TOWARDS UNDERSTANDING THE VALUE-CREATION IN AGILE PROJECTS

ZORNITZA BAKALOVA

PhD Dissertation Committee

Chairman and Secretary	
Prof. dr. ir. A.J. Mouthaan	University of Twente
Promotor	
Prof. dr. R.J. Wieringa	University of Twente
Assistant Promotor	
Dr. M. Daneva	University of Twente
Members	
Prof. Dr. ir. M. Aksit	University of Twente
Prof. Dr. J. van Hilegersberg	University of Twente
Prof. Dr. N.H.Madhavji	University of Western Ontario
Dr. J. Doerr	Fraunhofer IESE
Dr. T. Fehlmann	Euro Project Office

CTIT Dissertation Series No. 13-288
Center for Telematics and Information Technology (CTIT) P.O. Box 217 – 7500 AE Enschede – The Netherlands

The research in this thesis was supports by the Netherlands Organization for scientific Research (NWO) under the QuadREAD project

ISBN: 978-90-365-3605-9
ISSN: 1381-3617
DOI: 10.3990/1.9789036536059
Printed by: Ipskamp Drukkers

TOWARDS UNDERSTANDING THE VALUE-CREATION IN AGILE PROJECTS

DISSERTATION

to obtain
the degree of doctor at the University of Twente
on the authority of the rector magnificus,
Prof. Dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday, January 24, 2014 at 14.45

by

Zornitza Gueorguieva Bakalova

born on October 28, 1969

in Yambol, Bulgaria

This dissertation has been approved by:

Prof. dr. R.J. Wieringa (promotor)

Dr. M. Daneva (assistant promotor)

Summary

In recent years, iterative and incremental approaches for software development appeared as an alternative to the traditional, waterfall-style development. The reason for this is the large number of software projects in the past that failed to deliver useful products within budget, and struggled with changing requirements and scope creep. Meanwhile it is a common sense understanding that not all projects are predictable from the beginning. Market uncertainty and a fast changing business environment drives changes during the development of a software product.

One of the key characteristics of any agile approach is its explicit focus on Business Value. Although any software development method aims at creating a product and thus creating value, in agile software projects the value creation for the clients represents the essence and defines the focus of the process. Thus, the agile development process is a value creation process.

The agile methods allow for frequent decisions about the requirements that will be considered for implementation during the short development cycles called iterations. In practice this decision-making is implemented by the process of requirements prioritization and re-prioritization, performed at the beginning of each iteration.

This work is dedicated to exploring and understanding the process of value-creation for clients in agile projects, with a particular focus on the requirements prioritization and re-prioritization during a project, as an agile-specific value creation practice.

We performed a number of research steps to explore some of the current agile practices that seem to contribute to the value creation, and thus to distil knowledge that the agile practitioners apply and that might help to improve the agile practice.

Further, we studied in detail the agile prioritization process and identified the criteria, used in the decision-making process, and relations between the project context and the instantiation of the process.

In particular, we researched the following topics:

- How is business value perceived and measured in agile projects?
- What practices contribute to value creation in agile projects in different contexts?

- What concepts play a role in making re-prioritization decisions about requirements?

These questions represent the focus of our research activities. They lead and framed the formulation of our Research Questions and the research design.

The main contribution of our work to the research and practitioners' communities consists in the rich contextual description of the process of requirements prioritization in agile projects as well as a conceptual model of this process.

Acknowledgements

This journey took longer than anticipated, but here we are - at the final destination.

It has been a remarkable and unforgettable time for me in many respects, and it would not have been possible to go all this way without the help and the support of many great people. Here I want to thank you:

Maya – for being not only a supervisor, but also a mentor and a friend. Without your agility and competence it wouldn't have been possible to complete this work. I enjoyed our discussions on all possible aspects of research and life.

Roel, thank you for the inspiring conversations on philosophy of science, the big picture, generalizability and more. And for the flexibility allowing me to work from home.

I thank all committee members for accepting to be on the committee and for taking their time to read the thesis and to provide valuable comments and feedback.

Klaas, thank you for your support and trust. You were a pillar during my stay at UT.

Suse and Ida, you were always ready to help and found solutions to any problem. You made me feel at home.

Andrea and Luigi, thank you for the collaboration during the paper-writing. I learned a lot from you.

I also want to thank:

… all case study participants, who took their time to discuss their work with me, provided valuable insights and shared their experience and knowledge.

… Ivan, Arda and Klaas van den Berg, who made the work in the project a fun.

… all QuadRead business partners for showing interest in my research, participating in insightful discussions and providing constructive feedback.

… Dr. Tse and Dr. Schreamer for giving me the peace of mind to concentrate on research.

… Daphne and Andrea for their precious advices during the most challenging time of my life.

… Laura, Nelly, Siv, Virginia, Mohammad, Shahin, Roberto, Jelena, André, Silja, Pascal, Cams, Chen, Lianne, for making the IS group a creative, stimulating and pleasant place to work.

… Daniel, Esteban, Gunnar, Hape, Heiko, Holger, Jochen, Karsten and Paul, who taught me that even the best product owner is worth nothing without a great scrum team.

And last but not least, I want to express my appreciation for the love and support I have received from my family and friends.

Carmen, Franz, Margot, Maria, Nevenka, Niki, Sevi, Sophie - I am blessed to have friends like you.

I thank my parents Nelly and Georgi and my sister Julia for being there for me.

I am particularly grateful to my late aunt Dr. Diliana Vladeva for being a role model and a friend, and for encouraging me during the work on the thesis.

Rossen, thank you for taking care of our children during the time when I was away from home.

Ana and Alexander, you are my greatest motivation and inspiration. Thank you for being just who you are.

Contents

Chapter 1

Introduction and motivation

In this chapter we introduce the problem that we investigate in this thesis, explain our motivation to study this problem, and explain why it is worth to be investigated. Furthermore, we discuss the importance of the topic for the academic and practitioners' communities, and outline our contribution.

1 Introduction

In recent years, iterative and incremental approaches for software development appeared as an alternative to the traditional, waterfall - style development. The reason for this is the large number of software projects in the past that failed to deliver useful product within budget, and struggled with changing requirements and scope creep. Meanwhile it is a common sense understanding that not all projects are predictable from the beginning. Market uncertainty and a fast changing business environment drives changes during the development of a software product. A process is needed that allows teams to change direction based upon the customers' and market needs. In other words, response to changes should be a welcomed aspect of the development method. Economic downturn makes software development improvements even more urgent. This motivated software practitioners to look for alternative software development paradigms that could address and cope with those challenges. In 2001 a group of practitioners came up with the Agile

Manifesto [Agile]. This short document states the new way of thinking about the software development and captures the main characteristics of the so-called agile approaches: (1) creating small increments of working software in time-boxed iterations, (2) focusing on creating value for the clients and (3) responding to changes.

Since then, Agile Software Development Methodologies experienced wide acceptation and adoption in software developing companies. Certainly, the idea of iterative software development is not new and revolutionary. Methods, iterative in their nature, have been proposed and used for the last few decades. Such are, for example, the spiral model of B. Boehm [Boehm 86], the IBM's Rational Unified Process (RUP) [Kroll 03], or the Evo method [Gilb 85]. However, there are a number of characteristics that differentiate the agile approaches from other iterative methods.

One of the key characteristics of any agile approach is its explicit focus on Business Value (BV) [Abrahamsson 02]. Although any software development method aims at creating a product and thus – creating value, in agile software projects the value creation for the clients represents the essence and defines the focus of the project. Thus, the agile development process is a value creation process as ant development process is. Indeed, the agile community established a common understanding that (i) the main purpose of an agile project is to deliver maximum business value for the client and that (ii) agile approaches deliver business value fast and early in the project. This paradigm is not surprising, as demonstrating the linkage between investments in IT solutions and business benefits is becoming mandatory for an increasingly large number of organizations. Although this holds for any IT investment, it is particularly necessary in the context of agile software development, as new agile methodologies are being adopted and need to prove their merits. However, little is known about what Business Value is, and about the mechanisms that agile methodologies have in play that enable and lead to BV creation.

The literature of agile software engineering (SE) e.g. [Cao 08] has emphasized that value creation is attributable to the nature of agile projects. Agile software practices are credited with saving failing projects and increasing the success chance of many others. In fact, according to the 2011 CHAOS report from the Standish Group [Chaos 11], Agile projects are successful three times more often than non-agile projects. In the understanding of the agile SE community, the value delivered to the client lies not only in what the final software product represents, but also in the development process as such, which significantly contributes to the amount of value delivered. The accumulation of value is ensured by practices that are specific attributes of the agile methods only, in

particular the short iterations, the frequent response to changes, the active involvement of the clients and incorporating learning during the project.

The agile methods allow for frequent decisions about the requirements that will be considered for implementation at each iteration. In practice this is implemented by the process of requirements prioritization and re-prioritization, performed at the beginning of each iteration. As Gottesdiener [Gottesdiener] puts it: "Each release represents the culmination of a series of requirements decisions." The highest priority features (i.e. requirements in agile terminology) get implemented early so that most business value gets realized, while exposing the project to as low a risk as possible. According to the agile literature, e.g. [Ambler 02] [Beck 00] [Harris 06], a key tenet of agile processes is that the requirements are prioritized by a customer, customer team, or 'product owner' acting as a proxy for the end users of the intended system. The rationale behind this is that the client is the one who can make a judgment about the value of each requirement. Although the value creation is the core of the agile projects, researchers [Barney 08] [Petersen 09] in agile RE case studies found that the creation of software product value is only partly understood.

The observations above suggest that both agile companies and their clients would profit from a deeper understanding of BV and the phenomenon of BV creation in agile context. This represents the main research goal of our work. Below we provide detailed motivation for our investigation.

2 Motivation

2.1 Reliance on tacit knowledge

There exists already a substantial body of knowledge, dedicated to agile practices, in the form of books and papers. There is abundant literature that tells us what agile practices are, and how they work. However, the literature seldom provides insights on the 'why' – namely, in which cases would certain method or practice work, under which conditions, what is the context in which the methods are effective and will lead to desired effects.

Agile development is highly people-oriented activity that relies to a large extend on implicit or tacit knowledge. This originates in the following agile practices: (1) as extensive documentation does not necessarily represent value for the client, the decisions are rarely documented and more often than not the working software is the main

document of a project; (2) agile methods rely on highly skilled and motivated individuals that often make decisions based on their experience, knowledge and gut feeling; and (3) agile project management practices deal with knowledge exchange and transfer, such as daily stand-up meetings, involve face-to-face, verbal communication. This contributes to knowledge exchange among the team members; however such knowledge remains within the team. It is difficult to identify and disseminate good practices or to identify the rationale behind certain decisions, which leads to difficulties when reasoning about the application of single agile practices in concrete project settings, and generalizing on the impact of different practices across contexts.

2.2 The advanced state of the practice vs. research

It is characteristic of the agile community that the practitioners are the pioneers that drive the advancement of the agile methods.

The dissemination of the knowledge and ideas started from a few practitioners that came up with their own agile methods [Beck 00],[Sutherland 95]. These practitioners authored the first publications on agile methods - in form of books, articles and blogs. Other practitioners learned about the phenomenon from the first publications. Eventually, some of them also contributed to the body of knowledge by sharing their experiences with the community. However, the knowledge remains often anecdotic. It is difficult for external observers such as researchers, to understand the phenomenon.

Possible ways that researchers can follow to overcome this impediment is (1) by reading the literature authored by members of the agile community; or (2) by observing the process themselves. In their turn, researchers can publish the results of their investigations and thus contribute to dissemination and augmentation of the body of knowledge about agile value creation by:

1) Making knowledge about the phenomenon explicit and available to everybody;
2) Identifying good practices, and sharing them with the practitioners and clients, and
3) Proposing method improvements.

2.3 Disseminating the knowledge

To extract maximum benefit from an agile project, practitioners and clients would profit from a better understanding of the value-creating process and of the assumptions that are

made tacitly by the application of a method (or practices). For example, having an 'on-site' client is a well-known agile practice that leads to certain benefits. However, it is applicable only under the assumption that such 'on-site' client can be allocated as a resource. The question arises: what happens in those projects where the client's organization cannot afford to "free up" an employee to serve as 'on site' client?

Similarly, the assumptions behind a number of practices can be questioned. Some of those are:

- What happens in those projects where the client is supposed to, but in fact is not able to make decisions about the requirements? (For example, this could be the case if the client does not have the necessary knowledge, or is not aware of his/her needs.)
- How can the practice of 'good enough' documentation be implemented in an organization that is compliant with standards?

Thus, the following two requirements are an essential prerequisite for successfully applying agile methods:

1) making assumptions explicit;
2) identifying those project contexts where the assumptions would hold and those contexts where this is not the case.

As we see, both deal with the implicit character of the agile methods and practices. Thus, the considerations expressed above motivated us to ask ourselves: How can we – the researchers, contribute towards better understanding of the agile value-creation process?

2.4 Active client participation

A paramount characteristic of the agile methods is the involvement and reliance on clients' participation throughout a project. Clients are supposed to actively contribute to the process of value creation and in particular - to the decision-making about priorities. As the agile literature indicates, e.g. in [Augustine 05], never before in the software engineering history, the client has been that actively and constantly involved in the requirements reprioritization as he/she is in agile. When the client is expected to actively participate in the process by performing, among other tasks, the key task of prioritizing requirements, he or she must be aware of the facets of his/her role and thus would profit from an explication of the prioritization process.

In the following sections we state the problem that we investigate and the contribution we make for the practitioners' community.

3 Problem statement

This work is dedicated to exploring and understanding the process of value-creation for clients in agile projects, with a particular focus on the requirements prioritization and re-prioritization during a project, as an agile-specific value creation practice.

As we discussed in the motivation (section 2 of this Chapter), continuous and value-driven requirements reprioritization from client's perspective is one key aspect for the successful execution of agile software projects. A comparative study performed by other authors [Berteig 06] of this process and the prioritization practices in "traditional RE" indicates that one of the unique characteristics of agile requirements engineering consists in the intrinsic nature of the prioritization process and its value-based orientation. That is, prioritization is based mostly on business value, where the highest priority features (i.e. requirements in agile terminology) get implemented early so that most business value gets realized, while exposing the project to as low a risk as possible. However, researchers [Barney 08], [Petersen 09] in agile RE case studies also found that the creation of software product value through requirements prioritization decision-making is not completely understood.

We are addressing these challenges and issues by researching the following topics:

- How is business value perceived and measured in agile projects?
- What practices contribute to value creation in agile projects in different contexts?
- What concepts impact the re-prioritization decisions about requirements?

These questions represent the focus of our research activities. They lead and frame the formulation of our Research Questions and the research design.

We formulated two main Research Questions (RQs):

RQ 1. What concepts and relationships between those concepts characterize the value creation in agile projects? and:

RQ 2. How can the findings, gained in studying Research Question (RQ 1), be applied to other agile projects and how can they be used by practitioners and researchers?

The RQ 1. deals with understanding the phenomenon of the agile value creation as it happens in practice, while RQ 2. addresses the dissemination, generalizability and applicability of the extracted knowledge to other cases.

These research questions guide the study presented in this dissertation, and the structure of the results.

4 Research design

To answer our RQs we undertook a number of research studies. First, we explored the existing literature sources (Step 1). As we see further in this work, the results of this first step could not answer the first RQ 1. to our satisfaction. For this reason we undertook an explorative empirical study in different contexts. We performed a case study (Step 2), a modelling activity (Step 3) and a mapping activity (Step 4). The results are presented in Chapters 3 and 4.

To investigate the RQ 2., we not only analyzed the results obtained in the earlier phases of the study, but also undertook another research initiative and run a technical action research study (Step 5) to explore the applicability of our results to new project settings. The RQ 2. is discussed in Chapter 5.

We have indexed the research questions in the following way: the decomposed research questions that pertain to RQ 1 are numbered from RQ 1.1. to RQ 1.10., and those related to RQ 2 are indexed with RQ 2.1. to RQ 2.3. respectively. Table 1. below provides an overview of the research questions that have been investigated in the thesis, and maps them to the chapters.

At this point we want to point out that our interest in agile projects started before the studies described in this Chapter. We first looked at agile methodology as especially suitable for projects under uncertainty. In particular, we analyzed the similarities between real options analysis as another prominent method for decision-making under uncertainty, and the mid-course decision making in agile projects. This study can be considered as a preliminary and complementary step of the work described in this thesis, and for the sake of completeness we provide it in Appendix 2.

Table 1: Mapping between the decomposed research questions and the chapters of the thesis where they are discussed

Chapter	Research Questions
Chapter 3.2.: Understanding the concept of Business Value in a literature study	RQ 1.1. What concepts of business value are used in agile context, as described in the agile literature? RQ 1.2. In which way do agile projects create business value, according to the published agile literature?
Chapter 3.3.: Understanding the concept of Business Value in a case study	RQ 1.3. What concepts of business value do practitioners in the context of agile projects perceive? RQ 1.4. In which way do specific or individual practices influence the creation of business value? RQ 1.5. Do practitioner make value-driven decisions during agile development? If so, how decision-making is happening? RQ 1.6. How do developers combine value creation for their own organization with value creation for the client's organization?
Chapter 4.2, 4.3.: Understanding the Requirements Prioritization process in a case study	RQ 1.7. Who are the decision makers in the prioritization process? Which roles are involved and what are they responsible for? RQ 1.8. Which are the characteristics of the project settings that are essential for the way a requirements prioritization process is carried out in a project? RQ 1.9. Are there any other ways (beyond the selection of requirements) through which the requirements prioritization process adds value to the project?
Chapter 4.4: Identifying the gap between literature and practice	RQ 1.10. Which concepts of agile prioritization are shared in practice and in literature and how they are used to provide guidance for prioritization?
Chapter 5: Validation	RQ 2.1. Is the model from Chapter 4 usable? Here we mean: is the model understandable by the stakeholders? Can it be used in a software engineering context? RQ 2.2. Is the model useful and for what purposes? RQ 2.3. Which parts of the knowledge that we have explicated has been used in another project, and for what purposes?

5 Research Methodology

For the execution of our research plan, as outlined in Section 4 above, we exploited a number of research methods: systematic review of literature, case studies, technical action research. We will explain each of the methods in greater detail in the chapter where we apply it. In our opinion, this will make it easier for the reader to follow our narrative. In this section we want to explain the way we present the results of case study 1. This case study discusses many research questions pertaining to different facets of the topic under investigation – namely business value and requirements prioritization process in agile context. For this reason, we discuss the research questions in separate chapters. The results of the case study 1 are discussed in two different chapters, as presented below in Fig. 1.

We make the note that the case study composition and description is the same and is explained in Chapter 3, Section 3 – the first occurrence of the case study in the thesis.

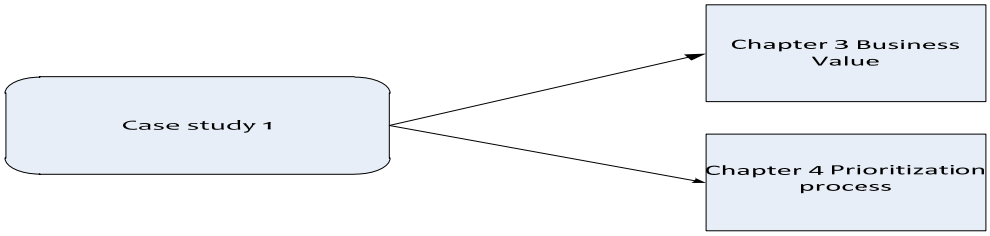

Fig. 1. Case study 1 and its results

6 High-level Structure of the thesis at a glance

In this section we provide the reader with an overview of the structure of the thesis and explain how the findings build upon each other.

Table 2. provides an overview of the Chapters and maps the research steps on them:

Table 2: Mapping between the chapters of the thesis and the research steps

Chapter	Research Steps				
	1	2	3	4	5
2. Background & related work	X				
3. Understanding the concept of Business Value	X	X			
4. Understanding the decision-making process		X	X	X	
5. Applying the findings to another project					X

After introducing and motivating our study, we investigate the topic of the understanding of *Business Value* in the agile literature. This study leads to the insight that certain agile practices contribute to the value creation. In particular, we identified and focused our further attention to the practices of:

- Continuous requirements prioritization,
- Active clients involvement,
- Waste reduction.

We furthered our research activity on each of these topics.
As our study is explorative in nature, the compound result consists in a set of insights, knowledge and lessons learnt.

As a last step we undertook an evaluation study to illustrate the applicability of the knowledge to other contexts and thus - its usefulness for the practitioners' community. We conclude with suggestions for future research. Fig. 2. captures the content in a diagram and makes explicit the relations between the themes discussed in the thesis.

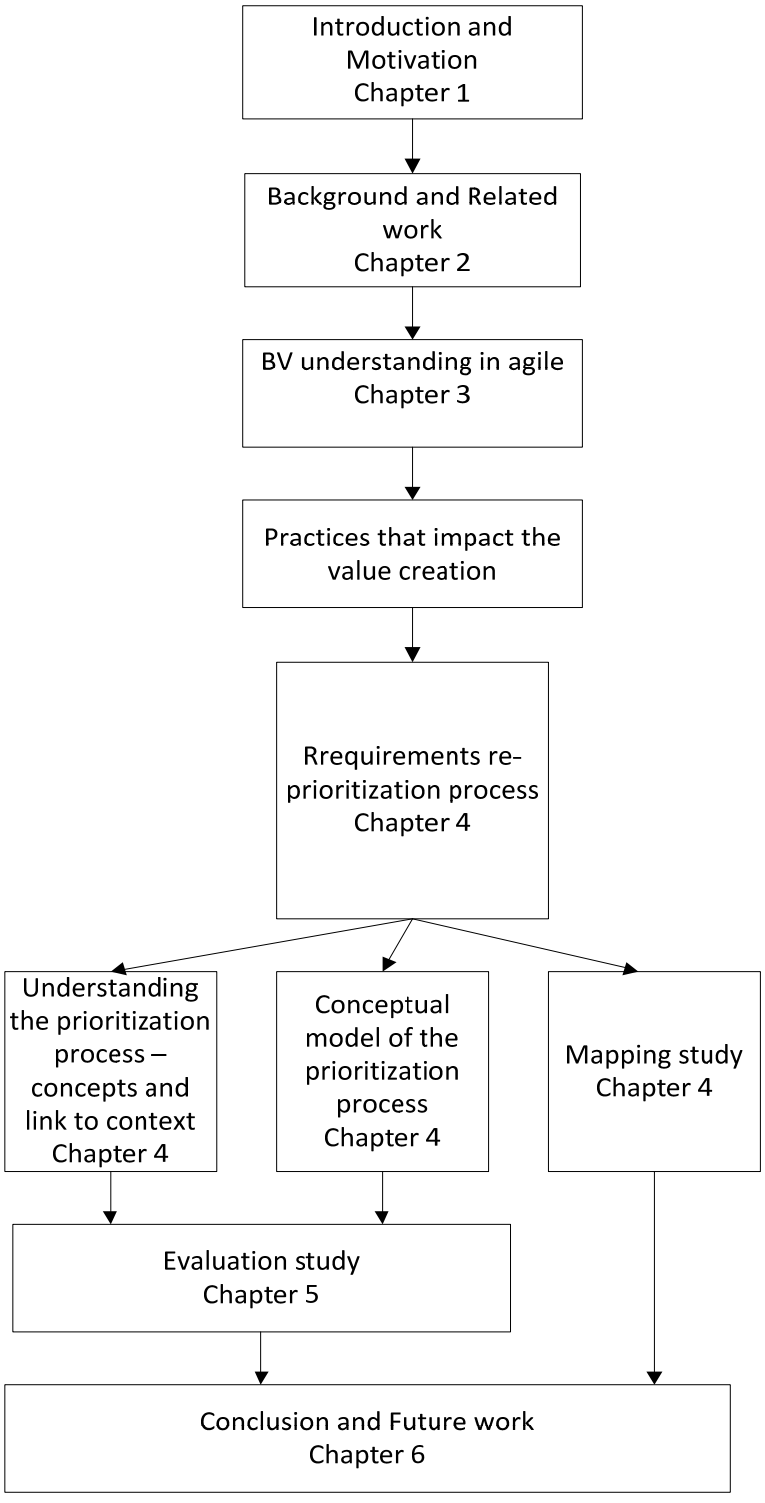

Fig. 2: High-level content of the thesis

7 Defining the Scope of the study

In this section we want to define what we understand under Agile software organization for the purposes of this study.

Agile methodologies encompass all facets of software development. This means that there are practices that pertain to all different software development and project management processes and phases. Obviously, this is too broad a scope to be discussed within one thesis.

In this work, we concentrate on understanding the mechanisms of the prioritization of requirements as a value-creating process. Our goal is to contribute to the contextualization of the agile practices. This means to explicate the application of an agile practice (i.e. prioritization) in different contexts. In doing so we uncover the assumptions that hold in the variety of contexts.

As described earlier, there is a wide variety of agile methods. Most of them are created and suggested by a single expert who promoted and disseminated the method further, e.g. [Beck 00]. On the other hand, our experience and the observations collected throughout this PhD study show that vast majority of companies that claim to work following the agile paradigm, don't follow exactly a concrete methodology. They rather implement a set of practices that best suit the company's context and culture. For this reason, we think that restricting our research to only one agile software development or project management method would represent quite a simplistic view on the topic. In order to make our results useful to a broader number of cases, we deliberately draw the line of our topic of study and took the following decisions:

- We don't restrict our investigation to a set of agile methods. Our study covers companies that claim to follow the agile philosophy as captured in the Agile Manifesto. Those are the iterative and incremental aspect of the development process, active client's participation and focus on the value creation.
- In spite of the broad set of methods and companies that we considered, we judged the agility of the companies before including them in our investigations and made sure we don't cover other iterative approaches like RUP.

We provide more details on the companies in the following chapters in the sections dedicated to the case studies settings and to validity and limitations.

8 Limitations

As the reader can see from the research questions and the structure of the thesis, the major part of our work is related to the study of business value and the process of requirements prioritization as a main vehicle for business value creation. It is obvious that we can't investigate within this thesis all possible ways in which an agile project creates business value for its clients. During our research on the topic we found it interesting and fruitful to have a closer look at other aspects of the value creation. In particular, we investigated the fit of Real Options Analysis (ROA) to agile projects. In Appendix 2 we take a closer look at the parallels between ROA and the agile decision-making situation and demonstrate that ROA could be used as an explanatory mechanism about the agile decisions about requirements under uncertainty. However, the question of decision-making under uncertainty is not central to our topic. We consider it as an interesting aspect and for this purpose we include it in the thesis.

9 Contribution

As explained in section Research Design, the investigation of the research topic included a number of related studies. Although they all were dedicated to different aspects of the agile value creation process, they helped to create an overall, clearer and explicit picture of the phenomenon. Thus the studies and, respectively - the results, are complementary. We aggregated the results of the separate studies and this lead to deeper understanding about the process of value-creation.

The main contribution of our work to the research and practitioners' communities consists in explicating the process of requirements prioritization in agile projects, as the major vehicle for value creation. We make the note that below we only present the main contribution and we will provide a detailed list with our findings in the concluding chapter.

The contributions for the practitioners are:

- Explication of tacit knowledge. In particular, we contribute by better understanding of the agile requirements reprioritization process in terms of concepts that impact the process, the decision-makers, and the link between the project context and the process instantiation.

- The conceptual model of the prioritization process not only made explicit the concepts that impact the process, but also the relations between them. We believe that this could help the decision-makers, and in particular the clients, to reason about their specific project situation. Furthermore, the model can be used as a decision-making framework that helps to structure the decision-making process and the discussions between the stakeholders. We present an example of such application in Chapter 5.

- The explication of the link between context and agile prioritization practices can help practitioners to become aware of their concrete project context and to consider using a sub-set of practices depending on the context.

The contributions for the research community are:

- A conceptual model of the agile reprioritization process.

- A conceptual model of the impact of organizational maturity on the clients' value creation in agile projects.

- An identification of a gap between the guidance that the literature provides for the prioritization process, and the real-life process as we observed it in a case study.

Below we provide an overview of the main contributions and the papers they resulted in.

1. Identifying the current requirements prioritization methods, and analyzing their weaknesses. The results are captured in the paper:

   Racheva, Z. and Daneva, M. and Buglione, L. Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products In: Second International Workshop on Software Product Management (IWSPM) September 2008

2. Investigating the understanding of Business Value in agile projects.

   Racheva, Z. and Daneva, M. and Sikkel, K. Value Creation by Agile Projects: Methodology or Mystery? In: 10th International Conference on Product-Focused Software Process Improvement (FROFES 2009) 17 June 2009

   Racheva, Z. and Daneva, M. and Sikkel, K. and Buglione, L. Business Value Is not only Dollars - Results from Case Study Research on Agile

Software Projects In: Product-Focused Software Process Improvement, 11th International Conference, PROFES 2010

3. Conceptual model of the agile decision-making at inter-iteration time

   Racheva, Z. and Daneva, M. Reprioritizing the Requirements in Agile Software Development: Towards a Conceptual Model from Clients' Perspective In: Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering, SEKE'2009, July 2009

   Racheva, Z. and Daneva, M. and Herrmann, A. and Wieringa, R.J. A Conceptual Model and Process for Client-driven Agile Requirements Prioritization
   In: Proceedings of the Fourth International Conference on Research Challenges in Information Science, RCIS 2010, May 2010

   Racheva, Z. and Daneva, M. and Herrmann, A. A Conceptual Model of Client-driven Agile Requirements Prioritization: Results of a Case Study In: IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), September 2010

4. An in-depth analysis of the requirements prioritization in agile context.

   Racheva, Z. and Daneva, M. and Herrmann, A. and Sikkel, K. and Wieringa, R.J. Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study In: 18th International IEEE Requirements Engineering Conference, October 2010

5. Identifying a gap between the state of the practice and the literature description of the agile requirements prioritization.

   Zornitza Bakalova, Maya Daneva, Andrea Herrmann, Roel Wieringa: Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature. In REFSQ, March 2011, pp. 181-195

6. Exploring the application of Real Options Analysis to the agile decision-making

   Racheva, Z. and Daneva, M. and Buglione, L. Complementing Measurements and Real Options Concepts to Support Inter-iteration Decision-Making in Agile Projects In: 34th EUROMICRO Conference on Software Engineering and

Advanced Applications (SEAA) - Software Management Track, 03 September 2008

Racheva, Z. and Daneva, M. How Do Real Options Concepts Fit in Agile Requirements Engineering? In: Eighth ACIS International Conference on Software Engineering Research, Management and Applications, SERA 2010 May 2010

10 Validity

In this section we discuss possible threats to validity of the results and how we planned to manage them.

We make the note that at each step and each separate sub-study we discuss separately the threats to validity that pertain to the

respective study, and the measures that we undertook to mitigate them. Further, for each study, we discuss how our threat management worked out, and how this affects the results that we obtained.

Research methodologists [Wieringa 09-1], [Yin 04], advise to consider and mitigate the following possible threats to validity:

- Construct Validity
- Internal Validity
- External Validity

10.1 Construct Validity

"Construct validity involves making inferences from sampling particulars of a study to the higher-order construct they present" [Shadish 02 p.65]. It is the extent to which the variables that we measure, represent the concepts we are interested in. In order to achieve high construct validity, it is important to operationalize in an unambiguous and detailed way the concepts of interest, and to assess the match between the operationalization and the concepts of interest.

In a case study construct validity is especially problematic because of potential investigator subjectivity. Yin [Yin 04], proposes three remedies to counteract this: using multiple sources of evidence, establishing a chain of evidence, and having a draft case study report reviewed by key informants.

We mitigate the threat of construct validity by taking the following actions:

- We made efforts to refrain from subjective judgments;
- Stayed as close to the interview scripts as possible, providing citations; even using "in-vivo" codes [Charmaz 07];
- Let case study participants and researchers review and discuss the data analysis and the reports with the findings [Racheva-1 10], [Racheva-2 10].

This way we checked that our data represent reality, rather than our own opinions. We constructed concepts bottom-up, from our data, using grounded theory. We attempted to apply the research method in a rigorous manner and thus to achieve construct validity by construction.

10.2 Internal Validity

In qualitative research, internal validity relates to the quality [Seale 99], rigor [Davies 02] and trustworthiness [Hipps 93] of data. The reader should be able to trace the results of the research back to the data. It should become clear that the results relate to the phenomenon under investigation and are not caused by other incidental causes.

We assure internal validity by the following measures:

- Reflexivity and self-checking, that is continuous reflection on the research process and evaluation of the data obtained [Pyett 03].

- Peer debriefing – discussions on the research methodology and the review of the data by other researchers familiar with the subject under investigation, as well as partial analysis of the data by two researchers and comparison of the results.

- Member checking – examination of data and interpretations by participants in the case study.

10.3 External Validity

The external validity deals with the generalizability of the results, i.e. deals with the question: are the results applicable to other cases, contexts and situations. According to Wieringa [Wieringa 12-1], we can rather talk about the re-usability of the results. In qualitative research, external validity can be discussed not in terms of statistical but in terms of analytical generalization. In this case it is a matter of the fit between the studied context and the context of the case of intended application of the results. In order to be able to successfully match her or his context to the concepts identified in our study and to the practices that we distilled, a practitioner has to identify the context first. There is a need to lay out the context, not in data terms, but in relationship terms – causal relationships between the concepts and the context. Changes in the contextual integrity need to be identified and the set of practices - re-assessed. In other words, the assumptions about particular context should be scrutinized and the questions should be asked: what happens if the context changes?

We are aware of the importance of this question. For this reason in Chapter 6 we propose as important area for future work the creation of a definition and a simple framework that could be used for comparing contexts and context matching.

10.4 A note about Conclusion validity

Conclusion validity [Wohlin 00] is the degree to which conclusions we reach about relationships in our data are reasonable. It focus on how sure we can be that the treatment we used in an experiment really is related to the actual outcome we observed. In quantitative studies, typically this concerns if there is a statistically significant effect on the outcome. In this thesis we discuss a case study that works only with qualitative and not with quantitative data. We therefore, follow the guidelines of Yin [Yin], according to which in qualitative studies, conclusion validity is part of the discussion about generalizability of the results. In our case, we have addressed the aspect of generalizability in Chapter 3, Section 3.

Chapter 2

Background, Related Work and Problem Investigation

This chapter provides the reader with background on agile software development and explains the most important terms and notions that we use in this thesis. The purpose is to set up and describe the context of our study. Furthermore, we provide an overview of the related work, performed by other authors, on the topics of: Value Based RE, Requirements Prioritization in agile projects, and methods for decision-making under uncertainty. Last, we explain what this knowledge means for our research and how we build upon it.

1 Introduction

The purpose of this chapter is to set up the context of the thesis by outlining the specifics of agile SD and agile decision-making, and to position our research against the studies performed by other authors. The chapter is structured as follows:

1) In Section *Background*, we explain agile SD, and the agile practices that are relevant for the topic of our study. That is – the agile requirements engineering (RE) process, the requirements prioritization process as a decision-making process about requirements, and the clients as the main decision-makers about requirements in agile projects. We refine the description from the higher level area

of the Agile SD, we zoom into the agile RE, and then – into the requirements prioritization process as a decision-making process. Graphically, the relation is presented at Fig. 3.

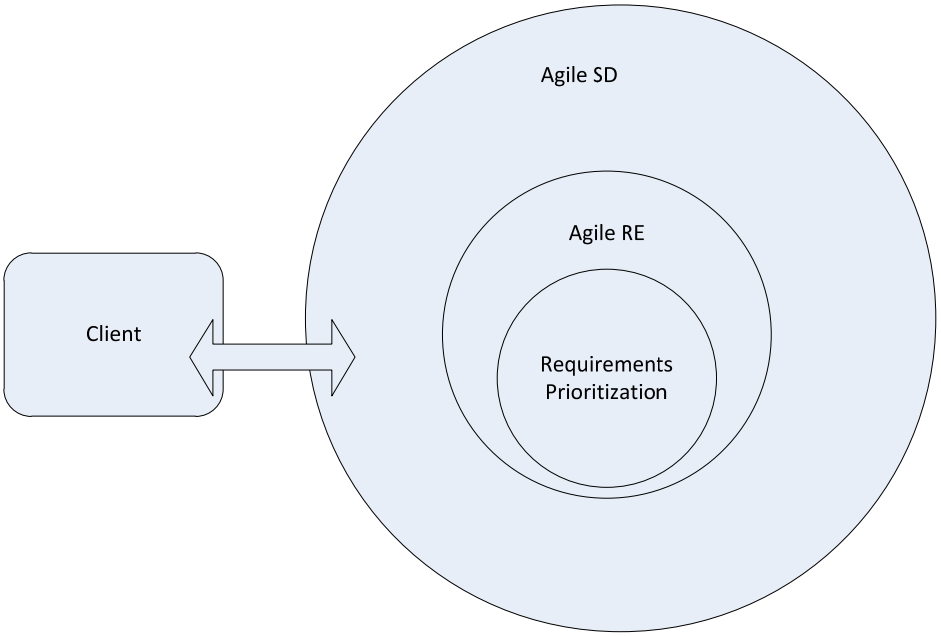

Fig. 3: Relation between the concepts covered in section Background

2) Section *Related work* summarizes the work, performed by other authors, on the topics of: Value Based RE, Requirements Prioritization in agile projects, and methods for decision-making under uncertainty. Next, we explain how we build upon these areas of knowledge, and in which aspects our research differs from the sources that we discuss in this section.

   Firstly, as the focus of this thesis is the requirements prioritization process in agile projects and how it contributes to the value creation, we looked at the current state of the agile RP methods as described in literature. Although there are a significant number of methods, as we will see in Section 3.1, they all rest on the assumption that the business value of the separate requirements can be determined, even more – is quantifiable and can be used as an input for the decision-making process. Furthermore, we observed that only two methods explicitly consider the viewpoint of the client. We could find no method description that considers broader project aspects during the decision-making, and that states explicitly how the decisions are made, what concepts and aspects from the project and the project environment impact them. All of the above motivates

the need of empirical study to better understand the phenomenon of agile inter-iteration decision-making.

As we'll see below in the analysis of the related sources on prioritization techniques in agile context, the authors of the methods and the papers don't discuss the nature of the decisions that the stakeholders face at inter-iteration time. The focus in these publications is on the process (in terms of steps) of obtaining a prioritized list with requirements from an initial list, under the assumption that all data and values, necessary to perform the steps, are available. However, there is no discussion on the decision-making situation that the stakeholders face while having to making prioritization decisions. In its essence an agile project deals with changing environment and is a means to manage uncertainty. In order to better understand what type of decisions the stakeholders face, we looked at methods for decision-making under uncertainty that find application in other fields.

