Requirements Eng
DOI 10.1007/s00766-007-0047-z

ORIGINAL RESEARCH

Industry evaluation of the Requirements Abstraction Model

Tony Gorschek · Per Garre · Stig B. M. Larsson · Claes Wohlin

Received: 27 December 2005 / Accepted: 26 March 2007

Abstract Software requirements are often formulated on different levels and hence they are difficult to compare to each other. To address this issue, a model that allows for placing requirements on different levels has been developed. The model supports both abstraction and refinement of requirements, and hence requirements can both be compared with each other and to product strategies. Comparison between requirements will allow for prioritization of requirements, which in many cases is impossible if the requirements are described on different abstraction levels. Comparison to product strategies will enable early and systematic acceptance or dismissal of requirements, minimizing the risk for overloading. This paper presents an industrial evaluation of the model. It has been evaluated in two different companies, and the experiences and findings are presented. It is concluded that the requirements abstraction model provides helpful improvements to the industrial requirements engineering process.

T. Gorschek (✉) · C. Wohlin
School of Engineering, Blekinge Institute of Technology,
P.O. Box 520, 372-25 Ronneby, Sweden
e-mail: tony.gorschek@bth.se

C. Wohlin
e-mail: claes.wohlin@bth.se

P. Garre
Danaher Motion Särö AB, 429-80 Särö, Sweden
e-mail: per.garre@danahermotion.se

S. B. M. Larsson
ABB Corporate Research, 721-78 Västerås, Sweden
e-mail: stig.bm.larsson@se.abb.com

Keywords Requirements abstraction model (RAM) · Industry trials · Technology transfer · Evaluation · Market driven product centered continuous requirements engineering · Requirements abstraction · Product management · Requirements triage

1 Introduction

Requirements engineering (RE) more and more transcends project boundaries as market-driven product development is becoming increasingly commonplace in software industry [1–3]. 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 [4, 5].

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.) [6–8]. Large volumes of requirements from multiple sources risk overloading companies unless they can handle incoming requirements in a structured way, dismissing some, and refining some prior to allocating them to a development instance [9]. 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) [3], 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 organization, of which the requirements engineer in this case is instrumental, is central to making sense of the requirements as they are processed [10].

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 was confirmed by results from two separate process assessment efforts conducted in cooperation with Danaher Motion Särö AB and ABB [11, 12].

In response, a market-driven product centered requirements engineering model was developed, the requirements abstraction model (RAM) [13]. RAM is designed toward 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. A brief introduction to the model is provided in Sect. 2.1.

This paper presents two cases of RAM tailoring, implementation and most important evaluation, conducted at Danaher Motion Särö AB and ABB. The main purpose is to give a brief overview of how RAM was tailored to fit two different organizations, and how the model performed in terms of usability and usefulness based on evaluations performed with professionals using it in their day-to-day work, thus establishing the relative value of using RAM [14]. The main objective and contribution of the paper is to show how a general RAM can be tailored to two different companies and in particular the focus is on an evaluation of the industrial introduction of the model. This is achieved through the collection of data from industry users of the model approximately six months after its introduction. Two main perspectives are evaluated and analyzed. First we study the usage of the model, i.e., if and to what extent the model and accompanying process supported the work efforts (*actions*). Second we study the requirements and the effect the new way of working had on them (*quality attributes*).

The paper is structured as follows. Section 2 gives some background information and a short introduction to RAM to increase the understanding of the rationale behind the tailoring and evaluations performed. Section 3 introduces the companies and the product development situations where RAM is used, and gives some background information as to domain and requirements engineering practices used prior to RAM implementation. The concept of model tailoring and the implementation of RAM at the companies are presented in Sect. 4. In Sect. 5, the study design is presented. Section 6 presents the results from the evaluation, and Sect. 7 presents the conclusions drawn.

2 Background and related work

The development of RAM was performed in close collaboration with industry. Figure 1 illustrates this process. Industry needs and possibilities for improvements were identified through several process assessments [11, 12]. The assessment results then acted as a basis for model development [13]. RAM was subsequently validated in several incremental steps, both in industry (small scale pilots) [15] and academia through several experiments using senior students as subjects [16, 17]. Each round of validation was used to refine the model in terms of contents and structure, as well as test issues relating to usability and usefulness.

While active collaboration and industry presence of the researchers in industry was an inherent part of the research conducted during the initial model development and validation, the evaluation presented in this paper was not based on researcher participation. That is, challenges and possibilities for improvement were identified though process assessment activities in industry, the gradual development of the model was performed in cooperation with industry, but the ultimate test of RAM was done by industry practitioners using it in their everyday work, without the presence or involvement of the researchers.

The evaluation presented in this paper aims to scope the success of the technology transfer in the form of implementation of RAM in industry. The large-scale industry trials allow us to validate the models' usability and usefulness in a non-simulated environment. Feedback obtained

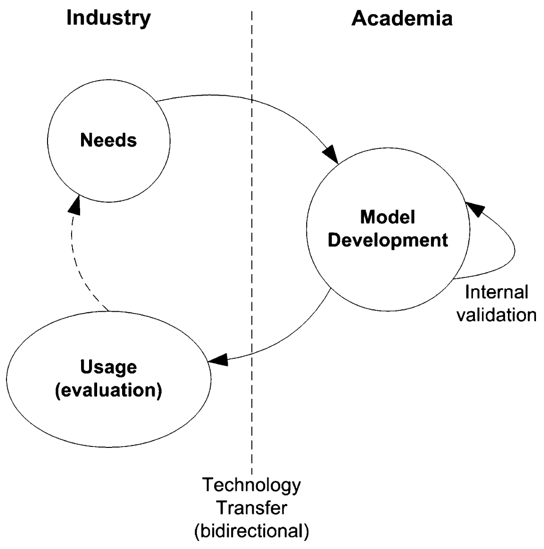


Fig. 1 Overview of RAM development

here will be used to further refine the model, completing the circle of technology transfer in which RAM was created. More details about the technology transfer process itself can be found in Gorschek et al. [15].

2.1 Introduction to RAM

The requirements abstraction model is a hierarchical abstraction model, and a method for working with requirements. It is 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. Figure 2 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, roadmaps, and 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.

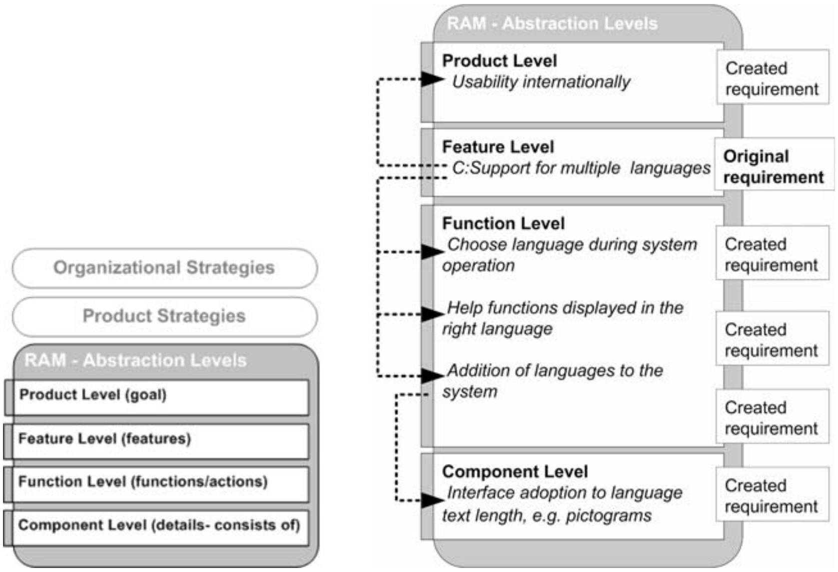
Briefly, the process followed while using RAM is that when requirements arrive they are placed and specified on an appropriate level by using good-example requirements suitable for the organization and product in question. These good-example requirements are specified during a RAM-tailoring workshop and there should be several good-example requirements for each main ''type'' of requirement typically encountered by the organization; examples can be several non-functional requirements, and functional software requirements. In addition to using good-example requirements a general comparison to the existing requirements base can be done for further guidance (i.e., placement can be seen as example-driven). This means that the present mass of requirements in general, and good-example requirements in particular, are used as decision support material for the specification 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 also breaking down high-level requirements to Function Level. This is done by creating new requirements in levels above and below and linking them to the original requirement. Figure 2 gives an example of this. The original requirement ''*C:Support for multiple languages''* (placed on feature level) is abstracted to product level through the creation of a new work-up requirement ''*Usability internationally''*, and broken down to 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. 2, if a new requirement comes in stating ''*C:Support imperial units*'' it could be placed on 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 product level and the new requirement can be linked to an already existing one.

The good-example requirements mentioned earlier are of course specified in the same manner, i.e., specified and worked-up. In fact, the example in Fig. 2 is a good-example requirement used at Danaher motion, one of the companies where RAM has been implemented.

During the work-up process, the original requirement is compared to product strategies/product roadmaps (as it is

Fig. 2 RAM abstraction levels and example of work-up



Organizational Strategies
Product Strategies
RAM - Abstraction Levels
Product Level (goal)
Feature Level (features)
Function Level (functions/actions)
Component Level (details- consists of)
RAM - Abstraction Levels
Product Level
Usability internationally
Created requirement
Feature Level
C:Support for multiple languages
Original requirement
Function Level
Choose language during system operation
Created requirement
Help functions displayed in the right language
Created requirement
Addition of languages to the system
Created requirement
Component Level
Interface adoption to language text length, e.g. pictograms
Created requirement

abstracted). 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 but rather the company wanted to limit the product market to the Scandinavian market. In this case ''*Usability internationally*'' would be ''*Usability 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 or not to include a requirement, since all requirements are directly linked to this level. If the new requirement is not supported on product level, it may indicate that the requirements should be dismissed, but it could of course also mean that a new product level requirement needs to be created.

The breakdown of the requirements stops on function level where the requirements are 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, but present since requirements in some instances were delivered in a very detailed form, and these requirements also needed 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. 2, 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.

Another example can be seen in Fig. 3. For Case A, a requirement comes in stating ''Use XML protocol''. As a ''requirement'' close to implementation/solution it is placed on component level. During work-up the underlying reason for the requirement becomes evident, i.e., in this case there is a need to send data to handheld devices from the product, thus a new requirement on function level is created *''Send data to handheld devices''*. The overall implication of this requirements demands that the product developed *''Support Communication with third party devices''*; thus an additional requirement is created on feature level. From an overall product perspective, supporting communication with third party devices is not self-evident, but the underlying reasoning is to support commissioning efficiency of the product at customer sites. This is enabled and usability is premiered through allowing service and installation technicians to use handheld devices during system setup. On the other hand, there is a second overall product decision to be made. The original requirement speaks of using XML as the communication language. As XML is an open standard the organization has to decide weather or not the system should allow communication with open standards or if some sort of lock-in is preferable. In Case A open standards are preferred; thus a new product level requirement is created *''Open Standards''*.

However, looking at Case B, using only a certain communication protocol (and not an open standard like XML) the organization can lock-in its customers to use a certain brand of handheld devices. If this were supported by product and company strategy the original requirement (to use XML) would be dismissed, but if the idea behind the requirement were good (i.e., offering handheld communication) the component level requirement would be modified to e.g., ''Use PCX protocol'' instead. This proprietary standard would still allow data to be sent to handheld devices, but as indicated by the feature level

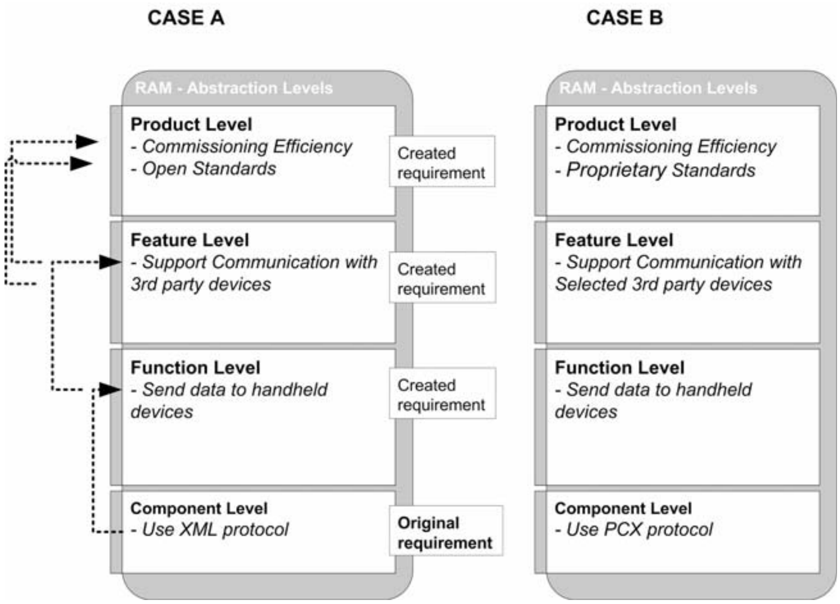


Fig. 3 RAM example of work-up of a component level requirement

requirement in Case B only selected third party devices would be supported. The ideas for the examples in Case A and B are taken from industry use of RAM, although modified to protect proprietary information.

