A Case Study of Requirements Management: Toward Transparency in Requirements Management Tools

Markus Kelanti, Jarkko Hyysalo, Pasi Kuvaja, Markku Oivo
Department of Information Processing Science
University of Oulu
Oulu, Finland
{markus.kelanti, jarkko.hyysalo, pasi.kuvaja, markku.oivo}@oulu.fi

Antti Välimäki
Metso Automation Inc
Tampere, Finland
antti.valimaki@metso.com

Abstract—**Requirements Management (RM) is a continuous activity that provides an interface between the requirements of engineering and other system development activities. Current literature offers an extensive set of general requirements for RM tools, and several RM tools are available that utilize these requirements. Interviews as a part of a case study to enhance the tool support reveal that the current RM tools do not provide enough transparency to the development process and its activities. The results from these interviews show problems (even with the basic features of RM tools) in decision-making support, reporting, and follow-up of development activities. This paper discusses the problems revealed in the interviews, and suggests further requirements for RM tools to address the problems with transparency.**

Keywords-requirements management; requirements management tools; transparency

I. Introduction

Requirements Management (RM) is one of the areas perceived as critical in collaborative product development [1], since RM ties together Requirements Engineering (RE) and other product development activities. Therefore, RM has an important role, and it needs adequate tool support for managing the requirements and sharing the information. These tools will ensure the success of product development.

Even though the fundamental activities of RM could be done manually with pen and paper, tools are necessary for practical reasons [2]. The RM tools may offer many features such as a general repository, the ability to import from other tools, communication capabilities, traceability links, change control mechanisms, and information sharing [3]. However, our findings from the interviews with industrial experts show that transparency is not fully taken into account in RM tools. Therefore, we focus the study on identifying transparency requirements that allow the RM tools to provide information about the ongoing status of the development process, enable easy access to relevant information, and make the process more visible and transparent. Thus, our research problem is: *What transparency requirements should be set for RM tools?*

We propose that transparency requirements should be added to the list of requirements for RM tools. Transparency is required in both RM itself and RM tools that will support the developers, help them become aware of the status of development activities and items, and achieve a common, shared understanding about the development goals. All these are necessities in decision making, and help achieve effective and open communication, among other positive impacts, which are all essential for successful, productive development. In short, transparency is the awareness and visibility of what is going on.

The importance of different aspects of transparency and awareness enabling transparency is also recognized in literature [4, 5, 6, 7, 8, 9]. For example, Herbsleb [4] states that if developers have no knowledge what the others are doing, it often leads to misunderstandings in communication content and of motivation. This lack of awareness also makes it difficult to track the effects of changes in distributed collaboration spaces. Transparency in RE in distributed development is especially critical as requirements often emerge from different organizations that challenge the process transparency [9].

Requirements for RM tools already exist in the literature [10, 11, 12]; however, literature about transparency in RM and RM tools is quite scarce. Our contribution focuses on this gap, and we complement the existing knowledge with a new viewpoint—transparency. An industrial case study was conducted in a large global company that develops process automation systems for industrial users. The case study was executed as part of the AMALTHEA project, and it consisted of 11 expert and manager interviews to cover the development process and tools used. The case company uses traditional and agile development methods simultaneously in the same product development project. This kind of setting emphasizes the need for transparency, as the findings of our case study show. The results of the focused interviews with the case company's personnel provided several requirements for transparency-related features and properties for RM tools.

The rest of the paper is organized as follows: Section 2 examines related work; Section 3 outlines the research process; Section 4 presents the empirical study and discusses its results and implications; and Section 5 concludes the study and summarizes the key findings.

II. RELATED WORK

A. RM Tools

RM is a process supporting other RE processes (elicitation, analysis, specification, and verification); it ensures that requirements are documented and traceable, and that changes are properly handled [11, 13, 14]. While requirements form the basis for other development activities, RM provides an interface between RE and the other processes, continuing through the whole product development cycle. Literature defines RM as "the structuring and administration of information from elicitation, derivation, analysis, coordination, versioning, and tracking of requirements during the complete product lifecycle [14]." Several tools are available for managing the RM process [15].

The RM process is generally supported by an RM tool comprised of people assuming roles and responsibilities, processes, and tooling. It also manages the artifacts of the software and systems development process [2]. The tool support should not force specific processes, but should support the developers' tasks and provide the functionalities needed in their work. Current RM tools need to be configured for specific RE and development processes [16].