Agile SD is explicitly linked to value creation as a development philosophy. In Section 3.2 we screen the related literature on Value Based RE to compare and position our work among the research performed by other authors on this topic.

2 Background

2.1 A note about terminology

Some of the agile methods use their own terms to name specific practices or artifacts. For example, in one of the most prominent agile project management methods – SCRUM, the iteration is called 'sprint', and the iteration backlog is called 'sprint backlog'. As we don't consider any particular agile method, we try to use as common a terminology as possible. However, for illustrative purpose and in the models derived from real-life data, we present examples of terms used in concrete methods, and provide their equivalent meaning.

2.2 Agile software development

Agile approaches to software project delivery and to software product development can be considered a paradigm, a project management philosophy, a culture, an attitude, and a state of mind. Compared to 'traditional' software development, they rest on completely different understanding about the values and principles that represent the foundation of

the development method. Namely, all agile methodologies rest on the 'minimalist' principle of organizing work in the software development process, meaning a conscious choice in carrying out those tasks which directly create value for clients and leaving out anything that is deemed "waste" [Dyba 08]. The latter refers to all work and work products not directly contributing to the development of the desired software, for example spending time on implementing features that are not specified by any user story or on producing an artifact not explicitly asked by the clients. The 'minimalist' principle is fundamental to the ability of the agile approaches to cope with project uncertainties. In that sense, this principle can be seen as a reaction to the 'plan-based' paradigm which assumes that problems are fully specifiable and that predictable solutions exist for every problem [Dyba 08].

All agile software approaches share the same 'minimalist' principle, but, despite that, not all of them are directly comparable in terms of scope and content. For example, an important distinction exists between agile software development (ASD) and agile project management (APM) approaches. While the first class of approaches are defined as "evolutionary approaches which are collaborative and self-organizing in nature, producing high-quality systems that meets the changing needs of stakeholders in a cost effective and timely manner" [Abrahamsson 02], the APM approaches are defined as "the work of energizing, empowering and enabling project teams to rapidly and reliably deliver customer value by engaging customer and continuously learning and adapting to their changing needs and environments". We make the note, however, that in this thesis we treat ASD and APM practices in the same way. That is, when we use the term 'agile practice' we mean a practice which can be part of either software development or project management. In the next sub-section, we narrow down the discussion to the concept of business value, as business value is what motivates the adoption of agile practices in the first place.

The agile process consists in short iterations and relies on fast feedback, incorporating new knowledge and embracing changes. The main idea of the iterative agile approach is represented on Fig. 4.

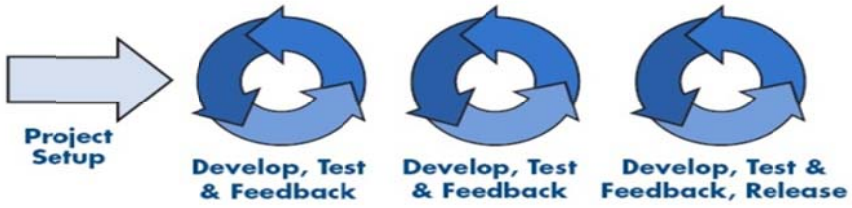

Fig 4: Agile Development Process – generic view

Source: www.definedconsulting.com

There is a number of different Agile Methodologies, for example Extreme Programming [Beck 00], Scrum [Schwaber 01], Test Driven Development [Beck 00], and others. They differ in the concrete practices that they follow. Comparing the methods and listing the practices they use has been done by others and we will not focus on this topic here [Dyba 08]. We want to make the note that for the purposes of our investigation we don't restrict ourselves to any particular agile method or approach.

2.3 Agile Requirements Engineering - Principles and characteristics from RE perspective

The Agile approaches, such as Extreme Programming or SCRUM, advocate requirements engineering through the software product development cycle in small and informal stages. That is, instead of engineering the requirements upfront, one lets requirements emerge during development. Agile software process practitioners deem this approach particularly valuable for software producers in a context that includes highly uncertain requirements, experimentation with new development technology, and clients willing to explore the ways in which an evolving product can help their business goals. Furthermore, this allows for the opportunity to incorporate changing requirements at each iteration, to react to new knowledge about the business realities, the development platform, or other project-related insights.
Agile requirements engineering has in place special mechanisms that make the reaction to changes possible. Those are:

- The dynamic character of the list with requirements (the project backlog and the iteration backlog)
- The process of requirements prioritization as a means to determine which sub-set of requirements shall be implemented during each iteration.

We elucidate these two mechanisms in more depth in the following two sections.

2.4 Agile requirements prioritization

Clearly, requirements prioritization is a part of any project, independently from the developing method. Yet, the purpose and the place of this activity are essentially different when we distinguish between 'traditional' and agile development. In a 'traditional' (e.g. gated or waterfall-style life cycle), it is about which features (i) to implement earlier than others, or (ii) to include in an earlier release. The premise is that the whole functionality cannot be implemented in the same time, but it will eventually be implemented. So it is a project-management activity from the developers' side. When asked about priorities in a

'traditional' project, the customer tends to qualify the majority of the requirements as *high priority*.

In contrast to 'traditional' development, agile projects rest on the understanding that the whole functionality will not be specified at the beginning of the project, and will be implemented in small increases (iterations). Furthermore, there is the possibility that part of the functionality will be eventually *not* implemented at all, because changes and learning are allowed. The problem, then, is: (i) how to decide on what to implement in each (next) iteration, and (ii) which requirements will deliver the maximum value to the customers as early as possible.

One of the biggest assets of an agile approach is that business value is delivered to the client throughout the project, and the return on investment is generated much earlier. Thus any changes in the requirements can be taken into consideration and implemented in the product soon after they emerge. This highlights the paramount importance of the prioritization activities. If we compare agile RE and 'plan-based RE', one notices two important differences [Abrahamsson 02]: (i) (re)prioritization in agile RE happens at inter-iteration time, which means that the project team anticipates and plans as many reprioritization sessions as the number of project iterations, and (ii) (re)prioritization in agile RE is based mostly on business value, that is, the highest priority features get implemented early so that most business value gets realized. In contrast to the traditional development, where the prioritization is a project management activity aimed at organizing the work during the project, in agile settings the prioritization assumes much more important role. From organizational tool it becomes value-creating mechanism. In traditional setting, the developing team and in particular the project management, business analysts and architects are the ones performing the prioritization. The nature of agile prioritization requires a shift in this respect. As business value is the main criterion when it comes to prioritization decisions, the customers are the one who know what goals should be achieved by employing the system under development. That's why they play a central role during the agile prioritization process.

Changes in the list of requirements for an iteration might occur for different reasons – new market or company realities or better knowledge about the value certain features deliver. This requires a dynamic prioritization process as well. This view is supported by Harris and Cohn [Harris 06]], who use tactics to minimize costs and maximize benefits through strategic learning and provide guidelines on how to optimize business value. They prove the necessity of adopting a dynamic approach to agile prioritization, in order to take into consideration the important aspect of learning in an agile project.

To summarize, the agile requirements prioritization process is:

- Dynamic and iterative
- Requires active customer participation
- Based on value considerations

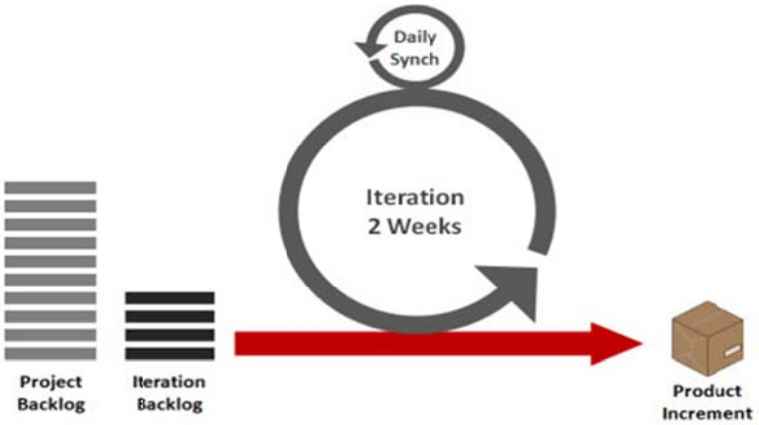

Fig. 5: Agile prioritization process

Source: www.dtsagile.com/Content/Iteration_*mechanics*.jpg

Fig. 5. is a visual representation of the agile process at requirements level. It shows that from the initial list with requirements – the project backlog, a sub-set has been chosen – the iteration backlog. After the implementation (and at the end of the iteration, respectively), a small part of the final product has been developed. This is represented by the 'product increment'.

To illustrate how agile projects proceed, we describe below an example of how Scrum and XP – two of the most popular agile project management methodologies [Schwaber 01], [Beck 00], treat requirements prioritization. Scrum is an iterative and agile incremental process model including values, artefacts, roles and meetings. The main roles in Scrum are:

- the "Scrum Master", who ensures that the Scrum process is used as intended and who enforces the project management rules;
- the "Product Owner", who represents the stakeholders;
- the "Team", a cross-functional group who perform the work activities as the actual analysis, design, implementation, testing.

The project starts with a product backlog which is an initial requirements list and is prioritized by business value. It also contains rough estimations of development effort. Business value is set by the Product Owner. Development effort is set by the Team. The project is executed in time units of iterations (the so-called "sprints") which are two to four weeks long. Each iteration starts with a sprint backlog which contains only those

requirements which are to be implemented during this sprint. We note that this sprint backlog should be frozen and not modified until the sprint is over. This means that (i) re-prioritization takes place during the sprint planning meeting at the beginning of each sprint only, and (ii) after this point of time no re-prioritization takes place during the daily Scrum meeting. At this meeting, business values and development effort of the requirements are re-estimated and the sprint backlog for the next sprint is created. This is performed by the Product Owner with the participation of the developers. At the end of a sprint cycle, two meetings are held: the "Sprint Review Meeting" (where the completed work is presented to the stakeholders) and the "Sprint Retrospective" (which serves the objective to make continuous process improvements).

Similarly, in XP each iteration is treated as a timebox with a fixed duration. The iteration planning, then, means: (i) how to decide on what to implement in each (next) iteration, and (ii) which requirements will deliver the maximum value to the clients as early as possible while minimizing project risk and respecting the project constraints. The decision-making process implementing the principle "business value first" is as follows: One starts with the requirement deemed to have the highest business value, and then checks whether it is within the (iteration) budget. If yes, one continues with the second-highest value requirement. Is it also within the budget? If so, one continues. If no, one stops. In XP's prioritization procedure – the so-called Planning Game – one complementarily applies the "business value first" principle and the "worst things first" principle [Beck 00] which means that requirements involving high technical risk should be implemented early. The rationale behind this is risk reduction: the earlier one implements the riskiest requirement, the earlier one gains more certain knowledge about this requirement's implementation effort.

2.5 The role of clients in agile decision-making and client-centric agile requirements prioritization process

As we saw in the previous section, one of the biggest assets of an agile approach is that business value is delivered to the client early and regularly throughout the whole project, and the return on investment is generated much earlier. Thus, any changes in requirements can be accounted for and built into the product at an early stage. This highlights the paramount importance of the RP activities. Moreover, in contrast to traditional projects where the RP is usually performed once and before the implementation phase, in agile context RP is an on-going process happening at iteration's start. This reflects the dynamics of the product backlog. The client is the key driver of this process, being supported by a project manager (called, for example, a 'scrum master'), while in the traditional

development this task is performed by the developer, with the participation of a project manager and other stakeholders.

The Agile Manifesto [Agile] explicitly deems the client's role critical in making decisions about "what to build". XP [Beck 00] recommends the following for the client's role:

1) The client is an integral part of the team and should be on-site with the team.

2) The client writes user stories and then discusses each requirement directly with the programmers.

3) The client is responsible for all business decisions, including RP.

4) The small 2-3 weeks iterations allow the clients to evolve their requirements based on concrete working software.

5) The client regularly tests the software to confirm it works as expected.

Although team work is a guiding principle in agile development, it is the client who makes the final decisions. In the decision-making process about requirements priorities, the development team takes the role of advisor by estimating cost and judging technical risk.

In this thesis, our focus is on the role of the client as described in item 3 of the above list. That is, we want to support the clients in their function as decision-makers, as well as other stakeholders that perform this function (e.g. Product Owners).

2.6 Requirements documents and requirements backlog - User stories as agile requirement documents

This section aims at outlining the specifics of the agile requirements documents.

The management and organization of the requirements differs significantly in agile and in traditional context. As we mentioned earlier, the agile methods don't propagate big requirements up-front. Instead, the requirements themselves are captured usually in a concise form, called User Story. Those are light-weight documents that capture the main functionality and value of the documented feature. The requirements for the whole project are called project backlog and it is represented as a stack of User Stories. From this stack the requirements with the highest Business Value are chosen for implementation in the upcoming iteration, and this sub-set of User Stories is called 'iteration backlog'.

User stories usually follow the following template: [Cohn 05]

As {role of the user of the system}

I would like to {function}

So that {value or business goal served by the demanded feature}.

Very often the USs are physically represented on cards and on the back of the card the acceptance criteria are documented.

The purpose of USs is to serve as a reminder for a conversation with the client on the topic, rather than as an exhaustive document that has the value of a contractual agreement.

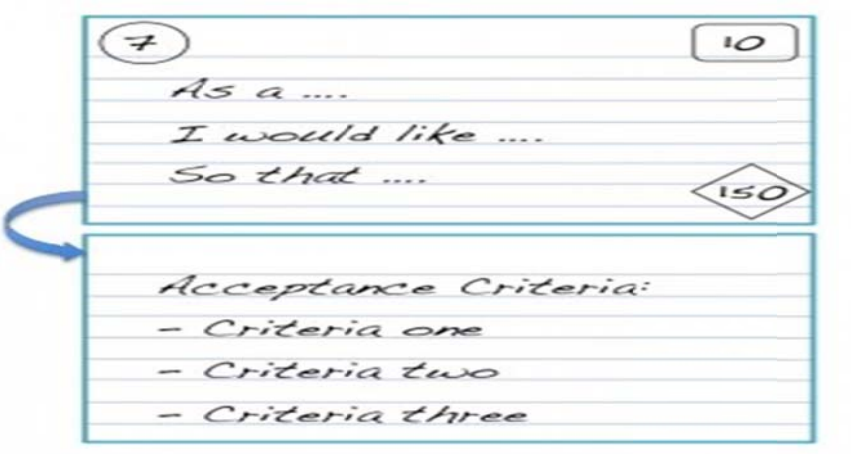

Fig. 6: Format of User Stories
Source: http://blogs.kent.ac.uk/agile/2009/09/22/planning-involving-your-customer-with-user-stories/

Although the agile methods are spreading fast in the industry and are getting increasing attention from the research community, the topic of the quality of USs seems to be under-researched. Advocates of agile methodologies claim that those reduce waste by implementing only those requirements that bring value for the customers. For USs to help reduce waste, they should satisfy six quality criteria [Cohn 05] the most important one being that USs should be Valuable (to customers or users). The literature sources on agile methodologies recommend that ideally, the value should be stated explicitly on the story card.

3 Related Work

In this section we summarize the research results of other authors that have worked on topics related to ours. We have investigated three domains:

- Current prioritization methods that have found application in agile projects. In this thesis we investigate the agile prioritization process as a value creating activity. Although we don't propose a new prioritization method, we build upon the knowledge on the existing prioritization techniques and use it twofold:
  - To capture the current state of the practice as documented in literature and to identify problems related to it. We present this analysis in section 4 of this chapter.
  - To identify the gap between the guidance that the literature sources provide with respect to requirements prioritization, and the prioritization practice as we observed it in a case study.
- Value Based Requirement Engineering (VBRE)

  Our research regards the RE process in agile projects and we look at it from the perspective of value-creation. For this reason we screened the VBRE literature to identify points of convergence and divergence between our research and those of other authors. By positioning our research among those of other RE authors we draw the conclusion that our work is complementary to those of other authors and represents an enrichment for the RE body of knowledge.
- Methods for Decision-making under uncertainty.

  As the agile approaches for SD are a means to cope with the project uncertainty, we looked at other practices for decision-making in uncertain and changing environment. We use this knowledge to identify possible parallels with other existing approaches. In our opinion, this would help the stakeholders, and in particular the clients, (i) to become aware of the type of decisions they face; and (ii) to possibly improve the outcome of their decision-making process by proposing to consider the decision situations as options. In Chapter V we discuss this question in-depth.

3.1 Current approaches for requirements prioritization in agile context

In this section, we present a summary of agile requirements prioritization methods that can be found in the literature. We use the knowledge that we synthesize in this section twofold: first, to understand the state of the practice as described in the literature, and second – to compare the methods with the model that we present in Chapter 4 and analyze the gap. We make the note that we don't modify the methods, nor do we suggest a new method later in the thesis.

3.1.1 The sources

We did a semi-systematic literature review to select the related work presented in this section. Our literature review included a broad search of academic and practitioners' information sources. We did a publication search using the five bibliographic databases: IEEExplore, ACM Digital Library, Google Scholar, InterScience and Citeseer.

We additionally included three professional magazines: the Agile Journal [AJ], and the platforms dedicated to software development and agile methods: Dr. Dobb's [Dr Dobb's] and InfoQ [InfoQ]. Our strategy for searching literature sources in these databases and magazines included (1) the use of subject keywords and (2) the use of citations. To define the search strategy we implemented the recommendations of Webster et al [Webster 02] for searching in literature databases.

First, the key words we used for our search were: *agile* plus one out of the following: *requirements*, *backlog*, *prioritization*, *inter-iteration decision-making*. We deliberately did not include any literature sources on non-agile iterative development methodologies, as RUP, for example. Second, we traced the references in the identified papers to get access to other relevant sources. To determine the relevance of these sources to our research, for each one, we reviewed the abstracts and the conclusions and we checked this information against the following four quality criteria for inclusion in the review: (1) the paper is on a agile RP, (2) the paper is credible, i.e. the method described is meaningful and intuitive to follow; (3) relevance for practice: the RP method potentially offers support for practical RP, (4) the paper adequately describes the context, in which the method is expected to be applicable; 'adequately' means that the reader can judge about the use of the requirements prioritization method in his/her own context. The publications were written in English only and included qualitative research papers as well as experience papers, from scientists and practitioners. We carried out the quality check by using these

criteria, which yielded 45 papers. Additionally, we analyzed books about agile development [Bleek 08], [Highsmit 04], [Hunt 06], [Lippert 02].

Altogether, our sources refer to 22 RP techniques, as indicated in Table 3.

We want to stress that we refer to these prioritization methods as agile ones, as we have found evidence about their use in an agile development context – i.e. in an iterative and incremental process, following an agile methodology like SCRUM, XP, etc. Nevertheless, the application of those methods is not restricted to agile settings, as many of them have been used in traditional software development as well [Highsmit 04].

3.1.2 The methods

Table 3: The RP approaches published in the literature sources

RP method	**References**
Round-the-group prioritization	[Berteig 06]
Ping Pong Balls	[Schwaber 04]
$100 allocation (cumulative voting)	[Leffingwell 03]
Multi-voting system	[Tabaka 06]
MoSCoW	[Oracle 07]
Pair-wise analysis	[Gottesdiener_1] [Karlsson 07]
Weighted criteria analysis	[Gottesdiener_1]
Analytic Hierarchy Process (AHP).	[Saaty 80]
Dot voting	[Gottesdiener_1]
Binary Search Tree (BST)	[Ahl 05]
Ranking based on product definition	[Fraser 02]
Planning Game	[Beck 00] [Karlsson 07]
Quality functional deployment (QFD)	[Crow 09] [Gottesdiener_1]
Wieger´s matrix approach	[Wiegers 99]

Mathematical programming techniques	[Li 07]
Kano Model	[Quality 93] [Cohn 05]
Relative weighting	[Cohn 05]
Theme Screening/Scoring	[Cohn 05]
Planning Pocker	[Cohn 05]
Binary Priority List (BPL)	[Bebensee 10]
Hybrid strategy	[Port 09]
Technique of bucketing requirements	[Patton 05]

Below we describe each of technique in terms of its core idea and context of use (whenever details are provided by the authors):

1. **Round-the-group prioritization** [Berteig 06]. Items are written on cards and placed in random order linearly either vertically or horizontally. The members of the group each take turns placing the items in the order they think is the proper priority order. While doing so, each person moving the cards is welcome to explain their reasoning. However, the other group members refrain from commenting on the new prioritization. This continues around the group as many times as it takes to find a stable order. The context of using this method includes group size of 3 to 8, and item list sizes less than 15.

2. **Ping Pong Balls** [Schwaber 04]. A fixed number of ping pong ball units are given to the group. The ping pong balls represent units of one dimension for prioritization such as value, risk or cost. The group discusses how to allocate ping pong balls to each item in a dynamic fashion until everyone agrees that the allocation makes sense. For very large lists, this is easiest to do in a spreadsheet with fewer people involved. This method is appropriate for projects where 1 to12 participants take part in the prioritization effort, and for more than 15 requirements.

3. $100 allocation (cumulative voting) [Leffingwell 03]. Stakeholders get a fictitious $100 to spend on requirements. After they allocate their money, tally the total for each requirement and then that total is divided by the number of stakeholders. The requirements are then rank-ordered, with the highest totals being most important.

4. **Multi-voting system** [Tabaka 06]. This method uses elements of cumulative voting. A person can put multiple votes on a single item and can withhold some or all of her votes. After everyone has "finished" voting, the facilitator calls for everyone to step back and think about the results. Some discussion is allowed about the consequences of the results. Finally, everyone is given an opportunity to move their votes. The optimal group of participants might include between 5 and 20, and the item list size should not exceed 50.

5. **MoSCoW** [Oracle 07]. This is a widely used method, close to the *Numeral Assignment Technique* for traditional prioritization, where the items are roughly classified in *Priority groups* depending on importance. The letters stay for: M - MUST have this, S - SHOULD have this if at all possible, C - COULD have this if it does not effect anything else, W - WON'T have this time but would like in the future. The importance of this method is that when prioritising the words mean something and can be used to discuss what is important.

6. **Pair-wise analysis** [Gottesdiener_1], [Karlsson 07]. According to this method, the requirements are ranked by comparing them in pairs until the top requirements emerge at the top of the stack. This method functions for relative small number of requirements, where direct comparison is possible.

7. **Weighted criteria analysis** [Gottesdiener_1] Criteria are defined, and weights are assigned- so that some have higher value than others. Numbers are assigned to each weighted criterion to arrive at a total score for each requirement.

8. **Analytic Hierarchy Process** (AHP). AHP was developed by Saaty and applied later to software engineering [23]. AHP is a method for decision making in situations where multiple objectives are present. This method uses a pair-wise comparison matrix to calculate the relative value and costs of individual requirements to one another. By using AHP, the requirements engineer can also confirm the consistency of the result. AHP can prevent subjective judgment errors and increase the likelihood that the results are reliable.

9. **Dot voting** [Gottesdiener_1]. Stakeholders are assigned sticky dots (which can also be colour-coded) to allocate to requirements. One counts the dots to narrow the list of requirements.

10. **Binary Search Tree** (BST) [Ahl 05]. This is an algorithm that is typically used in a search for information, and which can easily be scaled to be used in prioritizing many requirements. The basic approach includes the following steps: (1) Put all requirements in

one pile. (2) Take one requirement and put it as root node. (3) Take another requirement and compare it to the root node. (4) If the requirement is less important than the root node, compare it to the left child node. If the requirement is more important than the root node, compare it to the right child node. If the node does not have any child nodes, insert the new requirement as the new child node to the right or left, depending on if the requirement is more or less important. Steps 3-4 are repeated, until all requirements have been compared and inserted into the BST. (5) For presentation purposes, traverse through the entire BST in order and put the requirements in a list, with the least important requirement at the end of the list and the most important requirement at the start of the list.

11. **Ranking based on product definition** [Fraser 02]. This prioritization technique accounts for three important perspectives on product definition: the business, users, and technology. Using it, stakeholders rank the importance of each feature to the business, user experience professionals rank the importance of each feature to the users, and technical analysts rank the feasibility of implementing the feature. Once they have all ranked the features, the product team combines the different rankings and compares them to identify the most important features. The success of this technique is contingent on the involvement of the right people providing input on the right aspect of each item on the list. This simple process provides a quantitative, de-personalized way to arrive at rational, actionable, and — above all — realistic launch priorities.

12**. Planning Game** is a feature of extreme programming [Beck 00], [Karlsson 07] and is used with customers to prioritize features, based on stories. This is basically a variation of the Numeral Assignment Technique, where the customer distributes the requirements into three groups, "those without which the system will not function", "those that are less essential but provide significant business value," and "those that would be nice to have. The process is based on two criteria: business value judged by the customer and technical risk judged by the developers.

13. **Quality Function Deployment QFD** [Akao 94] [Crow 09] is a structured methodology for taking into account customer needs (i.e., the "voice of the customer"). You create a product planning matrix known as the "house of quality," which reflects both *what* (customer needs) and *how* (designer needs).

14. **Wiegers' matrix approach.** Karl E. Wiegers [Wiegers 99] describes a semi-quantitative analytical approach that uses a simple spreadsheet model to help estimate the relative priorities for a set of product features. This approach distributes a set of estimated priorities across a continuum, rather than grouping them into just a few priority

levels. This prioritization scheme borrows from the QFD concept of customer value depending on both the customer benefit provided if a specific product feature is present and the penalty paid if that feature is absent. A feature's attractiveness is directly proportional to the value it provides and inversely proportional to its cost and the technical risk associated with implementing it. All other things being equal, those features that have the highest risk-adjusted value/cost ratio should have the highest priority.

15. **Mathematical programming techniques for release planning.** Li et al [Li 07] use formal models to solve the release planning problem based on the precedence dependencies between requirements and the resources/ skills constraints. The authors propose mathematical programming techniques that integrate requirement scheduling into software release planning and provide a requirement selection and on-time-delivery project plan simultaneously. This approach goes beyond the process of prioritization.

16. **Kano** model [Cohn 05], [Quality 93] was created in the 1980's by Professor Noriaki Kano. The Kano Model's main objective is to help teams uncover, classify, and integrate 3 categories of Customer Needs and Attributes into the Products or Services they are developing. The 3 types of needs are classified depending on their ability to create customer satisfaction or cause dissatisfaction. When translated to software development, the three categories are [Rikkilä 10]: i) Must-have, mandatory features. It is necessary to develop these features just to enter the market. However, once the basic requirement is satisfied, customer almost doesn't care about the further improvement of the feature; ii) Linear or performance features. The better this kind of features is developed, the more customer is satisfied; iii) Excitement or "wow" features Customer won't be disappointed too much if these features are missing. However, the couple of exciter features can provide great customer satisfaction often adding a price premium to a product.

17. **Relative weighting** [Cohn 05] is prioritization approach that considers both the positive benefit of the presence of a feature and the negative impact of its absence. Each requirement considered for the next release is assessed in terms of the benefits it will bring if implemented and the penalty that will be incurred if it is not implemented. The benefit and penalty for each theme or epic are assigned values from 0 (low) to 9 (high).

18. **Theme Screening/** Theme scoring [Cohn 05] is a prioritization technique that can be used to prioritize themes (groups of user stories) and epics (large user stories) against one another. It can also be used to prioritize projects or products against one another. At the beginning, the selection criteria on which the priorities will be based, have to be

identified. Next, each of the selection criteria is weighted by assigning values. The criteria can be weighted by assigning values that add to 1.00 or by assigning any relative values such as 1, 2, 5, and 3 for a set of four selection criteria.

19. **Planning Poker** [Cohn 05] is a consensus-based technique for estimating, mostly used to estimate effort or relative size of tasks in XP. It is used for estimating a set of requirements – typically User Stories, by using estimation cards. Each participant obtains a deck with cards that have numbers on them. A typical deck has cards showing the Fibonacci sequence including a zero: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89; or other similar progressions.

20. **Binary Priority List (BPL)** [Ahl 05] is a binary search tree based technique for prioritizing requirements. The technique can be successfully used to prioritize requirements and is especially suitable for a medium amount of low-level requirements. According to [Ahl 05] BPL does not make the underlying inputs for the prioritization explicit. The results are rather based on the spontaneous intuition of the person applying it.

21. Daniel Port and Tung Bui [Port 09] propose a mixed agile and plan-based requirements prioritization strategies, that they name **hybrid approach.** The authors perform a simulation and reach the conclusion that the mixed methods of AG2 and HY outperform non-mixed methods PB and AG. However, it is not clear that simply any mixed strategy will be effective. Furthermore, the authors propose a seven-step model to look for an appropriate balanced and effective strategy. They have derived two mixed strategies – AG2 that takes a majority agile approach and adds a bit of Plan Based (PB) by including cost assessment and Pareto ordering in the prioritization, and Hybrid (HY) that takes primarily a PB approach but adds frequent (but heavily guided) reprioritization. The HY strategy appears to be the overall best performer, but the price for this approach is greatly increased prioritization effort as it includes the major activities of both the PB and agile strategies.

22. In addition to the techniques described above, our literature review revealed one practice, which can not be treated as separate method or technique, as it could be applied in combination with any other technique - the practice of bucketing requirements [Patton 05]. This means ”bucketing” groups of major functionality or areas of task support is sometimes easier than feature by feature prioritization.

The analysis of the prioritization methods in the literature, as summarized in this section, indicates that most of the methods are informal and subjective. We observe directions

like: 'If the requirement is less important than the root node, compare it to the left child node', or 'participants assign each requirement a number on a scale of 1 to 5 to indicate the importance of those requirements'. These are indications of the assumption that those, involved in the prioritization process, know the importance of each requirement.

As Harris&Kohn [Harris 06] argue, there is a number of problems related to these approaches. From the perspective of our study, the main deficiency of the methods in general is related to the insufficient support they provide for the client. This conflicts with one of the basic assumptions of the agile approach – namely, that the client is responsible for the prioritization decisions. Considering that this is a complex and responsible task, we would expect that there should be methods that would guide the decision-makers during the whole prioritization and re-prioritization process. However, the detailed look at the literature, as reported above in section 3.1., shows that this is not the case. Although there is a significant number of prioritization methods, most of them provide procedural guidance only, that is – how to obtain a prioritized list with requirements from an initial list. For example, "compare Requirement A with Requirement B and put the higher-priority requirement at the top of the list".

At the same time, a number of assumptions have been made implicitly by the authors of the method. Those are:

1) The decision-makers have all necessary information that is needed to perform the prioritization;

2) The decision-makers have the skills, knowledge and ability to perform the prioritization.

However, there is no evidence that these assumptions hold in all cases. A discussion on the context where the assumptions are valid, is missing in the literature.

As already said at the beginning of this section, we use the knowledge that we synthesize in this section twofold:

- To capture the current state of the practice as documented in literature and to identify problems related to it. We present this analysis in section 4 of this chapter.

- To identify the gap between the guidance that the literature sources provide with respect to requirements prioritization, and the prioritization practice as we observed it in a case study.

3.2 Value Based Requirements Engineering

The topic of value creation receives increasing attention in the research community [Rikkilä 10]. In SE, the sub-field of Value-based Software Engineering which focuses on the value analysis and value creation process in a software projects, has been gaining in importance [Boehm 81]. Drawing on the value-based SE theories, a Value-based approach in requirements engineering has been proposed. As B. Boehm [Boehm 12] puts it: "Value-based requirements engineering includes principles and practices for identifying a system's success-critical stakeholders; eliciting their value propositions with respect to the system; and reconciling these value propositions into a mutually satisfactory set of objectives for the system."

Aurum and Wohlin [Biffl] advocate a value-based RE approach that in its essence is about aligning clients' requirements, business requirements and technological opportunities when making requirements prioritization decisions. For example, recent studies by Barney et al. [Barney 09] investigated the release-planning process to create software product value through requirements selection. These authors identified the factors that determine the decisions about inclusion of certain requirements for implementation. Next, Rönkkö et al. [Rönkkö09] present three aspects of software – as a technology, as a design, and as an artifact. They use these aspects to divide the value concept into three components that are relevant for software developing companies and their clients: intrinsic value, externalities and option value. The authors propose a value-decomposition matrix as a vehicle to reason about the various aspects of value. We make the note however, that they take a broader look at the development of software products, without discussing a specific development method. As these authors stress, the vagueness of the concept of value seems to be a central problem in the Value-based requirements engineering.

These previous studies lead us to the conclusion that focusing on the topic of value creation is important, at the same time we observe that the studies do not differentiate between the concrete Requirements Engineering techniques used, and do not deal with the question how a particular technique contributes to the value creation.

3.3 Methods for decision-making under uncertainty

Erdogmus [Erdogmus 05-1] demonstrates that agile methods are especially appropriate for projects with high uncertainty. This, in turn, implies applying different decision-making approaches that take into account uncertainty. In this section we outline some of the existing methods in agile context.

There exist already a number of studies on decision-making under uncertainty in the domain of information technologies. Some authors address the problem of selecting the the best among a set of alternative IT investments, they propose the use of approaches like Utility Theory [Anand 93] or Balanced Scorecard [Scorecard 96].

Robustness analysis is an approach for structuring of problem situations in which uncertainty is high, and where decisions can or must be staged sequentially. The specific focus of robustness analysis is on how the distinction between decisions and plans can be exploited to maintain flexibility [Rosenhead 01].

From a more general perspective, uncertainty has been dealt with by applying fuzzy logic to model uncertainty, by considering intervals for the values of a given criterion and randomness in the actual value of the criterion by means of a probability distribution, which is needed to calculate the expected utility for the criterion values.

The topic of uncertainty in agile projects in particular has been addressed by other researchers. One category of papers deal with uncertainty of the development process itself, seen from technical perspective. An example of this kind of uncertainty is handling development time uncertainty in agile release planning [Logue 08] and dealing with uncertainty in development time and business value [Logue 1 08]. However, these papers consider the developers' perspective and are not directly related to the decisions that the clients have to make during a project.

Relevant for our topic of investigation is the class of papers, looking at the uncertainty of the projects from economic and value perspective. Other authors [Erdogmus 02] recognize that there are certain parallels between the Real-Options Analysis and the decision-making in agile projects. We make the note that in Chapter 5 we build upon this observation and extend it by analyzing the possible options that clients face in agile projects.

4 Research Goal

Section 3 suggests that focusing on the topic of value creation is important, at the same time to the best of our knowledge we could not find sources dealing with the question of the value creation particularly in agile context. Furthermore, the analysis of the existing prioritization methods showed that most of the methods are informal and subjective.

With our research we address these central questions by investigating what are the concepts that are considered during the decision-making, who are the decision-makers, and understanding how various context factors impact the decision-making. Thereby we put a significant emphasis on the topic of Business Value. This is so because another basic requirement for successful application of an agile approach is that at the beginning of an iteration the Business Value (BV) of each feature has to be estimated (calculated). The challenge is to make the knowledge or information, used by the experts to perform the estimations, explicit. The sources of such information need to be identified, as well as the criteria that define one requirement as more valuable than another. However, the assumption that the BV of a feature can be determined, doesn't seem to be fulfilled in majority of cases.

The purpose of our research effort is to make the decision making process more explicit, objective and systematic, and to increase the awareness about the issues and challenges down the road. In particular, we want to help decision-makers to become aware of the concepts and the relations between them that impact the prioritization decisions. Our results can be used independently to the concrete prioritization method.

Chapter 3

Understanding the concept of Business Value in agile context[1]

This chapter elucidates the concept of Business Value and the central role it plays in agile development. We present insights and details about the two empirical studies that we conducted – one systematic review of literature and one case study. Building upon our findings, we discuss which agile practices contribute to business value creation.

[1] This chapter is based on the following papers:

Racheva, Z. and Daneva, M. and Sikkel, K.(2009)Value Creation by Agile Projects: Methodology or Mystery? In: 10th International Conference on Product-Focused Software Process Improvement (FROFES 2009), 15-17 Jun 2009, Oulu, Finland. pp. 141-155.Lecture Notes in Business Information Processing 32. Springer Verlag. ISBN 978-3-642-02151-0 *and*

Racheva, Z. and Daneva, M. and Sikkel, K. and Buglione, L.(2010)Business Value Is not only Dollars - Results from Case Study Research on Agile Software Projects. In: Product-Focused Software Process Improvement, 11th International Conference, PROFES 2010, June 21-23, 2010, Limerick, Ireland. pp. 131-145.Lecture Notes in Business Information Processing 6156. Springer Verlag. ISBN 978-3-642-13791-4

1 Background - Business Value as a central concept in agile projects

Leading agile practitioners claim that the main purpose of an agile project is to deliver maximum business value to the client, and to do so fast and early in the project [Agile] [Boehm 86]. In other words, business value is what motivates the adoption of agile practices in the first place. Still, little is known about the nature of this concept and the process that leads to its creation and accumulation throughout a project.

The term Business Value (BV)is being used in management and financial economics as an informal term that includes all forms of value that determine the health and well-being of the firm in the long-run. In the context of agile development the term Business Value appears in the majority of publications at agile software development conferences (for example, the annual AGILE conference series) and in agile practitioners' fora (for example the online Agile Journal [AJ]). Typically, the term is used in phrases like 'companies should focus on delivering business value', or 'agile methods help deliver business value'.

That BV is central to the agile community is not surprising. What is surprising, however, is that there seem to be two contradictory views on the concept of business value. On one side, practitioners are occupied with how to measure the creation of business value through the software development process by translating anything valuable into dollar value. On the other side, in studies of the business value of IT, business value is modeled as a multidimensional concept. Requirements prioritization too assumes this, and have as their major aim to map value from these different dimensions to one dimension of prioritization. However, this is not acknowledged in the literature on business value in the agile community.

These observations motivated us to look deeper and in a more structured way at the agile literature and practice and get to know what is the definition of business value that is particular to the agile context and in which particular ways agile practices contribute to the value creation process. Our goal is to uncover such knowledge, to identify the different viewpoints presented in current agile software engineering literature and to uncover conceptual categories that are significant in developing a deeper understanding of the phenomenon of creating business value in agile software projects.

Following the discussion above, in this chapter the discussion is narrowed down with respect to the following: (i) the concept of Business Value, and (ii) the practices that lead

to value creation in agile context. In order to uncover the current understanding behind this commonly used term, we followed a two-phased approach.

During the first phase, we conducted a systematic literature review. The second phase consisted in a case study performed in agile software development organizations.

We make the note that we don't aim at providing new definition of BV. Rather, our investigation was driven by the intent to understand what is behind the buzz-word Business Value as it is used in agile context, as it is used remarkably often in agile literature. This is the reason why we don't consider or investigate synonyms of the term business value such as, for example, 'customer value'.

2 Business Value in the agile literature

The background provided above in Section 1 explains the choice of research questions for the series of studies.

The literature study, set out to answer two research questions (RQ):

RQ 1.1: What concepts of business value are used in agile context, as described in the agile literature?

