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

Maya Daneva, Andrea Herrmann
m.daneva@utwente.nl, andrea.herrmann@iese.fraunhofer.de

Abstract

In early phases of the software development process, requirements prioritization necessarily relies on the specified requirements and on predictions of benefit and cost of individual requirements. This paper induces a conceptual model of requirements prioritization based on benefit and cost. For this purpose, it uses Grounded Theory. We provide a detailed account of the procedures and rationale of (i) how we obtained our results and (ii) how we used them to form the basis for a framework for classifying requirements prioritization methods.

Key words: Requirements Prioritization, Software Economics, Non-Functional Requirements, Quantitative Prioritization, Grounded Theory

1. Introduction

Requirements prioritization based on importance has been as old as software engineering itself. Knowing the most important requirements for a software product is key to any software product improvement actions and gives an increased certainty that one is building in the most important functionalities and qualities, for example those with the highest benefit-cost-ratio. Therefore, the requirements engineering (RE) community devised a number of requirements prioritization methods (RPM) to support project organizations in their product improvement endeavors. In RE, there are multiple proposals for defining what the term 'importance' means. Two key factors are benefit and/or cost associated with each individual requirement [4],[18],[33],[34]. Consequently, early requirements prioritization can be supported by means of methods for predicting the cost caused and benefit added by single requirements. This reasoning about requirements prioritization from cost and benefit perspective lead us to the following research question (RQ): "What - in terms of methods and activities - is needed to prioritize requirements using cost and benefit predictions as criteria?" Our motivation to raise this question rests on the fact that, despite the awareness of the usefulness of cost and benefit information in requirements prioritization, the RE community did very little to consolidate existing knowledge on this topic and to position existing RPM in respect to how each one treats cost and benefit estimation for *single* requirements.

This work presents the results of the first of a series of research activities. This series has the objective 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 in this area. We focus on RPM used in early project 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.)

In this paper, we are set out to answer our RQ by applying the explorative research method of Grounded Theory (GT). In what follows, we first present this method, its application to the RQ and our main results, which we summarize in a conceptual model. We then highlight one specific aspect: how requirements dependencies are treated by RPMs. Next, we define a framework for classifying RPM, particularly based on benefit and cost predictions. We finish up with a discussion on the threats to validity, conclusions and future research.

2. Grounded Theory

The GT which we deployed is the qualitative research method developed by Strauss and Corbin [31] 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 GT 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 (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 [32]. The philosophical foundation of GT [31] and how it affects the researcher's choices in carrying out his/her work have been discussed in [7] and are beyond the scope of this paper. Here, we focus on the application of the GT process [31] and the results we obtained.

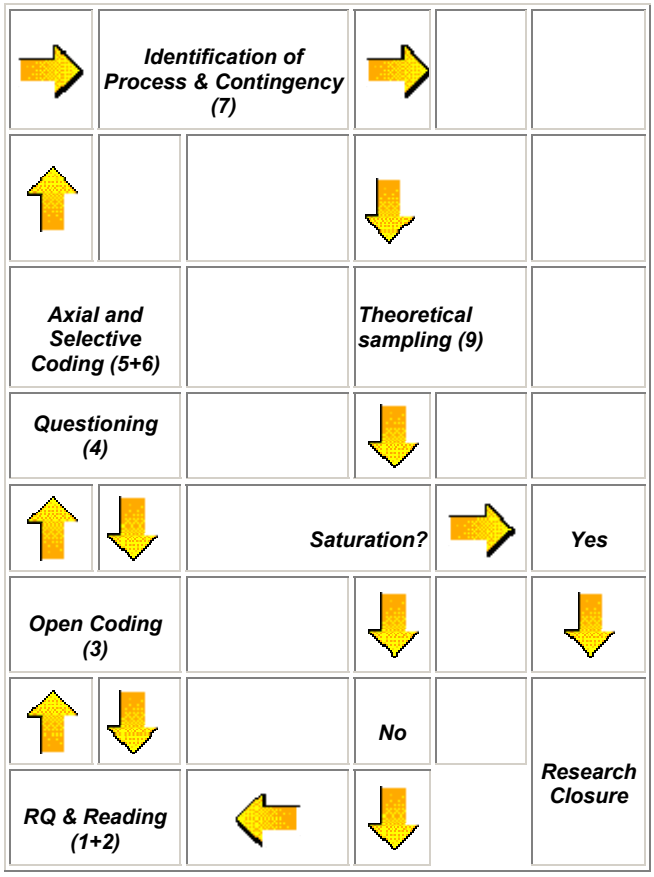

Fig. 1 The iterative framework-building process by means of GT.