As can be seen with the examples above the use of RAM demands that certain rules be upheld, called work-up rules, which are detailed below (for more details see Gorschek and Wohlin [13]):

- R1: *No requirement may exist without having a connection to the product level*

R1 can be met in one of two ways, one or more new *work-up requirements* are created, or the requirement in question is linked to already-existing requirements on an adjacent upper level. In either case, the original requirement is abstracted upward and can be compared (indirectly) to the product strategies/road maps.

In addition to abstraction, there may also be reason for requirements to be broken down enough to act as a basis for design (good-enough for project initiation). For a requirement to be detailed enough and on the right level of abstraction for this to be possible every requirement (on feature level or higher) has to be broken down to Function Level (testable and unambiguous). This creates the second work-up rule (R2):

- R2: *All requirements have to be broken down to function level*

As in the case of R1 this can mean creating one or more work-up requirements on adjacent levels, or linking the requirement in question to already existing requirements on lower adjacent levels. Either is acceptable as long as the requirement(s) on lower levels satisfy the upper level one.

Satisfy in this case pertains to the issue of breaking down a high-level requirement (from e.g., feature level) to function level, where the requirements on lower level together satisfy the original one to the extent of giving a foundation good enough for the initiation of realization (design) of the requirement. The main reasoning behind this is that requirements are meaningless if they cannot be delivered to a development effort (i.e., left on a too abstract level). Typically, R2 involves the creation of several work-up requirements being created.

It should be noted that the initial placement of a requirement is not final, i.e., in some instances it is necessary to iterate (go back) and rethink the initial placement of a requirement (as the work-up offers an analysis that may change the perception of the original requirement). This can also involve eliciting additional information from the requirement's source if this is possible.

During the work-up, it is important to stay true to the original requirement, or rather the intention of it. The creation of new requirements as a part of the work-up should not stray too far from the initial intention of the original requirement, and thus give rise to totally new requirements that are related to but outside the scope of initial intention. It is inevitable to create new requirements (especially if the original requirement is on a high abstraction level) as the work-up is designed to create new relevant work-up requirements to the extent that they satisfy the original requirement. However, this is not the same as including new requirements based on ‘‘this might also be a good idea’’ philosophy, as this could give rise to a mass of new requirements. As the model (and the usage of it) is aimed at offering support to professionals, it is also much dependent on the same professionals to assess how well it works. A recommendation when performing work-up of original requirements is always to ask the question ‘‘is this new (work-up created) requirement really necessary in order to satisfy the original requirement’’? If the answer is ‘‘yes’’ then there is no problem, but if there is uncertainty, the work-up should be stopped.

This does not mean that good ideas pertaining to new requirements should be discarded along the way in any case, but they should not be a part of the work-up, rather be specified as new original requirements on an appropriate level (and in turn get a work-up themselves).

2.1.1 Requirements work-up: discussion

Looking further into the process of work-up using RAM, several potential issues can be identified when applying the work-up rules (for more details see Gorschek and Wohlin [13]).

As mentioned before, no requirement may exist without having a connection to the product level (R1). This can imply that new work-up requirements are created on upper levels using lower level original requirements as a base. A potential issue during this activity is that an original incoming low-level requirement can give rise to (or be linked to) several requirements on a more abstract level. Two rules of thumb apply here. First, staying true to the intention of the original requirement is a priority. Inventing new requirements from scratch outside what is needed to work-up an original requirement can result in overwork of requirements at this early stage (product management). New requirements that are created as a part of the work-up process *are* separate requirements, but not *original* requirements. This distinction can seem somewhat trivial but it helps in keeping focus on expanding the meaning of the original requirements and the work-up of them instead of using the whole process as an excuse to create large amount of new independent requirements. Like all activities the requirements engineers, e.g., product managers, base this distinction on their expertise and judgment, and they must judge not only the impact on future development but also the impact on product strategy. Second, a new requirement that cannot be directly attributed to the work-up process, but still invented as a result of an original requirement idea, should not be dismissed, but rather stated as a new original requirement (inventor is the source), and then subjected to work-up of its own. Work-up has to be seen as a part of the analysis, refinement and specification of requirements, as well as a way to compare incoming requirements to product strategies.

Once the work-up is completed and there are distinct links from the original requirement upwards and downwards (if applicable) a requirement in the ‘‘chain’’ cannot be removed without considering the whole structure of the requirements. For example, removing a requirement on product level would result in that all requirements linked under it must either be re-linked to another requirement on Product Level, or all requirements in the chain have to be deleted as well.

The same consideration has to be taken when removing requirements on lower levels, i.e., removing a requirement on Function Level would demand an explicit decision stating that the above-linked requirement(s) can be satisfied after the removal (by other requirements linked to the feature level requirement in question), or the feature level requirement should also be removed. Looking at the example described in Fig. 2, this would mean that the removal of e.g., the function level requirement *‘‘Addition of languages to the system’’* would demand that an explicit decision be made that functionality for adding new ‘‘language sets’’ to the system should not be incorporated as a part of the feature of supporting multiple languages in the system. This in turn has implications. How should languages be added? Are some sets of them just added manually at development and so on? The decision as to when (in this case a feature level) a requirement is satisfied by lower level requirements is also a judgment call, but an explicit decision has to be made all the way up to product level regarding adequacy and satisfaction. Ultimately management (both business and technical) have to decide if a certain chain of requirements are complete and within the product strategies or not.

2.1.2 RAM: specification

Requirements abstraction model prescribes the use of attributes for the specification of requirements on all levels of abstraction. The reasoning behind this is to treat the requirements as objects with attributes attached to them, but also in different states. The tailoring of RAM to fit a certain organization (see Sect. 4) to a large extent controls the attributes used, although certain information is mandatory for the purpose of enabling traceability. Below a sample list of attributes can be seen.

It should be observed that every organization has different needs and possibilities to specify and utilize attributes. Although the attributes above are recommended a bare minimum could be Id, title, description, original, level, and relation/dependency for the purpose of RAM.

2.1.3 RAM: process

While the RAM model itself is primarily focused on the specification, placement, and work-up of requirements, there is an accompanying process focused on roles and responsibilities with regards to the handling of requirements. The process can be closely linked to the attributes described in Table 1 (for more details see Gorschek and Wohlin [13]).

Attributes 6, 7 and 8 are of traceability type, i.e., enabling certain roles to be linked to a certain requirement, ensuring that responsibilities are clearly not open to interpretation, on a requirement level rather than a document level, to avoid issues like certain requirements being neglected or even overlooked entirely.

As requirements are caught/elicited from multiple sources, everything from a person, to a market survey or a competitor analysis, can be the official ‘‘source’’ of the requirement (in the latter two the sources are the reports/documents produced). This makes the role of requirement owner important as this person is charged with the responsibility of seeing that the requirement is followed on through. A Requirement Owner is the representative of the requirement source when this role is silent, e.g., when the source is a survey or an external party like a group of customers. In some instances the requirement source and the requirement owner can be the same person (e.g., in the case of an internal engineer formulating the original requirement).

The requirements manager’s role is typically represented by the product manager charged with the responsibility of actually working with the requirement throughout its lifecycle. During the work-up of the requirement the requirements manager can utilize resources needed, e.g., asking system experts and/or domain specialists for input. The cooperation between the requirements manager, owner and source (when applicable) is especially important, as they possess, respectively, different perspectives and knowledge pertaining the requirement in question.

If a requirement is created as a part of the work-up (a work-up requirement not an original one), the attributes of requirement source/owner/manager are set to the requirements manager responsible for the work-up.

Attribute 10 (State) reflects how the requirement is handled in the product development organization and how it is set to different states reflecting the status. Figure 4

Table 1 RAM requirement's attributes [13]

Attribute	Description	Type
1. Id	Unique identification for each requirement	Mandatory (auto generated)
2. Title	Requirements title should be short and descriptive	Mandatory (manual)
3. Description	A short description of the requirement.	Mandatory (manual)
4. Benefit/rationale	Why is the requirement specified, and What is the benefit to the source and/or champion?	Mandatory (manual)
5. Risk/restriction	Lists potential risks with the requirement, e.g., technical risks, critical dependencies	Optional (manual)
6. Requirement source	This is a link to the source of the requirement. This can be a physical person, document, group, or meeting. The exactness depends on the information available	Mandatory (manual)
7. Requirement owner	A link to the person who ''owns'' the requirement and is responsible for the follow-up of the requirement. This person acts as the advocate of the requirement. This role is always held by an internal person in the product development organization	Mandatory (manual)
8. Requirements manager	An identification of the product manager responsible for the specification, placement, and work-up of the requirement	Mandatory (auto generated)
9. Relation/dependency	One or several links to other requirements on the same level of abstraction. This attribute's aim is to record important relations/interdependencies of different types between requirements. Every link can be augmented by a explanation in free-text	Optional (manual)
10. State	A requirement in RAM can have different states giving information of the status of the requirement	Mandatory (manual)
11. Reject reason	If a requirements state is set to ''Rejected Requirement'', i.e., it is deemed out of scope in relation to the product strategies, then this attribute records the rationale of the decision	Mandatory if requirement is rejected (manual)
12. Due date	This attribute's purpose is to ascertain that requirements are not forgotten, e.g., put as draft indefinitely. The manual usage of the attribute can be in case of, e.g., customer deadline.	Mandatory, auto set to 30 days if nothing else is specified
13. Version	Records version on requirements level rather than document level. Enables requirements' history to be viewed	Mandatory (auto generated)
14. Date of creation	Records the creation date of the requirement	Mandatory (auto generated)
15. Last changed	Indicates when the last change was performed	Mandatory (auto generated)
16. Original	Is the requirement an original requirement? (Y/N)	Mandatory (manual)
17. Level	What is the abstraction level placement?	Mandatory (manual)

offers an overview of the different states. Looking at states *A*, *B* and *C* is a part of the specification and work-up of the requirement and the continuous (project independent) requirements engineering, and is basically about drafting the requirement. The work done associated with the three states is specified in further detail in Table 2, where each separate step and sub-step is described.

In RAM v.1.0 (the version implemented and evaluated in this paper) the possible states a requirement can exist in are *A: draft requirement*, *B: rejected requirement*, *C: incompletely specified*, and *G: refined requirement*. A requirement can reach states *A*, *B*, and *C* during the continuous requirements engineering, i.e., the work done as a part of RAM action steps. State *G* is however reached as the requirement in question is subjected to further refinement and validation during the *dedicated* requirements engineering.

Dedicated requirements engineering is performed on a chosen subset of requirements after project initiation, and involves refinement by the development project and system test departments to assure testability and unambiguity as well as completeness of the requirements.

It should be noted that state *B* is dependent on the requirement being out of scope, i.e., that it is not in line with the product strategies. Generally the out of scope requirement is rejected off hand (and attribute *11: reject reason* is specified), but in some instances an alternate decision can be taken. If a requirement is considered out of scope but is important for any reason (e.g., an important customer is the source) an exception can be made. However, this exception is an explicit action and has to be approved by both the requirements manager and the requirement owner, as well as checked with upper management. The general rule is that all requirements not

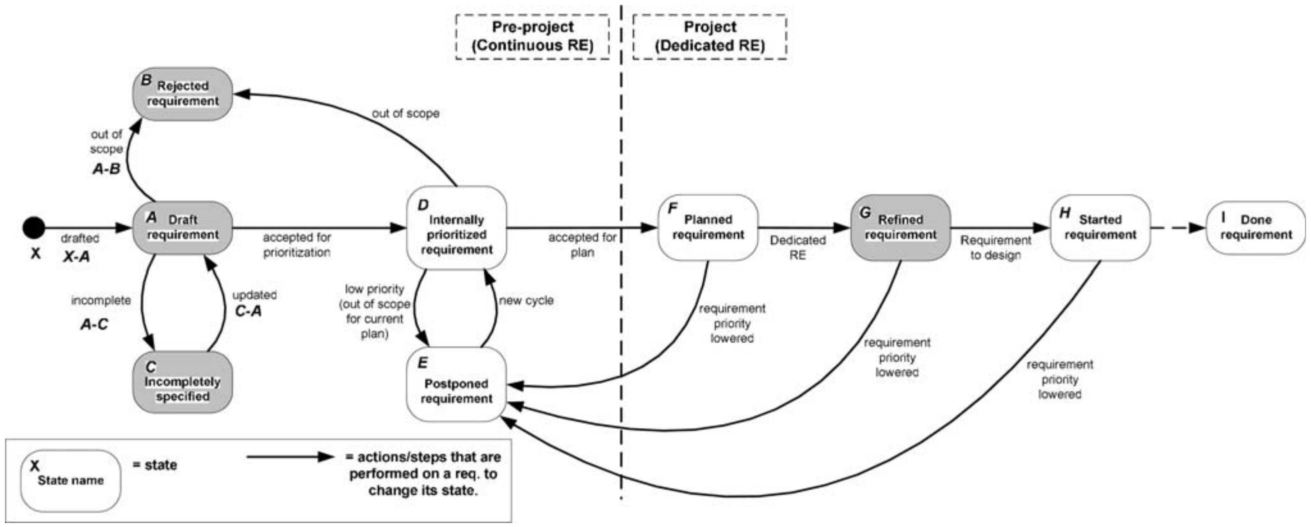


Fig. 4 Requirement states in RAM (example) [13]

