Linking Requirements and Testing in Practice

Conference Paper · September 2008

DOI: 10.1109/RE.2008.30 · Source: DBLP

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

Marko Komssi
F-Secure
17 PUBLICATIONS **279** CITATIONS
SEE PROFILE

Marjo Kauppinen
Aalto University
47 PUBLICATIONS **1,092** CITATIONS
SEE PROFILE

Linking Requirements and Testing in Practice

Eero J. Uusitalo[a], Marko Komssi[a], Marjo Kauppinen[a], Alan M. Davis[b]
[a]*Software Business and Engineering Institute,*
Helsinki University of Technology
[b]*Department of Information Systems, College of Business,*
The University of Colorado at Colorado Springs
{eero.uusitalo, marko.komssi, marjo.kauppinen}@tkk.fi, adavis@uccs.edu

Abstract

An increasing number of organizations are interested in binding requirements and testing more closely together. Based on a series of practitioner interviews conducted in five Finnish organizations, this paper presents a set of good practices that can be applied to create a stronger link between requirements engineering and testing. These practices include early tester participation particularly in requirements reviews, setting up traceability policies, taking feature requests from testers into account, and linking testing personnel with requirement owners. Due to reported hardships in implementing complete test traceability to requirements, communication links between testers and requirement owners are suggested in order to overcome the deficiencies of document links.

1. Introduction

A strong link between testing and requirements engineering (RE) can benefit both disciplines, but often this link is missing [1]. Furthermore, in our recent work with the industry, practitioners have shown interest in this topic with the goal of improving both areas of the software process. This paper presents a set of good practices that can be applied to bring RE and testing closer to each other in real product development projects, which will improve the likelihood of the resulting products satisfying customer needs.

The paper is based on interviews with practitioners as well as a literature review, both conducted during summer 2007. In the following sections, backgrounds of the interviewees and their companies are presented along with a brief summary of the research methodology used. The good practices found are then presented. Finally, analysis on the dependencies of practices is presented along with other observations and conclusions.

2. Background

The six interviewees represented five Finnish software companies, all working in different projects. As shown in Table 1, they operate in a variety of areas, and the roles of the interviewees ranged from R&D managers to testers. Four of the involved companies were product-oriented whereas company Delta's two consultants were involved in different customer-specific projects as external test engineers. Interviews

Table 1. Details of interviewees.

Company	Size (employees)	Domain	Interviewee role
Alpha	80	Enterprise security	R&D manager
Bravo	250	Network security	Test manager
Charlie	500	Sports instruments	Test manager
Delta(1)	70	Testing consultancy	Test engineer
Delta(2)	70	Testing consultancy	Test engineer
Echo	550	Computer security	Test engineer

were carried out in a semi-structured manner with two interviewers, ranging from 60 to 90 minutes in length. The interviews were recorded and transcribed.

A literature review was also conducted to determine previously suggested good practices concerning the link between requirements and testing activities.

3. Practices

This section presents discovered practices used in the industry that strengthen the link between RE and testing. The description of the practice, reported benefits and challenges are as reported by interviewees. Similar practices recommended in literature are also given. Table 2 shows how well each practice was in place, as reported by the interviewees (A to E in Table 2, as related to interviewees from Alpha to Echo in Table 1).

3.1. Early tester participation

This practice consists of involving testers in the planning stages of either the entire project or just the iteration at hand. Several interviewees reported that either the testers or the test managers (or equivalent) normally participate in this phase.

Benefits: Having testing personnel involved early ensures that

- Testing activities are properly taken into account in planning, i.e., budgeted for both in schedule and resources.
- Testers' domain and system knowledge are improved because of additional exposure to the subject.
- Requirements quality is improved.

Challenges: The general view of the interviewees was that testers are rarely available during this early stage of the project because they are assigned to other projects. In addition, the interviewees reported two other challenges, namely distributed development and the limits imposed by the applied software process.

In literature: Testers are encouraged to participate in requirement validation at an early stage according to [2]. It is even claimed in [3] that no software process can be successful unless testers are present from the beginning. One of the earliest references (1975) to the need for early tester participation appeared in [4].

3.2. Tester participation in requirements reviews

Requirement reviews were in place in all but one of the interviewees' companies. In two of the companies, testers participated in requirement reviews, whereas in the others only the test managers participated. Often the selection of specific test personnel to participate was based on their level of accumulated customer and domain knowledge. Some of the interviewees desired that individual testers participate in the reviews more often.

Benefits: Testers have a different viewpoint than requirements analysts or developers, which helps

- Surface deficiencies and omissions in requirements.
- Discover requirements that will be difficult to validate.

