Applying a Systematic Approach to Link Requirements and Testing: A Case Study

Conference Paper · December 2009

DOI: 10.1109/APSEC.2009.62 · Source: DBLP

Johanna Järvinen
Metso Corporation
1 PUBLICATION **20** CITATIONS
SEE PROFILE

Marjo Kauppinen
Aalto University
47 PUBLICATIONS **1,088** CITATIONS
SEE PROFILE

Eero J. Uusitalo
Aalto University
8 PUBLICATIONS **103** CITATIONS
SEE PROFILE

Applying a Systematic Approach to Link Requirements and Testing:
a Case Study

Johanna Kukkanen, Kyösti Väkeväinen
Metso Automation Inc.
Helsinki, Finland
{johanna.kukkanen, kyosti.vakevainen}@metso.com

Marjo Kauppinen, Eero Uusitalo
Software Business and Engineering Institute
Helsinki University of Technology
Helsinki, Finland
{marjo.kauppinen, eero.uusitalo}@tkk.fi

Abstract—**Requirements engineering and testing offer two complementary views of system development, and can, therefore, benefit from each other. This paper describes the lessons Metso learned from jointly improving requirements and testing processes. The goal of the process improvement was to increase the quality of R&D work. The new requirements and testing processes were piloted in an R&D project. The usage of the new processes removed the risk of doing double work, improved the visibility of the project status and the effectiveness of the project. The experiences of the pilot project suggest that the good results were mainly due to linking requirements and testing concurrently at process, people, and practice levels. It was essential that the new requirements and testing processes were integrated throughout the entire R&D project. In order to integrate the requirements and testing activities, critical roles and responsibilities were defined. Furthermore, we identified a set of good practices that helped perform the new processes.**

Software engineering; requirements engineering; software testing; process improvement; case study

I. Introduction

Requirements engineering (RE) and testing represent complementary views of a system and thus have a synergistic relationship [1], and both can benefit from taking the other discipline into account [2]. RE and testing can be linked by practices such as having testers participate in requirements inspections, testing requirements against business objectives and planning tests in parallel with requirements analysis [1-4]. The aforementioned practices are indeed being used in the industry in varying sets of practices which differ also in implementation [5]. However, practices alone do not define a process, but rather support the processes in place.

Published experiences on concurrently improving RE and testing processes seem to be scarce. Damian et al [6] present an interesting case study where RE processes were simultaneously improved with other process areas, including quality assurance. However, the focus of the work is on the effects of the RE process improvement. Further work by Damian and Chisan [7] reveals a major positive impact of RE process improvement on testing, resulting in improved testing productivity and more effective risk management.

This paper describes what Metso learned from the concurrent improvement of requirements engineering and testing processes. Metso is a Finnish company that develops and manufactures intelligent valve controllers and valves with over 28 000 employees globally. The results of this case study are based on a year-long development project where new processes, roles and practices were created and subsequently applied in a pilot R&D project.

The paper is structured as follows: Section II describes the initial state of the RE and testing processes. Section III gives an overview of the development project. The results of the process improvement and the lessons learned are described in Section IV. Section V presents the results of a self-assessment conducted both before and after the process improvement effort. In the final part, the key lessons learned are summarized.

II. Initial State of the Case

Metso is one of the leading developers and manufacturers of intelligent valve controllers and valves. Metso's products are used, for example, in energy, oil and gas industries as well as in pulp and paper industries. Valve controller products typically consist of software, firmware, electronics and mechanics. Part of the product offering is targeted for safety critical systems. Especially in the safety related cases, it is important that a link between requirements and test cases is created to show that the necessary requirements are met [8].

Our primary goal in Metso is to continuously improve both customer satisfaction and product quality. In addition, there is a need to run the R&D projects within the planned schedules and budgets. The main challenges in our R&D unit were related to the quality of our project work causing, for example, delays in schedules.

Initially, a general stage-gate type of R&D process [9] was in use. Detailed requirements and testing processes were not described in our R&D organization. R&D projects applied requirements and testing practices in different ways, i.e., no systematic and common practices were in use. Requirements and testing were not fully combined; testing was mostly based on the extensive experience of testers, instead of the requirements. The links between requirements and tests were loose, which led to difficulties in managing the quality of the system under development.

