

# Reprioritizing the Requirements in Agile Software Development: towards a Conceptual Model from Clients' Perspective

Zornitza Racheva, Maya Daneva

University of Twente, Netherlands, {z.racheva, m.daneva}@utwente.nl

**Abstract.** *Continuous and client-centric requirements reprioritization forms the very core of today's agile approaches. In this paper, we report on results of a grounded theory study on agile requirements prioritization methods. The outcome is a conceptual model for understanding the inter-iteration prioritization process from client's perspective. The latter is derived from the authors' experiences and by using empirical data, published earlier by other authors.*

**Keywords:** agile development, requirements prioritization, inter-iteration decision-making process, grounded theory.

## Introduction

Continuous requirements reprioritization from client's perspective forms the key of today's agile approaches. A recent empirical study [7] indicates that, with respect to requirements (re)prioritization, agile RE differs from 'traditional RE' in two ways: (i) (re)prioritization happens at inter-iteration time, which means the project team anticipates and plans as many reprioritization sessions as the number of project iterations, and (ii) (re)prioritization is based mostly on business value, that is, the highest priority *features* (i.e. requirements in agile terminology) get implemented early so that most business value gets realized. These two aspects of agile RE pose at least two challenges: (i) continuous reprioritization more often than not (especially when practiced without caution) leads to project instability, and (ii) clients, by and large, relate the concept of business value to features that meet their functional requirements, so non-functional requirements (such as scalability or security) that might initially appear secondary to clients, turn out critical for the operational success of the product. For example, redesigning the architecture of the software product at a late stage would add up to an over-expensive or a delayed project. In a context of a supplier network, these challenges may well aggravate further, as product managers make commitments based on process and product assumptions which are different from the ones of the development team. Yet, these assumptions might be essential to product success.

Our paper is a first attempt to respond to these two challenges. It proposes a conceptual model of the agile prioritization process from client's perspective. We

obtained it by applying a grounded theory approach. We make the note that we do not provide a new prioritization technique. Instead, we (i) redefine our view of requirements and their (re)prioritization by treating them from a clients' perspective, and (ii) we propose a model that reflects this specific focus and represents an unified approach to discussing the prioritization effort independently from the particular method that is used.

Our motivation for creating this model originates on the premise that the practices of continuous requirements reprioritization, with strong client participation, are a relatively recent phenomenon and because of this are only partially understood. As agile literature indicates, never before in the software engineering history, the client has been that actively 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 task, the key task of prioritizing requirements, he or she must be aware of the facets of his/her role and thus would profit from a decision-support vehicle available at his/her disposal. We think that a conceptual model can help the client in multiple ways: (i) to navigate through the agile process of delivering business value; (ii) to make explicit the tacit assumptions, used in different requirements prioritization methods; (iii) to identify the possible peaces and sources of information that might be of importance for the outcome of the prioritization and, consequently, for the project; (iv) to make the process more objective in the sense that having such a vehicle will allow also less experienced users to participate in the prioritization process and, they could do it with the confidence that they deliver a quality work.

The paper is structured as follows: Section 2 provides a discussion on the role of clients in agile RE and formulates our research question. Section 3 presents the Grounded Theory research method and Section 4 – its application and results. Section 5 evaluates our results and discusses the possible threats to their validity. Section 6 concludes the paper.

## Client-driven Requirements Prioritization and Reprioritization

The agile manifesto [26] deems the client's role critical in making decisions about "what to build". In the minimalist philosophy of XP – a prominent agile approach, the following is recommended for the client's role [5]: (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 prioritizing user story development. (4) The small 2-3 week iterations allow the user to evolve their requirements based on concrete working software. (5) The client regularly tests the software to confirm it works as expected.

Our focus in this paper is on item 3 of the above list, namely supporting the client when making prioritization decisions. Therefore, in this and the next sections, we re-focus the discussion on the role of clients' requirements prioritization (RP) in agile software development.

Clearly, RP 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 requirements (i) to implement earlier than others, or (ii) to include in an earlier release. The premise is that the whole functionality can not 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 client 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 implemented and delivered at once with the first release, and part of it will be eventually not implemented. 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 clients 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 into the product at an early stage. This highlights the paramount importance of the RP activities.

