Perspectives on Requirements Understandability — For Whom Does the Teacher's Bell Toll?

Mikael Svahnberg*, Tony Gorschek
School of Engineering
Blekinge Institute of Technology
PO Box 520, Soft Center, SWEDEN
Phone: +46 457 385811
[Mikael.Svahnberg | Tony.Gorschek]@bth.se

Magnus Eriksson†
Land Systems Hägglunds AB
SE-891-82 Örnsköldsvik, Sweden
Magnus.Eriksson@baesystems.se

Andreas Borg, Kristian Sandahl
Department of Computer and Information Science
Linköping University
[andbo | krs]@ida.liu.se

Jürgen Börster, Annabella Loconsole
Department of Computing Science
Umeå University
[jubo | bella]@cs.umu.se

Abstract

Software development decision makers use many different information sources as a basis for their decisions. One of these sources is the requirements specification, which is used in a large number of processes throughout the software development cycle. In order to make good decisions, the quality and completeness of the available information is important. Hence, requirements must be written in a way that is understandable for the different decision makers. However, requirements are rarely written with an explicit perception of how to make them understandable for different target usages. In this study we investigate the implicit assumptions of current and future requirements engineers and their teachers regarding which usages they perceive as most important when creating requirements. This is contrasted with industrial viewpoints of the relative importance of different requirements usages. The results indicate that the teachers and future requirements engineers have a strong focus towards in-project perspectives, and very little in common with the perspectives of industry managers. Thus, we are training students to serve as software developers, and not software engineering managers.

1 Introduction

The quality and usability of a software requirements specification depends on the understandability of the software requirements. The requirements are intended for different target audiences and different target usages. These different target uses require different information in a requirement for it to be usable (compare e.g. the information needed for product planning, project planning, and software design), and the requirement may need to be specified in a different way to be understandable in the different contexts it is going to be used. For example, the requirement "Support for Multiple Languages" may be understandable and useful for high level product planning, whereas a developer requires additional information regarding how many languages to support, whether the languages should be dynamically changeable, whether any languages with special characters should be supported, what to do when the display is too short for a specific language, etc.

Requirements engineering education and requirements engineering text books (e.g. [10, 11, 12, 18]) often describe different qualities that requirements must possess, e.g. complete, consistent, feasible, necessary, prioritised, unambiguous, verifiable, etc. One characteristic that is often also mentioned is that requirements should be understandable. It is, however, rarely discussed further what would constitute an understandable requirement, let alone for whom it would be understandable. Since the same books also discuss different users of the requirements (e.g. project management, project planning, end users, developers, customers, testers, documentation groups, etc.), one may implicitly assume

*Contact Author.

†The remainder of the author list is presented in alphabetical order of affiliation.

that the requirements should be made understandable for all these different users.

It is our experience that although requirements engineers may be aware of different users of requirements, the requirements are rarely created with a specific target audience in mind, let alone a collection of target audiences. Rather, the requirements are created and made understandable based on a general opinion of the requirements engineer that includes a conscious or subconscious assumption of the target audience. In other words, the requirements engineer has made a personal prioritisation of the importance of different requirements users, and creates the requirements to be understandable for the different users according to this prioritisation.

The main research question is thus:

- What are the usages that requirements engineers have in mind when creating requirements?

The implications of this from a decision support perspective stems from the fact that the requirements are used as decision support in a number of decision processes. If the requirements are not specified in an understandable way for a specific target audience it follows naturally that decisions are made based on a less than optimal material, and it may not even be possible to detect this by just studying the requirements. Hence, we need to understand the implicit assumptions that the requirements engineers have regarding the possible target audiences when specifying a requirement. This will enable decision makers at various positions to better assess the quality and completeness of the decision support material (i.e. the requirements specifications) for their specific usages. In addition, this gives the decision makers an insight into the perceptions of requirements engineers, which may be valuable during process improvement work or training efforts.