Requirements were not fulfilling all the typical quality criteria, such as completeness, verifiability and unambiguity [10,11], and test cases were also suffering from similar issues. No professional requirements and testing tools were in use, which allowed errors to manifest easily in the documentation.

Change management of the requirements, including the impact of changes to test cases, was non-systematic. This led to situations where the project personnel did not always know the latest requirements or the impact of changes to design and testing.

The current state of the processes is summarized in Section V of this paper. A development project was established to eliminate the above mentioned issues: to improve the quality of our project work and to create visibility to the R&D projects in general.

III. Description of the development project

An internal development project was established to create and implement R&D specific requirements and testing processes. Before starting the development project, a business case was calculated showing that the planned actions and the return on investments are beneficial for the organization. The improvement actions of the development project were:

- development of requirements and testing processes;
- identification of needed roles;
- training the responsible people;
- taking supporting tools into use;
- and piloting the processes, roles and practices in a selected R&D project.

The volume of the development project was approximately 2 000 hours. The duration of the project was one year. The project team consisted of R&D, service, marketing and operations representatives in order to accommodate the needs of different stakeholders. The main phases of the project are illustrated in Fig. 1.

We started the development project by performing a self assessment regarding requirements and testing processes, activities and practices. The self assessment was carried out by interviewing the product development team. The interview was conducted by using a questionnaire based on two templates, found in [1] and [12]. The template in [1] consisted of 20 questions in the area of requirements engineering. The Test Process Improvement template in [12] consisted of 20 questions concerning testing activities.

In Phase 2, processes for requirements engineering and testing were created based on the findings of the self assessment. Roles in the processes were also identified along with the needs for the requirements and testing tools, and the tools were selected.

In order to make the processes work in an effective way, the employees were trained to ensure competence in each role. This was carried out in Phase 3. Several methods were used, such as courses and workshops. In addition, installation and configuration of the requirements and testing tools according to the new processes also took place in this phase.

In Phase 4, the new processes, tools and roles were brought to practice. The piloting took place in an R&D project, where the goal was to create an intelligent valve controller. The pilot project took use of the new requirements and testing processes from the beginning of the project. New roles were also introduced in the pilot project organization. The selected requirements and testing tools were used to support the work. The amount of requirements in the project was approximately 2 000.

Part of the R&D work in the pilot project was outsourced. Consequently, relevant processes, tools and roles were implemented also in the suppliers' organizations. The required training was arranged for the subcontractors.

In Phase 5, the development project was brought to an end and the same self assessment as in Phase 1 was carried out. In addition, the lessons learned were identified.

IV. Lessons learned

A. *Integrated requirements and testing processes*

The main lesson learned is that the concurrent development of the requirements and testing processes ensures the information flow between requirements and testing processes throughout the entire R&D project life cycle. Well integrated processes form a basis for optimized use of resources e.g. roles and tools in our R&D projects.

We developed requirements and testing processes simultaneously to be run in parallel and to be integrated throughout the whole product development project. We wanted the information flow between the processes to be multidirectional in order to form a strong link between requirements and testing.

We followed the new requirements development, requirements management, testing development and testing management processes in our pilot project. An overview of the processes and data flows is presented in Fig. 2.

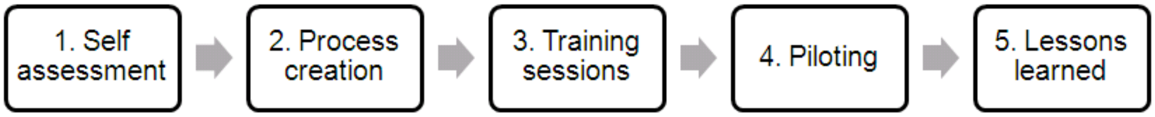

Figure 1. Phases of the development project.

