Using Defect Taxonomies for Requirements Validation in Industrial Projects

Michael Felderer
University of Innsbruck & QE LaB Business Services
Innsbruck, Austria
michael.felderer@uibk.ac.at

Armin Beer
Beer Test Consulting
Baden, Austria
info@arminbeer.at

Abstract—Quality of requirements is of great importance for the software development lifecycle as it influences all steps of software development. To ensure various quality attributes, suitable requirements validation techniques such as reviews or testing are essential. In this paper, we show how defect taxonomies can improve requirements reviews and testing. We point out how defect taxonomies can be seamlessly integrated into the requirements engineering process and discuss requirements validation with defect taxonomies as well as its benefits and the lessons learned with reference to industrial projects of a public health insurance institution where this approach has been successfully applied.

Index Terms—Requirements validation; defect taxonomies; requirements quality; requirements review; requirements-based testing; test management

I. INTRODUCTION AND INDUSTRIAL CONTEXT

Requirements engineering is the initial part of a software development process, and all subsequent steps are influenced by the requirements, making the quality of the latter an important factor for the overall quality of the developed system [1]. Low quality requirements may not only affect other requirements but also later phases and can cause follow-up defects in design, implementation and testing. For instance, incomplete requirements often lead to conflicting interpretations. The developers fill the resulting gaps in the requirements specification with their own ideas, while the testers might have a completely different view. As a result, the tests do not match the implementation and neither of them reflects the original ideas of the analyst. Thus, validation techniques applied to guarantee completeness of requirements as well as other requirements quality attributes such as comprehensibility, traceability or verifiability are key issues in the software development lifecycle. As many problems in the software development lifecycle result from low quality requirements, requirements validation techniques are one of the most effective techniques for ensuring successful development.

Although several validation techniques for requirements quality assurance such as requirements reviews or testing are available, there is still potential for improvement [1]. As we observed in industrial applications, defect taxonomies, i.e. hierarchical classifications of faults and failures, are a promising technique to improve requirements validation. On the basis of projects from the public health insurance domain we show how defect taxonomies can improve requirements review and testing. In practice, most defect taxonomies are only used for the a-posteriori allocation of resources to prioritize failures for debugging purposes or release decisions. But the potential of these taxonomies to validate requirements a-priori has so far remained unexploited.

In the public health insurance institution where requirements validation with defect taxonomies has been applied, a standard development process consisting of the phases of (1) analysis and design, (2) development, (3) testing, and (4) operating and maintenance, is in place. All projects follow an iterative and incremental development process performing phases (1), (2) and (3) in each iteration and incorporating the results from one release in the definition of the next release. A requirements and test management group supported by external consultants provides services for the application and improvement of tools and methods in all projects. This group has recently used defect taxonomies for requirements validation in several projects. Initially, defect taxonomies were only applied to improve requirements-based testing in order to design more effective tests in a systematic way [3]. It was observed that about 30% to 50% of the defects detected after the system test of the first iteration were related to anomalies in the system requirements specification (SRS). We therefore also began to apply defect taxonomies to the review of requirements and have developed a method that integrates defect taxonomies into the requirements engineering (RE) process with the aim of improving the requirements review and testing. We present this method and its preconditions for the established RE process together with the experiences we have gained by applying it in industrial projects within a public health insurance institution.

In this paper, we demonstrate the method on the basis of a specific project, named project A, to which it has been successfully applied. In project A, a web application was developed for refunding invoices of medical care and managing these cases. The project consists of 250 requirements, 45 use cases and 100 business rules. The development time is about two and a half years, it has 5 iterations and a project staff of about 20. About 60% of the budget is dedicated to the analysis/design and testing phases.

This paper is structured as follows. In the next section we present the basic concepts of requirements validation and give an overview of the process of RE with defect taxonomies. In Section III we discuss the creation and linkage of defect taxonomies and requirements in our approach, which is the
precondition for requirements validation. On that basis, we show in Section IV how requirements are reviewed and tested using defect taxonomies. Finally, in Section V we summarize and present the lessons learned.