RQ 1.2: In which way do agile projects create business value, according to the published agile literature?

To answer our research questions, we have performed a systematic review [Kitchenham 04] of literature.

2.1 Related work on systematic reviews in agile context

The literature study was informed by previously published systematic reviews [Abrahamsson 02], [Dyba 08], [Abrahamsson 03] in the domain of agile phenomena. Although these systematic reviews deal with agile phenomena, the research questions asked in them are different from ours. The earliest review [Abrahamsson 03] dates from 2002 and answers the question "What makes a development method an agile one?" This systematic review synthesizes existing literature to characterize the state-of-the-art practice and compare agile methods by pinpointing out their similarities and differences. Furthermore, a comparative analysis of nine agile methods was published in a report in

2002 [Abrahamsson 02]. We make the note that these two publications [Abrahamsson 02],[Abrahamsson 03] found scarce empirical support to exist for the nine reviewed methods.

The second systematic review [Dyba 08] dates from 2008 and its objective was to answer the questions "What is currently known about the benefits and limitations of the Agile Software Development (ASD)?" and "What is the strength of the evidence in support of these findings?" These authors also investigated what the implication of ASD studies are for the software practitioners and software engineering researchers. This systematic review identified four categories of ASD publications: (i) those pertaining to ASD adoption, (ii) to human and social factors, (iii) to customer and developer's perceptions and (iv) comparative studies of ASD processes and alternative ones. With respect to each category, the systematic review [Dyba 08] indicated a number of reported benefits and limitations of agile development. A key finding of this systematic review was that "the strength of evidence about benefits and limitations of agile software development is very low, which makes it difficult to offer a specific advice to industry. The authors concluded that the research community "needs to increase both the number and the quality of studies on ASD".

Clearly, the research questions of our SR were not the objectives of the previously published reviews. In this sense, the present study complements the existing research by other SR authors.

2.2 Research method

As Kitchenham [Kitchenham 04] puts it: "A systematic review is a means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest. Systematic reviews aim to present a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology."

We performed the systematic review by following the three phases of a systematic review covered in the guidelines of Kitchenham [Kitchenham 04]. Those phases are: (1) planning the review, (2) conducting the review, and (3) reporting the review. Below we address them and explain how we implemented them in our study.

Our systematic reviews started by defining a review protocol that specifies the research question being addressed and the methods that will be used to perform the review; they are based on a defined search strategy that aims to detect as much of the relevant

literature as possible. The search strategy should be documented so that readers can access its rigor and completeness.

Next, explicit inclusion and exclusion criteria have been defined to assess each potential primary study. The information to be obtained from each primary study including quality criteria by which to evaluate each primary study are specified as well.

As per SR the guidelines [Kitchenham 04], we used the RQs for determining the content and structure of the SR, for designing strategies for locating and selecting primary sources, for critically evaluating the sources, and for analyzing the results. We implemented the following SR process:

The following search string was used: *business value* AND one of the following: *agile, iterative*, *scrum*, *XP*. That is, the string was: ("business value" AND (agile OR iterative OR scrum OR XP)). We used these words because Scrum and XP are the widest-spread and best-known agile methodologies and often studies have been performed or papers written on the specifics of these methods in particular. This way we wanted to make sure that even if the title or the abstract do not contain the words 'agile' or 'iterative', we still will retrieve relevant sources that discuss the concrete agile methods Scrum or XP. This approach helped us to cover synonyms used in literature and the variety of agile software development and agile project management concepts.

We want to underline that we performed searches with the following alternative strings: *feature driven development* AND *business value*; *crystal clear* AND *business value*, because *crystal clear* and *feature-driven development* are also among the popular agile methods. However, they didn't return any relevant papers. We considered it important to proceed like this because no standardized, consistent terminology exists with respect to the topic of our study.

Our search strategy included the electronic databases Scopus, Web of Knowledge and Science Direct. This way we ensured that our search was applied to journals, magazines, conference/workshop proceedings, as Scopus covers publications by ACM Digital Library, IEEE Xplore, Elsevier, SpringerLink, and Dr.Dobb's which is one of the most popular agile practitioners online journals. We were looking for materials published since 2000, as this is the beginning of the spreading of the agile methodologies.

As the topic of business value in agile software development is closely related to the practice, we decided our search strategy to include the Agile Journal (www.agilejournal.com) which is the most popular practitioner-centric online publication

venue of the agile community. The Agile Journal publishes monthly issues with articles on various subjects concerning ASD and APM.

The Systematic literature review has been executed by the author with the participation of other two researchers. The author composed the search strings, defined the inclusion and exclusion criteria, identified the literature sources and screened all abstracts. Next, she read those primary studies that seemed to meet the inclusion criteria. The daily supervisor (Daneva) analyzed independently the abstracts and the results were compared. Differences were cleared in a discussion with the participation of another senior researcher (Sikkel).

Below we describe the inclusion and exclusion criteria that we applied in the study selection process, and the selection process itself.

2.3 Selection of the sources

To select papers from the retrieved results, the following inclusion criteria were used:

1) The paper has agile software development or APM as its central topic, that is – the authors discuss aspects of the development process, and this development process is agile.
2) The paper discusses the authors experience on what is BV or how various practices (one or combination of practices) contribute to the value creation;
3) The claims are clearly described and can be related to real-life.

We used these **exclusion** criteria:

1) If two papers publish the same framework, one of them is excluded.
2) The paper is not on IT application development, for example "An agile classroom experience" [Reed 08], [Rao 07].
3) Papers that are dedicated to studying the development of specific types of applications – e.g. ERP, and the word 'agile' appears only in phrases like 'in agile manner', have not been included in the study, that is, the paper is not on agile software development, for example "Transforming air traffic management system" [Davis 07].

Additionally we excluded any paper that is not accessible, as well as editorials, prefaces, posters, summaries of articles, and tutorials, workshops, and panels, as well as our own papers. We also did not include PhD theses and technical reports. The published sources

we reviewed were written in English only and included both qualitative and quantitative research, from scientists and practitioners.

From Scopus the result included 72 initial sources. Web of Knowledge returned in total 37 papers. The majority of them overlapped with the ones we already identified in Scopus. ScienceDirect returned 460 papers or book chapters. We sorted them by relevance and only the first 50 returns met our first or second inclusion criteria.
In the Agile Journal, the only search string we used was "business value", as we assumed that the publications in this forum would be relevant to the agile software development topic. The result was 56 articles. We acknowledge that these publications are authored by practitioners and are published in a non-scientific forum. For this reason we regard them as an anecdotal evidence. However, we found it important and justified to include them in our study for the following reasons: (i) The phenomenon of agile SD is relatively new; and (ii) We wanted to create as complete picture of the understanding of BV as possible. In the area of agile SD the practitioners are the pioneers that write books and disseminate the agile principles and practices.

We treated the selection of studies as a pre-review process in which a particular paper is screened and judged based on the inclusion and exclusion criteria. This process comprises four iterations, as shown in Figure 7. After identifying the potential sources, we have screened all titles and abstracts to extract the ones we consider relevant to our research effort. Only papers that met the inclusion criteria were used for the analysis in the next, categorization stage. To assure high internal validity of the study we followed at this initial step the recommendations of the methodologists [Kitchenham 04] by screening the sources by two researchers.
During the first phase, all 72 sources from Scopus were screened for inclusion/ exclusion by the author of the thesis and by another, senior researcher. We make the note that in the first phase we excluded only those papers for which it was possible to judge by the information provided in the title and in the abstract, if the paper is on agile SD and about IT development. This means that we didn't expect to find explicit definitions of BV or descriptions of agile practices straight in the abstract of the papers, instead we were looking for hints if the paper is related to the topic of our investigation, that is – does it seem to be relevant for the study.
The results of the first phase were compared and in case of discrepancies between the researchers' opinion the paper were classified as *undetermined* and was further analysed at the second phase. Next, the two researchers reviewed the papers classified as undetermined. For all other papers the full body of the paper was analysed. At this step we excluded those papers that we could not access. After the first two phases 37 papers

were included for analysis at phase three. We downloaded the body of the papers and searched manually for pieces of text indicating what the authors call value, or indications about value-creating practices. The author of the thesis analysed all 37 papers. The second researcher analysed 12 randomly selected papers out of these 37. The results were compared and there was a difference in the opinion in two cases. After a discussion on the application of the inclusion/ exclusion criteria, a consensus among the researchers has been reached.

We proceeded in a similar way with the 56 hits from Agile journal. As the majority of these publications don't have abstract, introduction or conclusion sections, all 56 materials were read by one researcher (the author of the thesis) in order to assess their relevance with respect to our study. Many of those materials turned out to be blog entries, news, or editorials. The number of papers that remained for analysis was 23.

To illustrate our selection process we take as an example paper [Balikuddembe 09]. It is an example of a paper that we didn't include as it does not satisfy our first inclusion criteria – namely, the paper is not explicitly on agile SD. Instead, this work draws from available literature in value-based software engineering, to provide an integrated approach for assessing overall value of a software project. The authors perceive value as the benefit realized in terms of profit or long-term business benefit that the software vendor derives from a project, given the project's overheads.

The final step of the sources selection resulted in identifying those papers, that are relevant to the questions of understanding BV, and the practices, contributing to value creation.

In section 2.3 below, we present our results pertaining to these questions and offer a discussion on them. We, then form answers to our research questions in Section 2.5.

We make the note that the first version of this systematic review was performed in 2008 and has been published in 2009 [Racheva PR 09]. In the section below we build upon this paper and complement the results with the findings that we obtained during the second version performed in 2011.

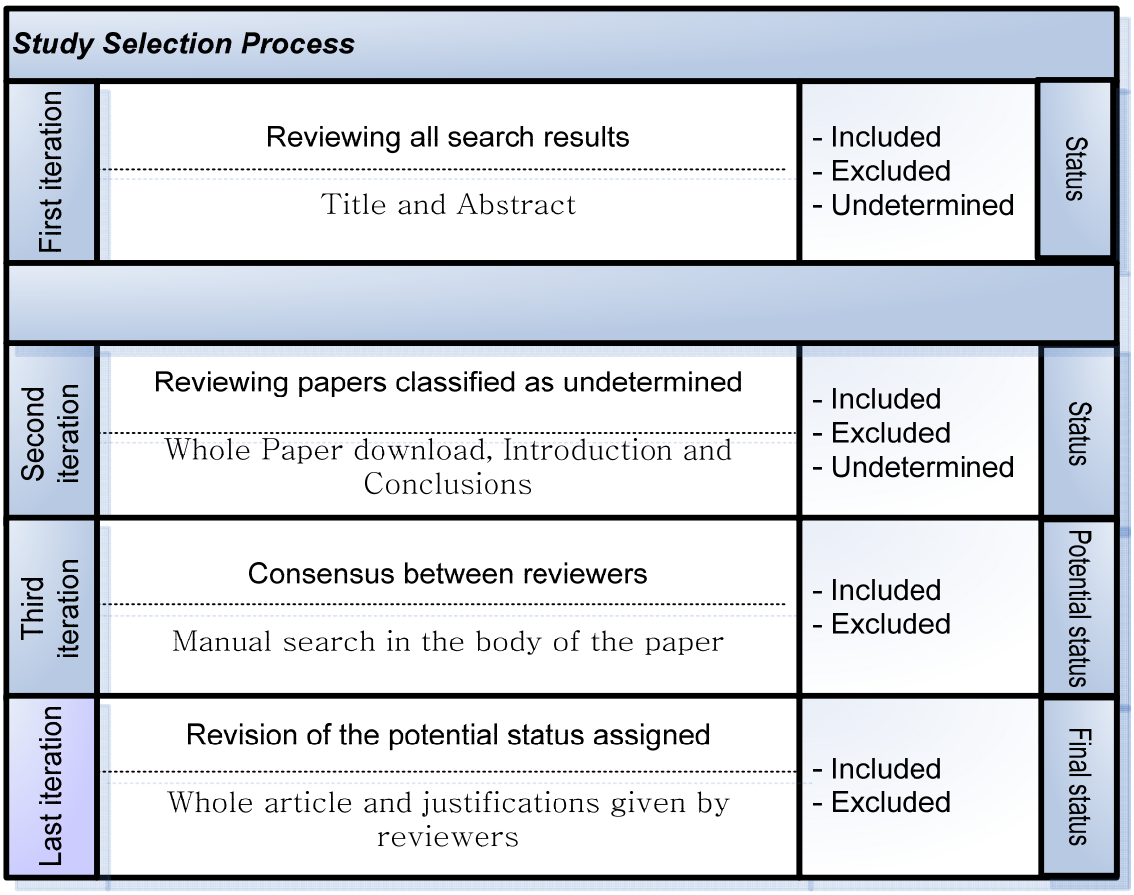

Fig 7: Study selection process

2.4 Results

Our first observation from reviewing the papers is that about two-thirds of them turned out to be irrelevant according to our inclusion criteria described above. Although a large number of materials in fact did contain the terms 'business value' and 'agile', we found that the meaning of 'business value' itself was not elaborated in either of the senses defined in the criteria.

Our SR indicates that the authors of the papers we reviewed consider business value a self-evident concept. It seems that business value concepts reflect condensed meanings of general terms which the authors of the papers assume everyone shares.

We found only eight papers that provide an explicit definition of business value in agile context. With exception of these nine papers [Barnett 07], [Patton 08], [Pettit 07], [Rawsthorne], [Rawsthorne 06], [Poole 07], [Logue 08], [Hartmann 06], [Szoke 11] in the

literature we reviewed, the understanding of business value was either implicit, or taken for granted.

For instance, paper [Lehto 09] does not include an explicit definition, neither a discussion on the nature of BV. However, implicitly the authors consider (i) business value to be expressed and operationalized trough business goals, and (ii) structure the discussion around release goals and iteration goals. In [Hashmi 07] the authors discuss the value creating practices of XP, such as the *benefit realization analysis*. There is no definition of BV, but implicitly the authors consider quality of the product and productivity as value creating. We discuss this source later with respect to RQ 1.2. and the value creating practices.

In what follows, we first discuss the definitions we catalogued from our review, and then we compare them to distil some characteristic features of the understanding of business value in the agile literature. Last, we present the results of our application of a coding process on the reading materials we deemed relevant. These results are conceptual categories which we think help understand and reason about the business value concept in agile project context.

2.4.1. Classification of the results

We found three classes of results with respect to the definition of business value: (i) definitions of the term BV; (ii) papers addressing the calculation of BV, and (iii) papers using synonyms of the term BV. We present these three sets of findings in the sub-sections below:

First, we identified d**efinitions of business value.** They are presented below in Table 4.

Table 4: Definitions and sources.

Authors	Definitions
Barnett [Barnett 07]	"…business value, as measured in business revenue, stock price, market share, or other business metrics. Value is in the eyes of the customer…"
Patton [Patton 08]	"Business value is something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service".
Pettit [Pettit 07]	"Business value is a communication vehicle: we use business value

	to communicate values, priorities, motivation".
Rawsthorne [Rawsthorne 06]	"Business value is what management is willing to pay for; value can only be defined by the ultimate customer. And it's only meaningful when expressed in terms of a specific product (a good or a service, and often both at once), which meets the customer's needs at a specific price at a specific time".
Poole [Poole 07]	"Might not be possible to define the business value of IT independently of other activities. What is business value: Business value = F(x) + F(y) + F(z) + ... That is, a complex function where we must balance multiple things ...while they are changing!"
Rico [Rico 09]	"Business value is measured in terms of costs, benefits, breakeven point, benefit to cost ratio, return on investment, net present value and real options."
Qumer [Qumer 07]	"Business Value is the return on investment"
Hartmann [Hartmann 06]	"Business value is best determined by business stakeholders and developers together, at the level of specific project components - features or groups of features that can be given a specific cost and customer value. Value is defined as software put into production that can return an investment over time. There are a number of ways to measure value. These include Net Present Value (NPV), Internal Rate of Return (IRR), and Return on Investment (ROI)."
Szoke [Szoke 11]	The business value term is being used in management and economics that includes all forms of value for the company in the long-run. It usually means anything that can be translated to money such as revenue, market share, and stock price – it is generally a multidimensional concept.

When looking at the definition we make the interesting observation that all definitions with exception of [Rico 09] originate from practitioners' articles. We explain this with the facts that (i) we could not find scientific publications, particularly dedicated to explaining the notion of value in agile context, and (ii) we believe that the authors assume that the concept of business value is self-evident because it is extensively studied in economic sciences.

Second, we list below two other papers that refer to a calculation of BV, without providing a definition:
1) Frank Padberg and Matthias Müller [[Padberg 04]] perform a study on how large is the impact of the lower pair productivity during warm up on the business value of the pair programming project. In the paper, they provide the following understanding about BV:
"A project has business value only if its net present value is positive. That is,
$NPV = AssetValue/(1 + DiscountRate)^{DevTime} - DevCost.$
With net present value, the cash returns of a project (AssetValue) are discounted at a certain rate per year, the DiscountRate." That is, these authors define BV trough Net Present Value.

2) A. A. Adamopoulos [Adamopoulos 10] proposes the use of a Business Value Model™ that should show the strategy of a project or organisation. In his opinion: "The premise is to identify the business value drivers and then define them. To define them creates a method to measure whether goals are attained. Each measurement is a possible business value driver that will drive the project in the right direction to maximize business value generated. From here we build a Business Value Model from business value drivers and define the relationships between business value drivers and how we'll use them to steer the project." However, he does not provide a definition of BV or BV drivers, as he calls them.

Third, in addition to the above definitions, we identified eight other publications [Alleman 03], [Cabri 06] , [Favaro 05], [Muller 03], [Pinheiro 08], [Rawsthorne06], [Sulaiman 06], [Concha 07] which discuss the topic of business value without using explicitly the term "business value" itself but terms synonymous to it. We list these seven for the sake of completeness:

1) three papers [Cabri 06], [Favaro 05], [Sulaiman 06] use the concept *Earned Value* in agile settings. All three base this concept on the earned value measure used in economic sciences, in order to track progress or velocity of an agile project. According to [Cabri 06], [Favaro 05], [Sulaiman 06], Earned Value is a project management technique that

compares, at a specific date, the project delivery percentage against budget used up percentage, thus estimating progress and performance of a project against the plan.

2) one paper [Pinheiro 08] uses the term *perceived business value*. According to the authors, this concept means the particular context of multiple projects and optimizing value in this case.

3) one paper [Rawsthorne 06] proposes the concept of *Earned business value* (EBV). It defines a measure that can be used to track the value of the requirements being delivered. The measure helps calculating the relative value of the work done compared to the whole project. Agile earned business value is a ration calculated by using the formula:

EBV = the-percent-of-value-delivered / the-percent-of-cost-consumed.

4) two published sources [Favaro 05], [Muller 03], use the term *Economic value* interchangeably with business value. The second source [22] defines the Economic Value through the net present value (NPV) in the formula:

NPV = AssetValue / (1 + DiscountRate) DevTime – DevCost

We note that the term 'Asset Value' (meaning the dollar returns of a project) is neither defined, nor traced back to tangible project characteristics. Instead, it is taken as a given in the calculations.

The term *business value goals* is used by the authors of [Concha 07] to achieve continuous risk visibility during a project : "At the start of the project, all business value goals (functionality, time-to-market, budget, and quality) must be established in terms of qualitative metrics, as well as potential losses incurred if a business value goal is not met."

2.4.2. Comparison of the concepts

Our comparison of the definitions presented in the previous section was done by following the guidelines of J. Webster [Webster 02]. We applied the following steps: we first identified the original authors' terms used in discussing business value and then, we compared them to see points of convergence and divergence to characterize these. In our comparison, we also checked for each definition the context of its intended use. Here we present a summary in a concept-centric approach. A conceptual category explicates ideas, event, or processes in our observations, which we collected while running the SR. We call these categories 'significant' because we believe we can use them to make an interpretative rendering that illuminates the studied phenomenon, namely *business value* in agile projects.

Below is a list of all identified concepts:

Qualitative definition of BV
Measuring BV (Quantitative definition)
Perspective of value
Level of granularity of BV
Vision
Goal
Subjective character of BV

We identify three different aspects that can help us better understand the phenomenon:
First, we observe that some authors regard BV as qualitative concept, while others are concerned with how to measure it and express it in quantitative terms.
Second, the BV is in the eyes of the stakeholders, that is - the BV is subjective, and is linked to the perspective of the stakeholders.

Third, BV is related to the stakeholders' goals that they want to achieve with the product, and with their vision.

The analysis produced these characterizing features of the business value below:

- **Business value in practice is considered both in qualitative and quantitative way**

Our observations from the reviewed sources do indicate that there are both qualitative and quantitative definitions of business value. Almost the half of the definitions define BV trough an economic (monetary) values such as NPV or ROI. However, we found no study suggesting that a quantitative definition of business value is used when authors attempt to see how much value is contributed by the deployment of an individual agile practice or by a group of practices. We could also find no study that provides evidence that business value and its accumulation over time has been tracked quantitatively throughout the project iterations. Clearly, if one is to see how agile development creates business value, one needs "to tie value back to some tangible gain for the business" [Patton 08]. For example, to "something that delivers profit to the organization paying for the software in the form of an increase in revenue, an avoidance of costs, or an improvement in service" [Patton 08]. However, our review indicates that tying back business value to gain is problematic.

This observation aligns with Pettit's opinion: "In an effort to justify IT investments, we have increasingly looked to quantify the "business value" of projects. While there is merit

in doing this, in practice it can generate more heat than light. There is a limit to the accuracy with which we can predict the future, how much data we can collect, and the extent to which business decisions can be expressed as mathematical formulas."

- **Business value tends to be subjective**

Our observations from the literature sources indicate that often, the term "value" is used subjectively. This can be observed in the definitions of Barnett, Rawsthorne, Pettit and Hartmann. Patton [Patton 08] illustrates clearly this by his experiences witnessing agile project stakeholders expressing value in the following ways: "I value something if it makes me feel good", or "If I'm representing the business, then I might view something that makes me feel good as a "business value".

In those cases where value is defined as a monetary measure, the authors assume that the result will be objective, based on data and calculations. However, when we look closer at those definitions we see that they require as an input for the calculations subjective estimations, for instance of value of an asset. Moreover, business value estimation requires a degree of trust: "There is a limit to the confidence we can place in business value numbers. There are time and resource limits to estimating, collecting, and analyzing results. In addition, estimates and analyses aren't facts. We can forecast revenue, but we must realize that forecasts may not materialize." [Pettit 07]. This means that business value is not an absolute "dollar value", as it is related to unpredictable and difficult to estimate parameters such as revenue, future market share, or market conditions.

- **Perspectives to consider when thinking of business value**

We identified that the understanding of business value is traced back to the perspectives of the two key groups of participants in the agile project and, in turn, their roles. As could be seen from the conceptual map, two main groups of papers emerged: (i) those dealing with creation of business value for the client organization and (ii) those discussing how a development organization can manage a portfolio of multiple and concurrent agile projects that are being undertaken for one or more client organizations [Rawsthorne 06]. The two groups of papers clearly indicated that each perspective represented a unique understanding of what is of value and how to achieve maximum value.

From the client's perspective, the value is defined by the clients themselves. Indeed, most of the literature sources we studied relate to business value as understood from the client's perspective.

Furthermore, from the perspective of an agile software development organization – that is, the software vendor, the management defines the relative business value of each

project in the portfolio of projects, which the organization is engaged in, as a software supplier. The management team typically uses projects' business value in the process of performing trade-off analysis and balance between resource demands coming from different projects. We make the note that in addition to the above, in case of a development team in a client-supplier contractual relationship, the value for the team is to satisfy the client's needs, so that the client will eventually come back the next time, which has a direct impact on the revenue of the developer [Logue 08]. This is different in the case of an IT-department within a company, where the IT-team has (i) to make business management happy, (ii) to help increase overall profit of the organization, and (iii) to balance between new development and other IT operations and maintenance tasks.

We make the note that we have consciously excluded the role of the end user. This is because, in the literature sources we reviewed, we could not find any evidence suggesting a linkage between the end user and the decisions influencing business value in agile projects. We believe that this is so because authors silently assume that the "customer" will take into consideration what is valuable for the end users in the client organization. Still, we think that this question is worth to be explored in detail in a future work, as it is very relevant for the value perspective.

It is interesting to note that more often than not, when agile projects refer to "customer", they mean a multi-stakeholder setting in a client organization. In such a setting, if requirements are prioritized and re-prioritized from the perspective of the "customer" at inter-iteration time, then the relative priority is given to each stakeholder group behind the label "customer" is the actual driver for the prioritization process. Patton [Patton 08] illustrates this point drawing on the matter that "different people consider different things valuable" and that "prioritizing work becomes a tug-of-war in those circumstances". (Patton [Patton 08] warns that "If we share a common idea of what's valuable, then we needn't pull in opposite directions.") Although this observation is not directly related to the nature of BV, we present it at this point as an important bridge to the topic of agile requirements prioritization that we investigate later in this work. [Ktata 09] argues that as business value is subjective, different stakeholders can perceive different things as valuable from their own perspective. Without a common understanding of business valuation in front of the strategy pursued, it's hard to find consensus on prioritizing sources of value. This finding is essential for our work as it motivated our next step – namely a case study that investigated the agile requirements prioritization process as it happens in commercial cases.

- **BV is related to the stakeholders' goals and vision.**

When trying to understand business value in agile projects we identified four conceptual categories which we deem significant when it comes to understanding the origin of business value and the process that leads to its creation. The categories we discerned are these:

1. *Vision*. Multiple indications [Barnett 07], [Alleman 03], [Favaro 05], [Gatz 07]] from literature suggest the creation of business value should be driven by the vision of the organization.

2. *Business goals*. Approximately half of the papers suggest that business value must be established from business deliverables often requiring input from a range of stakeholders [[Gatz 07], [Logue 08].

3. *Product goals*. The majority of the agile practitioners relate business value to software product goals. For example, [Cabri 06], [Gatz 07] cite experiences in which product goals were re-defined after the effect of the IT solution is known. Re-definition of business objectives after change in the project context is also possible [Gatz 07]. The authors mean, for example, change in the business environment, lows, competition. Each of these events might trigger a change in the business goals and consequently – in the defined objectives for the software.

4. *Product features*. Practitioners indicate that it would be of benefit if there is a way to quantitatively assess the business value of each feature of the software product. As Poole says [Poole 07] only by assigning business value, in hard currency, to each IT deliverable and even every feature of a deliverable, can business truly manage the relationship with IT effectively. The authors are concerned with the question how to measure the part of the whole business value (at project level) which is included (encapsulated) in each feature. For example, Rawsthorne [Rawsthorne 06] assigns to the whole set of features the value of 100%, and each separate feature is treated as a fragment of the whole functionality and, in turn, is measured in its relation to the whole. However, we found no study that describes a project in which this was done. Moreover, Logue [Logue 1 08] warns that: "The business value of stories can be particularly problematic for development teams to estimate, due to the high variation in possible monetary return."

Last, we want to stress that the business value of an IT solution tends to be dependent on non-IT business processes. Our observations from the reviewed publications suggest that business value might well be related to other aspects and processes of the business. The definitions of Patton, Rawsthorne and Hartmann suggest that the understanding of

BV is closely related to the concrete organization and its context. Poole [Poole 07] even warns that it might not be possible to define the business value of IT independently of other activities.
In the next section we discuss RQ 1.2. – the agile practices that contribute to value creation.

2.5 Agile practices that contribute to business value creation

In this section we address the second RQ of ours SR study – what agile practices contribute to the value creation.

There seems to be a common agreement in the literature that some practices help more the process of creating value than others [Favaro 05] [Gurses 07], Gurses [Gurses 07], highlights the importance of knowing the value that the particular agile practices create.

The authors of [Qumer 08] argue that BV has a great impact on the decision to adopt or discard any agile practice or set of agile practices. The value to a business of using an agile approach can be categorized as a business value delivered through agile software development as well as a business value delivered through appropriate governance.
The authors propose a bridge that aligns business goals and agile software development goals. A cost-benefit analysis metric is suggested to be helpful for the appraisal of business value delivered through working in an agile development environment.
This paper asks the question about the relation between business value and agile development, similarly to [Ktata 09]. In the authors' opinion: "today's concern has switched to business issues such as: What is the tackles the right product? How to identify sources of value? Why building the software? How to reduce waste (unused features)? How to improve decision making? How to track value progress? How to guide the project efficiently to make dependable software? an intensive-customer involvement Answering to the previous questions will ensure responsive software, delay obsolescence, reduce waste, increase added value, and achieve real business goals." However, the authors of these papers don't provide answers to these questions.

The results of the analysis of the sources are presented in Table 5 below.

Table 5: Results of the analysis of the sources

Source	Practice addressed/ described/ discussed
[Rico 09]	Just enough documentation; postponing implementation until last responsible moment
[Rico 1]	Requirements / USs prioritization
[Hansen 09]	Collaboration; small batches of work; continuous integration; global code reviews; automated testing; social aspects
[Ktata 09]	Prioritization
[Vidgen 09]	Co-evolutionary capability: collective mindfulness, team learning; sustainable work with rhythm, process improvement
[Hashmi 07]	Risk identification and mitigation at each iteration; iteration planning; refactoring; continuous feedback
[Concha 07]	Proposes framework for commitment management
[Ton 07]	Size of requirements and US can be linked and adapted to the BV
[Yap 06]	"highest value features first", "don't do anything extra", "eliminate waste" ; Value-based Investment Decisions, High Confidence Stories First, Incremental Story Delivery, Story Ownership
[Hartmann 06]	Using appropriate measurement
[Elssamadisy 06]	Functional testing
[Gottesdiener11]	The agile team

[AJ 11]	Guidance to write high-quality USs and acceptance criteria
[LINDERS]	Agile PI and its benefits
[Kundu]	Value of collaborative culture (of the team)
[Horton]	Technical issues e.g. visual control, managing queues
[Pieczko]	The BV of story boards
[Small]	Concurrent Testing
[Podelko]	Concurrent Testing, performance testing, continuous integration
[Leach]	Business driven development

Our analysis of the practices addressed, discussed and studied in the literature that we summarized in Table 5 can be divided in two groups:

First, directly related to the client perspective and the value creation. Here we find the following sources: [Rico 09] [Rico 1], [Ktata 09], [Concha 07], [Yap 06]. The practices addressed are: prioritization, agile process improvement, just-enough documentation, postponing implementation until last responsible moment; "highest value features first", "don't do anything extra", "eliminate waste"; Value-based Investment Decisions, High Confidence Stories First. This set of papers includes two frameworks created by researchers that aim at improving value creation [Qumer 08], [Concha 07]. However, it's problematic to classify these frameworks as truly agile practices, as they are a research proposition of theoretical character and there is no evidence of their application and benefit for real-life projects.

The second class of papers are concerned with agile practices from development perspective. Those are practices related to the team, the development technology and the development process – global code reviews; collaboration, automated testing, team

learning, refactoring, iteration planning, risk identification and mitigation, cloud technology, story boards, concurrent testing. We make the note that this set of practices include practices that are related to the software development process improvement as such, and are not related exclusively to agile SD. We make the observation that although some of the practices in this second category are of purely technical nature (e.g. refactoring, story board), all practices are oriented towards value creation in terms of quality improvement or waste reduction.

Next to identifying value creation practices, we observed indications about the way practices (or set of practices) added value. That is, what is the effect of the respective practice on the outcome of the project:

1) In [Hansen 09], the authors state that global code review and automated testing lead to reduced time for finding errors and thus – to waste reduction

2) Hashmi in [Hashmi 07]deems that risk identification, iteration planning, refactoring and continuous feedback lead to quality improvement, and

3) Linders [Linders] describes the benefits of agile process improvement, observed by the authors. Those are:

- Being able to deliver the right product with high quality, using frequent feedback
- Understanding the strengths & weaknesses of our processes, and the business value
- Alignment and streamlining of processes between several R&D centers
- Efficient ways for professionals to work together in a dispersed team.

These two sets of papers reflect also two different perspectives with respect to the value-creation practices. We clearly observe two groups of practices - on one side practices that are oriented towards value creation for the client, and on the other – practices that are related to the developers' perspective. This observation is similar to the observation that we made about the perspectives to consider with respect to the understanding about BV. It is interesting to note that all papers of the second category are authored by practitioners. We explain this observation with the pioneers role of the practitioners in the domain of agile SD and these papers are examples of the dissemination of the knowledge and experience in agile context.

We want to stress that although the authors claim that the individual practices contribute to the value creation, we could not find studies that estimate how much value a practice creates and what "more value" means in agile context.

2.6 Limitations

There are three main validity concerns pertinent to our SR: (i) our selection of publications to be included, (ii) our analysis of the data, and (iii) potential bias by the researchers.
As explained in section 2.2. (research method), approximately one-third of the selected papers were reviewed by a second researcher. This way we wanted to ensure that the analysis of the data is correct. For these papers, we consistently observed a consensus. Whenever there was disagreement, the points of disagreements were discussed until both researchers arrived at a consensus.

We believe that the threat to validity due to researchers' bias is minimal, because we have not: (i) published a study which is included as a primary source in the SR or (ii) are in a close research-collaboration relationship with the authors of included sources.

2.7 Summary of the results and implications

This systematic review study has addressed the questions:

RQ 1.1.: What concepts of business value are used in agile context, as described in the agile literature? and

RQ 2.1.: In which way do agile projects create business value, according to the published agile literature?

For RQ 1.1, our findings are that (i) the majority of papers in agile SE literature do not define explicitly the concept of business value in an elaborated way, (ii) the business value concepts rest on the definitions of *Earned Value*, *Net Present Value* or *Return on Investment,* as used in economic sciences, (iii) Practitioners offer definitions which translate business value into dollar value. However, we found that this 'translation' is problematic as there is no study that suggests how this could be done in practice. (iv)The notion of business value is slippery and highly volatile.
Why we couldn't find more profound answer to this RQ? At this stage we can only speculate at this. Our intuition says the fault isn't in the agile methods, but in the very

concept of business value, which turns out to be rather more slippery, subjective and volatile than the most of the authors of studied papers seem to assume implicitly.

For RQ 1.2., we have identified a set of practices that the authors of the publications link with value creation. Furthermore, some of the publications included in our review offer explicit evidence pertaining to the specific ways in which groups of practices create business value throughout the project.

In our view the identification of the practices is an important finding on its own. It motivates further studies on specific practices listed in our table. These results have direct implication for our research. We concentrated our next efforts on the process of requirements prioritization as a major vehicle for creating value for the clients. From the developers perspective value is created by minimizing waste during development, better product quality, and higher customer satisfaction. This result is not surprizing as agile SD is in its essence a process improvement methodology and its goals align with the goals of other process improvement initiatives that lead to higher organizational maturity. We build upon this finding and further study this process with respect to value creation.

Furthermore, our results have implication for the researchers as researchers and practitioners can investigate how exactly the practices create value, and in which context of application. The results of such studies can provide guidance to practitioners on which practices to adopt that would best serve their project needs and environment and eventually lead to better value creation.

When comparing our SR and the earlier published SRs [Abrahamsson 02], [Dyba 08], [Abrahamsson 03] we consider that our findings converge with the earlier published SRs in the following respect: similarly to the other authors, we found that the existing sources of definitions of business value are mainly practitioners' reports. As Abrahamsson et al indicated in 2002 [Abrahamsson 03], back at the time of their SR, the existing evidence consisted primarily of practitioners' success stories.

The results of our SR raise the question whether there is an existing representative body of knowledge on the subject, which might have been uncovered by means of other research approaches. Or is it time that researchers and practitioners look more closely at the phenomenon of value creation? This gives us the incentive to do further empirical research on how people make decisions in agile projects based on the concepts of value. To further investigate the understanding of business value and the value of agile practices, a different type of research is called for. This observation aligns with Dyba [Dyba 08], who reaches the conclusion that SR can be complemented with an empirical

investigation of the studied topic. Following the recommendation in [Easterbrook 08], as a next step of our research we conducted a case study to explore the topic of the understanding of BV and BV creation in real-life agile projects.

3 Investigating the State of the Practice by Means of a Case Study

3.1 Introduction

Building on the study presented in Section 2 of this chapter, we continued our investigation of the concept of BV by the means of a case study. Although we explored the same phenomenon, our research focus slightly shifts as the research approach allows for in-depth look into mechanisms and processes linked with the value creation.

The results of the case study presented in this section pertain to the following two issues: (i) the ways in which agile practitioners perceive the notion of business value, and (ii) the agile development practices that create value, in the practitioners' opinion. That is, we build upon the results of the SR and investigate the same topic with other means.

Below we formulate the RQs for the case study.

As a natural continuation of the SR is the question of how the practitioners that perceive BV, what does this term means to them. This is the repetition of RQ 1.1. from the SR, translated to new setting – a case study that collects data directly from agile practitioners.

RQ 1.3. What concepts of business value do practitioners in the context of agile projects perceive?

The results of the SR confirmed the statements of the agile proponents that the creation of BV for the clients is in the core of the agile methods. Still, it is difficult to judge whether this is an advertisement trick or a there is in fact a systematic approach that justifies this statement. As discussed in section 2.4 - Results, we couldn't find detailed answer to our RQ 1.2. **In which way do agile projects create business value?** As a first step towards studying this question, we address the topic of the agile practices that could contribute to value creation. This motivated the formulation of the following RQ for the case study:

RQ 1.4. In which way do specific or individual practices influence the creation of business value?

When it comes to regarding BV in real projects, it does not become clear from the SR results, how exactly BV is considered. Moreover, we could not find evidence that would show that the practitioners make decisions that are based on value considerations. To elucidate this issue, we posed the following RQ 1.5., that we regard as a refinement of the RQ 1.4.

RQ 1.5. Do practitioner make value-driven decisions during agile development? If so, how decision-making is happening?

One of the observations in the SR is that there are multiple perspectives about the value creation. Agile approaches explicitly talk about creating BV for the clients, however it is naïve and over simplistic to think that there is no link between the BV creation for the clients and the bottom line of the SD organization. For this reason, we posed the following RQ 1.6. for the case study:

RQ 1.6. How do developers combine value creation for their own organization with value creation for the client's organization?

In what follows we present the results of the case study that pertain to the topic of the understanding of BV and BV creation in agile projects. We make the note that the case study covered broader set of questions and in Section 3 we address only the RQs that we posed in this subsection.

3.2 The Research Method

The case study is a qualitative research method that excels at bringing us to an understanding of a complex issue or object and can extend experience or add strength to what is already known through previous research. Case studies emphasize detailed contextual analysis of a limited number of events or conditions and their relationships. Researchers have used the case study research method for many years across a variety of disciplines. Social scientists, in particular, have made wide use of this qualitative research method to examine contemporary real-life situations and provide the basis for the application of ideas and extension of methods. In this thesis we follow the guidance of Robert K. Yin [Yin] for conducting case studies. He defines the case study research method as "an empirical inquiry that investigates a contemporary phenomenon within its

real-life context; when the boundaries between phenomenon and context are not clearly evident; and in which multiple sources of evidence are used" [Yin, p. 23].

