden Seiten oft klare Anordnungen vermisst. Hier ist viel Überzeugungsarbeit nötig.

19	Agile Vorgehensmodelle müssen "straight from the book" eingeführt werden.	Die Mehrzahl aller Firmen führt nur Teile von Scrum oder ähnlichen Modellen ein oder passt diese von vorne herein an. Rollen werden umdefiniert, Meetings teilweise als obsolet angesehen und nicht durchgeführt (meist Retrospektive - obwohl eines der wichtigsten Instrumente). Daily Standups verkommen zu Rapportmeetings oder verlieren sich in Detaildiskussionen. Wenn die Methoden nicht mit ihren wenigen Regeln exakt durchgeführt werden, sind sie von vorne herein zum Scheitern verurteilt.
26	Die agilen Methoden insbesondere Scrum geben mit dem Wert „Working software over comprehensive documentation“ den source code Vorrang zu abgestimmten Anforderungen.	Dies bedeutet, dass insbesondere in agilen Projekte neue Kommunikationsstrukturen implementiert werden müssen und diese auch von alle Beteiligten befolgt werden müssen.

- Kontinuierliche Lieferung von Mehrwert -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
4	Anforderungen in der Agilen Softwareentwicklung müssen zum Sprintende ein nutzbares Inkrement ergeben	Steht so im Scrum Guide ;-)
11	Agil heißt, dem Kunden immer einen funktionsfähigen Stand zu liefern, der über die Zeit weiter im Umfang wächst. Hier die richtige Strategie zu finden ist schwierig und meiner Meinung nach eine der größten Verständnisprobleme von Leuten, die vorher Wasserfall gemacht haben. Henrik Kniberg hat das auf der Lean Kanban 2014 m.E. sehr schön erklärt: https://vimeo.com/116229546 ab 18:15	auch hier: nur so kann Agil wirklich funktionieren.
24	Distance to business people	The purpose of agile is to produce business value asap with minimum feature list. When business objectives are not clear or considered, it leads to a kind of projects like agile fixed price/scope. Only mechanics of agile requirement engineering will exist: methods, tools..etc. So, Waterfall/ Scrum models result. I do have not very positive experience with such projects.

- Rollen und Verantwortungsbereiche -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
1	Alle Stories zu einem Projekt aus einer Hand (dem Product Owner) zu bekommen.	Meist sind Projekte doch komplexer als dass sie von einem PO insgesamt erfasst werden können. Stories zu schreiben und diese zu priorisieren kann ein PO meist nicht wirklich objektiv, wenn diese nicht von einer Person erfasst worden sind.
2	Bei Scrum: Rolle des PO	PO muss priorisieren - sehr herausfordernde Rolle
6	Stakeholder müssen akzeptieren, dass viele Entscheidungen beim Team liegen.	Viele Entscheider, welche als Stakeholder in Projekte einbezogen werden, möchten auch bei Details mitentscheiden.
7	Laufende Erhebung der Anforderung	In klassischen Organisationen wurden Anforderungen in vorgelagerten Phasen erhoben und abgestimmt und dann häufig nicht mehr angepasst. Die daraus resultierenden Denkweisen im Umgang mit Anforderungen herrscht in vielen Organisationen noch vor. Das führt dazu, dass ein PO einerseits häufig unter Zeitdruck Anforderungen erheben und abstimmen muss und zudem erheblichen Aufwand hat, um seine Stakeholder im agilen Prozess nicht abzuhängen.
7	Erhebung von Anforderungen abgestimmt auf Sprint-basierte Entwicklung	Die Sprint-basierte Entwicklung führt dazu, dass ein PO zum Sprintbeginn (Planning) ausreichend Stories im Status "ready" haben muss, um den Sprint sauber planen zu können. Mit der parallelen Betreuung des laufenden Sprints kann dadurch erheblicher Zeitdruck entstehen.
9	Bei einigen Produkten kann es aufgrund konkurrierender Stakeholdergruppen nicht einen einzelnen Product Owner geben (politische Gründe)	Das Produkt sollte aus einem Guss sein und daher sollte es nur einen Product Owner geben
15	Requirements Engineering ist nicht mehr Aufgabe einer speziellen Person oder gar eine eigene Disziplin, sondern muss in Rollen wie Product Owner, Business Analyst oder UX Designer integriert werden	Grundlage für die Funktion von agilen Teams, zu viele hoch spezialisierte Mitarbeiter machen agile Entwicklung unmöglich
19	Rollen werden nicht gelebt oder missgedeutet.	Besonders die Rolle des Product-Owners und die des Scrum-Masters werden oft falsch gedeutet. Ein Product-Owner bestimmt grundsätzlich, welche Anforderungen mit welcher Priorität umgesetzt werden, ein Scrum-Master kümmert sich um den Scrum-Prozess ansich. Oft mischen sich Product-Owner in die Ausführung ein, kontrollieren die Mitarbeiter, verändern laufende Sprints, etc. Scrum-Master werden oft zur Mitarbeiter-Disziplinierung missbraucht.

- Bedürfnis nach Sicherheit -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
5	Ungenau/unvollständige Anforderungen zum Projektstart führen zu Folgeproblemen im Projektverlauf.	Meist folgt diesem eine Budget- UND Zeit-Überschreitung, was wiederum Diskussionen mit dem Kunden zur Folge hat.
6	Auch auf der Management-Ebene entsteht schnell der Eindruck einer Planungsunsicherheit, z.B. "welche Projekte werden im nächsten Jahr umgesetzt".	Obwohl die gerade genannte Planungssicherheit ohnehin meist illusorisch ist, weil "dringliche" oder "wichtige Projekte" dazwischenkommen, halten viele Entscheider an dieser Form fest.
13	Das finden eines geeigneten Grades an Stabilität.	Auch wenn Change explizit willkommen geheißen wird, sind sehr häufig wechselnde Anforderungen teuer (Motivation/Transaktionskosten)
19	Kontrollverlust - Projektleiter werden in ihrer eigentlichen Aufgabe obsolet.	Projektleiter sind in Scrum per definitionem nicht vorgesehen. Hier entstehen oft Ängste, die sich in Ablehnung manifestieren.
20	Die Anforderungen sind oft unpräzise und ändern sich während der Projektlaufzeit. Es kommen auch gerne neue Anforderungen hinzu. Werden die Anforderungen nicht einigermaßen präzise spezifiziert, nutzt das ein Kunde schon mal gerne aus, um möglichst viel Funktionalität bei gleichbleibenden Budget zu bekommen bzw. in der Gewährleistungsphase ist es wichtig Anforderungen genügend spezifiziert zu haben.	Es ist wichtig ein Mittelmaß an Spezifizierungsgrad zu finden um eine winwin Situation und Zufriedenheit für Kunde und Dienstleister zu erreichen.
23	Kosten von Anforderungsänderungen	Dem Nutzen der agilen Softwareentwicklung stehen auch die Kosten gegenüber. Ändern sich Anforderungen, ändern sich auch Kosten in der Entwicklung. Wenn Anforderungen erst spät (aufs Gesamtprojekt) bekannt werden, oder sich ändern (Festpreiskontext) führt das in der Regel zu einer Kostensteigerung.
24	Misunderstanding that Agile means a kind of requirements chaos or no documentation at all	This leads to resistance because of the perceived high risk or feeling danger of losing authority or role. Introducing agile RE will suffer from this resistance.
25	Herausforderungen bestehen in kostendeckender Durchführung von erfolgreichen Softwareprojekten. Diese entstehen oft durch fehlende Erfahrung bzw. fehlendes Verständnis von Rollen und Vorgehensweisen agiler Methoden auf Kundenseite. Erschwerte Kommunikation und unterschiedliche Erwartungen der Projektbeteiligten können die Folge davon sein. Zusätzlich erfolgt oft keine Priorisierung entsprechend der	Die genannten Herausforderung im ersten Abschnitts sind entscheidend für den Erfolg eines Softwareprojekts bezogen auf Kostendeckung, Lieferungumfang bestellter Leistungen, tatsächlich sinnvolle Funktionalität der Software und nicht zuletzt die Zufriedenheit des Kunden.

	umzusetzenden Kernfunktionalität wie z.B. des MVPs, sodass der Begriff agil eher mit der Bedeutung von planlos beschrieben werden kann.	
--	---	--

- Validierung von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
12	Validierung von Anforderungen	Idealerweise sollte man für jede Anforderung bevor sie in die Umsetzung geht, eine Validierung durchführen. Hierdurch kann sichergestellt werden, dass jede Änderung das Produkt tatsächlich verbessert und nicht nur scheinbar eine Verbesserung darstellt. Da eine Validierung, insbesondere durch die Nutzer des Produkts, zeitaufwändig ist, wird oft darauf verzichtet und auch im Nachhinein nicht mehr überprüft, ob die Änderung ggf wieder rückgängig gemacht werden sollte.
21	Da die umzusetzenden Anforderungen im Vorfeld detailliert werden, bleibt keine Zeit, um das Verständnis der dokumentierten Anforderungen noch einmal rückkoppelnd zu prüfen.	Missverständnisse lassen sich somit nicht umgehen
21	Implementierte Anforderungen werden nicht noch einmal mit dem jeweiligen Stakeholder gegengeprüft, sondern gleich im nächsten Release ausgerollt.	Durch eine Prüfung könnte unerwünschtes Feedback ausgeschlossen werden.
21	Durch die steigende Komplexität von Systemen und der globalen Verbreitung von Teams werden viele Anforderungen eingekippt. Oft bleibt keine Zeit, alle Anforderungen genau zu prüfen und zu entscheiden, welche weniger relevant sind und welche klarer definiert werden müssten.	So fehlen ggf. wichtige Punkte im Produkt.

- Requirements Engineering Methoden -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
1	Alle Stories zu einem Projekt aus einer Hand (dem Product Owner) zu bekommen.	Meist sind Projekte doch komplexer als dass sie von einem PO insgesamt erfasst werden können. Stories zu schreiben und diese zu priorisieren kann ein PO meist nicht wirklich objektiv, wenn diese nicht von einer Person erfasst worden sind.

5	Sich ändernde/widersprechende Anforderungen während des Projektverlaufs führen zu zeitlichen Verzögerungen und erneuten Abstimmungen zwischen den Entwicklungssträngen.	Das Managen der Anforderungen ist nicht nur zu Beginn wichtig, sondern auch während des Projektverlaufs.
5	Zu Beginn fehlende Style-Guides, Prototypen, etc. sind oftmals Ursache für eine schlechte UX, da sich in der agilen Softwareentwicklung Entwicklungsstränge meist unabhängig voneinander laufen und jeder Entwickler seine eigene (oder gar keine) Vorstellung von guter UX hat.	Eine schlechte/nicht einheitliche UX führt zu Unzufriedenheit beim Kunden und den End-Nutzern des Produktes.
8	Formulierung und Entscheidungen über Umsetzung der Anforderungen nicht in einem iterativen Wasserfallmodell	der Umsetzungsspielraum wird zu oft eingeschränkt durch einen vorgeschalteten wasserfallartigen Entscheidungsprozess (Analyse, Design) , der ein fertiges Ergebnis (bis ins Detail ausformulierte Anforderung mit Lösung) zur Umsetzung in den agilen Softwareentwicklungsprozess (reine Umsetzung) gibt, der damit auf einen kleinen Bereich im Unternehmen beschränkt bleibt
10	In großen Projekten, Programmen oder konzernweiten Applikation umfasst das klassische RE auch die Auswirkungsanalyse von Änderungen auf alle Komponenten (Abhängigkeiten, Plausibilitäten). Dies ist im Rahmen von agilen Ansätzen meines Erachtens oft nicht möglich, da die agilen Teams einen starken Selbstbezug haben.	In komplexen Umgebungen ist ein Konfigurationsmanagement essentiell, um die Integrität einer Software zu gewährleisten. Dieses mit agilen Vorgehensmodellen zusammenzubringen erscheint mir schwierig, weil die DNA der Vorgehensmodelle nicht zueinanderpasst. Ohne "hybride" Ansätze bekommt man die Modelle daher nicht angenähert.
10	Anforderungen im klassischen RE haben oft eine Prozesssicht oder einen Ursprung in etablierten Prozessen. Diese wiederum sind oft in der Ablauforganisation des Unternehmens verankert, welches das Umfeld für ein Projekt darstellt. Manche davon sind Leitplanken in dem Sinne, das sie aus einem Projekt heraus nicht verändert werden können.	Agilität wird oft als Bottom-Up-Phänomen erlebt. Einzelne Teams beginnen, sich agil aufzustellen, obwohl die sie umgebende Organisation klassisch konstruiert bleibt. Um den Projekterfolg nicht zu gefährden, sind nach meiner Erfahrung beide Perspektiven wichtig: die visionäre Perspektive des Teams, um Innovation zu treiben, aber auch die Perspektive auf das Ökosystem, in welches das Projekt eingebettet ist, um dessen Regeln und Limitierungen zu verstehen und damit umzugehen.
12	Auswirkungen der Ergebnisse einer Iteration auf die darauf Folgende (insbesondere bei Scrum)	Nach Abschluss einer Iteration gibt es eigentlich immer Auswirkungen auf die unmittelbar folgende Iteration. Sei es, dass Aufgaben nicht zufriedenstellend erledigt wurden und weiter bearbeitet werden müssen, oder dass durch die neu erledigten Aufgaben Fehler in anderen Bereichen auftreten, die wieder behoben werden müssen. Bei Scrum ist nun die Herausforderung, diese Auswirkungen zeitnah zu spezifizieren und für die Planung der nächsten Iteration vorzubereiten, ohne dass das restliche Team darauf warten muss.

15	Nicht-Funktionale Anforderungen, z.B. bzgl. Betriebsqualitäten oder Produktgestaltung werden genauso missachtet oder vergessen, wie in Wasserfallmodellen	Hat negative Einflüsse auf Produktqualität
15	Klassische Methoden zur Erhebung und Evaluierung von Anforderungen sind zu langsam. Es sind schnellere Methoden gefragt, bei denen die Erkenntnisse mit weniger Aufwand durch das ganze Team gewonnen werden. (Kürzer Durchlaufzeit, weniger organisatorischer Aufwand, Verteilung der Auswertung, Verzicht auf umfangreiche Doku)	Grundvoraussetzung für das Gelingen im agilen Umfeld
19	Kunden miteinbeziehen und Vorteile herausarbeiten.	Kunden sind oft zunächst skeptisch, wenn sie keine umfangreichen Pflichtenhefte und exakte Projektpläne mit definierten Meilensteinen bekommen, die letztlich ohnehin nicht der Realität entsprechen. Hier sollte wenn möglich der Kunde in die Prozesse des Requirement-Engineering und innerhalb von Scrum in die Reviews wenn möglich eng eingebunden werden.
21	Nicht immer sind die umzusetzenden Anforderungen zum Entwicklungsstart klar definiert, da sie im Vorfeld nur kurz dokumentiert wurden, und die Prioritäten gerade nicht auf diesen Anforderungen lagen. Dies liegt in der fehlenden Planung/Vorbereitung der Ziele der nächsten Sprints und Releases. Oft verändern sich da die Prioritäten, und unklare Anforderungen müssen schnell umgesetzt werden.	In der Praxis sind Prio-Verschiebungen normales Tagesgeschäft.
23	Pflege der Anforderungen	Da nicht alle Anforderungen am Anfang eines Entwicklung festgelegt, sondern erst über die Zeit detailliert werden, ist die kontinuierliche Pflege von Anforderungen extrem wichtig. Häufig werden Details besprochen, diese dann aber nicht in Tickets etc. integriert und existieren dann nur im Kopf von ein paar Beteiligten.
25	In Projekten mit langer Laufzeit stellt zusätzlich die Nachverfolgbarkeit von Anforderungen eine große Herausforderung dar. Dazu werden in der Praxis häufig Ticketing-Systeme wie z.B. Jira eingesetzt oder Excel-Sheets, in denen mit der Zeit der Überblick über die Änderungshistorie der Anforderungen verloren gehen kann.	Die genannten Herausforderung im ersten Abschnitts sind entscheidend für den Erfolg eines Softwareprojekts bezogen auf Kostendeckung, Lieferungsumfang bestellter Leistungen, tatsächlich sinnvolle Funktionalität der Software und nicht zuletzt die Zufriedenheit des Kunden.

26	Die klassischen Projektmanagementmethoden haben einem definierten Zeitpunkt an dem das RE durchgeführt wird. Bei den agilen Methoden gibt es diesen nicht.	Dieses stellt das Projektteam vor die Herausforderung zu definieren in welchen Rahmen und Umfang die Anforderungen aufgenommen werden.
26	Die agilen Methoden insbesondere Scrum geben mit dem Wert „Working software over comprehensive documentation“ den source code Vorrang zu abgestimmten Anforderungen.	Dies bedeutet, dass insbesondere in agilen Projekte neue Kommunikationsstrukturen implementiert werden müssen und diese auch von alle Beteiligten befolgt werden müssen.
26	Die Kommunikation und Einhaltung von nicht funktionalen Anforderungen (z.B.: Performance, Security, Architektur, Barrierefreiheit etc.).	Es gibt keine Kommunikationstechnik diese verbindlich für ein agiles Projekt festzulegen.

- Format der Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
1	Übersetzung eines Pflichtenheftes in Stories.	Wenn Anforderungen als ein Pflichtenheft von mehreren Personen verfasst worden ist, sollte Product Owner diese in geeignete Stories übersetzen.
4	Anforderungen in der Agilen Softwareentwicklung müssen Lean sein	Nur schlanke Anforderungsdokumente lassen sich mit vertretbarem Aufwand an sich ändernde Umgebungsfaktoren anpassen
7	Erhebung testbarer Anforderungen	Durch die Konzentration auf fachliche Stories kommt es häufig dazu, dass die Akzeptanzkriterien keine harten technischen Aspekte beinhalten. Die Testkriterien sind entsprechend weich und häufig schlecht auf die Anforderungen abgestimmt, woraus Qualitätsprobleme entstehen.
8	Anforderungen als Anforderungen (=Idee) zu formulieren und keine Lösungen als Anforderung zu formulieren.	Lösungsvorgaben schränken die Entwicklung in Bezug auf Kreativität in der Lösungsfindung ein.
10	Klassisches RE ist nach meiner Erfahrung kleinteiliger als User Stories. Besonders bei funktionalen Anforderungen liegen oft mechanische Prinzipien zugrunde, die bei eher prosaischen User Stories so nicht erfasst werden. Häufig geht es dabei auch um einfache Kriterien, um eine funktionale Anforderung im fertigen Produkt messen zu können.	Mechaniken und Messbarkeit drücken oft vertraute Kriterien der Fachseite aus und bewahren dort ein Gefühl der Kontrolle (Ursache-Wirkungszusammenhänge, bekannte KPIs, etc.). Diese sollten daher auch im agilen RE berücksichtigt und erfasst werden.
11	Die Anforderungen sollten nicht in der Art "der Button muss gelb sein" erfolgen, sondern es muss der Sinn und Zweck transportiert werden.	Nur so können die Teams auch die technische Umsetzung agil und schlank machen..

20	Die Anforderungen sind oft unpräzise und ändern sich während der Projektlaufzeit. Es kommen auch gerne neue Anforderungen hinzu. Werden die Anforderungen nicht einigermaßen präzise spezifiziert, nutzt das ein Kunde schon mal gerne aus, um möglichst viel Funktionalität bei gleichbleibenden Budget zu bekommen bzw. in der Gewährleistungsphase ist es wichtig Anforderungen genügend spezifiziert zu haben.	Es ist wichtig ein Mittelmaß an Spezifizierungsgrad zu finden um eine winwin Situation und Zufriedenheit für Kunde und Dienstleister zu erreichen.
20	Anforderungen bzw. Akzeptanzkriterien präzise genug zu erfassen, das eine Qualitätssicherung detaillierte Testfälle daraus ableiten bzw. durchführen kann.	Qualitätssicherung ist in komplexer und fehlerträchtiger Software wichtig. Darum ist es umso wichtiger das aus den Anforderungen bzw. Akzeptanzkriterien genügend Informationen bereitstehen, um eine genügend hohe Testabdeckung zu erreichen.
21	Schnelllebigkeit und schnelle Umsetzung lassen nur sehr selten Zeit, gezielt Anforderungen zu definieren. Oft bleiben diese oberflächlich.	Oberflächliche Anforderungen lassen viel Spielraum für die Umsetzung offen.
22	Das Formulieren von Zielen/Hypothesen/Problemstellungen anstelle der Vorgabe von Lösungen.	Nur klar geäußerte Ziele/Hypothesen/Problemstellungen inkl. des Kontexts führen zu passenden Lösungen (z.B. schnell und billig für eine ungeprüfte Hypothese vs. gründlich und langlebig für eine bewiesene Hypothese).
24	Lack of skill/knowledge/experience with agile way of specifying and managing requirements.	This leads to lower quality of documenting requirements though much effort is invested. Then after requirements are finally documented, the changes are not documented properly. Also, tracking changes in case of fixed price projects is not easily possible, because tools are either not known or not mature enough. Communicating requirements depends mostly on documents and less on face to face conversation. This increases risks of overseeing important expectations or challenges.
25	In Projekten mit langer Laufzeit stellt zusätzlich die Nachverfolgbarkeit von Anforderungen eine große Herausforderung dar. Dazu werden in der Praxis häufig Ticketing-Systeme wie z.B. Jira eingesetzt oder Excel-Sheets, in denen mit der Zeit der Überblick über die Änderungshistorie der Anforderungen verloren gehen kann.	Die genannten Herausforderung im ersten Abschnitts sind entscheidend für den Erfolg eines Softwareprojekts bezogen auf Kostendeckung, Lieferungumfang bestellter Leistungen, tatsächlich sinnvolle Funktionalität der Software und nicht zuletzt die Zufriedenheit des Kunden.

