Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework

Technical Report SWEHD-TR-2008-01

Andrea Herrmann, University of Heidelberg (Germany), Software Engineering Group
Maya Daneva, University of Twente (The Netherlands), Department of Computer Science

Version 1.0
5 March, 2008

**Software Engineering Group,
University of Heidelberg**

History

Version	Date	Change history
1.0	5 March 2008	Initial version

The „Software Engineering“ Group is part
Of the Institute of Computer Science of the
Ruprecht-Karls-Universität Heidelberg.
This group is led by
Prof. Dr. Barbara Paech.

Institut für Informatik
Neuenheimer Feld 348
69120 Heidelberg
paech@informatik.uni-heidelberg.de
http://www-swe.informatik.uni-heidelberg.de/

Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework

Andrea Herrmann, Maya Daneva

Abstract

In early phases of the software cycle, requirements prioritization necessarily relies on the specified requirements and on predictions of benefit and cost of individual requirements. This paper uses the Grounded Theory to investigate what is necessary to do requirements prioritization based on benefit and cost. The results of this analysis are activities and requirement properties. These form the basis for a framework for classifying requirements prioritization methods. This work is the first of a series of research activities aimed at the development of an agenda for future research in this area. It will be immediately followed by further literature research by using systematic review techniques.

1 Introduction

Requirements prioritization based on importance has been as old as software engineering itself. Much has been published on it and a number of requirements prioritization practices have been devised to increase its adoption in software organizations. The requirements engineering (RE) community knows multiple proposals for defining what the term 'importance' means. Two key factors are benefit and/or cost associated with each individual requirement [KR97], [Wie99]. Consequently, requirements prioritization can be supported by means of methods allowing the prediction of cost caused and benefit added by single requirements in a project.

Our middle-term objective is to sketch the problem of using benefit and cost information in support of requirements prioritization, to survey current solutions to this problem and their empirical evaluation, and to propose a research agenda to improve these solutions. We focus on requirements prioritization methods (RPM) used in early stages, when little is known about the architectural design and implementation of the requirement. (For later development phases, different benefit and cost estimation methods exist.)

We are set out to answer three research questions (RQ):

(1) What is needed for requirements prioritization based on benefit and cost estimates in early phases?

(2) Which of the activities of requirements prioritization based on benefit and cost estimation are currently supported by which published methods and how? and

(3) Which of these methods have been validated empirically so far and what was learnt from this validation?

We will approach RQ(1) by applying the Grounded Theory and plan to investigate RQ(2) and RQ(3) by using a systematic review of literature. The present report is our first report out of a series of several deliverables. Here, we treat RQ(1) and set the basis for our future literature research.

This report is structured as follows: Section 2 presents Grounded Theory and section 3 its application to RQ(1) with our main results: requirement properties and activities.

Section 4 analyzes requirements dependencies, their influence on requirement benefit estimation and how these are usually treated by RPMs. The results in section 3 and 4 are used in section 5 to define a framework for classifying RPM, particularly based on benefit and cost. Section 6 discusses threats to validity and section 7 concludes the report and describes future research.

2 Grounded Theory

The Grounded Theory which we deployed is the qualitative research method developed by Strauss and Corbin [StCo90] for systematically building theories in social research from qualitative data possibly drawn from case studies, from surveys and also from literature. As a research method, the Grounded Theory has two unique features: (i) that it is inductive in nature, which means that we as researchers have no preconceived ideas to prove or disprove and, thus, we are set to listen to the data rather than imposing preconceived ideas on the data, and (ii) that it relies on the concept of 'constant comparison', a process in which we constantly compare instances of data that we have named as a specific category with other instances of data, to see if these categories fit and are workable [Urqu01]. The Grounded Theory approach by Strauss and Corbin [StCo90] follows a constructivist research paradigm [Mil06], that is, (i) it stands for the collaborative co-construction of knowledge between the researcher and the participants in the studied phenomenon, and (ii) it assumes that there are multiple possible interpretations of the same data, each of which is potentially meaningful with respect to a specific context in which the phenomenon happens. This means, a researcher constructs theory as an outcome of the participants' interpretation on the participants' stories. (As Mills et al. [Mil06] indicate, the key issues for constructivist grounded theorists to consider in designing their research studies, is the process of developing a partnership with participants that enables a mutual construction of meaning during interviews and a meaningful reconstruction of their stories into a grounded theory model.) For us, the researchers, these underpinnings imply that we have to (i) recognize our bias and (ii) maintain objectivity as much as we can, when describing our position in relation to the data. This, in turn, has an implication on the validation plans (Section 6) for the theory which we would deliver by using this research approach.