Literature on the subject provides a comprehensive set of requirements for RM tools and their features [10, 11, 12, 17, 18]. There are also efforts that summarize the available requirements. For example, [14, 17] analyze the literature and classify the RM tool requirements into three categories from the viewpoints of users, project administrators, and IT system administrators. A summary of requirement topics for each category, according to [17], is presented next.

Requirements from the tool users' points of view cover the core functions of an RM tool:

- Information model, views, formatting, multimedia and external files, documentation of history, baselining, traceability, analysis functions, tool integration, import, change management and comments, document generation, collaborative work, checking for offline use, and web access.

Requirements from the project administrators' points of view cover the issues that are not core functionalities but are needed for managing large-scale projects:

- Users' roles and rights, size restrictions, workflow management, and extensibility.

The third category proposes requirements from the tool and the IT system administrators' points of view, which cover the issues related to availability, reliability, and data security:

- Database and encryption.

In addition to dedicated RM tools, most RE tools also support RM; however, their RM capabilities are often inadequate due to a lack of open data model mechanisms, which relate to the recording of user actions, modification of data structures, and standard format of data [18]. Although a wide array of dedicated RM tools is available, and the needs and requirements for RM tools have been recognized in the literature, problems remain with even the basic features of RM tools. For example, requirements for traceability and change management still seem to be difficult issues [19], and both relate strongly to transparency. Most RM tools do not provide adequate support for large distributed projects, nor support the management of large numbers of requests, nor facilitate collaborative RE [16, 20]. There are also usability issues [12, 14] and a lack of support for collaborative work [12].

B. Transparency

Besides these reported problems, we found that the aspects of transparency in RM tools are only partly discussed in the literature. Requirements concerning the awareness of the states of the process and work items are only briefly mentioned under different topics [21, 22]:

- Openness of communication and information sharing;
- Visibility of and access to data, documents, and work items;
- Visibility of decision-making processes and decisions;
- Visibility of processes;
- Transparency of collaboration; and
- Transparency of tools.

Awareness can be defined as the understanding of others' activities, which also provides the context for one's own activities [23]. It is suggested that awareness is the key to transparency [5], and awareness is particularly important in RM [21].

Relevant literature was studied to understand transparency and awareness in an RM context. The following synthesis is based on the literature study and the transparency-related topics that emerged. In the context of RM, transparency can be regarded as the awareness of the following topics:

- *Process support* [11, 12, 14, 17]: It is important to be aware of the states and the histories of software project tasks and the characteristic work activities that describe the environment within which they are performed [24]. Transparent RM tools enable workers to understand the context of their work, which helps them understand their own goals and relate them to others' goals and work. The main concerns are process states, progress, histories, and context.
- *Tooling and work items* [14, 17, 24, 25]: Awareness support is needed to provide information about development artifacts involved in RM in order to have a successful, distributed RM environment [21]. The main concerns are work artifacts, their states and changes, results, documents, data, and context.
- *Decision making* [21, 26, 27]: Awareness about the decision-making process is needed, and forums allow tracking the progress of the states of the requirements. This allows workers to be aware of the person who is working on a particular decision [21]. Forums can also keep track of RE decisions, their rationale, and their effects on software products [28]. The main concerns are decision-making forums, rationale, reasoning process, visibility, and documentation.
- *Collaboration and communication* [6, 22, 24, 25]: RM is often physically distributed work among stakeholders from various organizations [21]. It is important to know what others' roles and responsibilities are, and what they are doing, as it helps to coordinate the collaborative work and diminishes the problem of overlapping work. It is important in RM to understand dependencies, that is, to have the awareness of the other entities that are connected with the one that is being manipulated. This enables an individual to see the impact of one's work on those of others [22]. The main

concerns are visibility of others' actions, skills and competencies, and information access and exchange.

- *Organization and strategy* [6, 29]: Requirements to be implemented need to be synchronized with portfolios and roadmaps that are based on organizational strategy and goals. The RM tools must have the transparency towards organizations' strategies, visions, and goals. For example, Berggren and Bernshteyn [6] suggest "breaking down the strategy into definitive and meaningful components upon which individual employees can act." The main concerns are visions, goals, motives, portfolios, and roadmaps.

The areas of decision making, collaboration and communication, and organization and strategy are often omitted or not addressed extensively in RM tool literature.