- Klarheit von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
4	Anforderungen in der Agilen Softwareentwicklung müssen klar und verständlich sein	Unklare Formulierungen führen zu Unsicherheit in der Umsetzung
15	Der geeignete Zuschnitt von Anforderungen ist bei komplexen Geschäftsprozessen nachwievor schwierig. Vorgehen wie Use Case 2.0 helfen ein wenig, aber die Abneigung gegen Dokumentation in agilen Umgebungen behindert.	Komplexe Prozesse sind der Normalfall in der B2B-Softwareentwicklung
20	Die Anforderungen sind oft unpräzise und ändern sich während der Projektlaufzeit. Es kommen auch gerne neue Anforderungen hinzu. Werden die Anforderungen nicht einigermaßen präzise spezifiziert, nutzt das ein Kunde schon mal gerne aus, um möglichst viel Funktionalität bei gleichbleibenden Budget zu bekommen bzw. in der Gewährleistungsphase ist es wichtig Anforderungen genügend spezifiziert zu haben.	Es ist wichtig ein Mittelmaß an Spezifizierungsgrad zu finden um eine winwin Situation und Zufriedenheit für Kunde und Dienstleister zu erreichen.
21	Schnelllebigkeit und schnelle Umsetzung lassen nur sehr selten Zeit, gezielt Anforderungen zu definieren. Oft bleiben diese oberflächlich.	Oberflächliche Anforderungen lassen viel Spielraum für die Umsetzung offen.
21	Kein einheitliches Verständnis für die Dokumentation von Nutzer- und technischen Anforderungen, sowie warum dazu eine Begründung des Nutzens der Anforderung gehört. Nur so lassen sich später auch noch Entscheidungen für eine bestimmte Anforderung begründen.	Zu viel Interpretationsspielraum und Unklarheiten; halten späterer Prüfung nicht mehr stand

- Priorisierung von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
12	Auswirkungen der Ergebnisse einer Iteration auf die darauf Folgende (insbesondere bei Scrum)	Nach Abschluss einer Iteration gibt es eigentlich immer Auswirkungen auf die unmittelbar folgende Iteration. Sei es, dass Aufgaben nicht zufriedenstellend erledigt wurden und weiter bearbeitet werden müssen, oder dass durch die neu erledigten Aufgaben Fehler in anderen Bereichen auftreten, die wieder behoben werden müssen. Bei Scrum ist nun die Herausforderung, diese Auswirkungen zeitnah zu spezifizieren und für die Planung der nächsten Iteration vorzubereiten, ohne dass das restliche Team darauf warten muss.
14	Richtige Priorisierung nach diversen Gesichtspunkten wie Dringlichkeit, Geschäftswert, technischer Abhängigkeit...	Es müssen verschiedenste auf die Priorisierung einwirkende Faktoren bewertet und berücksichtigt werden.

18	Qualität vs. Quantität -> nicht nur Feature entwickeln, sondern auch Spielraum für sinnvolles Refactoring und gute Testautomatisierung lassen	Ein Produkt kann langfristig nur konkurrenzfähig bleiben wenn es den aktuellen technischen Standards entspricht, stabil ist und es möglichst kurze Releasezyklen ohne viel Overhead mit sich bringt.
20	Kunden fällt es schwer Anforderungen in einem Backlog zu priorisieren. Das ihnen oft nicht der Businessvalue, wonach unter anderem priorisiert wird klar ist.	Es ist wichtig ein Backlog zu priorisieren, um mit den "wichtigen" Dingen zu beginnen, die Mehrwert bringen, für einen technischen Durchstich sorgen und zur Kundenzufriedenheit führen sollten. Wichtig ist auch eventuell unwichtige Dinge weglassen zu können und/oder durch neuen wichtigere Dinge ersetzen zu können.
22	Die Limitierung der gleichzeitigen getanen Arbeit und die dafür nötige Priorisierung.	In fast jeder Situation ist die Arbeitskraft durch diverse Faktoren begrenzt (Geld, Orchestration, Abhängigkeiten). Zudem erzeugt starke Parallelisierung bei eng begrenzten Ressourcen meist großen Overhead. Daher muss es ein WIP Limit geben. Außerdem herrscht anfänglich große Unsicherheit über die "passende Lösung". Deswegen muss jeweils über "jetzt gerade wichtig" und "jetzt weniger wichtig" unterschieden werden, um schnell Risiken zu verkleinern und Unsicherheit zu beseitigen.
22	Das Finden von kleinen "passenden Lösungen", die gut genug sind. Das Abbrechen von Arbeit wenn das Ziel erreicht wurde.	Erkenntnis aus Studien: 80% des relevanten Nutzen stecken in 20% der Anforderungen. Projekte schaffen den größten Wertzuwachs in den ersten 20% der Arbeit.

- Verfeinerung (engl. Refinement) -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
2	Ausreichend Kommunikation	Weil sonst etwas herauskommt, was nicht gewollt war
3	Die immer komplexer und individuelleren Anfragen der Kunden jedes mal aufs neue zu identifizieren und in die IT-Welt zu übersetzen.	Ohne diese herangehensweise würden größere Softwareprojekte mit den heutigen Anforderungen scheitern
5	Fehlende Kenntnis des Stakeholders über agile Softwareentwicklung, Releases, etc.	Ein grundsätzliches Wissen beim Stakeholder ist wichtig, damit mit einer Sprache gesprochen werden kann und keine Missverständnisse auftreten. Durch den Begriff "agil" wird oft angenommen, dass "eh alles flexibel" sei, was dann wiederum ungenaue und unvollständige Anforderungen zum Start mit sich bringt.
7	Erhebung testbarer Anforderungen	Durch die Konzentration auf fachliche Stories kommt es häufig dazu, dass die Akzeptanzkriterien keine harten technischen Aspekte beinhalten. Die Testkriterien sind entsprechend weich und häufig schlecht auf die Anforderungen abgestimmt, woraus Qualitätsprobleme entstehen.
8	Formulierung der	zu oft geht Zeit verloren, weil Anforderungen und Mehrwert dem

	Anforderungen (Idee) hinreichend konkret für eine Umsetzung in der Entwicklung	Anforderer selbst nicht hinreichend klar sind, so dass unnötig Blindleistung produziert wird
21	Durch die Vielzahl der Stakeholder widersprechen sich teilweise Anforderungen für ein Feature.	Es benötigt Zeit die meistens nicht vorhanden ist, um zu detaillieren, welche Anforderung umgesetzt werden muss
23	Vollständigkeit von Anforderungen Part 2	Bei Entwicklungsstart sollten alle Details von Anforderungen bekannt sein. Ist die Planung allerdings nicht so kleinteilig, dass eine Funktion von einer Person umgesetzt werden kann, sondern ggf. das komplette Team an dieser Funktion arbeitet ist die Wahrscheinlichkeit groß, dass auch die Anforderungen, die in der Story umgesetzt werden sollen umfangreich sind. Hier besteht die Gefahr, dass nicht alle Details dieser Anforderungen bekannt sind (z.B. bei Designs) und erst während der Entwicklung (kurz davor) entdeckt werden. Diese schnell zu klären ist ggf. nicht immer möglich und kann zu Verzögerungen führen.

- Ermittlung von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
11	Änderungen als gut akzeptieren. Es ist nicht gleich ein Change Request, sondern durch Erfahrung mit dem Produkt und der Software lernen wir mehr. Dabei ergeben sich Änderungen. Wenn das nicht so wäre, dann hätte man nichts gelernt.	Auch das eines der Probleme die Leute aus "klassischen" Projekten nur schwer verstehen / berücksichtigen.
17	Anforderungsanalyse suggeriert, dass man alle Anforderungen vor der Realisierung finden und sammeln kann.	Agile Softwareentwicklung weiß, dass nicht alles am Anfang bekannt ist und sich Anforderungen ändern, neue hinzukommen und sich ein Produkt stetig weiterentwickelt.
21	Nicht immer sind die umzusetzenden Anforderungen zum Entwicklungsstart klar definiert, da sie im Vorfeld nur kurz dokumentiert wurden, und die Prioritäten gerade nicht auf diesen Anforderungen lagen. Dies liegt in der fehlenden Planung/Vorbereitung der Ziele der nächsten Sprints und Releases. Oft verändern sich da die Prioritäten, und unklare Anforderungen müssen schnell umgesetzt werden.	In der Praxis sind Prio-Verschiebungen normales Tagesgeschäft.
22	Das Zerlegen von großen Zielen/Hypothesen/Problemstellungen in kleine Schritte/Aufgaben.	Die "passende Lösung" ist eine Funktion der gesammelten Erfahrungen und des Kontext. Beides sind wiederum u.a. Funktionen der Zeit. Nur Messbarkeit von Zwischenergebnissen und kurze Feedbackzyklen führen zu der passenden Lösung.

- Transparenz von Anforderungen -

Expert	Welche Herausforderung sehen Sie im Umgang mit Anforderungen in der Agilen Softwareentwicklung?	Wieso erachten Sie diese Herausforderung als wichtig?
13	Schaffung des richtigen Transparenz-Grades für alle Beteiligten.	Nur wenn alle die für Ihre Arbeit und Ziele notwendigen aktuellen Informationen haben, kann per Selbstorganisation effektiv und effizient arbeiten.
19	Kunden miteinbeziehen und Vorteile herausarbeiten.	Kunden sind oft zunächst skeptisch, wenn sie keine umfangreichen Pflichtenhefte und exakte Projektpläne mit definierten Meilensteinen bekommen, die letztlich ohnehin nicht der Realität entsprechen. Hier sollte wenn möglich der Kunde in die Prozesse des Requirement-Engineering und innerhalb von Scrum in die Reviews wenn möglich eng eingebunden werden.
19	Mit Wahrheit umgehen können.	Agile vorgehensmodelle sorgen nicht dafür, dass Projekte nicht mehr scheitern, aber sie zeigen frühzeitig eventuelle Schief lagen auf und ermöglichen steuerndes Eingreifen. Führungspersonal möchte oft nicht mit unangenehmen Wahrheiten konfrontiert werden oder diese bis zuletzt nicht wahrhaben. Hier kommt ihnen ein Wasserfall/V-Modell, etc. eher entgegen.

Results Report Round 2 – original comments stated by experts

- Involvieren von Stakeholder und Nutzer -

Item 1.1 In der agilen Softwareentwicklung müssen Benutzeranforderungen und Nutzungsqualitäten in Zusammenarbeit mit direkten Nutzern (end user) des Produkts erarbeitet werden, um nicht an der Zielgruppe vorbei zu entwickeln.

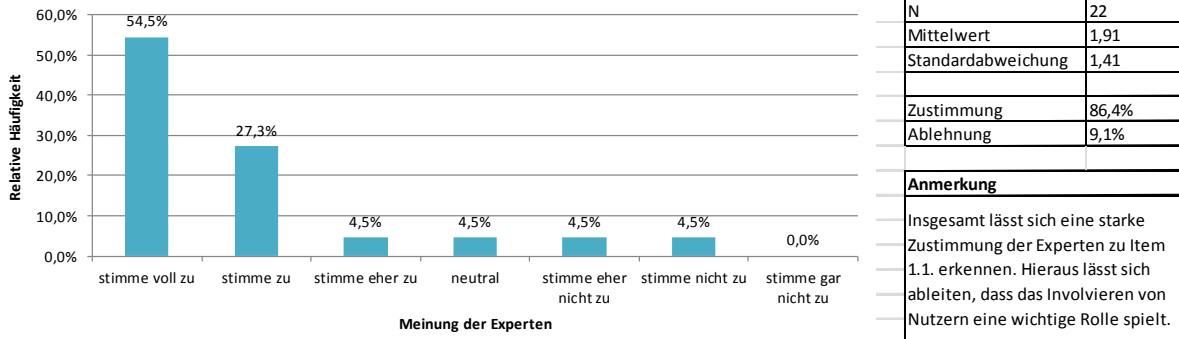


Figure 53: Item 1.1 – direct cooperation with end users

- (Experte 16) Hier kann ich weder zustimmen noch widersprechen, weil es zu sehr auf das Produkt ankommt. Es gibt viele Programme, wo man mit Hilfe von A/B Tests sich dann zusammen mit den Stakeholdern die Anforderungen erarbeitet, aber auch Software, wo die Idee erst vollständig mit dem Stakeholder erarbeitet wird bevor ein Enduser dieses zu Gesicht bekommt. Eine dritte von mir sehr geschätzte Methode sind User Cafés und Guerilla Testing mit Prototypen.
- (Experte 25) Wenn Endnutzer einer Software die PO-Rolle übernehmen, kann es dazu führen, dass bestimmte Features bzw. Anforderungen entsprechend der Erwartungen optimal umgesetzt werden und darüber hinaus oft sogar übererfüllt werden (Stichwort goldener Wasserhahn). Das kann dazu führen, dass für nachfolgende umzusetzende Features nicht genügend Zeit für die Umsetzung bleibt und der Überblick über den Gesamt-Scope aus den Augen verloren wird.

Item 1.2 In der agilen Softwareentwicklung müssen Stakeholder während der gesamten Entwicklung in regelmäßigen Iterationen eingebunden werden, damit die Produktentwicklung erfolgreich wird.

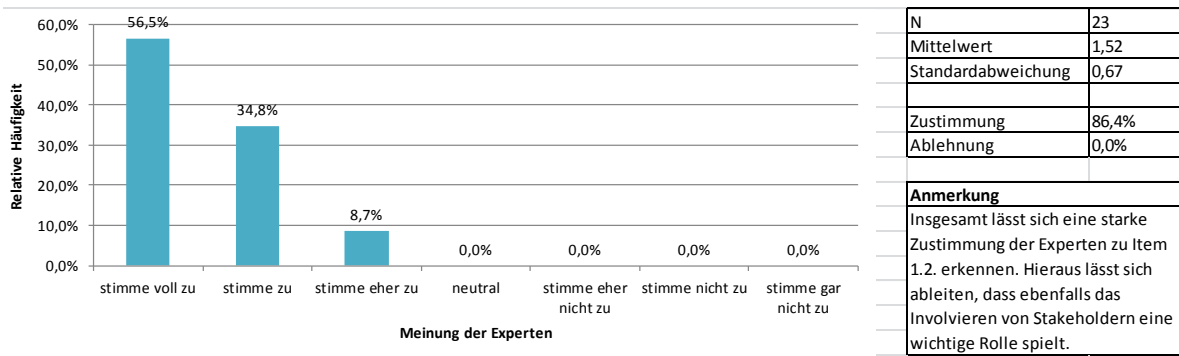


Figure 54: Item 1.2 – iterative involvement of stakeholders

Item 1.3 In der agilen Softwareentwicklung muss die Kommunikation der Anforderungen von Angesicht zu Angesicht erfolgen, damit wichtige Erwartungshaltungen nicht übersehen werden.

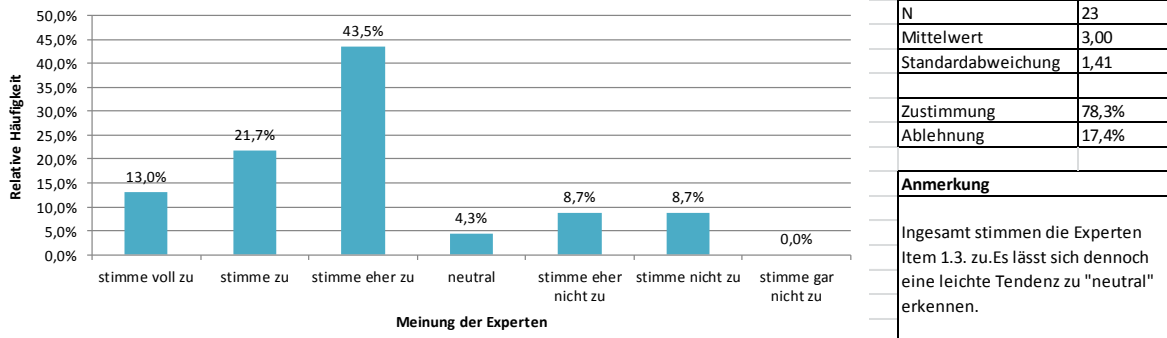


Figure 55: Item 1.3 – face to face communication

Item 1.4 In der agilen Softwareentwicklung müssen Anforderungen vor der Umsetzung mit Nutzern validiert werden, um zu evaluieren, ob die angestrebte Änderung das Produkt verbessert.

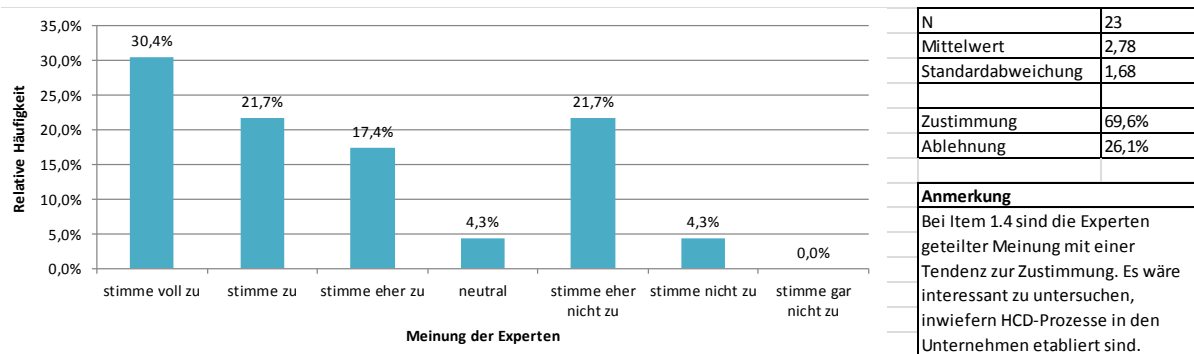


Figure 56: Item 1.4 – evaluation of requirements with end users

- (Experte 16) Auch hier können Varianten erarbeitet werden und diese im dann im Produkt getestet werden. A/B Testing ist häufig die beste Variante um die Entwicklung des Produkts im Auge zu behalten.

Item 1.5 In der agilen Softwareentwicklung werden umgesetzte Anforderungen nicht immer mit dem jeweiligen Stakeholder gegengeprüft. Dies führt dazu, dass Feedback erst nach dem Release entsteht.

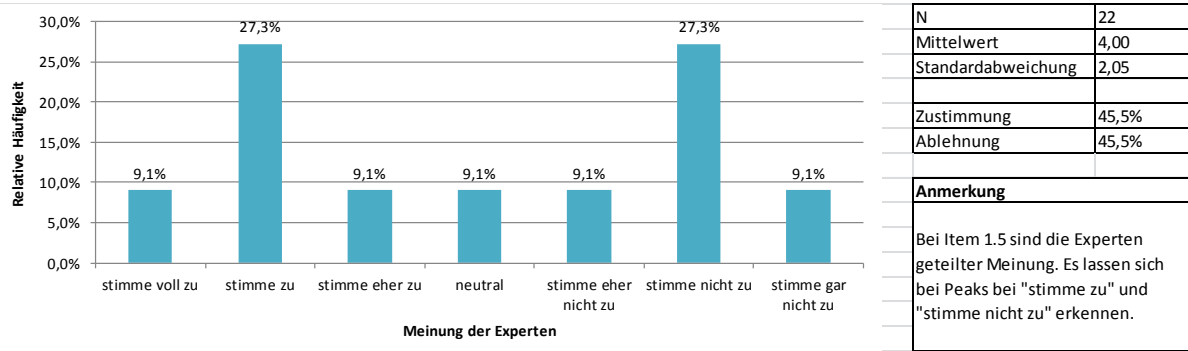


Figure 57: Items 1.5 – review of implemented requirements

- (Experte 23) Wenn Agilität richtig angewendet wird, wird die Anforderung gar nicht umgesetzt, wenn sie nicht zuvor vom Stakeholder validiert wurde.
- (Experte 24) Teil der Fragen lässt sich interpretieren. Z.B. In 1.5 war es mir nicht 100% klar, ob es um die Prinzipien oder Realität geht. Ich habe die Prinzipien angenommen.

Item 1.6 In der agilen Softwareentwicklung ist eine kontinuierliche Kommunikation notwendig, damit eine korrekte Umsetzung erfolgen kann.

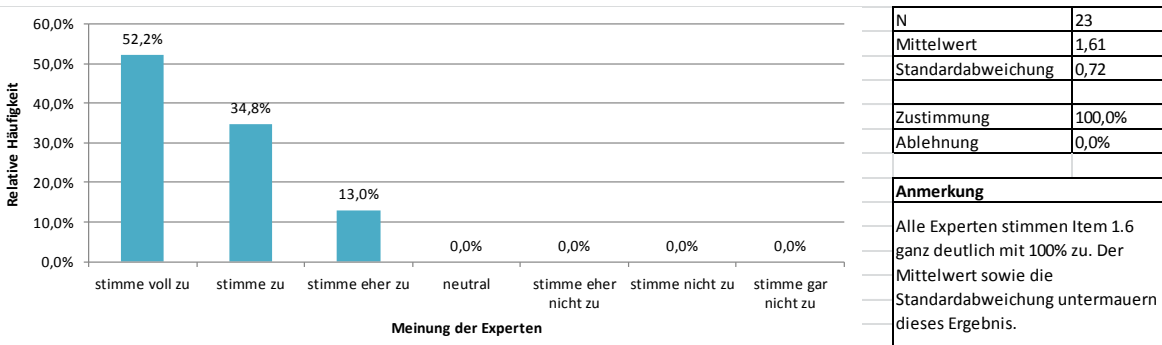


Figure 58: Item 1.6 – continuous communication

Allgemeine Anmerkung zu Involvieren von Stakeholder und Nutzer

- (Experte 9) Die Antworten hängen sehr stark von der Art des Produktes (Massenprodukt mit vielen unbekanntem Nutzern oder Individuallösung für wenige bekannte Nutzer) und der Art der Anforderung ab (Basis-, Leistungs- oder Begeisterungsfaktor).
- (Experte 13) Das Wörtchen MUSS finde ich persönlich schwierig. Alle diese Punkte verbessern aus meiner Sicht die Erfolgchancen. Mit MUSS wäre ich sehr vorsichtig.

- (Experte 15) Die Methoden zum Einbezug von Nutzern und Stakeholdern müssen in agilen Projekten deutlich leichtgewichtiger sein. Es geht meist eher darum eine Annahme zu verstehen und sinnvoll zu begründen, als zu 100% zu beweisen.