The steps of the Grounded Theory by Strauss and Corbin [StCo90] are as follows:

1.) Setting the research question, which means determining what we specifically focus on and what we want to know about it ([StCo90], p.38);

2.) Reading technical and non-technical literature to stimulate theoretical sensitivity ([StCo90], p.41f);

3.) Open coding: identification of concepts, grouping them to categories, identifying the categories´ properties and dimensions;

4.) Questioning for enhancing theoretical sensitivity;

5.) Axial coding, which means to connect categories, utilizing a coding paradigm involving causal conditions, phenomena, context, intervening conditions, action/ interaction, and consequences;

6.) Selective coding, which means to identify the core category, to write the story and story line;

7.) Identification of process (linking of action/ interaction sequences) and contingency (unplanned happening);

8.) Transactional analysis by conditional/consequential matrix: "The conditional matrix may be represented as a set of circles, one inside the other, each (level) corresponding to different aspects of the world around us. In the outer ring stand those conditional features most distant to action/ interaction; while the inner rings pertain to those conditional features bearing most closely upon an action/ interaction sequence." ([StCo90], p.161);

9.) Theoretical sampling: sampling of empirical data, using the concepts and categories found earlier. The objective is the discovery of as many relevant categories as possible, along with their properties and dimensions.

These steps can be and usually are traversed iteratively several times, because: "Constant interplay between proposing and checking […] is what makes our theory grounded!" [StCo90]. That means, the analysis of the data collected in one step helps to check the interpretations from the previous step.

The next Section indicates how we applied steps 1-9 and what results we obtained.

3 Results of the Grounded Theory

Applying the Grounded Theory turned out to be an iterative learning process taking 9 months of continual literature research and discussion. In our study, our aim as we were reading literature sources was to compare literature to the emerging theory in the same was as one would compare data (for example data collected through interviews) to the emerging theory. For investigating RQ(1), we did not interact directly with participants of prioritization processes, what might have been possible in a case study. As we are interested in what methods and steps are needed, we instead read technical literature to analyze what types of prioritization methods have been used and what their typical characteristics are. (This is consistent with the position, which Strauss and Corbin take in [StCo90] with respect to the role of literature in Grounded Theory studies).

Below we describe how we executed each step and what our results were:

1.) We dealt with RQ(1), namely: "What is needed for requirements prioritization based on benefit and cost estimations in early phases?"

2.) We reviewed at least 500 literature sources. These were identified based on a two-phase literature research. These two phases and their iterative interaction with the steps of Grounded Theory will be described below, in the end of this section. The first author focused on benefit estimation and the second -- on cost estimation. Our sensitivity also resulted from 9.5 years of authors' collective experience in research and case studies in RE as well as 14.5 years of collective work experience in software development.

3.) Open coding: This step was supported by diagrams which were exchanged and discussed between both authors.. The core concept is the requirements prioritization process. This process consists of activities which are performed on the *requirement*. The requirement is characterized by the following properties relevant with respect to RQ(1): (i) *type*, (ii) *estimated benefit* to stakeholders, (iii) *estimated size* of software that embeds the requirement, (iv) *estimated cost* to build what embeds the requirement, (v) *priority*, and (vi) *requirement dependencies*. Herein, the property '*type*' means a pair of two orthogonal qualities: 'functional/non-functional requirement (FR/NFR)' and 'primary/secondary requirement'. The *type* of a requirement can be one of the following pairs 'primary FR', 'secondary FR', 'primary NFR' and 'secondary NFR'.

Primary requirements directly provide benefit to the stakeholders, while secondary requirements are derived from primary requirements and constrain them [PW04]. Only few authors distinguish between these two types of requirements explicitly, while many do it implicitly. Poort and de With [PW04] assume that primary requirements are usually FR and that secondary requirements can be both FR and NFR. However, in MOQARE (Misuse-Oriented Requirements Engineering) [HePa05], [HePa07], also primary NFR are known. Hence, we can assume that FR/ NFR and primary/secondary are properties which are orthogonal to each other.

Priority turned out to be an ambiguous concept, not only in practice, as was found in the case study of [LKK04]. We were surprised to find that RPMs usually do not define what "priority" means. Reviewing literature, we identified the following types of priority criteria: (i) benefit if the requirement is implemented, (ii) importance of the stakeholder defining the requirement, (iii) dissatisfaction if the requirement is not implemented, (iv) cost, (v) risk and (vi) dependencies among requirements.

Benefit, size and *cost* estimation for individual requirements in the early life cycle phases was found to be theoretically challenging, because of the multi-fold *dependencies* among requirements and their benefit respectively cost. One vehicle for studying requirements dependencies is the *benefit function*, which is under-utilized in software engineering [ErFH06], [HePa06a], but commonly used in Mathematical Economics [Scho02]. In section 4, we analyze and discuss requirements dependencies and how RPMs account for them.