The changes in the list with 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 [19], 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. Their focus is particularly on incorporating learning and cost of change in the decision-making process.

Last, while in a traditional project the prioritization is usually performed once and before the implementation phase, in agile context it is an ongoing process, performed

in the beginning of each iteration, or even during the iteration; this reflects the dynamics of the project's backlog. The differences between the two settings are summarized in Table 1.

Table 1. Comparison of traditional and agile RP

Aspects of the RP process	Traditional (waterfall) development	Agile
Goals/Purpose of the RP	Project management-vehicle	- Vehicle to ensure that delivered business value is maximized at each iteration; - Scope definition vehicle at iteration level
When is RP performed	Typically once, after the analysis phase and before implementation	Before each iteration, at planning phase, or during iteration
Who is responsible for RP	Developer, with participation of project manager and other stakeholders.	Client is the key driver for choosing, being supported by Scrum Master (or Agile Project Manager) regarding the assessment of the technical feasibility of a schedule.

Building on the above discussion, we came up with the following research question (RQ): *“What are the key topics to consider when prioritizing the requirements from client’s perspective in agile projects?”*

## The Grounded Theory Approach

The term grounded theory [8] refers to a set of systematic guidelines for data gathering, coding, synthesizing, categorizing, and integrating concepts to conduct a theoretical analysis of an empirical problem. The name grounded theory points out to its fundamental premise that a researcher can and should develop theory from rigorous analyses of empirical data. As a qualitative research method, GT is distinctive in (i) that it is inductive in nature, which means that we as researchers have no preconceived ideas to prove or disprove data, (ii) that collection and analysis proceed simultaneously and each informs the other, and (iii) that constant comparative techniques treat possible disagreements between the emerging theory and newly collected information. A GT exercise of a studied topic starts with concrete data and ends with rendering them in an explanatory theory. From the very beginning, the researcher analyzes the data and identifies analytic leads and tentative categories to develop through further data collection. It is essential to note that, in this process, whenever the emerging theory disagrees with newly collected information from experiences or from literature, the researcher should not assume that the theory is wrong. Instead, the researcher seeks to extend the theory so that it makes sense of both

the data from the study and the data from the literature, because the key concern throughout a GT process is the fit of the theory to the data and its ability to make sense of actual experience.

Research methodologists [8,17,31] suggest that a theory derived from data is more likely to resemble what’s happening in reality, than a theory which is derived by putting together a set of concepts based on experience and solely through assumptions about how things in real life would work. As GT studies rest on the data, they are thought to enhance researchers’ understanding of a situation and provide a meaningful starting point for further action. The philosophical foundation of GT and how it affects the researcher’s choices in carrying out his/her work have been discussed in [8] and are beyond the scope of this paper. Here, we focus on the application of the GT process [8] and the results we obtained.

## The Application of Grounded Theory

For the purpose of our research, we used the GT guidelines by Kathy Charmaz [8]. We executed a research process which included the following steps: (1) identification and review of data sources from published literature, (2) initial and focused coding, (3) clustering and memo-writing, (4) conceptual modelling, and (5) theoretical sampling of empirical data, using the concepts and categories from our resulting conceptual model. The goal of steps 1-3 is the discovery of as many relevant categories as possible, along with their properties and dimensions. Step 4 is about the visual representation of the categories and their relationships, and Step 5 is about ‘saturating the categories’. Categories are considered ‘saturated’ when collecting fresh data no longer brings new theoretical insights nor reveals new properties of the categories in the conceptual model [8].

We traversed the steps 1-5 multiple times, as methodologists recommend [8], because: “Constant interplay between proposing and checking [...] is what makes our theory grounded!” [31]. That means, the analysis of the data collected in one step helps to check the interpretations from the previous step. In the subsections below, we indicate the execution of the steps along with the results we obtained from our application of GT.