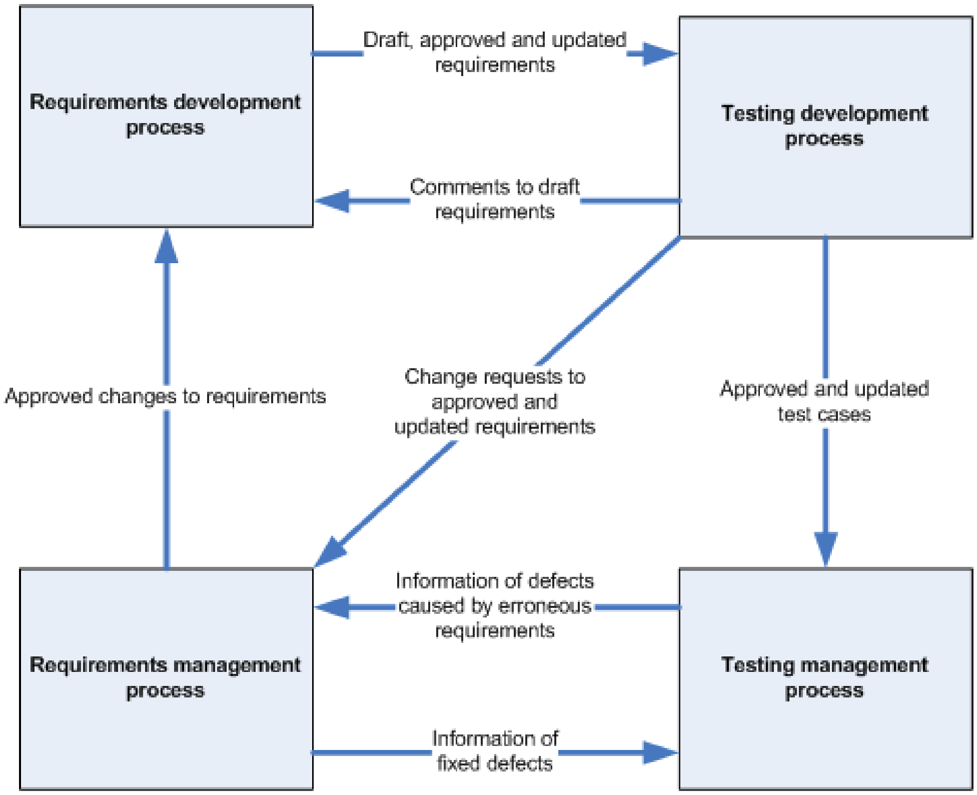
Requirements development process
Draft, approved and updated requirements
Testing development process
Comments to draft requirements
Approved changes to requirements
Change requests to approved and updated requirements
Approved and updated test cases
Information of defects caused by erroneous requirements
Requirements management process
Testing management process
Information of fixed defects

Figure 2. Relationships between requirements and testing processes.

During the requirements development process, we defined business, system and sub-system level requirements. Business level consists of main features as well as business, customer and user requirements forming a basis for the requirements development. System requirements and use cases describe the functional and non-functional requirements in the system level. The lowest and most detailed sub-system level consists of our software, firmware, hardware and mechanics requirements.

The system and sub-system level requirements were linked directly to test cases. Business level was agreed to be verified by field tests in customer sites and through system level test cases.

We gave the draft state requirements to the test planning team, enabling an early start for the test case development. Comments about the testability of the draft requirements were received directly from the testers. Based on the feedback we were able to finalize, review and approve the requirements.

We had the test cases and links between requirements and test cases reviewed by a review team before starting the test execution according to the testing development process. This revealed needs for changes to the already approved requirements.

We planned test rounds, executed the approved test cases and handled found defects as described in the testing management process. If a defect was caused by an erroneous requirement, it generated a change request to the requirement. This formed one of the feedback channels to requirements management in the later phases of the project.

We managed all change requests to the approved requirements through the requirement management process, where an impact analysis and decision making of approving or rejecting the changes took place. After the decision, all stakeholders were informed and the requirements and test cases were updated if needed.

B. *Roles in requirements and testing processes*

The main lesson learned is that defining roles to manage the integrated processes ensures that the right information is communicated to the right persons. Clear roles and responsibilities remove the risk for doing double work and ensure that all areas of product development are covered. Roles ensure that processes are performed in practice as planned.

Clear roles and responsibilities were needed to run the requirements and testing processes in parallel. Six roles related to linking requirements and testing were identified and they are presented in Table I. One person can have more than one role at the same time in smaller projects.