- Verständnis von Agile und Agilen Werten -

Item 2.1 In der agilen Softwareentwicklung fehlt häufig seitens der Stakeholder das Verständnis dafür, dass Anforderungen mit dem Entwicklungsteam verhandelbar sein müssen. Im Ergebnis werden teilweise nicht-optimale Lösungen implementiert.

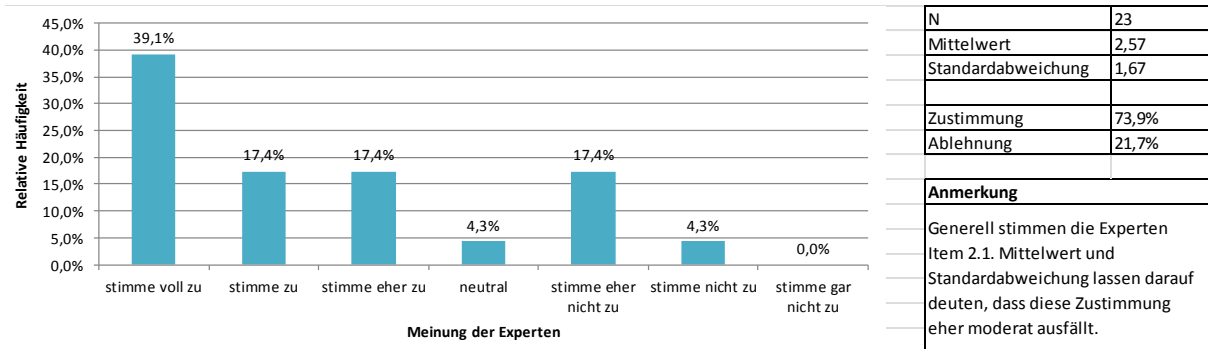


Figure 59: Item 2.1 – understanding of agile values

Item 2.2 In der agilen Softwareentwicklung muss eine große Anforderungsanalyse am Anfang (z.B. Auflistung, Schätzung, Priorisierung, Testbarkeit) vermieden werden, da dies zu einem wasserfallartigen Vorgehen in der Umsetzung führt.

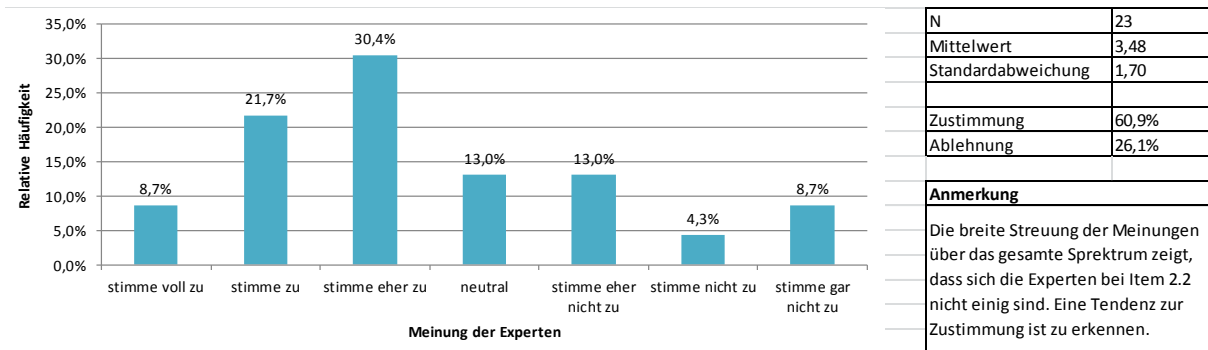


Figure 60: Item 2.2 – avoiding large requirements analysis in the beginning

- (Experte 24) Der Zweck der agilen Software Entwicklung ist es, schnelle Feedbackloops vom Endnutzern/Markt. Wasserfall zu vermeiden ist aus meiner Sicht an sich kein Grund.

Item 2.3 In der agilen Softwareentwicklung liegen viele Entscheidungen beim Entwicklungsteam. Stakeholder müssen akzeptieren, dass sie nicht bei allen Details mitbestimmen dürfen.

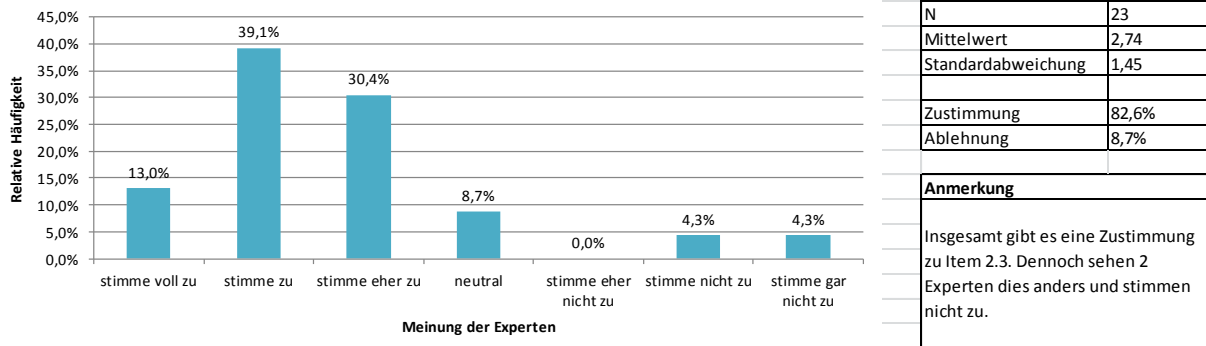


Figure 61: Item 2.3 – decisions of the development team

- (Experte 23) die Aussage ist insofern richtig, dass Stakeholder nicht in allen Details "mitspracherecht haben", allerdings sollten keine Entscheidungen vom Team getroffen werden, ohne Rücksprache, die die Funktionen/-umfang des Produktes verändern.
- (Experte 24) Ich habe die zwei Teile der Frage als zwei Fragen wahrgenommen, die gleich bewertet sein müssen.

Item 2.4 In der agilen Softwareentwicklung bekommt der Product Owner Probleme bei der Durchführung seiner Aufgaben, wenn in der Organisation Anforderungen in vorgelagerten Phasen festlegt werden.

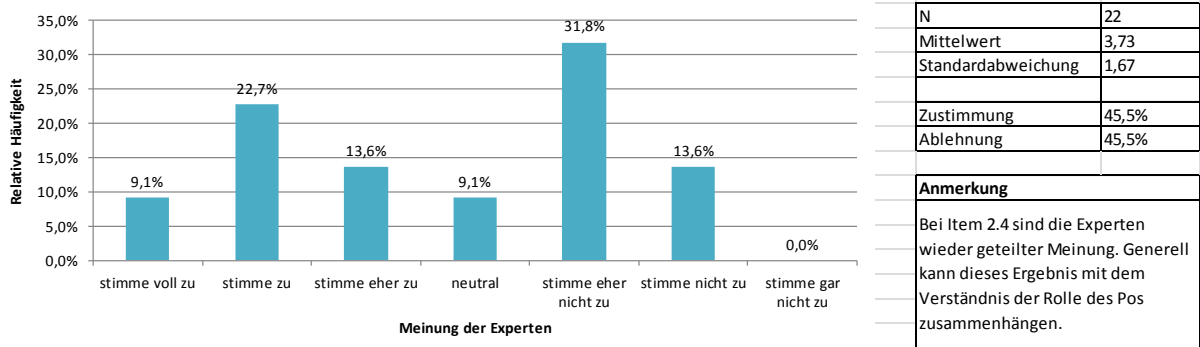


Figure 62: Item 2.4 – handling requirements within an organization

- (Experte 16) Verstehe ich nicht.

Item 2.5 In der agilen Softwareentwicklung kann es zu Fehlinterpretationen kommen und Agile mit Chaos der Anforderungen oder fehlende Dokumentation gleichgesetzt werden. Dies führt zu Widerständen, da zum einen große Risiken befürchtet werden und zum anderen die Angst vor einem Kontrollverlust besteht.

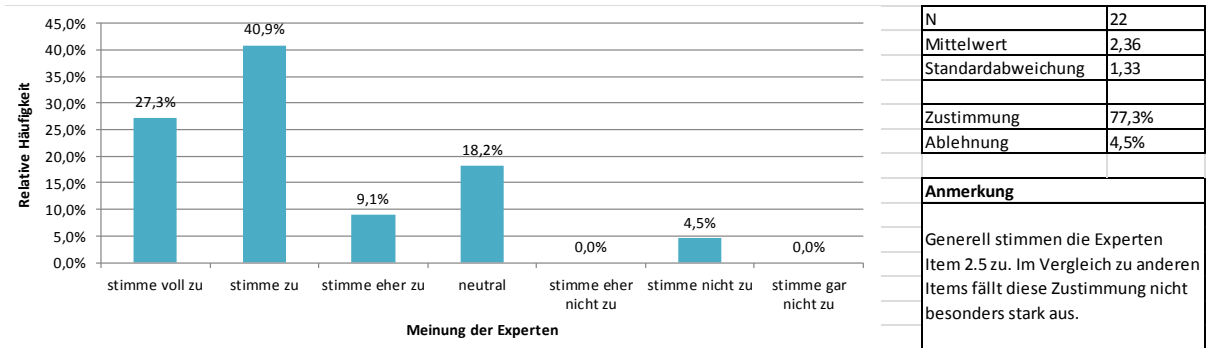


Figure 63: Item 2.5 – understanding agile values in terms of fears of losing control

- (Experte 16) Hier ist es einfach so, dass der Widerstand gegen die Agile Entwicklung im Web Umfeld nicht mehr existiert. Ich habe nur in Anwendungsentwicklung davon gehört, aber auch dort hat sich fast immer Scrum durchgesetzt. Im Webbereich eher Kanban.
- (Experte 18) Natürlich kann dieser Fall eintreten, kann aber durch eine klare Formulierung und Abstimmung der Anforderung mit den Stakeholdern vermieden werden. Epics/Stories sollten möglichst Interpretationsfrei formuliert sein.
- (Experte 24) Hier hängt es von der Transformationsphase der Organisation.

Item 2.6 In der agilen Softwareentwicklung ergeben sich aufgrund der Erfahrung mit dem Produkt Änderung in den Anforderungen. Oftmals suggeriert eine Anforderungsanalyse, dass alle Anforderungen vor der Realisierung gefunden und gesammelt werden können.

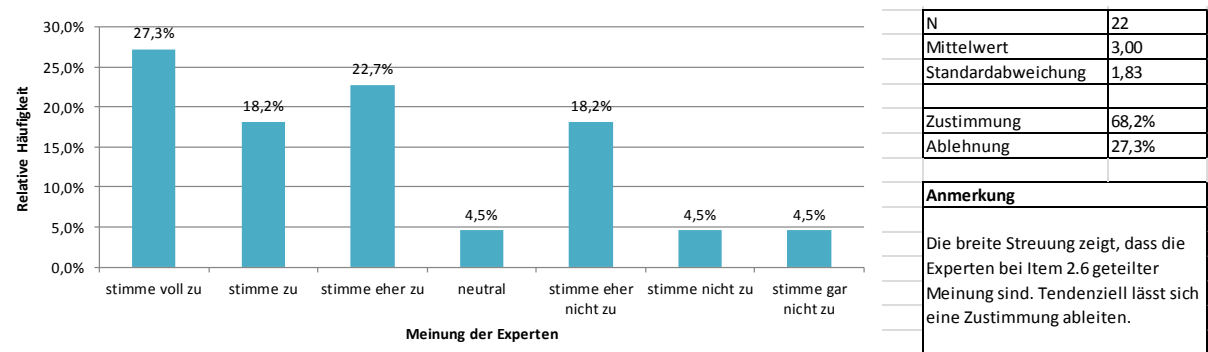


Figure 64: Item 2.6 – perception of requirement analysis

- (Experte 16) Ist das nicht sogar ein Widerspruch. Ja, die Produkterfahrung verändert das Produkt selbst relativ schnell. Vor der Realisierung ist dieses also nicht auffindbar.
- (Experte 18) Ich habe die Aussage nicht verstanden und daher keine Antwort.
- (Experte 24) Meine Antwort bezieht sich auf den zweiten Teil der Frage. Für den ersten Teil würde ich gegenteilige Antwort geben.

Allgemeine Anmerkung zu Verständnis von Agile und Agilen Werten

- (Experte 7) Das Entwicklungsteam entscheidet wie etwas technisch umgesetzt wird. Was fachlich gefordert ist, entscheidet der PO in Rückkopplung mit den Stakeholdern. Die Stakeholder müssen Einfluss auf Details haben, ansonsten kann die Akzeptanz des Produkts darunter leiden (und Devs treffen nicht immer die besten fachlichen Entscheidungen).
- (Experte 23) Es liegt in der Natur der Dinge, dass vor dem Beginn der Umsetzung, bzw. bereits vor dem eigentlichen Projekt Anforderungen (Geschäftsanforderungen) existieren. Es macht Sinn diese in einer vorgelagerten Phase zusammenzutragen und ggf. auch gegeneinander abzuwägen (Stichwort entgegengesetzte Anforderungen) um ein Bild des Produktes zu erhalten. Es ist richtig, dass in dieser Phase nicht alle Details geklärt werden sollten.

- Requirements Engineering Methoden -

Item 3.1 In der agilen Softwareentwicklung ist eine klassische Auswirkungsanalyse von Änderungen auf alle Komponenten nur schwer möglich, da oftmals der Überblick fehlt.

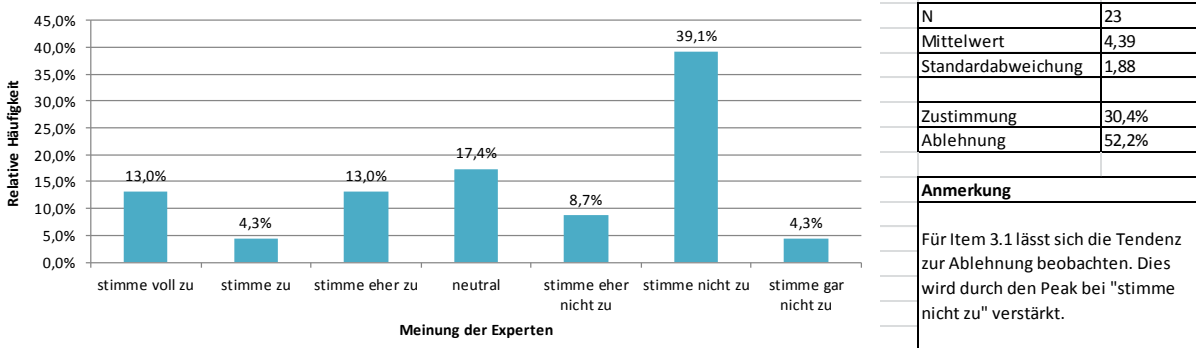


Figure 65: Item 3.1 – impact analysis of components

- (Experte 23) Wenn der Überblick fehlt, kann man nicht entscheiden welche Anforderungen den höchsten Business Value haben und damit auch die Auswirkungen deren Umsetzung.
- (Experte 24) Meine Antwort bezieht sich auf das Prinzip. In der Realität kann es doch richtig sein, weil die ganzheitliche Betrachtung manchmal fehlt.

Item 3.2 In der agilen Softwareentwicklung können fehlende Style-Guides, Prototypen, etc. dazu führen, dass Entwickler ihre eigene Vorstellung von guter User Experience umsetzen. Dies kann Ursache für eine schlechte User Experience des Produktes sein.

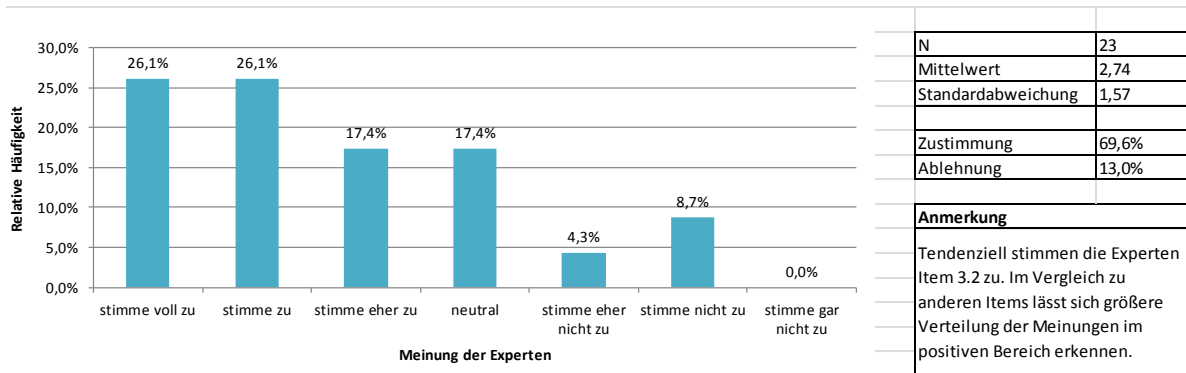


Figure 66: Item 3.2 – impact of styleguides and prototypes on user experience

- (Experte 13) Frage 3.2 ist für mich unabhängig vom Vorgehensmodell

Item 3.3 In der agilen Softwareentwicklung wird der Umsetzungsspielraum durch einen vorgeschalteten, wasserfallartigen Entscheidungsprozess eingeschränkt, da ein fertiger Prototyp zur reinen Umsetzung in den agilen Softwareentwicklungsprozess gegeben wird.

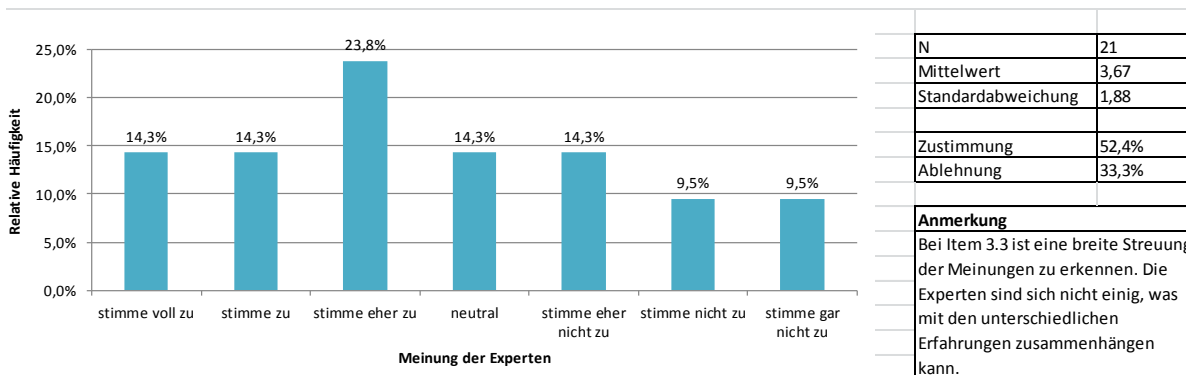


Figure 67: Item 3.3 – restricting solution finding with finalized prototypes

- (Experte 16) Ich habe es erlebt, aber immer selber dafür gesorgt, dass es beseitigt wird, weil es KEINE agile Entwicklung ist. Daher ist diese Aussage nicht korrekt.
- (Experte 23) Ich verstehe 3.3 nicht. Wer schreibt vor, dass es einen Prototypen geben muss?
- (Experte 24) Die Frage nimmt an, dass ein vorgeschalteter, wasserfallartiger Entscheidungsprozess zu einem fertigen "Prototyp" führt. Meine Antwort bezieht sich auf diese Annahme, obwohl es nicht immer der Fall ist.
- (Experte 25) In der Praxis fördern fertige Prototypen, die mit den Stakeholdern abgestimmt sind (d.h. POs, Entwicklerteam, Usability Experten) das gemeinsame Verständnis von konkreten Anforderungen. Aus meiner Sicht können damit Aufwände reduziert werden, die durch nachträgliche Änderungen entstehen. Diese "Einschränkung" wie sie in Punkt 3.3 genannt wird ist im positiven Sinne eher eine klar definierte Vorgabe.

Item 3.4 In der agilen Softwareentwicklung ist eine kontinuierliche Pflege der Anforderungen wichtig, da zu Beginn nicht alle Anforderungen feststehen und diese sich über den Projektverlauf ändern.

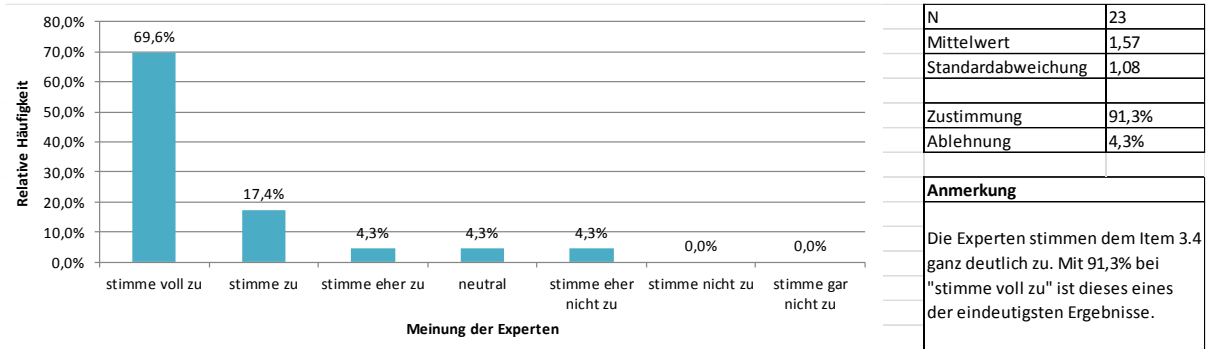


Figure 68: Item 3.4 – continuous requirements management

Item 3.5 In der agilen Softwareentwicklung sind klassische Methoden zur Erhebung und Evaluierung von Anforderungen oftmals zu langsam.