With hierarchical requirements specifications (see e.g. [8, 11]), it becomes even more important that the requirements are specified in an understandable way for the intended target groups. In these cases, different abstraction levels are targeted towards different users (i.e. decision makers) of the requirements, and it should ideally not be necessary to read requirements from other abstraction levels. A decision maker should ideally be able to use only one or two abstraction levels as decision support material, rather than having to sift through the entire mass of information. In order to accomplish this, mere implicit assumptions of the requirements engineers are no longer sufficient to be able to write understandable and complete requirements for the actual target audiences. By understanding the usages of requirements (the target users that e.g. industrial practitioners identify), it is possible to more accurately tailor different abstraction levels, and the requirements themselves, to the needs of the requirements users.

Table 1. Investigated Perspectives

Industry	*Academia*
Management	Teachers
Developers	Students

In this article we investigate the perspectives of four groups of people regarding what is important for them to make a requirement understandable. These groups are listed in Table 1. We investigate the views of two groups of industry practitioners (i.e., managers and developers), in order to get a view of what is considered important in industry. We investigate a group consisting of a large portion of the students educated in requirements engineering in Sweden during 2004, since these persons are the creators of requirements in industry during the coming years. Finally, we investigate a group consisting of a large portion of the academia involved in requirements engineering research and education in Sweden during 2004 and 2005, in order to understand the general opinions of the teachers of tomorrow's requirements engineers. This allows us to study the following questions:

- Is there a shared view of what makes a requirement understandable between industry managers and developers?
- Do the students' perspectives match those of the teachers?
- Do the academic perspectives match the industry perspectives?

With these questions, we investigate whether there is one single perspective in industry, and also whether the perspectives of future requirements engineers match the industry's need or whether further training is necessary. In addition, we get indications of whether academia need to adjust their perspective to match the needs of industry, and whether the requirements engineering education is effective in conveying that there are different requirements users that have different importance. If there is a mismatch between what is required by industry and what the future requirements engineers are being taught, the education needs to be adjusted, and further training in industry of the requirements engineers may be required. It may also be the case that academia, or the students, have a more up-to-date perception than industry, and in these cases it may be possible to more accurately direct training efforts toward industry.

The remainder of this paper is organised as follows. In Section 2 we describe our study design. In Section 3 we present and analyse the results of the study. Section 4 discusses these results. Related work is presented in Section 5, and the paper is concluded in Section 6

2 Study Design

Our study is primarily conducted using a survey approach. With the exception of the Industry Management-group, participants from each of the groups are asked to assign values to different perspectives on understandability, using the Cumulative Voting ($100-test) [12] prioritisation method. For the remaining group (Industry Management), we extract information from a publication by Wohlin and Aurum [19] into a comparable form, as detailed in Section 3.1.

2.1 Participants

The Industry Management group consists of people involved in the decision-making process that determines whether a requirement should be included in a specific release or not. Example of roles in the group are: product manager, project manager, line manager, user experience engineer and method specialists.

The Industry Developers group is composed of people actively working with requirements engineering in a number of companies that are collaborating with Blekinge Institute of Technology. Examples of roles in this group are: test engineer, developer, and product manager.

The Academia Teachers group consists of members of the SiREN network (Swedish Requirements Engineering Research Network[1]. This group consists of a mix of professors, assistant professors, and PhD students from six Swedish universities, and comprises a large portion of the active requirements engineering research and education in Sweden. In addition, a group of PhD students and lecturers in Software Engineering at Blekinge Institute of Technology participated. These are all active in teaching students in courses on topics such as project management, quality management, verification and validation, software architectures, and software design.

The Academia Students group consists of graduate and undergraduate software engineering and computer science students at Blekinge Institute of Technology, Linköping University and Umeå University. Together, the software engineering and computer science students from these three universities constitute a large portion of the students educated in requirements engineering in Sweden during 2004.

