Available online at www.sciencedirect.com
ScienceDirect
Information and Software Technology 49 (2007) 790–805
INFORMATION AND SOFTWARE TECHNOLOGY
www.elsevier.com/locate/infsof

A controlled empirical evaluation of a requirements abstraction model

Tony Gorschek [a,*], Mikael Svahnberg [a], Andreas Borg [b], AnnaBella Loconsole [c], Jürgen Börstler [c], Kristian Sandahl [b], Magnus Eriksson [d]

[a] Department of Systems and Software Engineering, School of Engineering, Blekinge Institute of Technology, P. O. Box 520, S-372 25 Ronneby, Sweden
[b] Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden
[c] Department of Computing Science, Umeå University, S-901 87 Umeå, Sweden
[d] Land Systems Hägglunds AB, S-891 82 Örnsköldsvik, Sweden

Received 22 September 2005; received in revised form 7 September 2006; accepted 22 September 2006
Available online 7 November 2006

Abstract

Requirement engineers in industry are faced with the complexity of handling large amounts of requirements as development moves from traditional bespoke projects towards market-driven development. There is a need for usable and useful models that recognize this reality and support the engineers in a continuous effort of choosing which requirements to accept and which to dismiss off hand using the goals and product strategies put forward by management. This paper presents an evaluation of such a model that is built based on needs identified in industry. The evaluation's primary goal is to test the model's usability and usefulness in a lab environment prior to large scale industry piloting, and is a part of a large technology transfer effort. The evaluation uses 179 subjects from three different Swedish Universities, which is a large portion of the university students educated in requirements engineering in Sweden during 2004 and 2005. The results provide a strong indication that the model is indeed both useful and usable and ready for industry trials.

Keywords: Requirements engineering; Requirements abstraction; Empirical evaluation; Requirements abstraction model; Pre-pilot validation

1. Introduction

Requirements Engineering (RE) more and more transcends project boundaries as market-driven product development is becoming increasingly commonplace in software industry [2,21,22]. Central activities in RE are performed pre-project as a part of for example the product management activities since the requirements flow is continuous and not limited to a specific development instance [10,25].

In this environment requirements come from several sources both internal (e.g., developers, marketing, sales, support personnel, bug reports etc.) and external (e.g., users, customers and competitors, often gathered via surveys, interviews, focus groups, competitor analysis etc.) [12,15,16]. Large volumes of requirements from multiple sources risk overloading companies unless they can handle incoming requirements in a structured way, dismissing some and refining others prior to allocating them to a development instance [23]. In addition to the volume, the requirements themselves are of varying quality, state of refinement, and level of abstraction. In traditional bespoke development (customer–developer) [22] a requirements engineer can actively elicit requirements and thus hope to control or at least substantially influence these aspects. In a market-driven situation this is seldom the case. Most requirements are already stated in one way or another when they reach the requirements engineer (e.g., a product manager). The knowledge and experience of the developing

* Corresponding author. Tel.: +46 457 385817; fax: +46 457 27125.
E-mail addresses: tony.gorschek@bth.se (T. Gorschek), mikael.svahnberg@bth.se (M. Svahnberg), andbo@ida.liu.se (A. Borg), bella@cs.umu.se (A. Loconsole), jubo@cs.umu.se (J. Börstler), krs@ida.liu.se (K. Sandahl), magnus.eriksson@baesystems.se (M. Eriksson).

0950-5849/$ - see front matter
doi:10.1016/j.infsof.2006.09.003

organization, of which the requirements engineer in this case is instrumental, is central to making sense of the requirements as they are processed [5].

Many requirements engineering best-practices, frameworks, and tools are adapted to suit a bespoke environment with traditional, project focused, customer–developer relationships. There is a need for the development and evaluation of RE practices and models that support professionals working with product planning and development (e.g., product managers) in a market-driven environment.

This paper presents an evaluation of some key aspects of such a model, the Requirements Abstraction Model (RAM). RAM was developed as a response to needs identified in industry during process improvement conducted in cooperation with Danaher Motion Särö AB and ABB Automation Technology Products [6,7]. The Requirements Abstraction Model is designed towards a product perspective, supporting a continuous requirement engineering effort. It can handle large quantities of requirements of varying degrees of detail and offers a structure and process for the work-up of these requirements (see Section 2.1). The main purpose of the evaluation in this paper is to assess the usefulness and usability of RAM in a controlled environment prior to widespread industry piloting. This is achieved by evaluating the following two aspects of the model. First, by using the model, engineers should be able to get an overview of an entire product by studying relatively few top level requirements. Second, engineers should be able to utilize the concepts of requirements abstraction as they decide whether new incoming requirements are in-line with product goals and strategies. This supports the need to handle large amounts of requirements continuously as they come in to an organization from various sources like market analysis, end-customers and sub-contractors by offering fast triage against the organizations strategies. The triage is performed early in the product management process as a prelude to the refinement and eventual prioritization and selection of requirements for implementation in projects, making it possible to dismiss requirements early prior to investing effort in to detailed analysis. This also alleviates the risk of requirements overload, a major risk in market-driven requirements engineering [4,10,23]. These aspects are further detailed as problem statements in Section 2.2.

Prior to describing the actual study and results a short introduction to RAM is given in order to increase the understanding of the rationale behind the evaluation. Therefore Section 2 gives an overview of RAM, the problem statement used in this evaluation, and related work. Section 3 details the planning, context, and overall design of the evaluation along with a sub-section detailing validity issues. The research questions are formally stated in Section 4, and the operation (execution) of the evaluation is detailed in Section 5. Section 6 presents the results and analysis with the help of descriptive statistics. Section 7 is devoted to revisiting the research questions and discussing them explicitly based on the results and the analysis. In Section 8 conclusions and plans for further work are presented.

2. Background and related work

This section gives a short description of RAM and related work regarding the concepts of abstraction in relation to RE work. For reasons of brevity details not central for the evaluation presented in this paper have been left out. For details, please see Gorschek and Wohlin [8].

2.1. The requirements abstraction model

RAM is a hierarchical requirements abstraction model and a method for working with this model based on the concept that requirements come on several levels of abstraction. Instead of flattening all requirements to one abstraction level RAM *uses* the varying abstractions of requirements, and orders the requirements hierarchically according to abstraction level. Fig. 1 shows four abstraction levels; *Product Level*, *Feature Level*, *Function Level*, and *Component Level*. The Product Level is the most abstract level and requirements here are considered abstract enough to be comparable to product strategies, and indirectly to organizational strategies. In the context of RAM, product strategies are rules, long and short-term goals, road-maps, visions pertaining to a product specified by management etc. Going down to Feature and Function Level the requirements become concrete enough to be used for estimations and as input to development.

Briefly, the process followed with RAM is that when requirements arrive they are placed and specified on an appropriate level by comparing the new requirement with the existing requirements base (i.e., example-driven placement). This means that the current mass of requirements is used as decision support material for the inclusion, exclusion, and placement of new requirements. Following this all requirements go through *work-up*. Work-up entails abstracting low-level requirements up to Product Level and breaking down high-level requirements to Function Level. This is done by creating new requirements in the levels above and below and linking them to the original requirement. Fig. 1 gives an example of this. The original requirement "*C:Support for multiple languages*" (placed on Feature Level) is abstracted to the Product Level through the creation of a new work-up requirement "*Usability internationally*", and broken down to the Function Level where three new work-up requirements are created as a part of the breakdown. In some cases requirements already present can be used for abstraction. As an example, in Fig. 1, a new requirement stating "*C:Support imperial units*" could be placed on the Feature Level and linked directly to "*Usability internationally*" as imperial units are used in Great Britain and the US in addition to SI units (metric system). In this case no new requirement has to be created on the Product Level and the new requirement can be linked to an already existing one.

During the work-up process the original requirement is compared to product strategies. This offers decision support regarding whether the requirement should be specified, refined, and kept in the repository, or whether the requirement should be dismissed. For example, let us assume that "*Usability internationally*" was not accepted since company wanted to limit the product market to the Scandinavian market. In this case "*Usability internationally*" would be "*Usability in Scandinavia*", and the new requirement "*C:Support imperial units*" would be dismissed as only SI units are used in Scandinavia. In other words, the product level ultimately decides whether to include a requirement or not, since all requirements are directly linked to this level. If the new requirement is not supported, this may indicate that a new product level requirement needs to be created.