Table 2 State steps in RAM [13]

Step ID	Step description	Sub-steps
X-A	The requirement (req.) is caught/elicited and drafted by the requirements manager. The work is done by the Requirements Manager, who utilizes relevant resources if the need arises (e.g., requirement owner, experts and so on are important parties in this process)	X-A-1: Specify attribute 1 to 4 (Specify) X-A-2: Determine on which level the original requirement is on (Place) X-A-3: Specify all attributes on relevant level X-A-4: Abstract and/or breakdown the original requirement according to work-up rule R1 and R2 (work-up) X-A-5: Validate requirement against requirement owner and requirement source
A-B	During (or rather, as the requirement is abstracted) work-up the requirement is checked against product strategy. Requirements deemed as not in line with strategies are deemed out of scope and thus rejected As a requirement is rejected, directly related requirements (work-up requirements) on adjacent abstraction levels either are removed as well or re-linked (enforcing R1 and R2)	A-B-1: Compare requirement (directly or indirectly) to product strategies. If the req. is not in line with strategy it is rejected A-B-2: If a requirement is rejected for any reason created work-up req. have to be evaluated if they should also be removed. If all are removed, nothing further is needed. However if a work-up req. is left on any level it has to be re-linked to requirements above and/or beyond for the work-up rules R1 and R2 to be enforced
A-C	During validation against the requirement source/ owner (X-A-5) the req. can be deemed incomplete. This can be a result of, e.g., incorrect initial placement, unsatisfactory work-up and so on	

rejected should be in line with the strategies formulated for a product. Exceptions to this rule should be kept at a minimum in order to use the full potential of RAM.

2.1.4 RAM summary

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 (triage) in the process, freeing resources to work on and refine relevant requirements that are in line with the product strategies, minimizing the risk of overloading the organization with irrelevant requirements [9].

(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 estimations, risk analysisetc. are based on requirements of appropriate abstraction level and contents. In addition, projects get good-enough requirements to base their development efforts on (e.g., testable and unambiguous [18]).

(III) Work-up of a requirement means that additional requirements may have to be created to get a connection to the top level. For example, if an incoming new requirement is placed on 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 certain level good enough to serve as a basis for project initiation (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 as detailed requirements are connected upwards through the levels.

(V) Requirements abstraction model is designed to be tool and notation independent. Any tool supporting the specification of attributes on a requirements level can be used for the implementation of RAM. In addition, as RAM is tailorable barring a minimum of information, the notation used can be adapted to fit an organization. For example, requirements can be accompanied by use cases of even formal parts if needed.

For reasons of brevity details not central for the tailoring and evaluations presented in this paper have been left out. For details, please see Gorschek and Wohlin [13].

3 The companies

The industry trials were conducted at two different companies, DanaherMotion Särö AB and ABB. Both companies are participating in a joint long-term (6-year) research project with the Blekinge Institute of Technology in the area of the process improvement and requirements engineering. The collaboration in requirements engineering started in the late 2002 with DHR, and ABB joined in the late 2003.

Each company is described briefly to get an idea of the organizations and the domains in which they operate.

3.1 DanaherMotion Särö AB (DHR)

DHR develops and sells software and hardware equipment for navigation, control, fleet management and service for automated guided vehicle (AGV) systems. More than 50 AGV system suppliers worldwide are using DHR technologies and expertise together with their own products in effective transport and logistic solutions to various markets worldwide. The headquarters and R&D Centre is located in Särö, south of Gothenburg, Sweden. DHR has 85 employees.

DHR has a wide product portfolio, as the ability to offer partners and customers a wide selection of general variants of hardware and supporting software that is regarded as important. Product managers oversee development and new releases of products.

Development projects range from 6 to 9 calender months , with a budget of 2,000–5,000 person–hours.

3.1.1 Requirements engineering practices prior to RAM implementation

DHR is certified according to SS-EN ISO 9001:1994 (currently working on certification according to ISO 9001:2000), but there have not been any attempts towards CMM or CMMI certification. Although during process assessment conducted at DHR it was summarized that there was a well established and successful requirements engineering process present albeit certain possibilities for the improvement were identified [11].

One of the main issues was the abstraction level and contents of requirements. The implication being that there was a need to handle requirements on different abstraction levels and enable improved traceability to goals and long term considerations. Large amounts of requirements, the need for enabling triage and traceability were also identified as improvements, as well as improving the division of responsibilities with regards to the RE process.

The tool used for requirements engineering was primarily MS Word and Excel. There was a standardized template in place for requirements specifications using attributes (title, description, source, relation). The specification was structured according to features, i.e., all requirements ‘‘belonging to’’ a certain feature was structured under a certain feature heading. Overall the specification used was not totally unlike the IEEE Standard 830–1998 template. In addition, the process itself was structured, documented, and following the project organization and planning.

3.2 ABB (ABB)

ABB is a leader in power and automation technologies that enable utility and industry customers to improve performance while lowering environmental impact. The ABB Group of companies operates in around 100 countries and employs about 102,000 people. The transfer of new methods for requirement engineering was performed with one of the ABB development centers in Sweden. The product development part of this organization has 200 employees, including development and product management. The organization is primarily working with product development, production and service, supporting ABB sales organizations as well as partners developing solutions for industrial use.

The introduction of RAM was made on the organization developing the controller part of the product offering. The controller includes electronics and software, and project typically involves development of functions in both hardware and software. Projects are divided into release projects with a typical duration of 9 months comprising 20–40 person–years, and functional development projects with duration from 3 to 24 months with a large diversity in effort.

Product management has the responsibility for the functionality of the controller, and orders development from the development organization. Over the past 5 years, the product development organization, including product management, has been re-organized several times. This indicates that there is a willingness and need to find improved working methods, also for the requirements engineering.

Process improvement is initiated and managed by process owners that are part of the development organization. Plans and progress are regularly monitored by senior management, and the interest in improving requirement engineering is steadily increasing.

3.2.1 Requirements engineering practices prior to RAM implementation

ABB is certified according to SS-EN ISO 9001:2000 standards. Although there was no formal certification according to e.g., CMM, ABB had a well established and successful product management and requirements engineering process in place prior to the implementation of RAM. The utilized a macro-enhanced MS Excel tool for their requirements handling utilizing attributes (title, description, state, relations). In addition to this prior to the specification of the requirements in excel they utilized an abstraction structure in the form of having different types of specifications in different stages. Early a MRS (market requirements specification) was created, and based on this one or several PRS (product requirements specifications) were created as a refinement and break-down of the MRS. This approach was rather advanced but had some inherent possibilities for improvement. The top–down structure did not always reflect the reality that requirements from the start come in to the organization on different levels of abstraction; thus the MRS to PRS structure either had to flatten all requirements to a certain level, or allow a mix of abstraction levels within the documents. In addition, traceability between individual requirements within and between the documents was not always explicit or easily attained.

4 Model tailoring

In the initial stages of RAM development, it was soon realized that one-size-does-*not*-fit-all. RAM as presented in Sect. 2.1 is generic in nature and is not intended to act as a prescriptive model with a set of best practices appropriate for all organizations and products. The model is intended to be a framework of principles on which continuous requirements engineering can be based. Several things need to be addressed prior to the model being set into operation in an organization. This can be seen as a tailoring of RAM to fit a specific product (organization), giving the adopting organization's members a chance to clarify critical issues and practices as well as to decide how RAM can be used to best suit their organization and products.

The tailoring of RAM was conducted in workshop format. During the workshop, summarized in Fig. 5, a selection of requirement engineers/product managers (Doers), engineers (Users) and managers collaborate in model tailoring and process definitions. These representatives are selected based on their roles and expertise by the local moderator/domain expert (top left in Fig. 5). As they are invited, they are asked to prepare for the workshop by reading some initial introductory materials, e.g., introduction to generic RAM, and bring artifacts such as domain specific requirements (Fig. 5, bottom left).

The main tool used in the workshop is real requirements from the specific domain. The requirements are used as examples driving the work conducted during the workshop. For example, the definition of attributes and abstraction levels are based on discussions generated as actual requirements are specified during the workshop. The workshop activities can be described as a combination of brainstorming session, requirements engineering/specification session, and process formulation.

All ideas and thoughts are written down and scrutinized until a least common denominator agreement is reached.

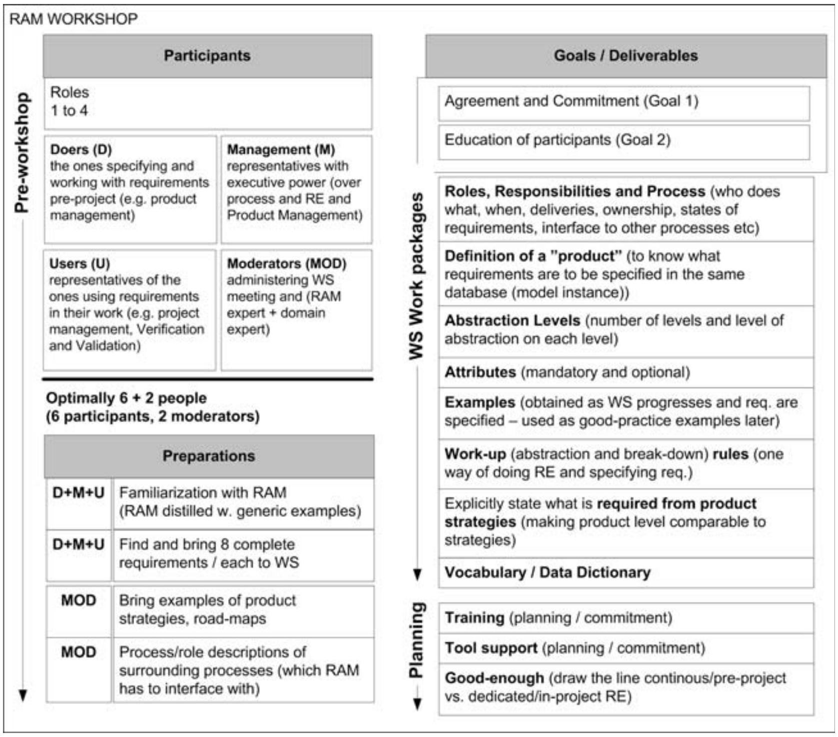

RAM WORKSHOP
Participants
Roles
1 to 4
Doers (D)
the ones specifying and working with requirements pre-project (e.g. product management)
Management (M)
representatives with executive power (over process and RE and Product Management)
Users (U)
representatives of the ones using requirements in their work (e.g. project management, Verification and Validation)
Moderators (MOD)
administering WS meeting and (RAM expert + domain expert)
Optimally 6 + 2 people
(6 participants, 2 moderators)
Preparations
D+M+U
Familiarization with RAM (RAM distilled w. generic examples)
D+M+U
Find and bring 8 complete requirements / each to WS
MOD
Bring examples of product strategies, road-maps
MOD
Process/role descriptions of surrounding processes (which RAM has to interface with)
Pre-workshop
Goals / Deliverables
Agreement and Commitment (Goal 1)
Education of participants (Goal 2)
WS Work packages
Roles, Responsibilities and Process (who does what, when, deliveries, ownership, states of requirements, interface to other processes etc)
Definition of a "product" (to know what requirements are to be specified in the same database (model instance))
Abstraction Levels (number of levels and level of abstraction on each level)
Attributes (mandatory and optional)
Examples (obtained as WS progresses and req. are specified – used as good-practice examples later)
Work-up (abstraction and break-down) rules (one way of doing RE and specifying req.)
Explicitly state what is required from product strategies (making product level comparable to strategies)
Vocabulary / Data Dictionary
Planning
Training (planning / commitment)
Tool support (planning / commitment)
Good-enough (draw the line continous/pre-project vs. dedicated/in-project RE)

Fig. 5 RAM workshop overview

The least common denominator speaks to pragmatics, e.g., if 20 attributes are suggested the number has to be scaled down to the most important ones (set as *mandatory* attributes) and additional ones that can be specified if need be (*optional* attributes), while some are dismissed altogether. In a similar fashion, all decisions reached at the workshop are compromises between what different groups need/want versus what is considered good enough for pre-project requirements engineering and what is practical in day-to-day work. The moderators direct the process, i.e., as experts in both the domain and the model they can assure that no critical issues are overlooked.