II. BASIC CONCEPTS AND PROCESS OVERVIEW

In this section, we present the basic concepts of requirements validation and give an overview of the process of requirements engineering with defect taxonomies.

A. Requirements Quality and Validation

The quality of requirements is determined by several attributes (also called criteria). Some key requirements quality attributes defined by Denger and Olsson [1] based on IEEE Standard 830-1998 [4] are completeness, comprehensibility, ranked for importance, right level of detail, traceability and verifiability. These quality attributes can be applied to individual requirements and to requirements documents.

Requirements validation comprises activities designed to secure the different quality attributes and includes verification activities. As there is considerable disagreement over the meaning of the terms “verification” and “validation” in the literature [5] and as these activities are almost inseparable in practice, we do not explicitly distinguish them. Instead we follow most RE literature sources and use the term “validation” to denote both validation and verification activities, i.e. the whole process of quality assurance for requirements [6]. Examples of good practice for requirements validation techniques are reviews, prototyping, specification of formal models and testing (see [7] and [8]). Validation techniques can be divided into one of two classes: constructive and analytical approaches [1]. Constructive approaches such as prototyping or specification of formal models ensure quality during the creation of requirements. Analytical approaches such as reviews and testing assess the requirements specification to check whether its requirements fulfill the quality criteria specified. In this paper, we only consider analytical requirements-validation techniques which check the quality of specified requirements with defect taxonomies.

Requirements reviews are manual processes designed to statically check the requirements specification for anomalies [8]. Reviews are characterized by the roles involved, the process, the reading technique and the quality criteria for the requirements [6].

Requirements-based testing is the process of planning, designing and executing test cases to dynamically validate whether the system fulfills its specification [1].

B. Requirements Engineering and Defect Taxonomies

In this section, we give an overview of the requirements engineering process steps specific for creating defect taxonomies and applying them for requirements validation. Fig. 1 illustrates the actively involved roles as well as the process steps with their input and output. The roles involved are analyst, domain expert and test manager. The analyst is responsible for eliciting, specifying, analyzing and reviewing the requirements. The domain expert has specific knowledge about the requirements, testing and possible defects in the problem domain, which is the basis for specifying requirements, creating defect taxonomies and linking them. A test manager is responsible for planning and controlling test activities. Besides the roles actively involved in the requirements engineering process, other roles also require high quality requirements. A developer and a tester are the consumers of the requirements specification and are interested in good quality in order to promote efficient coding and testing. The project manager is interested in the execution of the project on time and within budget and rigorous planning of releases in spite of iterative incremental development.

The process consists of five steps, where the highlighted process steps 4 and 5 are the validation steps using defect taxonomies.

In Step 1, the requirements are elicited, specified, and analyzed by an analyst and a domain expert. Input to this process step is the project description, including the project goals. We do not restrict the applied elicitation, specification and analysis techniques but require prioritized requirements together with additional requirements artifacts such as use cases, business processes, business rules, and GUIs as the output.

In Step 2, a product-specific defect taxonomy is created by a domain expert and a test manager. It is based on a generic defect taxonomy and consists of a hierarchy of defect classes, where each leaf defect category has an assigned severity value.

In Step 3, the requirements are linked to defect categories by a domain expert and a test manager. The defect categories assigned to a requirement reflect its possible defect sources and manifestations.

In Step 4, requirements are reviewed by the analyst or test manager checking quality criteria on the basis of defect taxonomies, requirements are improved and detected anomalies are listed.

In Step 5, requirements tests are planned by the test manager taking the linked defect categories into account. As a result, a test strategy which allows the definition of effective requirements tests as well as a detailed test effort estimation are created.

Details of process steps 1, 2 and 3 are discussed in Section III. Steps 4 and 5, which are highlighted in Fig. 1, address the
validation of requirements based on defect taxonomies. In Section IV we discuss these validation steps, i.e. review and testing based on defect taxonomies, in detail and highlight the benefits of using these taxonomies for requirements validation.