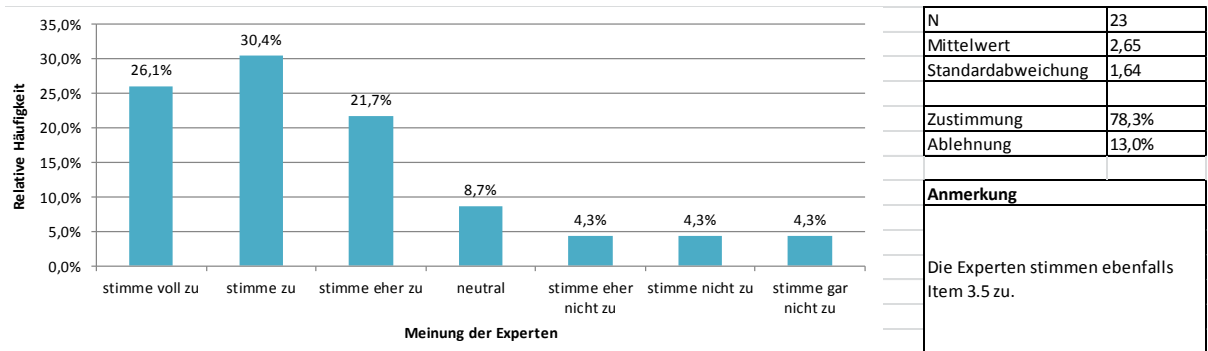


Figure 69: Item 3.5 – classical methods for elicitation and evaluation

Item 3.6 In der agilen Softwareentwicklung müssen Methoden zur Erhebung und Evaluierung von Anforderungen eingesetzt werden, bei denen die Erkenntnisse mit dem Entwicklungsteam geteilt werden.

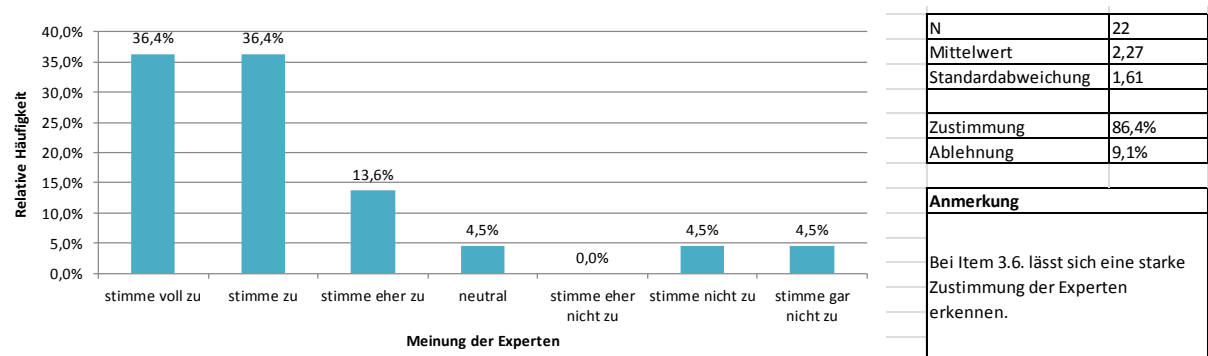


Figure 70: Item 3.6 – sharing insights with the whole development team

Item 3.7 In der agilen Softwareentwicklung werden die Änderungen der Anforderungen oftmals nicht nachvollziehbar dokumentiert, was zu einem Verlust über die Änderungshistorie führt.

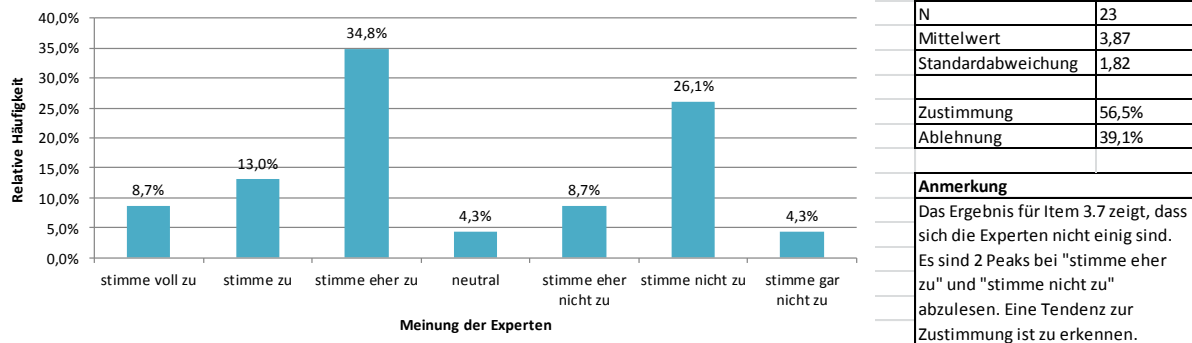


Figure 71: Item 3.7 – traceability of requirements

Item 3.8 In der agilen Softwareentwicklung gibt es im Vergleich zu klassischen Vorgehensmodellen keinen definierten Zeitpunkt, wann Requirements Engineering durchgeführt wird. Dieses stellt das Projektteam vor die Herausforderung, zu definieren, in welchen Rahmen und Umfang die Anforderungen aufgenommen werden.

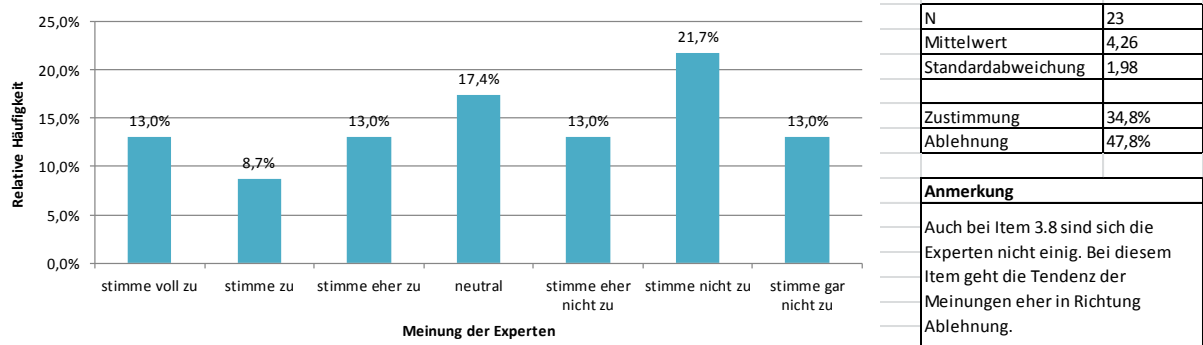


Figure 72: Item 3.8 – point of time for carrying out RE in agile projects

- (Experte 16) Auch hier möchte ich widersprechen, da es zur Agilen Entwicklung gehört die Anforderungen immer wieder festzustellen. Die definierten Zeitpunkte sind vor der Entwicklung durch PO und QA und nach dem Livegang durch Messungen und Tests.

Item 3.9 In der agilen Softwareentwicklung gibt es keine Methode mit der nicht-funktionale Anforderungen verbindlich festgelegt werden, wodurch die Kommunikation und Einhaltung von nicht-funktionalen Anforderungen problematisch wird.

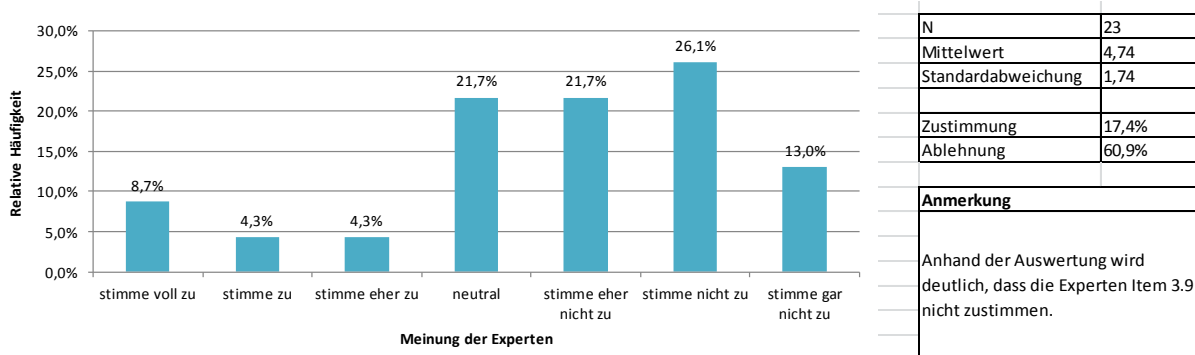


Figure 73: Item 3.9 – handling non functional requirements

- (Experte 16) Hier gibt es vielleicht keine Methode aber "Best Practises" die bekannt sind und auch angewandt werden.
- (Experte 23) es gibt verschiedene Ansätze diese Anforderungen festzuhalten (DoD, Smoketests, Lasttest, Penetrationstest, etc.) und bei der Entwicklung zu berücksichtigen

Item 3.10 In der agilen Softwareentwicklung müssen Anforderungsdokumente so gestaltet sein, dass sie sich mit vertretbarem Aufwand an sich ändernde Umgebungsfaktoren anpassen lassen.

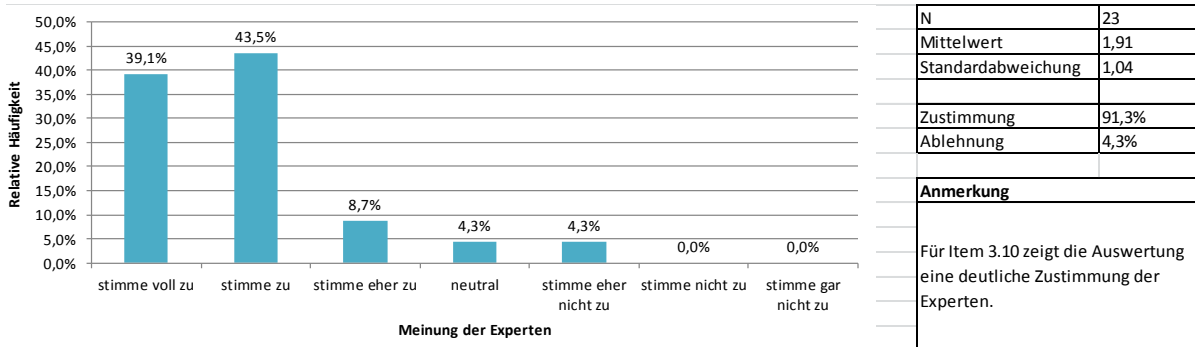


Figure 74: Item 3.10 – changes in requirements documentation

Allgemeine Anmerkung zu Requirements Engineering Methoden

- (Experte 15) Ein solides und leichtgewichtiges RE ist eine Grundvoraussetzung für agile Entwicklung. Ohne dass, wird es schnell chaotisch.
- (Experte 19) Viele der Punkte zeigen allgemeine Probleme in der Softwareentwicklung auf und haben nichts mit agil/nicht-agil zu tun. Auch agile Projekte lassen sich wunderbar dokumentieren und es steht auch nirgends, dass keine Styleguides, Prototypen, etc. verwendet werden dürfen. Im Gegenteil.

- Iterationsplanung und Aufwandschätzung -

Item 4.1 In der agilen Softwareentwicklung wird der Fokus auf die Detaillierung der Anforderungen für die zeitnahen Iterationen gelegt, um flexibel auf den weiteren Erkenntnisgewinn reagieren zu können.

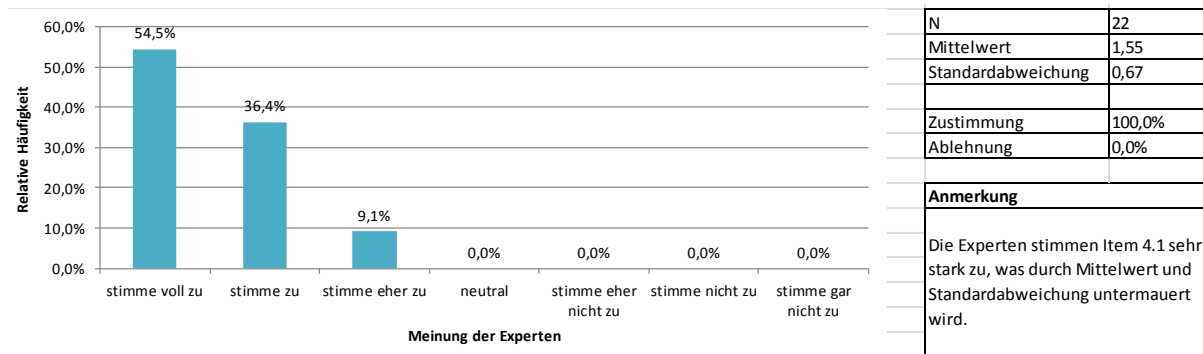


Figure 75: Item 4.1 – detailing requirements for short-term iterations

Item 4.2 In der agilen Softwareentwicklung sind die umzusetzenden Anforderungen zum Entwicklungsstart nicht immer klar definiert, da sich die Prioritäten oftmals kurzfristig ändern. Dadurch müssen Teams mit unklaren Anforderungen arbeiten.

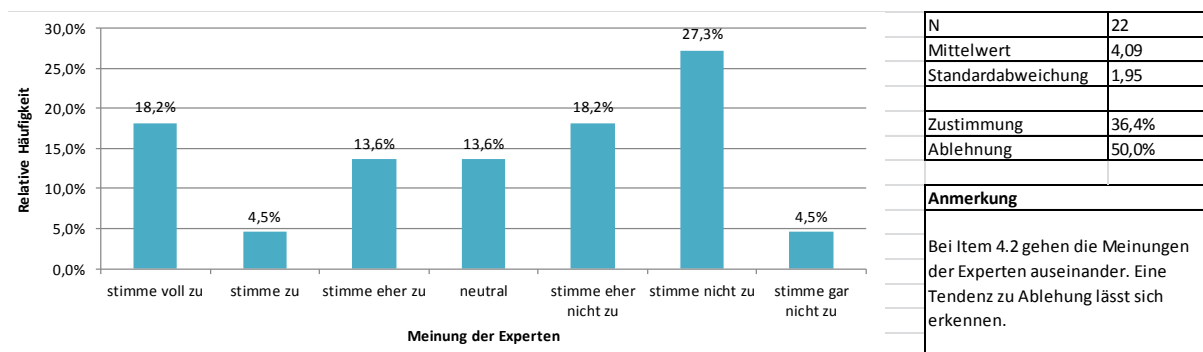


Figure 76: Item 4.2 – short-term changes in priorities of requirements

- (Experte 9) erster Satz stimmt, zweiter Satz stimmt nicht.
- (Experte 23) Bevor mit der Umsetzung begonnen wird, sollten die Anforderungen klar sein. Es müssen nicht alle Details bekannt sein, aber das "Was" muss bekannt sein.

Item 4.3 In der agilen Softwareentwicklung ist ein Ausblick auf die nächsten Iterationen für die Abstimmung mit angrenzenden Projekten wichtig. Jedoch darf der Ausblick nicht als verbindlich angesehen werden, um flexibel auf Änderungen reagieren zu können.

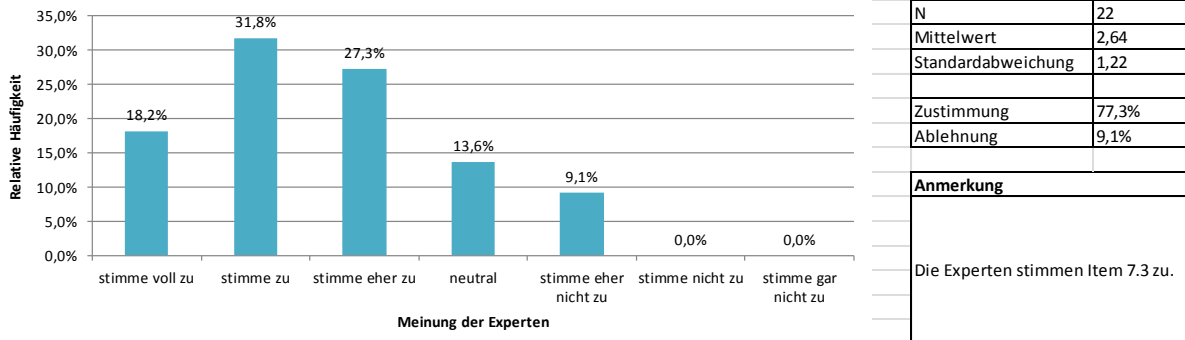


Figure 77: Item 4.3 – outlook on upcoming iterations

- (Experte 23) Es muss schon eine gewisse Verbindlichkeit gegeben sein, da sonst die Synchronisation der Anforderungen der Projekte untereinander schwierig wird.

Item 4.4 In der agilen Softwareentwicklung dürfen Aufwände nicht in Stunden geschätzt werden, um die verfügbaren Zeiten eines Teams nicht direkt mit der Summe geschätzter Stunden zu vergleichen.

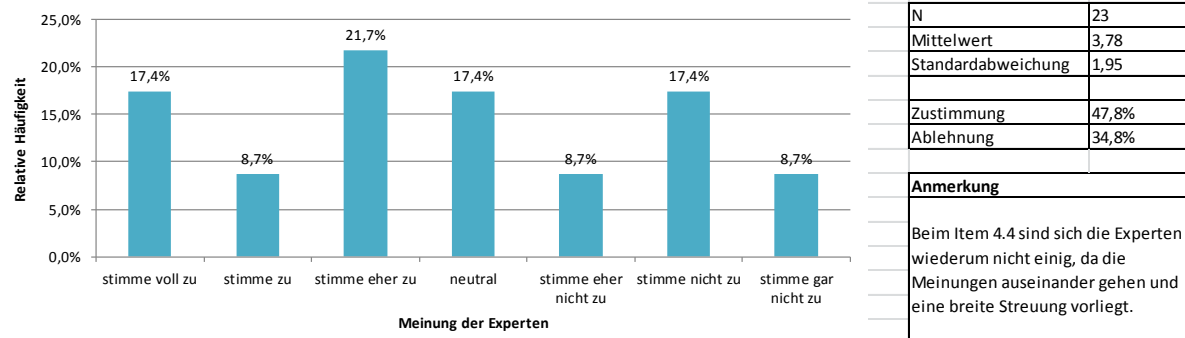


Figure 78: Item 4.4 – avoiding effort estimation in hours

- (Experte 18) es wird die Komplexität einer Anforderung geschätzt und nicht der Aufwand. Aufwand in Stunden ist eine sehr personenabhängige Schätzung. Bei der Betrachtung von "Komplexität" soll die personenabhängige Perspektive verringert bzw. eliminiert werden.
- (Experte 23) Egal in welcher Einheit Anforderungen geschätzt werden. Diese Einheit lässt sich immer in Stunden pro Einheit umrechnen.

Item 4.5 In der agilen Softwareentwicklung müssen Anforderungen so geschnitten sein, dass sie sich innerhalb einer Iteration umsetzen lassen und trotzdem einen Mehrwert für das Produkt bieten.

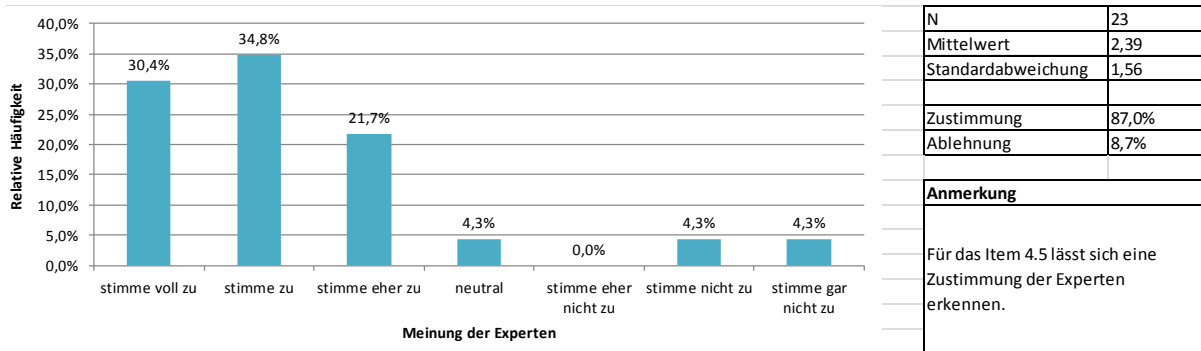


Figure 79: Item 4.5 – slicing requirements to fit into iteration

- (Experte 7) Die Aussagen beziehen sich m.E. nur auf timeboxed Ansätze wie Scrum, nicht auf agile Entwicklung an sich.

Item 4.6 In der agilen Softwareentwicklung sind Anforderungen teilweise so komplex, dass sie sich nicht innerhalb einer Iteration umsetzen lassen. Wenn sie über mehrere Iterationen hinweg umgesetzt werden, darf der Blick auf das Gesamtbild nicht vernachlässigt werden.

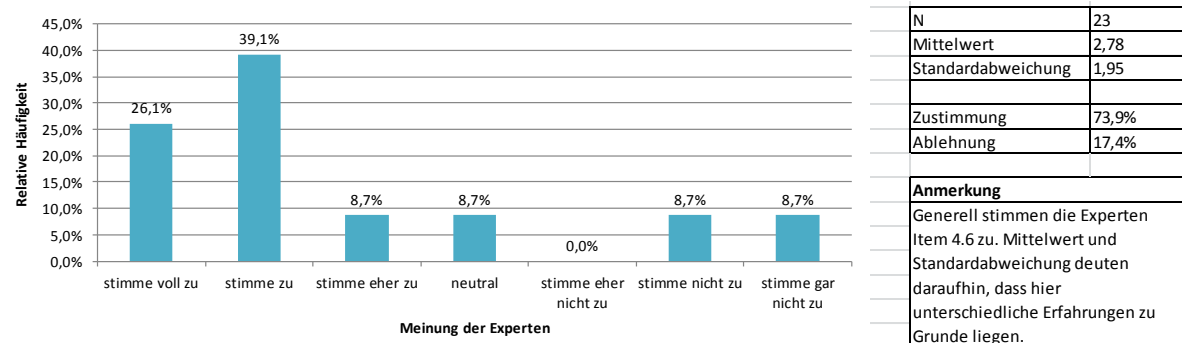


Figure 80: Item 4.6 – losing sight of the big picture due to complex requirements

- (Experte 7) Die Aussagen beziehen sich m.E. nur auf timeboxed Ansätze wie Scrum, nicht auf agile Entwicklung an sich.