### 4.1. The sources

In this study, the data used and constantly compared to the emerging theory is literature on agile requirements prioritization available via scientific digital libraries and prominent agile practitioners’ journals. We did a semi-systematic literature search using the five bibliographic databases: IEEEExplore, ACM Digital Library, Google Scholar, InterScience and Citeseer. We complemented

them with the following periodicals: the Agile Journal [1], and the platforms, dedicated to software development and agile methods: DrDobb’s [13] and InfoQ [20]. The key words we used for our search were: *agile, requirements, backlog, prioritization, inter-iteration, decision-making, business value, risk, cost, features*. We traced the references in the identified papers to get access to other relevant sources. To determine whether to include or not these sources to our GT research, for each one, we reviewed the abstracts and the conclusions and we checked this information against the following five 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 requirements prioritization, (4) the paper adequately describes the context, in which the method is expected to be applicable; ‘adequately’ means that the reader can replicate the use of the RPM in his/her own context; and (5) original paper: for each method, we searched at least its original publication; if an original paper is difficult to access, or is outside the RE field, we included another description from an RE author. The publications were written in English only and included both qualitative and quantitative research, from scientists and practitioners. We carried out the quality check by using these criteria, which yielded 42 papers eligible for inclusion and review in the GT process. These papers refer to 15 RP methods and one technique, as indicated in Table 2.

**Table 2. The RP approaches published in the sources used for the GT**

RP method	References
Round-the-group prioritization	[6]
Ping Pong Balls	[30]
\$100 allocation (cumulative voting)	[22]
Multi-voting system	[32]
MoSCoW	[16]
Pair-wise analysis	[18] [21]
Weighted criteria analysis	[18]
Analytic Hierarchy Process (AHP).	[29]
Dot voting	[18]
Binary Search Tree (BST)	[2]
Ranking based on product definition	[15]
Planning Game	[5] [21]
Quality functional deployment QFD	[11][18]
Wiegiers’ matrix approach	[33]
Mathematical programming techniques for release planning	[23]
Technique of bucketing requirements	[25]

## 4.2. The Conceptual model

The multiple iterations of coding, constant comparing of information from literature, and conceptual modelling in our GT process delivered two models, Model A, which is presented in Fig 1 and Model B, which is presented in Fig 2. Model A describes the agile RP process, while Model B elaborates on the conceptual categories related to making the RP decisions. We make the note that Model B (on Fig 2) is not meant as a refinement of Model A (on Fig 1). Instead, the purpose of Model B is to explicate and bring insights into the decision-making step, which is the core of the RP process.

Furthermore, both models take the perspective of the client, unlike RP authors [4] who adopt the perspective of the development team. We must note that the models take a 'big-picture' view to make explicit those pieces of information, necessary for the prioritization process.

To create these models we used the initial and focused coding practices described in [8]. This meant first to name the segments of data, and then to use the most frequent initial codes to "sort, synthesize, integrate and organize large amounts of data" [8]. We make the note that to us, the focused coding meant iteratively making decisions about those initial codes which the two authors deemed to make the most analytic sense to categorize the data, as Kathy Charmaz says, "incisively and completely". We complemented our coding with diagramming, which enabled us to visualize the connections among the conceptual categories and to see more clearly the relative strength or weakness of the relationships between the concepts. Our intensive diagramming activity was motivated by Adele Clarke [9] who contends that conceptual mapping preserves empirical realities and complexities. We drew diagrams and wrote notes, then reviewed them and dissected them meaningfully, while keeping the relations between the parts (that are dominant concepts, themes, and issues) intact. We followed this process, as it is meant to help the researcher to reduce and analyze data and direct him/her toward trends, themes, and patterns. Due to space

limitation, we do not provide a mapping between the literature sources we used to as input to build the model and the parts of the model derived from each source. We, however, plan to publish this in a separate paper in near future. Below, we describe the two models in more detail.

### 4.2.1. Model A

This model presents a generic prioritization process in terms of its inputs and outputs, as it is visible from the client's standpoint. We deliberately used concepts which make clear how the status of requirements changes – from 'Initial' to 'Prioritized', to "Sprint", to 'Implemented'. The input is the *Initial Project Backlog*, that is the total number of requirements upon the start of the project. Before the very first agile iteration, the client runs a RP technique, which produces *Prioritized Project Backlog*. This is an ordered list of the requirements (originally written in the *Initial Project Backlog*) according to their priorities. In agile settings, only a small portion of the upper part of this ordered list goes for implementation in the first iteration. (Iterations are called *sprint* in the jargon of Scrum - the most popular agile project management approach.) This small portion of prioritized requirements forms the so-called *Sprint Backlog*. Once the iteration is completed, the status of those requirements which are already implemented in the software product, changes to *Implemented requirements*. Those requirements which could not be implemented by the developer team are fed back into the project backlog and are subject to reprioritization before the new iteration starts. At that inter-iteration time, the client might decide to request a change to the requirements and this leads to a new reprioritization as well (this is the arrow from *Sprint backlog* to *Prioritized project backlog*). The client reprioritizes the project backlog, so that she/he knows the next portion of requirements which will go to the next *Sprint Backlog*. The relationship between the concepts *Prioritized Project Backlog* and *Sprint Backlog* - from the view point of the clients in agile projects, is elucidated in Model B (see Fig 2).