The GT steps are as follows:

(1) Setting the research question, which means determining what we specifically focus on and what we want to know about it ([31], p.38);

(2) Reading technical and non-technical literature to stimulate theoretical sensitivity ([31], 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;

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

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

(8) Transactional analysis;

(9) Theoretical sampling of empirical data, using the concepts and categories found earlier.

The objective of GT is the discovery of as many relevant categories as possible, along with their properties and dimensions. Steps 1-9 can be and usually are traversed iteratively several times [23] (Fig 1), because: "Constant interplay between proposing and checking […] is what makes our theory grounded!" [31]. That means, the analysis of the data collected in one step helps to check the interpretations from the previous step. Section 4 indicates the results we obtained from our application of GT.

3. Use of Literature

To us, GT was an iterative learning process taking 9 months of continual literature research and discussion. In our study, the data used and constantly compared to the emerging theory was scientific literature. Our iterative GT process included a two-phase literature research [7]: First, we selected a set of literature sources from RE and Software Engineering literature for the purpose of developing our conceptual model and classification framework (Fig 2). During this first phase, we analyzed about 400 recent publications (including their "related work" sections), which we found in conference/workshop proceedings and journals, as well as technical reports from RE schools, to trace back to primary studies. The list of these sources was the result of our literature research done in the last three years. We read and re-read text of these papers to uncover categories and inter-relationships (steps 2-7).

Second, we performed a systematic review [19], during which we analyzed other 100 publications, which we used for the specific purpose of testing (i) the completeness of the conceptual model and (ii) whether the classification framework helps classify the

sources and if so, to what extent [7] (theoretical sampling, step 9). Thus, the framework was finished up only after the second phase of literature review. Both phases yielded a total of 240 papers which met the following five 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); this is to ensure the paper directly adds up to answering the RQ.

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

(3) relevance for practice: the method potentially offers support for practical requirements prioritization,

(4) the paper adequately describes the context, in which the method is expected to be applicable; 'adequately' means that the reader can replicate the use of the RPM in his/her own context;

(5) original paper: for each method, we searched at least its original publication; if an original paper is difficult to access, or is outside the RE field, we included another description from an RE author.

The publications we selected in both phases were written in English only and included both qualitative and quantitative research, from scientists and practitioners. The results of both phases of literature research are published in [8].

4. Results of the Grounded Theory

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

Step (1+2) Given our RQ, we reviewed the 400 literature sources which formed the first phase of literature research. The first author focused on cost estimation and the second on benefit estimation and RPM in general. This choice was motivated by resource constraints and by each author´s specific expertise. Our 'theoretical sensitivity' resulted from 14.5 years of authors' collective experience in software development.

Step (3) Open coding: This step was carried out by creating an internal report (described in step 4) and diagrams which the authors exchanged and discussed. (These diagrams were intermediate versions of the activity diagram (Fig. 2) and visualizations of the benefit function discussed in Section 5.) In this step, the core concept, which emerged, is the requirements prioritization process; it consists of *activities* which are performed on each *requirement*. A requirement is characterized by the following properties relevant with respect to the RQ: (i) *type*, (ii) *estimated benefit* to stakeholders, (iii) *estimated size* of software that embeds the requirement, (iv) *estimated cost* to build the software, (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' or 'secondary NFR'. Primary requirements directly provide benefit to the stakeholders, while secondary requirements are derived from primary requirements, support and constrain them [20]. Only few authors distinguish between these two types of requirements explicitly, while many do it implicitly. In [25], it's assumed that primary requirements are usually FR and that secondary requirements can be both FR and NFR. However, there are other authors [9], [11] who regard NFR as primary requirements as well. Hence, we 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 one case study indicates [20]. We found that RPMs usually do not define what 'priority' means. Reviewing literature, we identified the following categories 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-faceted *dependencies* among requirements and their benefit and/or, respectively, cost. One vehicle for studying requirements dependencies and for classifying how RPMs account for them is the *benefit function* (see section 5). Such a function is under-utilized in software engineering [4], [10], but commonly used in Mathematical Economics [30].

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 to refine. An operationalization is a "possible design alternative for meeting NFR [or more generally: requirements] in the target system" [2].