Allgemeine Anmerkung zu Iterationsplanung und Aufwandschätzung

- (Experte 2) Für Scrum sind diese Fragen unter 4 korrekt, für Kanban nicht relevant
- (Experte 16) Es gibt viele verschiedene Meinungen zum Thema Aufwandsschätzung. Das bekannte Planningpoker ist allerdings schon lange überholt. Auch T-Shirtgrößen sind auf einem absteigenden Ast. In der Webentwicklung wird darauf geachtet, dass größere Projekte immer in viele kleine Tasks aufgeteilt werden, damit jederzeit abgebrochen oder unterbrochen werden kann. Das Gesamtbild ist dadurch auch zu jedem Zeitpunkt feststellbar.

- Format der Anforderungen -

Item 5.1 In der agilen Softwareentwicklung entsteht bei fachlichen oder technischen Abhängigkeiten zu anderen Teams erheblicher Koordinationsaufwand.

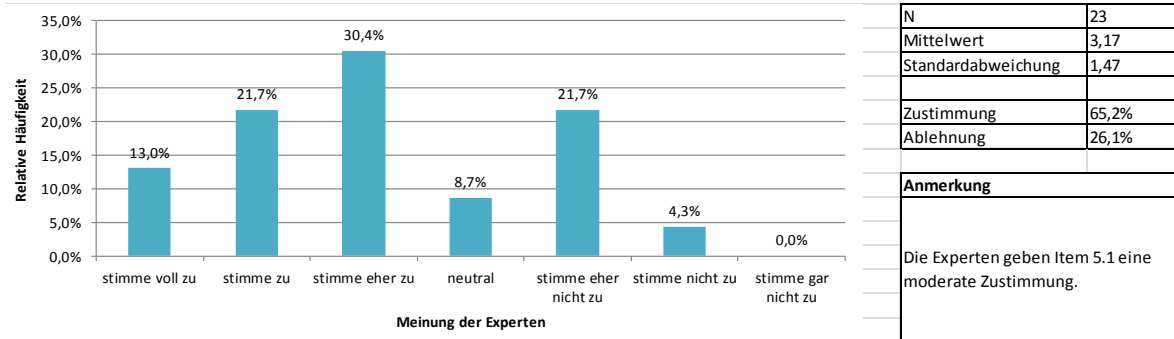


Figure 81: Item 5.1 – coordination effort due to dependencies

Item 5.2 In der agilen Softwareentwicklung müssen Anforderungen im Hinblick auf die bisherige Umsetzung analysiert werden, um Wechselwirkungen zu vermeiden.

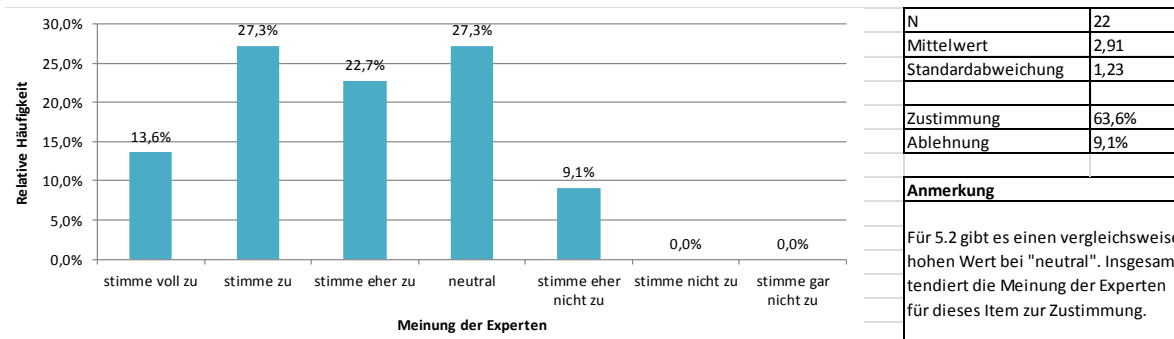


Figure 82: Item 5.2 – analyzing requirements to avoid interdependencies

Item 5.3 In der agilen Softwareentwicklung müssen Anforderungen als Ziele formuliert werden, die den Problemraum beschreiben, damit die Kreativität in der Lösungsfindung nicht eingeschränkt wird.

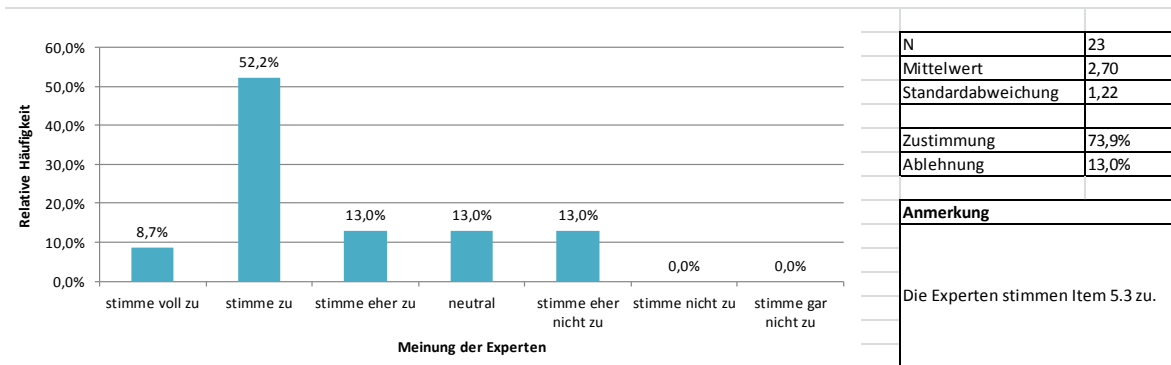


Figure 83: Item 5.3 – describing requirements in terms of objectives

Item 5.4 In der agilen Softwareentwicklung müssen Anforderungen so erfasst werden, dass zur Qualitätssicherung detaillierte Testfälle daraus abgeleitet werden können.

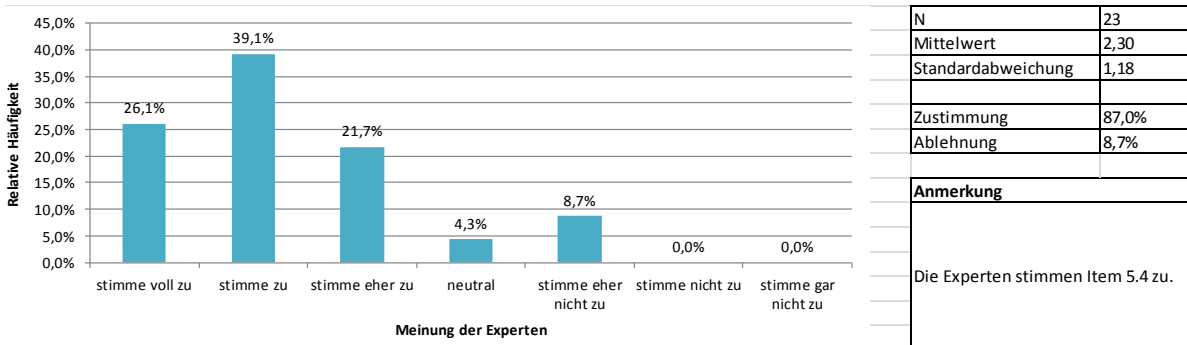


Figure 84: Item 5.4 – deviating test cases from requirements

Item 5.5 In der agilen Softwareentwicklung müssen Anforderungen klar formuliert und verständlich sein, um Unsicherheit in der Umsetzung zu vermeiden.

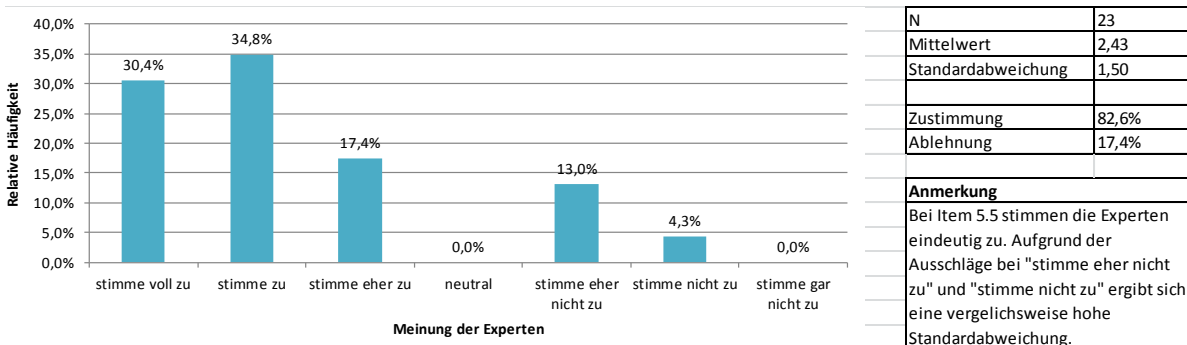


Figure 85: Item 5.5 – clarity of requirements

Item 5.6 In der agilen Softwareentwicklung muss zu den Anforderungen eine Begründung des Nutzens formuliert werden, damit der Mehrwert der Umsetzung klar wird und Entscheidungen für eine bestimmte Anforderung nachvollziehbar sind.

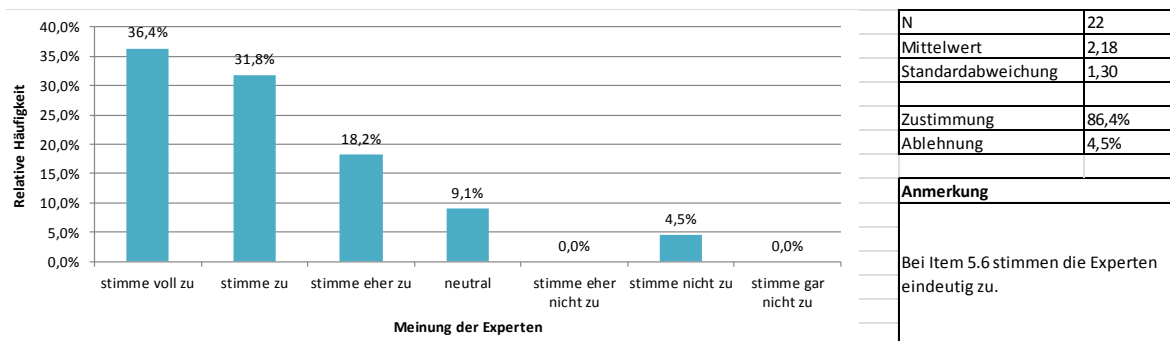


Figure 86: Item 5.6 – reasoning about benefits of a requirement

- (Experte 23) Es muss kein Nutzen formuliert werden, er hilft aber wie beschrieben Anforderungen nachzuvollziehen.

Allgemeine Anmerkung zu Format der Anforderungen

- (Experte 11) Viele Punkte treffen aber nicht nur auf "agile Projekte" zu. Klare und verständliche Anforderungen sind auch in "klassischen" Projekte essentiell :-)
- (Experte 12) Diese Aussagen gelten aber alle doch nicht nur für agile Projekte. Für mich sind dies keine Besonderheiten von agiler Softwareentwicklung.
- (Experte 15) Nicht-Funktionale Anforderungen müssen schon sehr klar und lösungsorientiert sein. Wenn z.B. eine Performance-Anforderung zu viel Spielraum lässt, wird sich das am Ende in einer schlechteren Performance niederschlagen.

Allgemeine Anmerkungen zur Umfrage

- (Experte 7) Viele Aspekte ändern sich im Laufe der Zeit und viele Aspekte hängen von den Erwartungen konkreter Auftraggeber/ den Entscheidern ab, die außerhalb des Teams arbeiten. Dennoch habe ich mich bemüht, aus der Perspektive "üblicherweise" zu antworten.
- (Experte 15) Bin gespannt auf die Ergebnisse :-)
- (Experte 20) Gute Fragen, die ein gutes Bild zur Agilität abgeben
- (Experte 24) Viel Erfolg :-)
- (Experte 26) 80-90% aller Aussagen im Fragebogen gelten genauso für klassische Entwicklungsmethoden.

Stakeholder und Nutzer

Item C5 In der agilen Softwareentwicklung ist es eine Herausforderung Benutzeranforderungen und Nutzungsqualitäten in Zusammenarbeit mit direkten Nutzern (end user) des Produkts zu erarbeiten.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Gilt dann, wenn bestehende Anwendungen digitalisiert, ausgebaut oder optimiert werden. Bei der völligen Neuentwicklung stellt sich die Frage ob die notwendigen Impulse von den Nutzern (der bisherigen Anwendung) kommen können. Um es mit den Worten von Henry Ford zu sagen: "If I had asked people what they wanted, they would have said faster horses."
5	Schnelles Veröffentlichen einer ersten Version, welche dann kontinuierlich verbessert wird. Zurückspielen der Erkenntnisse durch ein vorgelagertes UX-Board.
7	Bislang habe ich nur in zwei Projekten Nutzungstests durchführen können, wobei es sich in einem Fall nicht um echte Nutzer, sondern nur um Projektexterne handelte. Daher kann ich keine Empfehlung aussprechen.
9	Typische Verfahren aus dem Requirements Engineering - z. B. Hospitation bei der Arbeit der Nutzer
11	Problem ist nur manchmal, dass die Benutzer ihre Anforderungen ja selbst gar nicht so genau kennen. Daher Hypothesen aufstellen und diese dann an "echten" Nutzern verproben. Ggf. auch nur unter Laborbedingungen.
12	Usability Tests im Labor
13	Regelmäßige A/B-Tests und die Aufzeichnung und Auswertung von Nutzerverhalten. Einsatz von UX-Labs.
15	User Research-Methoden auf die Belange der agilen Entwicklung anpassen, z.B. * Methoden auf das Nötigste reduzieren * Auswertung im Team, keine Berichte * Finanzielle Hemmschwelle für Nutzereinbezug senken * Zugangsschwierigkeiten zu Nutzern verringern (z.B. durch Vorrekutierung oder Panels)
20	Interview und Nutzer bei der Arbeit beobachten und laut denken lassen, bei Nutzung des Systems welches z.B. abgelöst werden soll.
22	Feedback von echten Kunden durch User Labs, Prototypen, Friendly User Tests, Alpha/Beta/Silent Launches
23	Frühe Einbindung in den Produktentwicklungsprozess. Erstellung von Prototypen. Usability test.
24	Clickdummies oder Demos

Item C6 In der agilen Softwareentwicklung ist es eine Herausforderung Stakeholder während der gesamten Entwicklung in regelmäßigen Iterationen einzubinden, damit die Produktentwicklung erfolgreich wird.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
2	Kommt ganz auf das Projekt an :-) Alle Antworten vorher kommen auch total auf das Projekt an ;-)
4	Regelmässige Reviews, Demonstrationen oder Roadshows, ermöglichen von direktem Kontakt ohne Umweg über die oberen Hierarchie-Ebenen.
5	Hängt ganz stark vom Typus des Stakeholders ab.
6	Vorab sollten regelmäßige Jour Fixe verabredet werden, deren Agenda klar vorstrukturiert ist. Gerade am Anfang ist es sehr wichtig, dass kein Stakeholder sich übergangen fühlt - es ist nach meiner Erfahrung einfacher, die Runde schrumpfen zu lassen als sie im Nachhinein zu vergrößern
7	Reviews o.ä. durchführen und Stakeholder einladen. Dabei ist es wichtig, ein ansprechendes Format zu wählen und die Stakeholder im Vorfeld darüber zu informieren, warum die präsentierten Ergebnisse jeweils relevant für sie sind. Erfahrungsgemäß ist das Stakeholder-Engagement eher gering.
9	Einladung zu Sprint-Reviews.
11	Vorab ist zunächst zu klären, wer genau die Stakeholder sind und in welchen Intervallen sie ggf. hinzugezogen werden sollten.
13	Teilnahme an regelmäßigen Planungs- und Review-Terminen um Feedback und nützliche Informationen aus „erster“ Hand zu erfahren.
19	Stakeholder in Sprint-Reviews einladen und Inputs direkt zu berücksichtigen. So haben sie das Gefühl, ihre Mitarbeit ist wichtig und hat direkten Einfluss auf das Produkt.
20	Dem PO auf die Wichtigkeit sein Rolle hinweisen und mit Regeltermin zu den Scrum-Meeting einbinden.
22	Ziele definieren statt Lösungen vorschreiben, regelmäßige gemeinsame Reviews der Ergebnisse und Updates des Backlogs durchführen
23	Regelmäßige Reviews des entwickelten Produktes. Aktives Einholen/Einfordern von Feedback.
24	Nutzen für unterschiedliche Stakeholdergruppen klar machen. Z.B. Das Management interessiert sich dafür, wie und wann die Vision & Strategie in Features umgesetzt werden. Die Einbindung während des Entwurfs der Features und in der Review Sessions wird dann einfacher. Zweck der einzelnen Entscheidungen (strategische und taktische) transparent machen. Diesen mit den Stakeholdern besprechen und hoffen, dass sie irgendwann mitdenken :-)

Item C2 In der agilen Softwareentwicklung ist es eine Herausforderung, dass Stakeholder verstehen, dass das Entwicklungsteam selbständig (Detail-) Entscheidungen treffen darf.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Kommt stark auf den Kontext an, lässt sich so nicht beantworten.
6	Die Beteiligung der Stakeholder sollte eher in Form einer Informationsveranstaltung stattfinden - im Fokus sollte die Auswertung der erzielten Mehrwerte resp. der zu verzeichnenden Misserfolge und der Konsequenz für die weitere Produktentwicklung stehen. Konzepte, Entwürfe und Dummies sollten mit Bedacht präsentiert werden - das Produkt selbst dagegen häufig. Sofern es komplett neue Anforderungen gibt, sollten diese ggf. außerhalb des regelmäßigen (kurzen) Jour Fixe, aber in gleicher Runde, diskutiert werden.
7	Wichtiger ist es, einen PO zu haben, der tatsächlich entscheidungsbefugt ist, auf dieser Basis dem Team Spielraum gibt und die Entscheidungen anschließend auch vertritt.
8	regelmäßige enge Abstimmung mit den Stakeholdern, Präsentation von Lösungsvorschlägen
11	Das ist schon sehr schwierig. Hier muss man glaube ich mit gutem Beispiel vorangehen und verschiedene Lösungsalternativen durch das Team vorschlagen lassen, wie sich ein Feature umsetzen lässt.
12	Gutes Erwartungsmanagement gegenüber den Stakeholdern durch den PO hilft hier.
15	* Immer wieder Feedback an die Stakeholder geben, welche Auswirkungen es hat, wenn sie sich in Detailentscheidungen einmischen. * Scrum Master (wenn vorhanden) als Coaches für Stakeholder
19	Die Expertise des Stakeholders im fachlichen Bereich stärker herausstellen und in Reviews auf die Aufteilung der Concerns hinweisen. Das Team zwar Detailentscheidungen treffen lassen, den Stakeholder jedoch immer transparent über die Gründe aufklären.
20	PO bestimmt das „Was“ das Team das „Wie“
22	Stakeholder nach Zielen befragen und verschiedene Varianten für die Lösung anbieten, um aufzuzeigen, dass „seine“ Lösung eine von mehreren möglichen ist.
24	Coaching von Stakeholdern: den Zweck hinter Teamselbstständigkeit präsentieren und erleben lassen. Mit Beispielen zeigen, was in der Vergangenheit schiefgelaufen ist, als das Team nicht selbstständig entscheiden durfte.

Anmerkungen zum Thema 1 „Stakeholder und Nutzer“

- (Experte 4) Nicht jede Software hat einen (menschlichen) Nutzer. Z.B. agieren im Finanzbereich (Banken, Versicherungen, Aktienhandel) Anwendungen weitgehend autonom.
- (Experte 6) Die Balance zwischen Information und Nicht-Information der Stakeholder und deren Detailtiefe ist ebenso eine Gratwanderung wie die Beteiligung der Endanwender und der Entscheidung, bestimmte Wünsche umzusetzen oder eben nicht.
- (Experte 7) 1.1 Es ist immer eine Herausforderung den end user einzubeziehen, das ist keine Herausforderung, die speziell aus der agilen Vorgehensweise resultiert.
- (Experte 9) In unterschiedlichen Situationen fallen die Antworten unterschiedlich aus. Ich habe hier eine Tendenz angegeben, die nicht immer zutreffen muss. 1.1. trifft auch auf plangetriebene Entwicklung zu.

- (Experte 15) „Nicht-Funktionale Anforderungen sind in agilen Softwareentwicklungsprozessen genauso schwer „durchzusetzen“, wie in anderen Prozessen. Wenn es in einem Projektteam zu viele Techniker gibt, dann fallen Benutzeranforderungen und Nutzungsqualitäten schnell mal hinten runter. Einbindung von Anwendern und Stakeholdern ist dann kein Problem, wenn man dafür eine entsprechende Infrastruktur und Vorgehensweise geschaffen hat"
- (Experte 20) Zu 1.3. In der agilen Softwareentwicklung ist es eine Herausforderung, dass Stakeholder verstehen, dass das Entwicklungsteam selbständig (Detail-) Entscheidungen treffen darf. -> Wenn mit (Detail-) Entscheidungen damit das „Wie“- der Umsetzung gemeint ist, dann ein ja von mir.
- (Experte 21) Leider weiß ich nicht, was du mit „allgemeinen Herausforderungen“ verstehst. Deshalb meine Antwort zu „bin mir unsicher“.
- (Experte 26) „Für mich sind dies keine Herausforderungen der agilen Softwareentwicklung. In der klassischen SW-Entwicklung steht man vor den gleichen Herausforderungen.
zu 1.1) Kontakt zum Enduser ist immer schwer oder nur abstrakt gegeben.
zu 1.2) Ob die Iteration 2 Wochen - 4 Woche wie im agilen ist oder nach jedem Release nach 6 Monaten es ist immer schwierig die Stakeholder für die benötigte Zeit einzubinden.
zu 1.3) Auch wenn in der klassischen SW-Entwicklung mehr oder ausführlichere Inkremente erstellt werden sind diese nicht so detailliert, dass die Entwickler keine Entscheidungsfreiraum haben, weil dann könnte man die Sourcecode direkt selberschreiben. ^ Des Wegen sind für mich alle drei Aussagen mit „Nein“ zu beantworten, weil diese genauso für die klassische SW-Entwicklung gelten."

