Requirements Prioritization Based on Benefit and Cost Prediction: An Agenda for Future Research

Andrea Herrmann[*], Maya Daneva[+]
[*]*University of Heidelberg, Faculty of Mathematics and Computer Science, Software Engineering Group, 69120 Heidelberg, Germany*[1]
[+]*University of Twente, Department of Computer Science, PO Box 217, 7500 AE Enschede, The Netherlands*
m.daneva@utwente.nl

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 presents results of a systematic review of literature, which investigates how existing methods approach the problem of requirements prioritization based on benefit and cost. From this review, it derives a set of under-researched issues which warrant future efforts and sketches an agenda for future research in this area.

Key words: Requirements Prioritization, Software Economics, Non-Functional Requirements, Quantitative Prioritization, Systematic Review

1. Introduction

Requirements prioritization based on importance has been a popular concept in software engineering for more than 30 years. A number of requirements prioritization practices have been devised to increase its adoption in organizations. The requirements engineering (RE) community knows multiple proposals for defining what the term 'importance' means. Two key factors are benefit and cost associated with each individual requirement [1], [2]. Consequently, requirements prioritization can be supported by means of methods allowing the prediction of cost caused and benefit added by single requirements. The objective of the present article is to sketch the challenges of using benefit and cost prediction in support of requirements prioritization, to survey current solutions to this problem, 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. (For later development phases, different benefit/ cost estimation methods exist.)

We set out to answer the following research question (RQ): In which way is requirements prioritization based on benefit and cost estimation currently supported by which published method? We approached it by using a systematic review (SR) of literature, drawing on our earlier work [3] which yielded a framework for classifying RPMs. In what follows, Section 2 introduces this framework. Section 3 describes the SR of literature and Section 4 presents its most important findings. Section 5 and Section 6 discuss threats to validity - and the research agenda resulting from the SR, respectively.

2. Classification Framework

We derived the factors for classifying RPMs [3], [4] found in our SR by Grounded Theory [5], by analyzing scientific literature. The core concept of requirements prioritization 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 requirements prioritization based on benefit and cost: (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*

[1] now at: Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany, Andrea.Herrmann@iese.fraunhofer.de

1090-705x/08 $25.00 © 2008 IEEE
DOI 10.1109/RE.2008.48

dependencies, what in this context means that the degree of satisfaction of one requirement influences the cost caused or the benefit added by another requirement. The property '*type*' means a pair of two orthogonal qualities: 'functional/non-functional requirement (FR/NFR)' and 'primary/secondary requirement'. (The latter property is defined in Section 4.1.) The *type* of a requirement can be one of the following pairs 'primary FR', 'secondary FR', 'primary NFR' and 'secondary NFR'.

We specifically acknowledge those activities (see the ovals in Fig. 1) which determine one of the properties of a requirement. For example, the activity 'Estimate size (of requirement)' means determining the size of software it would take to realize the requirement. This paper classifies methods existing in the RE literature according to that activity in our framework which each method supports (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) almost all methods, as we will see later on, focus on one and only one activity. For examples of methods for each activity, see Section 4.