III. CREATING AND LINKING DEFECT TAXONOMIES AND REQUIREMENTS

In this section, we present steps 1, 2 and 3 for the integration of defect taxonomies into the requirements engineering process, i.e. the preliminary elicitation, specification and analysis of requirements, the creation of a product-specific defect taxonomy as well as the linkage of requirements and defect taxonomies.

A. Elicitation, Specification and Analysis of Requirements

Elicitation, specification and analysis are standard activities of requirements engineering [7] and also constitute the initial step of our process. Due to space limitations and because our approach is not bound to specific techniques, we do not discuss elicitation, specification and analysis activities in detail. We only present the necessary preconditions for creating defect taxonomies and linking them to requirements.

In our context, requirements are elicited, specified and analyzed by the analyst in cooperation with the domain experts on the basis of the project definition. The requirements are developed in several steps. Taking the evolving knowledge of domain experts and analysts into account, Fig. 2 shows the layers of requirements development and the created artifacts. The user requirements represented as business processes are defined on the basis of the vision and scope specified in the project definition. Finally, the system requirements are defined in the system requirements specification (SRS), which contains (functional and non-functional) requirements, use cases, business rules and graphical user interface (GUI) definitions. The SRS forms the basis for a design resulting in the system architecture which is outside the scope of this paper.

As a result of the elicitation, specification and analysis activities, the subsequent process steps demand requirements with assigned priority values and requirements artifacts such as use cases, business rules and user interface definitions. These preconditions are met by almost all established requirements development approaches, as the cost-saving potential of prioritized requirements driving requirements review and testing has been recognized [9]. Defect taxonomies and the validation based on them as presented in this paper can therefore be integrated in most requirements engineering processes.

B. Creation of a Product-Specific Defect Taxonomy

In this steps, a product-specific defect taxonomy is created on the basis of generic top-level categories which are then further refined to concrete low-level defect categories. The part of the defect taxonomy of project A relevant for the requirements of Table I is shown in Table II. This defect taxonomy is created on the basis of the vision and scope specified in the project definition. Finally, the system requirements are defined in the system requirements specification (SRS), which contains (functional and non-functional) requirements, use cases, business rules and graphical user interface (GUI) definitions. The SRS forms the basis for a design resulting in the system architecture which is outside the scope of this paper.

As a result of the elicitation, specification and analysis activities, the subsequent process steps demand requirements with assigned priority values and requirements artifacts such as use cases, business rules and user interface definitions. These preconditions are met by almost all established requirements development approaches, as the cost-saving potential of prioritized requirements driving requirements review and testing has been recognized [9]. Defect taxonomies and the validation based on them as presented in this paper can therefore be integrated in most requirements engineering processes.

B. Creation of a Product-Specific Defect Taxonomy

In this steps, a product-specific defect taxonomy is created on the basis of generic top-level categories which are then further refined to concrete low-level defect categories. The part of the defect taxonomy of project A relevant for the requirements of Table I is shown in Table II. This defect taxonomy already contains the assigned requirements and test design techniques (see Table III for their identifiers) which are added in later process steps but are - due to space limitations - already printed in this table. The top-level categories functionality, data, interfaces, logic and performance of this defect taxonomy are based on the types of the IEEE Standard Classification for Software Anomalies [10]. Each low-level defect category (DC) has an identifier, a description and a
severity value. Our experience of using defect taxonomies in several projects showed, that 4 to 9 sub-categories on each level are convenient and manageable. As our experience of creating defect taxonomies in previous projects showed [7], other defect taxonomies like the Beizer taxonomy [11] are also appropriate top-level categories. When creating defect taxonomies, defect data of completed projects and the feedback of affected roles such as developers or testers should be considered.