III. Research Process and Case Context

The case company uses a project-based approach to develop automation platforms for industrial automation purposes. Interviewees work in the development process, with the aim to improve and implement new functions in those platforms. The development process roughly follows this pattern: requirements elicitation, requirements feasibility analysis, project planning, product design, implementation and testing, and maintenance. In this development process, the purpose of requirements elicitation and feasibility analysis is to gather requirements from different stakeholders, evaluate their technical feasibility and business potential, and generate potential features for an automation platform. One or more features are selected in the project planning phase, where a project is created to implement the selected features. Product design, and implementation and testing are then done for that project. When the feature is released to the customer, it enters the maintenance phase.

A case study was initiated in the company to examine its current RE and RM practices and tools in order to improve them so they would better suit the developers' and managers' needs. The research process used in this case study is shown in Fig. 1.

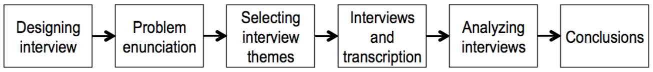

Figure 1. Research Process

At first, relevant research topics were identified with company representatives and researchers, and rough analyses were done on literature, company materials, company presentations, and other sources. Based on this information, the interview was designed to include 12 main questions, each with several sub-questions, to cover current development processes, practices, tools, pros and cons, and possible improvement proposals. A questionnaire template was created and improved in an iterative manner between researchers and the company representative. After a version that satisfied all parties was created, the actual interviews were conducted. The final questionnaire comprised 11 main questions covering the following topics: terminology, currently used methods and processes, tools, information needs and uses, responsibilities, and pros and cons perceived by the interviewees.

In total, 11 interviews were planned and performed in the case company. Seven of the interviewees were designers and engineers working in the development process, and four were managers from different levels in the organization.

The interviews were executed over a period of nine months. Seven interviews were conducted during the autumn of 2012, and four during the winter of 2012-2013. The duration of each interview was approximately 1.5 hours. The questionnaires were delivered to the interviewees in advance, so they could prepare themselves for the interview. Two researchers conducted the interviews, mostly face-to-face. All of the interviews were recorded and transcribed, and the researchers wrote a short summary of each transcription. The summaries and transcriptions were sent to the respective interviewees within two weeks, and they were given one week to validate the information. The feedback and change requests were taken into account during analysis, but only a few interviewees made any (minor) corrections or added anything to the summaries.

In the next step, the validated information was analyzed to find themes in the content. Nvivo 10 was used to store the interview data and to help facilitate the analysis process. Nvivo 10 was selected mainly due to the researchers' familiarity with the tool, its support for the different coding techniques applied for the data analysis, and theme identification. The interview data was auto-coded first based on the questions on the interview template. The next step was to analyze and code the data to find major themes from the interviews. The interviewees also reviewed these analyses individually, and a workshop was arranged with them to discuss the results further. Based on the interviews, analyses, reviews, and the workshop, we identified one major theme—transparency in RM tools. After this, the data was analyzed to find the transparency requirements. The researchers also analyzed and coded the data to identify other possible themes related to transparency, based on the literature review.

IV. Findings and Analysis

The case company uses three main tools to manage its requirements: Jira, Polarion, and a proprietary application developed in-house. The case company uses different development methods, depending on the system under development and the technology involved. Jira is mainly employed to manage agile projects, while Polarion is used only for safety-critical systems. The proprietary application is used to store information about requirements and features, and it supports other development methods applied in the case company (ranging from adapted waterfall methods to agile approaches). It offers basic information fields and the functionality to record, link, and store the data into a database accessible by users. The proprietary tool is currently used to manage all requirements and features. Other tools used in the process are mainly Microsoft Office products like Word, Excel, Visio, and PowerPoint, as well as some tools developed in-house for testing and demonstration purposes.

Based on the interviews, it became obvious that the current tools used in RE are unable to provide visibility, easy access to information, or knowledge about what is happening in the development process at any given time. Throughout the interviews, the respondents constantly raised the issue of their inability to access information relevant to their work. This problem causes unnecessary resource consumption in the decision-making, reporting, and follow-up areas of the development activities. These issues concentrate on the RM aspect of these tools, and after comparing them with the literature about RM tools [12, 14, 17, 21], we concluded that there is a gap within the transparency aspect of these particular tools used in the case company's RE process.

The following sections present the identified transparency requirements and the requirements that affect transparency, with references to existing literature, if the latter backs up the requirements. The sections are divided according to the transparency categorization presented in the related work section. The identified requirements are summarized in Table I.