Step (4) Questioning: During the whole GT process, the concepts proposed in step 3 by each author and examples of methods and publications for each concept category were gathered in an internal (unpublished) report. Each author regularly reviewed and questioned the sections written by the other author.

Step (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. In this work, the 'process' or 'story' are modelled in the notation of an activity diagram (Fig. 2): We found that benefit estimation demands a different approach for primary requirements (which by definition directly contribute benefits) and for secondary requirements (which by definition enhance benefits produced by primary requirements). Cost estimation usually is based on secondary requirements, and is done in two steps, supported by two types of methods: the activity '*Estimate size (of requirement)*' means determining the size of software it would take to realize one specific requirement and the activity *'Estimate cost (of requirement)'* means determining the cost it would take to implement the piece of software of that size.

Fig. 2 shows the prioritization 'story' [31] of each individual requirement, not necessarily of all requirements together. The role and treatment of requirement dependencies are discussed in Section 5.

We found that each activity corresponds to a specific type of methods. For example, most RPMs exclusively prioritize requirements by sorting (what includes negotiation among stakeholders). However, these RPM are not designed with a specific prioritization criterion in mind and do not offer any support for benefit or cost estimation/ prediction.

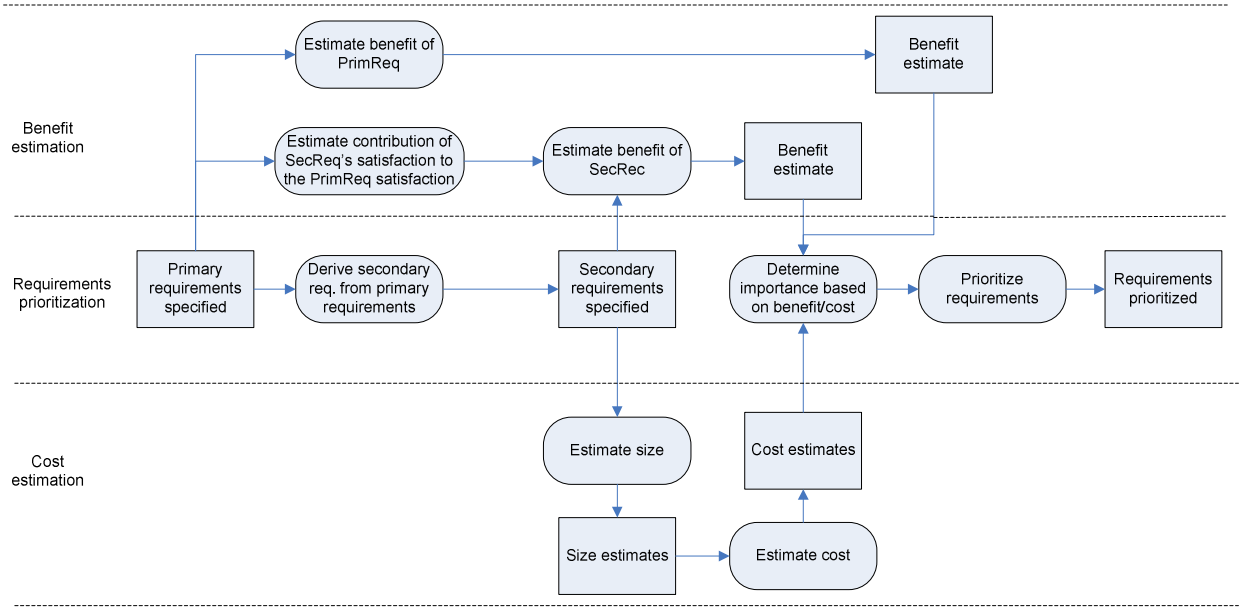

Fig. 2: Activity diagram depicting activities during requirements prioritization based on benefit and cost estimation

Step (8) Transactional analysis has not been done for this study yet.

Step (9) Theoretical sampling was done by means of a systematic review of literature. (In the future, we plan

to do it by applying the process shown in Fig. 2 in case studies.) In step 9, we checked the completeness of both the concepts and the properties of the concepts, identified in step 3. We also investigated which of them can be used for characterizing published RPMs. This question is answered by the classification framework presented in Section 6.