The break-down of the requirements stops on Function Level where the requirements are not perfect, but good enough to be used as decision support for estimation and risk analysis and as input to project(s) for realization. The Component Level is not mandatory in the model, and only exists since requirements in some instances are delivered in a very detailed form, and these requirements also need to be handled (i.e., specified and abstracted to assure that they are in line with the overall goals and strategies). In the example presented in Fig. 1, the Component Level requirement acts as extra information on a detailed technical level. For example, the interface has to be adapted to pictograms for the Chinese language. In this case, the Component Level sets a restriction that will accompany the requirements into development.

The features of using RAM can be summarized as follows:

(I) All requirements are compared to the product strategies, offering an assurance that requirements do not violate the overall goals set by management. This offers the possibility to dismiss requirements early in the process, freeing resources to work on and refine relevant requirements that are in line with the product strategies.

(II) All requirements are broken down to an abstraction level where they are good-enough for initiating a development effort (project(s)). This assures that projects get good-enough requirements to base their development efforts on (e.g., testable and unambiguous [13]).

(III) Work-up of a requirement means that additional requirements may have to be created in order to get a connection to the uppermost level. For example, if an incoming new requirement is placed on the Function Level and no appropriate requirement exists on the Feature Level, a new one has to be created. This Feature Level requirement in turn needs to be linked to the Product Level. This ensures that it is possible to follow a requirement through abstraction levels and assure that there is an explicit connection upwards to product strategies. In the same way every requirement is broken down to a level good-enough to serve as a basis for project initiation (i.e., Function Level). Requirements within a certain abstraction level are homogenous enough to be comparable with each other, which is a prerequisite for effective release planning and prioritization.

(IV) All requirements can be followed through several levels of abstraction giving a richer understanding of each requirement, and thus better decision support can be obtained for all professionals, from management to developers. Managers can for example study the most abstract levels and get a quick overview of the system, while developers can choose a more detailed view but still have an explicit connection to the overall goals of the product, since detailed requirements are connected upwards through the levels.

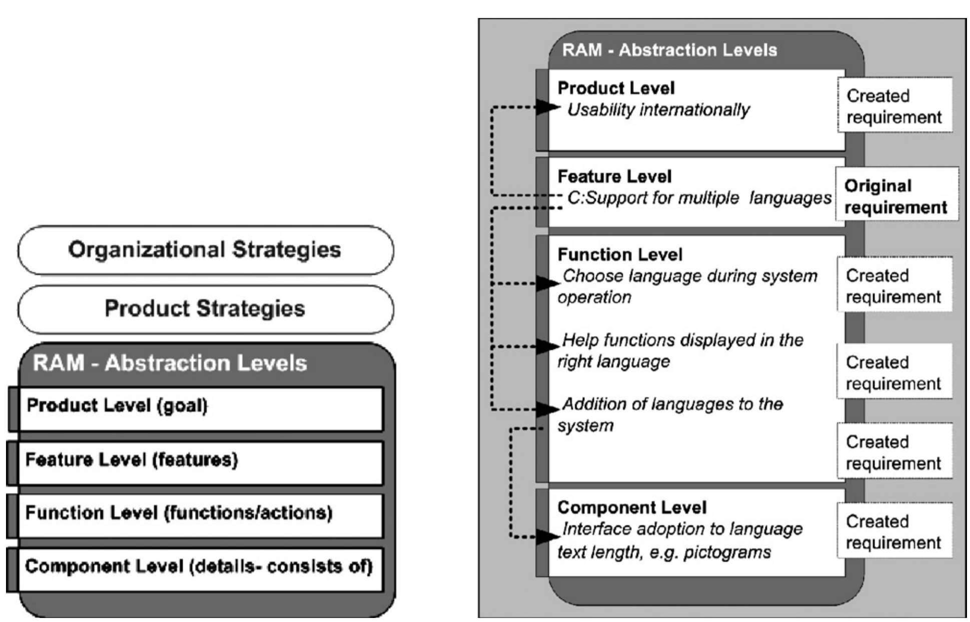

Fig. 1. RAM abstraction levels and example of work-up.

As an initial step in preparing RAM for industry use some of the underlying concepts of the model need to be evaluated in a laboratory setting. This is elaborated in Section 2.2.

2.2. Problem statement

The usefulness and the usability of RAM depend on several concepts that need to be assessed. The ones studied in this controlled evaluation are summarized below in two main parts.

(Part I) Top level (Product Level) requirements should be used to ascertain whether a new incoming requirement is in line with product strategies or not, acting as decision support for acceptance or early dismissal of a new requirement. Product Level requirements should therefore convey an overview ("big-picture") of the product (i.e., Product Level requirements should summarize and represent a considerable amount of requirements on lower levels).In Part I, we evaluate whether abstract Product Level requirements are general enough to summarize several requirements on lower levels, but still concrete enough so that it is possible to determine if they represent the product or not. If this is the case, Product Level requirements can be used to convey the aforementioned big-picture product overview.

(Part II) In Part II, we evaluate whether it is possible in practice, with relatively few abstract top-level requirements, to understand what features and attributes a product should and should not consist of. As new requirements come in it should be possible to:
– dismiss or accept a particular requirement using product strategies (explicitly expressed as Product Level requirements in this evaluation), and
– place the requirement on an appropriate level of abstraction using already present requirements on each level as indication of where the new requirement should be placed.

The usability of the model depends on these two points and thus the utilization of requirements abstraction. Requirements abstraction means that it should be possible to use relatively few abstract Product Level requirements to derive decisions across the entire requirements hierarchy with respect to inclusion or exclusion of new requirements. In addition, the ability to place new requirements on an appropriate level using already present requirements as decision support keeps the levels homogenous, i.e., requirements can be easily compared within the same abstraction level. The ability to place requirements in the model also preserves model consistency over time.

The controlled evaluation of RAM presented in this paper is centered on the aspects described in Part I and Part II, and they are stated formally as research questions in Section 4. In this study, we use software engineering and computer science students from three universities (i.e., Blekinge Institute of Technology, Linköping University, and Umeå University). This totals 179 participants from four different education programs, which is a large portion of the university students educated in requirements engineering in Sweden during 2004 and 2005.

2.3. Related work

The realization that requirements are often specified on different levels of abstraction and in varying stages of refinement has been recognized by others in both industry and academia [9,23]. We have confirmed this during process assessment and improvement efforts conducted at both Danaher Motion Särö AB and ABB Automation Technology Products [5].

Requirements on different levels of abstraction and at varying levels of refinement are considered by some as crucial input to the development in order to get a better understanding of what should be developed and why [23]. The basic notion is that both the abstract (long-term overview) and the detailed (giving context and the short term-view) are important [17,18], and the two used in combinations offer a better understanding.

The field of goal based requirements engineering (see [1,3,11]) is focused on the elicitation of goals intended to become requirements at some point, i.e., working from the top down. For example, Wiegers [24] and Lauesen [14] describe that requirements can be of different types pertaining to what they should be used for. This corresponds to the industry view of dividing requirements of different types into separate documents, from goal-like (abstract natural language formulations [19]) to technical design-like specifications. The focus is on that requirements are specified on different abstraction levels depending on usage, e.g., project type.

RAM is designed to operate in a market-driven product centered development environment. The volume of requirements that needs to be handled makes initial screening important in order to decrease the risk of work overload in the evaluation and realization process [23]. RAM is intended to utilize abstraction in a somewhat similar way to what Wiegers and Lauesen describe. It gathers all requirements for a product in one repository where requirements are linked between levels of abstraction. This makes it possible to perform fast screening and dismissal of requirements not in line with product strategies.