Requirements can also be linked to each other by hierarchical relationships like decomposition and operationalization. Decomposition refers to the process by which a complex FR or NFR is broken down into sub-requirements that are more specific, easier to conceive and refine. An operationalization is a "possible design alternative for meeting NFR [or more generally: requirements] in the target system" [CNYM00]. The activity of operationalization, i.e. the derivation of operationalizations, follows the objective to present a requirement in verifiable/testable form.

4.) Questioning happened between the authors of this paper. The conceptual model was further questioned by the findings of our systematic review. All relevant literature sources had to fit into the classification framework resulting from the conceptual model (described in section 5).

5.) Axial coding: the context in which we treat requirements is requirements prioritization. It is embedded in the software planning in early phases. Requirements prioritization can be done by individuals or by a team. Intervening condition can be, for example, requirements change, change of priorities or change of priority criteria, or new knowledge gained. In the coding process, the term 'intervening condition' refers to any "structural conditions bearing on action/ interactional strategies that pertain to a phenomenon. They facilitate or constrain the strategies taken within a specific context." [StCor91]. What in Grounded Theory is referred to as 'the process', are the requirements prioritization activities, and action/ interaction, which in our analysis corresponds to method input/output. Consequences of requirements prioritization are decisions which are based on priorities, like project management decisions, design decisions and test decisions.

5.) Axial coding, 6.) selective coding and 7.) process and contingency identification in our study didn't need to be as sophisticated as it is in sociological studies [StCo90]. As the core concept in this study is the requirement, we looked for actions (or RE *activities)* which may be performed on each requirement as part of their prioritization. The actions´ consequences as well as the "story" are modelled in an activity diagram (Fig. 1). We specifically acknowledge those activities (see the ovals in Fig. 1) which determine one of the properties of a requirement, but not necessarily of all requirements together. For example, the activity 'Estimate size (of requirement)' means determining the size of software it would take to realize the requirement. Activities lead individual requirements from one state to the other. We found that each activity corresponds to a specific type of methods.

We make an important remark about the prioritization activity. Its separation from benefit and cost estimation is not artificial; in fact, we found that existing RPMs usually are not specific for a certain prioritization criterion and do not offer any support for benefit or cost estimation. Instead, they manage the process of determining the requirements´ priorities by defining how requirements, their importance values and dependencies are treated, e.g. whether requirements are treated individually or pair-wise.

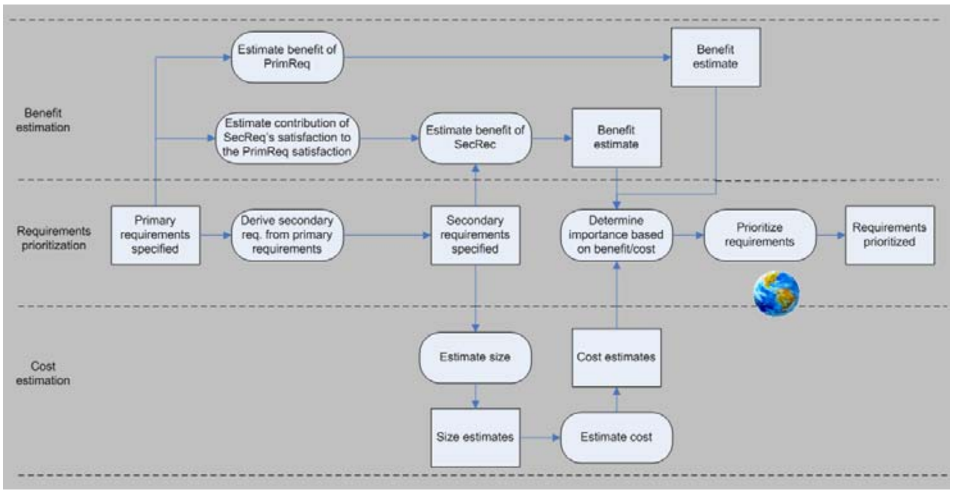

Fig. 1: Activity diagram depicting requirement states and activities during requirements prioritization based on benefit and cost estimation

8.) Transactional analysis has not been done for this study yet. At this stage of our research, we found it unnecessary as long as we treat only issues concerning the methods used. In the immediate future, we plan to also treat various aspects of the decision-maker´s situation in RE [AlPe04], e.g. organizational context and decision-maker´s characteristics. This extended research scope will, then, demand a transactional analysis. In this paper, we treat what is called in [AlPe04] ‘decision input’, ‘output’, and ‘decision-making activities’.

9.) Theoretical sampling was done by means of a systematic review of literature. (Though in the future, we plan to do it by applying requirements prioritization based on benefit and cost estimation in case studies.) During step 9, we checked the completeness of both the concepts and the concept properties identified in step 3, but also investigated which of them can be used for classifying published methods. The question, which of the concepts and their properties are useful to characterize RPMs, will become relevant with respect to RQ(2) and its answer is the classification framework presented in Section 4.

