Agile Requirements Prioritization: What Happens in Practice and What Is Described in Literature

Zornitza Bakalova[1], Maya Daneva[2], Andrea Herrmann[3], and Roel Wieringa[4]

[1,2,4] University of Twente, Computer Science Department, PO Box 217, 7500 AE, Enschede, The Netherlands
[3] Axivion GmbH, Stuttgart, Germany
{z.bakalova,m.daneva,.r.j.wieringa}@utwente.nl,
herrmann@axivion.com

Abstract. [Context & motivation] 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. **[Question/problem]** To gain better understanding of prioritization practices, we analyzed the real-life processes as well as the guidance that the literature provides. We compare the results of a literature research with the results of a multiple case study that we used to create a conceptual model of the prioritization process. We set out to answer the research question: "Which concepts of agile prioritization are shared in practice and in literature and how they are used to provide guidance for prioritization**."** **[Results]** The case study yielded a conceptual model of the inter-iteration prioritization process. Further, we achieved a mapping between the concepts from the model and the existing prioritization techniques, described by several authors. **[Contribution]** The model contributes to the body of knowledge in agile RE. It makes explicit the concepts that practitioners tacitly use in the agile prioritization process. We use this for structuring the mapping study with the literature and plan to use it for analyzing, supporting, and improving the process in agile projects. The mapping gives us a clear understanding of the 'deviation' between the existing methods as prescribed in literature and the processes we observe in real life. It helps to identify which of the concepts are used explicitly by other authors/ methods.

Keywords: agile development, requirements prioritization, conceptual model.

1 Introduction

In recent years, the agile methods enjoyed broad popularity and captured the attention of both the practitioners and the research community. Two of the key merits of these methods are the fast and early creation of value for the clients and the reduction of risk. This is ensured by practices that are specific attributes of the agile methods only, in particular the short iterations and the frequent respond to changes and learning during the project. The agile methods allow for frequent decisions about the

D. Berry and X. Franch (Eds.): REFSQ 2011, LNCS 6606, pp. 181–195, 2011.

requirements that will be considered for implementation at each iteration and in practice this is implemented by the process of requirements re-prioritization. As Gottesdiener [11] 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. As the agile literature agrees upon, e.g. [3],[6],[13], 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. Nevertheless, researchers [5], [16] in agile RE case studies found that the creation of software product value through requirements prioritization decision-making is only partly understood.

This paper presents a piece of work that is part of a series of studies about agile requirements prioritization. It builds upon our earlier publications [18],[20] in which we investigated the agile requirements prioritization (RP) process as described in literature [18], and as it happens in real-life projects [20]. This research now compares literature to practice and investigates and compares how complete and how detailed agile software engineering literature describes requirements reprioritization. This is a knowledge problem [26] aimed to identify the gap between practice and literature. We make the note that the key differences between our earlier work [18], [20] and this one consists in establishing the explicit mapping between the literature and the practice and in reasoning about its implications for research and practice.

This paper sets out to answer the following research question (**RQ**): "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 findings from a case study. In this paper we (i) first present a generic model derived from the case study and describe the conceptual categories that appear in it, and (ii) perform the mapping between these categories and existing techniques from literature, which is the main contribution of the paper.

This research represents a further step to contribute to the understanding of agile requirements reprioritization at inter-iteration time, and to assess the guidance the different RP methods provide. As per Alenljung and Person [2], a decision-making situation is "a contextual whole of related aspects that concerns a decision-maker", that is – in our case, the client or the product owner 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 the results, Sect. 6 is dedicated to validity threats, and Sect.7 concludes the paper.

2 Motivation

The practices of regular RP, with strong client participation, are a relatively recent phenomenon. In turn, they are only partially understood. Furthermore, the RP is an essential mechanism to maximize the business value (BV) for the clients and to accommodate changing requirements. We make the note that our previous study on BV creation brought us to think that we can not expect one universal definition of BV. In contrast, the notion of BV varies across projects and organizations, 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. It comes out of a human judgment that is based on competencies and deep knowledge of the client's domain and needs. For this reason, for the purpose of this study under BV we understand the client's perceived value of a requirement.