The selection of subjects from different companies and universities is made to ensure that the study results are not unduly influenced by local variations at a single company or university. Instead, the goal is to get an as comprehensive view of the current status in Sweden as possible.

[1]For more information, please see http://www.siren.lth.se/

2.2 Questionnaire Design

The questionnaire used for the study contains a number of perspectives on requirements understandability and the subjects are asked to rate the importance of each perspective by distributing 1000 tokens among the perspectives. Table 2 presents the questionnaire and the perspectives that the participants are expected to rate.

3 Results and Analysis

In this section we present and analyse the results of the study. We present and discuss each participant group, and compare them to the other participant groups.

3.1 Industry Management

Wohlin and Aurum [19] present a survey of criteria that are important when deciding which requirements to include in a specific project. This serves as a good example of a decision-making process based on the requirements that is primarily management-oriented. Wohlin and Aurum [19] present 13 criteria that 33 industry practitioners were asked to assess using the same method (i.e. the $100-test) that we use in this study. Because their survey has a different focus than the study in this article the results and perspectives cannot be compared directly (the selections of criteria and perspectives are, for example, depending on the foci of the studies), but it is possible to translate the results into something that is usable in the context of this article. A translation between the criteria used by Wohlin and Aurum and the perspectives used in this study is presented in Table 3.

In this table we see the criteria used, its importance (in %), and the perspective (from this study) that is closest to the criterion. This translation is based on the brief description of each criterion provided in the original article [19]. As can be seen, three perspectives are not used, i.e. Verification & Validation, Reuse, and Product Planning II, and one criterion is not mapped to any perspective, i.e. Support for Education / Training. The reason for this can be found in the differing foci of the two studies. The Industry Management column in Table 4 sums up the importance of the different criteria for each perspective. The perspectives that are not used are left blank, and the 1.7% stemming from the missing "Support for Education/Training" perspective are not included. All results in this table are normalised to sum up to 100, regardless of how many points were available for the participants in the respective sub-study.

3.2 Industry Developer

In total, 18 industry practitioners participated. A summary of their answers is presented under the Industry De-

Table 2. Perspectives Questionnaire

Perspectives (What do you think is important from a requirements understandability perspective?)

- Development Requirements understandability from a *Development perspective*, i.e. requirements being good-enough to base design on.
- Requirements understandability from a *Project planning* perspective, i.e. being good enough to base resource and time estimations on.
- Requirements understandability from a *Risk management* perspective, i.e. being good-enough to base risk analysis, impact analysis etc. on.
- Requirements understandability from a *Verification and validation* perspective, i.e. being able to base tests on requirements.
- Requirements understandability from a *Reuse* perspective, i.e. being able to reuse requirements and other parts associated with a requirement, e.g. description, risk analysis, estimations, etc.
- Requirements understandability from a *Product planning (I)* perspective, i.e. being able to be good-enough to give product planners a basis for strategic product decisions (e.g. what requirements to include in a release, what requirements to dismiss, etc.).
- Requirements understandability from a *Product planning (II)* perspective related to the company at large, i.e. being able to be used as a basis for establishing company strategy.

Table 3. Translation of Perspectives from Wohlin and Aurum

Criterion (Wohlin & Aurum)	*Value*	*Perspective (Our Study)*
1. Competitors	7.4	Product Planning I
2. Requirements Issuer	13.8	Product Planning I
3. Stakeholder Priority of Requirement	16	Product Planning I
4. Requirement Volatilitty	3.5	Risk Management
5. Support for Education / Training	1.7	
6. Development Cost / Benefit	11.1	Product Planning I
7. Resources / Competencies	7.7	Project Planning
8. Delivery Date / Calendar Time	14.8	Project Planning
9. System Impact	4.7	Development
10. Complexity	6.6	Development
11. Requirements Dependencies	5.8	Project Planning
12. Evolution	3.5	Risk Management
13. Maintenance	3.5	Development

velopers column in Table 4, with the exception of one participant that put 900 tokens on Verification & Validation and distributed the remaining 100 tokens on the other perspectives.