We separated the requirements manager and the test manager roles from the project manager's responsibilities to manage the new processes. In addition, requirements and test planners, test engineers and a defect coordinator had essential roles in the creation of the information flow illustrated in Fig. 2.

TABLE I. ROLES TO LINK REQUIREMENTS AND TESTING.

Role	Main responsibilities
Requirements manager	Planning, management and reporting. Participating in requirements and test case reviews. Managing change control meetings.
Requirements planner(s)	Creation of requirements. Participating in requirements reviews and change management meetings.
Test manager	Planning, management and reporting. Participating in requirements and test case reviews and change management meetings. Managing traceability.
Test planner(s)	Creation of the test case library and links from requirements to test cases.
Test engineer(s)	Executing tests and identifying missing or erroneous links, requirements and test cases.
Defect coordinator	Receiving information about defects from test engineers and allocating them to the right roles for fixing.

We arranged different kinds of training sessions regarding to the general processes, tools, test management, test planning, requirements management and requirements writing for the project team before the pilot project was started. The used methods were workshops with simulations, courses and hands-on training for tools and supporting materials. During the pilot project, we supported the project personnel in enhancing their abilities to work in the new roles and with the new practices. We used some of the methods also to train subcontractors.

C. *Good practices*

During the pilot project, we identified a set of good practices that helped the people responsible to perform the new processes. We also identified areas that require improvement and need to be solved in future projects.

1) *Communication*

Have requirements and test managers in the project management core team in addition to the project manager, system architect and designers. Our core team weekly meetings ensured good communication between the requirements and test management, as well as within the whole project team.

Improvement areas: We planned that the order of the design work and the execution of testing would be communicated with the requirement attributes ‘priority’ and ‘risk’. The use of these attributes was supposed to be beneficial when the rescoping of the project would be necessary. However, the priorities were not kept up-to-date, nor was the information put to use, because the benefits were not fully understood.

We also planned to indicate when the design was ready to be tested through the requirement status ‘implemented’. The use of the status was not successful due to inadequate training and lack of motivation, i.e., changing the status was perceived as additional effort among the designers. It generated failed tests because the functionality was not yet implemented. In addition, the metrics of the implementation status showed faulty progress.

These problems generated extra costs for the pilot project. The information regarding priorities and the readiness of the design was communicated through various means, for example, via email and in meetings.

2) *Metrics and visibility*

Define a variety of needed metrics in the requirement and test plans at the beginning of the project. We used metrics to follow the status of the requirements and verification processes during the pilot project. We gathered data regularly from the beginning of the project, which helped to predict changes and gave the means to react to them in due time. We presented the requirements and test metrics in weekly project meetings. This raised the visibility of the project status, which was perceived as one of the most important improvements in our lessons learned. One example of the metrics we used for monitoring the progress of the linking between requirements and tests is requirements coverage metrics, shown in Fig. 3.

Improvement areas: We followed the status of testing first through the requirements coverage metrics. These metrics gave the current status of how many requirements are covered with test case(s) and how many of them are in passed or failed states. However, it did not give us information on the quality of requirements coverage. We defined some new metrics to show how many test cases are under development and whether these test cases have been reviewed or not. Metrics showing the quality and maturity of testing needs to be developed further.

3) *Roles and responsibilities*

Agree that either the requirements or test manager is responsible for managing traces between requirements and test cases. We nominated the test manager for managing the traces. One person responsible ensured that there were no conflicting or overlapping traces and kept the test cases up-to-date in case of changes in the requirements. In addition, one person should be made responsible for managing defects. In our case, this was the role of the defect coordinator. Test engineers can allocate all found defects directly to the defect coordinator without confusion as to who the person responsible for fixing each defect is. This is important especially if subcontractors are involved in the

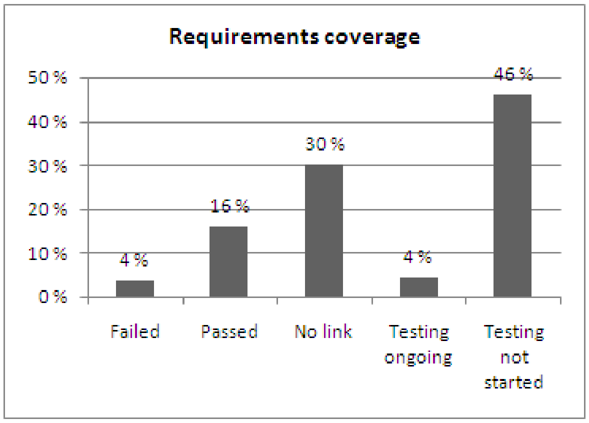

Figure 3. An example on how the requirements coverage gives visibility to the amount of links and to the test status.

project. In very large projects, more than one person may be required to take up these responsibilities.