Anforderungsmanagement

Item C4 In der agilen Softwareentwicklung ist eine kontinuierliche Pflege der Anforderungen eine Herausforderung, da zu Beginn nicht alle Anforderungen feststehen und diese sich über den Projektverlauf ändern.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
2	Refinement
4	An dieser Stelle ist die Antwort sehr einfach: Man muss es einfach tun.
5	Regelmäßiger Austausch mit allen Beteiligten.
7	Ständige Pflege des Product Backlogs.
8	Enge Zusammenarbeit mit den Anforderern, Kommunikation mit dem Team
10	Kano-Analyse, Theme Screening & Scoring, relative Gewichtung
11	Backlog grooming und mit dem Team diskutieren, ob Anforderung die Reife für die Umsetzung haben.
19	Im Produkt-Backlog müssen die Anforderungen bereits grob umrissen und priorisiert sein. Für das Sprint-Backlog müssen die Anforderungen detailliert und mit Begründung beschrieben werden. Sowohl Produkt- als auch Sprint-Backlog stehen ständig auf dem Prüfstand.
20	Prioritäten der Anforderungen ändern sich. Neue Anforderungen kommen hinzu die oftmals „super“ wichtig sind und schnell umgesetzt werden sollen. Das Team braucht genug Input das es sind die Anforderung in einem Sprint nicht umsetzen kann und es kein „done“ gibt.
22	regelmäßiges Review der Ergebnisse und Update des Backlogs mit Stakeholdern
24	Ich würde davon abraten Anforderungen zu früh zu schreiben, weil diese sich schnell ändern und Pflege wird mandatorisch. Grooming ist dafür da um Details zu besprechen und Entscheidungen zu treffen. In Kanban Arbeitsmodus (Grooming on Demand) wird die Anforderung sehr zeitnah entwickelt. In Scrum wird dies erst im folge Sprint. Neue Erkenntnisse und Risiken können erst im Grooming transparent werden. Dafür werden Spike Stories erstellt um die offenen Punkte zu klären. Dann wird die Anforderung asap umgesetzt. Pflege ist nicht notwendig. Spätere Änderung kann selbstständig dokumentiert (z.B. in Jira Ticket) werden. Das Ticket kann mit dem ursprünglichen Ticket verknüpft werden.
25	Die kontinuierliche Pflege der Anforderungen lässt sich durch Ticketing-Systeme wie z.B. Jira bewältigen, in dem pro Anforderung ein Ticket erstellt wird, auf das alle Stakeholder von der Erfassung über die Entwicklung, den Test bis hin zur Abnahme zugreifen können. Die Voraussetzung dafür ist das schnelle und einfache Auffinden von den zu pflegenden Anforderungen, was durch Tagging und Filtermechanismen, Gliederung der User Stories nach Epics erreicht werden kann. Mit den Möglichkeiten des Lifecycle-Managements, lässt sich der Status von Anforderung pflegen. Zusätzlich können Verknüpfungen zu anderen Anforderungen erstellt werden durch Verlinkungen, Anlegen von Subtasks etc. Ticketing-Systeme erlauben zusätzlich die Pflege von Scrum- und Product-Backlogs.

Item C12 In der agilen Softwareentwicklung ist es eine Herausforderung Änderungen der Anforderungen nachvollziehbar zu dokumentieren.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Auch an dieser Stelle ist die Antwort sehr einfach: Man muss es einfach tun.
13	Meiner Erfahrung nach nur bei Dienstleistungsverträgen und sicherheitskritischen Umgebungen wirklich relevant.
15	Wir arbeiten direkt mit den Anforderungen im Microsoft TFS. Es gibt keine gesonderten Dokumente mehr. In Verbindung mit einer Schablone für die Formulierung von Anforderungen, Definitions of Done / Ready und den Log-Funktionalitäten des TFS kriegen wir schon eine ganz passable Nachvollziehbarkeit hin ohne dass dafür ein gesonderter Dokumentationsakt erforderlich wäre.
20	Ist eher unwichtig, aber letztendlich ist es wichtig das die Anforderung den Stakeholdern und Nutzern entspricht. Gibt da Ansätze aus CPRE, wie man damit umgehen soll und Historie betrieben kann. Wichtig ist nicht die Historie zur Anforderungen, sondern das man bei ständigen Anpassungen nicht zu viel Geld verbrennt und es ein „Done“ gibt.
25	Es ist abhängig vom Projekt, in welchem Detaillierungsgrad Änderungen nachvollzogen werden müssen und ob es sich dabei um eine Herausforderung handelt. Zur Lösung können sich die Stakeholder auf ein Attributierungsschema und einen Prozess zur Anforderungsänderung einigen.

Item C8 In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungen im Hinblick auf die bisherige Umsetzung zu analysieren, um Wechselwirkungen zu vermeiden.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Auch an dieser Stelle ist die Antwort sehr einfach: Man muss es einfach tun.
13	Insofern wichtig, dass klar sein muss, wenn die Umsetzung neuer Anforderungen alten Anforderungen die ebenfalls noch bestehen, entgegenstehen. Meiner Erfahrung nach nur bei Dienstleistungsverträgen und sicherheitskritischen Umgebungen wirklich relevant. Es hilft automatisierte Tests mit Hinweis auf eine Anforderung zu haben.
15	Das birgt das Risiko, dass die bisherige Umsetzung einfach kopiert wird.
19	Siehe oben. Die bisherige Umsetzung in Verbindung mit Stakeholder-Input und UX-Tests ist das Maß der Dinge. Auch wenn dies mit an Sicherheit grenzender Wahrscheinlichkeit Änderungen zur Folge hat.
21	Ich kann an dieser Stelle nur für das Produkt sprechen, an dem ich im Bereich Netzwerksicherheit arbeite. Da hier viele Umsetzungen von Anforderungen Sicherheitslücken an anderer Stelle auslösen können, hat dies bei uns sehr hohe Priorität.
25	Der Einsatz eines Ticketing-System bietet Aufschluss über den Status der Umsetzung einer Anforderung durch das integrierte Lifecycle-Management. Der Status einer einzelnen Anforderung fließt ein in Burndown-Charts, womit Aussagen über die Umsetzung von mehreren Anforderungen oder Teilzielen getroffen werden können.

Item C14 In der agilen Softwareentwicklung ist es eine Herausforderung den Fokus nur auf die Detaillierung der Anforderungen für die zeitnahen Iterationen zu legen.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Ich würde grundsätzlich hinterfragen ob in agiler Softwareentwicklung Anforderungen detailliert sein können.
8	regelmäßige Iterationen mit Überprüfung des sich weiterentwickelnden Produkts
11	s.o.
12	Nur die Anforderungen für die anstehende Iteration (maximal für die zwei nächsten Iterationen) werden detailliert spezifiziert, alle anderen Anforderungen werden nur grob beschrieben.
15	Fokus ist der Schlüssel zum Erfolg.
19	Da sich die Priorisierungen und der Funktionsumfang ändern können (und werden) macht es keinen Sinn, Details im Voraus zu planen und zu beschreiben.
20	Sehr wichtig, das Team braucht genug Input das es die Anforderung im Sprint auch komplett umsetzen kann und es dazu ein „done“ gibt. Oftmals bemerkt ein Entwickler, dass Dinge vergessen oder Anforderungen hohe Umsetzungsaufwände erzeugen, aber mit leichten Umsetzungsänderungen sich diese minimieren lassen. Teilweise kommt zu keinem „done“ zum Sprintende und man muss sehen das die überschau bleibt und man Dinge trotzdem fertig bekommt.

Item C7 In der agilen Softwareentwicklung ist es eine Herausforderung, dass die umzusetzenden Anforderungen zum Entwicklungsstart klar definiert sind, da sich die Prioritäten oftmals kurzfristig ändern.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
2	Planning, weiterhin Möglichkeit zur Rückfrage mit dem PO
4	Wenn man das macht ist es nur dann agil, wenn alle Anforderungen sich jederzeit ändern können.
5	Für alle transparenter Ablauf, regelmäßiger Austausch mit allen Beteiligten.
7	Es muss ein gemeinsames Verständnis davon geben, welchen Detailierungsgrad eine Anforderung benötigt, damit die Umsetzung starten kann (Beispiel Definition of Ready, wobei diese m.E. auch implizit vorhanden sein kann).
8	enge Abstimmung mit dem Team bei der Formulierung der Anforderungen (ggf. Überprüfung und Anpassung)
12	Bei Scrum wird dies durch das Sprint Planning gelöst, indem dort die Anforderungen im Zweifel durch das gesamte Team genauer beschrieben werden.
13	Für Lösungen braucht es einen Rahmen. Je eher dieser Rahmen klar, desto besser. Man kann sich ihm auch über die Zeit nähern, kostet unter Umständen etwas mehr. Agile Methoden sind ja genau dafür gemacht, flexibel auf neue oder geänderte Anforderungen einzugehen.
24	Bitte siehe das Kommentar oben
25	Bevor die Entwicklung mit der Umsetzung einer Anforderung startet, müssen die Anforderungen bei der Entwicklung bekannt und verstanden sein. Im Grooming können Unklarheiten geklärt werden, wodurch die Definition von Anforderungen Geschäft wird. Es ist jedoch Projektabhängig, wie klar eine Anforderung definiert sein muss.

Item C15 In der agilen Softwareentwicklung ist es eine Herausforderung, einen Ausblick auf die nächsten Iterationen zu entwickeln, ohne diesen als verbindlich anzusehen.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	API driven Development, Microservices o.ä. können Abhängigkeiten reduzieren. Die Übrigen können im Rahmen von Skalierungsframeworks wie LESS oder Nexus behandelt werden.
7	Der PO muss eine Road Map pflegen und teilen.
15	Idealerweise gibt es da keine großen Abhängigkeiten. Die Teams sollten soweit wie möglich eigenständig agieren können. Dazu muss die Systemarchitektur entsprechend angelegt sein. Zu viele Abhängigkeiten zwischen Teams führen zu einer Unbeweglichkeit aufgrund der Abstimmbedarfe.
19	Man sollte zwar nicht „auf Halde“ entwickeln und planen, aber besonders in offenen Systemen ist ein Big Picture unumgänglich.
20	Die sollte Thema im Releaseplanning sein siehe (Scaled Agile Framework SAFE)
22	regelmäßiges Review der Ergebnisse und Update des Backlogs mit Stakeholdern, ggf. Update der Roadmap
25	Dieses Problem hatte ich in der Praxis bisher noch nicht.

Item C1 In der agilen Softwareentwicklung sind fachliche oder technische Abhängigkeiten zu anderen Teams eine Herausforderung, da erheblicher Koordinationsaufwand entsteht.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	API driven Development, Microservices o.ä. können Abhängigkeiten reduzieren. Die Übrigen können im Rahmen von Skalierungsframeworks wie LESS oder Nexus behandelt werden.
5	Ebenfalls regelmäßiger Austausch mit allen Beteiligten sowie ein guter Projektmanager.
6	Frühzeitig Ressourcen anfordern, während der Umsetzung möglichst täglich Rücksprache mit den Bearbeitern halten. Dies ist insbesondere dann wichtig, wenn die Bearbeiter im Tagesbetrieb erledigen, sofern sie komplett auf Auftragsarbeit geplant sind, ist eine Bearbeitung nach Plan einfacher.
7	Es muss geteiltes Wissen darüber bestehen, welches Team an welchem Produkt arbeitet und welche Themen sich gerade in der Umsetzung befinden und als nächstes anstehen. Product und Sprint Backlogs müssen entsprechend Team-übergreifend zugänglich sein.
9	Dies ist oftmals die größte Herausforderung, da andere Bereiche meistens nicht agil arbeiten und Planungsvorläufe erwarten. Ideal wäre die (temporäre) Einbindung eines Vertreters des anderen Teams; ansonsten hilft nur lange Vorplanung mit entsprechenden Risiken und Einschränkungen der Agilität. Falls die anderen Teams ebenfalls agil arbeiten sollten, helfen die Skaliermodelle, z. B. LESS.
12	Da fehlende Koordination zu anderen Teams zu teilweise erheblichen Zeitverzögerungen führt, müssen solche Abhängigkeiten entweder vermieden werden oder durch regelmäßigen Austausch mit anderen Teams und längerfristiger Planung der Iterationen vorbereitet werden.
13	Wichtig ist, dass die Beteiligten miteinander reden und ein gemeinsames Bild entwickeln und pflegen. Wenn das aufdecken und auflösen solcher Abhängigkeiten eintrainiert ist, stellt dies nur kleinere Hindernisse dar.
15	Siehe oben

19	Die Teams in Matrixform zu organisieren, also gemischte Teams aus den verschiedenen Gruppen. Falls dies nicht möglich ist, Scrum of Scrums, also ein wöchentliches Koordinations-Standup organisieren.
20	Da spielt frühzeitiges integrieren der Entwicklungsstände "CI" eine wichtige Rolle. Technische Abhängigkeiten sind eher zu identifizieren. Fachliche Abhängigkeiten schon schwieriger, die oftmals auch größere Aufwände erzeugen können, wenn man sie nicht im Blick hat. Hier auch wie oben im Release Planning (SAFE) die früh identifizieren und gute fachliche Schnitte der Architekturkomponenten, die von jeweiligen Teams umgesetzt werden, finden.
25	Z.B. durch Einrichtung von Gruppen zu wie Communities of Practices zu Team-übergreifenden Themen, durch regelmäßige gemeinsame Meetings mit Vertretern des Teams z.B. wie Architekturboards etc.

Anmerkungen zum Thema 2 „Anforderungsmanagement“

- (Experte 9) Herausforderung interpretiere ich hier als „Problem“. Problematisch sind die Punkte in der Regel nicht, müssen aber gekannt und beachtet werden.
- (Experte 16) zu 2.7 Ich würde hier sogar fast Nein sagen, weil Koordinationsaufwand nicht unbedingt durch die agile Entwicklung höher ist. Hierbei kann es allerdings zu Konflikten in der Priorisierung kommen. Während das eine Team sich ein Feature als nächstes Ziel gesetzt hat, hat das Team, welches Abhängigkeiten bereitstellt müsste diese als deutlich niedriger eingestuft und deswegen noch nicht gemacht.
- (Experte 23) Zu 2.6.: Die Herausforderung besteht darin allen Projektbeteiligten klar zu machen wo die Reise hingehen soll (kurz, mittel und langfristig) und sicherzustellen, dass alle diese Ziel verstanden habe und ihr Handeln dahingehen ausrichten, nicht nur auf die nächsten Iterationen bezogen.
- (Experte 24) Wenn ich „bin mir unsicher“ wähle, meine ich, dass es auf die Umgebung/Kontext kommt. Es kann ja oder nein sein.
- (Experte 26) Für mich gelten die Herausforderungen genauso für die klassische Entwicklung. Des Wegen „Nein“ keine Herausforderungen nur für die agile Softwareentwicklung.
zu 2.1) Über Change-Anforderungen oder Releases müssen in der klassischen Entwicklung Anforderungen nachgepflegt werden und es muss für Entwicklung und Test ersichtlich werden für Wann die Änderungen geplant sind, welche Status die Änderungen haben etc.
zu 2.5) Welchen Zusammenhang gibt es zwischen der Priorität und unvollständige Anforderungsdefinition. Insbesondere gibt es im agilen das „Planning“ in dem die Details noch nach justiert werden können.

Methoden und Artefakte

Item C20 In der agilen Softwareentwicklung ist es eine Herausforderung, dass die Erhebung und Evaluierung von Anforderungen im Projektkontext nicht schnell genug geht.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Kommt auf den Kontext an. Auch im agilen Vorgehen muss nicht alles sofort umgesetzt werden.
5	Viele kleine Nutzertests, statt großen, langwierigen Tests.
12	Sofern ein ausreichend umfangreiches Backlog mit bereits spezifizierten Anforderungen existiert ist eine schnelle Erhebung nicht erforderlich. Oft ist dies zu Projektbeginn jedoch anders, weshalb eine entsprechende Vorlaufzeit der Konzeption vor der Umsetzung empfehlenswert ist.
20	Oftmals ist der PO das „bottleneck“ und kommt seinen Aufgaben nicht nach, das Team kommt im Sprint ins stocken. Impediments entstehen beim Scrum Master.

Item C17 In der agilen Softwareentwicklung ist es eine Herausforderung Methoden zur Erhebung und Evaluierung von Anforderungen einzusetzen, bei denen die Erkenntnisse mit dem Entwicklungsteam geteilt werden.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Auch hier: kommt auf den Kontext an.
5	Ein geeignetes Darstellungssystem benutzen, z.B. internes Wiki.
19	Nur wenn das Team die Anforderungen und deren Nutzen kennt, ist eine sinnvolle Realisierung möglich. User stories sollten die Entscheidungsfindungswege für die Anforderungen enthalten und so dem Team zugänglich gemacht werden, um den Nutzen noch besser verstehen zu können.
20	Siehe CPRE, Clickdummy, Prototyping, User-Stories / Use-Cases
25	Nach der Anforderungserhebung und Evaluierung sollte der PO von einem Team-Sprecher eine grobe Bewertung zur Umsetzung einholen.

Item C16 In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungsdokumente so zu gestalten, dass sie sich mit vertretbarem Aufwand an sich ändernde Umgebungsfaktoren anpassen lassen.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
19	Statische Dokumente widersprechen agilen Methoden per se. Deshalb sind die Anforderungen in kleinen Paketen zu pflegen, bestenfalls in Cloud-basierten Tools wie Jira, Trello, o.ä.
20	Anforderungsdokumente nicht zu detailliert gestalten und die Software bzw. den Clickdummy auch als "Anforderungsdokumentation auf Detailebene" nutzen.
4	Lean Documentation, z.B. in Form von User Stories. Verzicht auf Feinspezifikationen.

Item C13 In der agilen Softwareentwicklung ist es eine Herausforderung nicht-funktionale Anforderungen verbindlich festzulegen.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Bei manchen (z.B. Lastfähigkeit) macht es Sinn, viele nichtfunktionale Anforderungen (Benutzbarkeit, Skalierbarkeit, Portierbarkeit, etc.) lassen sich aber kaum verbindlich festlegen.
9	Kommt auf die Qualitätsdimension an. Hierzu ein Artikel von mir: [Anonymisiert] ⁸
26	Was nützt mir die anspruchsvollste Software, wenn der Klick auf den „Weiter Button“ zum Absturz führt oder der Request zur DB mir Zeit zum Kochen eines Kaffees gibt?

Item C3 In der agilen Softwareentwicklung ist es eine Herausforderung bei der Umsetzung komplexer Anforderungen den Blick auf das große Ganze nicht zu vernachlässigen.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Erstmal definieren was das Große Ganze überhaupt ist. Erst dann kann man regelmässig überprüfen ob die aktuelle Entwicklung darauf einzahlt.
5	Regelmäßiges Einschwören der Beteiligten, alle Beteiligten über Änderungen informieren.
6	Zur Konzentration auf verwandte Themenbereiche/ Projekte mit Abhängigkeiten sollte ein zentraler Ansprechpartner für diese gefunden werden, so dass eine Abstimmung über Details oder auch Auswirkungen kurzfristig möglich sind.
9	Am Anfang muss immer eine Produktvision vorliegen, an die sich die Anforderungen orientieren.
11	Story-Mapping kann hier helfen, um die Verbindung der Stories besser zu verstehen.
12	Durch den Fokus nur auf die nächste Iteration passiert es leicht, dass man ein größeres Ziel aus den Augen verliert. Hier muss man regelmäßig „einen Schritt zurück“ machen und sich das gesamte Produkt betrachten, um dadurch die Prioritäten richtig zu setzen.
15	Visualisierung der Customer Journey zu Beginn des Projektes und immer wieder Kundeneinbezug, um das Problem, dass es zu lösen gilt nicht aus dem Blick zu verlieren.
19	Der Produkt Owner muss immer das Gesamtprojekt im Fokus behalten um keine sich ausschließenden Anforderungen zu kreieren.
20	Die Prioritäten der Items des gesamten Backlogs sollten nach dem Hauptziel ausgerichtet sein
22	Roadmaps, Systemskizzen, Prozessskizzen, Wertströme
25	Beim Projektstart können Epics oder Teilziele definiert werden, für die jeweils eine User Story Punktzahl vergeben wird für die Umsetzung. Bei Sprintwechseln müssen die Punktzahlen angepasst werden.