As the workshop is concluded, the main goal is to achieve agreement regarding the Deliverables and the initial commitment to training and tools support (see Fig. 5, right). Post-workshop the moderators summarize the results, formalize the model with descriptions, the process, and develop brief reference guides. These guides are lightweight manuals and include domain-specific examples. Tool acquisitions/adaptations are also important as well as the planning of training sessions.

One of the main overall goals of the workshop is to obtain commitment from all groups, especially management, as it is crucial for the successful process improvement [19–24].

4.1 Implementation

It is important to realize that the tailoring resulted in two different instantiations of the RAM model. Each of the companies had their own needs and preferences that dictated the tailoring but also the implementation of the tailored model. Looking at the model instantiations themselves many of the details, e.g., exactly what attributes are used; their descriptions, and details regarding process and roles and responsibilities are considered proprietary in nature. This implies that these details cannot be shared in this paper. The implementation of the tailored RAM instances at the two organizations also differed, mainly due to two factors, size of the organizations and time. DHR has a smaller organization than ABB making it possible to implement RAM in one increment. Further, DHR only has one product management organization (for AGVs). ABB on the other hand (as a larger organization) opted to do a stepwise implementation starting with one product management group dealing with one set of products. The implementations turned out to be rather similar despite the organizational differences, i.e., one product management group in each organization.

The time aspect revolved around that DHR had been involved in the development of RAM from the beginning (1 year longer than ABB), resulting in that the model had a somewhat longer time to maturate at DHR. In addition, the generic model was largely already formulated when ABB entered the stage, although this was not considered a problem due to RAM's tailorability.

A generic overview of the model as implemented at the companies can be seen in Table 3.

As mentioned, the technology transfer (process improvement effort) had continued longer at DHR than at

Table 3 Implementation overview DHR and ABB

Model/process aspect	Company	
	DHR (RAM instance called ‘‘DHRRAM’’)	ABB (RAM instance called ‘‘ABBR-RAM’’)
Abstraction levels	4 levels (full work-up as in generic RAM)	2–3 levels (with work-up in the form of headers. Headers imply that the hierarchy exists but full requirements are not specified on multiple levels but only on one level and headers created above. A header is similar to a requirements title on upper level)
Attributes	9 mandatory (3 auto generated)	10 mandatory (2 auto generated)
	6 conditional	1 conditional
	5 optional	3 optional
Possible requirement states	7 requirement states	5 requirement states + release status
Tool	CaliberRM	Excel (macrodevelopment adapted)
Main users of RAM (specification/analysis)	Process/model/RE expert, product management, project management	Process/model/RE expert, product management
Roles involved in pre-project RE (using requirements from RAM)	Product management, project management Requirements owner, technical expert	Process/model/RE expert, product management, requirements owner, reviewer (development, architecture), reviewer (verification and validation), technical expert, requirements receiver (project)

ABB. DHR used full work-up of all requirements, in approximately the same manner as described by the generic RAM model described in Sect. 2.1 (with the exception of having tailored abstraction of the levels) at the time of the evaluation. This was preceded by making sure that tools support was available. ABB on the other hand was at the time of the evaluation not implementing full work-up as they chose to do a stepwise implementation of RAM, starting with the specification of all requirements on *one* abstraction level using RAM attributes, then using feature level and above in a limited fashion. This involved creating headers/titles (not complete work-up requirements) under which requirements could be sorted. The implementation plan at ABB said to wait with the full work-up implementation until the product management organization got used to specifying requirements as objects (with attributes), but also improved tool-support was considered an advantage prior to implementing full work-up.

It is important to note that the companies have implemented different instantiations of RAM, and to a varying degree. In addition, the initial (pre-RAM) challenges facing the product management and requirements engineering organizations at the companies differed. Although, the common challenge of working in a product centered market-driven environment united them, the immediate improvement potentials were not the same. The focus of this paper and the evaluations presented is not a comparison between the companies. Rather the evaluation of the process improvements conducted, i.e., the implementation of RAM.

5 Evaluation design

The first step in the RAM evaluation was to make a selection of interview subjects for participation in the evaluations. The general idea was to select a representative sample of the roles involved in both using RAM to specify and analyze requirements, but also elicit information from the engineers involved in the pre-project requirements engineering. In addition, the receivers of the ‘‘RAM requirements’’ were considered an important source of information as they had a unique opportunity to assess e.g., requirements quality aspects. Table 4 lists the roles represented in the evaluation, as well as their organizational affiliation within the company. The ‘‘code’’ column refers the results presented in Sect. 6.

The sample chosen (the actual people representing each role) was based on seniority, experience and time spent using RAM. In some cases all people that had been using RAM in their work were interviewed. This was the case for product managers and development project managers. With regard to the selection of developers and verification and validation the selection was done in collaboration with on-site expertise where senior personnel were premiered. Although the titles of the roles are similar in the companies, the actual role content and responsibilities varied. The implication, with regard to the evaluation presented in this paper, being that a certain role may have been involved in performing different actions in the companies. This is discussed when relevant in the presentation of the evaluation results in Sect. 6. Each role is represented by one physical person.

Table 4 Roles participating in the evaluation

Role	Organization	Company	Code
Requirements engineer/specification	Product management	DHR + ABB	Subject RE
Product manager	Product management	DHR + ABB	Subject PM
Developer/systems architect	Project/development	DHR + ABB	Subject DEV
Verification and validation (system test)	Project/development	DHR + ABB	Subject V&V
Development project manager	Project/development	ABB only	Subject PL

The evaluations were conducted in interview form about 6 months after RAM was introduced at the companies. The reason for the time delay was that we wanted one project to be completed prior to the evaluation. This was the only way in which the subjects could render an informed expert opinion as to what positive (or negative for that matter) effects the use of RAM had. For example, rendering an opinion as to whether the requirements produced using RAM where better to base estimations on could not be done until the estimates were put to the test in the project. As preparation, the subjects were asked to prepare pertinent information and bring relevant documentation to the interview, e.g., test plans, requirements specifications and so on. This gave the subjects the not only the opportunity to get in the right ‘‘mindset’’ for the interview, but also the documents that were used by the subjects to motivate their answers.

The evaluation itself was divided into two main parts; each detailed below.

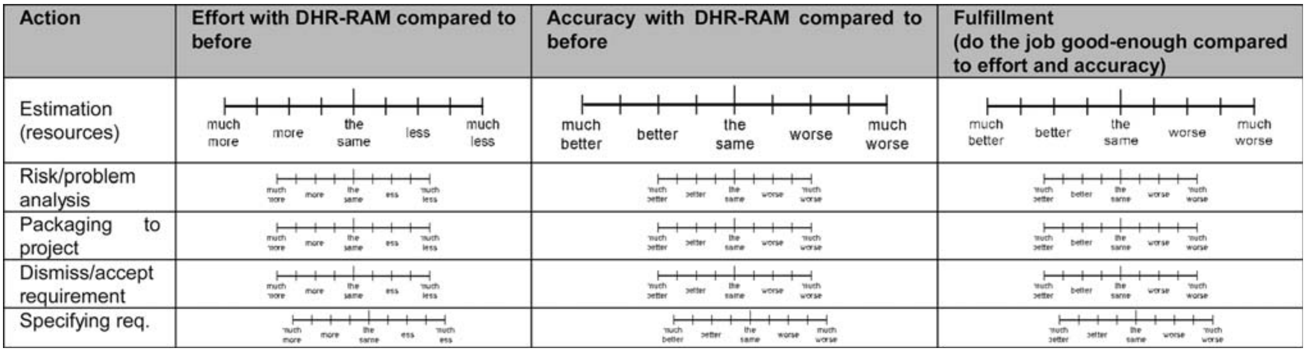
5.1 Part I

The first part focuses on RAM itself and the product management/requirements engineering work conducted using the model. It studies the activities, called ‘‘Actions’’ in the study, performed in the day-to-day work by primarily the product management organization. The actions evaluated are displayed in Table 5 in the same manner as they were presented to the subjects (Table 5 does not contain any answers, rather just illustrates the scales used—the results are presented in Sect. 6).

The subjects were asked to grade each action from three perspectives. First, the *effort* needed to perform a certain Action, for example how much effort did it take to perform ‘‘Estimations’’ using RAM and RAM requirements. Second, the *accuracy* of each action was elicited.

The third perspective speaks to the *fulfillment* of each Action when taking both effort and accuracy into account. The idea behind fulfillment is to catch implicit problems and offer a ‘‘reality-check’’. For example, if an Action takes ‘‘more’’ effort, and the accuracy is ‘‘better’’ it is not possible to ascertain if the improvement in accuracy is due to simply putting more effort into the action or if RAM has influenced the outcome. Fulfillment was interpreted as ‘‘bang for the buck’’ or return on investment by the participants. Fulfillment also gauges positive (or negative) spin-off effects in using RAM that are not covered by the concept of accuracy. For example, specification of requirements using RAM may require more effort and improve accuracy due to, e.g., structure, but if using RAM is cumbersome and usability is an issue fulfillment will be low. The opposite is also possible, e.g., the process with RAM encourages requirements engineering meetings, which in itself can lead to catching, e.g., dependencies between requirements thus avoiding overlap. In this case, the fulfillment might be higher that indicated by just

Table 5 Evaluation Part I, actions performed using RAM



Action	Effort with DHR-RAM compared to before	Accuracy with DHR-RAM compared to before	Fulfillment (do the job good-enough compared to effort and accuracy)
Estimation (resources)	much more, more, the same, less, much less	much better, better, the same, worse, much worse	much better, better, the same, worse, much worse
Risk/problem analysis	much more, more, the same, less, much less	much better, better, the same, worse, much worse	much better, better, the same, worse, much worse
Packaging to project	much more, more, the same, less, much less	much better, better, the same, worse, much worse	much better, better, the same, worse, much worse
Dismiss/accept requirement	much more, more, the same, less, much less	much better, better, the same, worse, much worse	much better, better, the same, worse, much worse
Specifying req.	much more, more, the same, less, much less	much better, better, the same, worse, much worse	much better, better, the same, worse, much worse

compiling the results of effort and accuracy. That is, fulfillment is a way in which we try to assess the total value, i.e., if the effort was worth the attained level of accuracy. Even if fulfillment is used as a way to catch the overall effect of the improvement on a certain action, all values with regards to effort, accuracy and fulfillment are presented separately as the summation is not formally correct; rather illustrative.

The scale used spans from ‘‘much more’’ to ‘‘much less’’ regarding effort, and ‘‘much better’’ to ‘‘much worse’’ regarding accuracy and fulfillment, as can be seen in Table 5 (the scales used are identical for every column). In each case, the subject has the opportunity to choose one neutral, four negative, and four positive alternatives. The comparisons performed are based on how product management/requirements engineering actions were performed prior to the implementation of RAM in the organization.

In addition to grading each Action, every subject was asked to explain his/her answers by commenting on their reasoning, effectively motivating their answers and assuring to the evaluators that the questions and action were understood as intended. This assured that the different subjects had the same understanding of each Action.

5.2 Part II

The second part focuses on the quality of the products of RAM, i.e., the requirements themselves. The subjects are asked to compare the requirements generated using RAM with requirements used prior to RAM implementation. Table 6 lists the requirements quality attributes evaluated using a similar scale as in Part I (all scales in Table 6 are identical and only miniaturized to save space). The choice of quality attributes evaluated comes from both academia [18, 25], but also from what was considered important during process assessments performed at the companies in question [11, 12]. The subjects were also asked to motivate their answers in the same way as in Part I.

5.3 Analysis Part I and Part II

For the purpose of comparison and analysis descriptive statistics are used. The answers obtained on the scales described under Parts I and II are transformed into numerical values as can be seen in Fig. 6. The less effort an action requires in Part I the lower the value (lower is better). Accuracy and fulfillment are treated the opposite way, i.e., better and much better are translated into positive values (higher is better).