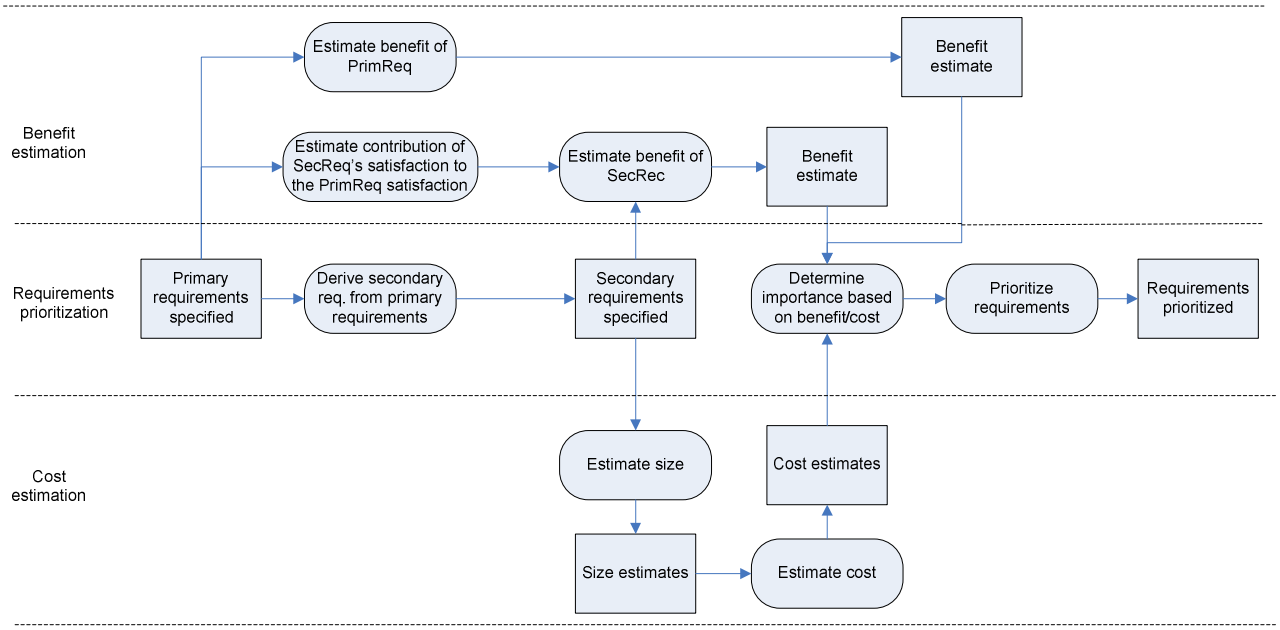

Fig. 1: Activity diagram depicting activities during requirements prioritization based on benefit and cost estimation

Fig. 1 explicitly separates the benefit and cost estimation activities from the prioritization activity which is merely concerned with determining priorities. Moreover, we will see in Section 4 that RPM can use arbitrary prioritization criteria and that the way each RPM treats dependencies is characteristic to it. Specifically, our framework distinguishes six ways of treating dependencies [3]. Thus, it lets us classify whether or not a RPM applies one of these six ways:

1) Each requirement's priority (or benefit, or cost) is assumed to be a fixed value: This approach disregards all dependencies among requirements. This is commonly done by state of the art RPMs, see Table 1.

2) Grouping requirements: Requirements are grouped into bundles in a way that each group is relatively independent of the others. This grouping takes care of the most important dependencies and disregards all others. The groups can be built on different levels to form a hierarchy of requirements [6], which in turn reduces the complexity of the estimation task when one first gets estimations for the groups relative to each other and, then, for the requirements within each group (as in [7]).

3) Using relative values instead of absolute: If benefit is to be compared to cost, it is ideal to express benefit in monetary terms (e.g. in $US) or in work hours saved. However, relative values have been found to be easier to estimate than absolute [8] and therefore often are preferred.

4) Pair-wise comparison: Some RPMs attribute one value per requirement, while others determine relative values by pair-wise comparison.

5*) Using discrete values instead of a continuous scale:* This means using a set of categories, e.g. an ordinal scale which ranks the requirements by their order of importance or a nominal scale like the values 1-2-3, or low/ medium/ high [2], or mandatory/ desirable/ inessential [8],[9].

6) Building benefit intervals: Some authors advocate that intervals be used, e.g. by estimating an optimistic, realistic and pessimistic value [10], [11].

We note that in the RE literature, RPMs are also distinguished according to other criteria, e.g.: ease of use, fault tolerance, or notation, and that such criteria are relevant when choosing a RPM for a specific purpose. However, such criteria are irrelevant with respect to our RQ. Furthermore, our literature research showed that the following criteria don´t seem to be factors suitable for grouping existing RPM: (i) the prioritization criterion supported (benefit, cost or others) and (ii) the type of requirements input into the RPM. This is discussed in [3] and [4].

3. Systematic Review

The overall objective of our SR effort is twofold: (i) to categorize the state-of-the-art methods supporting requirement prioritization based on benefit and cost information, and (ii) to identify under-researched areas that warrant future effort.

As per SR guidelines [12], we used the RQ (defined in the Introduction) for determining the content and structure of the SR, for designing strategies for locating and selecting primary studies, for critically evaluating the studies, and for analyzing their results. We, however, note that we didn´t strictly follow all SR guidelines from [12], [19], [69]. Specifically, 93% of the papers were classified by only one of the authors (as discussed in Section 5). And the results of our literature review (partly presented in this paper), also include the results of our previous literature research, like books and doctoral theses.

We used the following search strings: (1) non-functional AND requirements, (2) prioritization AND method, (3) prioritization AND approach, (4) cost AND size, (5) effort AND size, (6) requirements AND dependency, (7) value-based AND requirements, (8) value-oriented AND requirements, (9) requirements AND conflict, (10) requirements AND negotiation. These search strings are the result of a learning process, that is, we experimented with a variety of combinations of these words in order to test synonyms used in literature and to cover the concepts and properties of our classification framework. We had to proceed like this because no standardized, consistent terminology is used with respect to requirements prioritization, cost and benefit estimation. For example, work on value-based software engineering/ RE usually pertains to our RQ as well as work on requirements conflict resolution and requirements negotiation, which usually builds on requirements dependencies. We also reviewed work on requirements dependencies, because we found that RPM usually don´t treat them explicitly.