The reasoning in the previous paragraph motivated us for studying the agile RP process as it happens in real life and as described in literature. Moreover, we also made the observation that the agile literature [12],[13],[22] provides rather coarse-grained descriptions of the agile reprioritization process only [20], the literature is not complete and the RP methods are not described in such detail that a practicing software engineer can take them and immediately use them in his/her work. For example, we searched literature for specific information on how the process of value creation takes place in an agile project [17] and we could find no source that indicates how exactly this happens. In this work we investigate the guidance that the different RP methods described in literature provide to the decision-makers, by comparing literature sources with each other. We do this by using the concepts of our conceptual model as the common ground for our comparison. From this mapping and comparison, we can learn how complete and detailed is the published guidance.

Agile literature sources suggest, e.g. in [4], that never before in the software engineering history the customer has been that actively involved in the requirements reprioritization as he/she is in agile. However, our case study [19] indicates that in many cases 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. Given this backdrop, 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? Thus, 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 (e.g. 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 empirical research to investigate how agile requirements decision-making happens in practice, to structure research questions and empirical data. 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. This 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). (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.

3 The Research Method

In this section we provide a description of the case study that yielded the conceptual model. First, we conducted an explorative multiple-case study, applying the Yin's guidelines [27]. It included semi-structured open-end in-depth interviews with practitioners from 8 agile software development organizations. Second, we mapped the existing prioritization techniques to the categories identified in the case study.

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 questionnaire consisted of three parts: (i) questions referring to the prioritization practice in one concrete project; (ii) questions about the general prioritization practice in the company, based on the interviewees' experience; and (iii) questions about the role of value-consideration for (re)prioritization. Examples of the questions asked are: "Who performs the prioritization?", "What criteria do you consider?". 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
Total Number of Interviewees	**11**

3.2 The Data Analysis Strategy

In our case study, the data analysis was guided by the Grounded Theory (GT) method according to Charmaz [7]. It 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 5.

4 Results

4.1 The Conceptual Model

This section builds upon our previous work [18], where a preliminary result of our GT process has been presented. Here, we draw on this early result, extend it, elaborate - more in detail, the concepts involved, and discuss how we use the conceptual model to analyze the prioritization methods (Sect. 5.2).

Our 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 [4],[9],[12] adopting 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 seven aspects that the clients consider when making decisions on requirements priorities: 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. 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 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. 1). While the literature [3] 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 [13] advise *"Incorporate new learning often, in order to decide what to do next"*.