As per case study methodologists [Stake], [Simons], [Yin], a case study consists of the following steps that should be used:

- Determine and define the research questions
- Select the cases and determine data gathering and analysis techniques
- Prepare to collect the data
- Collect data in the field
- Evaluate and analyze the data
- Prepare the report

In this section we provide details about the concrete settings of the case study that we performed on the topic of understanding the Business Value in agile context. We explain the goals of the study, the context and the data collection and processing.

We conducted a multiple-case study [Yin] to explicate the decision-making process during an agile project in the context of changing requirements. The case study consisted of semi-structured open-end in depth interviews with practitioners that work in organizations that develop software by using agile approaches. The case study is a first step in discovering the way in which the agile requirements mid-course decision process contributes to the client's value creation.

The companies, included in the study, characterized themselves as following agile methodologies. Some of them did strictly follow Scrum principles such as daily stand–up meetings and release retrospective. More detailed discussion about the study participants can be found in section 'limitations'.

3.3 The Case Study Process and Participants

We executed a rigorous case study by performing the following steps: (1) Compose a questionnaire; (2) Validate the questionnaire through an experienced researcher; (3) Implement changes in the questionnaire based on the feedback; (4) Do a pilot interview to check the applicability of the questionnaire to real-life context. (5) Carry out open-end in-depth interviews with practitioners; (6) Sample (follow-up with those participants that possess deeper knowledge or a more specific perspective).

Each in-depth interview included a section with questions, related to the business value perception and value creation. Those questions were:

- What does business value of a requirement mean for you?
- At your meetings with your clients/product owners, do they explicitly discuss the business value of the requirements, so that all meeting attendees understand why some requirements are of higher priority than others?
- Is 'value' connected to the business goals which the clients want to achieve by deploying the software system? If so, in which way does 'value' connect to clients' business goals?
- When judging the value of the requirements, do clients also consider any other factors (e.g. cost, size, risk)?
- Has the desired value been quantified? If yes, how?
- In which way, in your experience, does the agile process add value to the client? Can you give a specific example from your practice?
- For yourself, as part of the developing side – do you consider the value for your own organization, or is it more important what the client wants?
- Do you share knowledge about business value creation within the organization?

We make the note that no substantial changes in our interview protocol took place after the pilot interview, so that the pilot interview could be considered part of the case study. The study included 11 practitioners who were working for eight different companies/public organizations:

- 1 middle size company in the Netherlands (3 cases, 3 participants)
- 2 small companies in the Netherlands (3 cases, 3 participants)
- 1 small company in Bulgaria (1 participant)
- 1 middle size company in Bulgaria (1 participant)
- 1 university in Germany (1 student project)
- 1 country-specific business unit of a large international company, Italy (1 participant)
- 1 department in a large governmental organization, Turkey, (1 participant).

The participants described a total of 11 projects. The application domains for which software solutions were developed represent a rich mix of fields and include banking, ERP for small businesses, health care management, automotive, content management system, online municipality services system. In each organization we interviewed one or more representatives that were directly involved in the decision-making and the development process. Many of the participants performed multiple roles in the team and thus had a wide overview of the end-to-end process. Table 6 presents the participating

companies in respect to size of the client's and developer's organizations, and Table 7 explains the primary role the case-study participants had in the studied projects.

Here we make the note that further in this work we refer to the same case study and will point to the case study description provided in this section.

Table 6: Case study sites.

	Small (less than 25 employees)	Middle-sized (25-200 employees)	Big (more than 200 employees)
Client's organization	4	2	4
Developer's organization	4	2	2

Table 7: Participants in the Interviews.

Interviewee's primary role	Number of interviewees
Project Manager	5
Developer	3
Product Owner	1
Client	1
Scrum Master	1
Total Number of Interviewees	11

3.4 The Data Collection

We collected data from our case study participants by carrying out in-depth interviews. According to research methodologists [Charmaz],[Yin], in-depth interviews are intensive conversations with a small number of respondents to explore their perspectives on a particular project, practice or idea. We used this data collection technique because it is deemed useful when a researcher needs detailed and context-specific information so that he/she explores an issue in depth.

The interviews took place between July 15 and Nov 10, 2009. Nine interviews were done in face-to-face meetings. Two interviews took place over the phone. Each interview lasted between 60 and 90 minutes. Each interviewee was provided beforehand with information on the research purpose, the research process and the rights and responsibilities of the participating case study companies. At the meeting, the researcher and the interviewee walked through the questionnaire which served to guide the interviews.

We make the note that in each interview, the interviewer (that is the author of this thesis) used her judgment and tact to decide how closely to stick to the interview guide and how much to follow up the practitioners' answers and the new directions they might open up. Throughout the data collection, the interviewer attempted to verify her interpretation of participant's answers. She also summarized the key data immediately following the interview. The data was then transcribed and analyzed, which is described in the next section.

3.5 The Data Analysis

The data analysis in this study was guided by the grounded theory method according to Charmaz [Charmaz], which is a qualitative method applied broadly in social sciences. This approach is explorative and well suited for situations where the researcher does not have pre-conceived ideas, and instead is driven by the desire to capture all facets of the collected data. On the next step the data can be used to build a theory. The data analysis followed the following steps (1) Coding, which was focused on attaching a 'code' to a portion of the text; (2) Clustering all portions of text with the same code; (3) Creating lists with codes and clustering them into families; (4) Identifying patterns, i.e. multiple occurrences of the same mechanisms or concepts. These steps were executed by the author of the thesis, in collaboration with another senior researcher from the research group. The two researchers worked independently at two different locations. Each of them read through the practitioners' responses and searched for themes and patterns that appear to be common among the practitioners. A third experienced colleague acted as a checker in the process of identifying patterns (step 4 of the list above). This process was chosen because we believe that this way we minimize the following threats:

1) Possible bias from the researcher;
2) Possible misinterpretation of the practitioners' opinions;
3) Omission or overlooking of data that would lead to codes and concepts.

3.6 Answers to the Research Questions

This section presents the results in an order corresponding to our research questions formulated in Section 3.1.

RQ 1.3. What concepts of business value do practitioners in the context of agile projects perceive?

Table 8 summarizes what the participants in the study perceive as a business value.

Table 8: Understandings of business value from the interviews

The business value…
"…is in the context of the main functionality: does the feature support our main scenario?"
"…what the organization will gain when we implement the requirement".
"…usually it is what they like to see, what is used most (from workflow perspective)."
"…what will it means if we implement this requirement – will the client become more efficient, more competitive, will it gain something"
"…is defined based on: how much the client uses certain feature; whether it works good, and to help them do their work (in this case – the work flow)"

Table 8 indicates that in our observations, many of the business value definitions are context-dependent, that is, a definition could be traced back to a specific context characteristic of a project. For example, one of the participants, who worked on a project in the context of a software suppliers' network, defined the term business value as: *"Business value is to allow the client develop the functionality for which he/she is dependent on us"*. This perception clearly demonstrated the linkage between the perceived business value by this interviewee and the role he plays with respect to his clients in the suppliers' network. Another interviewee shared that *"perceptions of business value vary from project to project, even if you have the same client on site in both projects"*. Examples as these brought us to think that we cannot expect one universal definition of BV. Moreover, practitioners also indicated that the definition of the same client would probably change from project to project, depending on (i) the different project-specific settings, (ii) the specific needs of the client (for example, the need to have highly reusable or highly scalable software), and (iii) the

market position of the client's organization. To us, this all indicated that multiple layers of business value are clearly observable in agile project organizations and that it might make good sense to look into these layers in order to understand the underlying mechanisms responsible for the variation of perceived business value across agile projects within an organization. We consider it intuitive to think that agile projects may well vary in terms of how much of an agile approach they adopt in the project delivery cycle, and this, in turn, leads to variations in the perceived business value of both the system being delivered (that is, the product) and of the way it is delivered (the process).

RQ 1.4. In which way do specific or individual practices influence the creation of business value?

All study participants agree on that agile development better suits the project objective to satisfy customer needs and, hence, it leads to increased customer satisfaction, regardless of other project context characteristics as level of customer involvement, organizational culture, type of product, level of risk and requirements volatility. More in detail, the answers by practitioners and the agile practices they addressed are presented in Table 9 below.

As suggested in Table 8, our observations from the interviews let us conclude that business value is created by a combination of agile practices in a project-specific context. For example, in short projects with limited resources and a short list with requirements, the client profits from the agile process through (1) the efficiency of the process, (2) the 'savings' made by the light-weight method, and (3) the ability to figure out early what they'll get and whether it is what they need. This approach is deemed by our participants important to obtain the best possible system for the money spent.

Another example is in a context of volatile or unclearly defined requirements. In this case, the value is ensured by the change management mechanisms and by incorporating learning loops in the process [Schwaber 04]. An interesting finding in our study was that the views by all participants agreed on that the agile paradigm has an effect on the social aspects of project delivery, such as work moral and atmosphere, as well as on the relationship between client and developing organizations.

RQ 1.5. Do practitioner make value-driven decisions during agile development? If so, how decision-making is happening?

While the concept of business value was deemed important to all participants, when it comes down to making requirements prioritization decisions at inter-iteration time, we

found a surprising result: nine out of eleven participants stressed the importance of, what they called, a '**negative** value'. In their requirements prioritization experiences, the line of reasoning that prevailed was not concerned with how much value a certain feature would add to the product, but instead with the question of how much it would detract from the product's value, if the developers would not implement this feature. (The amount value detracted was termed 'negative value'). The negative value thus is equivalent to loss of value or damage to the business. If something important is missing, its value is not zero, but it has a negative value. This is linked to damage for the business or impossibility to support business needs or processes. As one participant put it "...the question was always asked: how big is the damage if we don't have this or that requirement; how complicated would it be to use a work-around?...". Further in this work we will use the word 'damage' when we discuss this phenomenon.

In the experience of one practitioner, this line of reasoning reflects a professional pragmatic behavior especially in project contexts where the resources are very limited and the client is concerned about whether or not s/he derives maximum benefit from the project. This phenomenon, though, seems to be observable not only in small projects. One of the study participants reported about a project where a big client (a bank) demanded urgent implementation of additional requirements and re-scheduling of the project, because the losses the bank suffered from the lack of this functionality were bigger than the extra costs involved for development, re-scheduling and postponing the implementation of other functionality.

In the experience of one practitioner, this reasoning reflects a professional pragmatic behavior especially in projects that have very limited resources and clients preoccupied with whether or not they derive maximum benefit from the project. Unlikely to contexts in which scarcity of project resources is an important concern, in projects which enjoy 'more generous' budgets, practitioners agreed on that their decisions were driven mainly by value consideration, namely supporting the main functionality of the software system being delivered and keeping in mind the damage. We note that making decisions by considering 'negative value' sounds intuitive, as the scarcer the resources, the more conscious the project teams are on how to spend them.

Next, we observed that there is a link between the perception of value and the price for the implementation.

The clients want to see a link between the perceived value of a feature and the cost of implementation. They don't always understand the relation between the effort estimations of the developers and the perceived amount of work (from the viewpoint of

the client). For example, a client wanted an additional button. Estimations and analysis of the system architecture showed that this one button will require significant changes and was estimated at 20 hours of work. The client could not understand the rationale behind this estimation and debated it. The developer decided to look for a work-around to implement this button in a more time-efficient manner and thus please the client. Similarly, in another project the developer prepared a demo to better understand the client's needs. The client, on his turn, could not understand why he has to pay half a million for this one week of work, as the difference between the real system and the mock-up was not clear to him.

Last, the value-creation process plays an important role for the developers' organization, not only for the client's company. The agile practitioners' literature [Aurum 07], [Boehm 81] seems to share the opinion that the only value-creating considerations that drive the development decisions are those of creating value for the *client*. During this study we made the consistent observation that, more often than not, the value creation for the developers has been considered as well.

RQ 1.6. How do developers combine value creation for their own organization with value creation for the client's organization?

The practitioners shared the views that in the software project organizations, the developers regularly perform their own estimations and revise their understandings of how the business value from the client's standpoint relates to the bottom line of the developers' companies. They explained that this developers' value-conscious estimation happens, because of the pressure on the developers to maximize the value creation for the client, only while ensuring a descent level of profit for their own companies. This means that not all wants of the clients get implemented at inter-iteration time, and certainly, not all requirements that the client specified at project inception time are implemented. Overall, the practitioners agreed that the developers are active participants in the requirements decision-making processes. Their participation is deemed even stronger in cases of small projects, where the client is a small organization or company that does not possess knowledge in the IT domain and cannot afford paying extra for IT consulting services. Such a client may even find it very expensive to allocate a resource to the role of 'on-site client'. Often, it is economically unfeasible for the client organization to pool away a full-time employee from their every-day business and task him/her to serve 'on-site' in an agile project. In such a context, it happens that the client delegates the decisions influencing the value creation, to the developing team. Our case study participants indicated with certainty that a high level of trust is a prerequisite for such

cooperation. Some participants described situations where they even had to 'save the clients from themselves', meaning to prevent unwise decisions or suboptimal choices that will be harmful in long term. The practitioners motivated this course of action with their experience from previous projects at the client's site as well as their profound knowledge of the client's business domain. The developers also justified this behavior by their desire to create maximal value for the client and, thus, to contribute to a successful project. In the experience of our interviewees "this leads to high client satisfaction and good relationship with the client, which will, eventually, lead to future mutual projects". This observation represents an interesting point for further discussions and research, as it does not converge with the common understanding in the agile literature that the customer is responsible for making the (prioritization) decisions. We think, therefore, that knowing more about the variation in project contexts is key to understand how relevant project context characteristics possibly affect the choice of decision-making approach in a project.

Table 9: Agile practices and business value creation

Answers to the question of how agile software processes create business value	**Practices addressed:**	**Results in:**
"…The clients are included in the development process which enhances the understanding between the parties."	Client's involvement	Satisfied client, better relationship
"The process was adding value. The project included many relatively small requirements; there was a high through flow in the PB (product backlog) that you can not handle in a waterfall way."	Handling changing requirements	Creating a product that the client desires and that answers to changes
"The clients prefer to get something more often instead of one big thing once per year that might not be what they want."	Frequent releases	Satisfied client
"…by the efficiency of the process, the 'savings' made by the light-weight method, and by figuring early what they'll get and whether it's what they need. Gives the client peace of mind! Gives them the idea about what they'll get, at the same time they don't pay up front and don't have to sign a peace of paper; also they know that they can add something if they forget."	Close cooperation with the client Requirements' changes are allowed	Satisfied clients; Harmonious trustful relationship; peace of mind; less probability for requirements creep
"if it was not agile, we could have made a completely different system from what they want. Especially in this case where the requirements were not SMART. We discovered very early what they really want. Otherwise you need much more specific requirements. Also, for the developers – they have more voice, there is more interaction, they are happy. We have almost no cases of people that leave the company."	Small releases and frequent demos	The developers are happy! And work better; creating the right product, happy client, no waste of time and resources

"You can show very early what they can get; and you can manage expectations – what to expect and when."	Early release	Happy client, realistic expectations
"I don't believe in requirement documents; I think they are exactly as good as a card, and all the rest effort (specification, etc.) is a waste. Another good think is that you don't sign something up front. It is good for both sides, and for us to make expectation management; Agile makes products faster to market; you have de facto demo every 2 weeks, which is extremely helpful, as nobody can do everything right from the first time. It allows the client to collect feedback, observations and experience from the beta-versions, and so the first version in production is much better."	Slim RE process, less documentation, frequent releases, incorporate learning	Good use of resources, no waste, lower risk (do not sign something fixed up front), faster to market, creating the right product, higher quality
"You can make changes during the project; nobody knows in advance what they really want. This process helps them to see what happens, at an early stage."	Change management, early releases, incorporate learning and experience	Better product and right product via learning
"…to reduce the time for development…The project team is more cohesive, the experience is shared, also to the whole company."	Information sharing techniques	Faster time to market, better team work, information sharing
"The special benefit of agile is that the client can better influence/re-define what he gets for his money."	Client participation	Happy clients, more 'value for money'

3.7 Limitations

In this section we explicitly address the possible threats to validity of the observations and conclusions in the case study as per the recommendations of qualitative case study research methodologists [Wieringa 09-1], [Yin].
For this purpose, we used the checklist for case study researchers recommended by Runeson and Höst [Runeson 09]. First, the external validity addresses the generalizability of our observations and conclusions beyond the studied sample of companies, projects and participants. With this respect the following aspects of our study can be considered as possible threats to the validity: (i) the number of companies and projects that have been studied; and (ii) the choice of study participants. With respect to the first aspect, we admit that the scale of the study does not allow us to make statistically relevant observations. However, this was not the purpose of our study.

For a qualitative study, the question is rather [Wieringa 09-1], [Yin]: to what extent the companies included in the study can be considered as representative for a broader range of companies? This question addresses the second aspect. In the study, we deliberately included representatives of companies of different sizes, (although we admit that the majority of the companies were representatives of small and middle-sized businesses); business sectors and geographic location, as well as different stakeholders from projects with different sizes. Based on our observations, we think that some of the findings apply generally across the cases, despite the heterogeneity of the set of case studies. This was confirmed by participants in a panel discussion on the first results of the study. This gives confidence that the conclusions hold for other companies in similar context as well. It is for this reason we have searched for aspects that the cases have in common rather than aspects in which they might differ. We believe that the most important aspects of the context, in this case, are: geographic location of the company, company size and project size.

With respect to the validity of the conclusions that we made, we admit that researcher's bias is possible and can not be completely excluded. However, we undertook measures to minimize it. The analysis made by the author was checked with other researchers (Daneva, Sikkel) whose review ensured that the conclusions are traceable to the data pieces being collected.

We underline that we don't claim universality of the results across all possible settings and contexts. We are conscious about that the results might be different if a similar

study is performed in large companies/ projects, or in companies with different cultural background, or operating in different market and business environment, for example in in North America.

Furthermore, the choice of the companies represents a threat to the validity of our results. We asked the question: are the projects we sampled indeed agile? As we are analysing agile processes, we want to be sure that this is in fact what we are investigating. The important question to discuss is how we (the researchers) knew that the processes we studied were indeed agile. We took two steps to minimize the effect of this validity threat: we confirmed with all case study participants that they (or their team) did apply an agile methodology; the participants stated that their organizations were known as agile method adopters and that they were committed to use agile in the projects that we collected information about. Next, during the interviews, we consciously watched for confirmation of whether the interviewees indeed referred to examples of their experiences in agile projects (and not in projects that used other approaches). We acknowledge that the questions of where the line is between agile and other iterative and incremental approaches and which characteristics of the project should be observable in order to deem a team or a project agile, are philosophical in nature. These questions are, however, out of this thesis' scope. To the best of our knowledge, the projects we investigated in this study are truly agile projects in the sense of the Agile manifesto [Agile].

Last, our interviewees agreed on that the agile process helped them create rapport with their clients easier than it could have been possible in a project that uses a traditional delivery approach. The interviewees also agreed that the agile process makes it "much easier than ever before" to consistently maintain communication with the client organization. They felt that this all was instrumental to "making the clients happy". Clearly, one could raise the concern that these observations are biased, as they are provided by agile practitioners. More than 70% of the interviewees were seasoned practitioners with more than 15 years of software project experience and that they accumulated a large part of this experience while working for organizations with traditional software development approaches. So, they had enough practice to compare the both worlds. However, we admit that there is a possible threat to validity as we interviewed only those people that are currently engaged in agile development. Theoretically, there could have been people that came back to traditional approach. However, searching for such practitioners to involve in the case study was not a feasible option within the resources we had.

3.8 Conclusion

In Section 3 of this chapter we present the results of a case study on the understanding of agile software practitioners about the concept of business value and its creation during the development process. All participants in our explorative study expressed the opinion that agile projects make clients satisfied by the outcome of a project. This was a point of convergence across case study sites regardless of the significant variety in project characteristics (type of software project, organization size, culture) at these sites.
Our observations from the case study suggest that business value is more than just numbers. It comes out of a human judgment that is based on competencies and deep knowledge of the client's domain. An interesting question then is how a judgment about business value is formed and what tacit knowledge should be made explicit, so that knowledge about business value gets shared among developers and clients. Second, the existence of objective values to feed as input into the decision making process in agile projects and in particular in the prioritization methods, is questionable; instead, what is priority seems to be a combination of subjective value-based criteria.

Next, we note that making decisions based on the consideration of a possible damage to the value creation of the project in case a feature is not implemented (we borrow from the practitioners the concept of 'negative value' for this) sounds intuitive, as the scarcer the resources, the more conscious the project teams are on how to spend them. The concept of 'negative value' suggests that we may need to redefine the concept of business value in agile all together, for example by adding damage. Second, this concept clarifies the type of value that feeds into the decision-making processes in agile projects. The decision on requirements priority should be made both on achieving value and on avoiding loss. Particularly in those agile projects that are tight on resources, it is very important to avoid loss, as the customer cannot afford losing very much.

Third, this observation questions the assumption that the BV is the main driving consideration during the development process. Moreover, when analyzing the projects of the participants that referred to 'negative value' as a decision-making criterion we observe that they belong mainly to small and middle size projects, with fixed duration and/or fixed price-contractual agreement. This observation suggests that there might be a link between the context of a project and the instantiation of the value-creating practices used in a project.

With respect to the value-based decision-making we observed that the consideration of value as a decision-making criterion is complex. Furthermore, there is a *difference in the perception of value for the developers and clients*, and the value creation *for the developers* impacts the input from the developers, and thus the decision-making process. Clearly, developers and clients have some goals that ensure mutual benefits to incur e.g. "we want to make the client happy, so that he/she comes back", while other goals on the developers' side may not be related to one particular project or one particular client, and instead are related to issues like reuse, other concurrently running projects and distribution of resources for maximizing value for the supplier's organization.

We consider it an important finding of our case study that the value-creation process plays an important role for the developers' organization. In contrast to the literature sources [Aurum 07], [Boehm 81] that seems to share the opinion that the only value-creating considerations that drive the development decisions are those of creating value for the *client*, we made the observation that, more often than not, the value creation for the developers has been considered as well. Nevertheless, we consider it an interesting finding from the case study, as it explicates the importance of the both perspectives (vendors and clients), and the need of balance between the value consideration of these two groups stakeholders.

We need to consider more carefully in which ways development teams balance the client's business value with their own organizational bottom-line and consider this an important topic for future research on its own right

4 Concluding remarks

This chapter was dedicated to the study of the concept of *Business Value*. We presented the results of a systematic literature review (Section 2) and a case study (Section 3).

Our findings from the systematic review are that:

- the majority of papers in agile SE literature do not define explicitly the concept of business value in an elaborated way;
- there is evidence that certain agile practices contribute to the value creation;

We argue that the systematic literature provides insufficient insight into the essence of *business value* and therefore this concept needs to be investigated by the means of a case study in real-life setting.

The results of the case study extend the findings from the systematic literature review. Our observations suggest that the notion of *business value* is complex. It is often based on competences and knowledge of the client's domain. Furthermore, it might be linked to the project's context and to the viewpoint of the observer.

Next, we observed that *business value* is not always the main driving consideration during decision – making.

These findings makes us think that the existence of objective value to feed as an input into the decision making process might be questionable.

A closer look at concrete value-creating practices is needed, as well as at the concepts that impact them. We come back to this topic in Chapter 6, where we propose a direction for future research.

As already mentioned in Section 2.7, next we focus on the process of requirements prioritization as one of the value-creating practices in agile context. In next chapter we present a detailed look at this process, and at the concepts that are being considered during the decision making at inter-iteration time in agile projects.

Chapter 4

Requirements prioritization in agile projects[2]

In this chapter we first discuss the results of an empirical study of the agile requirements prioritization process and investigate the concepts that are important to consider during (re)prioritizing requirements at inter-iteration time. Next, we present a conceptual model which describes on an abstract, generic level, the concepts that seem to impact the agile prioritization process and the relations between them. Last, we provide a mapping between the concepts of the model and the RP techniques as described in literature.

[2] This chapter is based on the following papers:
Racheva, Z. and Daneva, M. and Herrmann, A. and Sikkel, K. and Wieringa, R.J.(2010)Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. In: 18th International IEEE Requirements Engineering Conference, 27 Sep - 01 Oct 2010, Sydney. pp. 147-156. IEEE Computer Society. ISBN 978-1-4244-8022-7
Racheva, Z. and Daneva, M. and Herrmann, A.(2010)A Conceptual Model of Client-driven Agile Requirements Prioritization: Results of a Case Study. In: IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), 15-17 Sept 2010, Bolzano, Italy. 39:1-39:4. ACM. ISBN 978-1-4503-0039-1
Zornitza Bakalova, Maya Daneva, Andrea Herrmann, Roel Wieringa: Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature. REFSQ 2011, pp. 181-195, Essen, Germany

1 Introduction

One of the findings of the systematic review on the topic of value creation that we presented in Chapter 3 section 2 is that agile projects create value by exploiting a number of practices. One of the practices that is of paramount importance is the choice of requirements that will be implemented at each iteration of a project, as this choice shapes the final product and determines the value, created for the clients. This choice of requirements is realized through the technique of frequent requirements re-prioritization (RP) that happens throughout a project, typically at the beginning of each iteration or during the planning phase of an iteration. For this reason we used the terms 'inter-iteration decision making' and 'mid-course decision making' as synonyms of 'agile requirements prioritization'.

In fact, RP is an essential mechanism of the agile projects to maximize the business value for the clients, and to accommodate changing requirements. The practices of regular RP, with strong client participation, are a relatively recent phenomenon. They appeared as an intrinsic ingredient of the agile methods. For this reason they are only partially understood. We consider our findings presented in this chapter as an important contribution to the existing body of knowledge on agile requirements prioritization, as it is an additional step towards understanding the way in which agile projects produce value to the clients (or to the product owners as clients' proxy) through the requirements prioritization activity.

In this chapter we make three contributions:

1) We present the results of an empirical investigation on the requirements prioritization process by means of an exploratory case study. The purpose of this study is to shed light on the real-life practice of requirements inter-iteration decision making. As we described in Chapter 1 Section *Methodology*, we use the same case study as in Chapter 3 Section 3. The case study is deliberately composed to investigate various research questions and in this chapter we address those RQs that pertain to the requirements prioritization process. We investigated real-world cases with the overall research objective to uncover how mid-course requirements prioritization takes place in industrial settings. We want to stress that the case study was exploratory because we had no pre-conceived ideas about the possible answers to these questions. Our purpose was to 'look under the hood', to observe and identify those concepts

and relations that drive the requirements prioritization as a value-creation process.

2) A conceptual model of the agile requirements prioritization process in terms of concepts and relations between them.

3) A mapping study that explicates a gap between the prioritization methods described in literature and the process as we observed it in the case study.

The chapter is structured in three parts, respectively:

1) In Section 2 we discuss the case study results and provide answers to the following research questions:

RQ 1.7. Who are the decision makers in the prioritization process? Which roles are involved and what are they responsible for?

RQ 1.8. Which are the characteristics of the project settings that are essential for the way a requirements prioritization process is carried out in a project?

RQ 1.9. Are there any other ways (beyond the selection of requirements) through which the requirements prioritization process adds value to the project?

2) Section 3 presents our conceptual model, and

3) In Section 4 we set out to answer the following research question: Which concepts of agile prioritization are shared in practice and in literature and how they are used to provide guidance for prioritization? We answer it by mapping the existing agile prioritization techniques discussed in Chapter 2 Section 3 to the categories that form our conceptual model of agile RP.

The studies were performed in the following order: (i) the literature review, that is already described in Chapter 2, (ii) the case study, and (iii) – the mapping study.

2 Agile RP process – case study results

2.1 Case study description

The case study results, presented in this section, came out from the same study that has already been described in Chapter 3, Section 3. We want to stress that although the participants in the case study are the same, the results in this chapter capture, summarize and analyze a different sub-set of the data collected in the study. As we explained in the case-study description, we performed an explorative study and the data collection was effectuated by the means of semi-structured open-end interviews. The interviews were led by a questionnaire that was prepared up front by the researcher. It consisted of three parts that deal with different aspects of the interviewee's experience with regard to the decision-making about requirements. The questionnaire was composed in such a way that first the focus of the interviewee was drawn to a concrete project from his/ her practice. Next, his/ her opinion and experience as a whole, was addressed. In the questionnaire this was implemented through two parts of questions that consisted the first and second part, respectively. The results presented in this section rest on the data collected in part 1 and 2 of the questionnaire. Those pertain to the questions related to the prioritization process, while the results in Chapter 3 reflect the results about the BV and present a view on the data obtained in part 3 of the questionnaire.

As we already explained above, here we'll not provide the description of the whole case study as this has been already done in Chapter 3 Section 3.

Below in Fig. 8 we provide the questions used to collect data on the topic of requirement prioritization practice. The whole case study protocol is available in Appendix 2.

Part 1: A concrete case of requirements prioritization.	*Q1. How often did re-Prioritization happen?* *Q2. Can you remember actual cases cases of (re)prioritization in the project?* *Q3. Do you know what triggered them? How did you proceed in these cases?* *Q4. Which factors played a role during the decision making?* *Q5. Are there any reprioritization decisions that were hard to take, or took a lot of discussion?* *Why were they difficult?* *What were the pros and cons* *what tipped the balance* *who in the end made the decisions*

	does it always work like that *was this a weird case and usually it works differently?* *Q6. Are there any reprioritization decisions that you regret? (i.e. where you think a wrong decision was taken? or one that looked good at the time but turned out badly)* *Q7. And some that you feel particularly good about? (i.e. where increased insight or serendipity made you change the direction of the project for the better?)* *Q8. Where did changing requirements initiate from? Where they initiated by the client or by the developers?*
Part 2: Prioritization process (general observations)	*Q1: Who is responsible for the decisions on priorities within one project and in each iteration?* *Q2: Are any processes used to help prioritize and select requirements for an iteration? For example, the Planning Game?* *Q3: Are any software tools used to help prioritize and select requirements for an iteration?* *Q4: Which methodologies/techniques are informally used to select and prioritize requirements, if any?* *Q5: Do you use explicit criteria for the prioritization? If yes, which?* *Do these criteria change from project to project or in different products? If yes, why? For example, if in one product the client is more important than in another, or if one project is more beneficial to your organization than other.* *Q6: Do you use any formal way to calculate the worth of a requirement, e.g. cost-benefit analysis, a value estimation formula, a risk estimation technique?* *Q7: Are you happy with the way requirements are currently prioritized at inter-iteration time? What do you like in the existing approach?* *Q8: Do you have cases, when a client imposes his preferred way of prioritizing the requirements and then you adopted this in the prioritization process for the project?* *And this let the team change the way you prioritize, which might be different from what you typically do in other projects? If so, which parts of the approach do you adapt? Why?* *Q9: Do you prioritize all the necessary work like refactoring, testing, etc., together with the clients' requirements? Do you need to explain the client why this is necessary?* *Q10: Do you consider dependencies between the requirements? If yes, how do you handle them at prioritization time?* *Q11: How did you keep track of these dependencies?* *Q12: in the past, have you experienced cases where:* *12.1 You had to educate the clients' representatives why prioritization is needed;* *12.2 The decision-makers did not have the information, necessary to do prioritization;* *12.3 The decision-makers did not know why and how to prioritize. If yes, please explain why, in your opinion, this happened.* *12.4 The stakeholders were not satisfied with the priorities and this had a negative effect on client's or yours organization?* *12.5 Where the client requested changes in the backlog, do you know why did it*

	happen? *12.6 The prioritization decision had impacted the system in a significant way (big refactoring or rework needed).*

Fig. 8: Questionnaire

2.2 Research Method

To obtain the concepts impacting the prioritization decisions from the case study data, we used the Grounded Theory (GT) research method. The Grounded Theory approach was first suggested by Glaser and Strauss [Glaser 86]. It is a qualitative method applied that finds broad application in social sciences to construct general propositions (called a "theory" in this approach) from verbal data. This approach is exploratory and well suited for situations where the researcher does not have pre-conceived ideas, and instead is driven by the desire to capture all facets of the collected data and to allow the theory to emerge from the data. Strauss and Corbin [Strauss 91] define it as follows: "The grounded theory approach is a qualitative research method that uses a systematic set of procedures to develop an inductively derived grounded theory about a phenomenon". The primary objective of grounded theory, is to expand upon an explanation of a phenomenon by identifying the key elements of that phenomenon, and then categorizing the relationships of those elements to the context and process. We consider GT to be an appropriate vehicle to use as we want to identify concepts that would help to understand a phenomenon, in our case – the agile requirement prioritization process. GT is a way to extract the theory of the world that our interviewees have. As our purpose is extracting this theory (consisting of a conceptual framework and generalizations about the real world believed by the participants), it is an appropriate approach to use in our case.
The application of GT consists of the following steps: open coding, axial coding and selective coding. Open coding is based on the concept of data being "cracked open" as a means of identifying relevant categories. Axial coding is most often used when categories are in an advanced stage of development; and selective coding is used when the "core category", or central category that correlates all other categories in the theory, is identified and related to other categories. That is, the application of the GT traverses the following phases: Developing Concepts, Core Categories and Describing the Results.

In this method, the data is constantly compared to the previous data items, until a point of saturation is reached, i.e., where new sources of data don't lead to a change

in the already identified concepts and relations between them, that is – the emerging theory in the terminology of the research methodologists.

2.3 Applying Grounded Theory

For the purpose of this research, we used the GT guidelines by K. Charmaz [Charmaz 07] in order to answer our RQ posed in the Introduction of this chapter. In essence, this exercise was a process of making analytic sense of the interview data by means of coding and constant comparison of pieces of data that were collected in the case study. To process the data in this study, we - the researchers, employed the following techniques for concept coding: a) writing memos for every interview summarizing key themes; and b) writing a 'researcher journal' that brings together key concepts across all the interviews.

The data processing followed the process described in Coding manual [Saldana 09], p.12-13, p. 48, by applying the generic coding steps:

1) Context capturing;
2) Descriptive coding;
3) In-vivo coding.

The author executed the steps, listed below. The daily supervisor (Daneva) coded independently two interview transcripts and her results were compared to the results of the author. Whenever differences were found, these were resolved in a discussion.

In particular, we traversed the following steps:

1. We read the interview transcripts and attached a coding word to a portion of the text (might be a phrase, a paragraph or a sentence), a coding word or phrase. They were intuitively chosen to describe in a condensed way the subject this portion of text relates to. This could be a concept (e.g. 'value', 'method'), or an activity (e.g. 'estimation') or abstract or social phenomena (e.g. communication, feelings). Some of the codes emerged directly from the descriptions of the project settings in which the agile requirements prioritization process took place, for example, settings such as 'the size of the team', or 'decision-maker'. Other codes, and respectively - concepts, emerged during the coding process as a result of observations that we didn't anticipate and concerned aspects of the process that we haven't explicitly addressed in the questionnaire. Those are for example 'problems that the developers encountered during the process' or 'feeling about the way of work'. We

clustered all pieces of text that relate to the same code in order to analyze it in a consistent and systematic way. A snapshot of the coding process can be shown on Fig. 9.

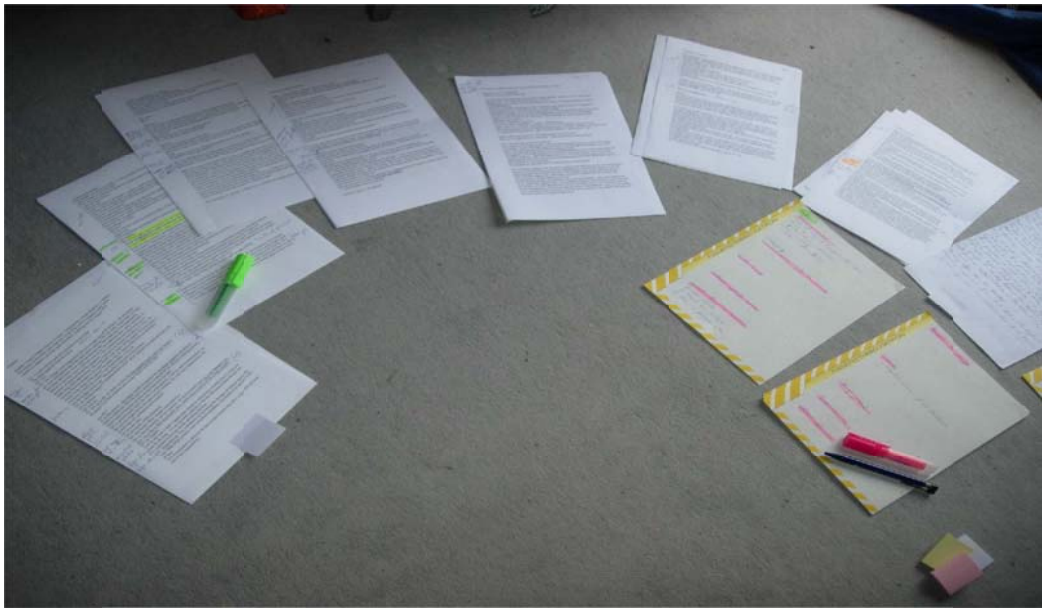

Fig. 9: Manual Coding Process

2. We collected in one place all portions of text that had the same code attributed to them. This way we collected all available to us information, pertaining to the same coded item. That is, all pieces of text coded with 'estimation' created a set of phrases and sentences that describe how estimation is being done. Then we summarized the information, gathered under each code, together with links to the source document.

3. Next, we created list with the codes and clustered them into families consisting of those codes that seem to be related to the same phenomenon or concept. For example, the family "Relationship with clients" consisted of the codes: *Communication with client, National culture, Corporate culture, Psychological issues.*

4. We identified patterns, i.e. multiple occurrences of the same relations or concepts throughout the transcripts. We captured these observations in a separate document, with links to the interview part where we made the observation. We created this document while working on the transcripts, in parallel with the coding procedure. This was an iterative process, i.e. this document was revised and complemented with other observations, done after the coding was completed. The document with the observations contains

relations that we observed to be in place between certain aspect of a process and project settings. The observations were ordered by number in the order they appeared on the list, for further processing. Here we make the note that we tried to remain as objective as possible by formulating them, and to stick with the underlying data. Nevertheless, we are aware of the fact that those observations and conclusions are made through the subjective lens of the researcher. We used this document at a later phase during the creation of a conceptual model of the prioritization process. The document served us to identify relations between the concepts on the model.