We used the Boolean "OR" operator to concatenate all these search terms and searched: 1 OR 2 OR 3 OR 4 OR 5 OR 6 OR 7 OR 8 OR 9 OR 10. Our search strategy included (i) journal publications in bibliographic databases (namely, ACM Digital Library, Compendex, IEEE Xplore, ISI Web of Science, Kluwer Online ScienceDirect – Elsevier, SpringerLink, Wiley InterScience) and (ii) a manual search in the following volumes of five conference/symposium proceedings: Requirements Engineering (RE) (2000, 2004-2007 [13]), Software Engineering (ICSE) (1999-2007 [14]), Empirical Software Engineering and Measurement (ESEM) (2007 [15]), Empirical Software Engineering (ISESE) (2003-2006 [16]) - all by IEEE Computer Society Press -, Mensura [17], published by Springer LNCS, and Requirements Engineering – Foundation of Software Quality (REFSQ) (1997-2007) [18]. Our choice of the above databases rests on that they are available to use due to our universities´ subscriptions to them. We performed the searches between Jan 15 and Feb 5, 2008, applying the search query individually to each source. Those databases, which didn´t allow queries composed of complex Boolean expressions, were queried by running separate

searches. Then, we used the union of the results obtained. We borrowed this practice from Mendes [19] who found it to work well in her SR. We applied the search query to the titles, abstracts, and keywords of the articles in the identified databases and conference proceedings. We excluded editorials, prefaces, summaries of articles and tutorials, workshops, panels and poster sessions. Our search strategy yielded a total of 240 papers which met the following quality criteria for inclusion in the review:

(1) the paper is on a method (e.g., an 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 supports requirements prioritization in practice,

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

(5) 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 expect that the fact that a work is cited and used by others is 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. Results of the Systematic Review

To synthesize the SR findings as well as results from former literature research, we mapped them against the classification framework (Section 2). This helped us clearly see which activities in requirements prioritization based on cost and benefit estimation are covered and which ones are supported little or not enough. We make the note that because of space limitation, not all methods described in the 240 papers from our SR are presented in this paper. We here focus on those examples we deem 'typical', meaning that they best illustrate how the activity can be executed.[2].

4.1 From primary to secondary requirements

Our SR showed that a distinction between primary and secondary requirements is seldom explicit, but it is often implicitly made, using different terminology. Poort and de With [20] define that (i) primary requirements are those to be demanded by the stakeholders who benefit from them and (ii) secondary requirements are those which are derived from and constrain the primary requirements. Primary requirements can be decomposed, operationalized, refined to, or supported and constrained by secondary requirements. In MOQARE (Misuse-Oriented Requirements Engineering) [21], [22], FR, business goals and quality goals play the role of primary requirements, and these are analyzed for deriving secondary requirements (the so-called countermeasures). The Defect Detection and Prevention (DDP) method [23], [24], makes a difference between (primary) requirements which the system is to satisfy and (secondary) PACT (an acronym for "Preventions, Analyses, process Controls, and Tests"). The IESE NFR Method [25], [26] uses quality attributes as primary requirements and derives "means". In goal-oriented RE methods [27], [28] we can interpret goals to be primary requirements and what these methods usually call "requirements" to be the secondary ones. This is also evident in [29], [30] and [31] whose authors recommend decomposing NFR into more refined NFR and additional FR as well as architectural requirements. We note that primary requirements sometimes form a hierarchy, like goals and sub-goals, or business goals and IT system goals.

Both primary and secondary requirements can be FR or NFR. The review by [22] found that the following types of secondary requirements could be derived from primary NFR: (i) new FR, extensions and constraints on FR, (ii) architectural requirements, (iii) constraints on project and software development, or (iv) constraints on administration or maintenance. Primary FR can also be decomposed into secondary requirements, as in the TORE (Task-Oriented Requirements Engineering) approach [32], or when goal-oriented RE methods analyze functional goals.

[2] The full list of papers and referenced methods can be found online at: http://is.cs.utwente.nl/

However, published case studies don´t indicate when the decomposition of primary FR should lead to secondary FR only and when to secondary NFR. We note that the difference between primary and secondary requirements depends on the viewpoint of the benefit estimator: All those requirements which s/he can estimate the benefit for, are primary. For instance, a software end user usually can estimate the benefit of FR or of quality goals, but not of architectural requirements.

4.2 Benefit estimates: primary requirements

We found no methods which estimate benefit for individual requirements. Instead, the methods described in the reviewed literature treat whole IT systems. Their predictions are either based on experience or on the use of methods as value chain analysis, process analysis, business cases, or sensitivity analysis, which estimate the benefit of whole systems, while we need benefit estimations for individual requirements. Often, it is left to those stakeholders, who experience the benefit, to estimate it.