The requirements work is performed by the product management organization, using RAM independently in a continuous RE effort designed for long term product development. Projects that realize requirements are a result of the work conducted, i.e., RE activities using RAM are not project initiated, but rather *initiates* projects as needed. This entails handling a large and steady stream of requirements continuously before any commitment or plan exists for realization of them. Early screening of requirements (comparing to product strategies) is only one part, another is the need to select requirements for realization in projects. Having requirements on different homogenous abstraction levels enables comparisons, prioritization, and planning on each level. This facilitates for product management to perform early screening and selection of requirements. This differs from e.g., Wiegers and Lauesen who focus more on requirements engineering in relation to projects.

3. Planning of the evaluation

In this section, we describe the overall parameters for the evaluation. These parameters are common for all parts of the study and research questions described in Section 4.

3.1. Context

The study is conducted in an academic setting, with the help of graduate and undergraduate students at Blekinge Institute of Technology, Linköping University and Umeå University. The evaluation was conducted as an exercise in different courses at the universities.

The intended target audience for RAM, however, is product managers with several years of experience in a specific domain and of a specific product. In the study, the subjects have no training on RAM, possess limited domain knowledge, are under time pressure, and have not seen the requirements before. There is thus a considerable gap between the intended target group and the sample used in this study. The subjects in our study can be expected to adopt a more surface oriented approach to the problem than product managers. We argue that this works to our advantage, since any effects that we measure are likely to stem from the instrumentation and the use of RAM, rather than previous experiences of the participants in the study. If RAM proves to be usable in the study, it would indicate that the model is able to decrease the dependency on individual persons' experience, knowledge, and methodology.

3.2. Subjects

As mentioned, the participants in the evaluation are software engineering or computer science students at the three participating universities. This group consists of graduate and undergraduate students. Some of the participants are international students, the rest are Swedish. The subjects have an average of 0.51 years of industrial experience, and consider themselves between novices and of moderate experience regarding requirements engineering, and between "moderate" to "skilled" in English.

3.3. Design

Prior to the controlled evaluation presented in this paper two test rounds and one live round were performed. An initial version of the study was constructed and assessed with the help of colleagues at Blekinge Institute of Technology. Based on the experience from this test round, the study package was revised. Specifically, it was reduced in size and the questionnaires were changed to provide a closed set of possible motivations, to facilitate analysis. This revised study package was then assessed a second time with participants from a network of Swedish requirements engineering researchers, SiREN[1]. After this, the controlled evaluation was performed live with participation of Software Engineering Master's students at Blekinge Institute of Technology. The exact same study package was then distributed to colleagues within the SiREN network in order for the controlled evaluation to be executed at their respective university.

The controlled evaluation consists of four parts, all run consecutively without any breaks between. The first part is a preparatory lecture where RAM and the concept of a hierarchical requirements specification are introduced, together with a presentation of the study and the research instruments. The remaining three parts of the controlled evaluation, further described in Section 4, consists of material and questionnaires to answer the research questions. All participants are given the same instrumentation, the same treatment, and the same questionnaires to answer. The study is estimated, after the pilot tests, to take a maximum of 3 h if no breaks are allowed.

3.4. Instrumentation

The instrumentation consists of three different questionnaires, further described in Section 4, and a requirements specification. This requirements specification details an intranet solution for a course management system. The system contains features such as news (information) in courses, file archives for courses, course calendars, course discussion forums, and management of course participants. The requirements specification is based on a 16,000 person–hour students' project. From this project different features have been added and removed to generate a requirements specification of adequate size. None of the subjects had any knowledge of this actual student project, although they are all familiar with this particular type of system. The requirements are constructed to be of moderate to good quality based on the authors' experience as software engineers and requirements engineers as well as requirements engineering researchers. This was confirmed through the two test rounds. To ensure that the requirements specification represents a complete system of adequate size (we estimate that it would take 15,000 person–hours or more to develop the system) and still be small enough to be usable in the evaluation (i.e., 120 requirements in total) many of the more detailed requirements have been omitted.

The requirements specification is structured hierarchically into four levels, denoted Product (containing 9.6% of the requirements), Feature (19.1%), Function (59.6%), and

[1] http://www.siren.lth.se/

Component level (11.7%). The levels are further discussed in Section 2.1. Each requirement contains the fields *Id* (a unique number where also the abstraction level can be discerned), *Title* (title of the requirement), *Description* (the actual requirement), *Rationale* (an explanation of the need for the requirement), *Restrictions* (any risks or restrictions on the requirement), *Relations to* (id and title of requirements that this requirement is related to), and *Relations from* (id and title of requirements that link to this requirement). The requirements are listed in tabular form as the example in Fig. 2 illustrates. In this figure we see a selection of requirements on all levels, starting with two product level requirements, two feature level requirements, four function level requirements, and two component level requirements. Each requirement contains the aforementioned fields.

3.5. Validity evaluation

The validity of the study is divided into four areas: conclusion validity, internal validity, construct validity, and external validity. Below we discuss these in further detail. More information about possible threats to studies can be found in Wohlin et al. [26] and Robson [20].

3.5.1. Conclusion validity

To ensure that the research instruments, including the posed questions, are of a good quality, two separate test rounds with the material, using requirements and software engineering researchers, have been executed before the "live" rounds (i.e., the studies at Blekinge Institute of Technology, Linköping University and Umeå University). Moreover, all participants received the same introductory lecture, were given the same material in the same order, and received the same additional instructions. It is thus unlikely that the instrumentation and the implementation of the treatment influence the results unduly. That being said, since we use the answers of human subjects the gathered measures are of course not 100% repeatable.

3.5.2. Internal validity

The participants may mature during the study. Specifically, this concerns their understanding of the requirements specification and how to read it. This means that the accuracy of the participants' answers may improve as they answer more questions. In effect, as the study progresses the participants' familiarity with the domain and the requirements specification becomes more and more alike the intended target group.

Level	ReqID	Title	Description	Rationale	Restrictions	Relations To	Relations From
Product	1 013	Product Interface	All access to the system must take place via the sytems own user interface, i.e. access from third party products is not allowed.	Control look and feel. Avoid security issues and compability problems.		1 040 Manage and Conduct a Course	2 084 Web-based User Interface
Product	1 040	Manage and Conduct a Course	The product shall provide functionality that supports a teacher (course administrator) to conduct a course, manage the course, and supports the participants in following course progress, course news, and accessing information related to the course as well as exchanging course related information	This is the core of the product.			1 013 Product Interface 2 032 Course Start Page 2 035 Course News 2 039 Course File Archive 2 067 Course Administration 2 107 Discussion Forum
Feature	2 032	Course Start Page	Each course shall have a course start page.	Information overview		1 117 Distribute Information about Courses 1 040 Manage and Conduct a Course	4 022 Link to Course 3 065 Course Start Page Contents
Feature	2 035	Course News	It shall be possible to attach news items to a course.	Keep the students informed about the course.		1 117 Distribute Information about Courses 1 040 Manage and Conduct a Course	3 038 Remove Course News 3 129 Access to View Course News 3 128 Access to Add, Edit and Remove Course News Items 3 127 Edit Course News 3 065 Course Start Page Contents 3 037 View Course News 3 036 Add Course News 3 069 Course Administration Contents
Function	3 036	Add Course News	It shall be possible to add news items to a course.			2 035 Course News 2 154 Support Swedish Alphabet	3 128 Access to Add, Edit and Remove Course News Items
Function	3 041	Add Files to Course File Archive	It shall be possible to add files to the file archive.		What if a file already exists with the same name?	2 039 Course File Archive	3 130 Access to change in Course File Archive
Function	3 042	Remove Files from Course File Archive	It shall be possible to remove files from the course file archive.			2 039 Course File Archive	3 130 Access to change in Course File Archive
Function	3 043	Download files from Course File Archive	It shall be possible to download files from the Course File Archive.			2 039 Course File Archive	3 131 Access to use Course File Archive
Component	4 018	Successful Login	When a user has successfully logged in, the product shall display the user's personal start page.	After user authentication, the user wants to start working.		2 020 Personal Start Page 3 012 Login	
Component	4 019	First login	The first time a user logs into the system the user's personal profile shall be shown.	The user may provide additional information about him/herself.	The user decides not to provide any additional information, which means that the user has an incomplete profile.	2 047 Personal Profile 3 012 Login	

Fig. 2. Example of requirements representation.