As was said earlier, the Grounded Theory process was applied iteratively. During our first phase of literature research, recent publications in conference/workshop proceedings and journals were analyzed, including their “related work” sections to trace back to primary studies. During step 2, publications about related issues were also analyzed, e.g. literature sources about quality analysis methods which evaluate architectural designs, and about collaborative RE. These readings helped us to define the scope of our study and to refine the RQs. Based on our reading of about 400 publications, the Grounded Theory was performed. The resulting classification framework was used to refine the RQ(2) and the search strings for a systematic review. This systematic review in a second phase of literature research identified further 100 publications which we deemed relevant for our research questions RQ(1) and RQ(2). Those publications which satisfied our inclusion criteria as described below, were classified according to this framework. This classification served the theoretical sampling (step 9), which tested the completeness of the conceptual model and of the classification framework. Thus, the classification framework was finished up only after the systematic review, and the framework helped us re-structure the systematic review results.

Both phases of literature research yielded a total of 240 papers which met the following seven quality criteria for inclusion in the review:

(1) the paper is on a RPM which treats individual requirements and includes estimation of cost and/or benefits for each individual requirement (and not for the system as a whole).

(2) the paper is credible, i.e. the method described is meaningful and intuitive to follow,

(3) relevance for practice: the method is useful, i.e. it potentially offers support for practical requirements prioritization,

(4) the paper adequately describes the context, i.e. there are enough details on the context in which the method is expected to be applicable.

(5) original paper: for each method, we searched its original publication; if this source does not describe a method clearly enough and in sufficient detail for readers to execute it, we cited a more comprehensive description;

(6) accessibility: if an original paper is difficult to access, or is outside the RE field, we cited another description from a RE author;

(7) cited by others: Additionally to reviewing work published in the above journals and conferences, we followed some references in the papers found to trace the original paper. We considered the fact that a work is cited and used by others as a hint on its usefulness.

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

4 Requirements Dependencies and Benefit Prediction

Requirements dependencies, although important in practice [RK97], are discussed by only few requirements prioritization authors, like [WA06], [Dav03], [PSR04], [LKK04]. Unlike other requirement properties (as priority or estimated cost), regarding the dependencies among requirements, we found the following: (i) they do not describe a property of one requirement, but a property of the relationship between at least two requirements, (ii) we could not find an activity or method which is specifically designed for coping with dependencies. Instead, our finding is that dependencies are implicitly treated by RPMs, and the way each method does this is characteristic to it. This led us to classify the RPMs from our review according to how they treat requirements dependencies. For justifying and developing criteria for our classification, we here include a mathematical model of how to reason about requirements benefit, based on the idea of a benefit function. (Analogous reasoning about cost estimation and all other prioritization criteria would lead to similar results.) We assume that a *benefit function* *B*(S) models the benefit provided by an IT system S in which a certain number out of *N* candidate requirements is realized, while others are not [HePa06a]. *B(S)* depends on S, not on its history. These are general assumptions without unnecessary restrictions.

Given RQ(1), we are interested in the benefit produced by single requirements. The incremental benefit of a single requirement A can be defined as $b_A(S) = B(S_A) - B(S)$, i.e. the benefit gain when adding requirement A to system S. This value depends on S; that is, on all other requirements which are assumed to be implemented in S. The incremental benefit is not additive, i.e. the benefit of a group of requirements is not the sum of the incremental benefits of the requirements. These are not only the most general assumptions, but also a consequence of dependencies among requirements. The explicit requirement benefit dependencies means, in turn, that benefit estimation for a single requirement only makes sense relative to a clearly defined system S, which is an idea of an ensemble of requirements which are supposed to be realized, and which we call the reference system [HePa06a]. It can be the status quo [XMC04], the mandatory requirements [REP03] or the perfect system where all requirements are assumed to be implemented [AHPR04].

The form of *B*(S) models all dependencies completely. Hence, when *B*(S) is known, all incremental benefits $b_A(S)$ (for all S and A) can be calculated. However, usually *B*(S) is unknown. Project realities pose many difficulties to the complete estimation of such a complex benefit function, for example: (i) a reference system must be explicitly defined (but not implemented) and communicated to all benefit estimators. This difficulty can be overcome by prioritizing requirements which are independent of each other (like in the experiments of [KBR04], [KTRB06]). (ii) the validity of estimations is limited to the specific reference system being used. Iterative re-estimations may be necessary for other reference systems. (iii) scaling-up a benefit estimation method might be difficult, as [BB05] observe that “In large, multi-attribute domains, it can be cognitively unmanageable for a user to compare full outcomes involving more than a handful of attributes”. (iv) many biases are known to impact predictions [TK74]. In practice, the benefit which is expected for adding a requirement was found to differ from the expected benefit loss when this requirement is not implemented [RR99]. However, both should be equal to $B(S_A) - B(S)$.