This results in higher quality requirements and consequently improved testability.

Challenges: It was reported that the suggestions made by testers were often in the wrong scope. In practice, this meant asking for too specific information about the features at a too early phase.

In literature: It is widely stated [1, 2, 5-7] that the testing personnel should take part in requirements reviews.

3.3. Test traceability to requirements

Four of the interviewees stated that their projects practiced some level of traceability between tests and documented requirements. However, all admitted that their traceability was incomplete, i.e., that some tests were not explicitly traced to any requirement, or some requirements were not explicitly traced to any tests.

Table 2. Requirement practices in place shown for each interviewee from Alpha to Echo.
(A solid box denotes that a practice was implemented thoroughly, a blank box denotes that it was not practiced, and a half-solid box denotes that it was implemented in a lesser form.)

Practice	A	B	C	D1	D2	E
Early tester participation	■	◧	◧	□	◧	■
Tester participation in requirements reviews	■	◧	◧	■	◧	□
Test traceability to requirements	◧	◧	◧	□	◧	□
Linking testers with requirement owners	◧	■	■	■	◧	■
Requirement suggestions by testers	◧	◧	□	■	□	■

Benefits: Traceability was reported to help in many ways:

- Improved test coverage, as it is more apparent which requirements the test cases cover.
- Efficiency of change management is improved because requirements change can be traced to appropriate test cases.
- Error removal is more efficient because testers can easily change their viewpoints between tests and requirements.

Challenges: Those interviewed declared that traceability between requirements and tests was rarely maintained in practice. This was caused primarily by failure to update traces when requirements change, due to stringent enforcement of schedules and budgets. In most cases, interviewees longed for better tool support for traceability. Some also noted that poor quality of requirements was a hindrance to maintaining the traces, since there is no guarantee how well the requirements covered the actual functionality of the product.

In literature: Setting traceability policies into place is recommended in several sources [5, 7-9]. Lindstrom [10] declared that failure to trace tests to requirements is one of the five most effective ways to destroy a project. Davis [11] reported that in one case a project manager refused to share the requirements document with testers for fear of prejudicing the test team.

3.4. Linking testers with requirement owners

All the interviewees agreed that requirements documents and test plans do not always provide enough information to enable testers to perform testing accurately and thoroughly. Without exception, interviewees stated that testers needed a way to retrieve additional information about the requirements. This means implementing person-to-person links between testers and requirement owners. In most of the interviewees' projects, testers had a direct link to requirement engineers when necessary. In two cases, this link was to the test manager, who had a greater understanding of the requirements than the testers. The test manager would consequently query requirements owners if needed. It is notable that when the link was to the test manager, the communication was perceived to be very intense at times.

Benefits: Open communication between testers and requirements writers during the development and testing process results in:

- Reduced assumptions made by testers.
- The ability to progress in spite of lower quality requirements.
- Increased reliability of test results and subsequent products.

Challenges: The authoritative source of information for requirements may be unavailable or overloaded, particularly toward the end of iterations. Individuals with deep knowledge of users and customers tend to work on requirements teams and move from project to project. Caution must be exercised to ensure the source of information used is indeed reliable.

In literature: The practice is promoted in [6], where its usefulness is underlined particularly when dealing with error handling. In addition, agile methods such as extreme programming emphasize the importance of communication between people working on the same artifacts [12]. Earlier instantiations of extreme programming [13] suggested that a customer (who by definition should be an expert on requirements) be present during the development process to respond to such inquiries from designers, coders, and testers. Of course, it is unlikely that a single individual could serve as an oracle for all the naturally occurring opposing stakeholder viewpoints. It is also unlikely that a customer could afford to sacrifice a truly important decision maker for such a passive task. In practice today, very few development organizations actually succeed in achieving this [14].

3.5. Requirement suggestions by testers

This practice was of great importance for the testing process of two of the interviewees' projects. The testers asked for features which helped in automating and performing tests, from low-level features such as command-line input possibilities to simulators for input data. Another two of the interviewees stated that testers are free to ask for requirements that will help in performing testing.

Benefits: Testability and possibilities for test automation are improved, resulting in reduced testing effort needed. With less effort required, it is more likely that testing is completed both on time and within the planned scope and scale.

Challenges: There were two challenges present: (1) As the requirements process progresses, requirements writers become more and more reluctant to add new features. Thus, the testers need to ask for features they think are important early in the process, and (2) when requirements triage [15] occurs, more weight is generally given to the opinions of customers and users than to those of the testers, resulting in a high likelihood that these features will be given lower priority.