Table 6 Evaluation Part II, requirements quality using RAM

Requirement Quality Attribute	
Completeness	much better · better · the same · worse · much worse
Un- ambiguity	much better · better · the same · worse · much worse
Testability	much better · better · the same · worse · much worse
Traceability	much better · better · the same · worse · much worse
Understandability / readability (why, context, big picture)	much better · better · the same · worse · much worse
Consistency	much better · better · the same · worse · much worse
Catching Dependencies Relationships between req.	much better · better · the same · worse · much worse
Redundancy (double information, unnecessary information)	much better · better · the same · worse · much worse
Structure (organization) find information etc	much better · better · the same · worse · much worse
Analysis base (can you base analysis/design on them)	much better · better · the same · worse · much worse
Conformance to business goals	much better · better · the same · worse · much worse

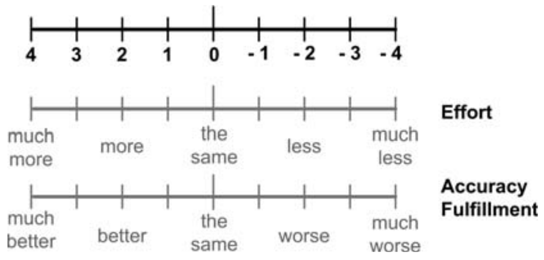


Fig. 6 Conversion of Parts I and II answers to numerical values

Table 7 Example of results Part I for actions

Action *X*				
	Effort	Accuracy	Fulfillment	Max: 12, min: –12
Subject *x*	0	2	3	5
Subject *y*	N/A	N/A	N/A	N/A
Subject *z*	1	2	3	4
Subject *n*	–2	1	4	7

The results are presented in the form of descriptive statistics (diagrams) and tables with numerical values created as illustrated in Fig. 6. An example is shown in Table 7. Effort, accuracy and fulfillment are specified by a subject and converted to numerical values as described previously. The last column (max: 12, min: –12) is a compilation of the values calculated with the following formula: accuracy – effort + fulfillment, giving a maximum of 12, and a minimum of –12 per subject with regard to a certain action. Using the example in Table 7 subject *x* would get a max/min of: 2 – 0 + 3 = 5, and in the case of subject n: 1 – (–2) + 4 = 7 (less being better in the case of effort to perform an action). The max/min compilation is a way to summarize the total level of perceived benefit concerning a certain action, allowing for easier comparison in diagram form as will be seen in Sect. 6. The use of numerical values converted from scales for performing operations such as summation can be questioned, as it is not normal in measurement theory. In the case of max/min the use is strictly illustrative, i.e., enabling concise presentation in diagram form collecting several aspects in one figure (as can be seen in Sect. 6). Care has been taken to present the actual values in parallel and the analysis of the actions are based on effort, accuracy and fulfillment, not the calculated max/min values.

In addition, not every subject had the possibility to answer every question or make estimates in all cases. This is due to what Actions the subjects were involved with. For example, the role of verification and validation (system test) was not involved in the Action of performing estimations on requirements during the pre-project requirements engineering. In this case, the answers will be grey and marked ‘‘N/A’’ as can be seen for subject y in Table 7.

5.4 Validity evaluation

In this section, we discuss the threats to the validity. We base this on the discussion of validity and threats to research projects presented in Wohlin et al. [26]. The validity threats considered are conclusion, internal, and external validity threats, respectively.

5.4.1 Conclusion validity

Each interview was done in one uninterrupted work session. Thus, the answers were not influenced by internal discussions about the questions during, e.g., coffee breaks.

The sampling techniques used for the static validation can pose a threat to the validity of the investigation. The subjects selected may not be totally representative of the role they should represent at the companies. The main assurance that this misrepresentation is minimal is the fact that the subjects were selected in cooperation with several senior managers with extensive knowledge and experience concerning the development processes and the personnel at the companies. In many cases the selection was based on what individuals had been using RAM at the companies. For example, all the product managers using RAM were selected to be a part of the evaluation. In this case no other individuals could have been selected.

5.4.2 Internal validity

As the discussions and evaluations of the RAM was performed with the different interview subjects, they were called upon to voice their opinions and views regarding changes to the requirements engineering and product management practices with regard to the implementation of RAM. As their answers were registered by the researcher this could have constrained people in their answers. This potential problem was alleviated by the guarantee of anonymity as to all information divulged during the validation, and that recorded answers was only to be used by the researcher, i.e., not to be showed or used by any other party. In addition, the researcher has worked with ABB and DHR personnel over a period of about three years, and has earned a degree of trust, as well as established a professional relationship that in all likelihood made it easier for the personnel to voice their views.

On the other hand, this raises another potential issue, namely, that the interview subjects could potentially allow their personal feelings towards the researchers to influence the answers. We believe that three aspects alleviate this threat. Fist, there was no personal relationship between the interview subjects and the researchers other than a professional one. RAM itself was not implemented and championed in industry by the researchers, rather by local industry professionals acting as champions.

Second, as the research and technology transfer conducted in the research cooperation with the companies has produced concrete changes and effects, all professionals involved have a vested interest in proper evaluation with focus on further improving all aspects of the development process in general, and RAM in particular. Being less than honest would yield problems as the evaluation is a basis for further improvements. The research collaboration between the companies and the researchers is based on iteratively transferring knowledge and technology from academia to industry, and as a part of this criticism and constructive feedback are well established and utilized means for process improvement evolution. Third, as a part of the interviews the subjects were asked to motivate their answers to assure that they did not misunderstand a question. This also gave the researchers the possibility to assure that the answers were well grounded in events encountered during the development process.

5.4.3 External validity

The external validity is concerned with the ability to generalize the results, i.e., in this case the applicability of RAM in industry at companies other than DHR and ABB, as some of the problems introduced as a motivation behind the conception of RAM (mentioned in the introduction of this paper and further detailed in the process evaluations at each company) to some extent could be general for organization faced with developing products for a market. As these organizations are faced with the same issues (e.g., large quantities of requirements on multiple levels of abstraction) it stands to reason that RAM could be usable and useful in more organizations.

However, strictly speaking, it is not possible to generalize the results from this evaluation based on the two cases of ABB and DHR; although from a perspective of transferability the process assessments presented in [11–13] can give an overview of the challenges facing the companies where RAM has been implemented. The tailoring of RAM as described in this paper (see Sect. 4) and in Gorschek and Wohlin [13], and in Gorschek et al. [15], makes it possible for any organization to adapt the ideas behind RAM to fit their organization. The evaluation method used in this paper can then subsequently be used to evaluate the process improvement.

6 Evaluation results

The evaluation results are presented by company, and divided according to Parts I and II.

6.1 DHR

The evaluation at DHR was performed in one session with individual interviews of about 1–1.5 h each. Subjects representing RE, PM, and DEV were all involved with aspects of the pre-project RE and the activities performed in relation to product management. The subject representing V&V was only involved in verification and validation activities (creation of test plan and test cases) and only answered questions in Part II.

6.1.1 Part I

In total five actions were evaluated (see Table 5) concerning the effort it took to accomplish the work, accuracy of the performed work, and the fulfillment achieved (gauging return on investment taking amongst other things effort and accuracy into account). The individual results are presented in Table 8.

Table 8 Results Part I for DHR Actions

	Effort	Accuracy	Fulfillment	Max: 12, min: –12
Estimation				
Subject RE	0	2	2	4
Subject PM	N/A	N/A	N/A	N/A
Subject DEV	1	3	2	4
Subject V and V	N/A	N/A	N/A	N/A
Risk/problem analysis				
Subject RE	0	1	1	2
Subject PM	0	1	1	2
Subject DEV	2	4	3	5
Subject V&V	N/A	N/A	N/A	N/A
Packaging to project				
Subject RE	-2	2	2	6
Subject PM	N/A	N/A	N/A	N/A
Subject DEV	3	3	2	2
Subject V&V	N/A	N/A	N/A	N/A
Dismiss/accept req.				
Subject RE	–2	2	2	6
Subject PM	–2	1	3	6
Subject DEV	–2	4	3	9
Subject V&V	N/A	N/A	N/A	N/A
Specifying req.				
Subject RE	2	2	2	2
Subject PM	0	2	2	4
Subject DEV	3	2	3	2
Subject V&V	N/A	N/A	N/A	N/A

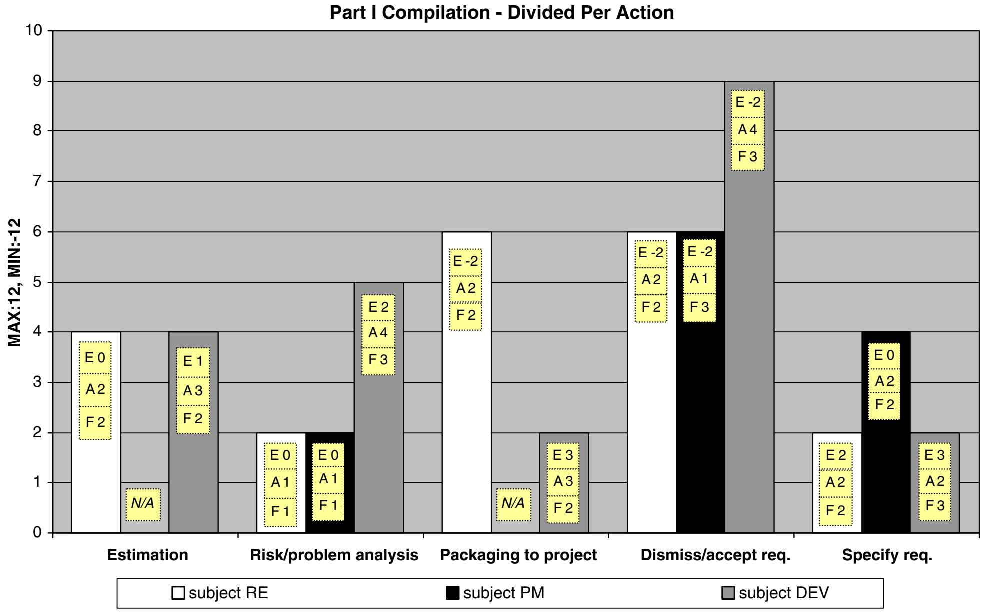


Fig. 7 Action compilation divided by action and subject

A general tendency observed is that although some effort increases are present it is predominantly compensated by increased accuracy and the fulfillment is in all cases positive.

A compact overview of the results is offered in Fig. 7. The bars represent the max/min compilation per subject (see Table 4) and Action. The tanned notes in the bars are the individual results of each subject, ''E'' standing for effort, ''A'' for accuracy, and ''F'' for fulfillment (for example, for the action of estimation subject RE has an effort of zero, accuracy of two and fulfillment of two, and a max/min of four indicated by the bar itself). In the cases where no answer was given (due to a subject not performing a certain action) an ''N/A'' note can be seen, this is to not confuse the absence of a bar with a value of zero (for example, subject PM did not perform the Action of Estimation). Observe that the *y*-axis scale in reality goes from –12 to 12, although only 0–10 is displayed in the diagram, as all the values are in that range.

Each action is analyzed in further detail below.

- *Estimation* in the case of DHR is performed by project personnel (DEV) and to some extent by the requirements engineer. Representatives for these roles claim improved accuracy and fulfillment. The main motivation is that the hierarchical structure of abstraction gives a better overview, in turn offering more information in total. Being able to traverse the levels gives a big-picture, while *all* requirements are broken down to a level where they are detailed enough to base estimations on.
- *Risk/problem analysis* is not performed as a separate activity; rather it is an implicit part of the analysis and specification of requirements (certain attributes specified for each requirements explicitly demands that limitations and risks be a part of the specification). RE and PM felt that there was no real increase in effort as the explicit risk and problem analysis was a part of the specification (the effort increase was put on the Action of specification). DEV felt that the breakdown of requirements, especially from feature to function level, gave early analysis on a more detailed level (as attributes, e.g., concerning risk/limitation had to be specified on function level and not just for abstract features). Although DEV felt strongly that accuracy was greatly increased, there was also an increase in work effort (E2). The total fulfillment was even greater than just a compilation of effort and accuracy due to that collaboration in RE meetings using the structured RAM requirements made it possible to catch problems not directly associated with requirements being specified, but in already existing products and legacy features.
- *Packaging to project* did not involve PM in this instance (was involved in the earlier phases of pre-project RE) and from a PM perspective the requirements going to a certain project were already packaged.

There seems to be a large discrepancy between RE and DEV regarding this action. Both agree that accuracy and fulfillment is better than before, but while RE considers less effort having been spent, DEV feels a noticeable increase in effort (which is what brings down the total max/min value for DEV). The increase in effort can be explained by two factors, both given by the subject as motivation for the answers. First the initiation threshold of using RAM (learning curve), and second a new practice of creating Implementation Proposals[1] prior to project start. The learning curve effect was felt especially by DEV as they were not used to being that active in requirements engineering. The practice of creating Implementation Proposals had nothing to do with RAM as such, but the activities were to some extent performed in parallel making the total effort pre-project greater.

- *Dismiss/accept requirements* was effort-wise felt to be substantially less by all subjects. The main motivation was that the abstraction levels offered much better overview of all requirements, enabling discussions focusing on abstract levels where the amount of requirements were manageable, but with the possibility to look at more detailed levels (following dependencies and relationships downward) when needed to see the implications of the decisions.

RE and PM mostly agreed with regard to increase in accuracy and a positive fulfillment. DEV felt a substantially greater increase in accuracy mainly due to the traceability of consequence, i.e., dismissing a feature level requirement was not performed using a gut feeling when gauging consequences, but rather explicit relationships could be followed down the abstraction levels giving a true picture of the decision being made.

- *Specify requirement* shows an effort increase for RE and DEV that can be attributed to two factors, learning curve in the case of DEV, but also the fact that more information is specified using RAM, and the use of abstraction levels, following the RAM process demands more effort. However, it should be observed that the effort increase was considered moderate in nature.

PM agreed with RE and DEV (they have the same level of accuracy and fulfillment increase) but did not feel a great increase in specification effort. This can be attributed to that the subject consciously subtracted the learning threshold when giving an answer to this question, i.e., looking at the comments/motivation offered when answering the learning threshold would have added moderately to the effort just like in the case of RE and DEV.