5. Requirements Dependencies

Requirements dependencies, although important in practice [27], are discussed by only few requirements prioritization authors [3], [20], [24], [33]. Unlike other requirement properties (as priority or estimated cost), we found the following: (i) requirement dependencies do not describe a property of one requirement, but of the relationship between at least two requirements, (ii) we could not identify an activity or method which is specifically designed for coping with dependencies. Instead, our finding is that (i) dependencies are treated by RPMs only implicitly, and (ii) the way each method does this is a characterizing property of this method. This finding led us to classify RPMs according to how they treat requirements dependencies. For developing and justifying criteria for our classification, we applied a mathematical model [30] of how to reason about requirements benefit, based on the concept of a benefit function. 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 [6], [7]. In the light of our RQ, we are interested in the benefit produced by single requirements, i.e. the benefit being gained when adding requirement A to system S. Our earlier analysis [7] showed the following consequences of requirements dependencies on benefit estimation: (i) benefit estimation for a single requirement only makes sense relative to a clearly defined 'reference system', which is an idea of an ensemble of requirements which are supposed to be realized [10]; (ii) requirements benefits are not additive, i.e. the benefit of a group of requirements is not the sum of the benefits of the individual requirements. However, these two observations and further factors lead to practical challenges in benefit estimating [1], [7]. Practical benefit estimation must be executed in a simpler way than determining a complete benefit function exactly in the N dimensional space; even if such simpler ways of estimating mean an approximation and the result of which deviated from the exact benefit value. Our review of RPMs shows that six types of simplifications are usually made, each with its specific advantages and disadvantages:

1) Each requirement´s benefit or importance is assumed to be fixed, independently of any reference system, and additive: The advantage of this simplification is that estimations need to be done only once and that the benefits of requirements can be added. This approach disregards all dependencies among requirements.

2) Grouping requirements: Requirements are grouped into bundles in a way that each group is approximately independent of the others with respect to the prioritization criterion (e.g. benefit). Such groups are used by many RPMs (without theoretical justification, though) (see overview in [7]). 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 [16], which in turn reduces the complexity of the benefit estimation task when one first gets estimations for the requirements groups and, then, for the requirements within each group (like in [21]).

3) Using relative values instead of absolute: Should benefit be compared to cost, it is ideal to monetize the benefit (e.g. in $US or work hours saved). However, often relative values (like 1/2/3 or low/medium/high) are preferred as they are easier to estimate than absolute ones [13].

4) Pair-wise comparison: Some RPMs attribute a value to each requirement, while other methods determine relative values by pair-wise comparison.

5) Using discrete values instead of a continuous scale: This means that the importance (or benefit) 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 in an order of importance or a nominal scale as the values 1-2-3, where the numbers signify names of categories.

6) Building intervals: Some authors advocate that intervals be used for the estimation, for example by doing an optimistic, realistic and pessimistic estimation [1], [3]. The merit of using intervals is that it can capture uncertainties.

The above six types of simplification are used in Section 6 to design a classification framework that should help position the existing RPMs with respect to

how each one treats requirement dependencies.

6. Classification Framework for Methods

The model in Fig 2, (resulting from the GT study) served to develop a classification framework capable of structuring the methods existing in the RE literature. This framework we used for classifying results of our literature research. Below, we introduce the classification factors that make up the framework.

Essentially, it classifies the existing methods on the basis of the *activity* which they support (Fig. 2). For example, some methods support the activity of estimating cost based on size estimation, or of deriving secondary requirements from primary requirements, or of prioritizing the requirements based on known importance values. 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 were found to 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* (FR or NFR, primary or secondary requirements) they take as input.

RPMs are characterized by the type of simplification they use to treat requirement *dependencies*. This classification refers to the six simplification described in Section 5. They correspond to six factors which indicate whether or not an RPM applies this simplification. For instance, some RPM include the simplification of pair-wise comparison, while others don´t. For each method in the literature, we analyzed whether it usually does apply a simplification or not. We also questioned whether the simplification could be used with this method. Most combinations of the six factors were found to be applicable in at least one RPM. We analyzed 15 basic RPM like Cumulative Voting [21] and Analytic Hierarchy Process [28]. All of them assume a fixed importance and priority value. Grouping explicitly is foreseen in two RPM, in eight it is not foreseen but can easily be included, while for the others this makes no sense. Almost all RPM use relative values, except for Cost-Benefit Analysis [22] which explicitly uses absolute ones and another which has been applied using both types of scales. Six out of 15 RPM apply pair-wise comparison. Ten RPM foresee discrete values, while the others use continuous scales. The latter could also use discrete values, except for Cost-Benefit Analysis [22]. Intervals of values are almost never used, except for one RPM which has been applied using intervals. Eight RPM are constructed in a way that the use of intervals is not possible, while for the others it is.