The instrumentation (i.e., the requirements specification described in Section 3.4 and the related artifacts described in Section 4) is designed to be of moderate to good quality to mimic the situation for the intended target group and should not influence the participants unduly. Moreover, to ensure that the participants answer in accordance with the research instruments and not according to their understanding of the domain of the system, we have intentionally left out some specific features that commonly occur in the domain, such as using standard USENET news clients, import and export users and data to other systems, and synchronization of calendars with handheld devices. All of the product statements and candidate requirements used in part I and II of the study (further detailed in Section 4) are common in the domain, and the course management systems that the students are in daily contact with have features equivalent to the product statements and candidate requirements. This ensures that the students provide a motivation for their answers based on RAM and the provided requirements specification and not according to their domain understanding or their opinion.

The motivation of the subjects may vary considerably, since they are students and do not have any real interest in the product. This may actually speak in favor of the study, since there is less incentive for the participants to perform well.

3.5.3. Construct validity

Since we are using a single requirements specification in this study, there is a risk of mono-operation bias, i.e., there is a risk that our choice of system and domain influence the results more than the method for working with the requirements specification that we wish to evaluate. As mentioned above, by intentionally leaving out commonly occurring features in the domain from the requirements specification we at least ensure that the participants follow the prescribed method and that we can detect if they do not.

We make no secret of the hypotheses in this study. Since there is only one treatment the only way the participants can influence the result is by deliberately answering wrong. This increases the risk that the evaluated methodology is deemed to be less useful than it is, which means that we may err on the cautious side when analyzing the results.

To avoid the risk that the wording of the questions may provide hints as to what the answer is, the questions in the questionnaires are formulated in a standard way. This is further described in Section 4.

3.5.4. External validity

To ensure the external validity and the ability to generalize the results, we use a requirements specification for a relatively large system in a fairly mature domain.

As discussed in Section 3.1, the knowledge and experience of the participants is less than that of the target audience (e.g., product managers in industry). To reduce this gap, we use a system from a domain that is familiar to the subjects. The target audience would also have training in using RAM and specifying requirements according to RAM, which the study participants do not have. In this study, we only focus on a few key aspects of RAM, and thus a complete training of RAM is not necessary. We also argue that more experience, knowledge, and training should not negatively impact the ability to work with RAM. In other words, any positive effects detectable in this study should be transferable to the target audience.

In the same way, it is our belief that a hierarchical requirements specification works best when used with a software tool. In this study we use paper printouts which may impact the readability and the ease by which the participants may access the information. Hence, also here any positive effects are transferable to the target audience and the target environment.

Moreover, we use subjects from three different universities, with different backgrounds, corresponding to a large portion of the software engineering and computer science students educated in requirements engineering in Sweden during 2004 and 2005.

4. Research questions

The study is divided into three parts, with research questions attached to each part. Part I and II directly correspond to the problem statement posed in Section 2.2 and concern the evaluation of RAM. Part III is a post-test designed to gather information about the subjects to ascertain if and how their background influences the results of the study.

4.1. Part I

(Research Question 1)

Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent do the top-level (most abstract requirements) connect to the rest of the requirements?*

(Test)

This research question is evaluated by:

(a) Removing the most abstract requirements (Product Level) from the requirements specification.
(b) Presenting the Product Level requirements as statements about the product.
(c) Asking the subjects to determine whether each statement is supported by the remaining requirements specification, and in particular which of the requirements on the lower level in the requirements specification that supports the statement.

(Collected metrics)

The questionnaire for this part of the study collects metrics on:

(1) Whether or not the participants consider the statement supported by the requirements specification as a defining factor of the system.

(2) Which feature requirements (if any) the participants think support the statement (we use this during the analysis of metric 1, since this motivates the participants' answers).
(3) How deep (i.e., to which abstraction level) the participants had to look in the requirements specification to reach a decision.
(4) Time to complete part I.

(Design)
The subjects are presented with 12 statements of which 7 should be included according to the study design. For each of these statements the subjects shall answer whether the statement is supported by the other requirements (on lower abstraction levels), and motivate their answer by linking the statement to one or several requirements. Being able to correctly connect the statements to the rest of the requirements would indicate that there is a strong connection between the Product Level requirements and the rest of the specification. In addition, the subjects are asked to answer how deep they had to look into the requirements specification to reach their decision.
(Comment)
This research question evaluates whether the linking of abstract top-level requirements to more detailed ones (Feature Level and down) is possible (i.e., whether it is possible to determine if an abstract statement is in line with the product or not). If this is the case, and if the abstract Product Level requirements are able to summarize lower level requirements, they can be used to get a big-picture product overview.

4.2. Part II

(Research Question 2)
Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent is it possible to determine if new requirements should be included in the requirements specification based on the limits set by Product Level requirements?*
(Research Question 3)
Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent is it possible to determine at what level of abstraction a new requirement shall be inserted*?
(Test)
We evaluate these questions by:

(a) Presenting the requirements specification in its entirety, i.e., all four abstraction levels.
(b) Providing a number of new candidate requirements, and for each of these asking the participants to determine:
- on what abstraction level this candidate requirement fits, and
- if it should be included or not in the system based on the already present requirements in the specification (governed by the top-level Product Requirements).

(Collected metrics)
The questionnaire for this part of the study collects metrics on:

(1) Whether the participants are able to decide (with a plausible motivation) whether to include each requirement or not.
(2) Whether the participants are able to place (with a plausible motivation) the requirement on a specific abstraction level.
(3) Whether the participants are able to link each candidate requirement to the existing requirements in the requirements specification. These links serve as a motivation for why a requirement should be included or not.
(4) The reason(s) the participants give as motivation when a requirement should not be included.
(5) Time to complete part II.

(Design)
The participants are presented with 10 candidate requirements of which 5 should be included according to the study design. The participants have the choice of placing each candidate requirement on Feature, Function or Component Level. Product Level is not offered as an alternative as these requirements are considered locked for the sake of the study. Subsequent to placing the requirement on a certain abstraction level the participants are asked to decide whether it should be included in the system or not, and to motivate their decision. If they choose to include the candidate requirement, it should be linked to the existing requirements on the abstraction level above by giving a direct or indirect link to one or several Product Level requirements. If they choose not to include the candidate requirement the participants are asked to motivate why, using the categories "Not supported by Product Level", "Undeterminable Level", "Cannot be linked to requirements on the level above", "Contradicts another requirement", "Unclear requirement", and "Other."

4.3. Part III

(Research Question 4)
What is the background of the participants in the study?
(Research Question 5)
Does the participants' background substantially influence the results of Part I and II?
(Test)
Post-test questionnaire with a number of subjective assessment questions.
(Collected metrics)
For each of the participants we gather the following information:

(1) Years in software engineering curriculum.
(2) Number of study points.
(3) Years of industrial experience in software engineering.

(4) English skills (rated: Novice, Moderate, Skilled, and Expert).
(5) Requirements engineering skills (rated: None, Novice, Moderate, Skilled, and Expert).
Assessment of domain understanding before and after the evaluation.
Assessment of ease of understanding and use of: abstraction levels, placement and work-up, RAM, and requirements representation.

(Comment)
The purpose of this part is to collect background information on the subjects in order to ascertain whether aspects like language skills, experience etc. influenced the results obtained in Part I and II. Although the subjects are students, their experience and skills often vary considerably, especially since the students are from three different universities, and some of them are international students. The subjects may have several years of industry experience in their background, their English skills may differ, and some have worked in development projects as requirements engineers. Thus, this post-test collects information that is used to describe the subjects in Section 3.2, and serves as a help to understand their answers during analysis.

5. Operation

5.1. Preparation

The subjects were not aware of the aspects that we intended to study, and were not given any information regarding research questions in advance. They were aware that it was a controlled evaluation in the area of requirements engineering. The evaluation ran over a period of 3 h, and all of the subjects were seated in the same room.

Introduction to the study was given during these 3 h in the form of a brief slide show presentation. In this presentation some basic concepts of RAM were shown pertaining to abstraction and levels, how the requirements are connected, etc. The focus of the presentation was put on presenting the instrumentation which consisted of:

- Purpose, i.e., evaluate RAM according to Part I and Part II (as seen in Section 2.2 and Section 4).
- Requirements specification (ordered hierarchically and with connections between the requirements across levels).
- The questionnaires used to gather the results, and how they should be interpreted.