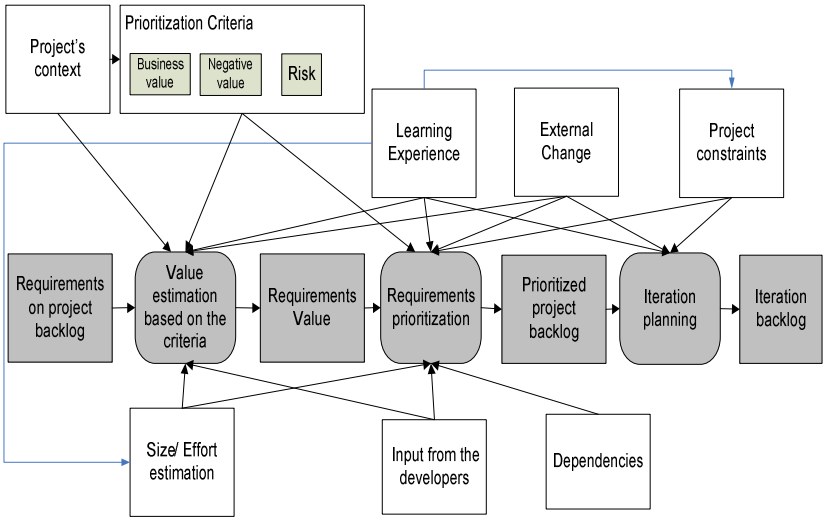

Fig. 1. Conceptual model of the agile prioritization process

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 following notes: First, we note that the iteration planning and the backlog of the follow-up iteration (i.e. the sprint backlog in the terms of some agile approaches), is out of the scope of this paper and is shown on the model for sake of completeness only. Second, we also note that in Fig 1, 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 the discussion on the nature of the relationships and the completeness of the set of relationships is outside the scope of this paper. Third, we traced the concepts back to the interview questions that we asked and the interview answers we collected. Because of space limitation, we do not provide information on this in this paper. However, we provide an illustration of this process by using the concept 'Negative value'. 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."

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

4.2 Mapping of the Existing Agile Prioritization Methods on the Model

In our previous work [20] 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 [20]. We, therefore, suggest interested readers either look into the [20], or contact the authors for receiving the complete list with references.

In this section we perform a mapping between the conceptual categories of the model in Fig. 1, 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 by reading the descriptions of the methods and

identifying those concepts that correspond to the ones in our model (Fig. 1). The result is presented in Table 2. Therein, the first column presents the 22 RP techniques. The other columns are named after the categories in the conceptual model in Fig. 1. 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 those RP method in the description of which we observe that the corresponding concept has been stated and used explicitly. Furthermore, in addition to the concepts that appear in the model, we have added an additional column S in Table 2 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 2 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 [12] 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."*

However, we found 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 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 [25] 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 2 we observe that:

1. 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 [9],[13] we assume this is incorporated rather implicitly in the methods, by means of their iterative nature and frequent decision-making cycles.
2. 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.

5 Discussion

Our observations in Table 2 confirm the finding discussed in our previously published paper [20], namely that the descriptions of RP techniques from the agile RE literature use mainly coarse-grained concepts. This becomes obvious when looking at Table 2, 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 [13] in agile process, no method describes how they should be considered. McDaniels and Small [15] plead for consensus-building that would lead to making decisions on requirements priorities. As per [15], 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 [21]. 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.

Table 2 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 [24], 'deliberation' implies equality

Table 2. Mapping between the concepts from the model in Fig. 1 and the prioritization methods

Concept → Prioritization method	Intuitive prioritization	Prioritization criteria, e.g. value, risk	Project context	Size/effort estimation	Input from developers	learning	external changes	Project constraints	Dependencies	Project backlog	value of requirement	Prioritized project backlog	Iteration/ Sprint backlog
Round-the-group prioritization	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											
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											
Eclipse Process Framework		y								y			y
Relative weighting		y											
Larman [9]		y											
Theme screening / scoring		y		y									
FDD		y								y		y	

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.

The implication for the 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.

6 Threats to Validity

We make the note that in this paper we propose a conceptual model. This model, as suggested by GT methodologists [7],[8], is not supposed to be validated against the data that has been used for the development of the model. According to GT methodologists [10],[23], 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, i.e. its fitness, we searched and included the so-called 'in-vivo' codes, as recommended in [7]. 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". Next, 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. The mapping of the conceptual categories with the existing prioritization methods used in practice shows that both the concepts themselves, as well as the level of granularity, are appropriate, as such mapping was possible and yielded meaningful results.

To minimize potential bias of the researcher, we considered also construct validity of our study. We followed Yin's [27] recommendations in this respect, by establishing a chain of evidence. First, the reports of the case study (e.g. partially published in [19]) showed clear links to the data, as well as reflected the link between the questions posed in the study protocol and the results. Second, we had a draft case study report reviewed by one key informant - one of the participants in the case study, who read and re-read multiple versions of the case study results. The third recommendation – using multiple sources of evidence, could not be implemented in the scope of our study, as interviews were the only source we consulted.