4.3 Benefit estimates: secondary requirements

Estimating benefit for secondary requirements usually includes two steps: the estimation of how much a secondary requirement´s satisfaction contributes to the primary requirements' satisfaction, and, then, the calculation of the secondary requirement´s benefit. We found two groups of methods quantifying the contribution of secondary requirements to the satisfaction of primary NFR: The first group of methods is *risk-reduction-oriented* as it defines secondary requirements in order to detect, mitigate or prevent risk posed to the satisfaction of primary requirements. For these secondary requirements, benefit can be quantified as the risk reduction which they affect, what is a common practice in security RE [33],[34],[35],[36]. Risk is defined as the product of 'probability of having something going wrong' and 'anticipated damage' [35],[36]. Next, the second group of methods *directly* quantifies the contribution of secondary requirements to the satisfaction of primary NFR. These methods often are specific to one or few quality attributes, e.g. reliability prediction models derive failure rate from the software defect rate, or the defect rate from quality assurance measures [37],[38], [39],[40].

With respect to FRs, our SR could not identify any method that rests on the derivation of benefits of secondary requirements from primary FR benefit.

4.4 Size estimation for requirements

Requirements-based cost estimation usually includes two steps: sizing the FR and NFR requirements and, then, estimating how much it would cost to implement the requirements of this size. We note explicitly that we consider only those methods which take requirements as their input (that is, they are applicable in early phases) and which relate size and cost to individual requirements (and not whole systems).

Our SR indicates that FR are sized by using functional size measurement (FSM) models that rest on Function Point Analysis; examples of commonly used FSM methods are [41],[42],[43]. We also found that FSM models are now being extended to cover 'size estimation' of NFR, provided NFRs are defined in operationalized form, that is in testable and verifiable terms [44], [45]. For example, the authors of [44] extend the COSMIC-FFP [41] FSM method to account for NFRs and, then, integrate it into the NFR framework [28]. In this proposal [44], the COSMIC-FFP counting rules take as their inputs the NFRs that are specified as goals in operational form. The counting output data (Full Function Points) are then provided to stakeholders who use them to make prioritization decisions.

Our SR suggests that in current FSM methods two perspectives could be taken in quantifying NFRs. As per [42],[43], to obtain size and effort numbers for the NFRs in a project, the NFRs must be first decomposed into a series of corresponding FRs. Then, a FSM method is considered to be the suitable vehicle for quantifying the contribution of NFRs to software size, and, ultimately, to the effort it would take to build the software project. These authors assume that it makes sense to decompose all NFRs into FRs. Recently, however, this assumption was questioned by researchers in goal-oriented RE [46],[47],[48], who put forward that not all NFRs should or can be decomposed into FRs. Specifically, these authors recommend NFRs not be decomposed into FRs if: (i) the NFRs are 'normative', that is, if they say how the actor is to interact with the system [47]; or (ii) the

NFRs serve as criteria for making architectural design choices; that is, the function of these NFRs is to help evaluate alternatives.

Table 1: Classification of 15 RPMs according to the criteria in Section 2. These 15 methods are basic RPMs, which most frequently discussed by prioritization authors and serve as building modules in more complex RPMs. 'X' stands for 'yes', the symbol '|' means that both alternatives have been found in literature, the symbol '-' means 'no (impossible or makes little sense)' and the symbol '○' means 'no, but can be included'.

	Fixed importance	Requirements grouping (instead of treating individual requirements)	Using relative values instead of absolute	Determining relative values by pair-wise comparison	Using discrete values instead of a continuous scale	Building benefit intervals instead of using one value only
Numeral assignment [8]	x	○	|	-	○	○
Cost benefit analysis [56]	x	○	○	-	-	-
Cumulative voting/ $100 test [57]	x	-	x	-	|	-
Priority groups [64], also called grouping/numeral	x	○	x	-	x	-
Top 10 Requirements [59]	x	-	x	-	x	-
Multi-Attribute Utility Theory [60], [61]	x	○	x	-	○	○
Weighting Method [61],[62]	x	○	x	-	○	○
Planning Game [9]	x	○	x	-	x	|
Analytic Hierarchy Process (AHP) [1],[63],[64]	x	○	x	x	x	○
hierarchy AHP [64],[65]	x	x	x	x	x	○
Outranking [61],[66]	x	○	x	x	○	-
Minimal spanning tree matrix [64]	x	-	x	x	x	-
Bubblesort [64],[67]	x	-	x	x	x	-
Binary search tree [64], [67]	x	-	x	x	x	-
Hierarchical Cumulative Voting [68]	x	x	x	○	x	○