A. Process Support

R1) RM tool shall provide information about the states of the process and tasks

Literature suggests that an integral part of the development process is the awareness of the states and histories of software project tasks and work activities [24].

Understanding the current state is needed to enable the developers to react to changes and unexpected events. It also builds a shared understanding, which is an integral part of cooperative development [24].

Both managers and engineers voiced the need to know the status of a task or a process. The most commonly mentioned situation for this is when customers request information about the development status, especially estimations for when a product will be ready for delivery or piloting. Currently, this information is not easily available, and sometimes months can pass before any information reaches the customer. On the other hand, management may need information about the status of a project to make estimations and check whether the schedule and resources are up to date. This information needs to be collected manually, since current tools are inadequate.

R2) RM tool shall only show the task-relevant information

Interviewees commented that some of the tools they use tend to display a lot of information: status, historical data, design documents, comments, and so on. This helps improve transparency, but if the information is irrelevant to the current task, it overwhelms the users. Another danger is when the information is not updated regularly in the RM tool, but kept in separate documents in other databases, on developers' PCs, or in emails. This problem was also revealed by interviewees. Therefore, to support transparency, task-relevant information must be available and easily accessible, without any additional effort.

R3) RM tool shall support the actual development tasks

Related to the relevant information for tasks, the RM tool should obviously support the actual development tasks. Some of the interviewees are more engaged with agile development methods, and they commented that both Jira and Polarion are more suitable for their work. According to them, both tools are better designed for the development tasks used in either agile processes or safety-critical applications. Only necessary information for a development task should be visible in the tools used by the developers.

R4) RM tool shall provide task guidance

Heuristic knowledge and providing ways-of-working to guide developers while performing systems development are needed. They are useful, for example, for decision-making purposes or activities needed to create the conceptual specifications of the system [30]. A transparent RM tool should not only help workers understand the work context and its goals, and relate them to others' goals and work, but also provide guidance about what kinds of information workers need to produce in the development tasks.

For example, the developers reported in the interviews that financial estimations are especially essential in several tasks, but it is very hard to estimate with the current tools and available information. These estimations are used in different parts of the process to make decisions, and it is important to know how to do those estimations, and in what format the information should be documented.

TABLE I. TRANSPARENCY REQUIREMENTS FOR RM TOOLS

Topic	#	Requirement for RM tool	Related work
Process support	R1	Provide information about the state of the process and tasks	[14, 17, 24]
Process support	R2	Show only the task-relevant information	
Process support	R3	Have task views that match the actual development tasks	[14, 17]
Process support	R4	Provide task guidance	
Process support	R5	Provide process guidance	
Tooling and work items	R6	Provide information about development artifacts	[24]
Tooling and work items	R7	Provide standard information templates for RE items	
Tooling and work items	R8	Support linking	[17, 23, 24]
Tooling and work items	R9	Maintain link validity	
Tooling and work items	R10	Enforce linking rules among items	
Tooling and work items	R11	Support traceability	[1, 11, 12, 14]
Tooling and work items	R12	Support version control	[11, 12, 14, 17, 24, 25]
Decision making[a]	R13	Provide the rationale and reasoning process for decisions	[24]
Decision making[a]	R14	Provide visibility of decisions and their documentation	[28, 29]
Decision making	R15	Be able to generate status reports from processes	[12, 14, 17, 21, 24, 25]
Collaboration and communication[a]	R16	Provide awareness of others' actions	[24, 25]
Collaboration and communication	R17	Provide support for information sharing between management and developers	
Collaboration and communication	R18	Enforce a coherent terminology for RE items	[24, 29]
Organization and strategy	R19	Support breaking down the strategy, vision, goals, and motives into work tasks	[6, 24, 25]
Organization and strategy	R20	Provide information about available resources, skills, and competencies	[21, 26, 27, 28]

[a] These requirements are suggested by the literature, but not specifically mentioned in the interviews.

R5) RM tool shall provide process guidance

Furthermore, when this information is produced, it is not clear where and how it is utilized. Management needs information about the current state, and to get it in the form they need, it has to be inserted in a certain format and from a certain viewpoint. If users are not presented with proper guidance to create information, including where and how it will be used, it will not be as reliable as it should be. This is especially true in RM tools, where accurate information is crucial. While these requirements do not directly support transparency itself, without them, the information will not serve its purpose and can even cause negative outcomes.