⁸ Link wurde aus Gründen der Anonymität von den Autoren entfernt

Anmerkungen zum Thema 3 „Methoden und Artefakte“

- (Experte 13) Wenn ich mit NEIN geantwortet habe, meine ich damit, dass es keine besondere Herausforderung ist, sondern die agilen Methoden dafür da sind, diesen Herausforderungen erfolgreich zu begegnen. Hat man sie verinnerlicht, sind sie keine Herausforderung mehr.
- (Experte 15) „Bei der Einführung von agilen Entwicklungsmethoden müssen auch die Methoden für User Research und Anforderungsmanagement an dieses Vorgehen angepasst werden. Wenn dem nicht so ist, wird es eine Herausforderung. Wir verzichten in der agilen Entwicklung auf Anforderungsdokumente und arbeiten direkt mit den einzelnen Anforderungen.
Nicht-funktionale Anforderungen: Die Festlegung ist kein Problem. Die Durchsetzung schon.“
- (Experte 16) „3.2 Meiner Erfahrung nach werden die Methoden bzw. Tools sogar von den Entwicklern bereitgestellt, sofern sich Anforderungen anhand von KPIs ableiten lassen.
3.4 Hier ist nicht klar was mit „verbindlich“ gemeint ist. Es wird nie etwas entwickelt, was sich nicht wieder ändern lässt. Wenn es darum geht diese „erfassbar“ zu machen, dann ist es wiederum ein grundsätzliches Problem der Entwicklung.“
- (Experte 20) zu 3.1 habe schon beides erlebt, das es nicht schnell genug ging kommt häufig vor
- (Experte 26) „Bis auf 3.4 sind dies für mich Herausforderungen, welche auch in der klassischen SW gelten.
zu 3.1) Ehr ein organisatorisches Problem bzw. zu wenig Personal (Anforderungsanalysten).
zu 3.2) Ehr ein organisatorisches- oder Kommunikationsproblem.“

Format der Anforderungen

- Item C9** In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungen als Ziele zu formulieren, die den Problemraum beschreiben, damit die Kreativität in der Lösungsfindung nicht eingeschränkt wird.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Kommt auf den Kontext an. Dazu ist der Begriff des Problemraums zu schwammig um daraus Handlungsweisen abzuleiten.
8	keine Lösungen vorgeben, evtl. schon im Vorfeld mit dem Team erarbeitete Lösungsansätze dokumentieren
11	Statt reinen User-Stories auch das Problem zu betrachten, was ein User / Kunde lösen möchte. Stichwort: Jobs to be done
12	Es gibt viele „Lückentexte“ als Formulierungshilfe, um dies zu vermeiden. Man muss allerdings trotzdem aufpassen, dass die vom Team gewählte Lösung allen allgemeinen Anforderungen (z.B. Sicherheit, Performanz) weiterhin entspricht.
13	Formulierung von Hypothesen, und Definition von Ergebnissen die die Hypothese bestätigen oder widerlegen. Fragen wie, welches Problem/Bedürfnis des Kunden lösen wir/wollen wir befriedigen?
15	Die Unterscheidung zwischen Problem- und Lösungsraum ist für Software-Entwicklungsteams schwierig, da sie in der Vergangenheit oft jahrelang in einer Mischform gearbeitet haben. Das ist ein Lernprozess, der durch regelmäßiges Feedback und Coaching betrieben werden muss.
19	So ist das Team frei bei der Umsetzung. Keine Umsetzungsregeln in User-Stories unterbringen.
22	Stakeholder intensiv nach Nutzen der Anforderung befragen, um das Ziel zu formulieren
25	Beim Projektstart sollten Ziele bzw. Teilziele definiert werden und während des Projektverlaufs angepasst werden.

- Item C11** In der agilen Softwareentwicklung ist es eine Herausforderung zu den Anforderungen eine Begründung des Nutzens zu formulieren, damit der Mehrwert der Umsetzung klar wird und Entscheidungen für eine bestimmte Anforderung nachvollziehbar sind.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Den Zwecksatz am Ende der User Story.
5	Aufnahme/Verlinkung der Begründung in das entsprechende Ticket.
12	Es hilft wahrscheinlich mehr, das Team ab und zu einen Usability Test beobachten zu lassen, damit sie darüber erfahren, wie die Nutzer über das Produkt denken.
19	Siehe oben. Nur so kann eine technisch durchdachte und innovative Lösung entstehen und vor allem können nur so die Werte der agilen Entwicklung gelebt werden.
20	Oftmals fällt es schwer das zu formulieren, manchmal macht es auch weniger sind das die Anforderung sehr kleinteilig ist.
22	Beispiel zeigen, wie eine Anforderung ohne Nutzendefinition inhaltlich korrekt aber völlig wertlos umgesetzt wurde / werden könnte
25	Im Grooming oder sonstigen Gesprächen kann Klarheit darüber geschaffen werden.

Item C18 In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungen so zu erfassen, dass zur Qualitätssicherung detaillierte Testfälle daraus abgeleitet werden können.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Im agilen Vorgehen würde ich die Vorformulierung detaillierter Testfälle grundsätzlich in Frage Stellen.
5	Bei sich ständig ändernden Anforderungen eignen sich regelmäßige Tests bei jedem Release, evtl. durch den Einsatz eines Crowd-Anbieters.
8	Einbeziehung der Tester in die Evolution der Anforderungen
10	Behat, Behaviour Driven Development
11	Akzeptanzkriterien formulieren und gemeinsam mit dem Team validieren. Dabei auf die neuen / wichtigen Dinge konzentrieren.
19	Das Testmanagement muss anhand der User-Stories abnahmefähige und messbare Testskzenarien entwerfen.
20	Oftmals reicht die User Story nicht aus und Kriterien zu Testfällen sind zusätzlich zu definieren

Item C19 In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungen klar und verständlich zu formulieren, um Unsicherheit in der Umsetzung zu vermeiden.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Ein großes Thema. Am wichtigsten ist die Klärung des Unterschieds zwischen „klar“ und „detailliert“.
8	enge Zusammenarbeit mit dem Team bei der Formulierung der Stories zur Umsetzung der Anforderungen
19	Hier können auch die Abnahmekriterien für das Testing zusätzlich zu den Stories wichtig sein.
20	Für das Team wichtig, durch starkes Herunterbrechen der Anforderungen sollte diese eh klar formuliert sind, wichtig ist das man den Kontext, in dem sich die Anforderungen befindet, nicht aus dem Blick verliert.
25	Antwort s. oben

Item C10 In der agilen Softwareentwicklung ist es eine Herausforderung Anforderungen so zu schneiden, dass sie einen Mehrwert für das Produkt bieten.

Experte	Empfohlene Lösung zur Bewältigung der Herausforderung
4	Value Streams, Impact Mapping, ROI-Kalkulation.
6	Dies ist aus meiner Sicht DER zentrale Punkt des agilen Ansatzes - es wird nur das umgesetzt, was einen Wert bietet.
9	Gut, wenn es klappt, aber nicht immer möglich und zwingend erforderlich: Stories können zu groß sein, als dass sie in einen Sprint passen. Werden sie dann kleiner geschnitten, kann bei den dann entstandenen einzelnen Stories kein fachlicher Mehrwert mehr vorliegen - (ich gehe davon aus, dass mit Mehrwert ein fachlicher Mehrwert gemeint ist). Es kann auch notwendig sein, Refactoring-Sprints durchzuführen. Ein Hinweis auf mangelnde Qualität in der Entwicklung zu zuvor, aber manchmal eben doch notwendig.
12	Es gibt zwar auch notwendige Tätigkeiten (z.B. Refactoring um die Wartbarkeit des Produktes zu bewahren), grundsätzlich sollten aber alle Anforderungen immer einen Mehrwert für das Produkt bieten. Ansonsten sind sie wertlos.
19	Durch sinnvolle Priorisierung.
22	Frage an den Kunden „wenn wir in 2 Wochen abrechnen müssten“, was sollten wir bis dahin unbedingt erreichen haben.

Anmerkungen zum Thema 4 „Format der Anforderungen“

- (Experte 11) Was generell nicht übersehen werden sollte, dass in der agilen SW-Entwicklung Anforderungen nicht nur verschriftlicht an die Entwicklung kommuniziert werden. Ziel ist es, dass der PO mit den Entwicklern über die Anforderungen spricht und dann in Form von Stories ein gemeinsames Verständnis festhält. M. E. müssen darin auch nicht alle Details festgelegt werden, wenn die beteiligten Parteien darüber ein Verständnis haben.
- (Experte 13) zu 4.4 Geschriebenes ist nur die Erinnerung an ein Gespräch oder die Einladung zu einem Gespräch. Das geschriebene Wort ist nicht interpretationsfrei.
- (Experte 15) Die Herausforderung besteht meiner Ansicht nach darin die richtige Granularität von Anforderungen zu finden. Es gilt Anforderungen so „umfangreich“ oder „fokussiert“ zu formulieren, dass eine Balance zwischen der Abdeckung aller Bedürfnisse (von Anwender bis Technik) und dem Aufwand in der Verwaltung der Anforderungen (Von der Erfassung bis zur Abarbeitung) entsteht. Damit kämpfen wir am meisten.
- (Experte 16) 4.1 Hier ist das Problem, dass jegliche Produktbeschreibung Lösungswege beschreibt, sobald Beispiele genannt werden um das Problem oder das Gewünschte zu erklären. Auch hier unabhängig von der gewählten Methode.
- (Experte 26) „Nein“ gelten für genauso in der klassischen SW Entwicklung. zu 4.1) „Kreativität der Lösungsfindung nicht einzuschränken“ ist für mich keine per se Anforderung von agilen Methoden. Ist dieses eine Anforderung bin ich der Meinung ist es in der klassischen SW-Entwicklung um einiges schwerer diese umzusetzen. zu 4.5) „Anforderungen schneiden“ bedeutet für mich, dass ich aktiv Anforderungen so erstelle, dass diese einen Wert haben. Dadurch gebe ich einer Anforderung einen künstlichen Wert, welche diese sonst nicht hätte.

– Journal Papers –

Schön, E.-M.; Sedeño, J.; Mejías, M.; Thomaschewski, J.; Escalona, M. J. (2017): Towards a Metamodel for Agile Requirements Engineering (<i>under review</i>)	
Abstract	<p>Context: Value delivery is becoming an important asset for an organization due to increasing competition in industry. Therefore, companies apply Agile Software Development (ASD) to be more competitive and reduce time to market. Using ASD for the development of systems implies that established approaches of Requirements Engineering (RE) undergo some changes in order to be more flexible to changing requirements. To this end, the field of agile RE is emergent and different process models for agile RE have arose.</p> <p>Objective: The aim of this paper is to build an abstract layer about the variety of existing process models by means of a metamodel for agile RE. Method: The metamodel was created in several iterations and relies on the evaluation of related process models. Furthermore, we derive process models for agile RE in industry by presenting instances of the metamodel in two different cases: one of the process models is based on Scrum whereas the other is based on Kanban. Results: This paper contributes to the software development body of knowledge by delivering a metamodel for agile RE that supports researchers as well as practitioners modeling and improving their own process models. Conclusions: We can conclude that the agile RE metamodel is highly relevant for the industry as well as for the research community, since we have derived it following empirical research in the field of ASD.</p>
Quality evidences	<i>n.a. since the paper is under review.</i>

Missler, T.; Schön, E.-M.; Escalona, M. J.; Thomaschewski, J. (2017): Towards A Model for Pair Design and Pair Testing based on the Characteristics of Pair Programming (<i>under review</i>)	
Abstract	<p>Pair programming is a method that is widespread in the field of agile software development (ASD) and is acknowledged as state of the art of programming. This article initially addresses the question of what constitutes the key attributes of pair programming. The extent to which these attributes can be applied to the development-related areas of human centered design (HCD) and the quality assurance (QA) of software will then be examined. The results of this consideration eventually lead to the presentation and consideration of a new model for the application of the attributes of pair programming in the context of pair design (HCD) and/or pair testing (QA). The discussion shows that a transferability and application is appropriate and that both areas benefit, particularly in terms of the product quality and product throughput time.</p>
Quality evidences	<i>n.a. since the paper is under review.</i>

Bader, F.; Schön, E.-M., Thomaschewski, J. (2017): Heuristics Considering UX and Quality Criteria for Heuristics. International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI), (in press)	
Abstract	Heuristic evaluation is a cheap tool with which one can take qualitative measures of a product's usability. However, since the methodology was first presented, the User Experience (UX) has become more popular but the heuristics have remained the same. In this paper, we analyse the current state of heuristic evaluation in terms of heuristics for measuring the UX. To do so, we carried out a literature review. In addition, we had a look at different heuristics and mapped them with the UX dimensions of the User Experience Questionnaire (UEQ). Moreover, we proposed a quality model for heuristic evaluation and a list of quality criteria for heuristics.
Quality evidences	The International Journal of Interactive Multimedia and Artificial Intelligence - IJIMAI (ISSN 1989 - 1660) provides an interdisciplinary forum in which scientists and professionals can share their research results and report new advances on AI tools or tools that use AI with interactive multimedia techniques. This international journal is not indexed by the Journal Citation Reports (Thomson Reuters) and therefore has no Impact Factor (2017).

Schön, E.-M.; Thomaschewski, J.; Escalona, M. J. (2017): Agile Requirements Engineering: A Systematic Literature Review. Computer Standards & Interfaces. Vol. 49, January 2017, pp. 79–91	
Abstract	Nowadays, Agile Software Development (ASD) is used to cope with increasing complexity in system development. Hybrid development models, with the integration of User-Centered Design (UCD), are applied with the aim to deliver competitive products with a suitable User Experience (UX). Therefore, stakeholder and user involvement during Requirements Engineering (RE) are essential in order to establish a collaborative environment with constant feedback loops. The aim of this study is to capture the current state of the art of the literature related to Agile RE with focus on stakeholder and user involvement. In particular, we investigate what approaches exist to involve stakeholder in the process, which methodologies are commonly used to present the user perspective and how requirements management is been carried out. We conduct a Systematic Literature Review (SLR) with an extensive quality assessment of the included studies. We identified 27 relevant papers. After analyzing them in detail, we derive deep insights to the following aspects of Agile RE: stakeholder and user involvement, data gathering, user perspective, integrated methodologies, shared understanding, artifacts, documentation and Non-Functional Requirements (NFR). Agile RE is a complex research field with cross-functional influences. This study will contribute to the software development body of knowledge by assessing the involvement of stakeholder and user in Agile RE, providing methodologies that make ASD more human-centric and giving an overview of requirements management in ASD.
Quality evidences	The quality of software, well-defined interfaces (hardware and software), the process of digitalisation, and accepted standards in these fields are essential for building and exploiting complex computing, communication, multimedia and measuring systems. Standards can simplify the design and construction of individual hardware and software components and help to ensure satisfactory interworking. This international journal is indexed by the Journal Citation Reports (Thomson Reuters) and its Impact Factor (2017) is 1.268.

Pfeiffer, T.; Hellmers, J.; Schön, E.-M.; Thomaschewski, J. (2016): Empowering User Interfaces for the Industry 4.0. Proceedings of the IEEE, Vol. 104, Issue: 5, May 2016, pp. 986 - 996	
Abstract	Industrie 4.0 (English translation: Industry 4.0) stands for functional integration, dynamic reorganization, and resource efficiency. Technical advances in control and communication create infrastructures that handle more and more tasks automatically. As a result, the complexity of today's and future technical systems is hidden from the user. These advances, however, come with distinct challenges for user interface design. A central question is: how to empower users to understand, monitor, and control the automated processes of Industrie 4.0? Addressing these design challenges requires a full integration of user-centered design (UCD) processes into the development process. This paper discusses flexible but powerful methods for usability and user experience engineering in the context of Industrie 4.0.
Quality evidences	<p>The most highly-cited general interest journal in electrical engineering and computer science, the Proceedings is the best way to stay informed on an exemplary range of topics. This journal also holds the distinction of having the longest useful archival life of any EE or computer related journal in the world. Since 1913, the Proceedings of the IEEE has been the leading journal to provide in-depth tutorial and review coverage of the technical developments that shape our world.</p> <p>This international journal is indexed by the Journal Citation Reports (Thomson Reuters) and its Impact Factor (2017) is 5.629.</p>

Schön, E.-M.; Escalona, M. J.; Thomaschewski, J. (2015): Agile values and their implementation in practice. International Journal of Artificial Intelligence and Interactive Multimedia (IJIMAI), Vol. 3, N°5, pp. 61-66	
Abstract	Today agile approaches are often used for the development of digital products. Since their development in the 90s, Agile Methodologies, such as Scrum and Extreme Programming, have evolved. Team collaboration is strongly influenced by the values and principles of the Agile Manifesto. The values and principles described in the Agile Manifesto support the optimization of the development process. In this article, the current operation is analyzed in Agile Product Development Processes. Both, the cooperation in the project team and the understanding of the roles and tasks will be analyzed. The results are set in relation to the best practices of Agile Methodologies. A quantitative questionnaire related to best practices in Agile Product Development was developed. The study was carried out with 175 interdisciplinary participants from the IT industry. For the evaluation of the results, 93 participants were included who have expertise in the subject area Agile Methodologies. On one hand, it is shown that the collaborative development of product-related ideas brings benefits. On the other hand, it is investigated which effect a good understanding of the product has on decisions made during the implementation. Furthermore, the skillset of product managers, the use of pair programming, and the advantages of cross-functional teams are analyzed.
Quality evidences	<p>The International Journal of Interactive Multimedia and Artificial Intelligence - IJIMAI (ISSN 1989 - 1660) provides an interdisciplinary forum in which scientists and professionals can share their research results and report new advances on AI tools or tools that use AI with interactive multimedia techniques.</p> <p>This international journal is not indexed by the Journal Citation Reports (Thomson Reuters) and therefore has no Impact Factor (2017).</p>

Schön, E.-M.; Hellmers, J.; Thomaschewski, J. (2014): Usability Evaluation Methods for Special Interest Internet Information Services. International Journal of Interactive Multimedia and Artificial Intelligence (IJIMAI), Vol. 2, N°6, pp. 26-32	
Abstract	The internet provides a wide range of scientific information for different areas of research, used by the related scientific communities. Often the design or architecture of these web pages does not correspond to the mental model of their users. As a result the wanted information is difficult to find. Methods established by Usability Engineering and User Experience can help to increase the appeal of scientific internet information services by analyzing the users' requirements. This paper describes a procedure to analyze and optimize scientific internet information services that can be accomplished with relatively low effort. It consists of a combination of methods that already have been successfully applied to practice: Personas, usability inspections, Online Questionnaire, Kano model and Web Analytics.
Quality evidences	The International Journal of Interactive Multimedia and Artificial Intelligence - IJIMAI (ISSN 1989 - 1660) provides an interdisciplinary forum in which scientists and professionals can share their research results and report new advances on AI tools or tools that use AI with interactive multimedia techniques. This international journal is not indexed by the Journal Citation Reports (Thomson Reuters) and therefore has no Impact Factor (2017).

Hellmers, J.; Thomaschewski, J.; Holt, E.-M.; Wriedt, T. (2012) Usability evaluation methods for a scientific internet information portal. Journal of Universal Computer Science (JUICS), Vol. 18, N° 10, pp. 1308-1322	
Abstract	Sharing information is important for the scientific community. Over the years the internet became the main information source due to its actuality, interactivity and flexibility. While the amount of available data grows, especially non-profit scientific internet pages often lack the user friendliness known from commercial offers, sometimes they also fail to focus on the users' needs. To analyze and improve the attractiveness of internet pages it became common to apply methods of usability engineering. But as it requires a certain amount of work it is usually done in 'big scale' for commercial offers. In this paper we would like to demonstrate the evaluation of a non-commercial scientific information internet portal using methods of usability engineering. For this an online User Experience Questionnaire (UEQ) in combination with web traffic analysis was used. We also would like to outline the experience made during the evaluation process, as well as some of the results.
Quality evidences	J.UCS - The Journal of Universal Computer Science - is a high-quality electronic publication that deals with all aspects of computer science. J.UCS has been appearing monthly since 1995 and is thus one of the oldest electronic journals with uninterrupted publication since its foundation. This international journal is indexed by the Journal Citation Reports (Thomson Reuters) and its Impact Factor (2017) is 0.546.

<p>Schön, E.-M.; Thomaschewski, J.; Escalona, M. J. (2017): Identifying Agile Requirements Engineering Patterns in industry by means of empirical research. In: Proceedings of the 20th European Conference on Pattern Languages of Programs (EuroPLOP 2017), (accepted)</p>	
Abstract	<p>Agile Software Development (ASD) is gaining in popularity in today’s business world. Industry is adopting agile methodologies both to accelerate value delivery and to enhance the ability to deal with changing requirements. However, ASD has a great impact on how Requirements Engineering (RE) is carried out in agile environments. The integration of Human-Centered Design (HCD) plays an important role due to the focus on user and stakeholder involvement. To this end, we aim to introduce agile RE patterns as main objective of this paper. On the one hand, we will describe our pattern mining process based on empirical research in literature and industry. On the other hand, we will discuss our results and provide two examples of agile RE patterns. In sum, the pattern mining process identifies 41 agile RE patterns. The accumulated knowledge will be shared by means of a web application.</p>
Quality evidences	<p>22nd European Conference on Pattern Languages of Programs. EuroPLOP is held in July 12-16, 2017 at Kloster Irsee in Bavaria, Germany.</p> <p>EuroPLOP is the premier European conference on patterns and pattern languages. Experience a unique way of knowledge sharing in the fields of software development, system design, human computer interaction, education, business and many more. EuroPLOP invites both academics and practitioners to participate when theory meets practice. Learn about the latest experience from the field and understand how and why good designs work.</p> <p>It is ranked as “B” in the CORE Extract of Conferences (The Computing Research and Education Association of Australasia).</p>

<p>Schön, E.-M.; Winter, D.; Escalona, M. J.; Thomaschewski, J. (2017): Key challenges in Agile Requirements Engineering. In: Baumeister H., Lichter H., Riebisch M. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2017. LNBP Vol. 283, pp. 37–51</p>	
Abstract	<p>Agile Software Development (ASD) is becoming more popular in all fields of industry. For an agile transformation, organizations need to continuously improve their established approaches to Requirements Engineering (RE) as well as their approaches to software development. This is accompanied by some challenges in terms of agile RE. The main objective of this paper is to identify the most important challenges in agile RE industry has to face today. Therefore, we conducted an iterative expert judgement process with 26 experts in the field of ASD, comprising three complementary rounds.</p> <p>In sum, we identified 20 challenges in three rounds. Six of these challenges are defined as key challenges. Based on the results, we provide options for dealing with those key challenges by means of agile techniques and tools. The results show that the identified challenges are often not limited to ASD, but they rather refer to software development in general. Therefore, we can conclude that organizations still struggle with agile transition and understanding agile values, in particular, in terms of stakeholder and user involvement.</p>
Quality evidences	<p>Proceedings of the 18th International Conference on Agile Software Development.</p> <p>The 18th International Conference on Agile Software Development (XP 2017) held in May 22-26, 2017, Cologne, Germany. XP is the leading international conference on agile and lean methods in software development. The conference aims to bring together industrial practitioners and researchers in the fields of agile software development, to collaboratively generate new insights, and to ultimately discover better ways of developing software.</p> <p>It is ranked as “B” in the CORE Extract of Conferences (The Computing Research and Education Association of Australasia).</p>