4.5 Cost estimation from size

FSMs often serve as dependent variables when developing cost (or effort) estimation models [49], might it be a model based on expert judgments, algorithmic estimation, or estimation by analogy [50], [51]. Our SR found, however, that very few cost estimation approaches have ever been used for the purpose of requirements prioritization. Very little is done to guide project managers and RE staff on which cost estimation practices work best, which are must-do, which can be safely dropped or merged in which contexts. In practice, most project managers still predict effort by using what is called a "delta method" [52], which suggests new projects be costed via their delta to previous projects: the effort associated with a new requirement is the amount of hours which went to the last project's requirements, multiplied by some factors modeling the recent and the new project contexts. This method was found simple, quick, and best of all, can take full advantage of local costing information. However, evidence for its use exists with respect to FR only and there is no indication of adopting this method to estimate the cost of implementing NFR. For NFR, there exists no clarity in the literature which approach to use in which context. So, cost analysts better stick to recommendations by software economics researchers to deploy multiple effort estimation techniques and compare results [50].

4.6 Requirement importance from benefit/ cost

As said earlier, benefit and cost both can be criteria for requirement prioritization or can be used in combination, e.g. when requirements are prioritized according to their net value (benefit minus cost)

[9],[35], benefit-cost-ratio [53],[54] or return on investment [55].

4.7 Requirements prioritization based on requirement importance

Our analysis of RPMs indicates that all of them assume that (i) stakeholders, at least tacitly, know the importance of requirements, or (ii) estimation methods are available for them to use. Our SR identified two methods only which explicitly define their prioritization criteria. Both use benefit and cost. These are: the cost benefit analysis [56] and the Planning Game [9]. All other methods, can use any importance/ priority criterion. We present our classification of 15 methods in Table 1, summarizing how they treat requirement dependencies according to the six factors from Section 2. We analyzed whether a method does apply an approximation or not, as described in literature. We also evaluated whether an approximation could possibly be used with it.

5. Threats to Validity

We considered the possible threats [12] to validity and took measures to counterpart them. First, the 'relevance' of the papers included in the SR and their classification could be questioned as in the SR (i) the first author was the only reviewer of papers for sections 4.2, 4.3, 4.6 and 4.7 and (ii) the second author was the only reviewer for cost and benefit estimation papers (see sections 4.1, 4.4, 4.5). We are aware of the SR guidelines [69] suggesting an individual classification be done by several researchers who then discuss the differences in each classification proposal by tracking rates of inter-researchers' agreement. However, this approach demands much time because all authors must read all papers. Because of resource constraints, the only viable option to us at the time was to (i) classify the papers individually and then (ii) have both authors read and discuss those papers only, for which classification turned out difficult. The latter account for 7% of the total sample. The classification of the other 93% of the papers was checked by means of an internal (unpublished) report. Each author regularly reviewed and questioned the part written by the other author. As we are concerned about the internal validity of the classification, in immediate future we plan to engage a third researcher in the role of checker, that is, s/he will review a subset of our papers and compare his/her classification with ours. This will increase the validity of the results and also avoid bias. We also note that the framework was developed in an earlier effort by the same authors and that the research process of framework formulation might have biased us in classifying the papers. Moreover, we are aware that our access to 'relevant' sources depended on the appropriateness of the search strings used. The fact that there is no standardized terminology used in RE posed a challenge. We treated their composition as a learning process. The list of search terms was adapted six times and search re-run with the new terms. For some search strings, we applied synonyms like "value-based" and "value-oriented". We also tentatively AND-combined the ten search strings pair-wise and queried the databases. The resulting list of papers had a reduced the number of items, which were less than 10% of the items resulting from using one search string alone. In half of the cases with pair-wise combined strings, the resulting paper list was empty or contained only one or two papers. This is a hint that our search strings are only slightly redundant. Moreover, we also accounted for concerns experienced by other SR authors [19], for example, that some key words are absent in the abstract or title, although the paper itself treats the topic searched for. Though, as we also searched for "prioritization" AND "method", we localized such papers. We achieved it at the expense of much screening work because the majority of papers in the resulting list (around 95%) were irrelevant to our RQ. However, the remaining 5% often didn´t use any of our other search strings. So, without these very general search strings they would not have been identified. We used the abstracts only for the decision whether the paper is relevant. The method classification itself was based on the complete paper.

6. Research Agenda Resulting from the SR

Drawing on the foregoing results, we identified an agenda for future requirements prioritization research. It includes six themes concerned with those issues which have been solved only partly or not at all:

1.) Concluding from section 4.1, it is under-researched when decomposing primary FR leads to

secondary NFR or when it makes sense to identify secondary FR only. Explicitly investigating these questions would advance our understanding of the relation between FR and NFR.