In literature: Test automation can save development time and make testing more effective, provided that it is adequately planned and used in a proper way. [6] In the past few years, the agile community [16] expanded their processes to include the idea that tests could actually suffice as requirements themselves. In such cases, the difference between testers and requirements writers becomes somewhat blurred.

3.6. Other practices

The interviewees were asked if the link of requirements and testing was more important on the level of linking people versus linking documents or artifacts together. Most were reluctant to choose one over the other. Still, one interviewee considered linking people of greater importance to his company's process. The remaining interviewees stated that the linking of the documents is essential as it builds the foundation for successful testing, but still saw the link of people as very important.

All of the interviewees emphasized the importance for testers to achieve superior knowledge of the product and its application domain. This assists testers in improved understanding of requirements and thus aids both in designing test cases and performing testing.

Some of the interviewees saw software process improvement, especially as applied to the requirements process, as a great enabler of testing. This demonstrates the interdependency between requirements and testing process. Without a proper requirements process, tester participation is difficult and the testing process improvement is inhibited.

4. Lessons learned

The practices discussed in this paper are not independent. Based on an analysis of the interviews, this section presents the interdependencies of the practices presented and discussion on the linking of documents versus the linking of people.

4.1. Interdependencies of practices

Figure 1 shows the interdependencies of the practices. Especially the first practice presented, early tester involvement, can be seen as an enabler for two of the other practices:

- Participation in requirements reviews. Up to a point, the earlier requirements are reviewed the more effective the reviews should be.
- Linking testers with requirement owners. Requirements analysts will typically perform most of their work early in the iterations, so chances for interaction between them and the testers are increased. Any face-to-face meeting should build trust between the parties, removing obstacles from communication.

The second practice, tester participation in requirements reviews,

- Facilitates requirements suggestions by testers. An involved tester is more likely to provide constructive feedback. Better understanding of requirements due to additional exposure to them can help identify testability issues.
- Enables linking testers with requirement owners. When participating in the reviews, testers will interact with requirement analysts, leading to increased trust as in the first practice.

The third practice, test traceability to requirements, is co-dependent on the fourth practice,

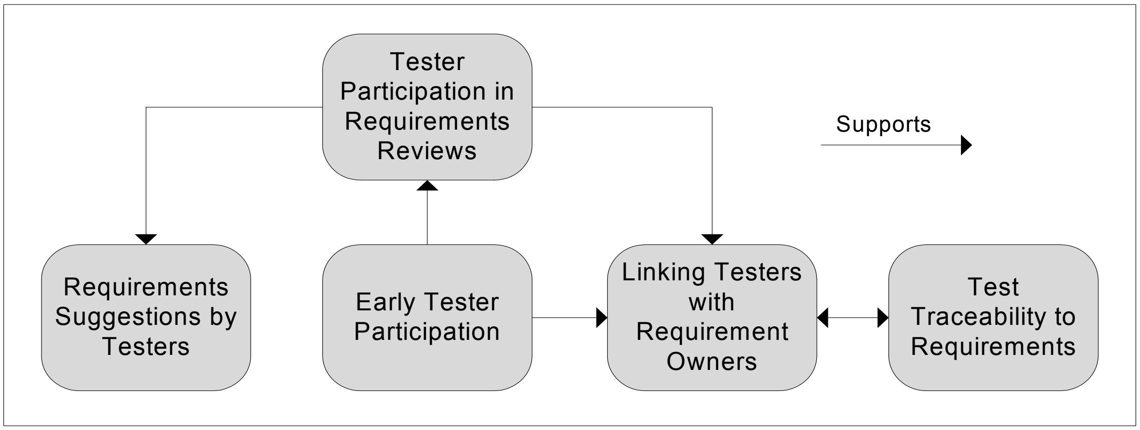
Tester Participation in Requirements Reviews
Supports
Requirements Suggestions by Testers
Early Tester Participation
Linking Testers with Requirement Owners
Test Traceability to Requirements

Figure 1. Interdependencies of practices.

linking testers with requirement owners:

- Having traceability lessens the need to communicate.
- Open communication channels lessen the need to have traceability.
- When requirements state the “owner” of the requirement, testers have the means to find the correct person to consult with when they face an ambiguous test case and/or an ambiguous requirement.

4.2. Linking people vs. linking documents

The two practices, “test traceability to requirements”, and “linking testers to requirement owners”, can be thought of as practices that link documents and people, respectively. Both share one common goal from the point of view of testing: Providing testers with an authoritative source of up-to-date, sufficient information about requirements in a timely manner.