In addition to this it was made very clear to the subjects that the requirements already present in the requirements specification (RAM) should govern their answers, i.e., the statements provided as a part of the questionnaire for Part I should be accepted or dismissed with regards to the requirements present in the specification, and not according to what the individual subjects “thought” or “wanted”. For Part II, the same general principle was emphasized. The placement of new candidate requirements on appropriate abstraction levels was to be example driven, using the requirements already present in the specification as “good examples”. The decision to accept a new candidate requirement into the product or dismiss it should be governed by the Product Level requirements and the ability to link the candidate requirement to them directly or indirectly.

5.2. Execution

At each university, the evaluation was executed in one day over a 4 h session (3 h plus one additional hour if some students should need extra time). The requirements specification used in the first part of the evaluation was handed out at the start of the presentation, to give the participants a chance to familiarize themselves with the format. After the presentation, the questionnaire for the first part was handed out. As the participants completed the first part, they brought the questionnaire and the requirements specification to the front of the room where they were given the material for part two (a new questionnaire and the complete requirements specification). The start time and delivery time was noted on the questionnaire. As part two was completed the participants were given the last part of the study to complete, after which they were allowed to leave the room. No breaks were allowed during this time. The mean and median times to complete Part I and Part II were around 30 min, the shortest times spent on each part were around 15 min and the longest were 50 min for Part I and 1 h for Part II. After 3 h all subjects were finished and had handed in all material. All subjects completed all parts of the study.

6. Results and analysis

As a first step in presenting and analyzing the collected data we use descriptive statistics to visualize the results, presented for each part (Part I and Part II). The results are then analyzed and a summary for each analysis step is presented. In Section 7 the research questions are revisited with the help of the analysis results.

6.1. Part I

In total 12 statements were given to the subjects, and they had to decide which of these 12 were in accordance with the product. The product was represented by (described by) requirements on Feature, Function and Component Level in the specification. Seven out of the twelve statements were statements designed to be supported by the requirements specification. During the analysis of Part I, we have considered those answers that are in line with the study design and aptly motivated as “correct”. If the answer is in line with the study design but missing a proper motivation (i.e., not linked to acceptable Feature Level requirements) or if the answer is not in line with the study design, the answer is considered “incorrect”.

Table 1
Number of correct answers and depth divided by subject

	# Correct answers	Depth
Average	11.22	2.34
Median	12.00	2.00
StdDev	1.291	0.5831
Min	4.00	2.00
Max	12.00	4.00

Table 2
Number of correct answers and depth divided by statement

Statement	# Correct	Average depth
S1	147	2.48
S2	170	2.42
S3	177	2.06
S4	153	2.71
S5	174	2.25
S6	170	2.06
S7	168	2.43
S8	170	2.43
S9	165	2.50
S10	171	2.51
S11	171	2.11
S12	172	2.06
Average	167.33	2.34
Median	170.00	2.00

Table 1 shows the average and median results for all 179 subjects, the number of correct answers (second column), and the depth measured in abstraction levels that the subjects had to use in formulating an answer. The depth is calculated following the levels presented to the subjects in Part I, i.e., 1, Product Level; 2, Feature Level; 3, Function Level; and 4, Component Level.

The subjects' average is 11.22 correct answers (out of 12 possible) with an average depth of 2.34 (where minimum = 2, maximum = 4). This implies that most subjects were able to use Feature level requirements to render a mostly correct decision regarding the statements.

Table 2 shows the results from Part I but divided by statement instead of subject. For every statement (S1, S2, etc.) the number of correct answers is displayed (maximum is 179 correct answers for every statement). In addition, the average depth for each statement is displayed.

The depth of the investigations in order to render an answer is better illustrated through a histogram, presented in Fig. 3[2]. Looking at the correct answers (black bars) most are centered on a depth of 2 (>73%). Looking at the incorrect answers however, the spread is more pronounced as most subject lie between depth 2 and 3. This implies that in the instances where the subjects gave an incorrect answer they also looked deeper in the specification, i.e., the Feature level was not enough and they had to look at the Function (~26%) or even the Component level (~17%).

[2] For the purpose of the histogram the values were normalized to enable comparison.

Table 3
Answers ordered according to valid and missing answers for correct and incorrect answers

	Valid		Missing		Total	
	N	Percent	*N*	Percent	*N*	Percent
Correct	1834	91.33	174	8.665	2008	100
Incorrect	121	86.43	19	13.57	140	100

Data for Fig. 3.

To check the validity of this implication a Mann–Whitney test was performed comparing the two populations (Correct and Incorrect) to check whether this was a coincidence. A significance of 0.000004 was obtained. This indicates that there is a significant difference between the mean values of the two groups, i.e., indicating that the difference between how deep the two populations look in the requirements specification is not a coincidence. An independent sample *T*-test gave a significance of 0.00006.

The Mann–Whitney test was chosen for the large differences in group size between correct and incorrect, see Table 3, and due to the fact that the data are not normally distributed (the Shapiro–Wilk normality test gives $p < 0.0000000001$, whereas 0.05 or larger would have indicated a normally distributed data set).

Table 3 presents a summary of the answers regarding depth, where *N* denotes amount of answers (as to what depth was used) for correct and incorrect determinations. Fig. 3 uses the data from the "valid" column. In some cases, the answers regarding depth used were missing (8.7% of the cases for correct and 13.6% for the incorrect, as can be seen under the "missing" column). In total 1834 answers that were correct specified their depth and 121 answers for incorrect specified their depth.

Summary of Part I

A clear majority of the subjects gave a correct answer regarding the statements, i.e., 93.5% of the answers were correct (2008 correct answers out of 2148 possible). Further, the subjects rendering a correct answer have not looked as deep into the specification (requirements hierarchy) compared to the subjects rendering an incorrect answer. The Feature level (level 2 from the top) was the most used level for correct answers, while both Feature and Function level were used when rendering incorrect answers. This implies that it is possible to relatively quickly make a correct decision regarding whether or not a particular statement is supported by the requirements specification. If there is hesitation (and a need for more information), there is a larger risk to make a mistake. Or to put it in other words, going down several levels might not improve the correctness of an answer.

6.2. Part II

In total the subjects were given 10 new candidate requirements (CRs). For each of the CRs, the subjects were asked to:

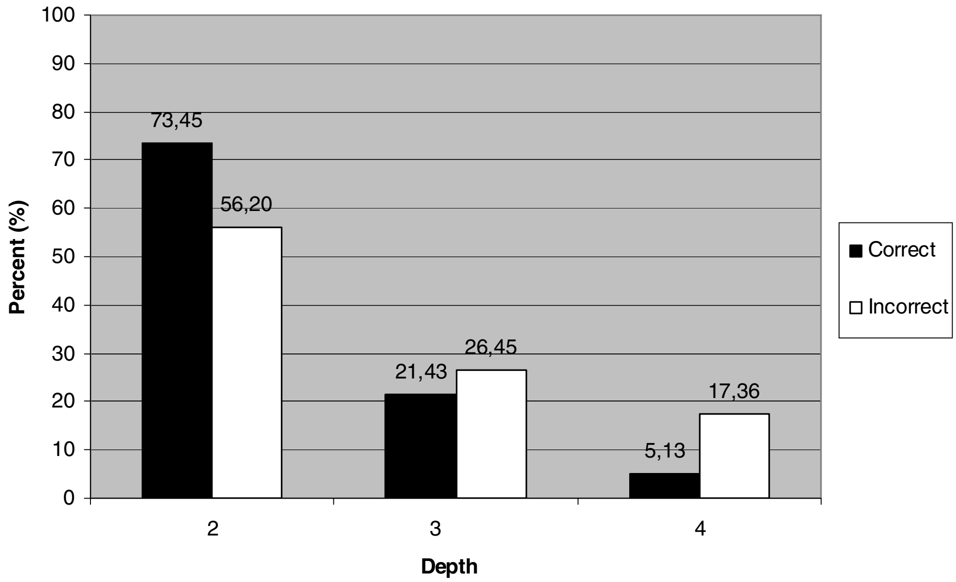

Fig. 3. Comparison of depth distribution between correct and incorrect answers.