Some further challenges in estimating have been identified in [BF00]: (v) unclear assumptions underlying the estimation – which calls for deploying interval estimates to account for uncertainties, (vi) when asked for past project data, few experts can recall historical numbers, (vii) estimation is a skill and needs training to master it, but most firms rarely have time for it.

Consequently, for pragmatic reasons, one can not expect the estimators to determine a complete benefit function. Practical benefit estimation must be executed in a simpler way. Our review of the requirements prioritization literature shows that six types of approximations are usually made, each with its specific advantages and disadvantages:

1) Each requirement´s importance is assumed to be fixed, independently of any reference system and summable: The advantage of this simplification is that estimations need to be done only once and that the benefits of requirements can be added, for instance: $B(S) + b_A + b_B = B(S_{AB})$. This approach disregards all dependencies among requirements. This is commonly done by state of the art RPMs.

2) Grouping requirements: Requirements are grouped into bundles based on a prioritization criterion in a way that each group is approximately independent of the others. Such groups are used by many RPMs (without theoretical justification, though), and are called features [RHN01], [Wie99], [NeTh02], [ZhMZ06], feature groups [RHN01], super-requirements [Dav03], classes of requirements [REP03], bundles of requirements [PSR04], categories [XMC04], User Story [Be00], super attributes [SKK97] or Minimum Marketable Features [DC03]. The benefits of requirements groups then are additive [BB05]. This simplification accounts for the most important dependencies and disregards all others. The groups can be built on different levels to form a hierarchy of requirements [KOR97], which in turn reduces the complexity of the benefit estimation task because one first gets estimations for the requirements groups relative to each other and, then, for the requirements within each group (like in [REP03]).

3) Using relative values instead of absolute: Should benefit be compared to cost, it is ideal to estimate benefit in a monetary terms (e.g. in $US). Work hours saved is also a measure which can be compared to implementation work hours. However, often such absolute estimations on a ratio scale are difficult due to high uncertainties. Therefore, relative values are preferred for estimation on an interval scale, where the difference between each consecutive pair of numbers is an equivalent amount, but there is no real zero value. Relative values are also known to be easier to estimate then absolute [Karl96].

4) Pair-wise comparison: One group of RPMs attributes one value per requirement, while the other group determines the relative value by pair-wise comparison.

5) Using discrete values instead of a continuous scale: This means that the importance (or benefit or cost) values are not estimated in real numbers, but only a finite number of values are used. This can be an ordinal scale which ranks the requirements by their order of importance or a nominal scale like the values 1-2-3, where the numbers signify names of categories (and have no numeral significance, that is 3 points does not mean three times the benefit as 1). Such nominal scales can be 1/2/3, or 1/2/3/4/5, or low/ medium/ high [Wie99], [LKK04], essential/ conditional/ optional [IEE98], [Wie99], mandatory/ desirable/ inessential [Karl96],[Be00], must/ need not/ needs more attention [Dav03]. These values depend less on the reference system If necessary, requirements of the same ordinal rank are prioritized against each other in a second step.

6) Building intervals b_A*€[,]*: Some authors advocate that intervals be used for the estimation, for instance by doing an optimistic, realistic and pessimistic estimation [Dav03], [BF00]. The merit of using intervals is that it can capture uncertainties of different types, e.g. of the reference system (when it's only partially known or slightly changed), of some external parameters, and unknown or differing perspectives of different stakeholders. We noticed, though, that in RE literature, different types of uncertainty are not distinguished.

The above six types of approximations in section 5 are used to design a classification framework that should help position the existing RPMs with respect to how each one treats requirement dependencies.

5 Classification Framework for Methods

Our results of the Grounded Theory study were used to develop a classification framework capable of structuring - according to a set of factors, the existing methods published in literature. In this section, we introduce the classification factors that make up the framework, which we later will use for classifying results of literature research.

In our framework, we classify methods existing in the RE literature on the basis of the *activity* which they support (Fig. 1). We chose this classification criterion for two reasons: (i) a method adds value by being integrated into the activities it is supposed to support, (ii) most methods, as we will see later on, focus on one and only one activity.

For different types of methods, we furthermore use the following classification factors (given in italic):

Benefit and cost estimation methods are characterized by the *type of requirements* they take as input. The input can be FR or NFR, primary or secondary requirements.

RPMs are characterized with respect to how they treat requirement *dependencies*. Six approximations have been described in section 4. This corresponds to six factors which indicate whether or not a RPM applies this approximation. .