2.) Section 4.2 indicated that in RE no methods could be found to estimate direct benefit for primary requirements. There is still space for research on whether and how well mature benefit estimation methods from other disciplines (e.g. business) can be applied to requirement benefit estimation.

3.) How the quantified contribution of secondary requirements to primary requirements can be used for quantifying benefit of secondary requirements remains an unanswered question (Section 4.3). No method was found, which explicitly puts these values together.

4.) Applying quantitative approaches to sizing NFRs is another potentially fruitful avenue. The use of a standard FSM model (e.g. COSMIC FFP) seems promising [44]. To advance in this direction, more research efforts are needed to confront further issues such as when to decompose NFRs to FR, and how to link the estimated size of operationalized NFRs to effort (see Section 4.4).

5.) Although methods exist for estimating cost of FR, based on their size, such relationships are under-researched for NFRs (see Section 4.5).

6.) Requirements dependencies are largely neglected in requirements prioritization, despite the fact that the RE community agrees on their importance. In section 2, we identified six ways of how RPMs usually treat them. We assume that there is a benefit function which exactly describes the benefit. When applying an approximation, there will result a deviation between the approximated and the exact benefit value. We are uncertain how much these two values deviate from each other. In practice, one often accepts to live with this uncertainty because the approximation saves time (when applying the method) and makes it easier to use it. It would be interesting to empirically compare two versions of RPMs, by applying it once with an approximation, and once without.

7.) Theoretically, 48 combinations of the six approximations are possible[3], but the cited 15 methods only apply 21 of these (Table 1). 24 are not implemented because all methods assume fixed priority values[4]. The three combinations which remain yet-not-tried-out are: (i) all six simplifications are applied, except for discrete values; (ii) fixed, relative and continuous values, without grouping or pair-wise comparison, either estimating intervals or (iii) single values. However, it is unclear whether this observation is coincidental, or whether these combinations make no sense for the practice of requirements prioritization.

7. Summary and Future Work

This paper investigated in which way existing methods support requirements prioritization based on early benefit and cost estimation. This was done by an SR. Answering this question served the objective to identify a research agenda on this topic. Two immediate 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 more items pertinent to empirical research; and (ii) we so far exclusively treated questions concerning method support. Research on further factors still remains to be done and will lead to further issues, e.g. concerning the role of the organization and of stakeholders in the prioritization.

8. References

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

[2] Wiegers, K. First things first: prioritizing requirements, Software Development, 7(9) Sept 1999.

[3] 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, 9 Feb 2008, http://www-swe.informatik.uni-heidelberg.de/research/publications/reports.htm.

[4] Daneva, M., A. Herrmann, Requirements Prioritization Based on Benefit and Cost Prediction: A Method Classification Framework, Proc. 34th Euromicro Conf., Track on SPPI 2008, to be published

[3] not 64, because pair-wise comparison only makes sense when relative values are estimated

[4] Except for Hierarchical Cumulative Voting; however, the other methods don´t add priority/ importance values, and therefore no major error is made.

