

# A Conceptual Model of Client-driven Agile Requirements Prioritization: Results of a Case Study

Zornitza Racheva, Maya Daneva  
University of Twente  
Netherlands  
Tel.: + 31 53 489 4344/ +31 53 489 2889  
{z.racheva, m.daneva}@utwente.nl

Andrea Herrmann  
University of Braunschweig  
Germany  
Tel.: +49 531 391 2275  
andrea.herrmann@tu-braunschweig.de

**ABSTRACT** Requirements (re)prioritization is an essential mechanism of agile development approaches to maximize the value for the clients and to accommodate changing requirements. Yet, in the agile Requirements Engineering (RE) literature, very little is known about how agile (re)prioritization happens in practice. Conceptual models about this process are missing, which, in turn, makes it difficult for both practitioners and researchers to reason about requirements decision-making at inter-iteration time. We did a multiple case study on agile requirements prioritization methods to yield a conceptual model for understanding the inter-iteration prioritization process. The model is derived by using interview data from practitioners in 8 development organizations. Such a model makes explicit the concepts that are used tacitly in the agile requirements prioritization practice and can be used for structuring future empirical investigations about this topic, and for analyzing, supporting, and improving the process in real-life projects.

## Categories and Subject Descriptors

D.2.1 Requirements/Specifications – *Methodologies, agile*

## General Terms

Management,

**Keywords** - Agile development, requirements prioritization, inter-iteration decision-making process, case study

## 1. INTRODUCTION

Regular and value-driven requirements reprioritization from customer's perspective is key to the successful execution of agile software projects. A comparative study [4] of this process and the prioritization practices in "traditional RE" indicates that,

with respect to requirements (re)prioritization, agile requirements engineering (RE) is unique in two ways:

(1) (re)prioritization happens at inter-iteration time, and (2) (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, while exposing the project to as low a risk as possible. Surprisingly, researchers [3,9] in agile RE case studies found that the creation of software product value through requirements prioritization decision-making is only partly understood.

This paper sets out to answer the following research question (RQ): What are the key concepts that are considered when prioritizing requirements, according to agile project practitioners? We answer it by carrying out an exploratory multiple case study. This research represents a further step to contribute to the understanding of agile requirements reprioritization at inter-iteration time. The result of our effort is a conceptual model that is independent from any particular prioritization method, and which describes on an abstract level the agile prioritization process. The model provides a generic framework for describing the client's decision-making situation while prioritizing the requirements. As per Alenljung and Person [1], 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.

The paper is structured as follows: Sect. 2 presents our motivation, Sect. 3 introduces the research method and Sect. 4 describes the results of its application. Sect. 5 discusses validity threats. Sect.6 concludes the paper. We want to make the note that interested readers can find a more detailed description of the case study context, related work and threats to validity in [10].

## 2. MOTIVATION

Our motivation for creating a model of agile requirements prioritization (RP) from a client perspective rests on the fact that the practices of regular requirements reprioritization, with strong client participation, are a relatively recent phenomenon and, in turn, are only partially understood. The agile literature provides rather coarse-grained descriptions of the agile reprioritization process [11] only. We expected that by performing an exploratory study we could better capture and explicate what happens in reality. As the agile literature indicates, e.g. in [2], 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 task, the key task of prioritizing requirements, s/he must be aware of the facets of his/her role and thus would profit from a

clear model of the prioritization process available to him/her. We think that a conceptual model can help the client 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. We also think that our model would help those RE researchers who are interested in carrying out case studies to investigate how agile requirements decision-making happens in practice, to structure research questions and empirical data.

### 3. THE RESEARCH METHOD

We conducted an explorative multiple-case study, applying the Yin’s guidelines [14] and using semi-structured open-end in-depth interviews with practitioners from 8 agile software development organizations.

#### 3.1 The case study process and participants

Our case study is performed in 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 semi-structured interviews with practitioners according to the finalized questionnaire; (6) Sample and follow-up with those participants that possess deeper knowledge or a specific perspective.

The case companies characterized themselves as organizations that follow agile methodologies. Some of them did strictly follow Scrum principles such as daily stand-up meetings and release retrospective. Most of them, though, applied a combination of agile practices without sticking precisely to a specific agile software development or project management approach.