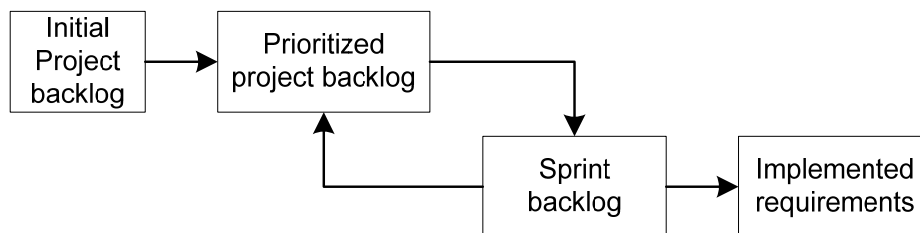


Fig. 1. Model A: the prioritization process from clients' perspective.

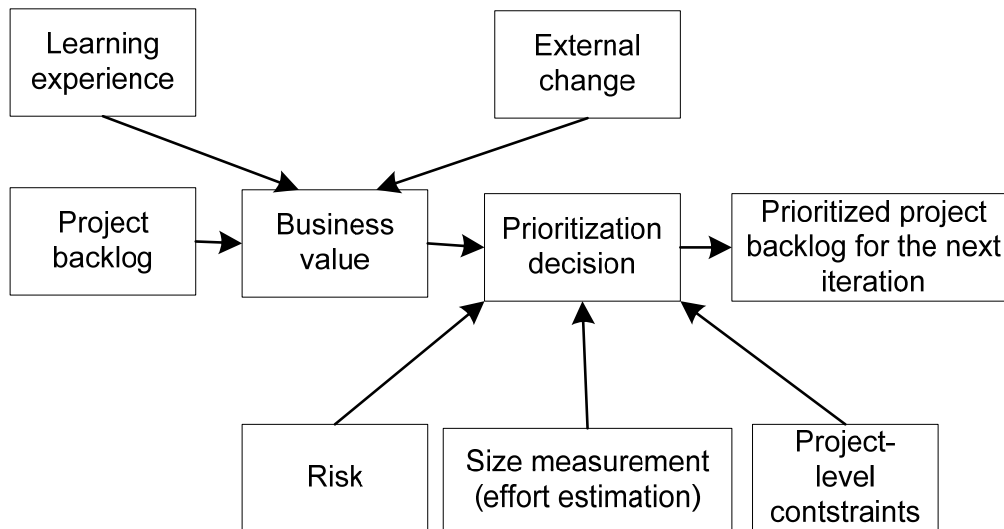


Fig. 2. Model B: topics to consider when making prioritization decisions

#### 4.2.2. Model B

This model is to help clients ‘zoom-in’ and see the aspects important for RP at inter-iteration time. As in Model A, in Model B we take a holistic perspective of RP. In contrast to Model A, Model B can be seen as a generic *framework for describing* the client’s decision-making situation while prioritizing the requirements. As per Alenjung and Person [3], a decision-making situation is “a contextual whole of related aspects that concerns a decision-maker”, that is – in our case, the client in an agile project. For example, one can use the conceptual categories of the framework (that is, Model B) to depict a specific client’s RP situation in a specific agile project, in a specific organization and, thus, take into account the topics important to be considered by the client when prioritizing requirements at inter-iteration time.

Furthermore, Model B shows the complexity of the decision-making from client’s perspective in agile RP. We observe, that the client, when prioritizing the *Project backlog*, explicitly or implicitly relies on tacit knowledge to estimate the *Business value* of each item (in the *Project backlog*). The estimation is qualitative (as it was already found in our previously published study on business value in agile [27]). Yet, the agile clients make a conscious effort to connect business value 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” [24].