6.1.2 Part II

The subjects were asked to evaluate the requirements quality from 11 different perspectives. In this part, the views of a new subject is introduced, i.e., V&V. V&V did not participate in the requirements engineering activities,

Table 9 Results Part II for DHR requirement quality attributes

	Completeness	Unambiguity	Testability	Traceability	Understandability/ readability	Consistency	Catching dep. relationships btw req.	Redundancy	Structure	Analysis base	Conformance to business goals	SUM per subject (max 44, min –44)
Requirement quality												
Subject RE	2	3	2	2	2	2	3	–1	2	2	0	19
Subject PM	2	2	0	4	4	2	0	0	0	2	1	18
Subject DEV	2	0	3	4	2	2	4	2	4	2	N/A	25
Subject V&V	2	3	3	4	2	3	0	4	3	2	N/A	26
Sum per quality attribute (max: 16, min: –16)	8	8	8	14	10	9	7	5	9	8	1	

[1] An implementation proposal is when the developers take one or several requirements and as a part of their analysis state example solutions taking, e.g., system architecture into account.

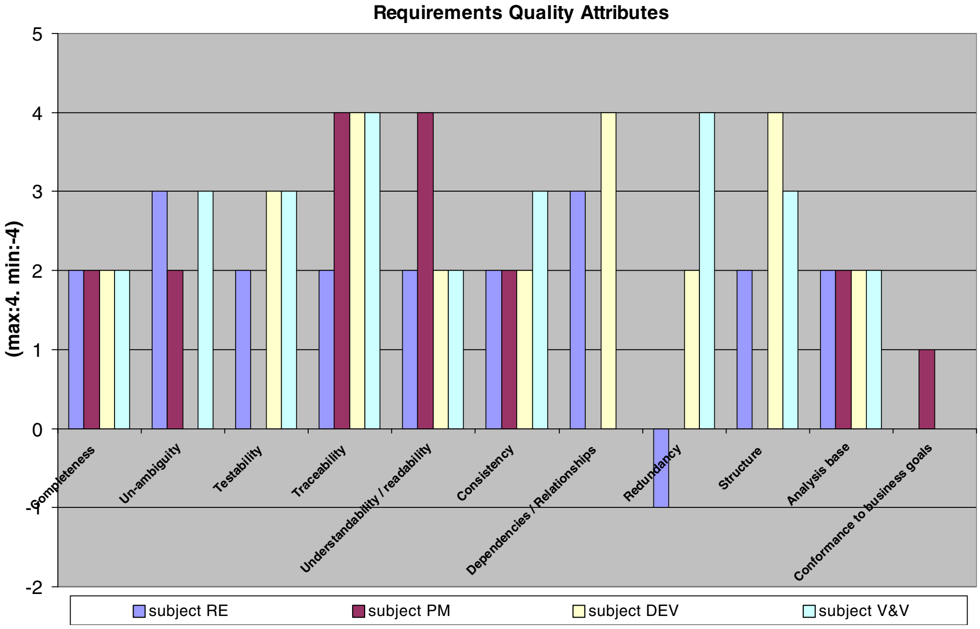


Fig. 8 Overview of requirements quality attributes

but is an expert in the artifacts produced, namely the requirements that are used to create test cases.

All answers for Part II are displayed in Table 9. The rows are summarized giving a total quality attribute value for the RAM requirements per subject (this is of course only to give an illustration, just like in the case of max/min in Part I, and not to be interpreted as absolute values). All positive values in the table indicate an increase in quality, where a four is maximum, and vice versa for a decrease in quality with a value of negative four in the worst case.

It is notable that the highest scores are obtained from DEV and V&V, users of the requirements within their development work. The high scores were obtained despite of the fact that they could not answer whether the requirements conformed to business goals, as they were not well versed in that aspect of product development. The total scores of RE and PM were similar.

The columns are also summarized displaying the overall score for every quality attribute.

Figure 8 gives a different overview of the results divided by quality attribute.

- *Completeness* (total score 8 out of 16). All subjects agree that substantial increase has been achieved with regards to completeness of the requirements. This is due to the use of attributes and more information being available than an individual requirement as the abstraction levels can be traversed offering overall big-picture and details in the form of linked requirements on lower levels.
- *Unambiguity* (total score 8 out of 16). All subjects except for DEV agree that the RAM requirements are less ambiguous than the previous ones. DEV scores it the same as before (zero). The main motivation offered is that in spite of more and better information (attributes and abstraction levels) the actual natural language formulation of the requirements needs to be improved.
- *Testability* (total score 8 out of 16). A general increase, although this increase was not confirmed by PM (giving it a zero). The main motivation being that from the PM perspective no direct difference in testability can be seen. This is however strongly opposed by both RE, DEV and especially V&V, who scores it high and sees a significant improvement in testability (3).
- *Traceability* (total score 14 out of 16). All subjects except RE agree that a very significant increase in traceability has been achieved. RE also scores it as a significant improvement but remarks that several traceability aspects are still lacking, due mainly to what perspective one has. For example, traceability to source/origin for DEV and V&V is fulfilled if the internally responsible for a requirement is listed. From the perspective of RE, the real external source is more interesting and not always traceable.
- *Understandability/readability* (total score 10 out of 16). All subjects indicate a significant improvement and they agree with regard to the level, with the exception of PM indicating a very large improvement. This is motivated mainly by the explicit abstraction of requirements and the possibility to gain an overview, and a connection to the product and business level.
- *Consistency* (total score 9 out of 16). All subjects see a substantial improvement in requirement consistency,

with V&V indicating a somewhat higher consistency than the other subjects do.

- *Catching dependencies/relationships* (total score 7 out of 16). The total score is brought down by PM and V&V who grade the improvement as zero in strong contrast to RE and DEV seeing a great improvement. The explanation for this is that while RE and DEV included dependencies and relations between abstraction levels in their score, PM and V&V did not. Both PM and V&V acknowledge that a very substantial improvement in catching and specifying dependencies exists upwards and downwards between abstraction levels, no additional effort or analysis has been put in catching dependencies between requirements on the same level (nor has it become worse than before).
- *Redundancy* (total score 5 out of 16). DEV and especially V&V see substantial improvements (less redundancy) in using RAM, while PM sees neither improvement nor degradation. RE however sees some added redundancy. The explanation for the two perspectives may be that RE and PM work with requirements actively, using and especially knowing about all information specified. DEV and V&V on the other hand concentrate and use predominantly parts of the total information specified (e.g., using primarily function level when specifying test cases). Thus, they are isolated from some information they may see as redundant in nature.

It should be observed that the overall score indicates an improvement with regard to redundancy in spite of the fact that the amount of information specified in total using RAM is much greater than previously.

- *Structure* (total score 9 out of 16). The improvement of this quality attribute is scored high to very high by all subjects except PM. The main motivation behind this is can be attributed to the fact that the PM in this instance was not involved in the actual specification of many of the requirements using RAM, and when being involved the learning curve of using RAM took away from the usability. The main motivation from RE not scoring it higher was attributed to the learning curve of the new tool used for requirements engineering.
- *Analysis base* (total score 8 out of 16). All subjects agree that the use of RAM (both attributes and abstraction levels) produces requirements of significantly better quality with regards to the analysis base.
- *Conformance to business goals* (total score 1 out of 8). At first glance, this score indicates a very low improvement in requirements' conformance to business goals. First, it should be observed that the maximum possible score for this quality attribute is eight, not 16 as DEV and V&V did not score it. Still one out of a possible improvement of eight is low; especially since RAM abstraction is intended to explicitly connect requirements to product strategies (through the use of abstraction). PM does score some improvement but both RE and PM remark that there is not an explicit mapping between the most abstract requirements and the product strategies and business goals. The main reason for this is that as RAM is relatively new to the organization the formulation of business and product strategies have not been performed in the manner needed for explicit use for mapping requirements. This work was planned and underway, but in the early stages at the time of the evaluation.

6.1.3 Summary and discussion: DHR

Part I overall shows some increase in effort being expended to perform the actions, but this is more than compensated by an increase in accuracy, especially when the learning curve of both RAM (new process, concept of abstraction, attributes) and a new requirements engineering tool are taken into consideration. Improvements can be seen across the board, with the ability to have fast dismissal or acceptance of new incoming requirements at the top. Estimation activities also show a large increase in accuracy.

Part II also shows improvements across the board. Traceability scores the highest and Understandability/readability comes in second. Completeness, unambiguity, testability, consistency, analysis base, and catching dependencies/relationships all score in the mid-range showing an overall substantial increase in requirements quality. The low-scoring quality attribute Conformance to business goals can to some extent be attributed to the short operation time of RAM in the organization as the routines for creating and using explicit product strategies and business and mapping to high level requirements is under way. In addition, it is interesting to notice that in many cases, agreement is high. For example, looking at completeness, traceability, understandability/readability, consistency, and analysis base the broad spectrum of roles all feel high increase in requirements quality. This is especially evident in the case of Completeness and Analysis base where the subjects agree totally, i.e., an increase of two (out of four). This indicates that the implementation of RAM managed to give a significant increase for multiple roles and thus multiple uses of the requirements. RE and PM use requirements for figuring out *what* is to be done and for product and project planning activities. V&V has the perspective of verification and validation, while DEV uses requirements to base solutions (how) on. An agreement in quality increase could suggest that the require-

Table 10 Results part I for ABB actions

	Effort	Accuracy	Fulfillment	Max:12, min: –12
Estimation				
Subject RE	–2	2	4	8
Subject PM	1	2	4	5
Subject DEV	1	0	1	0
Subject V&V	N/A	N/A	N/A	N/A
Subject PL	N/A	N/A	N/A	N/A
Risk/problem analysis				
Subject RE	N/A	N/A	N/A	N/A
Subject PM	–2	1	3	6
Subject DEV	1	1	1	1
Subject V&V	1	4	4	7
Subject PL	N/A	N/A	N/A	N/A
Packaging to project				
Subject RE	2	2	2	2
Subject PM	0	2	2	4
Subject DEV	N/A	N/A	N/A	N/A
Subject V&V	N/A	N/A	N/A	N/A
Subject PL	N/A	N/A	N/A	N/A
Dismiss/Accept req.				
Subject RE	2	2	2	2
Subject PM	-1	1	1	3
Subject DEV	2	2	2	2
Subject V&V	N/A	N/A	N/A	N/A
Subject PL	N/A	N/A	N/A	N/A
Specifying req.				
Subject RE	4	2	4	2
Subject PM	1	3	3	5
Subject DEV	1	1	1	1
Subject V&V	–2	2	2	6
Subject PL	N/A	N/A	N/A	N/A

ments on different abstraction levels suit the needs of multiple types of users.

Overall, the use of appropriate attributes and abstraction (work-up) of requirements results in a positive outcome both with regard to actions performed as a part of pre-project requirements engineering/product management and with regard to the requirements' quality. It should be noticed that all parts of the evaluation presented are relative in nature, i.e., an increase in accuracy, positive fulfillment, and improvement in quality all indicate the relative benefit achieved using RAM instead of previous practices.

6.2 ABB

The evaluation at ABB was performed in two sessions spanning over 2 days. Each individual interview lasted between 1 and 2 h. Subjects representing RE, PM, DEV and V&V were all involved with aspects of pre-project requirements engineering and the activities performed in relation to product management. Subject PL represents the project manager who was not involved in pre-project requirements engineering, thus only answered questions in Part II.

6.2.1 *Part I*

The individual results for Part I are presented in Table 10, and a more compact overview can be seen in Fig. 9. The bars (Fig. 9) represent the min/max compilation per action and subject. The tanned notes in the bars are the individual results of each subject, ''E'' standing for effort, ''A'' for accuracy, and ''F'' for fulfillment. In the cases where no answer was given (due to a subject not performing a certain Action) an ''N/A'' note can be seen, this to not confuse the absence of a bar with a value of zero. Observe that the *y*-axis scale in reality goes from –12 to 12, although only 0–9 is displayed in the diagram, as all the values are in that range.

Each Action is analyzed in further detail below.