3.3 Academic Perspectives

In total, 15 teachers (who are also active researchers) in the area of requirements engineering participated. A summary of their answers is presented under the Academia Teachers column in Table 4. A total of 179 students completed the questionnaire as a part of another experiment in requirements engineering courses at the respective university. A summary of their answers is provided under the Academia Students column in Table 4.

3.4 Analysis

Figure 1 presents another view of the results in Table 4. In this figure the results for each participant group is presented for each perspective. For the Industry Managers, we see that two perspectives stand out as being more important than the others, i.e. Product Planning I (i.e. that the requirements are good enough to give product planners a basis for strategic product decisions, such as which requirements to include in a release, which requirements to dismiss, etc.), and Project Planning (i.e. that the requirements are good enough to base resource and time estimates on). For the Industry Developers, we see that the Development and Verification & Validation perspectives are considered a most important.

The two academic groups do not have any pronounced differences between each other. However, when comparing to the industry perspectives, we see some interesting differences.

The first four perspectives (i.e., Development, Project Planning, Risk Management, and Verification & Validation) comprise 71% of the industry developers' points. Interestingly, these perspectives comprise 64.6 % of the teachers' points and 65.9% of the students' points. This is a clear contrast to the industry managers' distribution of points, where the first four perspectives only comprise 50% of the total points. This means that the students think as developers, not as managers. It is also alarming that this trend is visible throughout the remainder of the data, i.e. the teachers also think as developers, and do not consider the manage-

Table 4. Results Summary

Perspective	*Industry Management*	*Industry Developers*	*Academia Teachers*	*Academia Students*
Development	14.7	23.5	19.2	21.6
Project Planning	28.3	16.4	13.3	17.1
Risk Management	7.00	9.76	13.8	12.1
Verification & Validation		21.1	18.3	15.1
Reuse		5.18	8.75	9.70
Product Planning I	48.3	19.7	16.7	15.3
Product Planning II		4.23	10.0	9.05

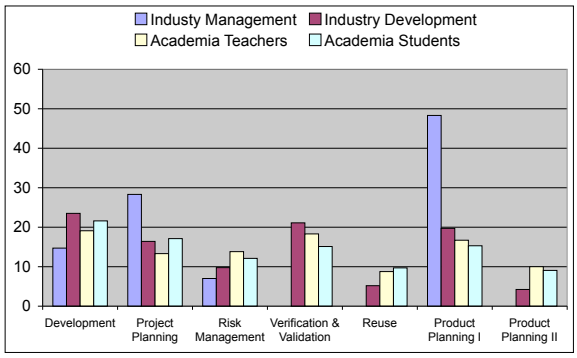

Figure 1. Results Summary

ment perspectives. Thus, we can suspect that the students are trained (not only in requirements engineering courses) to think and act as developers.

Some other interesting differences that we may study is for example that the students rate the Verification and Validation perspective as considerably (29.4%) less important than the industry developers. The reason for this may be that (a) the students have not yet been offered any courses in verification & validation, and/or (b) that the industry developers have more experience about the importance of this perspective.

Moreover, the academic perspectives rate Reuse and Product Planning II as almost twice as important as the Industry Developers do (These data points are not available for the Industry Managers). For the Reuse perspective, it may be the case that reuse is much advocated in academia and not so much practiced in industry, where hard deadlines and new features are much more pressing. A similar explanation could be given to the differences in the Product Planning II perspective: the industry developers can not afford the luxury to plan further than their current work log, and rely on that the managers do this type of strategic thinking for them, whereas the two academic groups have a more aloof view and have at least an initial awareness of that it is important to have a long-term perspective as well.