[5] Strauss, A.L., J.M. Corbin, Basics of Qualitative Research - Grounded Theory Procedures and Techniques, 6th print, Sage, Newbury Park, USA, 1991
[6] Karlsson, J., S. Olsson, K. Ryan, Improved Practical Support for Large-scale Requirements Prioritisation, REJ 2(1) 1997, pp.51-60.
[7] Ruhe, G., A. Eberlein, D. Pfahl, Trade-Off Analysis for Requirements Selection, Int'l J of SEKE 13(4) 2003, pp. 345-366.
[8] Karlsson, J. Software requirements prioritization, Proc. 2nd Int'l Conf. RE, 1996, pp.110-116.
[9] Beck, K., Extreme programming explained, Addison-Wesley, Upper Saddle River, USA, 2000.
[10] A.M. Davis, The Art of Requirements Triage, IEEE Computer 36(3) March 2003, pp. 42-49.
[11] B.W. Boehm and R. E. Fairley, Software Estimation Perspectives, IEEE Software, Nov/Dec 2000, pp.22-26.
[12] Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., El Emam, K., and Rosenberg, J., Preliminary Guidelines for Empirical Research in Software Engineering, IEEE Trans. on Software Eng., 28(8) 2002, pp. 721–734.
[13] www.re08.org, www.re07.org, etc.
[14] www.icse-conferences.org
[15]http://www.esem-conferences.org/
[16] http://www.cos.ufrj.br/%7Eght/isese2006.htm
[17] http://mensura2007.uib.es
[18] www.refsq.org
[19] Mendes E., A Systematic Review of Web Engineering Research, Proc. of Int'l Symp. on Empirical Software Eng. 2005, IEEE CS Press, pp. 498-507
[20] E.R. Poort, P.H.N. With, Resolving Requirement Conflicts through Non-Functional Decomposition, Proc. 4th Workshop on Software Architecture, 2004, 145-154.
[21] Herrmann A., B. Paech, Quality Misuse, Proc. Workshop on Requirements Engineering Foundations of Software Quality (REFSQ 2005), Essener Informatik Beiträge, Essen, Germany, 2005, pp. 193-199.
[22] Herrmann A., B. Paech, MOQARE: Misuse-oriented Quality Requirements Engineering, REJ 13(1) 2008, pp. 73-86.
[23] Feather, M.S.,S.L. Cornford, uantitative risk-based requirements reasoning, REJ 8(4), 2003, pp. 248-265.
[24] Feather, M.S., S.L. Cornford, J.D. Kiper, T. Menzies, Experiences using Visualization Techniques to Present Requirements, Risks to Them, and Options for Risk Mitigation", Int'l Workshop on Requirements Eng. Visualization, Minneapolis, 2006.
[25] Kerkow, D., J. Doerr, B. Paech, T. Olsson, T. Koenig, Elicitation and Documentation of Non-functional Requirements for Sociotechnical Systems, in: Requirements Engineering for Sociotechnical Systems, S. Maté, ed., Idea Group, 2004.
[26] Doerr, J., D. Kerkow, T. Koenig, T. Olsson, T. Suzuki, Non-Functional Requirements in Industry - Three Case Studies Adopting an Experience-based NFR Method", Proc. 13th Int´l Conf. on RE, 2005, pp. 373-384.
[27] A. van Lamsweerde, Goal-Oriented Requirements Engineering: A Guided Tour, Proc. 5th Int'l Symp. on RE, 2001, pp. 249-263.
[28] Chung, L., B.A. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Acadamic Publishers, 2000.
[29] Paech, B., A. Dutoit, D. Kerkow, A. von Knethen, Functional requirements, non-functional requirements and architecture specification cannot be separated, Proc. 8th Int'l Workshop on REFSQ,2002), Essener Informatik Beiträge Band 7, Essen, Germany, pp. 102-107.
[30] C. Ebert, Understanding the Product Life Cycle: Four Key Requirements Engineering Techniques, IEEE Software, May/June 2006, pp. 19-25.
[31] Gilb T., Competitive Engineering: A Handbook For Systems Engineering, Requirements Engineering, and Software Engineering Using Planguage, Butterworth-Heinemann 2005.
[32] Paech, B., K. Kohler, Task-driven Requirements in Object-oriented Development, Perspectives on Requirements Engineering, Kluwer Acad. Publ., 2003.
[33] Butler, S. Security Attribute Evaluation Method: A Cost-Benefit Approach, Proc. 24th Int. Conf. Soft. Eng.(ICSE 02), ACM Press, 2002, pp. 232–240.
[34] 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.
[35] Xie, N., N. R. Mead, P. Chen, M. Dean, L. Lopez, D. Ojoko-Adams, H. Osman, SQUARE Project: Cost/Benefit Analysis Framework for Information Security Improvement Projects in Small Companies, Techn. Note CMU/SEI-2004-TN-045, SEI, CMU, 2004.
[36] International Standards Organization ISO, Risk management – Vocabulary – Guidelines for use in standards, ISO Guide 73, International Standards Organization, Geneva, 2002.
[37] Laprie, J.C., K. Kanoun, C. Beounes, M. Kaaniche, The KAT (Knowledge-Action-Transformation) Approach to the Modeling and Evaluation of Reliability and Availability Growth, IEEE Trans. on Software Eng. 17(4) 1991, pp. 370-382.
[38] Brocklehurst, S., S. Littlewood, New Ways to Get Accurate Reliability Measures, IEEE Software 9(4) 1992, pp. 34-42.
[39] Wallace D., C. Coleman, Application and Improvement of Software Reliability Models, Proc. NASA OSMA Software Assurance Symposium 2001.
[40] Boehm, B., L. Huang, A. Jain, R. Madachy, The ROI of Software Dependability: The iDAVE model, IEEE Software 21(3) 2004, pp. 54- 61.
[41] Abran, A., J.-M. Desharnais, S. Oligny, D. St-Pierre, and C. Symons, COSMIC FFP – Measurement Manual,