### TABLE II. DEFECT TAXONOMY OF PROJECT A WITH ASSIGNED REQUIREMENTS AND TEST TECHNIQUES

<table>
<thead>
<tr>
<th>Top-Level Categories</th>
<th>DC Description</th>
<th>Severity</th>
<th>Assigned Requirements</th>
<th>Test Design Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality (Process, GUI, Navigation etc.)</td>
<td>1</td>
<td>Erroneous configuration of data display</td>
<td>critical</td>
<td>REQ_0011</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>GUI navigation, erroneous display of data</td>
<td>major</td>
<td>REQ_0012, REQ_0024</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Insufficient access to database</td>
<td>critical</td>
<td>REQ_0017, REQ_0062</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>Web Browser interaction</td>
<td>normal</td>
<td>REQ_0123</td>
</tr>
<tr>
<td>Data (Definition, Access, Processing)</td>
<td>1</td>
<td>Error in invoicing</td>
<td>major</td>
<td>REQ_0033</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Error in accessing data in invoice</td>
<td>critical</td>
<td>REQ_0024</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Obsolete data</td>
<td>major</td>
<td>REQ_0073</td>
</tr>
<tr>
<td>Interfaces</td>
<td>1</td>
<td>Error in enterprise business</td>
<td>major</td>
<td>REQ_0061</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Error in interface to booking component or SAP</td>
<td>critical</td>
<td>REQ_0005</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Error in interface to business line of bank</td>
<td>major</td>
<td>REQ_0140</td>
</tr>
<tr>
<td>Logic (Evaluation of business rules, Algorithms)</td>
<td>1</td>
<td>Error in calculating the status of service recipients (insurers)</td>
<td>major</td>
<td>REQ_0031, REQ_0018, REQ_0045</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Error in calculation of the refunding of medical treatments and therapies</td>
<td>major</td>
<td>REQ_0111</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Error in checking of invoices</td>
<td>critical</td>
<td>REQ_0024, REQ_0115</td>
</tr>
<tr>
<td>Performance (Throughput, Load, Response time)</td>
<td>1</td>
<td>Response time insufficient</td>
<td>major</td>
<td>REQ_0007</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Throughput insufficient</td>
<td>critical</td>
<td></td>
</tr>
<tr>
<td>Undefined</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C. Linkage of Requirements and Defect Categories

The test manager assigns the prioritized requirements to defect categories taking the experience of domain experts into account. The assignment of the requirements in Table I to defect categories is shown in Table II. For instance, REQ_0024 (recording of detailed invoice) is assigned to the defect categories F2, D2 and L3. There is an assignment to (1) F2, because the business rule BR_0020 defines that “entry fields of debit and credit can be changed only if the invoice is in an in-work state”, to (2) D2 taking potential failures of storing invoices in the database into account, and to (3) L3, because the correctness of the calculated amount of money is an important issue.

IV. REQUIREMENTS VALIDATION WITH DEFECT TAXONOMIES

In this section we present how defect taxonomies are used for requirements reviewing and testing and what the benefits of this approach are.

A. Defect Taxonomy Based Review of Requirements

In this step, defect taxonomies are used to review requirements. In the studied public health insurance institute a review process based on IEEE 1028 [12] is in place in which requirements are inspected by analysts, domain experts, test managers, developers and testers. The defect taxonomy based review complements the established review by additionally checking requirements quality criteria taking the defect categories and their linkage to requirements into account. The use of defect taxonomies allows additional anomalies to be detected compared to the standard review process based on IEEE 1028.

With defect taxonomies the following requirements quality criteria can additionally be reviewed for individual requirements, requirements artifacts or the complete SRS utilizing the assignment of weighted defect categories and requirements:

- **Completeness**: If all relevant aspects are considered in the SRS, then at least one requirement should be assigned to each defect category.
- **Ranked for importance**: The severity value of a defect category can be used to check the importance, i.e. the priority, of requirements assigned to the defect category. As a result the priority of a requirement may be enforced or revised.
- **Verifiability**: Via defect categories, requirements are linked to appropriate test design techniques (see Section IV.B for details). This enables the design of effective requirements tests and checks of requirements testability.
- **Traceability**: Via defect categories, requirements and the assigned requirements artifacts like use cases and business rules have direct links to tests and failures.
- **Comprehensibility**: The assignment of defect categories to requirements enhances the understandability of the latter, as defect categories provide examples of typical faults of the respective requirements.
- **Right level of detail**: If assigning appropriate defect categories to a requirement is difficult because it is impossible to assign defect categories at all or too many defect categories fit, the requirement may not be defined at the right level of detail.