Improvement areas: We discovered that the role of the quality assurance manager overlaps with the requirements' and test managers' roles. These roles had same responsibilities in requirements and test reviews as well as in change control board meetings. We must redefine the role and responsibilities of the quality manager compared to those of the requirements and test managers' in order to avoid double work and misunderstandings.

4) Review teams

Form cross-functional review teams. We took all viewpoints, for instance those of testing, service, operations, marketing and development into consideration and all parties were involved from the beginning of the project. It was important that each reviewer had a clear viewpoint to achieve extensive reviews and avoid double work. For example, test manager involvement in requirements reviews ensured reviewing requirements from the testability viewpoint, and vice versa.

Improvement areas: The whole review team does not need to be present in every review. Depending on the level and type of the requirements or tests, the representatives may vary. We discovered this after having some large requirements reviews in which some of the reviewers had no competence on the issue and in practice the meeting was wasting their time.

Keeping the focus of the conversation on requirements instead of discussing the design and technical details was another great challenge in the requirements reviews.

In addition, the lessons learned revealed a need for improvement in forming the test case review teams. We downsized the original cross-functional review team to the test manager and a designer due to lack of time. Test cases were reviewed, but it remained unclear, whether all viewpoints were taken into consideration. Tools should be configured to include the review process so that they show whether or not all the responsible roles have reviewed a particular item.

5) Reviews

Review requirements by using a walkthrough approach in face-to-face meetings. We ensured that the whole review team was familiar with the product concept, product scope and requirement contents by having the review meetings. Reviews were started as soon as the first requirement type was ready to be reviewed. This way, comments about the requirements were received already at the beginning. We arranged reviews systematically during the requirements development process to identify missing and erroneous requirements before giving the approved requirements to the test planners.

Improvement areas: We attempted to review the requirements and test cases individually before the face-to-face meetings but practice showed that not all reviewers were able to manage this successfully due to lack of time. Enough review time should thus be reserved in advance when planning the project schedule.

Another practice that remained unsatisfying was reviewing the requirements with subcontractors. We wrote and reviewed some of the requirements already before selecting the subcontractors, and therefore we had to revaluate them with subcontractors. Also the sub-system owners had their own reviews with subcontractors, after which the requirements were reviewed again internally. This doubled the workload.

6) Change control

Follow the change control process strictly. We handled all change requests to requirements in the change control board, which consisted of an adequate amount of different roles capable of making the decision. The change control process ensured that no changes took place without proper impact analysis and communication. The approved changes were updated and communicated to the test development and test management teams. The changes, impact analysis and decisions were well documented.

Improvement areas: Keeping the test cases up-to-date with the requirements was challenging. In the beginning, the approved changes were not made to the requirements and test cases immediately after the meeting, but were put on a task list. This resulted in test engineers using the old test cases. The changes must be made quickly both to the requirements and to the test cases to avoid unnecessary work.

Another area of improvement is handling change control practices with subcontractors. Change control meetings were internal, and separate meetings were arranged with subcontractors to discuss the changes. This caused additional work.

7) Tools

Use tools to create and manage links between requirements and test cases. Changes were effectively managed with the traceability functions in the tools. Automatic notifications after updates to requirements were sent to the test manager and to the test planners. After reading the updated requirements, it was possible to jump directly to the corresponding test cases to make the necessary changes. The tools also supported the gathering and reporting of the metrics.

Improvement areas: Initially, suppliers had access only to the requirements related to their design work. However, it was soon discovered that they also needed to know the other requirements of the project in order to understand the whole system and the design rationale.

V. RESULTS OF THE SELF ASSESSMENT

The progress in improving the quality of project work was studied by performing self assessment at the beginning and at the end of the development project, as described in Section III. The results of the self assessment that were related to linking requirements and testing are presented in Table II.