In the RE literature, RPMs are distinguished according to some other criteria, like: time consumption [13], [14], [15], [16], ease of use [14], [15], [16], [29], fault tolerance [15], notation [29]. These criteria all are relevant when choosing a RPM for a specific purpose. However, these are not relevant with respect to our RQ. Therefore, we do not use them here. Furthermore, our literature research showed that existing RPM can not be distinguished with respect to the following criteria: (i) the prioritization criterion supported (benefit, cost or others) and (ii) the type of requirements input into the RPM. We found that most RPM can use any prioritization criterion and that all RPM can apply to all types of requirements (FR or NFR, primary or secondary requirements).

7. 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 call 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 the systematic literature review on the same topic, namely the use of benefit and cost information in support of requirements prioritization. Using the framework [8], we 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, therefore, 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

GT analysis could be put in question as: (i) the first author was the only reviewer for cost-based requirements prioritization papers, and (ii) the second author was the only reviewer of benefit-based requirement 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 [12] 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.

Moreover, we also considered Glaser's criteria [6] for judging the credibility of an emerging theory that comes out of GT research efforts. Glaser (as well as other GT authors [5]) 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. We kept to the above GT principles, and so the main 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 in our immediate future to demonstrate the fit of the framework by using it in case studies.

Furthermore, modifiability of an emerging theory is concerned with the possibility to update it and extend it in the future. We 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, our framework will need some refinement/extension so that it's kept useful.

8. Summary and Future Work

The key contribution in this paper is a classification framework for methods used in the context of early requirements prioritization based on benefit and cost estimation. We derived it by applying GT. This effort was part of an on-going research aimed at increasing the understanding of how benefit and cost estimates are used in support of state-of-the-art requirements prioritization. For the immediate future, our mandate is to complete the systematic literature review [19] a summary of which is published in [8] to present how existing methods support the prioritization activities in our framework. This will serve the objective to identify a research agenda on benefit/cost-based requirements prioritization. We also plan three other steps to augment and/or refine the agenda: (i) we want to know which methods have been validated empirically and how, so that we add to our research agenda items pertinent to empirical research; (ii) our application of the GT so far exclusively treated questions concerning method support. We plan to carry out a transactional analysis for the RQ (as discussed in [6]) and expect it to lead to further issues, e.g. concerning the role of the organization/stakeholders in the prioritization; (iii) we plan case studies in companies' sites to demonstrate the adequacy, the relevancy and the modifiability of our framework.

9. References

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

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

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

[4] 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, Germany, 2006, pp. 39-66.

[5] 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.

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

[7] Herrmann, A., M. Daneva, Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework, Technical Report SWEHD-TR-2008-01, University Heidelberg, Version 1.0, 3 March 2008, http://www-swe.informatik.uni-heidelberg.de/research/publications/reports.htm.

[8] Herrmann, A., M. Daneva, "Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research", *Proc. 16th Int'l Conf. Requirements Eng., 2008*, Barcelona, Spain, 08-12th Sept 2008.

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

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

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

[12] 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.

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

[14] 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)

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

[16] Karlsson, L., P. Berander, B. Regnell, 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.*, Edinburgh, pp. 145-154.

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

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

[19] Kitchenham, B., *Procedures for Performing Systematic Reviews*, JTR, Keely University TR/SE-0401, ISSN: 1353-7776.

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

[21] Leffingwell, D., D. Widrig, *Managing Software Requirements - A Unified Approach*, Addison-Wesley, Reading, Massachusetts, USA, 2000.

[22] Nas, T.F., *Cost-Benefit Analysis: Theory and Application*, Thousand Oaks, Sage, USA, 1996.

[23] Pandit, N.R., "The Creation of Theory: a Recent Application of the Grounded Theory Method", The Qualitative Report, 2(4), Dec 1996.

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

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

[26] Ruhe, G., A. Eberlein, D. Pfahl, "Trade-Off Analysis For Requirements Selection", *Int'l J of Soft. Eng. and Knowledge Eng*., 13(4), 2003, pp. 345-366.

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

[28] Saaty, T.L., *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.

[29] Salinesi, C., 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.

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

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

[32] 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, 2001, pp. 104-140.

[33] 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, 2006

[34] Wiegers, K., "First things first: prioritizing requirements", *Software Development*, 7(9), 1999.