In the following paragraphs, we illustrate these review checks by anomalies identified in the SRS based on the defect taxonomy and its links shown in Table II.

**Completeness.** Because DC “P2” cannot be linked to a requirement of the SRS, the analyst becomes aware that a requirement for the throughput definition is missing.

**Ranked for importance.** REQ_0024 has a “high” priority which is confirmed by the severity values of the assigned defect categories (“major” for F2, “critical” for D2 and “critical” for L3). As a consequence, the underlying use case USC_create_edit_invoice and business rules BR_0020 have to be reviewed and tested with great care. In the case of REQ_0111, however, “medium” priority was assigned, but the DC D2 with “critical” severity was selected. As a consequence, the priority of the requirement may be adapted or its test depth may be increased. The latter was performed in the project.

**Verifiability.** In REQ_0024, the business rule used to calculate the amount of accounting was specified in natural language, leading to errors in coding and test design. Because DC L3...
(“critical” severity) was assigned, the testability of the requirement had to be improved by specifying decision tables. The selection of the test design technique TD4 (decision tables) also resulted in an improvement of the specification.

Traceability. All failures of the defect categories assigned to REQ_0024 can be traced back from the bug description in Bugzilla to the test cases which failed during system testing and to the relevant use case description and business rules.

Comprehensibility. The complexity of project A with its interfaces to external services and databases impedes the comprehensibility of the relationship between the different functions. In the use case descriptions connected to REQ_0024, for example, the alternative scenarios with error messages are more comprehensible when a defect category is considered.

Right level of detail. For the assignment of requirements to defect categories, the description of the requirement must be detailed enough. For example, REQ_0024 has to specify the calculation of the VAT of different countries. The REQ_0007 is not quantified to generate a load test definition.

The anomalies presented in the previous paragraphs were not identified during the established standard review process and show the benefit of our defect taxonomy based review technique to identify additional anomalies.

B. Defect Taxonomy Based Testing of Requirements

In this steps defect taxonomies are used to support all phases of testing requirements, i.e. test planning, design, execution and evaluation, which are in place at the studied public health insurance institution.

Test Planning. In the test planning phase the test manager defines a test plan describing the scope, approach, resources, and schedule of the intended test activities as well as a test strategy describing the applied test approaches. In our approach the test strategy is based on a list of test design techniques shown in Table III. These test design techniques are assigned to defect categories taking their focus of finding specific types of defects into account (see Table II).

<table>
<thead>
<tr>
<th>Table II. Test Design Technique Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Each test technique of Table III has three test strengths. The test strength refines a testing technique, e.g. by coverage criteria and is determined by the priority of requirements and the severity of defect categories. The test plan is based on a test effort estimation. How accurate the estimation of the test effort is, depends on the reviewed requirements artifacts. If only the project definition is available, the test effort is estimated by taking the experience with former projects into account. A detailed estimation requires requirements artifacts, the priority of the requirements, the severity of the defect categories and the selected test design technique. An example is given in Table IV. The use case USC_create_edit_invoice is linked to requirements of high and medium priority and defect categories of critical severity. Test strength 2 is consequently assigned. The number of test cases is calculated by designing abstract test cases. For instance, with the test design techniques TA1, TD4 and TD6 listed in Table III and taking the number of main and alternative branches of the use cases into account, the number of estimated test cases is 100. Thus, our approach supports a detailed test effort estimation enabling precise test planning.

<table>
<thead>
<tr>
<th>Table IV. Estimation of the Number of Test Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>USC_display_invoices</td>
</tr>
<tr>
<td>USC_create_edit_invoice</td>
</tr>
<tr>
<td>USC_interfaces</td>
</tr>
<tr>
<td>Performance</td>
</tr>
</tbody>
</table>

Test Design. In the test design phase test cases are derived from the defect taxonomy based test strategy fixed before, i.e. the linkage between weighted defect categories, prioritized requirements, and test design techniques is exploited to derive requirements tests from requirements artifacts. The defect taxonomy based test strategy allows the design of more effective tests, i.e. tests which find more failures on average, and reduces the total number of test [3]. In addition, the overall number of tests with severity major, critical or blocker already identified during testing increases.