4 Discussion

Given the focus of the original study [19], the extreme importance that the Industry Management group put on the product planning perspective is perhaps not surprising. Nor is it, in itself, surprising that Industry Developers have a different view, i.e. that the Development and Verification & Validation perspective are considered more important than the Product Planning perspective. What makes it interesting is that there is such a pronounced gap between the two groups. More practically, the Industry Developers expect requirements to be understandable in their specific context, i.e. Development and Verification & Validation, whereas the Industry Managers (in the context of deciding which requirements to include in a release) focus on Product and Project Planning.

Hence, there is not a shared view between managers and developers of what makes requirements understandable. Moreover, the data indicate that the perceived importance of different perspectives is at least role-dependent, but maybe also usage-dependent (one role may be involved in different usages of the same requirement). The results indicate that requirements are created with specific purposes in mind, especially Development, Product Planning I (Strategic Product Decisions), and Verification & Validation. It is thus for these purposes that requirements are primarily made understandable. We see that perspectives such as Risk Management, Reuse, and Product Planning for Company Strategies (Product Planning II) are considered to a lesser degree when creating requirements. Decision makers seeking to use requirements for these purposes should thus be made aware that the decision support material may not be completely adequate.

From an education perspective, the results are more alarming. That teachers and students share a similar view may indicate that the education is effective in conveying the

academic opinion[2]. However, the data also indicate that the education only focuses on a single industry perspective, namely the development perspective. Obviously, this gives an unbalanced view of what is important for requirements understandability. In fact, the conclusion is that researchers/teachers as well as newly educated requirements engineering students have a single-minded view of how requirements are used and for whom requirements are written, i.e. requirements are written for the purpose of in-project activities such as development and testing. Pre-project activities (especially important in market-driven requirements engineering) receive no special focus.

An alternative explanation is that a large portion of requirements are product-specific and on a relatively detailed level (e.g., [8] reports that 57% of the requirements are on the abstraction level of specific functions or even implementation details). These detailed requirements are almost per definition unsuitable for decisions regarding company strategies. Moreover, there is too much information available to be able to quickly sift through to the requirements that are relevant for making strategic decisions. This means that for *some* requirements, the product planning perspective is very important, but for the large majority of (detailed) requirements it is not. However, since the question was asked generically (i.e. "what do you think is important from a requirements understandability perspective?"), the answers can also be expected to be on a generic level, where each requirement should be understandable from the prioritised perspectives, even though the particular perspectives may not be in focus for every requirement.

One way to clarify the intended usages of requirements is to use hierarchical requirements specifications, where some of the more abstract levels are targeted towards e.g. Product Planning for Company Strategies, and the more detailed levels are used for Development and Verification & Validation. In such a case it is important to emphasise to the requirements engineer that, depending on the level of abstraction, different information is included in requirements to make them understandable for their purposes (e.g. Product Planning for higher level requirements and Development for lower level requirements). Thus, the abstraction level of a requirement provides valuable information to the requirements engineer regarding its usage, and hence for whom it must be made understandable. If the requirements engineer understands and uses this, the value of the entire collection of requirements is increased from a decision support perspective. First, it is possible to base decisions on a subset of the requirements collection (e.g. a subset consisting of only one or two abstraction levels). Second, the requirements on a certain abstraction level are specified with certain usages and target audiences in mind. Third, because of the different needs of different usages and target audiences, more of the information that is required for a certain type of decisions may need to be present in a given subset of requirements.

[2]The astute reader will observe that we have not shown that the education in itself is effective, only that the students share their teachers opinions. Whether this was also the case *before* the students were subjected to the education is not evaluated. Strictly speaking, and from a teacher's perspective, we can only state that the students do not leave their education with unreasonable opinions (concerning requirements understandability).