1. Place the CR on an appropriate abstraction level *and* motivate this placement through links to one or several requirements on the abstraction level above. We call this "abstraction level placement".
2. Decide whether or not to include the CR in the product *and* indicate with links that the CR is in line with other already present requirements in the specification, *or* motivate their decision for not including the CR. We call this "include decision".

The "product" in this case was represented and described through the requirements already present in the hierarchically ordered specification (on four levels: Product, Feature, Function, and Component).

Table 4 gives an overview of the results divided by candidate requirements (CR1, CR2, etc.). The answers are divided into two main categories, visible in the table as *Correct* and *Fail*. These two main categories are divided into subcategories, further detailed below. By dividing the main categories into several sub-categories, we are able to further scrutinize the answers. It should be observed that in Part II, the subjects have to provide four things for every CR; abstraction level placement, include decision, and motivations for these two. Hence, an answer can be correct or wrong to different degrees, depending on which parts of the question the subjects manage to complete correctly. These distinctions are important for the analysis and the reason for the sub-categories.

Correct

The correct category contains those answers where the subjects have managed to use RAM and the requirements specification in a correct way. This means that the subjects have been able to place the CR on an acceptable abstraction level and motivate this by linking to other requirements, and have also been able to make and adequately motivate an include decision. The correct category is divided into two sub-categories, i.e., *correct according to study design and with proper motivation*, and *correct with proper motivation.*

Table 4
Summary table showing correct and failed answers

Candidate requirement	Correct		Fail		
	According to study design and with proper motivation	With proper motivation	Include-OK	Level-OK	Not-OK
CR1	88	47	14	9	21(1)
CR2	27	16	42(7)	56	38(2)
CR3	35	1	51(5)	75	17(1)
CR4	48	31	8	60	32(3)
CR5	56	20	44(7)	41	18(1)
CR6	31	36	33(2)	34	45(7)
CR7	86	44	17	18	14(3)
CR8	88	16	45(8)	21	9
CR9	87	29	17	20	26(3)
CR10	91	7	62(11)	4	15
SUM	637	247	333	338	235

Correct according to study design and with proper motivation
This category holds the answers that are exactly as intended during the study design, i.e., the abstraction level placement and the include decision are the same as the study design intended. The motivation is also specified according to study design.

Correct with proper motivation
This category is similar to the category above, but the links used for motivating the answers (i.e., abstraction level placement and include decision) are not always the same as the ones pre-specified during the study design. The links (and motivation for them) stated by the subjects were carefully scrutinized and considered to be acceptable. This implies that the subject either stated links that were not caught during the study design, or that the requirements themselves are possible to interpret differently than what was done in the study design. However, the central issue with the answers in this category is that both the placement of each CR and the corresponding include decision are ok, and the motivation for these decisions (in the form of links or exclusion motivations) are fully acceptable. During the analysis, correct answers missing proper and well motivated links were *not* given the benefit of the doubt, and were instead put in one of the Fail category.

Summarizing the correct category, 884 answers out of a total of 1790 are correct in that they have acceptable abstraction level placements and include decisions, using acceptable motivations. Of these, 637 answers are entirely according to study design, and an additional 247 use a different but proper motivation.

Fail
The second category, "Fail", holds all the answers where some aspects of RAM or the hierarchical requirements specification have been misunderstood (or instances where the participants forgot to fill in information), resulting in answers that do not have an acceptable motivation. Within this category there are three sub-categories, in order to detail which aspects have been misunderstood or understood. These three categories, "Include-OK", "Level-OK", and "Not-OK", are further detailed below. It should be stressed again that the analysis has been conducted with the objective of finding as many faults as possible with the answers, and when there has been room for doubt the answer has been placed in the category least favorable for the study (i.e., as far to the right as possible in Table 4).

Fail(Include-OK)
Fail(Include-OK) indicates that the include decision is properly motivated by links, but that the abstraction level placement is not sufficiently motivated (please note that the placement itself may be according to the study design, but if a proper motivation is missing we place the answer in this category). Numbers within parenthesis in this case indicate missing information regarding abstraction level placement. For example in the case of CR2, 42 answers are deemed Fail(Include-OK) and out of these seven answers are missing information regarding abstraction level placement.

Fail(Level-OK)
The next sub-category Fail(Level-OK) is the opposite to Fail(Include-OK). The CR abstraction level placement is sufficiently motivated but the include decision lacks an acceptable motivation (as above, the actual abstraction level specified may be according to the study design, but not properly motivated).

Fail(Not-OK)
Fail(Not-OK) includes all answers without any acceptable motivation for neither the abstraction level placement nor the include decision. Numbers within parentheses in this case indicate missing information regarding abstraction level placement or include decision. As these answers are considered entirely wrong no further detail is given about answers in the Fail(Not-OK) category.

Summarizing the Fail category, in 333 out of 906 answers in the fail category the subjects managed to make an acceptably motivated include decision, in 338 answers an acceptably motivated abstraction level placement, and in 235 answers both the abstraction level placement and the include decision were missing or not acceptably motivated.

Summary of Table 4
Table 4 shows that the subjects answered correctly regarding both abstraction level placement and include decision in $637 + 247 = 884$ cases (49.4%). In $637 + 247 + 333 = 1277$ cases (71.3%) the CR was accepted as a part of the system or dismissed as not being part of the system, all with a proper motivation. In $637 + 247 + 338 = 1222$ cases (68.3%) the concept of requirements abstraction was correctly used with proper motivations.

Studying Subjects' Consensus
In this section, we investigate whether or not a general consensus can be seen in the subjects' answers. In an industrial setting the subjects would have cooperated through, e.g., consensus meetings, and in this section we discuss possible outcomes of such a work practice. In addition some answers from Part II (CRs) are analyzed further.

For the sake of the discussion in this section, we consider the study design to be produced by domain experts with considerable training in RAM, as well as the particular requirements specification, and can thus be considered as being "correct". The reason for this is to have a benchmark to compare the subjects' consensus against.

Table 5 shows the average results of the subjects' answers with respect to include decision and abstraction level placement. The first column indicates the answer intended in the study design for every CR, "1" indicating that the CR should be included in the product, and "0" indicating that it should not. Coding the subjects' answers to "1" or "0" in the same way enables us to cal-

Table 5
Average (consensus) answers in comparison to study design regarding include decision and abstraction level placement (significant deviations marked bold italics)

	Study design include decision 1 = Include 0 = Do not Include	Subject average include decision (Consensus)	Study design abstraction level placement	Subject average abstraction level placement (Consensus)
CR1	1	0.93	3	2.75
CR2	***0***	***0.47***	***2***	***2.75***
CR3	***0***	***0.50***	3	2.60
CR4	***1***	***0.44***	2	2.16
CR5	0	0.35	3	2.77
CR6	***1***	***0.43***	3	2.99
CR7	1	0.84	3	2.71
CR8	0	0.17	3	2.80
CR9	1	0.80	3	3.07
CR10	0	0.10	3	2.66

culate an average for each CR. For six of the CRs the subjects' consensus is fairly close to the answer intended in the study design. For example in case of CR1 "1" is to be compared with "0.93". This means two things. First, it means that the subjects' answers are in line with the intentions of the study design (i.e. expert product managers' opinions). Second, the relatively high number means that there is a large consensus among the students that this CR should be included. If this had been a requirements discussion meeting among managers there is a good chance that the consensus would be to include the requirement in the product.

In four cases such a consensus meeting could render the wrong decision (i.e., for CR2, CR3, CR4, and CR6), since there is less than 2/3 majority for the decision. There is no case where a clear majority of the subjects have an opposite opinion to the study design. Rather, the include decisions could go either way as the subjects' opinions were divided equally on the issues. Below, these four CRs are discussed further.

CR2
In the case of CR2 the subjects that want to include the CR motivate their decision through linking to already present requirements in the specification. CR2 is (unintentionally) formulated such that it can be interpreted to fit with already present requirements. The same is true for the opposite, i.e., a subject average of 0.47 could go either way although there is a tendency to dismiss the CR, as originally intended in the study design.