The application of the two practices is somewhat dependant on the software process in use: unsurprisingly, an agile process emphasized the importance of the people link. However, those using more traditional processes (i.e. waterfall or iterative) still stated that the people link is an important addition to the link of documents.

Both of the links have their pitfalls: The people link can get overloaded in situations where the information needed is concentrated on one person only, or if a middle man exists in the link. On the other hand, traceability is perceived to be too cumbersome to be implemented completely, mostly due to lack of tool support and the inefficiency of change management processes – even apparent error corrections take time to officially clear the process, so the actual and correct requirements information is often not in the requirements documents.

It appears that the links of documents and people hardly make each other redundant and that they should be applied in support of each other. The problems arising from incomplete traceability can be alleviated by linking people together, thus shifting some of the responsibility for information flow from the level of documentation to the level of communication.

5. Conclusions

One of the major factors in performing successful testing was accumulated tester knowledge about the product and the problem domain. Testers will understand higher level requirements better and need less additional information, possibly diminishing the reliance on the link between testing and requirements.

The most important function of the link of requirements engineering and testing is to ensure the flow of information about requirements to the testing process. Most other benefits gained from the link can ultimately be derived to the flow of information: higher quality requirements mean that tacit knowledge is made explicit and available. Especially the two practices, test traceability to requirements, and linking testers with requirement owners, focus on the information flow. Traceability links information on the level of documents, often providing the first place for testers to look into, while personal links mostly provide a supplementary means to fill up any gaps in the information.

6. Future work

Five out of six interviewees were primarily focused on testing in their daily jobs. Most interviewees reported deficiencies in the requirements process to be a large hindrance to linking requirements and testing together. Future research is needed to find out what issues requirement engineers and developers see in binding requirements and testing together and how they think this can be accomplished.

Additionally, further analyzing how well the presented practices work in relation to various variables could be interesting. One such example is whether encouraging tester participation in requirement reviews is more useful when dealing with specific kinds of requirements. Furthermore, return on investment (ROI) analysis could be helpful in assessing the suitability of the practices for a given situation.

Bringing the roles of requirement analysts and testers closer to each other is an interesting idea. The skill set of a tester can be useful in constructing requirements that are unambiguous and testable, whereas the domain knowledge possessed by the analyst is invaluable when testing is performed. A total integration of the roles may be unrealistic or even unnecessary, but there are indeed promising synergies to exploit in the roles of testers and requirements analysts.

7. References

[1] D. Graham, "Requirements and Testing: Seven Missing-Link Myths," *IEEE Software,* vol. 19, pp. 15-17, 2002.

[2] R. R. Young, *Effective Requirements Practices.* Addison-Wesley, Boston, MA, USA, 2001.

[3] M. Ramachandran, "Requirements-Driven Software Test: A Process-Oriented Approach," *ACM SIGSOFT Software Engineering Notes,* vol. 21, pp. 66-70, 1996.

[4] J. B. Goodenough and S. L. Gerhart, "Toward a Theory of Test Data Selection," *ACM SIGPLAN Notices,* vol. 10, pp. 493-510, 1975.

[5] R. Black, *Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional.* Wiley, New York, NY, USA, 2007.

[6] C. Kaner, J. Bach and B. Pettichord, *Lessons Learned in Software Testing: A Context-Driven Approach.* Wiley, New York , 2002.

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

[8] R. Stevens, P. Brook, K. Jackson, S. Arnold, *Systems Engineering: Coping with Complexity.* Prentice Hall, Great Britain, 1998.

[9] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide.* Wiley, Chichester, 1997.

[10] D. Lindstrom, "Five Ways to Destroy a Development Project," *IEEE Software,* vol. 10, pp. 55-58, 1993.

[11] A. M. Davis, *Just Enough Requirements Management: Where Software Development Meets Marketing.* New York, NY, USA: Dorset House Publishing Co., Inc., 2005,

[12] A. Cockburn, *Agile Software Development: The Cooperative Game,* 2nd ed., Addison-Wesley, 2002.

[13] K. Beck, *Extreme Programming Explained: Embrace Change.* Upper Saddle River, NJ, USA, Addison-Wesley, 2000,

[14] L. Cao and B. Ramesh, "Agile Requirements Engineering Practices: An Empirical Study," *IEEE Software,* vol. 25, pp. 60-67, 2008.

[15] A. M. Davis, "The Art of Requirements Triage," *Computer,* vol. 36, pp. 42-49, 2003.

[16] S. W. Ambler, *Agile Modeling.* Wiley, 2002.