The client assesses the *Business Value* of the requirements in the project backlog based on his/her current knowledge and *Learning Experiences* within the agile project as well as any changes (see the box *External*

*changes* in Fig 2) occurring in the business environment of the organization. An example for an external change can be a merger between the client’s organization and another organization. Both the client’s continuous learning throughout the project and the dynamic environment in which the client organization operates can make - from one iteration to another, some requirements more valuable than others. (It is possible that *External changes* can even render some requirements irrelevant).

Model B suggests that there are four aspects which the client considers when making his/her decision on requirements priorities: *Business Value*, *Risk*, *Size Measurement/Effort Estimation*, and *Project-level Constraints*. These four aspects are important to the way and the possibilities for a client to execute the decision-making step. We make the note that the agile RP literature sources converge on that the *Business value* is the dominating RP criterion. We also observed that some RP methods used the notion of ‘importance’, or relative importance of a feature, compared to other requirements, instead of ‘value’. Still, when reading about the application of the RP method, we understand that estimations of ‘value’, is the implicit prerequisite for these prioritization methods. In addition to Business Value, the client considers *Risk* due to development instability. Accommodating highly-volatile requirements, which in turn, means accommodating instabilities in the development process is an inherent aspect of the agile development process. As a matter of fact, the strong focus on business value and on continuous reprioritization of the requirements is the key to successfully coping with instability and volatile requirements.

Next, the client considers *Estimated Effort* based on functional size when making decisions on priorities for the next iteration. Size is based on the user stories and can, for example, be expressed in story points [10].

Another aspect which can be a consideration during the decision-making is a *Project-level Constraint*. This can include, e.g. budget constraints, fixed market-driven deadline or human resource constraint.

Last, the *Prioritized Project Backlog* is the ordered list of requirements which the developer team should act upon in the next iteration.

We make the note that Model B fits the contextual whole of those related aspects which concern the client when using any of the 15 RP techniques covered in the literature sources for our GT study (see Table 2). This means that a client could use Model B to reason about his requirements prioritization context when using any of these techniques. Clearly, not all of the elements in Model B are necessarily present in each prioritization effort – i.e. some of them are optional depending on the project’s context or on the method used. For example, *Risk* (due to instability and highly volatile requirements) is usually a serious consideration in the later project iterations, for example when a large portion of the budget has already been consumed, or when a critical delivery deadline is approaching.

### 4.3. Theoretical sampling and saturating the concepts

This section briefly discusses the purpose of our theoretical sampling and how we carried it out. As per methodologists [8, 12, 17, 31], theoretical sampling means a quick and focused collection of pinpointed data once the researcher has a first set of conceptual categories to direct his/her theoretical sampling. In our study, we tentatively conceptualized relevant ideas which hinted to areas to probe with more information. We selectively looked for people and online forums on agile software development to shed light into what could be the boundaries of our conceptual categories. To get access to people, we used our own professional networks and agile-focused workshop venues (for example, the agile workshop co-located with the International Conference on Software Engineering in 2008 in Leipzig, where the authors presented the very first draft of the conceptual model in Fig 2). Specifically, we involved three practitioners from companies, when we were trying to figure out (i) how clients define, estimate and use size in agile RP context, (ii) how clients define business value, and (iii) how clients (or product owners) manage sprint backlogs in the context of agile projects in supplier networks. These practitioners (Luigi Buglione from large IT-solution providing company, Thijs Munsterman from a mid-sized agile software development company, and Erlend Engum from a small agile developing company, who helped out in understanding (i), (ii) and (iii)

respectively) brought insights into the variation in the meanings of our conceptual categories (*Size Measurement*, *Business Value*, and *Risk*). Checking our concepts against the empirical realities of the practitioners was instrumental to understand how, when, and why the meanings of our categories vary.

Similarly to this, our screening of published experiences in prominent agile blogs (for example [28]) and forums (for example [14]) contributed to the identification of those categories which we overlooked (for example *Project-level Constraints* and *Learning Experience*), or under-analysed (for example *Risk*).