Furthermore, we make the note that we wanted to create a conceptual model about prioritization from the perspective of creating business value for the client, yet the majority of our interviewees were from development teams (e.g. we had only one client and one product owner who explicitly served as clients' proxy). We asked these professionals to put themselves 'in the shoes of the clients' when we discussed how agile prioritization creates clients' value. Nevertheless, we are conscious about that we obtained developers' perceptions only regarding the concept of clients' business value. It might be, therefore, well possible that if we had interviewed clients exclusively, we could have obtained some other categories in addition to those that we already have in the model. We consider this is an interesting study that concerns the expendability of our conceptual model and we plan it as research for the future.

The choice of the companies participating in the study could represent a threat to the validity of the results in a number of respects:

(i) As we are interested in the phenomenon 'agile prioritization', we want to be sure that this indeed is the context of the studied companies. We relied on the information provided by the companies' representatives and on our own observations, and 'mapped' them to the principles stated in the Agile Manifesto [1] in order to identify the agility of a company. The companies varied in respect to size, level of organizational rigor and hierarchy, and thus – in level of agility.

(ii) The choice of the companies was not motivated by any other criteria except the one – to be agile. The authors relied on their professional and personal network to establish contacts with the companies.

(iii) The studied projects are not representative for all the possible ways in which prioritization is performed in agile organizations. We, however, consider that our findings can be observable in companies and projects that have similar contexts to those included in our study only.

We make the note that while the conceptual model considers the perspective of the client, the analyzed literature treats prioritization from the development team's perspective. We, however, think that this does not pose an issue because we are aware that we investigate the client's perspective as it is seen with the developers´ eyes, because in our case study, the 10 out of 11 interviews were made with representatives of the development team.

Last, we make the note that although this study used the model produced by our previous study [18][20], we do not think that this represents a major threat to validity . The initial model was based on literature sources, authored by agile experts. For this reason we expected to find the same concepts in the interviews. We can expect that practitioners, who say they follow an agile methodology, are familiar with the literature and try to work in a way which is consistent with it and with the underlying concepts. Thus, the initial model cannot be regarded as preconceived ideas in the sense of the GT.

7 Summary and Outlook

This paper made two contributions: first it investigated the concepts that are important to consider when practitioners work on (re)prioritizing agile requirements at

inter-iteration time, and second, it mapped these concepts against 22 agile RP techniques described in literature. The results of our effort are, respectively: (1) a refined conceptual model which describes on an abstract, generic level, the concepts that seem to impact the agile prioritization process, and (2) a table with mappings between the concepts of the model and the methods as described in literature.

Our conceptual model was created by applying GT. The model explicates the RP in agile projects. It presents the state of the practice described by concepts that we discerned from interviews with 11 practitioners. The model provides a generic framework for describing the decision-making situation while prioritizing the requirements. We used it 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. It helps to identify which of the concepts that we identified are used explicitly by other authors/ methods. Furthermore, we identified clusters of methods and make a suggestion to use a discursive approach for those methods that rely on implicit, tacit knowledge.

We think that the results can be of value in at least two ways: (1) to serve as a roadmap for further empirical research to investigate the level of literature guidance on the decision-making, needed in different contexts, and (2) it can be used as a framework to provide better guidance to practitioners and allow for better motivated, discourse-based process.

Acknowledgments. We are indebted to the anonymous reviewers for their detailed and constructive comments that added to the improved quality of this paper. This research has been funded by the Netherland's NWO under the QUADREAD project.