We used the codes twofold: (i) to structure the knowledge about requirements prioritization that we extracted from the case-study participants and to answer the RQs that we posed at the beginning of this section; and (ii) to identify those concepts that are considered by the stakeholders during requirements prioritization, as an input to the model that we present in Section 3.

2.4 Results

In this sub-section we present the results of the case study that pertain to the following research questions that we posed in the introduction to this chapter.

2.4.1 Decision-makers in the prioritization process

RQ 1.7. Who are the decision makers in the prioritization process? Which roles are involved and what are they responsible for?

Our findings indicate that in our cases the developer plays a much more important role in practice than what is recommended in the literature. Below we provide examples from the interviews:

To the question: "Who is responsible for the decisions on priorities within one project and in each iteration?" we received following answers:

P1: "… in this project – the client; usually in practice the developer proposes priorities in order to reduce his own risk, and to propose a material for the decision-making sessions, which helps to keep the sessions shorter and more focused."

P2: "Prioritization is usually done by the developers based on the following two criteria:

1. the availability of people;
2. the sequentiality of the stories.

The process consists of initial planning for the project, where the client is involved and the priorities are defined. Afterwards the developers took care of that…The developers decide on priority, because the client does not know about the impact new requirements have on the project."

P3: "When making decisions we always consider what is reasonable to be implemented, from our understanding."

P4: "I am the one who prioritized, (a product owner/ business analyst at the developers site) based on: Who called and how many times! And what the developers think is good for the system as a whole – what would better serve the clients and will make them happy, as an end-effect. "

P5: "Q1: the project leader and the developers, plus the client. Customer's relations are essential; we want to make them happy but we don't just do whatever they ask for. Instead, we try to understand what his problems are, and his domain, so that we (the application) can better serve their **needs,** for example by smart architectural decisions. They gain as well from this approach. Also, when we know the goals, we can advise the client better, based on our past experience from other projects. For example, if they want something that we know from past experience that it didn't work."

As the citations show, the practitioners' opinions converge on that the developers are active participants in the requirements decision-making processes. We observed the following situations: (i) the decisions were delegated fully to the developers; (ii) the clients required changes or faster implementation of certain functionality, without participating in other prioritization activities; (iii) the client participated during the project in a traditional way, e.g. by approving changes to budget. We provide some details about these cases in the following sub-sections. In only one of the investigated projects the client was the one driving the prioritization and making the decisions about the requirements that will be included for implementation in the next project iteration. In this project, the client had significant previous experience in software development projects.

2.4.2 Developer's participation

We observed that the participation of the developers in the decision-making processes is stronger in small projects, where the client is a small organization or company. First, such clients don't possess knowledge in the IT domain and can't

afford paying extra for IT consulting services. They may even find it very expensive to allocate a resource to the role of 'on-site client'. Often, it's economically unfeasible for the client organization to pool away a full-time employee from their every-day business and task him/her to serve 'on-site' in an agile project. In such a context, it happens that the client delegates the decisions influencing the value creation, to the developing team. In one of the projects that we investigated the developers took over the decision-making because the client (car dealership) didn't possess the time and abilities to analytically reason about the system he needed. This makes us think that there are certain constellations of contextual factors that will always lead to delegating the decisions to the developing organization.

Second, often the developers possess knowledge both in development and in the respective subject domain, as teams are specialized in developing a specific class of applications (e.g. banking, health-care, ERP, etc.). Some participants described situations where they had to 'save the clients from themselves', to prevent unwise decisions or suboptimal choices that will be harmful in long term. The experience from previous projects as well as the profound knowledge that the developers had of the client's domain justified this course of action. "Also, when we know the goals, we can advise the client better, based on our past experience from other projects. For example, if they want something that we know from past experience that it didn't work." The developer was led by the desire to create maximal value for the client, to contribute to a successful project. In the experience of our interviewees "this leads to high client satisfaction and good relationship with the client, which will, eventually, lead to future mutual projects".

Our case study participants indicated with certainty that a high level of trust is a prerequisite for such cooperation. One project manager reported: "Customer's relations are essential; we want to make them happy but we don't just do whatever they ask for. Instead, we try to understand what their problems are, and their domain, so that we (the application-builders) can better serve their needs, for example by smart architectural decisions. They gain as well from this approach …"

In our case study, the interviewees went further to explain why developers are so strongly participating in the decision-making. In their view, the developers' company is the one to make sure that the project delivery process runs in a way that is profitable for the company. If developers accommodate all wishes which clients might come up with at inter-iteration time, the company may find it not sustainable in

the long run. This observation raises the question about value considerations for the developers, discussed in detail in [Racheva PR 10].

The matter that developers strongly participate in the prioritization and decision-making gives us the hint that agile and traditional requirements engineering processes may not be that different regarding who prioritizes the requirements.

2.4.3 Client's participation

The observation we made is that, in contrast to the agile best practices, in most of the cases we observed that the developers are those who prioritized, and the involvement of the clients was to approve the plan/give comments. Only in few cases we saw that the client is really capable/interested/aware of the agile way of defining priorities, and thus able to decide. We provide possible reasons for this observation in the discussion of this section.

Next, our interviewees indicated that the client's judgment of importance regarding a specific requirement might not always be representative for the client's organization as a whole (despite the matter that the client is supposed to represent the organization and speak on behalf of all users). In such situations, it might well be the case that the client consciously or unconsciously manipulates the developers to implement specific requirements. The developer has no possibility to collect more objective information about the situation and to judge the extent to which s/he could trust the client's sense of priorities. One participant reported her experience in a case of a client who asked for a certain report. According to the client, this report was 'very important'. It took significant efforts on the developers' side to prepare it. Later it turned out, that this client's representative was the only person in the whole company reading this report.

2.4.4 Characteristics of the project's settings that influence the prioritization process

RQ 1.8. What are the characteristics of the project settings that are essential for the way a requirements prioritization process is carried out in a project?

We found that the requirements prioritization processes vary regarding the forms of *clients´ participation and collaboration in the process*. In the following subsections we discuss two context factors, which seem to be responsible for this variation. Those are: *size of the client's organization*, and *size of the project* in terms of resources (budget and time).

We make the note that the observations that we discuss below are result of our analysis of the projects that have been discussed in the case study, as we didn't ask the case study participants explicitly about the relations between the process instantiation and the context of the projects.

Impact of the size of the clients' organization

The observations concerning the size of the client's organization and the process are summarized in Table 10. First, in all cases where the client was a small company, except one, the decisions were delegated completely (in 2 cases), or mainly (1 case), to the developer. The special case was a small academic project where the professor served as a project manager and as a client. In this case the client considered herself experienced and led the whole process. In the other three cases, the developers took up the responsibility to oversee the work process and provide a solution. As one participant put it: "…we tried to look from the perspective of different users in order to understand what they need first…"

Table 10 : Variation in the prioritization process based on size of client's organization

Size of clients' organization	Process specifics
Small client company	The client can't allocate resources for participation and in the most cases does not possess the knowledge needed.
Middle-sized client company with rigid structures (e.g. government or municipality case)	Client's cooperation is limited, don't allocate resources, don't agree to client on site or even to developer on site. The behavior changes only after few iterations where they see the benefits of the agile methodology.
Large client (e.g. banking sector)	The relationship becomes more strictly defined, changes require participation of higher-level management.

Second, in the two cases which we observed to have middle-sized organizations as clients, there were serious problems with the client's participation. In one of these two cases (namely, in a municipality), the client did not want to provide resources for either a 'client on site' or a 'developer on site'. The project manager that we interviewed even went on writing to ask an advice (what to do in such case) from S.

Ambler, a prominent agile practitioner who authored a large number of publications on agile practices. The answer was: "you have to have on-site client." In the other case of a middle-sized organization, there was a tension between the product owner (PO) (that is, the client's representative from the developing organization) and the developing team, as the PO persistently wanted the continuous implementation of new functionality without considering the technical feasibility.

In the middle-sized organizations we observed that the client's behavior eventually changes during the project, but no earlier than when a few iterations pass and the benefits of the agile approach become visible. A similar situation was observed in two of the big organizations. One project manager reported that: "…Now a 'we' feeling emerged, in contrast to the polarized 'them-us' attitude from the beginning of the project. Now we started doing the prioritization together and they learned how to prioritize…"

Third, in all cases of big clients, we observed a more strictly defined decision-making process, including multiple levels of client's involvement, reaching up to the higher management when it comes to decisions concerning the project development. The big clients were four in total: three were in the financial sector and one in the automotive sector. In essence, all four projects deployed a requirements prioritization process that was a blend between traditional and agile. Our observations that a blended process is used in large organizations converges with reports in the literature on hybrid methods [Fitzgerald 06].

Impact of the factor resources

When it comes to resources, we observed that small projects with very limited resources aim at creating a product with a certain minimum of functionality that is absolutely necessary in order to satisfy the client's needs. From the 10 projects we investigated, five were with strictly limited resources: in terms of time (one academic project), schedule (two projects), and money (two projects). Out of the five projects, four projects had small organizations as project clients, and one project had a middle-sized company as its client. In four of the projects the order of implementation of the features was almost irrelevant to the value creation because the projects were relatively short (2–5 months). In these four cases, of paramount importance was a minimum of functionality that needed to be implemented within the available resources. Without this functionality the product would be of no value (or very low value) for the clients. Table 11 captures our findings in this respect.

Table 11: Variation in the prioritization process based on resources available

Project resources	Process specifics
Very (and strictly) limited resources in a small project	Necessary minimum of functionality is absolutely required by the end of the project. Prioritization serves to choose those requirements that are crucial for supporting the main goal of the client. (This is often coupled with choices about implementation solutions).
Bigger project where additional resources can be considered	The prioritization serves to choose the highest value requirements for the next iteration.

The observations in subsections 1 and 2 suggest that the interplay between both factors represents a specific project constellation that requires (or pre-determines) the instantiation of the prioritization process.

Throughout the interviews, it became explicit that there is a link between the project's settings and the way the decisions are made, i.e. how the value creation process is organized. In all projects where the client's company was a small company, the decision making was deliberately delegated to the developer. It could be a product owner, a project manager or another representative of the developing team that was responsible for the communication with the client. Furthermore, these observations demonstrate that the context of a project plays a significant role when it comes to the instantiation of the project's processes. It becomes obvious that not all agile projects can be considered to operate in the same way with respect to the value-creation processes and practices used in the project. This opens up further questions that deal with the relation context – project instantiation. As this is a very broad topic, we could not address it in all its facets within the boundary of this thesis. In Chapter 6 we investigate in depth some of the contextual factors and their impact on the value creation process.

2.4.5 Role of the prioritization process

RQ 1.9. Are there any other ways (beyond the selection of requirements) through which the requirements prioritization process adds value to the project?

The use of the prioritization in agile context is not limited to selecting the most important/valuable requirements for the upcoming iteration. Our study revealed two other aspects that are very important for the project's outcome: building the right

product and incorporating new information and learning on-the-fly. Our participants indicated that in a context of volatile or unclearly defined requirements, the prioritization process ensures value by the change management mechanisms and by incorporating learning loops in the process. As a participant observed, "… if we would not have followed this approach, we could have made a completely different system from what they want. Especially in this case where the requirements were not SMART (specific, measurable, attainable, relevant and time-bound.… ." Furthermore, the participants agreed that the agile prioritization process provides a natural way to react to new information and knowledge that become available in the course of the project. This might be a deeper insight about the requirements, or new knowledge about technical feasibility. For instance, in one of the projects in our study the initial planning was made with certain assumptions about functionality provided by open-source libraries. In the course of the project it became clear that these assumptions do not hold and the scope of the project was changed, and, respectively, the requirements were re-prioritized. Another participant reported that the requirements became more clearly defined at later stage in the project.

2.5 Discussion on the results

Our exploratory study of the agile requirements prioritization process yielded a few findings regarding essential aspects of requirements (re)prioritization in agile projects. The interesting about these findings is that they deviate from what agile literature says about these aspects, and thus reveal a new perspective when looking at the agile RP process. Overall, these findings revised parts of our understanding about the roles of the client and developer, and the prioritization criteria used. Additionally, our results suggest that there is a link between project context and RP instantiation. We already addressed the value-based criteria in Chapter 3 and below we discuss the other two aspects.

First, **with respect to the decision-makers**, the evidence from the case study shows that, in contrast to the documented agile best practices in the literature [Cohn 05], in most of the cases the developers are those who made inter-iteration decision making. Our interviewees agreed that more often than not the involvement of the clients consisted mainly of approving the plan/giving comments. Only in few cases practitioners were able to provide evidence that the client is really capable/interested/aware of the agile way of defining priorities, and thus able to navigate the functionality by the mid-course decision-making process.

That the developers strongly participate in the prioritization and decision-making gives us the hint that agile and traditional requirements engineering processes may not be that different, compared to what originally was thought, regarding who prioritizes the requirements. Our case study suggests that in agile (as well in traditional) contexts, we can find examples of clients who essentially rely on developers to prioritize their project requirements; we, therefore, think that the difference between agile and traditional processes is not with respect to who prioritizes the requirements, but: (1) with respect to what competencies and (tacit) knowledge those, who prioritize, have of their client's business, (2) with respect to whether the client is able to participate in the process. Our interviewees suggested that the developers, who 'saved the clients from themselves' are experienced professionals (e.g. in the words of one interviewee, with 10 to 15 years of experience in IT systems delivery in a specific business sector) and this might indicate that for agile prioritization to be led by developers, it should include highly-competent and experienced people. As already noted in the previous section, the developers rely on their own estimations and understandings, even on common sense, in order to maximize the value creation for the client. The situations in which the developers had to 'save the clients from themselves' exemplify that some of the core underlying assumptions about the value creating practices in agile context might not apply to all projects. The observations that we made in the case study make us question some of these assumptions, in particular the assumptions that:

1) The client is responsible for the value creation; the client always participates and provides input about the value of each requirement: the client is on-site, is competent, is able to resolve issues rapidly and provide the developers with timely information about the client's desires and especially about the value created by each requirement;

2) The client is capable of making decisions with respect to value creation; the client is the one taking the final business-related decisions and in particular the prioritization decisions, based on business value;

3) The developer takes a subordinate role with respect to the decision making related to value creation; the developers proceed with the implementation based only on whatever the client desires, i.e. the value-creation for the client is driving the process.

The results of our study suggest that these assumptions are not present in all projects. First, having an on-site client was not always feasible. Whenever it was feasible, we

found that developers may have to work with a client that for various reasons may not be in a position to provide timely information about requirements priorities. The detrimental impact of involving such a client in the requirements prioritization process may vary.

What could be the reason for the inability of the client to perform her/ his role as suggested by the agile literature? In our opinion one possible explanation is that the clients is not aware of his/her needs or can't express them in a way that can drive the developing process. In some cases, notably with small companies, the client doesn't have the resources to get deeply involved and has no alternative but to trust the developer to do a good job. Particularly sharp examples of this are cases where the client might want something that has proven not to work, but is not aware of that, and the developer gives advice based on his/her experience.

Additionally, we think that the clients might not be aware of: 1) their role; 2) the process; 3) the type of input they need to provide; 4) the impact their participation has on the overall project outcome. Last but not least, the client might have no experience whatsoever with IT solutions.

Further in this chapter we address points 2 and 3 from the list above by proposing a conceptual model of the prioritization process that could help clients to become aware of the process and by making explicit the concepts that impact the decision-making.

Furthermore, the observation that clients feel their knowledge of requirements priorities is limited (when it comes to inter-iteration decision-making), opens up a question to those researchers that develop and evaluate requirements prioritization methods. The existing prioritization techniques rest on the assumption that clients are aware of the mechanics behind the application of requirements prioritization techniques and, as a minimum, they are conscious about their role of providers of the input that feeds into these techniques. It appears that our case study findings question the extent to which this assumption is realistic. Indeed, requirements prioritization methods take for granted that there is measurable business value to provide input into the methods. Now, looking at our case study findings, the suspicion grows that these objective values may not always exist and are difficult to make. Our findings are indicative for a limitation being present in the current requirements prioritization methods, and therefore, we think that future research is warranted to understand those cases in which the assumption is not realistic. We propose this topic to the community for future research.

We make the note that in projects where the client's organization is large and is represented in the project by an IT department or by consultants [Larman 08], the reality is quite different from the observations in our case study. The findings that we present in this thesis relate to small and middle-size companies as this was the context of the companies that participated in our case study. However, in the context of large projects and organizations the client is also experienced and competent IT practitioner. This makes us think that the context of the agile project plays a central role with respect to client's participation.

We can conclude from our case study the following points:

First, while an agile software company lets clients prioritize requirements, the requirements decision-making process can take place only when the client's interest to make changes along the way is in balance with the developer's interest for a sustainable business. While in large companies and projects the vendor organization mitigates its own risk by estimating the technical debt, in small and middle companies this could happen by active participation of the developers during the mid-course decision-making.

Second, the prioritization process instantiation varies across projects at different client companies and those variations seem to be linked to project characteristics such as size of project and size of clients' organization. This means that the understanding of the roles in the process of value creation as described in literature might hold only for a sub-set of all agile projects.

An implication of these findings is that that we have to reframe the assumptions that have been taken by default in the literature. Furthermore, the research and practitioners' communities might consider to extend the existing approaches to agile requirements prioritization for value creation. In Chapter 7 (Conclusion and Future work) we composed a number of RQs which investigation would contribute further to better understand the agile prioritization process.

2.6 Threats to validity

The limitations of our case study and the threats to validity has been discussed in Chapter 3 Section 3.7.

3 Conceptual model of the agile prioritization process from clients' perspective

3.1 Motivation

As we saw in the previous section, our case study indicates that different stakeholders might take over the RP, depending on the context. In many agile projects the developers or their representative (e.g. a product owner), are actively involved and more often than not are leading the inter-iteration decision-making process, keeping in mind the value-creation for the clients. That's why we felt motivated to study the decision-making - as perceived by the developers, with client's goals in mind. We think that more clarity is needed in respect to: What do the decision-makers need to consider in order to create more value for the clients / stakeholders? In our opinion, a decision-maker would profit from a clear model of the prioritization process available to him/her. We think that a conceptual model can help the decision-maker (client or proxy) in at least three ways: (i) to navigate through the agile process of delivering business value; (ii) to make explicit the tacit assumptions in different RP methods; (iii) to identify those possible pieces/sources of information important to the outcome of the prioritization and, consequently, to the project. The model and the concepts would help the decision-makers independently from the concrete role they have in a project. And – moreover – we think that it could help to strengthen the position of the client in those projects where the client is less experienced with agile methodologies or with IT projects in general.

3.2 Concepts that appear in the agile prioritization process

With respect to the prioritization process, the process of coding yielded the following seven aspects: *Project Context*, *Prioritization criteria*, *Effort Estimation/ Size Measurement, Learning Experience*, *Input from the developers, Dependencies* and *External Change*. Iteration planning additionally considers *Project Constraints*. Below we explain each of these conceptual categories, and their impact on the RP process.

1. During the case study, we observed that the prioritization process itself varies significantly in terms of participants involved, prioritization criteria applied, purpose and frequency of the prioritization. The interviewees shared that, in their view, the

variation depends to large extent on the context of the project. We represented this variability in the model by the concept ***'Project Context'***. It includes those project settings such as 'size of the project' or 'size of the client's organization', and is used to explicate the impact of these settings on the prioritization process. In the projects of our practitioners, the concrete instantiations of the prioritization processes were deemed to be linked with these contextual settings. For example, our interviewees observed that in projects with similar contexts, the instantiated prioritization processes are similar in respect to who are the *decision-makers* and the *amount of participation of the different* parties in the process.

2. All interviewees agreed on that the *project context* has a significant impact on the '***Prioritization Criteria'.*** We observed also that they all consider the ***Business Value*** the dominating RP criterion, whereby *Business Value* is estimated by the customer alone. In some projects we observed one recurring question being asked at requirements reprioritization time: *"Is a requirement absolutely necessary to support the main usage scenario?"* This question implies a notion of 'damage to the client' or 'negative value to the client' in the case the requirement is not implemented. One study participant said: *"All features that belong to the main usage scenario were considered mandatory and needed to be included in the product. This drove the decision-making process."* In addition to Business Value, the client in some projects considers the ***Risk*** caused by a requirement's implementation.

3. In the experience of the interviewees, the client considers '***Estimated Size'*** based on functional size when making decisions on priorities. The estimation of *Size/ Effort* impacts the value estimation as well. For example, a participant put it this way *"If we give a high estimation for certain requirement (in terms of time / cost), it happens that the client starts considering this requirement as less important as previously thought."* We make the note that size, effort, cost and risk are estimated by the developers and provided to the clients for their decision-making. From the client's perspective, size is a given – though potentially uncertain – input.

4. Another 'building block' in the RP process appeared to be *the developer's perspective* (box '***Input from the Developer***' in Fig. 10). While the literature [Ambler 02] deems the role of the developers for the RP process secondary, the case study revealed a different situation. In the majority of the cases the developers were the more influential party, providing advice and alternative solutions, but also taking into considerations the interests of their own organization (such as 'possible reuse of the

requirement', 'importance of the project for the organization', 'available resources at the moment').

5. The conceptual category ***'External Change'*** stands for those events that happen during the project and impact the company, the business environment or the product under development. Such changes can impact the value of requirements. The interviewees deemed the external changes be one of the reasons for clients' requirements change requests.

6. The category ***'Learning Experiences'*** represents new insights acquired by both the clients and the developers during the project, such as new knowledge about technical solutions, or new insights about the desired functionality of the product under development. They impact the value estimation, the prioritization decisions and the size estimation. For example, while working in a project that we investigated, the developer learned about the exact functionality of open-source software that he intended to use. This new insight triggered changes in the initial estimations and thus in the priorities of the requirements. Learning is an in-built principle in agile development. Harris and Cohn [Harris 06] advise *"Incorporate new learning often, in order to decide what to do next"*.

7. The ***'Project Constraints'*** such as duration, release date, budget, velocity and available resources, impact both the prioritization decisions and the iteration planning.

8. ***'Dependencies'*** between requirements can be of different nature – e.g. chronological or architectural dependencies. Both clients and developers express the dependencies that have to be considered, from their perspective.

9. The ***'Project Backlog'*** means the list with requirements for the projects. Prioritized Project Backlog is the ordered list of requirements, and a sub-set of it (called iteration, and in some agile methods - sprint backlog) is to be implemented in the next iteration. 'Prioritized' means to assign a requirement a priority, which during iteration planning translates into an order of implementation: i.e. starting with the requirements with the highest priority, so many requirements are chosen for the iteration backlog as can be implemented within the next iteration and project constraints.

We make the note that the concepts identified in this sub-section serve as a foundation for the conceptual model of the prioritization process presented in Section 5.3.

3.3 The model

The concepts and the relations between them that we discerned in the previous section, resulted in the following generic model of the RP process.

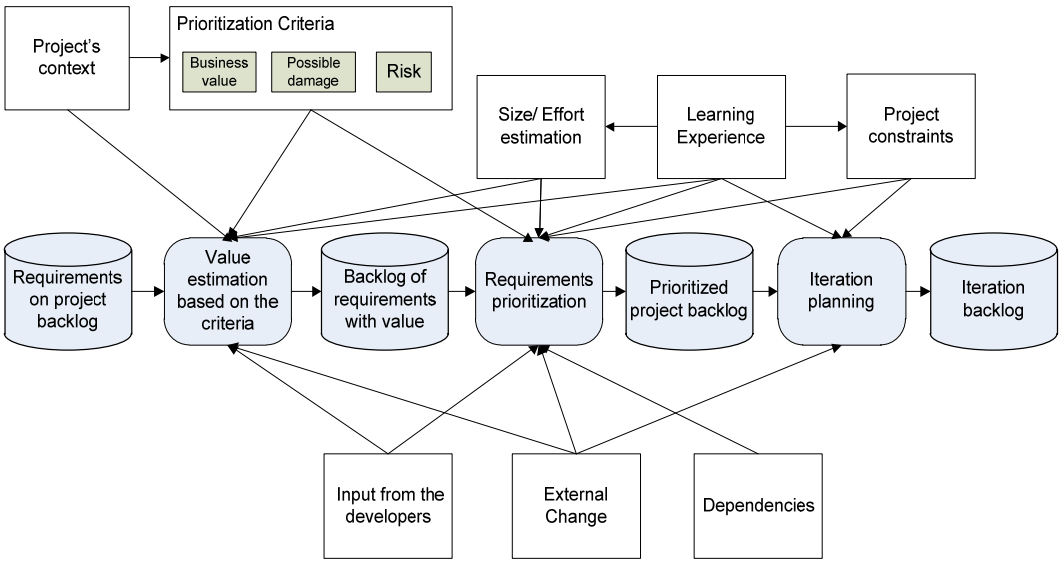

Fig. 10. Conceptual model of the agile prioritization process

Legend:
Arrows mean that the item, towards the arrow points, is impacted from the item where the arrow origins.
Rounded Boxes represent *actions.*
Cylinder Boxes represent *artifacts* (such as value estimation, list with requirements)
Square Boxes represent concepts that impact the main workflow.
The dark boxes and the arrows between them represent the main workflow of the prioritization process.

We make the following notes:

First, we also note that in Fig 10, arrows reflect relationships between the concepts. For example, the 'learning experience' impacts the size/effort estimation. This is so because with the progress of a project the developers learn to better estimate both the amount of work they are able to perform in one iteration, as well the concrete effort (in hours), or the size of a requirement (e.g. in story points). The learning also is about the mapping factor of story points to effort in hours/ days. This leads to more correct estimations for the following iterations. We make the note, however, that we don't claim completeness of the set of relationships.

Second, we traced the concepts back to the interview questions that we asked and the interview answers we collected. We provide an illustration of this process by using the concept 'Negative value' (or damage). This concept originated from two questions: "Which factors played a role during the decision making?" and "Do you use explicit criteria for the prioritization?" The concept was derived based on the following statements of our interviewees: "We considered how big the damage will be if a requirement is not implemented. We call this 'negative value', "Is a requirement absolutely necessary to support the main usage scenario?", and "How angry will the client be if certain feature is missing."

It is important to note that the model does not aim to represent the RP process in terms of a sequence of steps performed by the decisions makers. Why is this so? First, our purpose is to explicate the process by identifying the concepts that are considered during the decision-making by the stakeholders. That is, we want to look under the hook, to understand why certain decisions are made, based on what criteria, and what factors impact them. Merely observing a sequence of steps leading to prioritized list with requirements would not give us a hint about the "why? ". Having said that, it becomes clear that the model is compatible with any RP technique, because we represent the activity "prioritizing requirements" as one box on the model. For the purpose of our research what is in this box (i.e. which technique in particular has been applied in a certain case), is not of interest. As we reasoned in Chapter 2 Section 3 (Background on RP in agile), the RP methods described in literature give prescription on how to obtain a prioritized list with requirements from an initial list. This means that these methods deal with what is in this box. On the contrary, our model does not prescribe any process or propose a new technique, but instead just describes what we found in the case study data. This means that a decision-maker could use this conceptual model as a framework for reasoning about his/her RP process independently of his/her concrete context. Clearly, not all of the elements in the model are necessarily present in each RP process – i.e. some of them depend on the project context. For example, one can use the concepts of the model to depict a specific client's RP situation in a specific project, in a specific organization and, thus, take into account the topics important for clients to consider in RP at inter-iteration time.

We make the note that we don't claim completeness of the model. Completeness still should be validated by new case studies and by collecting new data that could lead to new concepts. In fact, we believe that studying more projects in different contexts

would reveal additional aspects of the RP process, that we could not cover in this study; for example large projects and large organizations.

To conclude, we want to stress that the model takes the perspective of the client, unlike other RP authors [Augustine 05], [Cohn 05], [Gottesdiener-1] adopting the perspective of the developers. This model is to help clients or clients' proxies to 'zoom-in' into the prioritization process and see those concepts which are important to consider in RP at inter-iteration time. It describes what happens in all those RP processes about which we learnt from the participants in the case study. In the model we take a generic perspective of RP, that is, it abstracts from the use of a specific RP approach.

The implication for the researchers is that, in our opinion, the model would help those RE researchers who are interested in carrying out empirical research to investigate how agile requirements decision-making happens in practice, to structure research questions and empirical data.

3.4 Limitations

The conceptual model described in section 4.3.3, as suggested by GT methodologists [Charmaz 07], [Clarke 05], cannot be validated against the data that has been used for the development of the model. We can only evaluate the resulting model against the evaluation criteria of the GT methodologists, which we do in this section. The validation of the model (i.e. how well it describes the phenomenon in question) could be a subject of a follow-up empirical investigations, e.g. by the means of future case studies.

Research methodologists [Strauss 91] emphasize that when a researcher builds up a conceptual model by using a qualitative approach as GT, the researcher should evaluate its emerging model in terms of three key criteria: (i) adequacy, (ii) fitness (or relevance) and (iii) modifiability.

Adequacy of the result of the GT process is to be assured by applying the set of techniques and analytical procedures in the GT. Examples of these techniques are to adhere as closely as possible to the GT principles and processes, to evaluate similarities and dissimilarities among the interview data pieces, and to check for any category, property or property value that might have been overlooked. We made conscious effort to keep these GT principles. The data was coded independently by the author of the thesis and by another researchers. We must be clear on a validity concern arising from the fact that most of the time the researchers worked away from

each other at different locations and could not do much joint coding. However, the individual coding processes led to the same results.

The relevance of the conceptual model to researchers is to be judged regarding how it fits the situation, that is, whether it helps individuals familiar with the phenomenon (in this study, agile RP) to make sense of their experience and to manage the situation better. We took two steps to make sure we preserve the meanings of the clients in agile projects: (1) we consistently engaged ourselves in diagramming activity, (2) we searched and included the so-called 'in-vivo' codes, as recommended in [Charmaz 07]. These are special terms from the world of the practitioners in the studied context, which are assumed that everyone "knows and shares" them, which flag condensed but essential meaning, and which reflect assumptions that frame some actions. In our case, examples of in-vivo codes, associated to clients in agile RE, are "negative value" (meaning the damage in case the requirement is not implemented), and "gold-plating for the client" - meaning that the team is virtually available to leave in any requirement that the client insists on. We looked into the implicit meanings behind these terms, which brought us to the Model on Fig. 10.

Furthermore, modifiability of an emerging theory is concerned with the possibility to update it and extend it in the future. In our study, modifiability means the ability to replace the abstract concepts in the model by more specific terms, for example, when used in a specific project with specific agile method. We made a conscious effort to maintain a balance between keeping the concepts abstract enough - so that the theory can serve as a general explanation, and making sure the concepts do not get too abstract as to lose their sensitizing characteristics. In our view, we should keep our framework open as it makes more sense to invite other researchers to use it and test it, only after this, to strive for all-inclusive and general results. We think that if industrial uptake of agile software development practices increases and more knowledge on the client's role and the client-developer interaction modes becomes available, our framework will need further refinement and extension so that it's kept useful.

We point out that like other qualitative research approaches, the GT approach implies the risk that the researchers assume that the conceptual categories are saturated, when they might not be. Following Charmaz [Charmaz 07], we asked the interviewed practitioners to expose us to many examples of their agile RP practice, while carrying out the theoretical sampling. This increased our understanding of the empirical world and helped us discern variations in the conceptual categories we use

to describe the agile RP from client's perspective. We checked our concepts against the empirical realities of these practitioners and this was instrumental to understand how, when, and why the meanings of our categories vary. We stopped our re-coding when we noticed that further continuing of our work on the data did not bring new ideas nor opened up new ways to think of the properties of our conceptual categories. In GT, this state is called 'saturation' of the resulting conceptual model [Charmaz 07].

Last, it is important to discuss the external validity of the model. External validity deals with the question whether our findings are shared by other interviewees too, in future case studies. More generally, this is the question is this model shared by anyone involved as a decision maker in agile software development. In chapter 6 we address this question and make a first attempt at validation of generalizability (usability and utility) of the conceptual framework.

In the following Section 4 we use the model presented in this section, to identify the gap between the guidance that the literature provides on the topic of agile RP, and the practice, captured in the model.

4 Identifying the gap between practice and literature

As explained in the introduction of this chapter, this section sets out to answer the following research question RQ 1.10. **Which concepts of agile prioritization are shared in practice and in literature and how they are used to provide guidance for prioritization?** We answer it by mapping existing agile prioritization techniques to the findings from the case study. We performed this mapping study after creating the conceptual model. In it we map the conceptual categories, that appear in the model, and the existing techniques from literature, as presented in Chapter 2 Section 3. The goal of this study is to identify which concepts of agile prioritization are shared in practice and in literature, and to understand if there is a gap between the guidance for prioritization that literature provides to practitioners, and the prioritization process as observed in a case study. The result is meant to help to: (i) map different techniques and concepts to each other; (ii) analyze the level of guidance the different method descriptions provide to practitioners (in terms of those concepts that are explicitly used), and to (iii) be used as a framework for structuring the discussion about requirement priorities in an agile project and thus lead to explicit

and better motivated requirements choices. The results under (i) and (ii) can be used to analyze the current situation of the literature guidance and thus draw conclusions about possible improvements of the RP practice in terms of better method descriptions that could help the stakeholders in a project, and in particular the clients.

4.1 Mapping of the existing agile prioritization methods on the model

In Chapter 2 Section 3 - Related work, we identified from the literature 22 prioritization techniques that are being used in agile context. Here we don't provide motivation for the choice of the literature and references to the sources where the techniques are described, as this has been already discussed in Chapter 2.

In this section we perform a mapping between the conceptual categories of the model in Fig. 10, and their presence in the existing prioritization methods. By means of this mapping, we will see which of the conceptual categories (that we discerned in the case study and that constitute the model) are in fact used by other authors and techniques. The mapping is performed in panel with 3 experts with at last 17 years of experience in RE and project management – Andrea Herrmann, Luigi Buglione and Maya Daneva. The result is presented in Table 12. Therein, the first column presents the 22 RP techniques of Table 3. The other columns are named after the categories from the conceptual model in Fig. 10. A row in the table is to indicate those concepts that a particular technique supports and does this up to a certain extent, i.e. the concept appears explicitly in the description of the technique. In the table, we populate the cells with the symbols 'y' to mean that RP method corresponding to that row uses the concept corresponding to that column. Furthermore, in addition to the concepts that appear in the model, we have added an additional column S in Table 12 to acknowledge that a description of a method indicates the use of tacit knowledge in the requirements prioritization. In this column, we place in the symbol 'x' to mean those methods where we identified that the decisions are made based on implicit, subjective opinion of the decision-maker ("intuitive prioritization"). We make the note that the empty cells in Table 12 mean that we could not find explicit indication about the use of the concept. For example, the second row is about the method *Ping Pong Balls*. From the description we discern that this technique uses value and risk as prioritization criteria ('y' in the first cell), the context of suitability of the methods is described ('y' in the second cell), and cost is considered as well. We proceeded analogically with all methods and concepts. We make the note that most of the techniques are not described in the literature in great detail. Further, they don't discuss

explicitly what concepts drive the prioritization decision. For example, the 'Round-the-group' prioritization, and the 'Ping Pong Balls', take the subjective judgment of each participant as an input into the decision-making process, without discussing why each participant estimates one requirement (or feature) to be of higher priority than another. The majority of the descriptions of the techniques are focused on the steps that transform an initial list of requirements into a prioritized list, i.e. in which order they shall be executed, and say almost nothing about the considerations used to determine the priority order itself. For example, Gottesdiener [Gottesdiener-1] says about the Pair-wise analysis: *"You successively rank requirements by comparing them in pairs until the top requirements emerge at the top of the stack."*

Table 12: Mapping between the concepts from the model in Fig. 10 and the prioritization methods from Table 3.

Concept → Prioritization method	Intuitive prioritizatio	Prioritizati on criteria.	Project context	Size/effort estimation	Input from developers	Learning	external changes	Project constraints	Dependenci es	Project backlog	value of requiremen	Prioritized project	Iteration/ Sprint
Round-the-group prioritizati on	x		y						**y**				
Ping Pong Balls	x		y	y									
$100 allocation cumulative voting		y											
Multi-voting system	x	y	y										
MoSCoW	x	y											
Pair-wise analysis		y								y			
Weighted criteria analysis		y											
Analytic Hierarchy Process (AHP)		y		y									
Dot voting	x	y								y			
Binary Search Tree		y											

Concept → Prioritization method	Intuitive prioritizatio	Prioritization criteria	Project context	Size/effort estimation	Input from developers	Learning	external changes	Project constraints	Dependencies	Project backlog	value of requiremen	Prioritized project	Iteration/ Sprint
Ranking based on product definition		y			y					y	Y		Y
XP: Planning Game/ Poker		y		y	y	y		Y			Y		Y
Quality function deployment (QFD)		y	y							y	Y	y	Y
Wieger´s matrix approach		y	y	y	y			Y		y	Y		
Mathematical programming techniques		y		y				Y			Y		Y
Technique of bucketing requirements	x	y											Y
Kano Model		y											
Relative weighting		y											
Larman		y											
Theme screening / scoring		y		y									
FDD		y								y		y	

4.2 Discussion

Our observations in Table 12 confirm the finding discussed in Chapter 2, namely that the descriptions of RP techniques from the agile RE literature use mainly coarse-grained concepts. This becomes obvious when looking at Table 12, as it was possible to populate only part of the cells in the table. This means that our conceptual model is at a finer level of detail compared to the levels that the authors of the 22 techniques considered when describing their approaches. Moreover, our conceptual model

reveals that in practice there are many more concepts that impact the prioritization decisions than those concepts that literature describes. Also, only few methods among the 22 that we investigated and that were described in literature, explicitly take the client's perspective – those are the Kano model and the QFD. In fact, literature treats requirements reprioritization very superficially and often does not give a complete cook book recipe. For example, although it is always emphasized that learning and context are important [Harris 06] in agile process, no method describes how they should be considered.

Furthermore, our analysis shows that there are almost no methods described in the literature that explicitly state the criteria on which the decisions are based and the influence of the context. Nor there is any indication about who is or should be involved in the decision-making process. We think that a possible reason for this finding could be the nature of the agile decision-making itself, where the team is empowered and self-organized and where team members' tacit knowledge plays a significant role. Further, our observations indicate that some of the methods don't strive for perfection in the sense that their authors mean them to be universally useful. Instead, these methods are just 'good enough' for certain application contexts. Wiegers [Wiegers 99] is one of the very few who explicitly states the criteria used and that these criteria he uses in his approach are not the only one that play a role during prioritization. For this reason he warns practitioners that the scheme he proposes should not be considered as the only method to set priorities. Moreover, he advises to use this approach to decide about 'negotiable' features only, i.e. the ones that are not in the top-priority category. The core features shall be included anyway.

Another reason for the low level of detail of the methods described in literature might be that the practitioners who are authors of the compared methods consider that it requires only common sense to execute the prioritization, and they trust the team to do it right without much guidance.