COSMIC implementation guide to ISO/IEC 19761:2003, École de technologie supérieure – Université du Québec, Montréal, Canada, 2003
[42] FISMA, Experience Situation Analysis, Finnish Software Metrics Association, 2001, http://www.fisma.fi/wp-content/uploads/2006/09/fisma_ situation _ analysis_method_nd21.pdf
[43] ISBSG, Practical Software Estimation, 2nd Edition, Int´l Software Benchmarking Standard Group, 2006.
[44] Kassab, M., O. Ormandjieva, M. Daneva, A. Abran, "Non-Functional Requirements: Size Measurement and Testing with COSMIC-FFP", Proc. Int'l Conf. on Software Process and Product Measurement (MENSURA), UIB Press, pp. 247-259.
[45] Clements, P., The Value of Software Architecture, Proc. of the IFIP Int'l Conf on Software Architecture (WICSA2008), SEI, CMU, US.
[46] M. Glinz, Rethinking the Notion of Non-Functional Requirements, Proc. 3rd World Congress for Software Quality, Munich, Germany, 2005.
[47] Wieringa, R., The Declarative Problem Frame: Designing Systems that Create and Use Norms, Proc. 10th IEEE Int'l Workshop on Software Specification and Design, IEEE Computer Society Press, 2002, pp. 75-85.
[48] Mylopoulos, J., Goal-oriented Requirements Engineering, Keynote speech at the 14th Int'l Conf. on Requirements Eng., IEEE Computer Society Press, 2006.
[49] Finnie, G., G. Wittig, J.-M. Desharnais, A Comparison of Software Effort Estimation Techniques Using Function Points with Neural Networks, Case-based Reasoning and Regression Models, J of Syst & Soft 39, 1997, pp. 281-289.
[50] Shepperd, M., Software Project Economics: a Roadmap, ICSE 2007: Future of SE, pp.304-315.1
[51] Boehm, B. C. Abts, S. Chulani, Software Development Cost Estimation Approaches - a Survey", Annals of Software Eng. 10, 2000, pp. 177-205.
[52] B. Boehm, B, Safe and Simple Software Cost Analysis, IEEE Software, Sept/Oct, 2000, pp. 14-17.
[53] Kazman, R., J. Asundi, M. Klein, Quantifying the Cost and Benefits of Architectural Decisions, ICSE, 2001, pp. 297-306.
[54] In, H., R. Kazman, D. Olson, From Requirements Negotiation to Software Architectural Decisions, Proc. From Software Requirements to Architectures Workshop (STRAW), 2001.
[55] Nejmeh B., I. Thomas, Business-Driven Product Planning Using Feature Vectors and Increments, IEEE Software 19(6) 2002, pp. 34-42.
[56] Nas, T.F., Cost-Benefit Analysis: Theory and Application, Thousand Oaks, Sage, USA, 1996.
[57] Leffingwell, D., D. Widrig, Managing Software Requirements - A Unified Approach, Addison-Wesley, Reading, Massachusetts, USA, 2000.
[58] Berander P. A. Andrews, Requirements Prioritization, Engineering and Managing Software Requirements, A. Aurum and C. Wohlin, eds., Springer, Berlin, Heidelberg, Germany, 2005, pp. 69-94.
[59] Lauesen, S., Software Requirements - Styles and Techniques, Addison-Wesley, NY, 2002.
[60] Keeney, R.L. and H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Trade-Offs, Cambridge University Press, 1993.
[61] Salinesi C., E. Kornyshova, Choosing a Prioritization Method - Case of IS Security Improvement, Forum Proc. of 18th CAiSE´06, Luxemburg, 2006, pp.51-55.
[62] Keeney, R.L. Foundations for Making Smart Decisions", IIE Solutions 31(5) 1999, pp. 24-30.
[63] Saaty, T.L., The Analytic Hierarchy Process, McGraw-Hill, New York, 1980.
[64] Karlsson, J., C. Wohlin, and B. Regnell, "An evaluation of methods for prioritizing software requirements", Inf &Soft Techn 39, 1998, pp. 939-947.
[65] Davis, A., Software Requirements: Objects, Functions and States, Prentice-Hall, NJ, 1993.
[66] Roy, B., Multicriteria Methodology for Decision Aiding, Kluwer Acad. Publ., Dordrecht, 1996.
[67] Aho, A.V., J.E. Hopcroft, and J.D. Ullmann, Data Structures and Algorithms, Addison-Wesley, Reading, MA, USA, 1983.
[68] Berander P., P. Jönsson, Hierarchical Cumulative Voting (HCV) - Prioritization of Requirements in Hierarchies, Int'l J of SEKE. 16(6) 2006, pp. 819-849.
[69] Jørgensen,M., M.J. Shepperd, A Systematic Review of Software Development Cost Estimation Studies, IEEE Trans. Software Eng. 33(1) 2007, 33-53.