4.1 Validity

Some threats to the validity of the study exists. First, the questionnaire is very brief and does not provide much information to the participants. There is thus a risk that the participants may have misunderstood what is meant by the different perspectives. On the other hand, the perspectives are fairly commonplace in most software development processes, which means that most participants have a good idea of what is meant by them anyway. Second, the participating companies may not be representative of all types of companies, and other companies may have other priorities between the perspectives. The questionnaire is also distributed to individuals within the company and may thus not represent the company's opinions of what is important. In addition, we have made an interpretation of the perspectives used by Wohlin and Aurum [19], and translated these into the perspectives used in this article. Although we are reasonably confident that the translation is accurate, there is a risk that we may have misunderstood something in the translation process.

5 Related Work

Some work has been done on connecting requirements engineering and decision making (e.g. [1, 2, 3, 7, 16, 17]). In these articles the connection made is that requirements engineering is a decision rich activity, and can benefit from more structured decision processes. Examples of decisions made include defining the scope of a product, quality assessment of the requirements themselves, requirements prioritisation, requirements negotiation, and release planning. These are all requirements engineering activities that in fact consist of a large number of decisions. Central for these articles is that the studied decisions are made as a part of the requirements engineering activities. In this article we instead focus on the produced requirements as artifacts that are used in other decision processes "outside" of the requirements engineering process, for example project planning, verification & validation, and product management.

Several sources describe different aspects of requirements management where it is important to take conflicting objectives into consideration (e.g. [4, 5, 6, 9, 14, 13, 15, 17]). These studies, and others like them, motivate our work since there is a need to understand the different viewpoints and objectives both when writing requirements, and

to understand what the authors of a requirement had in mind when reading requirements.

6 Conclusions

In a survey of decision problems in requirements engineering, Ngo-The and Ruhe [3, chapter 12] discuss the need for more research that empirically investigates and describes the current state of practice. We argue that this does not only concern how to model and view requirements engineering as a decision making or decision support activity, but also to study how the outcomes of requirements engineering activities are, and can be, used as decision support in subsequent development activities. This article makes a contribution in this respect, providing software engineering decision makers with further insights into the (often implicit) assumptions that requirements engineers make concerning the target audiences of their software requirements, and the importance of these target audiences.

In this article, we study the assumptions that requirements engineers have concerning for what purposes requirements will be used by investigating the importance placed on making the requirements understandable for different usages. We investigate and compare the opinions of industry managers, industry developers, academic requirements engineering researchers/teachers, and requirements engineering students. The results indicate that:

- Requirements engineers do indeed have certain usages in mind when creating requirements. This is visible in the importance they place on making the requirements understandable for the different usages. The results indicate that the Development perspective is generally ranked highly, as is the Verification & Validation and the Product Planning I perspective. Other perspectives, such as Reuse, Product Planning for Company Strategies, and Risk Management are not considered to be very important. In our study we also confirm findings from another survey [19], i.e. that Product Planning for Strategic Product Decisions is an important perspective when deciding which requirements to include in a particular release.
- There is a large difference between the priorities of Industry Managers and Industry Developers. Whereas the former consider it most important that the requirements are made understandable for product and project planning purposes, the latter consider development and verification & validation purposes as being most important. In fact, this is a dichotomy between pre-project activities and in-project activities, where the managers are more concerned with the pre-project activities and the developers focus more on the in-project activities.
- Academia seem to be focusing primarily on the in-project usages of requirements, i.e. Development and Verification & Validation. There is a lack of training of future requirements engineers who understand the pre-project activities and the importance of these. In other words, academia is training industry developers. Considering that a large portion (3/4) of the studied students are at a master's level, we argue that academia should focus more on educating software engineering managers.

6.1 Future Work

This study is performed on a limited selection of industry partners, and is limited to requirements engineering industry practitioners, researchers, teachers, and students in Sweden. Replication of the study with more industry partners and other academic and student participants (including non-Swedish participants) would increase the validity of the results.

Within each of the perspectives in this study, there are several uses of the requirements. Future work includes defining in more detail how requirements are used in different decision processes, and how requirements (and the requirements' specification) should be formulated to best support the decision processes. Wohlin & Aurum [19] is an example of such a study, where the criteria that are important for deciding in which project to include a requirement are investigated.