<p>Sedeño, J.; Schön, E.-M.; Torrecilla-Salinas, C.; Thomaschewski, J.; Escalona, M. J.; Mejías, M. (2017): Eliciting Agile requirements using Context-based Persona Stories. In: Proceedings of WEBIST the 13th International Conference on Web Information Systems and Technologies. SCITEPRESS - Science and Technology Publications, pp. 196–203 (WEBIST2017)</p>	
<p>Abstract</p>	<p>In recent years hybrid approaches focusing on user needs by integrating Agile methodologies (e.g. Scrum, Kanban or Extreme Programming) with Human-Centered Design (HCD) have proven to be particularly suitable for the development of Web systems. On the one hand, HCD techniques are used for requirements elicitation and, on the other hand, they can be utilized to elicit navigation relationships in Web projects. Navigation is one of the basic pillars of Web systems and also a fundamental element for the methodologies within the Model-Driven Web Engineering (MDWE) field. This paper presents an approach to model Agile requirements by means of integrating HCD techniques into Agile software development. We contribute to the software development body of knowledge by creating the concept of a Context-based Persona Story (CBPS) and formalizing it through a metamodel. Our approach covers the modeling of users and stakeholders by personas as well as the visualization of the context of use by storyboards. The attributes of the context of use enable us to elicit acceptance criteria for describing the scope of an Agile requirement.</p>
<p>Quality evidences</p>	<p>13th International Conference on Web Information Systems and Technologies. WEBIST is held in April 25-27, 2017 in Porto, Portugal.</p> <p>The purpose of the 13th International Conference on Web Information Systems and Technologies (WEBIST) is to bring together researchers, engineers and practitioners interested in the technological advances and business applications of web-based information systems. The conference has five main tracks, covering different aspects of Web Information Systems, namely Internet Technology, Web Interfaces and Applications, Society, e-Communities, e-Business, Web Intelligence and Mobile Information Systems.</p> <p>It is ranked as “C” in the CORE Extract of Conferences (The Computing Research and Education Association of Australasia).</p>

<p>Schön, E.-M.; Winter, D.; Uhlenbrok, J.; Escalona, M. J.; Thomaschewski, J. (2016): Enterprise Experience into the Integration of Human-Centered Design and Kanban. In: Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016) – Vol. 1: ICSOFT-EA , pp. 133-140</p>	
<p>Abstract</p>	<p>The integration of Human-Centered Design (HCD) and Agile Software Development (ASD) promises the development of competitive products comprising a good User Experience (UX). This study has investigated the integration of HCD and Kanban with the aim to gain industrial experiences in a real world context. A case study showed that requirements flow into the development process in a structured manner by adding a design board. To this end, the transparency concerning recurring requirements increased. We contribute to the body of knowledge of software development by providing practical insights into Human-Centered Agile Development (HCAD). On one hand, it is shown that the integration of HCD and Kanban leads to a product with a good UX and makes the development process more human-centered. On the other hand, we conclude that a cross-functional collaboration speeds up product development.</p>
<p>Quality evidences</p>	<p>11th International Joint Conference on Software Technologies. ICSOFT 2016 is held in July 24-26, 2016 in Lisbon, Portugal.</p> <p>The purpose of the ICSOFT-EA 2016, the 11th International Conference on Software Engineering and Applications, is to bring together researchers, engineers and practitioners interested on information technology, with a practical focus on software engineering and applications. The conference tracks are “Enterprise Software Technologies”, “Software Project Management”, “Software Engineering Methods and Techniques”, and “Distributed and Mobile Software Systems”.</p> <p>Software technologies are essential for developing any enterprise information system, encompassing a large number of research topics and applications: from programming issues to the more abstract theoretical aspects of software engineering; from software security to mobile applications; Distributed systems, ubiquity, software quality as well as project management and other related topics are included in the scope of ICSOFT-EA.</p> <p>It is ranked as “B” in the CORE Extract of Conferences (The Computing Research and Education Association of Australasia).</p>

Schön, E.-M. (2016): A Methodology for Agile Requirements Engineering based on a Pattern Approach. CAiSE 2016 - Doctoral Consortium	
Abstract	Agile Software Development (ASD) is used facing the challenge to reduce time to market and to deliver systems, which meets customer expectations. In ASD, Requirements Engineering (RE) is carried out in an iterative manner and therefore established approaches have to be adopted with strong focus on stakeholder and user involvement. The results of a Systematic Literature Review (SLR) show that there are different ways to carry out Agile RE approaches, what increases heterogeneity among them. This thesis studies how a commonly accepted framework for Agile RE looks like. To this end, the initial phase for creating a methodology based on a pattern approach will be performed. Therefore, the design science research methodology is applied to carry out the research. The designed artifact is a metamodel that describes the Agile RE process at an abstract level. It represents the main part of the methodology and will be evaluated through an international qualitative study.
Quality evidences	<p>28th International Conference on Advanced Information Systems Engineering. CAiSE 2016 is held in June 13-17, 2016 in Ljubljana, Slovenia.</p> <p>The CAiSE'16 Doctoral Consortium will be the 23rd Doctoral Consortium of a series held in conjunction with the CAiSE conference series. It is intended to bring together PhD students working on foundations, techniques, tools and applications of Information Systems Engineering and provide them with an opportunity to present and discuss their research to an audience of peers and senior faculty in a supportive environment, as well as to participate in a number of plenary sessions with Information Systems academics.</p> <p>It is ranked as "A" in the CORE Extract of Conferences (The Computing Research and Education Association of Australasia).</p>

Winter, D.; Holt, E.-M.; Thomaschewski, J. (2012): Persona driven agile development - Build up a vision with personas, sketches and persona driven user stories In: Á. Rocha, J. A. Calvo-Manzano, L. Paulo Reis und M. Pérez Cota (Hg.): Information Systems and Technologies (CISTI). Proceedings of the 7th conference held in Madrid	
Abstract	In the beginning of a user centered agile development process, three important elements have to be defined. 1) A software product with a defined set of features, 2) the intended usage context and 3) the future software product user. The clear definition of these elements right from project start will lead to a more accurately developed software product. A combination of well-known tools like "personas" and sketches of the context of use as well as requirements as persona driven user stories make a comprehensive basis. During an iterative visioning phase, also referred to as sprint 0, directly before starting the first sprint, those tools can be integrated to assure a complete software product definition process. That will ensure that, even in this early stage, the agile project is focused on user needs represented by "personas".
Quality evidences	<p>7th Iberian Conference on Information Systems and Technologies. CISTI 2012 is held in June 20-23, 2012 in Madrid, Spain.</p> <p>CISTI is a technical and scientific event, whose purpose is to present and discuss knowledge, new perspectives, experiences and innovations in the Information Systems and Technologies field.</p>

<p>Uhlenbrok, J.; Schön, E.-M.; Winter, D.; Thomaschewski, J. (2015): User Experience in Kanban – Case Study: Erfahrungen aus dem Relaunch eines Internetportals. In: Endmann, Fischer & Krökel (eds.), Mensch und Computer 2015 - Usability Professionals. Berlin: De Gruyter Oldenbourg, pp. 84-94</p>	
Abstract	<p>Kanban ist eine agile Projektmanagementmethode, die auf geringe Durchlaufzeiten während des Entwicklungsprozesses abzielt. Bereits im Jahre 2013 haben wir uns mit der Herausforderung beschäftigt, wie eine Integration von Human-Centered Design in Kanban gestaltet werden kann. Ein wesentlicher Punkt dabei ist es, einen Gesamtüberblick über die zu bearbeitenden Aufgaben zu erhalten. Hierzu haben wir verschiedene Methoden vorgestellt, welche den Kanban-Prozess um Human-Centered Design Aktivitäten erweitern.</p> <p>In einer Case Study zum Relaunch eines Internetportals wurden die Methoden im interdisziplinären Team produktiv eingesetzt, um ein Produkt mit positiver User Experience zu entwickeln. Nach Abschluss des Projektes wurden Interviews mit den Projektbeteiligten durchgeführt, Ergebnisse zusammengefasst sowie Vor- und Nachteile herausgearbeitet.</p>
Quality evidences	<p>This national conference (held in September 06-09., 2013 in Stuttgart, Germany) is combined of the research track Mensch-Computer-Interaktion (MCI) and the practitioners track Usability Professionals (UP). Mensch und Computer aims to bring together industrial practitioners and researchers in the fields of Human-Centered Design and User Experience, to collaboratively generate new insights.</p>

<p>Winter, D.; Schön, E.-M.; Uhlenbrok, J.; Thomaschewski, J. (2013): User Experience in Kanban – Die UX-Karte ausspielen. In: Henning Brau, Andreas Lehmann, Kostanija Petrovic und Matthias C. Schroeder (eds.): Usability Professionals 2013. Stuttgart: German UPA, pp. 220-224</p>	
Abstract	<p>Die agile Projektmanagementmethode „Kanban“ zielt auf geringe Durchlaufzeiten ab und richtet damit den Fokus der Entwicklung auf das Moment der aktuellen Aufgabe. Um Produkte mit einer positiven User Experience zu entwickeln, muss bereits der Gestaltungsprozess auf die Bedürfnisse des Menschen ausgelegt sein. Dadurch werden agile Projekte vor die Herausforderung gestellt, sowohl die einzelnen Aufgaben als auch das gesamte Produkt zielgerichtet und nutzerzentriert zu realisieren. Dieser Beitrag soll zeigen, wie trotz des eingeschränkten Gesamtüberblicks dennoch eine Nutzerzentrierung bei der Entwicklung realisiert werden kann. Hierzu wird vor allem die Frage geklärt, wie sich Human-Centered Design Aktivitäten in Kanban integrieren lassen und an welchen Stellen im Softwareentwicklungsprozess Evaluationen der User Experience durchgeführt werden können.</p>
Quality evidences	<p>This national conference (held in September 08-11., 2013 in Bremen, Germany) is combined of the research track Mensch-Computer-Interaktion (MCI) and the practitioners track Usability Professionals (UP). Mensch und Computer aims to bring together industrial practitioners and researchers in the fields of Human-Centered Design and User Experience, to collaboratively generate new insights.</p>

Holt, E.-M.; Winter, D.; Thomaschewski, J. (2012) Von der Idee zum Prototypen - Werkzeuge für die agile Welt. In: Brau, Lehmann et al. (eds.) 2012 – Usability Professionals 2012. Stuttgart: German UPA e.V., pp. 22-27	
Abstract	Damit in agilen Entwicklungsprojekten, basierend auf einer Idee, ein nutzerzentrierter Low-Fidelity Prototyp entwickelt werden kann, bietet es sich an, die Idee zunächst zu einer Vision zu konkretisieren. Hierbei handelt es sich um einen iterativen Prozess, für den im Human Centered Design unterschiedliche Werkzeuge zur Verfügung stehen. Der vorliegende Beitrag stellt eine Auswahl von Werkzeugen zur Konkretisierung der Vision vor, die sich in agilen Prozessen als Best Practices bewährt haben. Dabei werden konkrete Vorlagen und Einsatzmöglichkeiten für diese Werkzeuge erörtert.
Quality evidences	This national conference (held in September 09-12., 2012 in Konstanz, Germany) is combined of the research track Mensch-Computer-Interaktion (MCI) and the practitioners track Usability Professionals (UP). Mensch und Computer aims to bring together industrial practitioners and researchers in the fields of Human-Centered Design and User Experience, to collaboratively generate new insights.

Holt, E.-M.; Winter, D.; Thomaschewski, J. (2011): Personas als Werkzeug in modernen Softwareprojekten. In: Brau, Lehmann, Petrovic und Schroeder (eds.): Usability Professionals 2011. Stuttgart: German UPA e.V., pp. 40-44	
Abstract	<p>In der agilen Softwareentwicklung haben User Stories einen hohen Stellenwert. Häufig beschreiben sie die Anforderungen aus Anwendersicht, ohne dabei den Anwender weiter zu konkretisieren. Dies führt dazu, dass der Anwender ein theoretisches Konstrukt bleibt und dem Entwickler hypothetisch erscheint. Zur Steigerung des Verständnisses des hypothetischen Anwenders bietet sich die Persona-Methode als Ergänzung zu User Stories an. Personas ermöglichen es, ein realistisches Bild des Anwenders zu vermitteln und können in allen Teilen des Entwicklungsprozesses vom Verständnis der Anwenderbedürfnisse über Usability-Tests bis hin zu Akzeptanztests herangezogen werden. Sie ermöglichen für alle Projektbeteiligten eine einheitliche Sichtweise auf den Anwender und schaffen insbesondere bei den Software-Entwicklern ein besseres Anwenderverständnis während der Implementierungsphase.</p> <p>Den Teilnehmern wird im zugehörigen Tutorial vermittelt, wie Personas auf eine Art erstellt und genutzt werden können, die durch einen hinreichenden Realismus die Entwicklung von Software in allen Teilen des Entwicklungsprozesses unterstützt.</p>
Quality evidences	This national conference (held in September 10.-14., 2011 in Chemnitz, Germany) is combined of the research track Mensch-Computer-Interaktion (MCI) and the practitioners track Usability Professionals (UP). Mensch und Computer aims to bring together industrial practitioners and researchers in the fields of Human-Centered Design and User Experience, to collaboratively generate new insights.

Hellmers, J.; Holt, E.-M.; Thomaschewski, J. (2011): Personas für das wissenschaftliche Internetportal ScattPort zum Thema Lichtstreuung. In: Jens Hellmers, Thomas Wriedt und Jörg Thomaschewski (eds.): 2. Symposium Wissenschaftliche Portale und Informationsnetzwerke. Hannover: Technische Informationsbibliothek und Universitätsbibliothek, pp. 29–37	
Abstract	In der Entwicklung von Software oder Internetauftritten besteht das Problem, dass sich die Entwickler oftmals kein hinreichend geeignetes Bild von den potentiellen Anwendern machen können. Dabei besteht dann die Gefahr, dass das Produkt an den Bedürfnissen der späteren Nutzer vorbei entwickelt wird. Die Verwendung der Methode Personas ermöglicht es dem Entwickler, die Anforderungen der Benutzer während des Entwicklungsprozesses stärker zu berücksichtigen und somit die genannte Problematik zu umgehen. In diesem Artikel wird die Entwicklung von Personas für ein wissenschaftliches Internet Informationsportal beschrieben und dargelegt, welche Überlegungen dabei eingeflossen sind.
Quality evidences	This national symposium was held in April, 11 in 2011 Bremen, Germany at the Institut für Werkstofftechnik

Holt, E.-M.; Hellmers, J.; Thomaschewski, J. (2011): Usability Engineering am Praxisbeispiel gruendungsachse.de. In: Jens Hellmers, Thomas Wriedt und Jörg Thomaschewski (eds.): 2. Symposium Wissenschaftliche Portale und Informationsnetzwerke. Hannover: Technische Informationsbibliothek und Universitätsbibliothek, pp. 17-28	
Abstract	Ein erfolgreiches Internetportals basiert auf der Erfüllung der spezifischen Benutzeranforderungen, die beim Besuch des Internetportals bewusst und unbewusst vorhanden sind. Somit müssen Inhalte, Design, Bedien- und Interaktionskonzepte bereits bei der Planung an die Bedürfnisse der zukünftigen Benutzer angepasst konzipiert werden. Oftmals unterscheiden sich noch immer die Ziele und Bedürfnisse der Benutzer von denen der Seitenanbieter, beispielsweise wenn Strukturen abgebildet werden, deren Zusammenhänge für den Benutzer irrelevant sind. Der Anbieter des Internetportals wird ohne valide Kenntnisse um die zukünftigen Benutzer deren Bedürfnisse selten derart befriedigen können, dass diese das Internetportal wiederkehrend benutzen. In dieser Studie wird das theoretische Modell „Usability Engineering“ an einem Praxisbeispiel dargestellt. Die Ergebnisse der Studie geben konkrete Handlungsanweisungen zur Konzeption von Internetportalen anhand des spezifischen Beispiels „gruendungsachse.de“ und zeigen die eingesetzte Methodik zur Berücksichtigung unterschiedlicher Benutzergruppen auf.
Quality evidences	This national symposium was held in April, 11 in 2011 Bremen, Germany at the Institut für Werkstofftechnik

– Scholarships –

fem:talent Scholarship	
Information	The fem:talent scholarship is granted by the University of Applied Science Emden/Leer with the aim to increase the number of women in top scientific positions.
Funder	University of Applied Sciences Emden/Leer

– R&D Projects –

Red Temática para el desarrollo de soluciones software de calidad en entornos PLM	
Project information	Person in charge: María José Escalona Cuaresma Type of Project: Plan Estatal 2013-2016 Retos - Proyectos I+D+i Reference: TIN2013-46928-C3-3-R Start date: 01/01/2014 End date: 31/12/2016
Funder	Ministerio de Economía y Competitividad

Explorando Soluciones Guiadas para Sistematizar el Aseguramiento Temprano de la Calidad del Software	
Project information	Person in charge: María José Escalona Cuaresma / Manuel Mejías Risoto Type of Project: Plan Estatal 2013-2016 Retos - Proyectos I+D+i Reference: TIN2016-76956-C3-2-R Start date: 30/12/2016 End date: 29/12/2019
Funder	Ministerio de Economía y Competitividad

Mecanismos Guiados en Etapas Tempranas para la Mejora del Software	
Project information	Person in charge: María José Escalona Cuaresma Type of Project: Plan Nacional del 2014 Reference: TIN-2013-46928-C3-3-R Start date: 01/01/2014 End date: 31/12/2016
Funder	Ministerio de Ciencia e Innovación

Development of an information and software portal for light scattering on particles	
Project information	Person in charge: Thomas Wriedt (University Bremen, DE) Type of Project: DFG Project Reference: INST 1216/1-2 Start date: 01/08/2007 End date: 31/07/2011
Funder	Deutsche Forschungsgemeinschaft

Fachkräfteinitiative „Ems-Achse – beste Köpfe, beste Chancen“: Teilprojekt 3: Entrepreneurship-Offensive	
Project information	Person in charge: Hermann Wocken (Wachstumsregion Ems-Achse e.V.) Type of Project: Projekt zur Regionalentwicklung Reference: - Start date: 01/06/2010 End date: 31/05/2013
Funder	N-Bank

- Industry Projects -

Development of a SCADA system in wind energy 11/2015 until 05/2017	
Project assignment	<ul style="list-style-type: none"> • Requirements engineering for redevelopment of existing SCADA system • Responsible for adopting agile techniques in the international environment of SCADA development • Establishing an agile requirements engineering process model • Coordination of requirements workshops with different stakeholder groups using various elicitation techniques.

Product and process manager in e-commerce, 07/2015 - 11/2015	
Project assignment	<ul style="list-style-type: none"> • Interim management in technical product management with focus on analyzing and conception of product and process requirements for web shop • Creation of technical solutions considering requirements from IT and further departments • Managing projects autonomously and reporting the project status to division manager and general manager • Working in an agile environment as a Product Owner for an agile development team

FinTech portal for startup, 05/2015 - 07/2015	
Project assignment	<ul style="list-style-type: none"> • Development of a FinTech portal for digital services and responsive design. • Working in an agile environment as a Scrum Master for an agile development team • Responsible for coaching of the Product Owner with regard to Human-Centered Design, prototyping and creation of user stories.

IT Service Management with ITIL v3 for availability, capacity und configuration management – Transition of service IT infrastructure in seafaring industry and logistics, 01/2015 - 04/2015	
Project assignment	<ul style="list-style-type: none"> • Requirements engineering for reporting standards. • Developing of reporting standards and guidelines in collaboration with the client

Traffic information portal in government – 01/2012 - 09/2014	
Project assignment	<ul style="list-style-type: none"> • Consolidation of existing web portals into one integrated solution which delivers information about the current traffic situation to its users • Supporting the proposal team during the europe-wide bid procedure • Working in an agile environment as a Product Owner for an international agile development team • As project manager responsible for coordination of external vendors during the transition phases of the platform

Portal development for conference services in telecommunications – 01/2012 - 07/2012	
Project assignment	<ul style="list-style-type: none"> • Requirements Engineering and User Experience Design for a conference service portal • Process optimization based on contextual inquiries • Documentation of results in forms of workflow descriptions and prototypes • Creation of high-fidelity prototype using Axure • Creation of visual designs using Photoshop

User Experience and Usability, various – since 08/2009	
Project assignment	<ul style="list-style-type: none"> • User experience design and usability evaluation of various software products, websites und mobile applications

– Research fellowship –

January-June 2017. Six months of research fellowship in the University of Applied Science Emden/Leer, invited by Dr. x.

– Supervision of students´ projects –

In between 2014 and 2017, I supervised in sum 14 students´ projects at the University of Applied Sciences Emden/Leer. The projects belong to bachelor and master program of computer science and the topics are the following:

- Heuristische Evaluation (Arbeitstitel) in SS2017
- Entwicklung eines Modells um UX-Schulden in der agilen Softwareentwicklung erfolgreich aufzudecken und zu behandeln (Arbeitstitel), in SS2017
- Adaption agiler Requirements Engineering Methoden für ein traditionelles Projektumfeld (Arbeitstitel), SS2017
- Vorgehensweise zur Entwicklung eines Software Prototypen (Arbeitstitel), in SS2017
- Portal zur Darstellung von Agile Requirements Engineering Patterns, in SS2017
- Einführung von agilen Software-Entwicklungsprozessen im konservativen industriellen Mittelstand, in SS2017
- Heuristics considering UX and quality criterias for heuristics, in WS2016/2017
- Entwurf eines modellgetriebenen Softwareentwicklungsprozesses (Arbeitstitel), in WS2016/2017
- UX-Schulden - Wie können UX-Schulden in der agilen Softwareentwicklung erkannt und behandelt werden?, in SS2016
- Pair Design & Pair Testing - Ist das Konzept von Pair Programming auf das Produktdesign und die Qualitätssicherung von Software übertragbar? in SS2016
- Produktentwicklung nach der Lean Startup Methode „Am Beispiel einer Online-Marketing-Plattform für PC-Spiele“, in WS2016/2017

- Matrix agiler Modellein, in WS2015/2016
- Barrierefreiheit von Webpräsenzen SS2014
- Evaluation von Prototypen für ein webbasiertes System zur Erstellung und Auswertung von Fragebögen, in WS2013/2014