In eight out of nine areas a significant improvement was gained. The pilot project reached a good or excellent level in all of the nine areas. The initial and achieved states are explained in the following, according to [1] and [12].

TABLE II. RESULTS OF THE SELF ASSESSMENT RELATED TO LINKING REQUIREMENTS AND TESTING.

Area of improvement	Initial state	Achieved state
Requirements as basis for testing	None	Excellent
Training and experience	None	Good
Verify mutual understanding	Good	Good
Evaluate requirements quality	Weak	Excellent
Requirements traceability	Weak	Good
Change management	None	Excellent
Requirements baseline	None	Good
Test case reviews	Weak	Good
Testing metrics	Weak	Good

1) Requirements as basis for testing

Initial state: We had no direct relationship between tests and requirements.

Achieved state: We created and managed links between requirements and test cases with the help of tools. The test manager reviewed the requirements from the testability viewpoint. The test cases were based on requirements and reviewed. The testing progress was measured by requirements coverage.

2) Training and experience

Initial state: We had a little experience but no specific training in requirements engineering.

Achieved state: We had a professional requirements manager with several days of training.

3) Verify mutual understanding

Initial state = Achieved state: We created prototypes when appropriate to verify mutual understanding between different stakeholders.

4) Evaluate requirements quality

Initial state: We passed around the requirements specification in order to get feedback, but no formal process was followed.

Achieved state: We followed the review process, which was managed by the requirements manager. Requirements were reviewed by a cross-functional team, for example, from the testability viewpoint.

5) Requirements traceability

Initial state: We knew where many of the requirements came from but sources were not documented and traces were not maintained.

Achieved state: All our requirements had an identified origin and they were linked to test cases in the specified level.

6) Change management

Initial state: We had uncontrolled changes, due to which the project scope and schedule were difficult to manage.

Achieved state: We followed a change control process and used a tool to collect, store and communicate change requests. Changes were analyzed and decided upon in the change control board meetings.

7) Requirements baseline

Initial state: We had no baselining practice in use.

Achieved state: We defined an initial requirements baseline but we did not always keep it current as changes were made over time.

8) Test case reviews

Initial state: We did not follow a formal test case review process or strategy. Risk analysis was missing.

Achieved state: We had a formal review process in use. Risks were managed.

9) Testing metrics

Initial state: We did not define testing metrics in the test plan. Our test related data was not easily available.

Achieved state: We defined testing metrics in the test plan. The metrics were easily available and regularly monitored in project meetings.

Improved areas have a positive effect to the quality of the project work and product quality. In our case, linking requirements and testing increased the accuracy of planning of testing, which reflected on the improved control and estimation of project costs and schedule. Increased visibility enabled to follow the project status and react to changes and delays early. Our pilot project managed to keep its schedule with an accuracy of 5% at the time when approximately 80% of the project gates were executed.

In case of the product quality, we executed mainly requirements based testing, which gave us visibility to the testing status and product maturity. In addition, we validated the concept with customers to get feedback on the requirements and product concept at the early stages of the pilot project. The feedback gave us valuable information on how to make the user interface of the product more usable. Acceptance and feedback from customers was also received from early field tests in customer sites, which confirmed that the product meets the customer needs.

VI. DISCUSSION AND CONCLUSIONS

In this paper, we have presented how we applied a systematic approach to linking requirements and testing. The results showed that it is beneficial to perform linking in three different levels: 1) linking of processes, 2) linking of people and 3) applying good practices.

We observed that when requirements and testing processes are developed together, they form a solid basis for systematic project work as well as linking requirements and testing. Concurrent development reduces the risk of overlapping processes and roles, and it also prevents the formation of gaps in the processes. In addition, the implementation of processes, roles and the selection of supporting tools becomes easier.

To make the link between people concrete, two critical roles were identified: the role of the requirements manager and the role of the test manager. The linking of roles ensures smooth communication between the relevant parties. The most important responsibilities of requirements and test managers to enhance the communication are reviews.

Definition and follow-up of the metrics, management of traceability with the help of tools, organized reviews and establishment of the change control board were found as the most essential good practices to link requirements and testing in our case.