Furthermore, in Table 12 we observe that:

- Learning is treated explicitly only by Extreme Programming (XP). This observation is surprising, given the fact that many authors deem the explicit use of learning between two iterations the main advantage of the agile paradigm [Cohn 05][Harris 06] we assume this is incorporated rather implicitly in the methods, by means of their iterative nature and frequent decision-making cycles.

- External change – although an important aspect in agile development, is not mentioned even once. It seems that the published methods do not discuss how external changes influence reprioritization. Our gut feeling is that it is included implicitly in the implementation of the processes because in the case study we found that this is a tacit consideration which the developers do take into account.

4.3 Implications for Research and Practice

In this section we outline the implications of the results of the mapping between literature and practice, presented in Sections 4.2 above.

Firstly, we point out that Table 12 represents: (i) a new knowledge as it makes explicit the gap between the descriptions in the literature and the process as experienced by practitioners in real life projects; (ii) it can be eventually used as a framework to structure a deliberate decision-making process by providing the concepts that can be used to frame the discussions. The concepts of our model can serve as objects of the decisions to be made and could be the topic of a meeting. As per [Webler 95], 'deliberation' implies equality among the participants, and an orientation towards resolving conflicts in consensual way. In its core, this is the nature and the goal of the agile prioritization. McDaniels and Small [McDaniels 03] plead for consensus-building that would lead to making decisions on requirements priorities. As per [McDaniels 03], a deliberative process rests on a common understanding of the issues based on the joint learning experience of the decision makers with respect to systematic (e.g. explicit) and anecdotal (e.g. tacit) knowledge. Example of such a process is the communicative process that promotes rational value disputes [Rippe 99]. The decision-making on priorities is governed by establishing rules of a rational discourse, a specific form of a dialogue in which the stakeholders that make the decisions have equal rights and duties to present their claims and test their validity. These rules also define the role and relevance of both systematic and anecdotal knowledge for making choices.

The implication for practice is that the mapping between the methods and the concepts allow for a better-motivated and explicit rational-discourse-based process, that includes the concepts from the model.
The implications for the research community is that more research is needed in order to understand: (i) do the practitioners need more detailed guidance about the decision-making process, and if so – for which methods / decision-makers, and

project contexts, and (ii) how the assumptions behind the RP techniques (quantitative and discourse-based) shape the outcomes of the decision-making and which technique is better in which agile context.

5 Conclusion

In this chapter we made three contributions: (1) discussed the results of an empirical study of the agile requirements prioritization process and investigated the concepts that are important to consider when practitioners work on (re)prioritizing agile requirements at inter-iteration time; (2) presented a conceptual model which describes on an abstract, generic level, the concepts that seem to impact the agile prioritization process in terms of concepts that impact the decision-making and the relations between them, and (3) provided the reader with a mapping between the concepts of the model and the RP techniques as described in literature. We used the model to map different literature sources, methods and terminologies to each other, by identifying the use of the concepts from the model in the methods from literature. The mapping table that we obtained gives us a clear understanding of the 'deviation' between the existing methods as prescribed in literature and the process we observe in real life.

These results contribute to the body of knowledge on agile RE in the following ways:

1) By explicating the RP process as it happens in real-life projects;

2) By outlining perspectives for improvements of the state of the practice.

The answers to the RQs and the model can be used twofold: (1) to help researchers and practitioners to identify best practices and distill some guidelines for practitioners, and (2) to serve as a roadmap for further empirical research to investigate the fitness of different RP approaches to different contexts.

The conceptual model presented in this study can help the client in agile projects in multiple ways: (i) as the model makes the process explicit, the client can navigate through it in order to optimize and influence the delivering of business value; (ii) the clients can become aware of the assumptions that are used tacitly in different RP methods; (iii) the model helps to identify those possible pieces and sources of information important to the instantiation of the prioritization process and the outcome of the prioritization and, consequently, to the project, and (iv) the model

offers a big-picture view of the diversity of the prioritization process and this can help each practitioners to map the reality of her/his own project to the model and thus to be able to reason about his/her own context.

Furthermore, our study indicates the presence of relations between concepts that impact the prioritization decision-making and that might not be visible to the decision-makers in the agile projects, and particularly to the client. Those relations are not part of any explicit process description and refer to: (i) the variety of prioritization criteria and the motivation to choose a specific criterion as the prioritization criterion in a specific project, (ii) the important role of the developers and their own criteria when proposing requirements' priorities. Because the agile community postulated that the client is the key decision-maker, it seems logical to conclude that the client should be aware of these relations and their impact during the decision-making if he/she wants to run an effective requirements prioritization process. If the client makes assumptions about the process, it might happen that the real project situation is quite different and it's of benefit to the client to explicitly address these assumptions and the real process. Moreover, the client needs to have visibility about the real process, even if these issues are related to the developer (or the developer's process).

In Chapter 5 we provide an illustration of how the findings, presented in this chapter, can be used by the stakeholders to support the decision-making in an agile project.

Chapter 5

Empirical Evaluation

As already outlined in the Introduction, in the context of this thesis we understand the research in a sense of a process of acquiring knowledge. The previous chapters were dedicated to describe in detail our contribution as an increment to the globally available knowledge on the topic of agile RE. In this chapter we focus on the question of the usefulness and usability of the acquired knowledge for the research and practitioners' communities. We conducted an evaluation study, using the research method of technical action research, with the purpose to illustrate how the acquired knowledge can be used in other projects and to analyze its usability and relevance.

1 Introduction

As already stated in the Introduction of this thesis (Chapter 1), the main contribution of our work consists in acquiring better understanding about the studied phenomenon (that is, the agile value creation and in particular – the reprioritization process at inter-iteration time) beyond what is already known in the agile literature. The nature of our study called for assessing the validity of our findings at each step. (See the validity sections in chapters 3 to 5.) Naturally, the question about the usefulness of the knowledge arises, in other words - is there utility component to it. We asked ourselves: How can the insights, gained in the course of our study, be of further help for the practitioners? To answer this question we undertook another

study by applying the research method of technical action research. The goal of this evaluation study is to illustrate the usefulness of the acquired knowledge for the research and practitioners' communities by demonstrating how the acquired knowledge can be used in academic and industrial projects to support the decision-making about requirements in a project.

We want to make the note that by explicating knowledge we already have made a contribution towards better understanding of the studied phenomenon by: (1) identifying concepts that impact the decision-making process, (2) identifying causal relationships between concepts; (3) identifying relations between the instantiation of the process and the context of a project. These results in themselves aid the stakeholders to gain higher awareness and understanding about the process of decision making and the factors impacting the value creation, and eventually to take deliberate steps towards improving them. Now we go one step further and address the issue of the usefulness of the knowledge. In this chapter we provide an example of the application of our findings in another academic or industry project.

We want to underline that the validation at this point can only be considered very preliminary. For use in actual projects performing a larger-scale evaluation study is required.

The rest of this chapter explains the Technical Action Research (TAR) as a research method, describes the researched case, and evaluates the results.

2 Research Method

2.1 Technical Action research as a research method

Technical Action research (TAR) is a validation method, used in design science. It starts with an artifact and tests it under conditions of practice by solving concrete problems with it [Wieringa 12]. The core of the method, as the authors explain, is that the researcher plays three roles, namely of artifact developer, artifact investigator, and client helper. We chose this validation method, because it is particularly suitable for testing the applicability of artifacts, e.g. conceptual models, to real-life problems. As Wieringa and Morali underline: "The aim of this way of using action research in design science is to bridge the gap between the idealizations made when designing the artifact and the concrete conditions of practice that occur in real-world problems." The TAR is especially intended to test the relevance of artifacts.

These characteristics of TAR make it the method of choice in our case, as we start with the artifact – in our case the prioritization model from Chapter 4, and validate its relevance, usability and usefulness for other projects. This validation method has been successfully used by other authors in similar settings [Morali 10].

In the sections below we demonstrate how we applied the TAR.

2.2 Research questions of the evaluation study

In this study we were driven by the following research question:

RQ 2. How can the findings, gained in studying Research Question (RQ 1), be applied to other agile projects and how can they be used by practitioners and researchers? In other words: In which way can the knowledge, presented in the Chapters 3 and 4 and the conceptual model, presented in Chapter 4, be used in a real-life project?

This question is deliberately formulated at higher level of granularity in order to help the reader to grasp the purpose of the study. Following the Nested problem-solving approach of Wieringa [Wieringa 09], we refine the RQ 2. into the following sub-questions:

> **RQ 2.1. Is the model from Chapter 4 usable?** Here we mean: is the model understandable by the stakeholders? Can it be used in another software engineering context?
>
> **RQ 2.2. Is the model useful and for what purposes?** Our working hypothesis is that a stakeholder – in our case the project manager (in other cases this could be a product owner or product manager), could use the model to identify in a systematic way his priorities.
>
> This way we could judge the relevance of the model for the practice: the stakeholder could find what is relevant in his particular case, and which aspects he should consider, in order to determine his priorities.
>
> **RQ 2.3. Which parts of the knowledge that we have explicated has been used in another project, and for what purposes?**

3 Problem Investigation

3.1 The project – stakeholders and goals

At this point we want to underline that so far in this thesis we presented observations pertaining to different projects where, understandingly, the context varies. The limited resources that we had allowed us to choose only one project where we could run a technical action-research study. We investigate a small project that involved three Master's Students at Eindhoven University of Technology and a Project Manager from the University of Twente. The team was self-organized. The project manager fulfilled also the role of a client. He was responsible for defining the functionality and the scope of the project. Being as well the client, he was interested in an efficient work, i.e. – in extracting maximum value within the resources of the project. The project was with fixed duration of 3x 140 person-hours, plus the effort of the project manager that he estimated to be at average 3 hours per week.

3.1.1 Goals

The application that was under development is called "The personal Chief Security Officer" (pCSO). It aims at creating and supporting a risk management process for consumers of Facebook. Risk management is the "coordinated activities to direct and control an organization with regard to risk" [ISO 31000]. The goal includes creating of On-line Privacy Controls that:

- Force people to think about their goals, possible threats and mitigations;
- Motivate people to spend time on privacy management by:

  - Investigating assets and their threats (likelihood x impact = risks)
  - Taking deliberate decisions with respect to the risks, i.e.
    - Accept (do nothing)
    - Transfer (buy insurance)
    - Mitigate (put controls)
    - Avoid (discontinue activity)

- Motivate people to adopt it.

Furthermore, the application should be easy to use.
The whole process is iterative and goal driven, and involves tradeoffs (plan-do-check-act).

3.2 The initial process

3.2.1 Schedule of the project

The project had two phases. The first phase lasted from September till December 2010, the second was from December 2010 till June 2011.

The process of the initial phase was ad-hoc. There were weekly meetings. The project manager gave general directions without interfering much. This didn't contradict with the goal of this phase – to have an open-end software that worked, without aiming at concrete set of functionality.

The second phase, however, required better focus on achieving a specific goal. As the resources became limited (the number of hours remaining), the project manager faced the problem of optimal use of the remaining time to reach more concrete results. The question arose what methods and practices could be used to "get there".

This was the motivation of the project leader to consider and explore the implementation of a set of agile practices in the second phase of the project. This is where we joined the project.

3.2.2 Similarities with industry projects

Although the developers were master students and the project took place in an academic environment, it shared the following commonalities with real-life projects:

- Foremost, a working application was expected by the end of the project. Otherwise the students would not get the full credits for this project.
- The resources were limited, i.e. it would not be possible to add more developers to the team in case of need, nor it was possible to extend significantly the amount of time they spend on the project.
- The deadline was sharp and non-negotiable.
- The requirements and the functionality were not clear from the beginning and were subject to changes and clarification during the project. Here we mean that they would not only change, but it was not clearly known and defined what the product should do, as the functionality emerged during the development.

4 Our implementation of TAR

In the action case, the role of the researcher (namely the author of this thesis) consisted in providing guidance with respect to selected aspects of the development process and in particular - the requirements engineering activities.

The researcher served as an external consultant for the project, discussed issues with the project manager, evaluated problems and suggested possible solutions for improvement. The suggestions were evaluated and decisions about the actions that will be undertaken have been decided upon. These steps were repeated during the course of the project. For the purpose of evaluating the results of the research, an interview with the project manager was performed at the end of the project. It consisted of open-end in-depth questions.

We performed the following steps:

- First, we observed the current process and discussed the experienced problems or improvement desires with the project manager.
- At the next step we analyzed the problems and considered a *treatment* for improvement .
- Third, we planned with the project manager possible improvement actions. This corresponds to the step *agree on implementation plan*.
- Next, the stakeholders applied the improvement suggestions to the project.
- Last, we analyzed the changes and evaluated the effects of the *treatment* on the project.

This cycle was performed three times in the period January – April 2011.

The process of all three cycles was similar: first, the researcher and the project manager (PM) analyzed the current state of the project, the challenges and issues that are being faced at that stage, and discussed possible solutions. Then, the PM implemented the solutions that pertained to the project management, updated the requirements backlog, and informed the team members about changes in the requirements, and the assignments to each team member. Then, the project continued with the implemented changes for about a month, when the next cycle started. We want to make the note that it is difficult to draw a clear cut border between the different phases within a cycle for at least two reasons: (i) the

improvements couldn't be implemented all at once, and (ii) time was required for the solution proposals to start working and to observe their impact.

Although the three cycles were similar in their structure, the topics that have been discussed and addressed, differ.

The first cycle started in January 2010 with meetings between the researcher and the PM. First, we observed and captured the initial situation at the time of joining the project. This happened during two discussions with the project manager, each lasting about 60 minutes. During the meetings the researcher became familiar with the goals of the project, the process that have been deployed so far, and the vision of the PM for the product development. The following project documents were analyzed as well:

TO_DO list;
Risk Tree;
Development Roadmap.

As a result of the meetings, a list with problems that have been experienced at that stage, has been created, and possible improvement suggestions have been discussed. In the meetings participated the PM and the researcher. The other team members – i.e. the developers, learned about the changes, discussed at the meetings, from the shared project documentation. This process was chosen for two reasons: (1) because the developers were not responsible for decisions except at implementation and technical level, and (2) the team was not co-located. That is, it could cause increased coordination effort to organize meetings for all team members.

During the observation and analysis phases the results of the previous solution actions were evaluated, and new were proposed. Between the meetings, e-mail discussions between the PM and the researcher have been used to clarify things. The PM was then responsible for updating the project documents based on the outcome of the discussions, and the update versions were shared with the project team members.

We make the note that, according to the schedule, the project continued beyond the three cycles that we have observed and supported. We draw the line of our observations at the end of the third cycle because the PM was satisfied with the way the project functions and felt that he can continue managing it this way till the end (two months later).

4.1 Evaluation of the initial process

As a first step, we analyzed the RE process at the stage when we joined the project. This happened by interviewing the PM, comparing the current practices, applied in the project, to agile best practices, by using our knowledge in the area of agile RE. We make the note that some of the problems were identified as such by the PM, and others were suggested by the researcher as possible areas for improvement. Jointly, the problems were related to the following aspects of the requirements engineering process: *identification of requirements*, *prioritization of requirements*, *choice of requirements to be implemented in the next iteration*, and the *requirements management*. Two of these problems were related to requirements prioritization, the other two were more general requirements engineering problems. We discuss all four of them to give accurate picture of the project context.

Our first observation was that the requirements engineering process was rather based on the gut feeling of the project manager.

1. **Requirements identification and elicitation:** The project manager was responsible for the identification of the requirements. The process, however, didn't follow a structured approach. Instead, the process was ad hoc, and was driven by the rough idea that the project manager had about the overall purpose of the product.

2. **Requirements prioritization:** At the time when we joined the project the purpose was to implement the basic functionality. For this reason not much attention was paid to the process of requirements prioritization and the order in which the functionality should be implemented. No explicit prioritization criteria were applied.

3. **Choice of requirements for the next iteration:** There was no structures approach for the choice of requirements. The developers implemented them in the order they considered meaningful to achieve basic architecture and functions. There was no clear cut between iterations.

4. **Requirements management**: Requirements' descriptions were kept mixed together in the same document with other project information such as implementation planning and technical details. Although the team worked in well-coordinated manner, there was no prioritized list with tasks to be worked on at each point in time.

4.2 Treatment design

We now describe how we mitigated the problems listed above. The main focus was on the process of requirements selection and prioritization. The treatment (i.e. our advice how to improve the current situation), emerged in a discussion with the project manager, and it was based on application of the conceptual model of requirements prioritization from Chapter 4 to the project's context. This application consisted in a mapping between the concepts of the model and the project settings. We used the model applying the following procedure: we traversed the concepts and at each step asked the questions: *Is this concept relevant for our project, i.e. does it impact the value and thus – the priority of a feature? What happened and what did we learn in-between after the last decision-making, that could impact or change the priorities?*

The mapping was done by the researcher (the author of this thesis), in a discussion with the project manager, in the following manner: We walked through all concepts one after another and identified the importance of each concept for this particular project. Next, we traced all links from one concept to another, and analyzed a possible impact on other concepts. This procedure was performed at the beginning of our involvement in the project for the following purposes:

1. To investigate if the concepts of the model make sense for a new project and can they be easily observed;
2. To structure the following discussions with the PM that aimed at familiarizing him with the agile prioritization process;
3. To make the PM aware of the concepts that might impact the decision-making, and to identify possible improvement areas in his project.

Below we present the results of this mapping, and illustrate how exactly we used the concepts of the model for structuring the discussion and analyzing the concrete project settings.

4.2.1 Results of the mapping between the project and the conceptual model

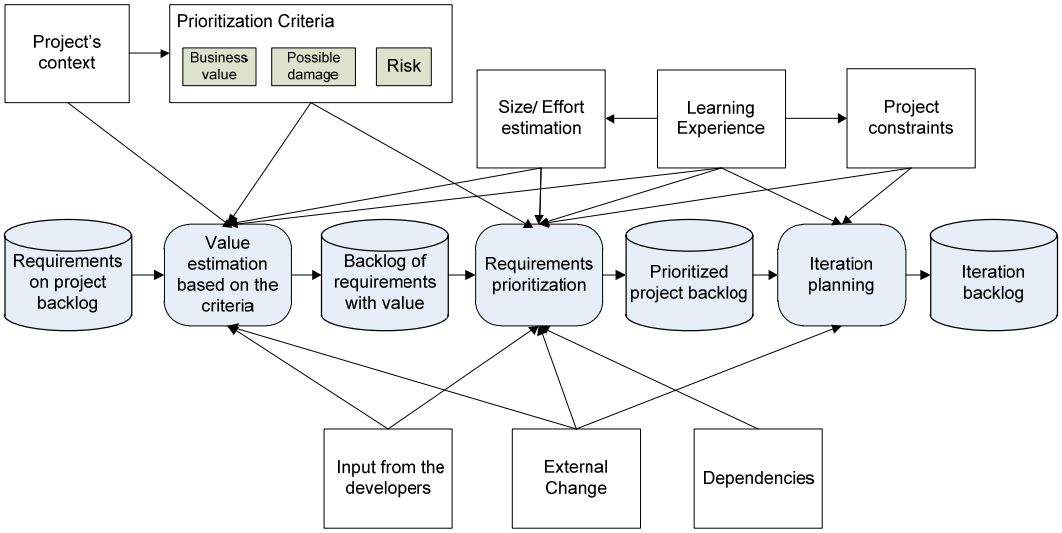

Fig 10, borrowed from Chapter 4

In Table 13 below we present at a glance the results of the mapping in terms of the concepts of the model and whether they were observed in the project or not.

Table 13: Concepts of the model, observed in the project

Concept from the model	Observed in the project or not
Prioritization criteria	yes
Learning experience	yes
Dependencies between requirements	no
External Changes	no
Size/ Effort estimation	yes
Input from the developers	no
Project constraints	yes

Next, we provide details on the mapping.

First, we considered the possible prioritization criteria.

1. *Business Value and possible damage* in case a feature is not implemented (also called *Negative value* by some participants in the case study), have been considered as prioritization criteria. We observed as well, that the potential damage is not an issue at this phase of the project. We explain this observation with the nature of the product – completely new solution that aims at providing novel, innovative functionality to potential users, that is – people with a Facebook account that want to improve their online security. According to the PM, the potential damage to the users in case when certain feature is not implemented, cannot be estimated upfront. On the opposite - all features were considered from the perspective of the value they add. As it was difficult to estimate the value of the single features, the decisions were driven mainly by one consideration: *do we absolutely want to have this functionality in the product before the end of the project*. The *risk to finish on time* was also used as a prioritization criterion, especially during the later iterations of the project. In the words of the PM, "…*remaining time* was the criterion what will be in and what not…".

   To wrap up, the following main criteria were driving the decision-making process:

   - How much a requirement contributes to the project goals.
   - How precise a requirement is. Under 'precision' we understand here is it possible for the developers to understand a requirement and to implement it within one iteration. (If it was not precise enough, it was refined.)
   - Resources (Remaining time.)

2. *Learning Experience* appeared in the project in the form of two usability tests with early users. The results of the tests provided insights about both functionality and usability issues. This led to the inclusion of new requirements on the list and to a change in the list with priorities.

3. We identified that *External Changes* were not relevant for this project for two reasons: (i) the type of functionality, developed during the project, is not related to market or economic situation, as it was completely new; and (ii) the project was relatively short and during that time it was unlikely that other, competitive products, would appear. However, we acknowledged that possible

changes in privacy policy of Facebook might have impacted the business case and the functionality of our product. In this respect, the presence of this concept helped to reason not only about the current situation, but also to think of possible scenarios.

4. *The Project constraints* in terms of labor resources and schedule played an essential role during the decision making. We discussed this also under the *Risk* as prioritization criterion.

5. At the beginning of the second phase of the project (January 2011) we considered including *Size/ Effort estimation* on per-User Story basis as an integral part of the requirements and project documents. However, we were confronted with two practical problems: Firstly, it was very difficult to make any estimations because the developers were not very experienced, we didn't have any historical data to work with, and neither the PM nor the team had experience with similar functionality. Secondly, we couldn't predict the impact of the introduction of such estimates on the team moral. As we worked with students, we didn't want to "scare them, or exercise pressure, as we don't pay them", as the PM said. Because of this, in his opinion, any size estimation would have been useless, as the tasks took as much time as they took.

6. The *input from the developers* in this case didn't play a role for decision-making process for the following reasons: (i) they were less experienced than the PM in any software development respect, and (ii) they didn't have decision power, except for some technical solutions. This observation aligns with one of the projects from the case study described in chapter IV, that shared a similar context. Here we want to make note about an observation that we found interesting to share: the decision-maker (in this case – the PM), had to consider the *skills* of the developers as a decision criterion. For example, it turned out that the developers don't possess the skills needed to implement some of the features. This required that the PM finds an innovative solution. The work-around that was found was that the developers will refactor the code in such a way that the client can make changes to some data himself later-on.

7. *Dependencies* didn't appear to be an issue for the this project. We explain this with the relative simple and restricted functionality.

What we learnt is: it is possible to observe the concepts in a new project. Furthermore, these concepts make sense also for external observers, in this case the

PM. We will discuss this finding in more depth later in Section 5, where we answer the research questions.

To wrap up, we list the advices that we proposed and that were implemented in the project, that is the *treatment* in the terminology of [Wieringa 12].

1. With respect to requirements identification and prioritization, we applied a structured and explicit approach for organizing the discussions about the requirements.

2. The requirements were prioritized on regular basis, using the concepts from the model to decide on per-week basis on the set of requirements (iteration backlog).

3. Some of the User stories were difficult to estimate. Such stories with unknown scope and effect were split into 2 parts – 1) research story – to learn about the story, and 2) the actual implementation. The purpose of this practice was to reduce the uncertainty and the risk, both technical and in terms of time resource needed.

   Additionally, with respect to the overall requirements engineering process, we proposed and implemented a User-Driven approach to identify relevant product features. In particular, we performed user-tests to obtain input from target users about the Goals and Threats they estimate to be of high and of less high significance.

4.3 Results

In this section we present the state of the project after the application of our advice to the project. We judge about the changes that occurred based on an interview with the project manager that took place in June 2011. The nature of the interview was open-end semi structured.

According to the project manager, the following changes were made during the course of the project.

1. The choice of requirements became explicitly driven by three criteria:

- How much a requirement contributes to the goals.
- How precise a requirement is. (If it was not precise enough, it was refined.
- Resources (Remaining time.)

As the PM put it: 'The goals were defined and then operationalized.'

2. The learning during the project occurred by means of incorporating the results from brainstorming sessions, feedback from discussions with colleagues, and the results of two tests with users. "These discussions led to the decision to include also positive risks [note: in terms of the product – vocabulary under *Risk* here we understand mitigating risks that Facebook-user might be exposed to] in the scope of the product. This decision impacted the schedule and complicated the architecture and the dependencies."

3. The implemented requirements management changes led to clearer implementation order. "The new set of project and requirements documents and the requirements' organization … provided structure." The PM sees additional, positive side effect: the new requirements organization and the way requirements are decided about had positive impact on the team members as well. They knew what is expected of them at all times. The definitions of 'done' was set. The requirements were divided into business USs and technical USs. This helped to clarify who is responsible for what. The developers had to focus on the technical USs, while the project manager sees the goal all the time and specifies the priorities based on it.

4. With respect to the final result the opinion of the project manager is that it is better than it would have been while working in the old way. However, we acknowledge that this is a subjective opinion, and it cannot be compared or measured to a case without intervention from the researcher. In our opinion, the BV creation was positively impacted because the goals became clearer and the choice of requirements-to-be-implemented within the boundaries of the project resources were driven by clear criteria that were known to the whole team.

5. Last but not least, the changes in the requirement management process gave the PM explicit control over all processes: "There was better and more control over the process, the direction and the status, at all time." The PM became confident that he had the instruments to steer the value-creation process and finish within project boundaries with a satisfying functionality. Additionally, he realized that he had to re-think what to ask from the team members. 'As our resources are limited, the new way how the requirements were organized provided clear view on the resources and the remaining time. We focused on the functionality that in our opinion was most useful and desirable for the potential users.'

6. We make the note that some of the suggestions that we made with respect to the requirements engineering practices impacted as well some project management practices. In particular, the change in the tool support resulted in new organization of the project documents. They were spit into the following categories – tasks for current iteration, responsibilities, backlog, steps (reflecting high-level functionality and goals of the product), refined requirements (lower level requirements, divided into functional and non-functional), time administration (for time-tracking purposes), 'done' (list with completed tasks). Another document was dedicated to capture the test results and observations, and other product- relevant data. Requirements backlog, was re-organized. The technical details were separated from the decisions on the USs at business level.

5 Answering the Research Questions

Below we aggregate the findings into answers of the refined research questions that we posed in Section 2.2. of this Chapter.

RQ 2.1. Is the model from Chapter 4 usable? Is the model understandable by the stakeholders? Can it be used in a software engineering context?

In order to answer this question, our goal was to observe whether the model is usable in other projects, different from the ones described in the multiple case-study. In particular, it was important to analyze (i) is the model and its application understandable by the stakeholders, and (ii) can the model be used to analyze the decision-making situation in real-life software engineering context. In order to do that, we first analyzed the new project while searching for the concepts, that appear on the model, and second, tried to map those concepts on to the TAR case. We presented the results of the findings and the mapping in section 4 above.

It is important to note that the main source of the information, beside the project's documentation, were the interviews and the discussions with the project manager. The fact that he could understand without special knowledge the meaning of the concepts, as well as the way we apply the model, gave us an indication that the model is understandable by other people, i.e. by the stakeholders in a project. This answers the first part of the RQ 2.1.

Furthermore, our results show that the researcher and the project manager could identify the concepts of the model in a new project, and that mapping results in meaningful observations. An important finding is that some of the concept were relevant for the project that we investigated, while others were only of scamp significance. Summarizing our observations we find that the concepts/aspects: *prioritization criteria, project constraints, learning* were most relevant. On the other hand, less relevant were the *external changes*. We explain this with the nature of the product. As a new and innovative application, the user experience is paramount to understand what would help the user to solve her problems and to identify those features that are most valuable (learning). We make the note that in this specific case we couldn't apply *the potential damage* as a main prioritization criterion because the nature of the application and the state of the product didn't allow for determining such potential damage in case a requirement is not implemented. However, given the project's context, we explicitly analyzed the prioritization situation from this perspective.

RQ 2.2. Is the model useful and for what purposes?

Our working hypothesis was that stakeholder – in our case the project manager (in other cases this could be a product owner or product manager), could use the model to identify in a systematic way his priorities and could receive indications about which aspects to consider, in order to determine his priorities.

The results presented in section 4 suggest that the model can be used to support the decision-making process in agile projects in at least three ways:

> 1) to analyze the context of a project and the possible impact of different factors;
>
> 2) the model is an useful instrument to structure the discussions between the stakeholders (with respect to requirements decisions) by keeping the focus on the goal;
>
> 3) the concepts of the model help the decision-makers to identify those aspects of their project that are relevant for the value creation in each concrete case. That is, we can use them as a checklist and make sure that nothing will be omitted by the decision-makers.

These observations show that the model is useful in at least one case, as it could be used to support the decision-making process in one of the three ways listed above. This is an indication about the relevance of the model for this case.

RQ 2.3. Which parts of the knowledge that we have explicated has been used in another project, and for what purposes?

Here we mean the results of our study, described in Chapters 3 and 4. Those aspects are:

1) the categories of the conceptual model, which was elaborated earlier in this chapter, in particular the prioritization criteria.
2) the knowledge about the relation between the availability of resources and the prioritization criteria.
3) the awareness about the relation between the type of the project and the role and the abilities of the decision-makers. Our observations overlap with the other academic project that was included in the case study (see Chapter 3 and 4) – namely that in this case the PM makes the decisions. This observation suggests that the conclusions might be generalizable to other similar projects.

The findings listed above show that understanding the link between the contextual factors and the instantiation of the process is essential as it was used most, next to the concepts of the model that helped to structure the DM process.

We observed also that the knowledge was used to better organize the requirements engineering process. In particular it was applied to improve the requirements elicitation and management, the identification of useful features, and to achieve more objective decision-making by stating explicit criteria. Our results suggest that: (i) the decision-maker became more aware of different aspects of the context that could possibly affect the project; (ii) these aspects were taken explicitly and actively into consideration during the discussions, and (iii) the decision-making process becomes more focused and oriented towards the goals.

6 Limitations and future work

In this section we discuss the limitations of our study and reason about generalization to other projects.

Although we can not draw quantitative conclusions based on only one case, we argue that the observations and results from the TAR case are insightful and indicative for the applicability of the thesis' findings to broader context. We address the following threats to validity: **construct validity, conclusion validity and external validity.**

1) **Construct validity:** it deals with the question *Are the results independent of the observer?* With respect to the construct validity we analyze in what possible ways could the researcher be biased when making observations and drawing conclusions.

One significant threat to validity originated from in the fact that the researcher (and author of the model) is the one that use it in the technical action research case as well. This does not tell us whether that other people can use it as well. Still, we observe that the project manager had no difficulty to understand what was meant, and could map easily the terms to the aspects of his project.

Other possible limitations are listed below:

- The result of the mapping between the aspects of the project and the categories of the model could be potentially biased by the view of the researcher. As the mapping was performed by the researcher only, we have not validated the results by an external observer. However it is our understanding that the mapping reflects the projects' reality because at each step we went back to the description of each category and asked ourselves the question: Do we observe the same thing in the current project?

- The conclusion about the usefulness of the model as a means to structure the decision making process about requirements might also potentially be biased. We admit that the opinion of the PM might be considered to be subjective. However, we assume that his professional qualities and experience allow him to assess correctly the impact of the application of the model on the project. Furthermore, both the researcher and the PM made the same observations with respect to this question.

- The mapping and the illustration of how a discussion within a project can be structured around the model, can be regarded as an analytical exercise that, in the observed case, provided a positive answer to the question: *Is such mapping possible?* This means that the results of the study can be considered as an illustration, and one possible way of applying the findings of the thesis.

2) **Conclusion validity** (observation validity)

The question that has to be answered is: *How are we going to manage the risk that we draw the wrong conclusions (observations) from our data?*

We may have drawn wrong conclusions and the changes in the decision-making process and in the requirement engineering process might be a result from other influences. However in our opinion this threat is minimal because of the following considerations:

- There have been no other changes in the projects' settings and environment during the period of the TAR case.
- The project duration was quite short which makes unlikely that other factors like increased experience or extensive learning could have affected the project.

However, we want to underline that this was a real-life case, that didn't take place in an idealized conditions. As the researcher provided advice regarding more general aspects of the RE process of the project, it is possible that other changes in the RE process have impacted the aspects that we analyze – i.e. the decision-making process about the requirements.

3) **External validity** (generalizability)

The external validity deals with the question of applicability of the results to other cases. We can not argue that our results are universal and applicable to any other case. However, we think that similar projects might (i) implement the findings in a similar way as explained in this illustrative case, and (ii) might be affected in a similar way by the introduced changes. Under 'similar' here we understand a project that shares characteristics like size, experience of the team, limitation of resources.

Furthermore, we think that the results could be applicable to other kinds of project where explicit criteria for choice of features/requirements are needed.

In our opinion, there is no risk of generalizing to the wrong cases as all agile projects share the understanding of iterative development, continuous prioritization and accommodation of changes during the development.

At this point we want to address the possible threat to validity of the results that might originate in the academic nature of the project. We admit that at first glance this might relativize the results when transferred to real-life projects. However, in our opinion, the possible threat to validity in this respect is minimal. Although the project took place in an academic environment, it shares the characteristics of industrial projects, as discussed in Section 3.2. For this reason we claim that it can serve for our illustration purposes.

Last but not least, we want to address the applicability of the results to other stakeholders in a project. In the TAR case it was the project manager who benefited from the results. Further work must show whether our conclusions are generalizable to all decision-makers in a project, regardless of the role they take in a concrete case. Those are the clients, but also representatives of the developers such as project managers and product owners. Indirectly, all stakeholders of a project will benefit indirectly from the improved process and product. We cannot see reasons why not, but empirical evidence is needed to support such a generalizability claim.

7 Conclusion

In this chapter we presented a technical action research study aimed at illustrating the usefulness and usability of the knowledge, distilled and presented in this thesis, to other projects.

The results suggest that:

1) Although our conceptual framework does not contain any causal generalizations, it is useful because we demonstrated that the knowledge, explicated in the conceptual framework, helped in the studied case. It was used to find the relevant factors in the project and the project manager was able to identify the mechanisms in place that work in his project, that is - the aspects that he should consider in this individual case.

When reasoning by analogy, we can conclude that such identification could happen in similar projects as well.

2) The conceptual model of the agile prioritization and re-prioritization process makes sense for other practitioners because it can be used as an instrument for structuring the discussions on decision making.

Project participants and the project's outcome can benefit from being aware of the concepts that impact the process and the relations between them.

We make the note that we might not have uncovered all possible ways in which the results of the thesis could be useful. However, we consider the question of usefulness as being positively answered.

The results listed in the previous section bring us to the following observations:

1. The prioritization, based on clear and explicit criteria, was perceived as a powerful means for achieving the project goals. It helped structure the decisions about requirements.

2. The criteria linked the importance of a requirement to its value for achieving the project goals. Although the value of the single product features could not be assessed in measurable (countable) entities, the consideration of value reframed the discussions about the requirements.

3. The concepts from the model helped to make the decision-making process explicit and structured. Although not all concepts appeared to be relevant for the project, discussing them helped the PM to consider factors that he might have otherwise omitted. For example, the impact of possible external changes.

4. We observed a causal relation between the factor "experience of the developers" and the decisions that PM had to make.

Although we could stop at this point with the discussion of the case, we would like to mention one additional result of the study: new questions arose from this study that could motivate further research and would complement our results.

Such questions are:

- How can Business Value be considered and tracked during a project?

- Can single requirements be linked to (a part of) the Business Value?

- Is there an observable relation between the context of the project and the prioritization decisions? Such observations would serve to gather a set of observations that could lead to formulation of hypothesis about causal relationship or to identify new relations.

In chapter 6 (Conclusion and Future work) we address these questions in more detail.

Chapter 6

Conclusion and Future Work

In this chapter we summarize the contributions of this dissertation in relation to the research questions discussed in Chapter 1. Furthermore, we highlight open issues and offer directions for future research on the subject of value creation in agile projects.

1 Conclusions

As today's business world is very dynamic and unpredictable, a high level of flexibility is required during the product development. This means that the goals that have been identified at the beginning of the product development, can be re-defined during the development process. A re-definition of business objectives after change in the project context such as change in the business environment, laws, or competition can occur as well. Each of these events might trigger a change in the business goals and consequently – in the defined objectives for the software. These phenomena are the primary motivation for the application of the agile approaches, and in particular – for their light-weight nature and the ability to respond to changes. The idea of focusing on business value is pivotal in the agile paradigm, yet in which way this value is created seems to evade precise description.

The research goal of this thesis is to gain deeper understanding about the agile value creation process. We performed a number of research steps to explore some of the current agile practices that seem to contribute to the value creation, and thus to distill knowledge that the agile practitioners apply and that might help to improve the agile practice.

The dissertation asked two main research questions:

RQ 1. What concepts and relationships between them characterize the value creation in agile projects? and

RQ 2. How can the findings, gained in studying Research Question (RQ1), be applied to other agile projects and how can they be used by practitioners and researchers?

The answers of these questions were found by means of literature reviews and case studies.

We can cluster our findings in two main groups:

1. Insights about the notion "Business Value", and
2. Better understanding of the requirements prioritization process as a value creation practice.

With respect to the term "Business Value" we discovered that:

- The majority of papers in agile software engineering literature do not define explicitly the concept of business value in an elaborated way.

- The business value concepts rest on the definitions of *Earned Value*, *Net Present Value* or *Return on Investment,* as used in economic sciences.

- Practitioners offer definitions which translate business value into dollar value. However, we found that this 'translation' is problematic as there is no study that suggests how this could be done in practice.

- Business value is more than just numbers. It comes out of a human judgment that is based on competencies and deep knowledge of the client's domain. The existence of objective values to feed as input into the decision making process in agile projects and in particular in the prioritization methods, is questionable; instead, what is priority seems to be a complex combination of subjective value-based criteria.

- There is a *difference in the perception of value for the developers and clients*, and the value creation *for the developers* impacts the decision-making process. This explicates the importance of the both perspectives (vendors and clients), and the need of balance between the value consideration of these two groups stakeholders.

- We have identified a set of practices that the authors of the publications link with value creation.

We have increased the body of knowledge on agile requirements prioritization by making the following contributions:

- We presented a conceptual model of the on agile requirements prioritization process, which describes on an abstract, generic level, the concepts that seem to impact the process in terms of concepts that impact the decision-making and the relations between them. Thus we explicated the RP process as it happens in real-life projects.