We stopped our theoretical sampling process when we noticed that further acquisition of data from real project experiences 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 [8]. We however, acknowledge that this judgement about the point at which we stop the theoretical sampling might be subjective. Therefore, in the immediate future, we are planning case studies on real projects with companies in which we will use Model B as our framework to describe the contextual whole of the related aspects that concern the client when prioritizing the requirements at inter-iteration time.

## Evaluation of the GT results

Research methodologists [12, 17, 31] emphasize that when a researcher builds up a theory by using a qualitative approach as the GT, it makes more sense for the researcher to assess its resulting theory in terms of explanatory power than in terms of generalizability. As a conceptual model based on GT is always context-dependent (and this is reflected in the categories), methodologists do not propose that the GT findings are generalizable beyond the defined boundary of the study. To study explanatory power, we considered Glaser’s three key criteria for evaluating the emerging theory: adequacy, fitness (or relevance) and modifiability. Adequacy is to be assured by applying the set of techniques and analytical procedures in the GT, for example, adhering as closely as possible to the GT principles and processes, coding the data independently by each researcher before re-coding them in joint work discussions (in order to ensure the highest possible degree of inter-coder reliability), consulting literature to evaluate similarities and dissimilarities of the resulting theory to extend literature and to check for any category, property or property value that might have been overlooked. We made conscious effort to keep these GT principles, however, we must be clear on a validity concern arising from the fact that most of the time the two authors worked away from each other at two different locations and could not do much joint re-coding.

The relevance of the results 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, requirement prioritization) - either as researchers or as 'lay observers' - to make sense of their experience and to manage the situation better. To make sure we preserve the meaning of the clients in agile projects, we made the conscious choice to search and include the so-called 'in-vivo' codes, as recommended by Kathy Charmaz [8]. These are special terms from the world of the individuals involved 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 software development, were "backlog" (meaning those requirements in an agile project, which are subjected to the implementation - for the whole project, as well as for immediate future iteration, 'project backlog' and 'sprint backlog' respectively) and "sprint" (meaning an individual agile iteration in a project). We looked into the implicit meaning behind these terms and this in fact was what brought us to Model A on Fig 1. Another measure we took in order to keep our conceptual modelling effort in sync with real experiences was our consistent engagement in diagramming activity, details on which were presented in section 4.2.1. Beyond these two steps (using in-vivo codes and diagramming), in our immediate future research, we plan to demonstrate the fit of the framework by using it in case studies.

Furthermore, modifiability of an emerging theory is concerned with the possibility to update it and extend it in the future. 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 do think that if industrial uptake of agile software development practices increases and more knowledge on the client's role and the client-develop interaction modes becomes available, our framework will need some refinement and extension so that it's kept useful.

Last, we point out like other qualitative research approached, the GT approach implies the risk that the researchers assume that the conceptual categories are saturated, when they might not be. Following Charmaz [8], we remained open at all times to any new literature source and whenever we felt we were getting stuck, we stepped back and re-coded the earlier collected information and looked for new leads. We also looked at many cases of agile RP, while carrying out the theoretical sampling and 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.

## Conclusions

The contribution of this work is a conceptual framework which is a grounded theory explicating the requirements reprioritization in agile software development. This conceptual model fills a gap in the current agile software engineering and agile requirements engineering literature which lacks comprehensive studies on agile prioritization. Our conceptual model is a first proposal only. However, we think that it opens up for other researchers to explore the area and to accumulate support for - or a challenge to, the proposed theory. Our immediate future step is to carry out case study research in agile companies in the Netherlands.

## Acknowledgement

This research has been funded by the Netherlands Research Foundation (NWO) under the QUADREAD project and under CARES project. The authors would like to thank the practitioners Luigi Buglione, Thijs Munsterman, Erlend Engum, the colleagues Roel Wieringa, Klaas Sikkel, Klaas van den Berg, Siv Hilde Houmb, and the participants of the APSO workshop at ICSE 2008 for sharing ideas on the topic of agile requirements prioritization. We also thank the anonymous SEKE reviewers whose comments and suggestions brought us to an improved version of this paper.