B. Tooling & Work Items

R6) RM tool shall provide information about development artifacts

In software development, the artifacts are mostly documents and code. Literature suggests that awareness support is needed to provide information about development artifacts involved in RM in order to have a successful, distributed RM environment [21]. This awareness provides up-to-date information to stakeholders for better decision making.

Generally, the results from the interviews indicate a clear need to access information regarding any item in the development. These items include a single requirement, project status, use case, original request, and so on. The main reason is quite clear—interviewees need more information in order to perform their tasks. They often also need old documentation, previous work items, or other items linked to the item they are working on. This is true for both engineers and managers, and both commented that it is important to access information about a single item in order to learn its status, who is working on it, and generally understand its status.

R7) RM tool shall provide standard information templates for RE items

In the process, standard templates are used for documenting needs and requirements that contain basic information necessary for determining business potential, technical feasibility, and other relevant information for decision making. This is considered a good practice in general, but interviewees pointed out that these templates need to match the information needs of the tasks or activities at hand.

R8) RM tool shall support linking

The literature also discusses how one's work may impact those of others [22]. This includes artifacts and associated tasks, collaborators, and the concurrent work context of collaborators and resources [24]. Awareness of the context and others' actions makes it possible for developers to structure their interactions and cooperative processes, and to provide a context for one's own activities [23, 24].

Interviewees generally agreed that one of the main functions in the tools they use is the ability to link different items. This functionality is considered necessary to show dependencies and relationships among different requirements, features, and products. The ability to link different items is essential to the developers, particularly, how changes they introduce will affect different parts of the platform they develop. Since many developers work on a single platform or product, it is important to know the relevant items others are working on.

R9) RM tool shall maintain link validity

Another challenge related to linking different items in a tool is that the links sometimes connect to the wrong versions of the development artifacts. This can cause wrong versions to be implemented and tested. It also becomes increasingly difficult to search for information. This is especially true when data are searched after some time, and the item is not in the fresh memory. Developers clearly need to access valid information that points to the correct, updated version. If the validity is ignored, the link itself becomes useless. If this functionality is ignored, it can lead to situations where wrong versions are used in the work, and conflicts will arise.

R10) RM tool shall enforce linking rules among items

However, just enabling functionality to link and keep the links up to date is not enough. Interviewees also commented that linking practices should be enforced to keep the links coherent and understandable. Current tools in the case company allow anything to be linked in several different ways, with no generally accepted conventions for their use. This has led to unnecessary complexity with the database and tool, as individuals follow their own preferences. It was suggested that there should be rules and restrictions on the kinds of links to be used and the ways they should be described. The RM tool should enforce these rules to maintain cohesion, which will enable better transparency.

R11) RM tool shall support traceability

Traceability is one of the basic functions and requirements for RM tools. Traceability is needed to maintain and follow the relationships among requirements and design, implementation, and test artifacts [10]. With good tool support, traceability could enable analysis that would otherwise require more effort [17].

This is also one of the key functionalities, according to interviewees. On several occasions, interviewees mentioned that lack of traceability is troublesome because it hides what has already been done for a requirement. When this happens, they have to investigate what has been done in order to understand how the item has been developed in the past and where it originated. Testing would benefit if they could trace the requirements back to their original sources to see how things should work in the system.

R12) RM tool shall support version control

Enabling traceability has also led to a demand for proper version control, since this is lacking in most of the current tools. Without version control, it would be hard to know what has been done for any given item in the process.

C. Decision-making

R13) RM tool shall provide the rationale and reasoning process for decisions

To support decision making, the RM tool should provide identified criteria for evaluating the achievements. Moreover, decisions need to be explained and transparent for all relevant stakeholders. This improves the overall effectiveness of the RE process and provides understanding about the nature of the decisions made. It is necessary to keep track of RE decisions, including their rationale and effects on the product [28].

For managers and developers, the decisions are made mostly among the relevant parties, and the rationale is generally available for the interested stakeholders. Even in this case, knowing the rationale for a decision is still important, and the interviewees mentioned the times when they might need to communicate results to a customer.

R14) RM tool shall provide visibility of decisions and their documentation

The literature suggests ways to provide visibility of decisions. Decisions need to be documented and fed back into the system, so the workers can benefit from the experience [21]. Decisions also need to be integrated into organizational information systems; this allows them to be better understood by relevant stakeholders [26].