CR3
The case of CR3 is similar to the case of CR2. Several subjects link to already existing requirements as motivation for their decision to include CR3. Since the formulation of CR3 is such that this was possible, and the fact that there is no explicit requirement contradicting CR3 (e.g., on Product Level) half of the subjects answered that it should be included. It should be noted that there is no explicit support for CR3 either, which is the motivation behind the decision during study design that CR3 should not be included. In a consensus discussion (e.g., a requirements meeting) for CR2 and CR3 the issue of uncertainty would be discussed and should optimally yield a decision to either add a requirement on the Product Level explicitly supporting the inclusion of the CRs, or a slight reformulation of existing requirements to explicitly exclude them. The Product Level requirements are of course subject to change in a real product management/requirements engineering situation.

CR4
In the case of CR4 the subjects were more inclined to dismiss requirements than was anticipated during the study design. Although the study design provides opportunities for including a requirement like CR4, the subjects did not agree with this since explicit support for CR4 was missing in the Product Level requirements (according to the subjects).

CR6
In the case of CR6 the divergent view of the subjects can be explained to some extent with that several of the subjects misunderstood or misinterpreted CR6. The requirement is a Function Level requirement and hence rather technically specific. A lack of knowledge regarding web technology led many subjects to believe that the CR contradicted other requirements already present in the requirements specification, in particular that the use of a web interface to the system precluded the use of IP-address locking, which is not the case.

In the two rightmost columns of Table 5 the abstraction level intended by the study design and the average of the levels specified by the subjects are presented. Overall, the subjects' consensus and the study design intentions are comparable, with the exception of CR2, where the subjects' consensus places the CR on level 3 (Function Level) while the intended level was level 2 (Feature Level). Because the requirements model is example-driven, there is often not a definitive answer whether a requirement should be placed on a specific level or on one of the neighboring levels. In this case, there are examples on both the feature and the function level of requirements that can be considered similar to CR2.

In summary, the consensus whether or not to include a candidate requirement is most of the time comparable to

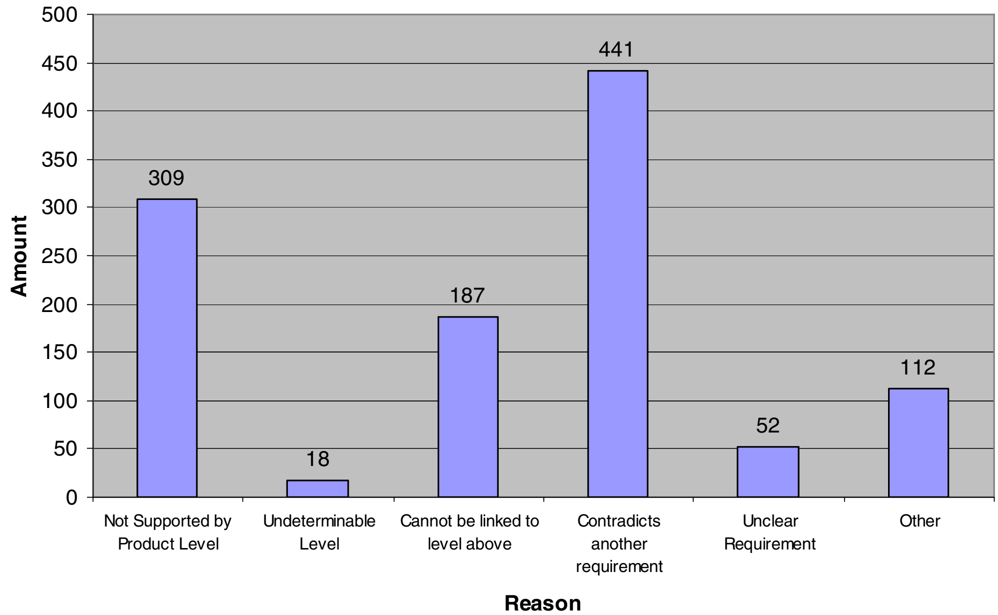

Fig. 4. Reasons given by subjects when they chose not to include a CR.

what was intended in the study design. The exceptions to this can be traced to lack of domain knowledge among the subjects, alternatively some insufficiencies in the study design (i.e., the requirements themselves could be misinterpreted). In these cases the subjects are divided fairly equally between accepting and dismissing a CR, as could be expected.

The consensus regarding abstraction level placement is close to the intention of the study design in all cases but one (CR2).

Reasons for Dismissing CRs

Fig. 4, finally, gives an overview of the reasons given by the subjects for dismissing a CR rather than including it in the product. Three motivations are in clear majority, i.e., that the CR is not supported by any requirements on the product level, the CR cannot be linked to any requirements on the abstraction level above, and that the CR contradicts another requirement already present in the requirements specification. These three reasons were also the ones used in the study design.

We also see that the remaining three categories (i.e., undeterminable level, unclear requirement, and other comments (free text comments)), that represent perceived problems in the study design, were rarely used and constitute a total of 16%. It should be noted that the subjects were able to mark more than one reason for excluding a candidate requirement.

In summary, the subjects are able to clearly and in accordance with the opinions of experienced product managers discern the reasons for dismissing a CR.

7. Research questions revisited

7.1. Part I

(Research Question 1)

Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent do the top-level (most abstract requirements) connect to the rest of the requirements?*

In the evaluation 93.5% of the answers are correct. The subjects giving correct answers used requirements at level 2 (Feature Level) to render decisions, while the subjects that answered incorrectly had a depth of between 2 and 3. This indicates that in order to get a correct answer it is in general only necessary to go as deep as Feature Level. This, in turn, indicates that Product and Feature Level are tightly coupled, which is a prerequisite for RAM and the idea behind abstraction levels.

The large amount of correct answers indicates that Product Level requirements are indeed general enough to summarize several requirements on lower levels. However, they are still concrete enough to determine whether they represent the product or not, and can thus be used to get a system overview.

7.2. Part II

(Research Question 2)

Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent is it possible to determine if new requirements should be included in the requirements specification based on the limits set by Product Level requirements?*

A total of 71.3% of the answers include or exclude the candidate requirements using acceptable motivations. In other words, more than 2/3 of the answers are correct in terms of a well motivated include decision. This indicates that the ability to compare requirements with similar requirements on the same abstraction level and the necessity to be able to link a new requirement to the existing hierarchy indeed supports the decision process. Time constraints prevented us from producing a non-hierarchically structured requirements specification to run a control group. However, offering all of the requirements structured according to for example functionality would have forced the subjects to read through much more information (requirements) before being able to render an include decision. Looking at the results regarding the relation between depth of study and answering correctly in Part I, there is indication that more information (larger amount of requirements) as a basis for a decision does not render a better decision, rather the opposite. In addition, there exists a multitude of ways to structure requirements, thus it is not possible to compare RAM structured requirements to a specific "normal" or "default" structured set of requirements.

Strictly speaking we cannot say that RAM structured requirements are better or worse than the current situation, since it depends on what the "current" situation is. However, the results indicate that it is much better than answering at random[3], and in conjunction with the answer to research question 3 we are convinced that it is a real help for product managers.

(Research Question 3)
Given a requirements specification ordered hierarchically according to the level of abstraction of the requirements, *to what extent is it possible to determine at what level of abstraction a new requirement shall be inserted?*

A total of 68.3% of the answers place the candidate requirements on their intended abstraction level using acceptable links to other requirements. It thus seems to be relatively easy to determine the abstraction level of a requirement by studying the "good examples" that the existing requirements specification provides.

7.3. Part III

(Research Question 4)
What is the background of the participants in the study?
(Research Question 5)
Does the participants' background substantially influence the results of Part I and II?

These questions were mostly used in the analysis of the results. The information gained regarding research question 4 is found in Section 3.2. For research question 5, we were unable to find any difference between the performance of participants with more experience and those with less experience (industry experience or experience with requirements engineering). It should be noted, however, that the differences in terms of experience between the participants were not very large (a standard deviation of one year for the industry experience). The homogeneity of the group may be the reason why no difference was found.