- We provided a mapping between the concepts of the model and the RP techniques as described in literature. We used the model to map different literature sources, methods and terminologies to each other, by identifying the use of the concepts from the model in the methods from literature. The mapping table that we obtained gives us a clear understanding of the 'deviation' between the existing methods as prescribed in literature and the process we observe in real life.

- We analyzed the agile decision-making situation and draw a parallel to the real-option analysis method. In doing so, we provide one possible way to explain how to consider the decision-making situation that the stakeholders face during the agile inter-iteration requirements prioritization and reprioritization process – namely, in terms of options.

We started our journey with a literature study on the topic of *business value* and its creation in agile project. The study gave indications that BV is created by a set of agile practices. However, we found no publication in the literature which suggests how exactly certain agile practices and groups of practices add more value and even what "more value" means in agile context. This motivated an additional study on the agile practices with the purpose to identify those practices that impact the value creation.

The key distinguishing feature of the agile methods is re-prioritization, based on an assessment of business value that is changing over time. The idea that re-

prioritization is driven by calculating a cost function can be discarded as overly simplistic; it is evident that non-trivial decision making is involved. The key question, then, is how this decision making takes place. To understand this, we performed a series of case studies to identify the concepts that impact the prioritization decision-making. We acknowledge that there are other agile practices and activities that contribute to the value creation, beyond the practice of continuous requirements re-prioritization. However, our motivation to look at prioritization as value creating activity originates in its focus on the clients' perspective and its relation to the clients' goals.

Our observations in the case study brought us to the conclusion that business value is created by a combination of agile practices and mechanisms at play in a project-specific context. For example, in short projects with limited resources and a short list with requirements, the client profits from the agile process through (1) the efficiency of the process, (2) the 'savings' made by the light-weight method, and (3) the ability to figure out early what they'll get and whether it is what they need. This profit-making mechanism is deemed by our participants important to obtain the best possible system for the money spent.

Further, we studied in detail the agile prioritization process and identified the criteria, used in the decision-making process, and relations between the project context and the instantiation of the process.

A client would gain a maximal benefit of an agile approach in a context of volatile or unclearly defined requirements. In such cases, the agile paradigm ensures the creation of value by the change management mechanisms and by incorporating learning loops in the process, which represent the agile requirements management process, coupled with the prioritization as a decision-making vehicle under uncertainty. To better understand the decision-making situation under uncertainty, we drew a parallel between the agile decision-making situation and the Real-Options theory. We argued that the agile mid-course decision-making situation can be regarded as decisions about options, analogically to the real-options approach.

As a result of these observations we formulated lessons learnt that could help other practitioners and clients in agile projects: (1) to reason about their project context, and (2) to make optimal choices to improve the decision-making.

Finally, we illustrated one possible way of how practitioners could apply our findings to their own project context. For this purpose we performed an action research study.

In the next section we outline possible topics for future investigation, that we distilled during our research.

2 Future Work

Value creation in agile projects is a complex topic. Many forces play a role in it. In this work we provided a closer look at some of the concepts and practices that impact it. We identified concepts that seem to add to the value creation, and we reasoned about the mechanisms behind the phenomenon.

Still, the results presented in this work a only a small step towards understanding the agile value-creation practices. Our observation shows that the knowledge is incomplete and fragmented and there is a need to further pursue and extend the current understandings about the agile process in general, and about the value-creating activities in particular.

Below we provide a list with topics and questions that, in our opinion, are worthwhile to investigate by the research community.

2.1 Quality of the User Stories as agile requirements

Although the agile methods are spreading fast in the industry and are getting increasing attention from the research community, the topic of the quality of agile requirements engineering artifacts – the user stories, seems to be under-researched. Advocates of agile methodologies claim that they reduce waste by implementing only those requirements that bring value for the customers. For user stories to help reduce waste, they should satisfy six quality criteria, according to the INVEST principle [Cohn 04], the most important one being that user stories should be Valuable (to customers or users). The literature sources on agile methodologies recommend that ideally, the value should be stated explicitly on the story card. Our research, however, revealed that business value evades explicit specification: we found that business value more often than not is present only implicitly or assumed as a tacit knowledge of those involved in agile requirements engineering. In our opinion, there is need for

further empirical study motivated by the goal to get a deeper understanding of how the agile requirements, used in practice, specify business value and how agile project teams members (both customers and developers) reason about customers' value and customers' value-creation. We suggest that the agile requirements artifacts be analyzed with respect to: (i) format, (ii) content, and (iii) level of granularity of the documented business value statement. The results of such study would help identify and distill good practices for writing useful user stories that do support the value creation process. Well written user stories with explicit value statements could help to identify unnecessary requirements and thus reduce waste for the customers. Furthermore, they can help for value-driven decisions-making on requirements priorities at inter-iteration time.

2.2 Dependencies

In Chapter 4 we have identified Dependencies as one of the factors impacting the prioritization decisions. In the case study the practitioners identified the dependencies as one of the prioritization criteria. Although the guidelines for writing good agile requirement documents (user stories) say that the stories should be independent, in practices this turns out to be unrealistic for the majority of projects. Most of the prioritization methods that we list in chapter 2 usually do not take into account dependencies among requirements. Those methods which do acknowledge for dependencies are the ones which describe requirements on several levels of granularity (e.g. AHP). We propose that dependencies are considered multi-dimensionally – they might be chronological from developer's and client's perspective, as well as architectural in nature from developer's point of view. We have made a difference between architectural and chronological dependencies, as we consider both clients' and developers' perspectives. A chronological dependency from clients' point of view could be different from chronological dependency from developers' perspective, as the first one is based on business reasons– and –processes, while the second one has technical reasons and might require completely different sequence of implementing features. Furthermore, an architectural dependency should not necessarily be of purely chronological character – e.g. architectural decisions about the development platform will influence further implementation of features or quality attributes.

We are aware of the existence of dependencies and the huge impact they have, especially in large projects. Studying them and how practitioners deal with dependencies forms a cluster of research questions in our agenda for future

investigation. For the sake of simplicity, we'll not consider at this stage dependencies among features, but we are aware of their existence. This will form a cluster of research questions in our agenda for future investigation.

2.3 Implicit assumptions

In our opinion, an important cluster of research questions that can be furthered is related to the assumptions made implicitly in agile projects. Our results suggest that the descriptions made in the agile literature often take certain project constellations to be the rule and don't further question them nor they provide advice on how to deal in those cases that deviate from the rule. One such example is the observation that we made in the case study, that a small client very often can't dedicate an employee to serve as on-site client in a project. We think that further investigation is needed to create a check-list that could help to identify what the specific conditions of a concrete project are and choose the best practices according to it, instead of following pre-conceived and tacit assumptions.

2.4 Identifying Project Context

The arguments in Section 2.3 lead us directly to the next topic for future research, namely the identification and description of the project conditions in agile. There is a need to question and re-think the general understanding and assumptions about the universality of agile processes across contexts. As already discussed in Chapter 4, the context of a projects determines the concrete instantiation of the decision-making process with respect to the participants and the decision criteria. A means for explicit description of the context can contribute to the identification of sets of projects across which certain conclusions/ observations/ practices can be generalized. Additionally, the question arises: In which project contexts are we likely to observe that certain assumptions are realistic?

In order to be able to successfully match her or his context to the concepts identified in our study and to the practices that we distilled, a practitioner has to identify the context first. There is a need to lay out the context, not in data terms, but in relationship terms – causal relationships between the concepts and the context. Changes in the contextual integrity need to be identified and the set of practices - re-assessed. In other words, the assumptions about particular context should be scrutinized and the questions should be asked: what happens if it changes? We are aware of the importance of this question. For this reason we propose as one

important goal for future work to be the creating of a definition and a simple framework that could be used for comparing contexts and context matching.

2.5 Identifying value of requirements

As already discussed in the Chapter 5, there is still no method that allows to take into consideration the value of each requirement during the decision-making process. Such method would require a means (1) to link the value of a feature/ requirement to the overall value of a product; and (2) to calculate the value in monetary expression. We acknowledge that in real-life projects this might be difficult or even unfeasible to operationalize these ideas. We made some initial suggestions towards simplifying the real-life case that could help to address the problem. For example, we propose to make the assumption that (1) the Reqs R1, R2, …Rn are independent from each other for the purpose of the BV estimation. Still, the use of this approach in a concrete RP method, opens a lot of questions such as: i) what level of granularity of the Rs should be adopted, ii) how the non-functional requirements can be decomposed and represented. How will each requirement be taken into consideration/represented during the decision-making process?

2.6 Grasping the nature of Business Value

A cluster of research questions for the future deal with the nature of the term Business Value. We saw that Business Value is more than just numbers. It comes out of a human judgment that is based on competencies and deep knowledge of the client's domain. An interesting question then is how a judgment about business value is formed and what tacit knowledge should be made explicit, so that knowledge about business value gets shared among developers and clients.

Perceived business value varies from project to project, as projects vary in terms of amount of agile practices they use. Does any relationship exist between the extent to which a project is agile (i.e. the amount and the combination of agile practices) and the perceived business value? If so, what kind of relationship is it?

2.7 Value creation from developers perspective

The case study revealed that the value-creation process plays an important role for the developers' organization. The agile practitioners' literature [Aurum 07], [Boehm 81] seems to share the opinion that the only value-creating considerations that drive

the development decisions are those of creating value for the *client*. During this study we made the consistent observation that, more often than not, the value creation for the developers has been considered as well. We need to consider more carefully in which ways development teams balance the client's business value with their own organizational bottom-line. We think this is an important topic for future research on its own right.

A next step could be to identify techniques that possibly suggest adaptations in the agile requirements prioritization process, so that both clients' and developers' organizations profit. For example, extensions that ensure a better understanding for the rationale behind the decisions and enhance the value-creation throughout the project.

2.8 Applying existent methods for value creation to agile projects

In the economics and management studies there are methods that make explicit how business value is created. For example, Deming's Value Chains [Deming 86] explains how business value from different parts of the organization creates a value chain. He defines Transfer Functions that transform e.g. low level value, prefabrics, into value on another level.
We believe that similar processes can be observed and analyzed in agile software development as well, and consider this as an interesting topic for future work on value creation and value measurement in agile context.

To conclude, more insights are needed on extracting and disseminating good practice both with respect to clients and developers, about the application of the practices in context, and on proposing improvements that could overcome some of the drawbacks of the agile practices, in particular the high importance of tacit, implicit and subjective knowledge. Possible improvements could be method extensions that ensure a better understanding for the rationale behind the decisions and enhance the value-creation throughout the project.

Appendices

Appendix 1

List of abbreviations

APM agile project management

ARP agile requirement prioritization

ASD agile software development

BV business value

DM decision making

NPV net present value

ROA real option analysis

ROI return on investment

RP requirements prioritization

SD software development

US user story

XP eXtreme Programming

Appendix 2

Real-Options Thinking and Agile Mid-course Decision-Making

Preliminary study

In this Appendix we analyze the fitness of the application of Real-Options thinking to the decision making situation that the stakeholders in agile projects face at inter-iteration time. This is a preliminary study on mid-course decision making in agile context. We argue that the prioritization decisions can be regarded as options and provide examples from the literature and from our case study. Our overall objective is to understand how real options thinking fits into the agile requirement engineering process from the perspective of the client. Our conclusion is that there is a parallel between two decision-making situations – the decisions that the stakeholders in an agile project face at inter-iteration time, and the capital investment decisions for which real options theory has been applied. In our opinion, the agile decision-makers could contribute to higher value creation in a project by regarding sets of requirements as options and by explicitly considering and comparing options.

1 Introduction

In this thesis our focus is on understanding the business value creation for the clients in agile projects. We were looking for means to explain the value creation process during a project and to make it more explicit. As we saw in chapter 3, the value creation, among others, depends on the decisions about requirements, taken in the course of the project. For this reason we first investigated the process of dynamic requirements prioritization as a specific decision making instrument pertaining to agile projects. Next, we looked at methods and techniques, used for supporting decision-making in an uncertain context, and that have found application in other domains and disciplines. Our purpose is to provide the stakeholders in an agile project with an explanation of their decision-making situation. We think that this will help the stakeholders to understand and reason about the choices they face. For this reason, in this chapter we set out to investigate the fitness between the real-options

theory as a method for modeling decision-making under uncertainty, and the decisions about the set of requirements that will be implemented at each iteration of an agile projects.

This analysis is from the perspective of the decision-makers. We investigate what kinds of options can be formulated in the context of the inter-iteration decision-making in agile projects. Furthermore, we analysed the published experiences and the experience of the practitioners that we have interviewed in the case study, to determine whether options have been considered.

First, we identify the options that exist for the agile software project decisions. Then, we analyze how practitioners can incorporate options thinking into their decision-making processes.

We want to underline that this analysis is not about applying a new class of mathematical models. Instead, we look at it as a way of re-framing the discussion about client's spending and investment decisions in terms of options.

Below we explain our motivation for investigating the relation between the two decision-making situations – the decisions that the stakeholders in an agile project face at inter-iteration time, and the capital investment decisions for which real options theory has been originally applied.

2 Motivation

Many software development projects nowadays face uncertain business or market conditions. Especially in recent years the rapid development of technologies makes it difficult to anticipate the business success of a software product. The customer faces uncertainty with respect to market acceptance, competition, business environment, new devices or even legislation. For projects in this particular context of the customers, Agile techniques are recognized as most suitable [Erdogmus 05]. More specifically, the inter-iteration decision making on requirements priorities in agile projects is meant to account for organizational uncertainties.

Indeed, the agile practices of dealing with changes during the projects, reacting to new information and insights, and incorporating learning, would bring most benefit to those projects that face a dynamic and changing environment. The decision makers in such projects have to make decisions at each step of the project (i.e. – at each

iteration and between iterations), based on the current level of knowledge and insights that the stakeholders have about the project and the economic environment. At the same time, it is difficult or impossible to predict the future context of a project. Moreover, the decisions about the continuation of an agile project could be viewed as investment decisions. The clients in an agile projects, (which are the investors in this case), have the right to decide how to proceed with their investment at each step of the project. Here we make the note that the results of the case study that we presented in Chapter 4 suggest that the developers take a decisive part in the inter-iteration decision-making in some of the agile projects. For the analysis in this chapter it is not of importance who performs the decisions about requirements prioritization, as long as it happens with the clients' interests in mind. The decision-maker might be a product owner, a scrum master or another client proxy.

Although agile methods is a way to make decisions under uncertainty, there is no explicit explanation of how this happens, that is – how exactly uncertainty is acknowledged and managed during a project. For this reason we looked closer at decision-making methods used in other fields where decisions about investments in uncertain and dynamic environments need to be made. As we saw in Chapter 2 Section 3 - Related work on decision-making under uncertainty, there are number of methods that have been proposed for such contexts. One of these methods is the real options analysis [Amram 99] (ROA).

We think it is worthwhile exploring the use of ROA-concept as an explanatory decision-making vehicle in agile projects because:

- It comprehends uncertainty and it responds to the dynamics inherent in agile project context.
- It allows for incremental expenditures while focusing on the critical pieces of software functionality essential to accomplish the project mission.
- It rests on the understanding that not all requirements and architecture design options are of equal value.
- It supports the clients in the context of a spectrum of possibilities rather than in the context of a single or three (the best, likely or worst case) discrete set-ups, and it facilitates reprioritization as client's realities unfold over time.

We make the note that when thinking of decisions as options, we don't look at the totality of all decisions related to project management, that need to be made during a

project. We restrict our analysis to the decisions about requirements only, and in particular – about the set of requirements that the client considers for implementation at each step of the project. Furthermore, we don't present a full – blown application of the ROA method to agile projects. That is, we look only at the decisions about requirements as comparison between sets of requirements, representing options. We do not use mathematical modeling of the options. That's why we talk here about Real-Options *thinking /or approach* instead of Real Options Analysis.

3 What is ROA?

Real Option Analysis [Luehrman 98] [Amram 03] is first known as a decision support technique in the area of capital investments. A real option is the right, but not the obligation, to undertake some business decision, typically the option to make a capital investment. Real options capture the value of managerial flexibility to adapt decisions in response to unexpected market developments.

The real options method applies financial options theory to quantify the value of management flexibility in a context of uncertainty. If used as a conceptual tool, it allows management to characterize and communicate the strategic value of an investment project.

In this context, the concept of 'real' means adapting mathematical models used to evaluate financial options to more-tangible investments. Since 1999, this concept has found its way into the area of appraising IT investments [Amram 99],[Childs 98]. The core of the ROA for IT assets consists of: (i) the identification and the assessment of optional components in a project, and (ii) the selection and the application of a mathematical model for valuing financial options that serves to quantify the current value of choosing these components for inclusion at a later time. Optional components are project parts that can either be pushed ahead or pulled out at a later point in time when new information becomes available to the decision-makers. The option, therefore, is the right but not the obligation to spend a budget or put resources on a project.

When transferred to our context – namely reasoning about decisions in agile projects, we observe strong parallels. Each agile project consists of iterations and at the end of each iteration a piece of working software is being developed. Eventually, the client in a project will be satisfied with the current functionality and can terminate the project.

From the point of view of the client, each upcoming iteration poses the question: what is the best possible investment that we can make at this stage of the project. This question includes both the possibilities to choose between different requirements to be developed during the next iteration, or to decide to terminate the project.

Clearly, the first step in re-orienting our way of looking at agile projects is to identify the options that exist for the software project decisions. Only then it will be possible for practitioners to incorporate options thinking into their decision-making processes. Thus, in this chapter the discussion is focused on point (i) above, namely: the identification and the assessment of optional components in a project and formulating them as options in agile context. To this end, we are set out to answer the following research question (RQ): **What kinds of clients' options can be formulated for the context of agile projects? What kinds of options have been most discussed in the published experiences and in the experience of the practitioners?** We make the note that point (ii) is outside the scope of this chapter and of the thesis.

4 Research process

In order to assess the fitness of ROA for agile decision-making, we applied two-phased research approach: a structured literature review [Armitage 08] [Fink 05] and a case study [Yin 04]. First, we looked into the experiences published previously in the agile software literature, to collect and analyze examples of how developers and clients reason together on those requirements which create the most business value. Second, we collected and compared the experiences from the agile practitioners in our case study. We make the note that here we refer to the same multiple case study that we discuss in Chapters 3 and 4, and as explained in Chapter 1 Section 5 – Research Methodology.

5 Options and agile software projects

Other authors have already addressed the intrinsic similarity between the way agile projects function, and real-options theory [Erdogmus 02],[Erdogmus 05]. We build upon their observations and extend them. Erdogmus [Erdogmus 05] demonstrates

that agile methods are especially appropriate for projects that happen in uncertain contexts (for example those mentioned in the Introduction).with high uncertainty. Such contexts imply the use of decision-making approaches, e.g. ROA, decision theory [Pratt 95], or robustness analysis [Rosenhead 01] that are able to account for the various uncertainties imposed on the project in a customer organization . In the software engineering literature, there is awareness of the use of ROA in the agile software development [Erdogmus 02],[Poppendiek 06],[Williams 07]. The works of Erdogmus and Williams [Erdogmus 02],[[Williams 07], demonstrate applications of options-thinking to projects that use the XP agile development method. These authors put forward two XP value propositions, namely, that (i) 'delaying the implementation of a fuzzy feature creates more value than implementing the feature now' and that (ii) 'small investments and frequent releases create more value than large investments and mega-releases' [Williams 07, p.13]. Furthermore, the authors of [Poppendiek 06] describe the options associated with delaying decisions and illustrate their points by using the Microsoft strategy (circa 1988). Matts at al. [Matts 07] has put option thinking at work in XP and Scrum contexts. An important aspect in the analyses by these authors [Erdogmus 02],[Matts 07] is their focus on the viewpoint of the developers. The client's role and decision-making process has received only marginal attention. As in this thesis we explicitly consider the clients' perspective, our work can be regarded as complementary to the results of these authors.

6 Definition of options in agile context

How do we formulate options in an agile context? For the purpose of this work, under the term "option" we understand *a set of features that is considered for implementation in an iteration*. In this definition, we do not make a difference between functionality, quality of the product or documentation requirements. Each piece of work that the client requires from the developers has an impact on the resources spent (e.g. budget, time), and thus on the outcome of the project. In our opinion this definition reflects the real situation in an agile project. The stakeholders can choose between different sets of features, and try to maximize the value at each step – i.e. to choose this set that, as a whole, represents the highest value. What remains important is to consider a dynamic decision-making process, typically taking part in the beginning of each iteration when clients compare alternative options (i.e. sets of features) and select the one to be implemented within the iteration that follows.

As stated in the Chapter 4 of this thesis, when regarding a project from a client's perspective, what we see is a sequence of iterations. Each one can be considered as a mini-project, representing a relatively independent and closed entity. At the beginning, it is a prioritized list with requirements and, at the end, it is working software, with certain features and qualities. The value, created during each iteration, can be viewed as the compound amount of the value of the requirements (e.g. features) that have been implemented during that iteration. As we suppose that the goal of a project is to maximize the overall business value, the choice of requirements at each iteration is paramount and their value should be maximized.

We make the note that we are aware of the fact that maximizing the BV at each iteration might not necessarily lead to a maximum possible BV at the end of a project, as this could lead to local optimizations only. However, we don't investigate this topic here and regard it as future work.

7 Reasoning about the value of an option

When reasoning about different options it should be possible to compare the options in terms of the value that each options brings to the project. For this reason in this section we illustrate a possible calculation of the value of an option by offering one possible way for presenting the value. In real-life projects precise estimations are difficult to make. For this reason, the purpose of this section is to help the reader understand how reasoning about options and comparison between them could be implemented by stakeholders.

First, we assume that the business value of each feature can be estimated. We are aware of the fact that such estimation or, even more - a precise calculation of the value of a feature (or a requirement) as a portion of the final value of a product, is not feasible for the majority of the project. However, for the purposes of the analysis in this chapter we make the simplifying assumption that each feature or requirement has a value that can be defined in some entities. In our opinion this would help the reader to understand the way we consider the value of the options further during the discussion.

As we don't aim at applying a new class of mathematical models, we illustrate our analysis by using one possible way to calculate the Business Value of a feature. We build upon the Earned Business Value (EBV) [Rawsthorne 06] measure that was

suggested in 2006. This measure was presented as an answer to criticism to agile methods as too developer-centric and providing difficult-to-understand feedback to businesses [Rawsthorne 06]. EBV is a project management vehicle that measures "how done we are from a business perspective" – namely the percentage of the known business value that is coded up and running. The method considers the total BV of the final product to equal to 100%) and then calculates the value of each requirement/ feature as a portion of this total value. Usually, the work-breakdown structure (WBS) of the project is used as a representation of the project with all its features.

We use this measurement technique because it aligns with our purpose to estimate, compare and reason about value of options. Furthermore, (i) it is a customer-oriented technique, not technical one; (ii) its calculation uses *stories* as smallest units of value; they are visible from both the inside and outside the project and represent the interface between the client and the developer; (iii) while suggested as a PM tool from developer's side, it can easily be adapted to serve clients' estimations as well; (iv) it uses weights to the WBS legs and stories, that are assigned by the business or client.

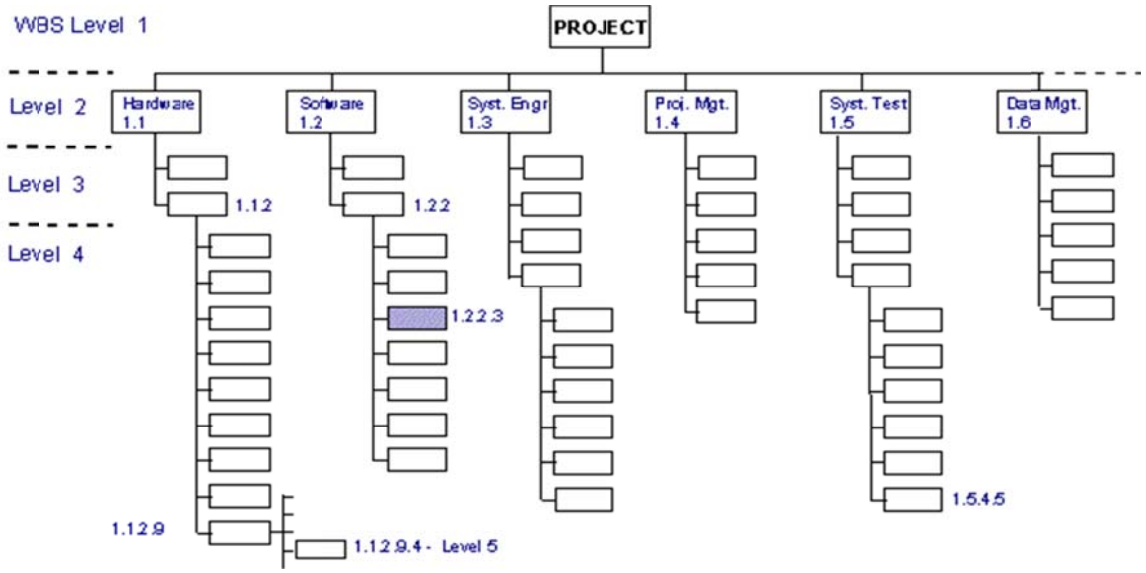

Fig. 11: Example of a work breakdown structure [source http://www.hyperthot.com/pm_wbs1.gif]

For calculating the value of the options that the client has, we use the part of EBV measurement technique, which measures the BV of each feature. We call it the BV technique. The BV technique includes the following calculation steps: First, the client assigns additive weights to the buckets of the WBS, which represent features or other organizations of stories. (In Fig. 11. we provide an example of a WBS.)

Given this assignment, we can now calculate the Business Value (BV) of any WBS bucket, including stories. The idea is simple and has two parts. First, the value of the whole tree is set 100% (or 1), which means that implementing all branches will yield all the value. This value (of 1) is assigned to the top of the tree. Second, as we move down the tree, each bucket's value is the appropriate percentage of its parent's value, as compared to its 'siblings' - the other children of its parent. The calculation is recursive, and uses the formula (1) below:

$$BV(bucket) = BV(parent) x \left[\frac{wt(bucket)}{wt(bucket) + \sum_{siblings} wt(sibling)} \right] \quad (1)$$

Once the BV per story is calculated, we can calculate the BV for sets of stories, that form options, by adding up their BV.

We suggest that the BV of the stories should be calculated at the beginning of each iteration, or each time changes occur that impact the value of requirements or the list with requirements. Knowing the value of the separate stories, different options (sets of stories) that could be considered for implementation in the upcoming iteration, can be discussed, and their values can be compared.

8 Perspectives on Options

From the clients' point of view, real-options thinking can be deployed to prioritize the requirements at the start of each iteration so that the delivery of business value is optimized. Suppose, the business value (BV) for each individual requirement is known to the client, s/he can re-arrange the requirements in sets that form options. Clearly, an option will be worth having when the cost of setting it up is less than its BV (which in our case is the sum of the BVs of all requirements that form the option). The client can, then, compare the advantages of each option and select the one that has the optimal BV. The client can wait to the last responsible moment (as it is called in [Poppendiek 06]) to make his decision on the set of requirements to be implemented and this allows her/him the chance to incorporate late breaking information and consider alternative sets of requirements. The term 'responsible' means that the client needs to understand the last point of time to make a decision without affecting the delivery of the project.

Second, from developers' perspective, the real-options thinking can support the implementation prioritization process. For example, the authors of [Matts 07] report on a practice of XP and Scrum developers who defer the decision about which story to develop until just before the coding starts. This allows them to incorporate information that arrives at the last moment, such as a new client request. In fact, the Scrum Backlog provides a forum [Matts 07] where any idea for functionality can be recorded *without requiring an immediate commitment to build it*.

When discussing the 'options' in this chapter, we explicitly take a specific viewpoint into account, namely the perspective of the decision-makers at the inter-iteration time which usually is the client or client's proxy.

In the next section we elucidate the question of the different options, that can be considered by the client.

9 Formulating client's options

First, to formulate client's options in agile inter-iteration time, we consulted the ROA literature and identified the options that have been considered in the sources. Next, we looked at literature in agile software engineering to find examples of options thinking. Despite the fact that we found no publication explicitly discussing how options thinking can be a useful decision making vehicle from client's perspective, we should note that the agile literature we searched does provide examples when decision-makers think of requirements in terms of options. We catalogued the clusters of organization-specific experiences to discern kinds of options which seemed to be considered and used in the requirements reprioritization processes discussed in the publications which we analyzed. Next, we mapped the options that we discerned from the ROA-literature [Baecker 04] to the observations that we made in the agile publications.
The result of this analytical process is presented in Table 14.
According to [Baecker 04] there are the following types of real options: Waiting-to-Invest option, Exit option and Learning/Flexibility type of options which includes 4 options addressing various flexibility-centric scenarios. In the first column of the table we list these options, and in the second column we map them to the options we found to exist from client's perspective.

Table 14: Description of Options

Option	Description
Waiting-to-Invest (postpone)	Wait to determine whether to implement certain requirements without imperiling the potential benefits.
Exit (Abandon)	Abandon the project (terminate at the current stage).
Learning/Flexibility, that can result in:	
Scope up	Add new functionality or quality features, not scheduled previously.
Scope down	Remove already implemented or negotiated features.
Switch	Change or re-arranging the stack with requirements.
Growth	Decisions that will possibly open up future opportunities for growth of the client organization.

10 Examples of client's options in agile projects

In this section we provide examples of the options listed in Table 14 from the agile literature.

1. The option of Waiting-to-Invest (Postponing): the project of ThoughtWorks (an agile coaching firm) and a major US insurance company, working on re-writing a large Java application used in support of core business processes, is a case in point [Tengshe 07]. In this project, the clients structured their business requirements in so-called 'epics' which are a compound story framing a software feature in context of a business scenario. At inter-iteration time, the client went through the release plan's epic list and marked each one as 'Must have', 'Should have', Could have', and 'Won't have (this time around)'. Epics not part of the initial release plan and deemed lower in priority had been deferred until a future release.

2. The option of Exit (Abandoning): In many agile projects, the client has the right to cancel at the end of any phase, receiving the working, tested software from all phases completed so far. The experience of a Control System Manufacturer [Mahanti]

indicates how clients can cancel a project early if they find it is not going as expected and thus loose minimal investment; for example, a project review found that only 20% of the projected business value had been achieved, which was used by the clients to conclude that the project should no longer be pursued.

3. The option of Scoping-up: This is an inherent part of any agile process and the varieties of features or functionality pieces that might be added in any iteration, all depend on the types of stakeholders on the client's side involved. As [Ambler 05] indicates, operations and support people, architects, regulatory compliance auditors, senior management, all may change their requirements.

4. The option of Scoping-down: Yahoo!'s Mixd project [Gatz 07] illustrates the use of this option. In their social mobile product experiment, the Yahoo! Advanced Product team cut features and learnt how removing complexity in the product and removing assumptions about users allowed people to use it in unintended way. The team prioritized the features by asking if the feature was absolutely necessary to help the users achieve their goal. They ruthlessly cut on those features that diluted the key focus of the product. Dropping those features that they had specified earlier was a major conceptual shift, but turned out to be an easy shift to make, as it eliminated development complexity.

5. The option of Switching: because agile applications are developed 'in vertical slices instead of horizontal ones' [Poppendiek 06] (this means each iteration includes a little bit of RE, of architecting, of codding, and of testing; this contrasts the 'traditional software development lifecycle model that assumes all RE is done upfront and only when it's over the next stage of the projects starts), the client never receives 100% of one tier completed before moving to the next one. This lets him/her switch some features and hook them together differently from the original set up. For example, at Sabre Airlines Solutions, clients compared alternative sets of features and switched to 'simplified functionality', and at the beginning of each iteration they deemed an alternative set of requirements [Poppendiek 06].

6. The Growth option can be thought of as an option on an option (if I realize this option, then I open another option). It represents the opportunity to make decisions that will, in the future, lead to opening up other opportunities for increasing business value. This aligns with Ambler's observation in [Ambler 09] that one should not optimize too locally. From a client's perspective – one presumes that the client pays for the software because it wants to achieve certain benefits in the future. Whether such growth will be realized will depend on many factors, and one of them is the IT product. Thus, the growth for the client's organization will depend on the selection of

the features-to-be-included in the product. The growth from the developer's perspective is related to the process of IT development and considers those agile practices that will help the growth of the developers' organization in the future.

11 Investigating the consideration of options in a case study

Building upon the definition that we provided in Section 6 and the examples of options from Section 10, we investigated the consideration of options and the application of option-thinking in real-life projects. For this purpose we used our multiple case study that we already discussed in Chapters 3 and 4. Here we analyze the results from the perspective of option-thinking and asked ourselves the question: what options can we identify in the studied projects? Below, we present those observations from the case study, that pertain to this question.

As we explained in Chapter 3, the case study aims at explicating the inter-iteration decision-making process in context of agile projects and changing requirements. We strive to better understand the way in which the agile requirements mid-course decision process contributes to the client's value creation. We studied how requirements prioritization and decision-making on priorities happen and what are the factors that play a role in it. As part of the investigation on how companies prioritize requirements in agile projects, we observed examples of options-thinking as practiced by agile teams.

The interview questions that provided insights about the option thinking were:

(i) Who decides about priorities?
(ii) What criteria are applied?
(iii) What were the reasons for changes in the requirements backlog?
(iv) Has value been explicitly considered?
(v) How does the agile process create value for the client?
(vi) Does the developer consider other factors in making decisions, a part from the value for the client?

Below we provide examples of application of option-thinking from the case study.

1. The *waiting to invest (postpone)* option was not observed in any case. In the following section we discuss possible explanations for this observation.

2. The *exit* (*abandon)* option has not been observed in the projects involved in the case study. Moreover, the practitioners that we interviewed didn't refer to any project that has been terminated prematurely. In our opinion, there might have been a number of explanations for this, originating in the nature of the projects. In the majority of the companies that we investigated, the contractual agreement between the vendor's the and the client's organization had been made on fixed-price basis. This means that the overall scope of the project had been outlined at the beginning of the project and it would make little sense for the clients (from economical perspective) to cancel the project before they get the functionality they desired (as delivery risks are beard by the software vendor in fixed-price contracts). Our gut feel suggests that the fixed-price character of the contracts influences the decision-making process in agile projects and we consider it as a topic for future research in its own right.
3. The *scope up* option is the natural evolution of an agile project, where new functionality and features are proposed and included in the project. Our case study, however, comprised mainly bespoke development projects, coupled with fixed-price or fixed-delivery-date contract. In this constellation we could not observe cases where the scoping up happened as an intrinsic part of the project, as the functionality was decided upon (and priced) at the beginning of the project. One of the case study participants underlined that: "We had a budget of days and knew exactly which requirements to implement in this time. Because of the fixed budget all bigger changes were managed via a request for change. This means, that when the clients want a change, it's not envisioned in the planning; that's why the higher management (of the client) needed to approve additional budget for it.."
   In our opinion, the *scope up* option can be exercised in cases where budget and resources are available, and the client desires further functionality.
4. The *scope down* option has been observed in a number of cases.
   Our observations suggest that this option is executed in cases where new knowledge, insights or learning during a project forces the stakeholders to re-estimate the resources needed to complete the previously planned amount of work. It resulted in significantly larger effort for implementing part of the functionality and this in turn caused scoping down of the overall functionality. Lower priority features or nice-to-haves have been cut. This is especially the case in projects with fixed resources (money or time), or with hard deadline. Particular example of this option is the case in the academic project where the participants had to cut initially planned features and functionality. The reason

was the realisation that the print function cannot be implemented as originally thought. This happened during the scrum planning meeting. This insight meant significant additional work. Because of the fixed-delivery date some of the other features in the backlog had to be cut.

5. The *switch* option has been applied naturally across projects. The examples below show the different facets in the application of the option in real-life projects:
   One of the case study companies that implemented a management system for health-care practitioners decided to move a sub-system for financial operations earlier in the implementation schedule. This switch happened later in the project when the client realized that the end of the quarter is coming and they would need the new system for their accounting purposes. This functionality has been considered to be of higher value at the time of the decision than the rest of the system.

   The change was driven by the desire of the developers to support as early as possible the needs of the clients, in this case – their taxation operations that happen to be needed because the end of the quarter. Certainly, the team could have left this functionality for later stage of the project, as was initially scheduled. The switching, however, saved the end client a lot of effort.

   We observed the following varieties in applying the switch option:

   (1) Sets of features have been switched as whole blocks. This example shows that the options can be considered not only with respect to their end-value for the client at the time of the delivery of the product, but also the current value of different options plays a role for the decision making. Moreover, this example shows that the option thinking, applied to an agile project, has an added value as the agile projects offer the flexibility to deliver part of the functionality and thus the clients profit as early as possible from the project.
   (2) The position of single features in the project backlog has been changed, as a result of learning, or postponing.
6. The *growth* option – As growth is an option on an option, we observed the application of this option to be common sense in the agile projects of our case study organizations. Our case study participants motivated this with the circumstance that their agile projects offer the possibility to build upon the current functionality at each stage of the projects and to enlarge and expand the product, depending on the will of the client. An example is the process,

applied in a small agile team, where we observed that the option thinking is being applied with the criteria to fit the feature selection into the available resources. At each iteration, the team developed those features, for which minimal use of resources was estimated. This is as well a growth option, as the project can be extended on the current grounds, whenever more resources are available.

12 Discussion on the observations

While in our previous section in this chapter, we formulated options and we found examples in our case study organizations that these options are meaningful, in this section we reflect on our observations. After compiling the information and mapping pieces of practitioners' evidence from our case study against the options presented in Table 14, we identified points of convergence and divergence between what we found in the first stage of our research (the insights from literature) and the next stage (namely, the case study). In the remainder of this section, we summarize these points.

First, the case-study observations suggest that options are almost always considered. We observed that the developers' decisions are often driven by options thinking, where the clients' perspective was taken seriously into account. This consideration, however, took place in an implicit or tacit way.

Second, we must note that not all types of options in Table 14 were equally considered by all practitioners. As we noted in the section above, most often the options of switching and postponing have been applied, while abandoning has not been observed at all.

Third, we consistently observed that the options were not stated in quantitative terms, but are, instead, explicitly or implicitly taken into account during the decision-making process.

Fourth, the evidence from the case study shows that, in contrast to the agile best practices, in most of the cases the developers are those who made inter-iteration decision making. As we explain in Chapter 4, only in few cases practitioners were able to provide evidence that the client is really capable/interested/aware of the agile way of defining priorities, and thus able to navigate through the functionality by the mid-course decision – making process. According to the practitioners, the developers' company is the one to make sure that the project delivery process runs in a way that

is profitable for the company. In those cases where the clients could not make the decisions, due to different reasons (as discussed in chapter 4), the developers are the ones that took over this process. In such cases, the option thinking, if applied, would consider rather the options from developers' perspectives and, potentially, would contribute for the value created for the developers instead of the value for the customers.