In the RE literature, we also observed that RPMs are distinguished according to the following additional criteria:

- time consumption total or per decision [Karl96], [KaWR98], [KBR04], [Karl06]
- number of comparisons or decisions to execute [Karl96], [KaWR98],
- standard deviation of the priorities for the same requirement [Karl96],
- perceived trustworthiness of the method or subjective reliability of results [Karl96], [KaWR98]
- ease of use [KaWR98], [KBR04], [Karl06], [SaKo06]
- fault tolerance [KaWR98]
- accuracy [KBR04], [Karl06]
- granularity (fine/ medium/ coarse), sophistication (complex/ easy) [BeAn05]
- takes into account multi-views [SaKo06]
- notation (textual explanation, mathematical formula, function) [SaKo06]
- tool to indicate if a COTS is available to support the method [SaKo06]

These criteria all are relevant when choosing a RPM. However, these are not relevant with respect to our RQ(1), and according to the decision-making framework of [AlPe04], these do not describe activities, input or output of decision-making. Therefore, we do not use them here.

Clearly, readers may argue that there are criteria which might sound relevant for classifying a certain type of method and which we did not include. Our motivation for not using these criteria rests on the result of the literature research, which showed that existing methods can not be distinguished with respect to these criteria. More in detail, we found that RPMs could not be classified according to the following criteria:

- the prioritization criterion which the authors of the RPM suggest be used (benefit, cost or others). We found that most methods can use any prioritization criterion.
- the input into the RPM. In our literature research, we observed that whether RPMs allow FR or NFR as input or primary or secondary requirements was not essential. The RPM found during the literature research can all be applied to all types of requirements.

6 Threats to Validity

We considered the possible threats to validity and took measures to counterpart them. A key validity concern is the degree to which the set of classification factors is complete. We consider it complete if the factors sufficiently account for the major differences between the methods found. We judged the completeness of the framework when using it for classifying the results from a systematic review of literature, which we undertook recently for investigating RQ(2). The review is on the same topic, namely the use of benefit and cost information in support of requirements prioritization. It will be published separately and is in the process of being finalized. However, we do have some preliminary observations from this first experience in using the framework: we indeed found, that the classification framework well supported the classification of the literature sources. Both authors used it without any need of additional concepts, properties or property values to be added to the set of factors. We, however, acknowledge that this judgment is subjective. The factors could have been more fine-grained, as can be concluded from the observation that methods falling in the same group still differ. It is possible that researchers posing different research questions, might want to include further factors to our classification framework.

Second, the 'relevance' of the papers included in the Grounded Theory analysis (as well as in the SR) could be put in question as: (i) the first author was the only reviewer of benefit-based requirement prioritization papers and (ii) the second author was the only reviewer for cost-based requirements prioritization papers, thus it was not assessed whether each author's tabulation and application of the selection criteria are correct. Because of resource constraints, working this way was the only viable option to us at the time. Some papers, for which classification turned out to be difficult, were read and discussed by both authors. We are aware of approaches by other researchers [JoSh07] who do an individual classification by several researchers and then discuss the differences in each classification proposal by tracking rates of inter-researchers' agreement. However, this approach demands much time because all researchers must read all papers.

Another validity threat is that methods might have been classified erroneously and, in the worst case, this could lead to incompleteness of the concepts used. That there is no standardized terminology used in RE was a challenge.

We also considered Glaser's criteria [Glas92] for judging the credibility of an emerging theory that comes out of GT research efforts. Glaser (as well as other GT authors [Este07]) put forward three key criteria for judging 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. Clearly, a validity concern arises from the fact that the two authors could not do much joint re-coding due to their limited resources (as mentioned earlier in this section).

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. We plan, as an activity in the near future, to demonstrate the fit of the framework by using it in a SR process and in case studies.

Furthermore, modifiability of an emerging theory is concerned with the possibility to update it and extend it in the future. We chose deliberately to keep our framework open and, in our view, it makes more sense to invite other researchers to use it and test it, then to strive for all-inclusive and general results. We believe that if industrial uptake of requirements prioritization practices increases, the framework we propose in this report will need some refinement/extension so that it's kept useful in the course of time.

7 Summary and Future Work

This paper investigates what is needed for requirements prioritization based on early benefit and cost estimations. This research was done by applying Grounded Theory. Its results were used to derive a classification framework for methods which are used in the context of requirements prioritization based on benefit and cost estimation.

This paper is the first of a series of several papers aimed at increasing the understanding of how benefit and cost estimates are used in support of state-of-the-art requirements prioritization. The papers we plan in the immediate future will describe the results of systematic literature reviews about requirements prioritization. Next, we will investigate how existing methods support the needs described in this paper, which means the activities and different combinations of classification criteria. This will also be done by a systematic review. Finding these answers will serve the objective to identify a research agenda on requirements prioritization based on benefit and cost estimation. Two further future steps are planned to augment and/or refine the agenda: (i) we are interested to know which methods have been validated empirically and how. This is to add to our research agenda items, which are pertinent to empirical research; and (ii) our application of the Grounded Theory so far exclusively treated questions concerning method support. A transactional analysis as discussed in section 3 for RQ(1) still remains to be done and will lead to further issues, e.g. concerning the role of the organization and stakeholders in the prioritization. We are also interested in looking into companies' sites and their requirements prioritization practices and using case study research methods to demonstrate the adequacy, the relevancy and the modifiability of our framework.