We were able to gain significant improvements in the pilot project. We can identify the following main factors for these significant improvements. There was a major change in the organization from line organization to matrix organization. This enabled effective and clear roles in project teams with a mandate to execute the project. The second main factor was previous experiences from overdue R&D projects. We can reliably assume that these experiences created need for improvement, which significantly reduced change resistance in the organization. The third factor was the commitment and support from the management. The fourth main factor was inviting the representatives from marketing, R&D, service and manufacturing to the development project. This again reduced change resistance. The last main factor was emphasizing the easy usability of the requirements and testing tools. This decreased the barrier to take tools into use by personnel.

Previous studies [5,13] have discovered similar types of good practices and linking of roles as were found in this case study. Our findings also emphasize the importance of concurrent development of requirement and testing processes. We were able to identify practices in areas of communication, metrics, reviews and change control in addition to the practices given in [5].

The linking of requirements and testing is one factor that helps a project manager to run a project effectively, and it enables a project to meet its planned schedule. In our case, the R&D pilot project has succeeded to keep its schedule, which has been a major improvement in comparison to the results of previous projects [14].

Areas to be improved are related to the usage of requirements attributes such as priority, risk and 'implemented' status. In addition, keeping test cases up-to-date during the requirements change process is one of the challenges. There is also a risk that the testing performed is only requirements-based testing. It is important to recognize the need of other testing techniques in different phases of an R&D project. Moreover, fluent practices with subcontractors must be developed.

In this paper we have presented how we applied a systematic approach to link requirements and testing. Currently we are interested in other factors, which may shorten the product development time and improve the quality.

ACKNOWLEDGMENT

Grateful acknowledgement is given to coworkers in Flow Control's Smart Products R&D unit. Mr. Mika Nissinen and Dr. Kari K. Hartikainen are acknowledged for the financial side of this project.

REFERENCES

[1] K.E. Wiegers, Software Requirements, 2nd ed. Microsoft Press, Redmond, WA, 2003.

[2] D. Graham, "Requirements and Testing: Seven Missing-Link Myths," IEEE Software, vol. 19, iss. 5, 2002, pp. 15-17, doi: 10.1109/MS.2002.1032845.

[3] R. Black, Pragmatic Software Testing. Wiley Publishing, Inc., Indianapolis, 2007.

[4] R.R. Young, Effective Requirements Practices. Addison-Wesley Longman Publishing Co., Inc., Boston, 2001.

[5] E.J. Uusitalo, M. Komssi, M. Kauppinen and A.M. Davis, "Linking Requirements and Testing in Practice," in Proceedings of the 16th IEEE International Requirements Engineering Conference (RE'08), 2008, pp. 265-270, doi:10.1109/RE.2008.30.

[6] D. Damian, J. Chisan, L. Vaidyanathasamy and Y. Pal, "Requirements Engineering and Downstream Software Development: Findings from a Case Study," Empirical Software Engineering, vol. 10, iss. 3, 2005, pp. 255-283, doi: 10.1007/s10664-005-1288-4.

[7] D. Damian and J. Chisan, "An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity, Quality, and Risk Management," IEEE Transactions on Software Engineering, vol. 32, iss. 7, 2006, pp. 433-453, doi:10.1109/TSE.2006.61.

[8] IEC 61508-2:2000, Functional safety of electrical / electronic / programmable electronic safety-related systems.

[9] R.G. Cooper, "Overhauling the new product process," Industrial Marketing Management, vol. 25, iss. 6, 1996, pp. 465-482, doi:10.1016/S0019-8501(96)00062-4.

[10] A.M. Davis, Just Enough Requirements Management – Where Software Development Meets Marketing. Dorset House Publishing, New York, 2005.

[11] I. Hooks and K. Farry, Customer-Centered Products. Amacom, New York, 2001.

[12] T. Koomen and M. Pol, Test Process Improvement, Addison-Wesley Longman Publishing Co., Inc. Boston, 1999

[13] S. Konrad and M. Gall, "Requirements Engineering in the Development of Large-Scale Systems," in Proceedings of the 16th IEEE International Requirements Engineering Conference (RE'08), 2008, pp. 217-222, doi:10.1109/RE.2008.31.

[14] Metso Project Portfolio Database, Internal database.