- *Estimation* in the case of ABB is performed by RE and PM with input from DEV and other experts as needed. Both RE and PM claim a substantial improvement in accuracy and fulfillment. In addition RE sees less effort being expended using RAM. The main motivation is that the hierarchical structure gives a better overview and that the requirements are more specified utilizing attributes. DEV sees a moderate increase in effort but none in accuracy. Despite this, DEV claims positive fulfillment as discussions in relation to estimations solve other small issues and assumptions are questioned.
- *Risk/problem analysis* did not involve RE in the case of ABB, but mainly PM, DEV and V&V were used as experts, e.g., consulting them as needed. PM sees a substantial decrease in effort needed to perform the Action, and a moderate increase in accuracy. DEV on the other hand sees a moderate increase in effort. DEV motivates this by the adoption of a structured process for the activity, which has an initial learning curve. V&V also sees a moderate increase in effort but a substantial increase in accuracy and fulfillment. This is mainly due to the use of attributes, which forces the explicit specification of information that could have been missed previously. Another contributing factor is that V&V is involved earlier in the process allowing for aspects such as verifiability to be premiered.
- *Packaging to project* involved RE and PM. Both of them see an increase in accuracy in using RAM. The only difference is that RE sees an increase in effort. The

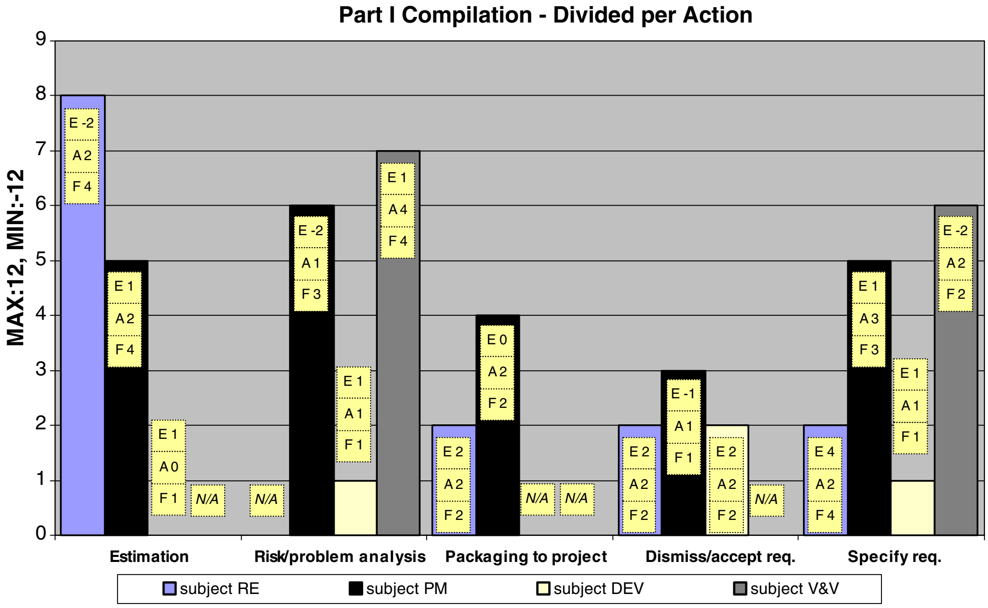


Fig. 9 Action compilation divided by action and subject

increase in effort was motivated by the subject and can be attributed to a double learning curve. The first part of the learning curve is due to that RE did not have extensive experience in requirements specification at the time. The other part of the learning curve can be attributed to the implementation of RAM itself. (These two aspects can be seen repeated in the rest of the actions for the subject.) It should be observed that although effort and accuracy are equally increased the fulfillment was relatively high, indicating that the subject realized that the issue of effort was not mainly due to problems with usability or usefulness of RAM.

- *Dismiss/accept requirements* RE and DEV both considered that a substantial increase in effort was necessary to systematically dismiss or accept requirements using RAM, although an equivalent increase was perceived with regards to accuracy. The effort increase for RE was mainly due to learning curve, but for DEV the effort increase could be explained by the overall participation in this activity. DEV called the RAM early acceptance/dismissal of requirements a ‘‘pre pre-study’’, a new activity from DEV’s perspective. Although the increase in effort and the equivalent increase in accuracy seemed to cancel each other out, RE and DEV both indicated that it was worth it (indicated by fulfillment) as they realized that the second time around effort would probably be less as the learning curve was not as steep. In addition, RE felt that a positive spin-off effect was obtained, in the form of that overlapping requirements were dismissed earlier as the requirement levels were homogenous enabling better comparison between requirements.

In the case of PM, the effort was moderately less for early dismissal and an equally moderate increase in accuracy. The positive outcome was mainly motivated by an increase in requirements quality, and that all (a clear majority) of the requirements were comparable with regards to abstraction and detail. This increased requirements understanding, making it easier to dismiss/accept requirements at this early stage.

- *Specify requirement* shows a large effort increase for RE and a moderate increase for PM and DEV. However, V&V indicates an effort decrease. The decrease can be attributed to that better and more structured specification prior to V&V active involvement decreases effort for V&V.

All subjects indicate a substantial increase in accuracy, except DEV, which indicates a more moderate increase.

6.2.2 Part II

The subjects at ABB evaluated requirements quality from 11 different perspectives. In this part the views of a new subject is introduced, PL. PL did not participate in the requirements engineering activities, but uses requirements as input to the development projects, which are under PL’s responsibility in terms of planning, control and general administration.

All answers for Part II are displayed in Table 11. The rows are summarized giving a total quality attribute value for the RAM requirements per subject. All positive values in the table indicate an increase in quality, where a four is maximum, and vice versa for a decrease in quality with a value of negative four in the worst case.

Table 11 Results Part II for ABB requirement quality attributes

	Completeness	Unambiguity	Testability	Traceability	Understandability/ readability	Consistency	Catching dep. Relationships btw req.	Redundancy	Structure	Analysis base	Conformance to Business Goals	SUM per subject (max 44, min -44)
Requirement quality												
Subject RE	4	0	3	3	3	0	0	0	0	3	0	16
Subject PM	3	1	2	2	2	0	0	–1	0	3	0	12
Subject DEV	1	0	N/A	2	1	0	0	0	0	2	N/A	6
Subject V&V	2	2	4	3	2	0	1	–1	0	2	0	15
Subject PL	4	4	2	4	2	2	2	0	1	2	2	25
Sum per quality attributes (max 20, min –20)	14	7	11	14	10	2	3	–2	1	12	2	

It is noticeable that the highest score is obtained from PL, which uses the requirements in development efforts. DEV scores lowest values. RE, PM, and V&V score in the same range.

The columns are also summarized displaying the overall score for every quality attribute.

Figure 10 gives a different overview of the results divided by quality attribute.

- *Completeness* (total score 14 out of 20). All subjects (except DEV) agree that a substantial increase has been achieved with regard to completeness of the requirements. This is mainly due to the utilization of attributes and to some extent homogenous abstraction of the requirements on an abstraction level suiting RE, PM, V&V, and PL. The exception to this is expressed by DEV, who in general feels that the increase in quality is moderate to none for most quality attributes (barring some exceptions). The main motivation given was that the level of abstraction on which the requirements are specified are not optimal, even if some improvements can be seen. DEV feels that for the purpose of development some improvement can be seen (e.g., analysis base) but overall the requirements can be specified on a level more appropriate for development activities, i.e., broken down further.
- *Unambiguity* (total score 7 out of 20). PM, V&V and in particular PL feel that there has been improvement with regard to requirements' ambiguity. DEV feels that the requirements are too abstract to say they are less ambiguous. PM and especially RE feels that the natural language specification of the requirements text itself can be improved.
- *Testability* (total score 11 out of 16, not specified by DEV). The general consensus (with exception of DEV who does not feel qualified to answer) is that from the perspective of testability a clear improvement has been obtained. PL comments that although an improvement has been achieved there is still room for more improvement.
- *Traceability* (total score 14 out of 20). All subjects agree that a significant increase in traceability has been achieved. This impression is underlined by PL quoting especially traceability to requirements source/origin and requirements specifier as an improvement.
- *Understandability/readability* (total score 10 out of 20). All subjects indicate a significant improvement (except DEV). DEV quotes the need to have more general information about the problem and goals (why) of the requirements to improve understandability.
- *Consistency* (total score 2 out of 20). All subjects indicate that no improvement has been obtained with regard to consistency. The main motivation for this is

that there is a focus on individual requirements and not a conscious effort to specify requirements in a way that several requirements as a whole are consistent amongst each other. PL feels that some improvement has been achieved, as the requirements are more thoroughly worked-through.

- *Catching dependencies/relationships* (total score 3 out of 20). As in the case of consistency the total score is very low (i.e., no great improvement). The reasons are to a large extent the same as in the case of consistency, namely, that the focus is on individual requirements.
- *Redundancy* (total score –2 out of 20). Redundancy scores the lowest of all quality attributes, with PM and V&V actually indicating a slight decline (more redundancy) in comparison to pre-RAM. Once again, the main reason given by the subjects revolves around the fact that there is focus on individual requirements and that the same information sometimes has to be specified in several requirements, as it is important for those several requirements. This is often information handling goal-like information, or general risks and restrictions and so on.
- *Structure* (total score 1 out of 20). The improvement of this quality attribute is scored very low. In general, the consensus is that no improvement has been made. The reason for this can be traced to less than optimal tool support, and the lack of full sorting of requirements under more abstract ones.
- *Analysis base* (total score 12 out of 20). All subjects agree that the use of RAM (attributes and more detailed analysis and specification) has produced requirements of significantly better quality in this aspect.
- *Conformance to business goals* (total score 2 out of 16, not specified by DEV). The main reason for this relatively low score of improvement can be accredited to the fact that the abstraction level of the requirements resides too far from product strategies. This makes comparison very difficult.

6.2.3 Summary and discussion: ABB

Part I shows an overall increase in action accuracy. Positive fulfillment is also a general tendency even if the level of effort sometimes rivals the increase in accuracy, indicating that the extra effort is considered a good investment. The learning effort for RAM in general is low, although there is a substantial effort increase in the case of RE with regard to general requirements specification as the person filling the role was new to the task. A general piece of feedback often accompanying the ratings performed was that improved communication between the product management organization and the product organization was achieved using RAM. The positive effects of this were hard to quantify and not necessarily caught by this evaluation.

Part II displays some substantial improvements in requirement quality. Completeness, traceability, testability, analysis base, and understandability/readability all score high to fair improvements in comparison to previous requirements engineering/product management practices.

The quality attributes of completeness, unambiguity, consistency, catching dependencies/relationships, redundancy, structure, and conformance to business goals all score very low on improvement. The overall explanation for this by the subjects themselves is the fact that full work-

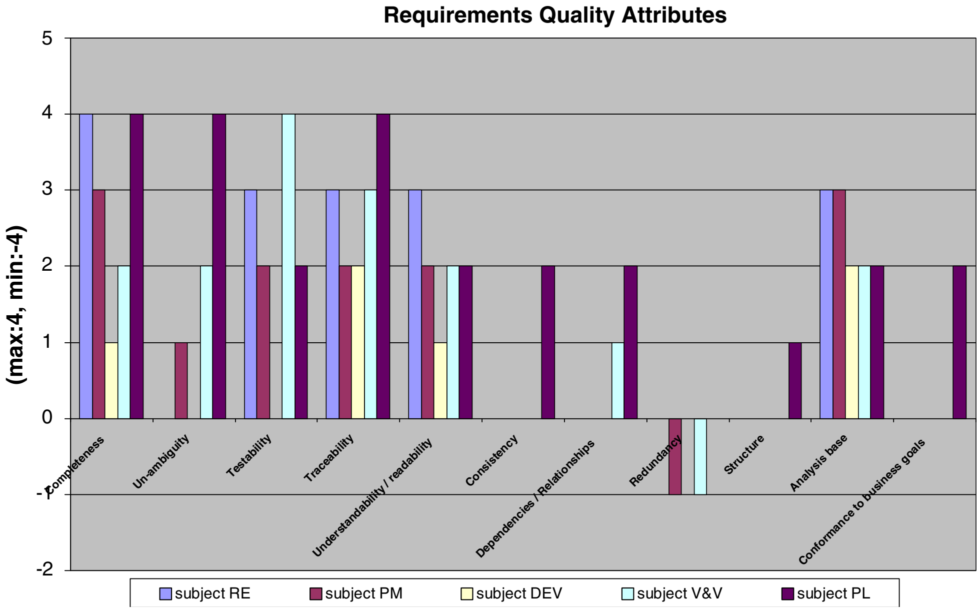


Fig. 10 Overview of requirements quality attributes

up of requirements according to RAM has not yet been implemented (they specify requirements on one level only, and use headers on the levels above). This was a conscious decision by ABB, planning for a stepwise implementation of which the first step is evaluated here. The implication being that the requirements created (one level of abstraction) suited certain roles and thus certain work efforts more than others in this phase of RAM implementation. For example, DEV saw some increase but not at all as high as e.g., PL or RE.

The use of full work-up (abstraction, and break-down) creating new requirements as needed is by no means a silver bullet taking care of all problems. However, the predominant feeling amongst the subjects was that this could further increase accuracy and decrease the effort needed to complete actions, as well as substantially improve requirements quality. Examples of this are specified below:

- Completeness, unambiguity, understandability/readability could be improved as requirements of different abstraction (specified on different levels) explicitly linked across levels offer different views of one or several requirements. In total offering a more complete picture, increasing understandability (especially the *why* sought by DEV), as the levels can be traversed. This gives the subject the choice of abstraction, and in total gives more information.
- Looking at consistency, dependencies/relations, and redundancy, the full use of abstraction levels could make it possible to store general information (that is true for several requirements) in one or at least in a few requirements on a more abstract level. This would lessen Redundancy, as similar information does not have to be specified repeatedly. In addition, full use of abstraction levels in RAM enables (and forces) a more holistic view. Related requirements on a lower level all connect to a more abstract requirement on an upper level. For example, a feature level requirement generally has several requirements on Function level that ‘‘belong to it’’, i.e., they have to be implemented in order for the Feature level requirement to be fulfilled. This encourages a group view of requirements (lessening the focus on just individual requirements). This may increase Consistency and make Dependencies/relations more explicit.