Customers often present their needs and wait for the company to react to them; all of the interviewees pointed out that customers should be told the reasoning for the decision when it finally comes. Interviewees expressed that this information should be available in the RM tool, either directly visible for the customer or for the developers to inform the customer.

R15) RM tool shall be able to generate status reports from process

One of the main concerns for managers is that the current company tools do not allow them to generate status reports such as project status, feature status, portfolio, and overall status reports from several projects. They commented that they can access some of the necessary information in the existing tools, but the tools should only provide the information they need and not just everything that is available. Due to the lack of this kind of functionality, the management has to collect the information by asking each project manager individually in order to generate the reports themselves.

Managers also expressed a clear need for constant reporting support from the tool. They especially need up-to-date reports on the various projects they are managing in order to track problems, delays, and progress in general. Project managers need to communicate information to upper management and customers about the schedule and progress. Portfolio reports, project reports, or feature status reports were all mentioned as important. The RM tool would therefore need to synthesize reports on the basis of need.

”I think that this kind of upper-level project management is not possible with the current tool. And this kind of overview to all projects is missing. One has to pick up the pieces of information to create the overview. That is the biggest shortcoming in the tool, in my opinion.” (Interviewee)

D. Collaboration & Communication

R16) RM tool shall provide awareness of others' actions

The RE is inherently distributed [21]; thus, there are awareness needs in RE and RM. In collaborative work, it is important to know what others' roles and responsibilities are, and what they are doing, because it helps diminish the problem of overlapping work. It is also highly relevant to have knowledge of others' interactions with the space and its artifacts. This helps with understanding who is working with what artifact and the artifacts of interest [7, 24].

While it is not necessary to know what a single developer or manager is doing at a certain moment, interviewees mentioned the need to generally know what is happening. This information is considered useful for making plans for future projects and for usage of resources, from management's perspective.

R17) RM tool shall provide support for information sharing between management and developers

Interviewees also said that transparency among different units, developers, and management would result in better understanding about the business and the real-world use potential of the products. This is not only tied to RM tools; often they are the tools used by management, while developers are the most important source of information in this area. Therefore, to establish proper transparency through information sharing, the RM tool needs to enable information flow from developers to management.

R18) RM tool shall enforce a coherent terminology for RE items

The relevance of information changes across different contexts; thus, the context should always be understood. As previously mentioned, understanding their work context enables workers to understand their own goals and relate them to others' goals as well. For example, Basili *et al.* (2007) suggest that “context specification is an important part of defining goals and deriving measures, since it prevents drawing wrong conclusions from the analysis” [29]. The evolving internal and external state of information characterizes the situation of entities in a shared environment [24].

During the interviews, the understanding of RE concepts (such as requirements, features, RE, and RM) varied from one interviewee to another; they often had different terms for similar concepts. Between the engineers and managers, this does not cause too much trouble because they are able to communicate face-to-face, but when they communicate with someone in another location, these differences are a potential source of misunderstanding.

E. Organization & Strategy

R19) RM tool shall support breaking down the strategy, vision, goals, and motives into work tasks

Transparent goals help the collaborative work and improve efficiency by reducing redundant work. Strategy transparency can be stated as “breaking down the strategy into definitive and meaningful components upon which individual employees can act" [6]. Strategies, visions, goals, and motives should be transparent and understood at all levels of work, and defining the portfolios and roadmaps based on organizational strategy and goals is suggested.

Interviewees expressed a need to see the plans and short-term roadmap for any automation platform they develop. They commented that it helps them decide what is needed and what areas they should prioritize. If this functionality would be available in the RM tool itself, it would remove the need to use time and other tools to find the information they need in their work.

R20) RM tool shall provide information about available resources, skills, and competencies

An integral part of the process is the awareness of the expertise of the developers working on the project [24]. A clear understanding about the availability of the talent pool in the organization enables the alignment of talents with the organizational strategy and development tasks.

During the interviews, both managers and developers expressed the need to access information regarding the available resources and competencies within the company. Managers need better information about the resources available for project planning, so they can satisfy the customers' needs and schedule the releases. Developers need to know about persons who can provide further information or clarification for requirements, in case the existing information is not sufficient.

V. Conclusion and Future Work

In this paper, we studied RM and RM tools in a large global organization that develops process automation systems. In a collaborative setting, different organizations, or even teams within an organization, may use various development methodologies and tools, causing challenges for RM; thus, support for transparency is required.