8 References

[AHPR04] Arora, A., D. Hall, C.A. Pinto, D. Ramsey, and R. Telang, *An ounce of prevention vs. a pound of cure: How can we measure the value of IT security solutions?*, Carnegie Mellon CyLab, 2004.

[AlPe04] Alenljung, B., A. Persson, "Supporting requirement-based decision-making in the software engineering process: A position paper", *10th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2004)*, B. Regnell, E. Kamsties, and V. Gervasi, eds., Essener Informatik Beiträge Band 9, Universität Duisburg-Essen, Essen, Germany, pp. 63-68.

[BeAn05] Berander, P., A. Andrews, Requirements Prioritization, pp. 69-94, in: Aybüke Aurum, Claes Wohlin (Eds.), *Engineering and Managing Software Requirements*, Springer, Berlin, Heidelberg, Germany 2005.

[BB05] Braziunas D., C. Boutilier, "Local Utility Elicitation in GAI Models", *Proc. 21th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-05)*, AUAI Press, Arlington, Virginia, 2005, pp. 42-49.

[Be00] Beck, K., *Extreme programming explained*, Addison-Wesley, Upper Saddle River, USA, 2000.

[BF00] Boehm B.W., R. E. Fairley, "Software Estimation Perspectives", *IEEE Software,* Nov/Dec 2000, pp.22-26.

[CNYM00] Chung, L., B.A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Kluwer Acadamic Publishers, 2000.

[Dav03] Davis, A.M. "The Art of Requirements Triage", *IEEE Computer*, vol. 36, no. 3, March 2003, pp. 42-49.

[DC03] Denne, M., and J. Cleland-Huang, *Software by Numbers: Low-Risk, High-Return Development*, Prentice-Hall, 2003.

[Dey93] Dey, I. *Qualitative Data Analysis: A user-friendly guide,* Routledge, London, 1993.

[ErFH06] Erdogmus, H., J. Favaro, M. Halling, "Valuation of Software Initiatives Under Uncertainty: Concepts, Issues, and Techniques", *Value-Based Software Engineering*, S. Biffl, ed., Springer, Berlin, Heidelberg, Germany, 2006, pp. 39-66.

[Este07] Esteves,J., I. Ramos, J.Carvalho, Use of Grounded Theory in Information Systems Area: an Exploratory Analysis, *European Conference on Research Methodology for Business and Management*, 2007, pp. 129-136.

[Glas92] Glaser B. G., *Basics of grounded theory analysis: emergence vs forcing*, Mill Valley, Ca.: Sociology Press, 1992.

[HePa05] Herrmann, A., B. Paech, "Quality Misuse", *Proc. Workshop on Requirements Engineering for Software Quality, Foundations of Software Quality (REFSQ 2005)*, Essener Informatik Berichte, Essen, Germany, 2005, pp. 193-199.

[HePa06a] Herrmann A., B. Paech, "Benefit Estimation of Requirements Based on a Utility Function", *Proc. Workshop on Requirements Engineering for Software Quality, Foundations of Software Quality (REFSQ 2006), Essener Informatik Beiträge, Band 11*, Essen, Germany, pp. 249-250.

[HePa07] Herrmann, A.,B. Paech, "MOQARE: Misuse-oriented Quality Requirements Engineering", RE Journal, vol. 13, no. 1, Jan 2008, pp. 73-86.

[IEE98] IEEE, *Std. 830-1998: IEEE Recommended Practice for Software Requirements Specification*, IEEE, Washington, 1998.

[JoSh07] Jørgensen, M., M.J. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies", *IEEE Trans. Software Eng., vol.* 33, no. 1, 2007, pp. 33-53.

[Karl96] J. Karlsson, "Software requirements prioritization", *Proc. 2nd Int'l Conf. Requirements Eng.*, 1996, pp.110-116.

[Karl06] Karlsson, L., *Requirements Prioritisation*, SERG Software Engineering Research Group, Lund University, Sweden, 2006, http://serg.telecom.lth.se/research/packages/ReqPrio/ (last updated: 9 Aug 2006, last visit: 21 Jan 2008)

[KaWR98] Karlsson, J., C. Wohlin, and B. Regnell, An evaluation of methods for prioritizing software requirements, *Information and Software Technology*, vol. 39, 1998, pp. 939-948.

[KBR04] Karlsson, L., P. Berander, B. Regnell, and C. Wohlin, „Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning", *Proc. 8th Int'l Conf. on Empirical Assessment in Software Eng. (EASE 2004)*, Edinburgh, Scotland, pp. 145-154.

[KOR97] Karlsson, J., S. Olsson, K. Ryan, "Improved Practical Support for Large-scale Requirements Prioritisation", *RE Journal*, vol. 2, no. 1, 1997, pp.51-60.