It should be noticed that the lack of the improvement of certain aspects of the requirements engineering/product management process (as can be seen above) was predicted when the stepwise implementation plan was decided post the RAM tailoring workshop. The evaluation results confirm these predictions (made by the process improvement responsible personnel, not the users of RAM). An improved tool support is also a factor here. The use of an appropriate tool was considered a prerequisite to full RAM implementation. The acquisition and training in tool usage will be a part of the continuation of RAM implementation at ABB.

6.3 Comparison of RAM evaluation results at DHR and ABB

The comparison presented here is not aimed at comparing the companies. The objective is solely to compare the experiences in using RAM.

Substantial improvements in Action accuracy and fulfillment, as well as requirements quality can be seen in the evaluation results for both DHR and ABB. However, some interesting differences merit notice. At DHR the different subjects (representing the span from product management to development) are in agreement to a larger extent than at ABB. This is especially evident in Part II where most roles at DHR see a very similar increase in requirements quality (see Fig. 8). At ABB (see Fig. 9) the rated quality increase fluctuates from generally high when asking RE, PM and PL, to low when asking DEV; the main reason for this can be attributed to the fact that a stepwise implementation of RAM was chosen at ABB. In the first step (evaluated in this paper), requirements were formed on one abstraction level only, resulting in that the increase in quality was perceived as substantially greater by the roles using requirements on that abstraction level. DEV in ABB’s case saw some improvements compared to before, but the requirements were specified on a somewhat too abstract level. At DHR the full work-up of requirements offers requirements on different levels of abstraction, each targeted at a different group, thus offering requirements appropriate for, e.g., RE and PM, but also for DEV.

Another noticeable difference between the two is that although DHR uses full work-up and thus in reality specifies more requirements relative to the incoming number, the redundancy is less than at ABB. Intuitively one might draw the conclusion that more specification equals more redundancy, but this does not seem to be the case. The reason for this can be found in the work-up itself. As requirements are abstracted and broken down the more abstract requirements gather the general information that is true for all detailed requirements under them. This results in that information of a general (but important) nature does not have to be specified in every detailed requirement, but maybe only once or twice in a more abstract requirement related to the detailed ones. Looking at Part I, the efforts needed to perform the Actions are in many cases lower at DHR than at ABB, this despite of full work-up.

The differences above do not come as a surprise to the process improvement team, and where expected by all team representatives, and as the second step of implementation is underway further improvements are anticipated.

It is important to notice that the process improvement activities and the evaluation presented in this paper in no way can be generalized to other areas. Process improvement activities at ABB have a long and positive tradition, and the maturity of other processes cannot be inferred in any way using the results presented here. The improvements seen in each company are only relative within the companies themselves and thus an improvement at DHR cannot be compared to an equal improvement at ABB.

7 Conclusions

The overall results of the evaluations indicate that the implementation of RAM at DHR and ABB has yielded substantial increases in both; the accuracy of the practices (actions) performed in requirements engineering/product management, and in requirements quality. It is only natural that these improvements have a price, which can be observed in some effort increase over the board, although there are also examples of the opposite. A learning curve effect can be used to explain some of the increase, but for the most part increased accuracy and quality will have some cost as more work is performed, although in the case of DHR and ABB, these costs are very moderate in total. The implications of improved quality and accuracy should also yield positive effects in the long run. For example, avoiding requirements overload by having improved acceptance/dismissal of requirements at an early stage will almost certainly save resources in development, enabling more ‘‘good’’ requirements to be implemented. One example of this is the possibility to reduce the number of pre-studies performed if rough estimates can be used to prioritize requirements, and enable the organization to focus on a set of requirements that is feasible. Improved estimation accuracy (as the requirements and the process itself supports estimation activities) will thus enable better resource planning overall. Higher quality requirements in general should also decrease defects in later stages of development.

It could be argued that the implementation of any good requirement engineering practice would yield the same result as presented in this paper. Without actually implementing, e.g., two models in parallel, it is impossible to dismiss this possibility. However, several things can be said about RAM that could indicate some advantages. The concepts behind RAM are based on using the reality facing industry (e.g., large amounts of requirements, multiple abstraction levels and limited resources) to enable scalability. This is of course not proven totally as of yet, but based on the evaluations performed indications are positive. Actual practitioners have used and are using RAM in real development situations.

Requirements abstraction model is based on needs identified in industry but does not push a predefined one-size-fits-all set of practices and rules on all organizations looking to adopt it. Rather the model is tailorable, maintaining certain concepts (use of abstraction and attributes) but adaptable enough to fit different environments. In addition, RAM is tool independent, but requires tool support to be scalable and practical. The evaluation of RAM at both ABB and DHR indicates that the tailoring aspect of RAM has worked, although it should be realized that the environments are not totally heterogeneous (nor are they homogenous).

It is very important to notice that both DHR and ABB are very successful companies in terms of their domain, have excellent engineering practices, and are ultimately populated by professionals. In addition to this, they have the maturity to realize that the improvement is always possible. These facts actually reflect positively on the evaluations as both companies had good requirements engineering/product management practices prior to RAM implementation. This implies that RAM was benchmarked against mature and working practices.

From a more academic perspective the evaluation can be seen as a case study where close collaboration with industry produced a requirements engineering model. It was tailored, implemented, and subsequently evaluated in a real industry environment by industry professionals. There are of course inherent challenges, with industry trials void of direct involvement by researchers. Availability of metrics is one as extensive measurement programs are rare, making expert opinion a valuable data source. From an academic standpoint real industry trials often implies relinquishing control as the researchers ‘‘hand-over results’’, which might explain why the number of large-scale pilots of new models and methods in industry are limited in amount (at least in comparison to the amount of models and techniques suggested by researchers). The evaluation presented in this paper shows that it is possible to perform large-scale industry trials without researcher participation, and an example how the results can be measured through the use of data available utilizing expert professionals.

RAM was evaluated in two separate organizations, a quite large and complex undertaking as extensive industry trials require time and resource investments both from the companies, but also from the researchers in terms of preparation and acquisition of commitment. An important lesson learned is that risk minimization for the involved industry partners is a crucial prerequisite. RAM was developed and refined in several steps prior to industry trials (the main focus of this paper). Thus, it is not enough to invent a model or technique and go out and test it. Rather, extensive and long-term collaboration with industry, basing the solutions on real industry needs is a prerequisite for getting commitment for trials as it builds trust in addition to making the results industry relevant.

As with any process improvement activity, there are confounding factors when evaluating an improvement. As improvement activities commence (e.g., the workshop, presenting RAM), it gives rise to added awareness, increases knowledge, and creates curiosity. These things in themselves can improve practices independent of the ''official'' process improvement, although all of this can be seen as a part of the technology transfer process, and ultimately any improvement is a good thing.

The evaluation presented in this paper can be seen from several perspectives. The evaluation was requested, and can be used, by the companies to gauge the effects of RAM implementation, catching both positive effects and possibilities for further refinement. The effects that have been evaluated from two perspectives, work performed (actions), and requirements quality (quality attributes) offer a good-enough indication for further commitment to process improvement using the concepts of RAM. In addition, a critical factor that was tested through the large-scale pilots evaluated in this paper was scalability of RAM. Testing RAM in a controlled environment (e.g., through experiments) or in a limited fashion (e.g., small pilots involving researchers as support) was not good enough for industries to commit to using RAM and incorporating the concepts as a part of their official development process permanently.

8 Future work

There are several efforts presently underway at both DHR and ABB, and plans for the continuing evolution of RAM as a framework for requirements engineering and product management support. They are described briefly below.

At DHR, RAM is maturing and the initial learning curve with regards to RAM and the use of new tool support is subsiding, allowing for a natural adoption of the new way of working. A part of this is of course a constant monitoring of the process and RAM in order to tweak and adapt the model to the needs. A part of this is to refine the measurement programs at the company to minimize the effort needed to monitor the process improvement. The next step at DHR is to include explicit support for requirements prioritization and packaging of requirements in RAM. In addition, in-project requirements engineering (e.g., technical specification and general requirements maintenance) will be assessed and streamlined to suit the needs of the organization.

At ABB, the next step consists of tool acquisition and implementation of full RAM work-up at the PM units where it is used today. A second part is to implement RAM in several other neighboring PM groups (an activity proceeded by additional RAM tailoring). Monitoring and stepwise refinement of all RAM instances implemented is also crucial and will be performed continuously, based on formal and informal evaluations.

RAM in general will continue to evolve. This will be achieved through lessons learned from industry implementation, but also through several experiments conducted in laboratory environment testing new concepts prior to industrial trials.

References

1. Ruhe G, Greer D (2003) 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, pp 262–271
2. Butscher SA, Laker M (2000) Market-driven product development. Mark Manage 9(2):48–53
3. Sommerville I (2001) Software engineering. Addison–Wesley, Essex
4. Wieringa R, Ebert C (2004) Guest editors' introduction: Re'03: practical requirements engineering solutions. IEEE Softw 21(2):16–18
5. Karlsson L, Dahlstedt Å, Natt Och Dag J, Regnell B, Persson A (2003) Challenges in market-driven requirements engineering: an industrial interview study. In: Proceedings of the 8th international workshop on requirements engineering: foundation for software quality (REFSQ'02), Universität Duisburg-Essen, Essen, Germany, pp 101–112
6. Kotler P, Armstrong G (2001) Principles of marketing. Prentice-Hall, Upper Saddle River
7. Lehmann DR, Winer RS (2002) Product management. McGraw-Hill, Boston
8. Mintzberg H, Ahlstrand BW, Lampel J (1998) Strategy Safari: a guided tour through the wilds of strategic management. Free Press, New York
9. Weber M, Weisbrod J (2003) Requirements engineering in automotive development: experiences and challenges. IEEE Softw 20(1):16–24
10. Gorschek T (2004) Software process assessment and improvement in industrial requirements engineering. Licentiate thesis no. 2004:07, ISBN 91–7295–041–2. Blekinge Institute of Technology, Ronneby. http://www.ipd.bth.se/tgo/_licentiate/papers/Licentiate
11. Gorschek T, Wohlin C (2003) 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, pp VI.1–VI.14
12. Gorschek T, Wohlin C (2004) Packaging software process improvement issues: a method and a case study. Softw Pract Exp 34(14):1311–1344
13. Gorschek T, Wohlin C (2006) Requirements abstraction model. Requir Eng J 11(1):79–101
14. Morris P, Masera M, Wilikens M (1998) Requirements engineering and industrial uptake. In: Proceedings of the 3rd inter-

national conference on requirements engineering. IEEE, Los Alamistos, CA, pp 130–137

15. Gorschek T, Garre P, Larsson L, Wohlin C (2006) A model for technology transfer in practice. IEEE Softw 23(6):88–95
16. Gorschek T, Svahnberg M (2005) A controlled empirical evaluation of a requirements abstraction model. Inf Softw Technol (can be obtained by email)(in press)
17. Gorschek T, Svahnberg M, Borg A, Börstler J, Eriksson M, Lonconsole A, Sandahl K (2005) A replicated controlled empirical evaluation of a requirements abstraction model. Inf Softw Technol (can be obtained by email)(in press)
18. Kotonya G, Sommerville I (1998) Requirements engineering: processes and techniques. Wiley, New York
19. El Emam K, Goldenson D, McCurley J, Herbsleb J (2001) Modeling the likelihood of software process improvement: an exploratory study. Empir Softw Eng 6(3):207–229
20. Rainer A, Hall T (2003) A quantitative and qualitative analysis of factors affecting software processes. J Syst Softw 66(1):7–21
21. Herbsleb J, Zubrow D, Goldenson D, Hayes W, Paulk M (1997) Software quality and the capability maturity model. Assoc Comput Mach Commun ACM 40(6):30–40
22. Herbsleb JD, Goldenson DR (1996) A systematic survey of cmm experience and results. In: Proceedings of the 18th international conference on software engineering, IEEE, Los Alamitos, CA, pp 323–330
23. Conradi R, Fuggetta A (2002) Improving software process improvement. IEEE Softw 19(4):92–100
24. Basili VR, McGarry FE, Pajerski R, Zelkowitz MV (2002) Lessons learned from 25 years of process improvement: the rise and fall of the NASA software engineering laboratory. In: Proceedings of the 24th international conference on software engineering (ICSE02), ACM, Orlando, pp 69–79
25. Sommerville I, Sawyer P (1999) Requirements engineering: a good practice guide. Wiley, Chichester
26. Wohlin C, Runeson P, Höst M, Ohlson MC, Regnell B, Wesslén A (2000) Experimentation in Software Engineering: An Introduction. Kluwer Academic, Boston, MA