Based on our findings from the interviews and literature, RM tools should support transparency and provide the features needed for awareness creation. This paper has presented a set of necessary requirements for RM tools to support transparency. We have categorized these requirements under the following topics: process support, tooling and work items, decision making, collaboration and communication, and organization and strategy. We have also emphasized those transparency requirements that are already included in the requirements list for RM tools, but are still regarded as inadequately addressed.

A. Case Validity and Limitations

Study validity was addressed in several ways. Construct validity was dealt with through an extensive literature review, comparison of previous findings with current research using multiple sources of evidence, and utilization of key sources as reviewers. Internal validity regarding cause-effect relations was handled via multiple sources of evidence and iterative research, which gradually built the outcome. External validity involving the generalization of the results was tackled by having different organizational units as evaluation platforms. While the interviews were conducted only in one company in the automation domain, the literature supports the findings in different domains. However, a study in other organizations may introduce new requirements for transparency. The purpose of this study is not to suggest statistical generalizations but to enable generalization of the results to cases that have common characteristics. For further generalization, more studies are required. Finally, reliability was managed with rigorous research protocol, documentation, data collection procedures, and peer reviews.

B. Implications for Research and Practice

These results should interest both researchers and practitioners, since transparency requirements for RM tools are not extensively discussed in the literature. This study provides insights for academic research and lays the groundwork for further scholarly inquiry, for example, in validating the results in other domains and development phases.

Practitioners could learn to understand the importance of transparency in RM and RM tools, and thus have those requirements implemented in the tools. If transparency is addressed adequately, it can also benefit the practitioners by enabling better decision making and information flow in the development processes. Transparency will also help the development process and improve product quality, as well as the efficiency of the development.

C. Areas for Future Work

There is still a place for further work, and our intention is to validate the findings in the telecommunication and automotive industries. We also aim to have transparency requirements taken into account in applications other than RM tools. Additionally, RM tools should still be able to monitor and provide support for users, even if different development methods are used to build the systems. The needs of different development methods are another area for future work. Finally, we intend to implement the requirements in a prototype tool for practical validation and evaluation purposes in a follow-up study, where we will also examine how currently available RM tools conform to the transparency requirements presented in this paper.

Acknowledgment

This research is supported by the European ITEA2 program with national funding from Tekes (the Finnish Funding Agency for Technology and Innovation). The authors would like to thank AMALTHEA project partners for their assistance and cooperation.

References

[1] J. Hyysalo, P. Parviainen, and M. Tihinen, "Collaborative Embedded Systems Development: Survey of State of the Practice," Proc. IEEE Symp. Engineering of Computer-Based Systems, (IEEE 13), 2006, pp. 130–138.

[2] O. Gotel and P. Mäder, “Acquiring tool support for traceability,” in Software and Systems Traceability. Springer London, 2012, pp. 43–68.

[3] K. E. Wiegers, “Automating requirements management,” in Software Development, 7.7, 1999, pp. 1–5.

[4] J. D. Herbsleb, “Global software engineering: The future of socio-technical coordination,” in Future of Software Engineering, IEEE Computer Society, 2007, pp. 188–198.

[5] M. Beaudouin-Lafon and A. Karsenty, “Transparency and Awareness in a Real-time Groupware System,” Proc. ACM Symposium on User Interface Software and Technology, (ACM 05), 1992, pp. 171–181.

[6] E. Berggren and R. Bernshteyn, “Organizational Transparency Drives Company Performance,” Journal of Management Development, 26.5, 2007, pp. 411–417.

[7] C. Gutwin, S. Greenberg, and M. Roseman, “Workspace Awareness in Real-time Distributed Groupware: Framework, Widgets, and Evaluation,” Proc. International Conference on Human–Computer Interaction: People and Computers (11), Springer London, 1996, pp. 281–298.

[8] R. E. Grinter, “Using a Configuration Management Tool to Coordinate Software Development,” Proc. Conference on Organizational Computing Systems, (ACM), 1995, pp. 168–177.

[9] D. C. Gumm, “Distribution dimensions in software development projects: a taxonomy,” in Software, 23.5, IEEE, 2006, pp. 45–51.

[10] I. Sommerville and P. Sawyer, Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 1997

[11] G. Kotonya and I. Sommerville, Requirements Engineering: Process and Techniques. John Wiley & Sons, 1998

[12] M. Lang and J. Duggan, “A tool to support collaborative software requirements management,” in Requirements Engineering, vol. 6, no. 3, 2001, pp. 161–172.