Each interview lasted 60 to 90 minutes. Each interviewee was provided beforehand with information on the research purpose and the research process. At the interview meeting, the researcher and the interviewee walked through the questionnaire which served to guide the interviews. The study included 11 practitioners who described a total of ten projects (two practitioners worked on the same project holding different roles). The application domains for which these practitioners developed software solutions represent a rich mix of fields including banking, health care management, automotive industry, content management, online municipality services, and ERP for small businesses. The information about the participating companies and specialists is summarized below:

- 1 middle size company in the Netherlands (2 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 German university (1 student project)
- 1 large consultancy in Italy (1 participant)
- 1 IT department in a large governmental organization in Turkey (1 participant)

Table 1 explains the primary role the case-study participants had in the studied projects.

**TABLE 1. Participants in the Interviews.**

Interviewee’s primary role	Number of interviewees
Project Manager	5
Developer	3
Product Owner	1
Client	1
Scrum Master	1
<b>Total Number of Interviewees</b>	<b>11</b>

#### 3.2 The data analysis strategy

In our case study, the data analysis was guided by the Grounded Theory (GT) method according to Kathy Charmaz [5], which is a qualitative approach applied broadly in social sciences to construct general propositions (called a “theory” in this approach) from verbal data. GT 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. In essence, this 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. Constant comparison means that the data from an interview is constantly compared to the data already collected from previously held interviews, until a point of saturation is reached, i.e., where new sources of data don’t lead to a change in the emerging theory (or conceptual model).

We first read the interview transcripts and attached a coding word to a portion of the text – a phrase or a paragraph. The ‘codes’ were selected to reflect the meaning of the respective portion of the interview text to a specific part of the RQ. This could be a concept (e.g. ‘value’, ‘method’), or an activity (e.g. ‘estimation’). We clustered all pieces of text that relate to the same code in order to analyze it in a consistent and systematic way. The results of the data analysis are presented in Fig. 1 and discussed in Section 4.

### 4. RESULTS

The multiple iterations of coding, constant comparing of information from the interviews, and conceptual modeling in our GT process yielded the model presented in Fig 1. Its purpose is to explicate and bring insights into the decision-making, which is the core of the RP process. The model takes the perspective of the client, unlike other RP authors [2,8,13] who adopt the perspective of the developers. This model is to help clients ‘zoom-in’ into the prioritization process and see those concepts which are important to consider in RP at inter-iteration time, including context. 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.

Our case study results suggest that there is a consensus among the practitioners that there are five aspects that the clients consider when making decisions on requirements priorities: *Business Value*, *Effort Estimation/ Size Measurement*, *Learning Experience*, *Input from the developers* and *External Change*. Iteration planning additionally considers *Project Constraints*.

Below we explain each of these conceptual categories, that are reflected in the model, 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 interview participants 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 that the case study participants converge on that the **Business Value** is 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. We termed this criterion **'negative value'**. 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 mentioned "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. 1). While the literature [11] 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.

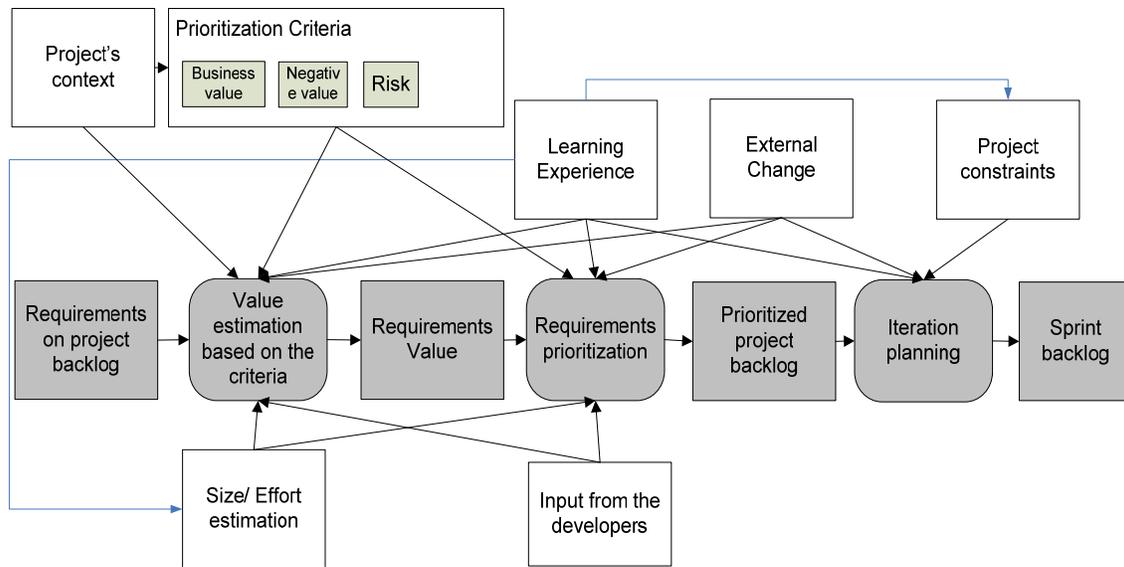


Figure 1.