[KR97] Karlsson J., K. Ryan, "A Cost-Value Approach for Prioritizing Requirements", *IEEE Software* 14(5) 1997, pp. 67-74.

[KTRB06] Karlsson,L., T. Thelin, B. Regnell, P. Berander, and C. Wohlin, "Pair-wise comparisons versus planning game partitioning-experiments on requirements prioritisation techniques", *Empirical Software Eng.*, vol. 12, no. 1, 2006, pp. 3-33.

[LKK04] Lehtola, L., M. Kauppinen, and S. Kujala, "Requirements Prioritization Challenges in Practice", *Proc. 5th Int'l Conf. On Product Focused Software Process Improvement (PROFES)*, 2004, pp.497-508.

[Mil06] Mills J., A. Bonner, K. Francis, The development of Constructivist Grounded Theory, Internatoional Journal of Qualitative Methods, 5(1), March 2006.

[NeTh02] Nejmeh B., I. Thomas, "Business-Driven Product Planning Using Feature Vectors and Increments", *IEEE Software*, vol. 19, no. 6, 2002, pp. 34-42.

[PSR04] Papadacci,E., C. Salinesi C. Rolland. "Payoff Analysis in Goal-Oriented Requirements Engineering", *Proc. Workshop on Requirements Engineering for Software Quality, Foundations of Software Quality (REFSQ)*, 2004.

[PW04] Poort E.R., P.H.N. With, "Resolving Requirement Conflicts through Non-Functional Decomposition", *Proc. 4th Workshop on Software Architecture*, 2004, pp. 145-154.

[REP03] Ruhe, G., A. Eberlein, and D. Pfahl, "Trade-Off Analysis For Requirements Selection", *Int'l Journal of Software Eng. and Knowledge Eng*., vol. 13, no. 4, 2003, pp. 345-366.

[RHN01] Regnell,B., M. Höst, J. Natt och Dag, P. Beremark, and T. Hjelm, "An Industrial Case Study on Distributed Prioritisation in Market-Driven Requirements Engineering for Packaged Software", *Requirements Eng.,* vol. 6, 2001, pp. 51–62.

[RK97] Ryan K., J. Karlsson, "Prioritizing Software Requirements in an Industrial Setting", *Int'l Conf.on Software Eng.*, 1997, pp. 564-565.

[RR99] Robertson, S., and J. Robertson, *Mastering the Requirements process*, Addison-Wesley, 1999.

[SaKo06] Salinesi, C., and E. Kornyshova, Choosing a Prioritization Method - Case of IS Security Improvement, *18th Int'l Conf. on Advanced Information Systems Engineering* CaiSE 2006, June 5-9, Luxemburg, Forum Proceedings, pp. 51-55.

[SKK97] Stylianou, A.C., R.L. Kumar, and M.J. Khouja, "A total quality management-based systems development process", *ACM SIGMIS Database*, vol. 28, no. 3, June 1997, pp. 59-71.

[Scho02] Schofield, N., *Mathematical Methods in Economics and Social Choice*, Springer, 2002.

[StCo90] Strauss, A.L., and J.M. Corbin, *Basics of qualitative research - grounded theory procedures and techniques*, 6th print, Sage, Newbury Park, USA, 1991.

[TK74] Tversky, A., D. Kahneman, "Judgment under uncertainty: Heuristics and biases", *Science*, vol. 185, 1974, pp. 1124–1131.

[Urqu01] Urquhart, C. "An Encounter with Grounded Theory: Tackling the Practical and Philosophical Issues", In: E. Trauth (ed.), *Qualitative Research in Information Systems: Issues and Trends*, Idea Group Publishing, 2001, pp. 104-140.

[WA06] Wohlin C., A. Aurum, "Criteria for Selecting Software Requirements to Create Product Value: An Industrial Empirical Study", *Value-Based Software Engineering*, S. Biffl, ed., Springer, Berlin, Heidelberg, 2006

[Wie99] Wiegers, K., "First things first: prioritizing requirements", *Software Development*, vol. 7, no. 9, September 1999.

[XMC04] Xie, N., N. R. Mead, P. Chen, M. Dean, L. Lopez, D. Ojoko-Adams and H. Osman, *SQUARE Project: Cost/Benefit Analysis Framework for Information Security Improvement Projects in Small Companies*, Technical Note CMU/SEI-2004-TN-045, Software Engineering Institute, Carnegie Mellon University, 2004.

[ZhMZ06] Zhang, W., H. Mei, H. Zhao, "Feature-driven requirement dependency analysis and high-level software design", *RE Journal*, vol. 11, no. 3, June 2006, pp. 205-220.

Document Information

Title	Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework
Date	05 March 2008
Version	1.0
Status	
Distribution	http://www-swe.informatik.uniheidelberg. de/research/publications/reports.htm