[13] K. E. Wiegers, Software Requirements. Microsoft Press, 2000.

[14] M. Hoffmann, N. Kuhn, M. Weber, and M. Bittner, “Requirements for Requirements Management Tools,” Proc. IEEE International Requirements Engineering Conference, (IEEE 12), 2004, pp. 301–308.

[15] INCOSE The International Council on Systems Engineering, “Tools survey: RM tools,” http://www.incose.org/ProductsPubs/products/rmsurvey.aspx (referenced February 8, 2013.)

[16] A. Finkelstein and W. Emmerich, “The Future of Requirements Management Tools,” Information Systems in Public Administration and Law. Austrian Computer Society, 2000.

[17] D. Beuche, A. Birk, H. Dreier, A. Fleischmann, H. Galle, G. Heller, D. Janzen, I. John, R.T. Kolagari, T. von der Maßen, and A. Wolfram, “Using Requirements Management Tools in Software Product Line Engineering: The State of the Practice,” Proc. Conference on Software Product Line Engineering, (IEEE 11), 2007, pp. 84–96.

[18] J. M. Carrillo de Gea, J. Nicolás, J. L. Fernández Alemán, A. Toval, C. Ebert, and A. Vizcaíno, “Requirements Engineering Tools: Capabilities, Survey and Assessment,” Information and Software Technology, vol. 54, no. 10, 2012, pp. 1142–1157.

[19] G. Sabaliauskaite, A. Loconsole, E. Engström, M. Unterkalmsteiner, B. Regnell, P. Runeson, T. Gorschek, and R. Feldt, “Challenges in aligning requirements engineering and verification in a large-scale industrial context,” in Requirements Engineering: Foundation for Software Quality, Springer Berlin Heidelberg, 2010, pp. 128–142.

[20] P. Laurent and J. Cleland-Huang, “Lessons learned from open source projects for facilitating online requirements processes,” in Requirements Engineering: Foundations for Software Quality, Springer Berlin Heidelberg, 2009, pp. 240–255.

[21] D. Damian, J. Chisan, P. Allen, and B. Corrie, “Awareness Meets Requirements Management: Awareness Needs in Global Software Development,” Proc. International Workshop on Global Software Development, International Conference on Software Engineering, 2003, pp. 7–11.

[22] M. A-D. Storey, D. Cubranic, and D.M. German, “On the Use of Visualization to Support Awareness of Human Activities in Software Development: A Survey and a Framework,” Proc. ACM Symposium on Software Visualization, (ACM), 2005, pp. 193–202.

[23] P. Dourish and V. Bellotti, “Awareness and Coordination in Shared Workspaces,” Proc. ACM Conference on Computer Supported Cooperative Work, (ACM), 1992, pp. 107–114.

[24] I. Omoronyia, J. Ferguson, M. Roper, and M. Wood, “A review of awareness in distributed collaborative software engineering,” Software: Practice and Experience 40.12, 2010, pp. 1107–1133.

[25] W. Prinz, H. Löh, M. Pallot, H. Schaffers, A. Skarmeta, and S. Decker, “ECOSPACE – Towards an Integrated Collaboration Space for eProfessionals,” Collaborative Computing: Networking, Applications and Worksharing, IEEE, 2006, pp. 1–7.

[26] G. Ruhe, “Software engineering decision support–a new paradigm for learning software organizations,” Advances in Learning Software Organizations, vol. 2640, Springer Berlin Heidelberg, 2003, pp. 104–113.

[27] C. Jensen and W. Scacchi, “Collaboration, Leadership, Control, and Conflict Negotiation and the netbeans.org Open Source Software Development Community,” Proc. Hawaii International Conference on Systems Sciences, (IEEE 38), 2005, p. 196b.

[28] A. Aurum and C. Wohlin, “The fundamental nature of requirements engineering activities as a decision-making process,” Information and Software Technology 45.14, 2003, pp. 945–954.

[29] V. Basili, M. Lindvall, M. Regardie, C. Seaman, J. Heidrich, J. Münch, D. Rombach, and A. Trendowicz, “Bridging the gap between business strategy and software development,” Proc. International Conference on Information Systems, (ICIS 28), 2007, pp. 1–16.

[30] C. Rolland, C. Souveyet, and M. Moreno, “An approach for defining ways-of-working,” Information Systems 20.4, 1995, pp. 337–359.