References

1. Agile Manifesto, `http://agilemanifesto.org/principles.html`
2. Alenljung, B., Person, A.: Portraying the Practice of Decision-making in Requirements Engineering: a Case Study of large Scale Bespoke Development. Requirements Engineering Journal 13, 257–279 (2008)
3. Ambler, S.W.: Agile Modeling - Effective Practices for eXtreme Programming and the Unified Process. Wiley, New York (2002)
4. Augustine, S.: Managing Agile Projects. Prentice-Hall, Englewood Cliffs (2005)
5. Barney, S., Aurum, A., Wohlin, C.: A Product Management Challenge: Creating Software Product Value through Requirements Selection. Journal of Software Architecture 54, 576–593 (2008)
6. Beck, K.: eXtreme Programming Explained: Embrace Change. Addison-Wesley, Reading (2000)
7. Charmaz, K.: Constructing Grounded Theory: a Practical Guide through Qualitative Research. Sage, Thousand Oaks (2007)
8. Clarke, A.: Situational Analysis: Grounded Theory after the Postmodern Turn. Sage, Thousand Oaks (2005)
9. Cohn, M.: Agile Estimating and Planning. Prentice-Hall, Englewood Cliffs (2005)

10. Glaser, B.G.: Basics of Grounded Theory Analysis: Emergence vs Forcing. Sociology Press Mill Valley (1992)
11. Gottesdiener, AView To Agile Requirements, E EBG Consulting, Inc., http://www.ebgconsulting.com, http://ebgconsulting.com/Pubs/Articles/AViewToAgileRequirements-gottesdiener.pdf
12. Gottesdiener, E., At a Glance: Other Prioritization Methods, EBG Consulting, Inc., http://www.ebgconsulting.com, http://www.ebgconsulting.com/Pubs/Articles/At%20a%20Glance-Other%20Prioritization%20Methods-supplement-EBG%20Consulting.pdf
13. Harris, R.S., Cohn, M.: Incorporating Learning and Expected Cost of Change in Prioritizing Features on Agile Projects. In: Abrahamsson, P., Marchesi, M., Succi, G. (eds.) XP 2006. LNCS, vol. 4044, pp. 175–180. Springer, Heidelberg (2006)
14. McDaniels, T., Small, M.J.: Rsk Analysis and Society: an Interdisciplinary Characterization of the Field. Cambridge University Press, Cambridge (2004)
15. McDaniels, T., Small, M.J.: Risk analysis and society: an interdisciplinary characterization of the field, Timothy. Cambridge University Press, Cambridge (2003)
16. Petersen, K., Wohlin, C.: 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, 1479–1490 (2009)
17. Racheva, Z., Daneva, M., Sikkel, K.: Value creation by agile projects: Methodology or mystery? In: Bomarius, F., Oivo, M., Jaring, P., Abrahamsson, P. (eds.) PROFES 2009. LNBIP, vol. 32, pp. 141–155. Springer, Heidelberg (2009)
18. Racheva, Z., Daneva, M., 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), Bolzanao, Italy (September 2010)
19. Racheva, Z., Daneva, M., Sikkel, K., Herrmann, A., Wieringa, R.: Do we Know Enough about Requirements Prioritization in Agile Projects: Insights from a Case Study. In: The Proceedings of Requirements Engeneering 2010, Australia (2010)
20. Racheva, Z., Daneva, M., Herrmann, A., Wieringa, R.: 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. IEEE, Los Alamitos (2010)
21. Rippe, S.: Demokracy and Environmental Decision-making. Environmental Values 8, 75–98 (1999)
22. Schwaber, K.: Agile Project Management with SCRUM. Microsoft Press (2004)
23. Strauss, A.L., Corbin, J.M.: Basics of Qualitative Research - Grounded Theory Procedures and Techniques. Sage, Newbury Park (1991)
24. 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, pp. 35–86. Kluwer, Boston (1995)
25. Wiegers, K.: First Things First: Prioritizing Requirements. Software Development 7(9) (1999)
26. Wieringa, R.J.: Relevance and problem choice in design science. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 61–76. Springer, Heidelberg (2010)
27. Yin, R.K.: Case Study Research: Design and Methods, Thousand Oaks (1984)