8. Summary and conclusions

In this study we evaluate aspects of a hierarchical requirements abstraction model – RAM – as a preliminary step before conducting full scale industry piloting. Using industry as a laboratory through pilots is a powerful tool and a part of building trust and commitment prior to technology transfer in the form of process improvement activities. However, prior to using valuable and hard to obtain industry resources we consider it prudent to evaluate the ideas in an academic laboratory setting. Initial evaluation of the model in academia intends to investigate whether or not industry piloting is mandated, or whether RAM needed further refinement or even redesign before directly involving industry in validation activities. It is thus important to get early indications whether or not the fundamental features inherent in the current model are useful and usable, and if the assumptions implied by the model hold.

RAM is intended to aid product managers and others involved in requirements engineering in getting an overview of a product using relatively few abstract Product level requirements. Using the same Product level requirements as a basis (and the requirements linked to them), managers should be able to include or dismiss new incoming requirements in a repeatable way. That is, the include decision should not be dependent solely on the person performing the work. Moreover, managers should be able to place new requirements on appropriate abstraction levels in the model.

The participants in this evaluation were given a relatively large amount of requirements (120 in total) and were asked to accomplish a considerable amount of work in a short period of time. The participants were expected to form an understanding of the concept of requirements abstraction and hierarchically structured requirements specifications, understand the domain (read and understand the requirements), and then solve the tasks in Part I and II. Very little training of the model was given to the participants, and they also possessed little prior knowledge regarding the domain when compared to a product manager. Considering these aspects and the time spent the results produced are encouraging and point towards both high usability and usefulness of RAM.

The characteristics of industry are also relevant as real-life usage of RAM should be easier than during the evaluation. In industry, requirements engineering is not performed in isolation (as was the case in the evaluation); regular meetings as well as official and unofficial conversations and discussions help in sharing views and forming consensus as well as a shared understanding. From this perspective the results obtained in the evaluation are even more promising. The consensus results presented in Table 5 indicate that had there been cooperation and exchange between the participants even better results may have been achieved, since cooperation often yields better results than

[3] A rough calculation yields 1/2 chance to make the correct include decision, 1/3 chance for correct abstraction level placement, approximately 1/5 chance to link to at least one appropriate requirement on the level above. This would yield approximately an 1 in 30 chance to "guess" correctly.

working individually. In addition, industry product managers are often senior practitioners well versed in both their specific domain and in the field of requirements engineering. Given this some of the mistakes made by the subjects participating in the controlled evaluation, where lack of domain knowledge was evident, could probably have been avoided in an industry setting.

As indicated by the results of the controlled evaluation in this article, the concept of abstraction and the usage of abstraction levels seems to be fairly intuitive. This is also useful in a scope outside of RAM. For example, the possibility to obtain homogenous requirement abstraction levels using RAM may facilitate requirements prioritization, as requirements on the same level of abstraction are easily compared in contrast to comparing requirements on different levels of abstraction.

A potential issue with the model is that since the abstract requirements on the upper levels (especially Product Level) are few, their individual impact on the product is considerable when they are used as the principal decision support (getting a quick overview of the product). This implies that the demands on these requirements are high in terms of being well-formed and unambiguous. In addition they need to be explicit. In this study, lack of explicit support among the product level requirements was a motivation often used by the participants to dismiss a candidate requirement.

In industry, we may assume that the users of RAM possess domain knowledge as well as at least minimal training in using RAM together with some tool support. These factors should help ensure even greater usability and usefulness of the model than was observed during the controlled evaluation in this article.

8.1. Future work

As a future study we intend to run control group evaluations performing the same tasks but on requirements specifications structured in several “traditional” ways (e.g., structured according to functionality, stakeholders, or use cases). The comparisons of the results could give further information regarding the usefulness and usability of RAM, after which the model may be considered mature enough for industry piloting. Several of these evaluations are planned for the near future and preparations for some of them are already underway.

References

[1] A.I. Anton, Goal-based requirements analysis, in: Proceedings of the Second International Conference on Requirements Engineering, IEEE, Los Alamitos CA, 1996, pp. 136–144.

[2] S.A. Butscher, M. Laker, Market-driven product development, Marketing Management 9 (2000) 48–53.

[3] J. Castro, M. Kolp, J. Mylopoulos, Towards requirements-driven information systems engineering: the Tropos project, Information Systems 27 (2002) 365–389.

[4] A.M. Davis, The art of requirements triage, IEEE Computer 36 (2003) 42–49.

[5] T. Gorschek, Software process assessment and improvement in industrial requirements engineering, in: School of Engineering – Department of Systems and Software Engineering, Blekinge Institute of Technology, Ronneby, 2004, p.154.

[6] T. Gorschek, C. Wohlin, Identification of improvement issues using a lightweight triangulation approach, in: Proceedings of the European Software Process Improvement Conference (EuroSPI’2003), Verlag der Technischen Universität, Graz, Austria. Download at: http://www.bth.se/fou/Forskinfo.nsf/, 2003, VI.1–VI.14.

[7] T. Gorschek, C. Wohlin, Packaging software process improvement issues – a method and a case study, Software: Practice and Experience 34 (2004) 1311–1344.

[8] T. Gorschek, C. Wohlin, Requirements abstraction model, Requirements Engineering Journal 11 (2006) 79–101.

[9] S.A. Higgins, M. deLaat, P.M.C. Gieles, Managing requirements for medical IT products, IEEE Software 20 (2003) 26–34.

[10] L. Karlsson, A. Dahlstedt, J. Nattoch Dag, B. Regnell, A. Persson, Challenges in market-driven requirements engineering – an industrial interview study, in: Proceedings of the Eighth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ’02), Universität Duisburg-Essen Essen, Germany, 2003, pp. 101–112.

[11] E. Kavakli, P. Loucopoulos, D. Filippidou, Using scenarios to systematically support goal-directed elaboration for information system requirements, in: Proceedings of the IEEE Symposium and Workshop on Engineering of Computer-Based Systems, IEEE, Los Alamitos CA, 1996, pp. 308–314.

[12] P. Kotler, G. Armstrong, Principles of Marketing, Prentice Hall, Upper Saddle River NJ, 2001.

[13] G. Kotonya, I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley, New York, 1998.

[14] S. Lauesen, Software Requirements: Styles and Techniques, Samfundslitteratur, Fredriksberg, 2000.

[15] D.R. Lehmann, R.S. Winer, Product Management, McGraw-Hill, Boston MA, 2002.

[16] H. Mintzberg, B.W. Ahlstrand, J. Lampel, Strategy Safari : A Guided Tour Through the Wilds of Strategic Management, Free Press, New York, 1998.

[17] C. Potts, Requirements models in context, in: Proceedings of the Third IEEE International Symposium on Requirements Engineering, IEEE, Los Alamitos CA, 1997, pp. 102–104.

[18] C. Potts, I. Hsi, Abstraction and context in requirements engineering: toward a synthesis, Annals of Software Engineering 3 (1997) 23–61.

[19] S. Robertson, J. Robertson, Mastering the Requirements Process, Addison-Wesley, Harlow, 1999.

[20] C. Robson, Real World Research: A Resource for Social Scientists and Practitioner–Researchers, Blackwell Publishers, Oxford, 2002.

[21] G. Ruhe, D. Greer, Quantitative studies in software release planning under risk and resource constraints, in: Proceedings of the International Symposium on Empirical Software Engineering (ISESE), IEEE, Los Alamitos CA, 2003, pp. 262–271.

[22] I. Sommerville, Software Engineering, Addison-Wesley, Essex, 2001.

[23] M. Weber, J. Weisbrod, Requirements engineering in automotive development: experiences and challenges, IEEE Software 20 (2003) 16–24.

[24] K.E. Wiegers, Software Requirements, Microsoft Press, Redmond WA, 1999.

[25] R. Wieringa, C. Ebert, Guest editors’ introduction: RE’03: practical requirements engineering solutions, IEEE Software 21 (2004) 16–18.

[26] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlson, B. Regnell, A. Wesslén, Experimentation in Software Engineering: An Introduction, Kluwer Academic, Boston MA, 2000.