A more challenging future work is to use the knowledge gained to change the way we teach our students, so that we may create requirements engineers that are able to serve as software engineering managers — not only developers.

References

[1] A. Aurum and C. Wohlin. Applying decision-making models in requirements engineering. In *Proceedings of Requirements Engineering for Software Quality*, Essen, Germany, 2002.

[2] A. Aurum and C. Wohlin. The fundamental nature of requirements engineering activites as a decision-making process. *Information and Software Technology*, 45(14):945–954, 2003.

[3] A. Aurum and C. Wohlin, editors. *Engineering and Managing Software Requirements*. Springer Verlag, Berlin Germany, 2005.

[4] B. Curtis, H. Krasner, and N. Iscoe. A field study of the software design process for large systems. *Communications of the ACM*, 31(11):1268–1287, 1998.

[5] A. Dutoit and B. Paech. *Handbook of Software Engineering and Knowledge Engineering*, chapter Rationale Management in Software Engineering, pages 787–815. World Scientific Publishing Company, 2002.

[6] K. El-Emam and N. Madhavji. A field study of requirements engineering practices in information systems development. In *Proceedings of the 2nd IEEE International Symposium on Requirements Engineering*, Los Alamitos CA, 1995. IEEE Computer Society.

[7] R. Evans, S. Park, and H. Alberts. Decisions not requirements: Decision-centered engineering of computer-based systems". In *Proceedings of the International Conference on Engineering and Computer-Based Systems*, pages 435–442, 1997.

[8] T. Gorschek and C. Wohlin. Requirements abstraction model. *Requirements Engineering Journal*, 11:79–101, 2006.

[9] E. Kamsties. *Engineering and Managing Software Requirements*, chapter Understanding Ambiguity in Requirements Engineering, pages 245–266. Springer Verlag, 2005.

[10] G. Kotonya and I. Sommerville. *Requirements Engineering — Processes and Techniques*. John Wiley & Sons, Chichester UK, 1998.

[11] S. Lauesen. *Software Requirements — Styles and Techniques*. Pearson Education, Harlow UK, 2002.

[12] D. Leffingwell and D. Widrig. *Managing Software Requirements — a Unified Approach*. Addison-Wesley, Boston MA, 2000.

[13] L. Lehtola, M. Kauppinen, and S. Kujala. Requirements prioritization challenges in practice. In *Proceedings of Product Focused Software Process Improvement (PROFES 2004)*, pages 497–508, Berlin Germany, 2004. Springer Verlag.

[14] M. Lubars, C. Potts, and C. Richter. A review of the state of the practice in requirements modeling. In *Proceedings of the IEEE International Symposium on Requirements Engineering*, pages 2–14, Los Alamitos CA, 1992. IEEE Computer Society.

[15] U. Nikula, J. Sajaniemi, and H. Kälviäinen. A state-of-practice survey on requirements engineering in small-and-medium-sized enterprises. Technical report, Lappeenranta University of Technology, 2000.

[16] B. Regnell, B. Paech, A. Aurum, C. Wohlin, A. Dutoit, and J. N. och Dag. Requirements means decisions! — research issues for understanding and supporting decision-making in requirements engineering". In *Proceedings of the First Swedish Conference on Software Engineering Research and Practice*, pages 49–52, Ronneby, Sweden, 2001.

[17] G. Ruhe. Software engineering decision support — a new paradigm for learning software organizations. *Advances in Learning Software Organizations*, pages 104–113, 2003.

[18] K. Wiegers. *Software Requirements*. Microsoft Press, Redmond WA, 1999.

[19] C. Wohlin and A. Aurum. What is important when deciding to include a software requirement into a project or release. In *Proceedings of the International Symposium on Empirical Software Engineering*, Noosa Heads, Australia, 2005.