Fifth, throughout the interviews with the case-study practitioners, it became explicit that there is a link between the project's settings and the way the decisions are made. As we saw in chapter 4, in those projects where the client's company was a small company, the decision making was deliberately delegated to the developer. It could be a product owner, a project manager or another representative of the developing team, that was responsible for the communication with the client. The option-based reasoning made us think that how exactly a project organization would apply it on a project would depend on the context. That is, the context would determine who is the decision-maker, what decision-making criteria are used, and which view-point will dominate. However, more study is needed on the possible options in the different cases.

13 Conclusion

In this appendix we analyzed the agile decision-making situation and draw a parallel to the real-option analysis method. In doing so, we provide one possible way to explain how to consider the decision-making situation that the stakeholders face during the agile inter-iteration requirements prioritization and reprioritization process – namely, in terms of options.

We have shown that it is possible to use real-options thinking, that others have proposed this too, and that we have seen some use of it in our case study organizations.

The discussion above suggests that in order to determine the business value of a feature, both clients and developers are building on experts' knowledge in the domain of quantifying business value of information technology by using options-thinking. Methods based on options-thinking let organizations account for multiple aspects of IT adoption, as operational, strategic managerial benefits from financial, customers' or process perspectives. In the literature sources we reviewed, and in the case study

projects, we could find examples of such benefits: improve process efficiency, reduced cost, increased productivity and better customer-needs satisfaction. These examples indicated to us that it might be worthwhile further exploring the use of real options concepts as a decision-making vehicle. We therefore consider it a possible line for future research for the agile software engineering community to pursue.

Finally, we want to stress that although we reach the conclusion that there is a semantic parallel between the agile inter-iteration decision making and a systematic approach such as the real-options theory, we acknowledge that the practical application of this approach as-is (e.g. as applied in Benaroch [Benaroch 02]), is not feasible for most of the projects. There are decision-making situations where formulating options and comparing them is not possible or the value of the options can not be estimated reliably enough, thus applying this approach could lead to highly speculative results. In fact, we could not find a single case where different options were explicitly listed and compared to each other in terms of the value they would create for the clients. However, making the option thinking explicit could help the stakeholders to understand the decision space and eventually reason about the decisions they make in an explicit way.

Our analysis suggests also that applying options thinking might not be equally suitable for all kinds of projects. The full potential of this method can be used in those projects where the decision-makers are genuinely open to explore, compare and consider all possible options, including those of terminating projects. In contrast, we think, that this might not be the case in fixed-price projects where developers bear the delivery risks and therefore outline the requirements upfront [Daneva 13]. Nevertheless, such projects also could possibly benefit from applying option thinking in order to reason about possible implementation options, or to compare alternative implementation scheduling options for sets of features to be done in an iteration.

Appendix 3

Case Study Protocol

Conceptual Framework for a case study on requirements prioritization for value creation in an agile project.

The case study is composed as an **explorative** study, i.e. it investigates the state of the practice, and is not intended to test or validate a method or a theory. The outcome of the research will be a conceptual framework (which may also include a list with hypotheses to be tested in future research).

1. Research problem investigation.

A. **Research goa**l. What will the research results be used for: description,

evaluation, diagnosis, prediction, or something else?

The results will be used for:

1. *Description of the current practice of requirements prioritization in agile projects.*
2. *Gaining an insight into the understanding of value and the extent to which it is taken into consideration when making prioritization decisions during a project.*
3. *Evaluation of the effectiveness of the prioritization methods used, if a formal procedure is involved. (A method, a process, a practice).*
4. *Base for improvement suggestions of the practice – e.g. suggesting application of a structured approach. What we expect to improve?- The understanding of the stakeholders as well as of developers about the value of the features, and the impact of priorities decisions on the outcome of the project. Further on, a practice can be proposed about how to make value-driven decisions.*
5. *Identification of further problems (not explicitly mentioned in this document)*

B. **Problem owner**. Who are the stakeholders in the research problem, i.e. who wants to know?

The researcher and the participating companies are the problem owner as both we and the company want to know – on our side we identified it as an under-researched topic in the literature, and after discussions with the company it became clear, that these questions are important for the company as well.

C. **Unit of study**. What do we want to know something about?

Our unit of study is the process/practice of requirements prioritization in an agile project. It consists of two parts – (i) the outcome the final list with user stories (requirements) to be implemented in an iteration, and (ii) the process: the process of decision-making about prioritization. The first part is observable, while the observation of the second is not trivial. On the other hand, the first part on itself can not provide an insight into the mechanisms of the decision-making, and in particular into the consideration of value and other decision criteria that play a role.

What distinguishes the unit of study from other possible units of analysis?

The first unit of study – the list with requirements, is uniquely distinguishable. The requirements prioritization decision-making process can be mixed with decisions about, for example, architectural design. We can put a border and say, that each time there is a decision about the priority of requirements, it is an instance of the unit of study. We can refine the units of study by dividing them into value – driven and not value-driven. Here we have to make a decision about how to deal with cases where value is linked to other criteria, for example cost, or risk.

How do we count units of analysis; how do we know that we have one, two, etc.?

We define our unit of study as all the instances of decision-making, in which the list with requirements-to-be-implemented are defined or changed, before and during one iteration. Each time when there is a change in the list, we suppose that an instance of our unit of analysis has taken place.

When does a unit of study come into being and how long does it exist?

The artefact – list with requirements, starts existing after the first decision about the priorities of the requirements is made. It can exist forever. The prioritization process exists until the list with requirements is defined or changed.

D. Population. The population is the set of all possible units that satisfy the definition.

Our population is the set of all lists with requirements during a project, together with all decisions about the priorities. If it is not possible to obtain information or to analyze a whole project, we can restrict the population to a number of iterations.

E. Research questions. What do we want to know about the unit of study?

RQ1: How the concept of value is used in the process of requirements prioritization?

RQ2: How the requirements prioritization process helps the agile team deliver value to the client?

RQ3: How the concept of value is used in other decisions during a project?

These questions and topics deal with the decision-making from both client's and developers' perspective. The questions can be refined trough the following sub-questions:

- What does the priority mean to the stakeholders?
- How is a priority determined – is a structured method used, who is involved , who makes the decision about the priority?
- Based on what ? Which aspects of the requirements are important in the decision-making process?
- Is value considered? In which way?
- Are metrics (or other quantitative units, e.g. story points, or other estimations) used in support of these decisions?
- How frequently is this procedure performed? Are there events which can trigger new prioritization and new decisions on the project's development?
- How the prioritization affects the developing cycle? (This concerns technical aspects.)

F. Unit of data collection. This is the source of data about the unit of study. If the unit of data collection consists of people, then these are often called subjects.

These are the interviews with the case study participants.

G. Environment of data collection.

At the project site, in the office, over telephone.

H. The instrument of data collection.

J. Conceptual framework. Which concepts will we use to ask and answer the research questions?

Concepts:	*List with requirements (per iteration and as a whole)*
	Priorities
	Value
	Measurement
	Decisions about the priorities
	Rationale behind the decisions
	Impact of priority on development (and architecture).

2. Research design. What are we going to do to answer the RQ?

1. *Creating Questionnaire.*
2. *Interviewing practitioners.*
3. *Analysis of the information collected during the case study.*

Appendix 4

List of Figures

Appendix 5

List of Tables

References

[Abrahamsson 02] Abrahamsson, P., Salo,O.,. Ronkainen,J.,Warsta, J.: Agile Software Development Methods: Review and Analysis, VTT Technoical Report, 2002.

[Abrahamsson 03] Abrahamsson, P.,Warsta, J., Siponen, M.T.,Ronkainen, J.: New Directions on Agile Methods: a Comparative Analysis, Proc. of ICSE, 2003, IEEE CS Press.

[Adamopoulos 10] Adamopoulos, A., Managing Organizational Change, http://alexadamopoulos.wordpress.com/2010/07/19/, last accessed in Aug. 2013.

[Agile] Agile Manifesto, http://agilemanifesto.org/principles.html

[Ahl 05] Ahl, V. An Experimental Comparison of Five Prioritization Methods, Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden, 2005.

[AJ 11] How Do I Write Requirements Using Stories and Acceptance Criteria? http://www.agileconnection.com/article/how-do-i-write-requirements-using-stories-and-acceptance-criteria-part-one, 2011, last accessed in Aug. 2013

[AJ] Agile Journal, http://www.agilejournal.com

[Alleman 03] Alleman, G.B. Henderson, M. Seggelke, R Making Agile Development Work in a Government Contracting Environment Measuring Velocity with Earned Value, in Proc. of the Agile Development Conference, 2003. pp. 114- 119

[Ambler 02] Ambler, S.W., Agile Modeling - Effective Practices for eXtreme Programming and the Unified Process, Wiley, New York, 2002

[Ambler 05] Ambler, S.W., Measure Me Wisely, Dr Dobbs's Journal, July 2005, http://www.ddj.com/architect/184415360, last accessed in Aug. 2013

[Ambler 08] Ambler, S., Complex requirements on an Agile Project, Dr Dobbs, Oct 31, 2008, http://www.ddj.com/architect/211800534?cid=Ambysoft

[Ambler 09] Ambler, S., The Manifesto for Software Craftsmanship, Dr Dobbs's Journal, March 2009; http://www.drdobbs.com/architecture-anddesign/216200102

[Ambler 1] Ambler, S., Answering the "Where is the Proof That Agile Methods Work" Question, http://www.agilemodeling.com/essays/proof.htm

[Amram 03] Amram, M., and K. N. Howe (2003), Real Options Valuations: Taking Out the Rocket Science

[Amram 99] Amram, M., Kulatilaka, N., Real Options: Managing Strategic Investment in an Uncertain World. HBS Press, Cambridge, 1999, ISBN 0875848451

[Anand 93] Anand, P., Foundations of Rational Choice Under Risk. Oxford: Oxford University Press. ISBN 0-19-823303-5. 1993

[Armitage 08] Armitage, A. and Keeble-Allen, D. "Undertaking a Structured Literature Review or Structuring a Literature Review: Tales from the Field." *The Electronic Journal of Business Research Methods* Volume 6 Issue 2 2008, pp. 103 - 114, available online at www.ejbrm.com

[Augustine 05] Augustine, S., Managing Agile Projects, Prentice-Hall, 2005

[Aurum 07] Aurum, A., Wohlin, C.: A Value-Based Approach in Requirements Engineering: Explaining Some of the Fundamental Concepts, In Proceedings. of Requirements Engineering: Foundation for Software Quality REFSQ, LNCS, vol. 4542, Springer, Heidelberg (2007), pp. 109-115.

[Baecker 04] Baecker*, P.,* Hommel*, U. (2004): 25 Years* Real Options Approach *to Investment* Valuation: Review and Assessment, in: Zeitschrift für Betriebswirtschaft, Vol. 74, No. 3 supplemental issue (2004), pp. 1-53

[Balikuddembe 09] Balikuddembe, J.K., Bagula A., . "Business viability assessment of potential software projects: an empirical study with the CASSE Framework". in Proceedings of the ASEA2009 Conference, December 2009, pp. 229-236

[Barnett 07] Barnett L.: Agile Projects Must Measure Business Value, Jan 2007, Agile Journal, http://www.agilejournal.com/content/view/211/76/

[Barney 08] Barney S., Aurum A., C. Wohlin, A Product Management Challenge: Creating Software Product Value through Requirements Selection, Journal of Software Architecture, 54 (2008), pp. 576-593.

[Barney 09] Barney, S., Wohlin, C., Aurum, A.: Balancing software product investments, In: Proc. of the Symposium on Empiricl Software Engineering (ESEM), pp. 257-268 (2009)

[Bebensee 10] Thomas Bebensee, Inge van de Weerd, and Sjaak Brinkkemper Binary Priority List for Prioritizing Software Requirements, In Proceedings of REFSQ'2010. pp.67~78

[Beck 00] Beck, K. eXtreme Programming Explained: Embrace Change, Addison Wesley, 2000.

[Benaroch 02] Benaroch, M, Managing Information Technology Investment Risk: A Real Options Perspective. J of Management Info Systems, 19(2), Fall 2002, pp. 43-84.

[Berteig 06] Berteig, M., Methods of Prioritization, March 20, 2006 in Agile Advice Online Practitioners Forum, http://www.agileadvice.com/archives/2006/03/methods_of_prio.html

[Biffl] Biffl, S., Aurum, A., Boehm, B., Erdogmus, H., Grünbacher, P. (Eds): Value-based Software Engineering, Springer, Heidelberg (2006)

[Bleek 08] Bleek, W-G, H. Wolf, Agile Softwareentwicklung: Werte, Konzepte und Methoden. dpunkt, Heidelberg, 2008.

[Boehm 12] Boehm, B. Value-based software engineering: Overview and agenda, http://csse.usc.edu/csse/TECHRPTS/2006/usccsse2006-639/usccsse2006-639.pdf, accessed on 30.07.2012

[Boehm 81] Boehm, B.: Software Engineering Economics, Englewood Cliffs N.J., Prentice-Hall Inc., (1981)

[Boehm 86] Boehm B, "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes", "ACM", 11(4):14-24, August 1986

[Burnard 91] Burnard, P. (1991) A method of analysing interview transcripts in qualitative research. Nurse Education Today; 11: 461-466. 1991

[Cabri 06] Cabri A., M. Griffiths, Earned Value and Agile Reporting, in Proc. of AGILE Conf 2006, p. 6

[Cao 08] Cao, L., Ramesh, B."Agile Requirements Engineering Practices: An Empirical Study," IEEE SOFTWARE, Vol. 25, 01, pp. 60-67, JANUARY/FEBRUARY, 2008

[Charmaz 07] Charmaz, K. Constructing Grounded Theory: a Practical Guide through Qualitative Research, Thousand Oaks CA, Sage, 2007.

[Childs 98] Childs, P. D.; Ott, SH.; Triantis, A.J. Capital Budgeting for Interrelated Projects: A Real Options Approach, J of Financial & Quantitative Analysis, 33(3), 1998, pp 305-335.

[Clarke 05] Clarke, A. Situational Analysis: Grounded Theory after the Postmodern Turn, Thousand Oaks, CA, Sage, 2005.

[Cohn 04] M. Cohn, User Stories Applied: For Agile Software Development. Addison-Wesley Longman, Amsterdam, 2004

[Cohn 05] Cohn, M.: Agile Planning and Estimating, Prentice Hall Inc, (2005)

[Concha 07] Concha, M., Visconti, M., Astudillo, H.: Agile Commitments: Enhancing Business Risk Management in Agile Development Projects. XP 2007: 149-152

[Crow 09] Crow, K. Customer-focused Development with QFD, URL: http://www.npd-solutions.com/qfd.html, Accessed on August 26, 2009.

[Daneva 13] Daneva, M. van der Veen, E. Amrit, C. Ghaisas S. Sikkel K, Kumar R. Ajmeri N, Ramteerthkar U., Wieringa, R. Agile requirements prioritization in large-scale outsourced system projects: An empirical study, in The Journal of Systems & Software, Volume 86, Issue 5, Pages 1153-1464 May 2013

[Davies 02] Davies, D. Dodd, J. Qualitative Research and the Question of Rigor, Qualitative Health Research, 12(2): 279-289, 2002

[Davis 07] Davis, D., Zhao, J. Transforming air traffic management systems using SOA-based business (Mission) driven development methodology, 52nd Annual Conference of the Air Traffic Control Association – 2007, pp 18-25

[Deming 86] Deming, W.E.: Out of the crisis. Massachusetts Institute of Technology, Massachusetts, USA, 1986

[Dr Dobb's] Dr Dobb's portal http://www.ddj.com/

[Dyba 08] Dyba, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: a Systematic Review, Journal of Information and Software Technology, 50, 2008, pp 833-859.

[Easterbrook 08] Easterbrook, S., Singer, J., Storey, M-A., Damian, D., Selecting Empirical Methods for Software Engineering Research, in Guide to Advanced Empirical Software Engineering, Springer, 2008, ISBN 978-1-84800-043-8 (Print) 978-1-84800-044-5 (Online)

[Elssamadisy 06] Elssamadisy A., Whitmore, J., Functional Testing: A Pattern to Follow and the Smells to Avoid, PLoP Pattern Languages of Programs 2006 Conference Proceedings, 2006

[Erdogmus 02] Erdogmus, H., Favaro, J., Keep Your Options Open: Extreme Programming and Economics of Flexibility. In Extreme Programming Perspectives, M. Marchesi et al (eds). Addison-Wesley, 2002, URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.1786

[Erdogmus 05] Erdogmus, H. The Impact of Learning and Flexibility on the Economics of Iterative Development. IEEE Software, Nov/Dec 2005.

[Erdogmus 05-1] Erdogmus, H. The Economic Impact of Learning and Flexibility on Process Decisions, IEEE Software, Nov/Dec 2005, pp.76-83

[Favaro 05] Favaro, J.M. That Elusive Business Value: Some Lessons from the Top, in Proc of Int Conf. on XP, 2005, LNCS Springer, p. 199.

[Fink 05] Fink, A. (2005) *Conducting Research Literature Reviews,* London, Sage.

[Fitzgerald 06] Fitzgerald, B. G. Hartnett, K. Conboy, “Customising Agile methods to Software Practices at the Intel Shannon”, European Journal of Information Systems (EJIS), 2006, 15(2), pp. 200-213.

[Fraser 02] Fraser, J., Setting Priorities, April 23, 2002, URL: http://www.adaptivepath.com/ideas/essays/archives/000018.php

[Gatz 07] Gatz S. A., G. Benefield, Less, Never More: Launching a Product with Critical Features and Nothing More, In: Proc of AGILE Conf, 2007, IEEE CS, pp. 324-327.

[Gilb 85] Gilb, T. Evolutionary Delivery versus the "waterfall model" In Newsletter ACM SIGSOFT Software Engineering Notes, Volume 10 Issue 3, July 1985, Pp. 49 – 61, ACM New York, NY, USA

[Glaser 86] Glaser B G, Strauss A L 1967 The discovery of grounded theory. Aldine, New York; Strauss 1986

[Gottesdiener 11] Gottesdiener, E., Gorman, M. It’s the Goal, Not the Role: The Value of Business Analysis in Scrum, http://www.agileconnection.com/article/its-goal-not-role-value-business-analysis-scrum, , 2011 last accessed in Aug. 2013

[Gottesdiener] Gottesdiener, E., A View To Agile Requirements, EBG Consulting, Inc. www.ebgconsulting.com,

[Gottesdiener-1] Gottesdiener, E., At a Glance: Other Prioritization Methods, EBG Consulting, Inc. www.ebgconsulting.com, http://www.ebgconsulting.com/Pubs/Articles/At%20a%20Glance-Other%20Prioritization%20Methods-supplement-EBG%20Consulting.pdf

[Gurses 07] Gurses L.: Increasing Business Value by Adopting Agile Methods, 08 May 2007, Agile Journal, http://www.agilejournal.com/content/view/410/

[Hansen 09] Hansen, M. T., Baggesen, H. From CMMI and Isolation to Scrum, Agile, Lean and Collaboration. In Proceedings of AGILE, Conference – 2009, pp. 283-288. IEEE Computer Society

[Harris 06] Harris, R. S., M. Cohn, Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects. XP 2006: pp. 175-180

[Hartmann 06] Hartmann, D., Appropriate agile measurements: Using metrics and diagnostics to deliver business value, In Proceedings Agile 2006 Conference, pp. 126-131

[Hashmi 07] Hashmi , S.I., Bail, J. Software Quality Assurance in XP and Spiral - A Comparative Study Proceedings of The International Conference on Computational Science and its Applications, ICCSA 2007, pp. 367-374

[Highsmit 04] Highsmit, J. Agile Project Management, Addison-Wesley (2004)

[Hipps 93] Hipps, Jerome A. Trustworthiness and authenticity: Alternate ways to judge authentic assessments. Paper presented at the annual meeting of the American Educational Research Association, 1993

[Horton] Horton, K., Beaver, G. Driving Enterprise Agility from the Program Management Office, http://www.agilejournal.com/articles/columns/column-articles/3099-driving-enterprise-agility-from-the-program-management-office

[Hunt 06] Hunt, J., Agile software construction. Springer, London, 2006.

[InfoQ] InfoQ software development community http://www.infoq.com/agile

[ISO 310000] ISO 310000 http://www.iso.org/iso/home/standards/iso31000.htm

[Karlsson 07] Karlsson, L., Thelin, T., Regnell. B., Berander, P., Wohlin, C., Pair-wise Comparisons versus Planning Game Partitioning-experiments on Requirements Prioritisation Techniques, Empirical Software Engineering, 12(11), 2007

[Kitchenham 04] Kitchenham, B., Procedures for Undertaking Systematic Reviews, Joint Technical Report, Computer Science Department, Keele University (TR/SE-0401) and National ICT Australia Ltd. (0400011T.1), 2004.

[Kroll 03] Kroll, P., Kruchten, P. Rational Unified Process Made Easy, The: A Practitioner's Guide to the RUP, 2003

[Ktata 09] Ktata, O. Lévesque, G.,. Agile development: issues and avenues requiring a substantial enhancement of the business perspective in large projects.. Proceedings of the 2009 C3S2E conference on - C3S2E '09, page 59-66, 2009

[Kundu] Kundu, A., A Tale of Two Product Owners, http://www.agileconnection.com/article/tale-two-product-owners, last accessed in Aug. 2013

[Larman 08] Larman, C.Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum, Addison-Wesley, 2008.

[Leach] Leach, M., Put Business Analysts to Work for You!, http://agile.techwell.com/articles/weekly/agile-practitioners-put-business-analysts-work-you, last accessed in Aug. 2013

[Leffingwell 03] Leffingwell, D., Widrig, D., Managing Software Requirements, 2nd ed. Boston, Addison-Wesley, 2003

[Lehto 09] Lehto, I., Rautiainen, K., Software development governance challenges of a middle-sized company in agile transition, In SDG '09 Proceedings of the 2009 ICSE Workshop on Software Development Governance, Pages 36-39, 2009

[Li 07] Li, C., Akker, J.M. van den, Brinkkemper, S., Diepen, G. Integrated Requirement Selection and Scheduling for the Release Planning of a Software Product, In Proc. of REFSQ '07, Spinger LNCS, pp. 93-108, 2007

[Linders] Linders, B. Agile Process Improvement, http://www.agilejournal.com/articles/columns/column-articles/5847-agile-process-improvement, last accessed in Aug 2011

[Lippert 02] Lippert, M., S. Roock, H. Wolf, , eXtreme Programming in Action: Practical Experiences from Real World Projects, John Wiley & Sons , 2002

[Logue 08] Logue, K., K. McDaid, Agile Release Planning: Dealing with Uncertainty in Development Time and Business Value, Engineering of Computer Based Systems, 2008. In: Proceedings of 15th Annual IEEE International Conference and Workshop, 2008, pp:437 – 442.

[Logue 1 08] Logue, K., McDaid K., Handling Uncertainty in Agile Requirement Prioritization and Scheduling Using Statistical Simulation., In: AGILE, 2008, pp. 73-82

[Luehrman 98]Timothy Luehrman: "Investment Opportunities as Real Options: Getting Started on the Numbers". Harvard Business Review 76, no. 4 (July - August 1998): 51-67.; "Strategy as a Portfolio of Real Options". *Harvard Business Review* 76, no. 5 (September–October 1998): 87-99.

[Mahanti]Mahanti, A., Challenges in Enterprise Adoption of Agile Methods –a Survey, J of Computer & Information Technology, 14(3), 196-207

[Matts 07] Matts, C., Maassen, O., “Real Options" Underlie Agile Practices, June 8 2007, URL: http://www.infoq.com/articles/real-options-enhance-agility

[McDaniels 03] McDaniels, Mitchell J. Small, Risk analysis and society: an interdisciplinary characterization of the field, Timothy November 2003, Cambridge University Press

[Morali 10] Morali, A., Wieringa, R.J.: Risk-based confidentiality requirements specification for outsourced it systems. In: Proceedings of the 18th IEEE International Requirements Engineering Conference (RE 2010), Sydney, Australia, Los Alamitos, California, pp. 199‑208. IEEE Computer Society (September 2010)

[Muller 03] Muller , M., Padberg , F., On the Economic Evaluation of XP Projects ACM SIGSOFT Software Engineering Notes Volume 28 , September 2003,, SESSION: Software process and workflow, Pages: 168 - 177

[Oracle 07] Getting Started With Use Case Modeling, An Oracle White Paper, May 2007 http://www.oracle.com/technology/products/jdev/collateral/papers/10g/gswUseCaseModeling.pdf

[Padberg 04] Padberg, F., Müller, M.,: Modeling the Impact of a Learning Phase on the Business Value of a Pair Programming Project. In Software Engineering Conference, 2004. 11th Asia-Pacific APSEC 2004: pp. 142-149

[Patton 05] Patton, J., Finding the Forest in the Trees, In Proc of the IEEE Conf. on OOPSLA, 2005, pp: 266 – 274.

[Patton 08] Patton ,Jeff. Ambiguous Business Value Harms Software Products, IEEE Software, 25(1) , January/February 2008.

[Petersen 09] Petersen K., C. Wohlin, A Comparison of Issues and Advantages in Agile and Incremental development between State of the Art and an Industrial Case, Journal of Systems and Software, 82 (2009), pp. 1479-1490.

[Pettit 07] Pettit R., Business Value Applied: Aligning The Day To Day With Business Imperative, 04 January 2007, Agile Journal, http://www.agilejournal.com/content/view/206/33/

[Pieczko] Pieczko, S., Agile? Waterfall? How about WetAgile? http://www.emqus.com/index.php?/emq/article/agile_waterfall_how_about_wetagile_7781, last accessed in Aug. 2013

[Pinheiro 08] Pinheiro, C., Maurer, F., J. Sillito, Adopting Iterative Development: The Perceived Business Value, Lecture Notes in Business Information Processing, Agile Processes in Software Engineering and Extreme Programming, Prod of the 9th International Conference on XP, 2008.

[Podelko] Podelko, A., Agile Performance Testing, http://www.alexanderpodelko.com/docs/Agile_Performance_Testing_CMG08_presentation.pdf last accessed in Aug. 2013

[Poole 07] Poole, M., Business and IT – A Marriage Made in Heaven? 06 October 2007, Agile Journal, http://www.agilejournal.com/content/view/627/76/

[Poppendiek 06] Poppendiek, M., T. Poppendiek, Implementing Lean Software Development: from Concept to Cash, Addison- Wesley, 2006, ISBN 0321437381

[Port 09] Port, D., Tung Bui: Simulating mixed agile and plan-based requirements prioritization strategies: proof-of-concept and practical implications. EJIS 18(4): 317-331 (2009)

[Pratt95] Pratt, J.W., H. Raiffa, R. Schlaifer, Introduction to Statistical Decision Theory, MIT Press, Cambridge, MA, 1995 ISBN 0585223149

[Pyett 03] Pyett, P.M. Validation of qualitative research in the “real world.” Qualitative Healh Research, 13(8): 1170-1179, 2003

[Quality 93] Center of Quality Management Journal (1993) Special issue on Kano's Methods for Understanding Customer-defined Quality, Vol 2, No 4, Fall 1993

[Qumer 07] Qumer A., Defining an integrated agile governance for large agile software development environments, In XP'07 Proceedings of the 8th international conference on Agile processes in software engineering and extreme programming, 4536 LNCS, pp. 157-160, 2007

[Qumer 08] Qumer, A., Henderson-Sellers, B., A framework to support the evaluation, adoption and improvement of agile methods in practice, In Journal of Systems and Software, Volume 81 Issue 11, November, 2008, pp. 1899-1919

[Racheva PR 09] Racheva, Z. and Daneva, M. and Sikkel, K. Value Creation by Agile Projects: Methodology or Mystery? In: 10th International Conference on Product-Focused Software Process Improvement (FROFES 2009), June 2009, Oulu, Finnland, Pp. 141-155

[Racheva PR 10] Racheva., Z., M. Daneva, K. Sikkel, L. Buglione, "Business Value Isn't only Dollars- Results from Case Study Research on Agile Software Projects", in Proceedings of PROFES 2010, 11th International Conference on Product-Focused Software Process Improvement, Limerick (Ireland), June 2010,

[Racheva-1 10] Racheva, Z., Daneva, M., Sikkel, A. Herrmann, R. Wieringa, Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study, in the Proceedings of Requirements Engeneering 2010, Australia

[Racheva-2 10] Racheva., Z., M. Daneva, A. Herrmann, R. Wieringa, "A conceptual model and process for client-driven agile requirements prioritization", in the Proc. of 4th International Conference on Research challenges in Information Science (RCIS), Nice, France, May 2010, IEEE

[Rao 07] Rao, N: *Entrepreneurial Digital Photography* - A Case Study for Design Research Method in the Emerging Indian Market. HCI (11) 2007: 193-200

[Rawsthorne 06] Rawsthorne, D., Calculating Earned Business Value For An Agile Project, Agile Journal, June 2006, http://www.agilejournal.com/articles/articles/calculating-earned-business-value-for-an-agile-project.html

[Rawsthorne] Rawsthorne D., Managing the Work in an Agile Project, http://www.netobjectives.com/files/resources/downloads/ManagingTheWork.pdf

[Reed 08] Reed, P.: An Agile Classroom Experience. AGILE 2008:Pp. 478-483

[Rico 09] Rico, D.F., Sayani, H.H., Sone, S.: The Business Value of Agile Software Methods: Maximizing ROI with Just-in-Time Processes and Documentation. J. Ross, Ft. Lauderdale, (2009)

[Rico 1] Rico, D, What is the ROI of agile vs. traditional? An analysis of XP, TDD, Pair Programming, and Scrum (Using Real Options), http://davidfrico.com/rico08b.pdf, last accessed Aug. 2013

[Rikkilä 10] Rikkilä, J.,New Approach for Managing Lean-Agile Development: Overturning the Project Paradigm, Lecture Notes in Business Information Processing, 1, Volume 65, Lean Enterprise Software and Systems, Part 3, Pages 139-150, 2010

[Rippe 99] Rippe, Schaber, Demokracy and Environmental Decision-making, Environmental Values 8, 75-98, 1999

[Rönkkö09] Rönkkö, M., Frühwirth, C., Biffl, S.: Integrating Value and Utility Concepts into a Value Decomposition Model for Value-Based Software Engineering. In: 10th International Conference on Product-Focused Software Process Improvement, Springer, Heidelberg (2009): pp 362-374

[Rosenhead 01] Rosenhead, J., Robustness Analysis: Keeping Your Options Open, In J. Rosenhead et al. (eds.) Rational Analysis for a Problematic World Revisited: problem structuring methods for complexity, uncertainty and conflict, Wiley, Chichester, 2001, pp. 181-207, ISBN 978-0-471-49523-9

[Runeson 09] Runeson, P., M. Höst, "Guidelines for conducting and reporting case study research in software engineering", Journal of Empirical Software Engineering, v.14 n.2, April, 2009, Springer, Pp 131-164

[Saaty 80] Saaty, T.L., The Analytic Hierarchy Process, McGraw-Hill, New York, 1980.

[Saldana 09] Saldana, J. (2009). The coding manual for qualitative researchers. London: Sage

[Schwaber 01] Schwaber, K., Beedle, M.: *Agile Software Development with Scrum*. Prentice Hall, Upper Saddle River 21. Oktober 2001, ISBN 978-0130676344, S. 36.

[Schwaber 04] Schwaber, K., Agile Project Management with SCRUM, Microsoft Press, 2004.

[Scorecard 96] The Balanced Scorecard: Translating Strategy into Action, Harvard Business School Press, Boston (1996)

[Seale 99] Seale, C. Quality in Qualitative Research. Qualitative Inquiry, 5 (4):465-478, 1999

[Shadish 02] Shadish, W.R., Cook, T.D., & Campbell, D.T. Experimental and Quasi- Experimental Designs for Generalized Causal Inference. Boston : Houghton- Mifflin, second edition, 2002

[Simons] Simons, H., Case Study Research in Practice, 2009

[Small] Small, B., Gadnis, C. "The Value of Concurrent Testing", http://www.ebizq.net/topics/saas/features/12307.html, last accessed in Aug. 2013

[Stake] Stake, R.E., The Art of Case Study Research, 1995

[Strauss 91] Strauss, A.L., J.M. Corbin, Basics of Qualitative Research - Grounded Theory Procedures and Techniques, Sage, Newbury Park, 1991.

[Sulaiman 06] Sulaiman, T. Barton, B. Blackburn, T. AgileEVM – Earned Value Management in Scrum Projects, in Proc. of AGILE Conf 2006, pp. 10

[Sutherland 95] Sutherland, J; Schwaber, K.Business object design and implementation: OOPSLA '95 Workshop Proceedings. The University of Michigan. p. 118. ISBN 3-540-76096-2, 1995

[Szoke 11] Szoke, A. Conceptual Scheduling Model and Optimized Release Scheduling for Agile Environments Journal of Information and Software Technology (53), 2011, pp. 574-591

[Tabaka 06] Tabaka, J., Collaboration Explained: Facilitation Skills for Software Project Leaders. Addison Wesley 2006.

[Tengshe 07]Tengshe, A., Establishing the Agile PMO: Managing Variability across Projects and Portfolios, Proc. of the Agile Conf. 2007 (Agile 2007), Washington, DC, USA. IEEE CS, pp. 188-193

[Ton 07] Ton, H., A Strategy for Balancing Business Value and Story Size, AGILE '07 Proceedings of the AGILE 2007, IEEE Computer Society Washington, DC, USA, pp. 279-284

[Vidgen 09] Vidgen, R, Wnag, X. Coevolving Systems and the Organization of Agile Software Development, In Journal Information Systems Research, Volume 20 Issue 3, September 2009, pp 355-376

[Webler 95] Webler, T., "Right" discourse in citizen participation. An evaluative yardstick. In Renn, Webler, & Wiedemann (EDs.), Fairness and competence in citizen participation. Evaluating new models for environmental discourse ,1995, (pp. 35-86). Boston/ Dordrecht: Kluwer

[Webster 02] Webster J., R.T. Watson: Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly 26(2): (2002)

[Wiegers 99] Wiegers, K., First Things First: Prioritizing Requirements, Software Development, 7(9) 1999;

[Wieringa 09] Wieringa, R., Design science as nested problem solving, In SRIST '09 Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology, 2009

[Wieringa 09-1] Wieringa, R. Design Science and Software Engineering. ICSOFT (1) 2009: 5

[Wieringa 12] Wieringa, R., Morali, A., Technical Action Research as a Validation Method in Information Systems Design Science, DESRIST, pp. 220-238, 2012,

[Wieringa 12-1] Wieringa, R.J. Designing technical action research and generalizing from real-world cases. In: Proceedings of the 24th International Conference on Advanced Information Systems Engineering (CAiSE 2012), 25-29 June 2012, Gdansk, Poland. pp. 697-698.Lecture notes in computer science 7328. Springer Verlag. ISSN 0302-9743 ISBN 978-3-642-31094-2, 2012

[Williams 07] Williams, M., Jay Packlick, Rajeev Bellubbi, Scott Coburn, How We Made Onsite Customer Work - An Extreme Success Story, Proc. of the Agile Conf. 2007 (Agile 2007), Washington, DC, USA. IEEE CS, ISBN 0-7695-2872-4, pp. 334-338

[Wohlin 00] Wohlin, C., P. Runeson, M. Höst, M.Ohlsson, B. Regnell and A. Wesslén, Experimentation in Software Engineering – An Introduction, Kluwer Academic Publisher, 2000

[Yap 06] Yap, M.: Value Based Extreme Programming. In Proceedings of the AGILE 2006, pp 175-184.

[Yin 04] R.K.: Case Study Research: Design and Methods, Sage (2004)

Samenvatting

Iteratieve en incrementele methodieken voor softwareontwikkeling zijn de laatste jaren opgekomen als alternatief voor de traditionele ontwikkeling volgens het watervalmodel.

Reden voor deze verandering is dat in het verleden te veel softwareprojecten geplaagd werden door budgetoverschrijdingen, veranderende eisen, en scope creep (het telkens bijna ongemerkt oprekken van de scope van een project). Ondertussen is het in de IT-wereld geaccepteerd dat niet alle projecten vooraf geheel voorspelbaar zijn. Onzekerheden in de markt en een voortdurend veranderende omgeving leiden tot bijstellingen tijdens de ontwikkeling van een product.

Een van de essentiële karakteristieken van alle zogenaamde agile benaderingen van softwareontwikkeling is de expliciete focus op business value. Uiteraard zijn alle softwareontwikkelmethoden erop gericht een waardevol product op te leveren, maar in een agile aanpak is het creëren van business value de kern en de focus van het ontwikkelproces. Een agile ontwikkelproces is een waardecreatieproces. In agile methoden wordt gewerkt met korte ontwikkelcycli, iteraties genaamd. Tussen deze iteraties wordt telkens vastgesteld welke eisen in de volgende iteratie geïmplementeerd zullen worden. Beslissingen daarover worden genomen door de prioriteiten van eerder opgestelde eisen te heroverwegen.

Doel van deze studie is het proces van waardecreatie in kaart te brengen en te begrijpen. De focus ligt daarbij op het voortdurend heroverwegen van prioriteiten van eisen aan het op te leveren product. Dit heroverwegen van prioriteiten is de manier waarop waardecreatie specifiek in agile processen is ingebed.

In verschillende deelonderzoeken hebben we enkele van de agile praktijken die bijdragen aan waardecreatie nader onderzocht, om er zo achter te komen welke kennis de beoefenaren van agile methodieken in de praktijk gebruiken om waardecreatie te realiseren. Uiteindelijk doel is om aan de hand van deze kennis de agile methoden nog verder te kunnen verbeteren.

Door een gedetailleerde studie van het proces van heroverweging van prioriteiten hebben we de criteria achterhaald op basis waarvan beslissingen genomen worden, evenals de relatie tussen de context en de uitvoering van het proces.

Bij het definiëren van nadere onderzoeksvragen en het ontwerpen van het onderzoek hebben ons laten leiden door de volgende hoofdvragen:

- Hoe wordt business value in agile projecten gedefinieerd en gemeten?
- Welke praktijken dragen bij aan waardecreatie in agile projecten in verschillende contexten?
- Welke concepten spelen een rol bij besluitvorming over heroverweging van prioriteiten?

De belangrijkste bijdragen van dit onderzoek voor research en praktijk zijn (1) een uitgebreide beschrijving van het proces van heroverweging van prioriteiten in zijn specifieke context en (2) een conceptueel model van dit proces.