The conceptual model

Learning is an in-built principle in agile development. Harris and Cohn [8] 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. 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 *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 sprint backlog as can be implemented within the next iteration and project constraints.

The iteration planning and the resulting sprint (i.e. iteration) backlog, is out of the scope of this paper and is shown on the model for sake of completeness.

As indicated earlier, the resulting model is compatible with any RP technique. It does not prescribe any process or propose a new method, but instead just describes what we found in the case study. This means that a client 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. The model's completeness still should be validated empirically, e.g. by new case studies.

## 5. THREATS TO VALIDITY

We make the note that in this paper we propose conceptual model **only**. This model, as suggested by GT methodologists [5,6], is not supposed to be validated against the data that has been used for the development of the model. According to GT methodologists [7,12], we can *only* evaluate the resulting model against the three evaluation criteria of GT: (i) adequacy, (ii) fitness (or relevance) and (iii) modifiability. We ensured adequacy of the result of the GT process by applying the set of techniques and analytical procedures in the GT. We adhered as closely as possible to the GT processes, coded the data independently by each researcher before re-coding them in joint work discussions. To ensure that the conceptual model makes sense to both researchers and practitioners, we searched and included the so-called 'in-vivo' codes, as recommended in [5]. These are special terms from the world of the practitioners in the studied context, which are assumed that everyone "knows and shares" them. 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), "project backlog", "iteration backlog".

Last, the modifiability of an emerging theory is ensured by the level of granularity that we chose for the model. 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.

## 6. SUMMARY AND OUTLOOK

This paper investigated the concepts that are important to consider when reprioritizing agile requirements from clients' perspective at inter-iteration time. We used GT to derive from case study data a conceptual model that explicates the RP in agile projects. This model fills a gap in the current agile RE literature, which lacks comprehensive studies on how agile RP happens in real life. The model presents the state of the practice described by concepts that we discerned from interviews with 11 practitioners. Our conceptual model is a first proposal only. It is descriptive, not prescriptive. However, we think that in its current state the model can be of value in at least two ways: (1) to help researchers and practitioners to identify and classify best practices and distill some guidelines for practice, and (2) to serve as a roadmap for further empirical research to investigate the fitness of different RP approaches to different contexts.

This study accesses fine-grained information that described the RP processes in companies. We found our fine-grained pieces of information about RP did not always agree with course-grained descriptions of the RP process in the literature. When we made practitioners' knowledge explicit in this case study, we found that there are variations in the meanings of important RP concept, for example, 'Priority Criterion' and 'Business value', and that certain variations are seemingly related to project context characteristics. This motivates us to do follow-up case studies in order to make the tacit practitioner's knowledge explicit.

## 7. ACKNOWLEDGMENT

This research has been funded by the Netherlands NWO under the QUADREAD project.

## 8. REFERENCES

- [1] 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.
- [2] Augustine, S., Managing Agile Projects, Prentice-Hall, 2005.
- [3] 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.
- [4] Cao, L, Ramesh B., Agile Requirements Engineering Practices: An Empirical Study, IEEE Software, Jan/Feb, 2008 pp. 60-67.
- [5] Charmaz, K. Constructing Grounded Theory: a Practical Guide through Qualitative Research, Thousand Oaks CA, Sage, 2007.
- [6] Clarke, A. Situational Analysis: Grounded Theory after the Postmodern Turn, Thousand Oaks, CA, Sage, 2005.
- [7] Glaser, B. G., Basics of Grounded Theory Analysis: Emergence vs Forcing, Sociology Press Mill Valley, 1992.
- [8] Harris, R. S., M. Cohn, Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects. XP 2006: pp. 175-180
- [9] 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.
- [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 Engineering 2010, Australia, in print
- [11] 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
- [12] Strauss, A.L., J.M. Corbin, Basics of Qualitative Research - Grounded Theory Procedures and Techniques, Sage, Newbury Park, 1991.
- [13] Wiegers, K., First Things First: Prioritizing Requirements, Software Development, 7(9) 1999.
- [14] Yin, R.K., Case Study Research: Design and Methods (1984)