## References

- [1] Agile Journal <http://www.agilejournal.com/>
- [2] Ahl, V. "An Experimental Comparison of Five Prioritization Methods." Master's Thesis, School of Engineering, Blekinge Institute of Technology, Ronneby, Sweden, 2005.
- [3] Alenljung, B, A. Person, Portraying the practice of decision-making in requirements Engineering: a Case Study of large Scale Bespoke Development, Requirements Engineering journal, 2008, 13, pp. 257-279.
- [4] Augustine, S., Managing Agile Projects, Prentice-Hall, 2005.
- [5] Beck, K. eXtreme Programming Explained: Embrace Change, Addison Wesley, 2000.
- [6] 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](http://www.agileadvice.com/archives/2006/03/methods_of_prio.html)
- [7] Cao, L, Ramesh B., Agile Requirements Engineering Practices: An Empirical Study, IEEE Software, Jan/Feb, 2008 pp. 60-67.
- [8] Charmaz, K. Constructing Grounded Theory: a Practical Guide through Qualitative Research, Thousand Oaks CA, Sage, 2007.
- [9] Clarke, A. Situational Analysis: Grounded Theory after the Postmodern Turn, Thousand Oaks, CA, Sage, 2005.

- [10] Cohn, M "Agile estimating and planning", Prentice Hall,20
- [11] Crow, K.,: "Customer-focused Development with QFD",  
URL: <http://www.npd-solutions.com/qfd.html>
- [12] Dey, I. Grounding Grounded Theory, San Diego Academic Press, 1999.
- [13] Dr Dobb's portal <http://www.ddj.com/>
- [14] Forum, Agile Journal,  
<http://www.agilejournal.com/forums>
- [15] Fraser, J., Setting Priorities, April 23, 2002, URL:  
<http://www.adaptivepath.com/ideas/essays/archives/000018.php>
- [16] Getting Started With Use Case Modeling, An Oracle White Paper, May 2007  
<http://www.oracle.com/technology/products/jdev/collateral/papers/10g/gswUseCaseModeling.pdf>
- [17] Glaser B. G., Basics of grounded theory analysis:emergence vs forcing, Mill Valley, Ca.: Sociology Press,1992.
- [18] Gottesdiener, E., At a Glance: Other Prioritization Methods, EBG Consulting, Inc. [www.ebgconsulting.com](http://www.ebgconsulting.com)
- [19] Harris, R. S., M. Cohn: Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects. XP 2006: pp. 175-180
- [20] InfoQ software development community  
<http://www.infoq.com/agile>
- [21] 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
- [22] Leffingwell, D., Widrig, D., Managing Software Requirements, 2nd ed. Boston, MA: Addison-Wesley, 2003
- [23] Li, C., Akker, J.M. van den, Brinkkemper, S. & Diepen, G. (2007). Intergrated Requirement Selection and Scheduling for the Release Planning of a Software Product. In Proc. of REFSQ '07, Spingre LNCS, pp. 93-108.
- [24] Patton ,Jeff. Ambiguous Business Value Harms Software Products, IEEE Software, 25(1) , January/February 2008.
- [25] Patton, J., Finding the forest in the trees, Conference on Object Oriented Programming Systems Languages and Applications, 2005, San Diego, CA, USA, pp: 266 – 274.
- [26] Principles behind the Agile Manifesto, 2001, URL:  
<http://agilemanifesto.org/principles.html>
- [27] Racheva, Z., M. Daneva, K. Sikkil, Value Creation by Agile Projects: Methodology or Mystery? to appear in the Proceedings of PROFES 2009 Conference, June 15-17, Oulu, Finland, Lecture Notes of Computer Science
- [28] Rally Dev Agile Blog <http://www.rallydev.com/agileblog>
- [29] Saaty, T.L., The Analytic Hierarchy Process, McGraw-Hill, New York, 1980.
- [30] Schwaber K., Agile Project Management with SCRUM, Microsoft Press, 2004.
- [31] Strauss, A.L., J.M. Corbin, Basics of qualitative research - grounded theory procedures and techniques, 6th print, Sage, Newbury Park, USA, 1991.
- [32] Tabaka, J., Collaboration Explained: Facilitation Skills for Software Project Leaders. Addison Wesley 2006.
- [33] Wiegers, K., "First Things First: Prioritizing Requirements," *Software Development*, 7(9) 1999.