Test Execution. As the number of test cases is reduced, the overall test execution time decreases. Additionally, the test cases can be prioritized based on the requirements priority and the severity of the defect category if test execution resources are limited. Experience gained during the execution of tests and the evaluation of failures detected by these tests, are used to adapt the defect taxonomy and to optimize the number of test cases of the next releases and for regression testing purposes.

Test Evaluation. To check the quality of the system and the test progress not only the ratio of the passed tests to the overall number of executed tests but also the failures and their severity have to be considered. As the failures are traceable to defect categories the severity values of failures which are assigned by testers and have varying accuracy, can be checked and adapted via the severity value of its assigned defect category. This enables more precise statements about the quality of a release and decisions whether additional hotfixes or iterations are needed.
V. SUMMARY AND LESSONS LEARNED

In this paper, we showed by application in industrial projects how defect taxonomies can be used to validate requirements. Based on (1) prioritized requirements, (2) a product-specific defect taxonomy and (3) links between them, requirements reviewing and testing can be improved. The lessons learned for requirements reviewing and testing using defect taxonomies are summarized in the following paragraphs.

Prerequisites to RE. Defect taxonomy based requirements validation demands requirements with assigned priority values and requirements artifacts like use cases and business rules as result of the initial requirements elicitation, specification and analysis activities. These preconditions are met by most requirements development approaches in use and shows the versatility of our validation approach.

Defect taxonomies. Defect taxonomies are the central artifact of our validation approach and their suitability influences the quality of the results. By creating and using several defect taxonomies we learned that 4 to 9 sub-categories on each level are manageable and that each sub-category requires a clear meaning which is supported by additional exemplary defects for each sub-category. To guarantee complete, comparable and orthogonal defect taxonomies, new ones should be created on the basis of application type specific or standardized defect taxonomies such as the IEEE taxonomy. Additionally, available defect data of completed projects and the feedback of affected roles like developers or testers should be considered.

Tool support. In the studied public health insurance institution professional tools for RE, modeling, test management and defect management are used. The creation of the defect taxonomy as well as the linkage of requirements (exported from the RE tool), test design techniques and failures (exported from the defect management tool) is done with spreadsheets. This works well in our projects as spreadsheets are easy to customize, frequently used and only edited by a few persons (one analyst, one domain expert and one test manager).

Quality of requirements. With defect taxonomies especially the quality attributes completeness, ranked for importance, verifiability, traceability, comprehensibility and right level of detail can be reviewed for individual requirements, requirements artifacts or complete SRS. In project A, we found additional anomalies for all these quality attributes compared to the standard review process based on IEEE 1028 and took appropriate countermeasures to increase the quality of requirements and consequently also in the released product.

Requirements-based testing and testability. Defect taxonomies can seamlessly be integrated into a standard requirements test process and support all its phases. Test planning is supported by a detailed test effort estimation, test design by more effective tests, test execution by prioritizing tests as well as reducing their number, and test evaluation by more precise statements about the quality of a release. In addition, the number of regression tests can be optimized when software is changed or developed in several iterations. The assignment of test design techniques to requirements via defect taxonomies fosters the adaptation of requirements and requirements artifacts to improve their testability.

Defect Detection. Defect taxonomy supported requirements validation significantly reduces the number and severity of defects detected in operation. First, defect taxonomy based reviews increase the number of anomalies detected in the SRS. Second, defect taxonomy based tests are more effective and the overall number of tests with high severity increases. Finally, comparing the severity values of failures and linked defect categories provides better decisions support whether hotfixes are needed prior to release.

Costs-Benefit considerations. The main benefits of using defect taxonomies for requirements validation are (1) the increased quality of the requirements, tests and especially the released product as well as (2) the increased process quality providing decision support for the release and test process. The main costs of the approach are (1) the effort to create and maintain the defect taxonomy and its links as well as (2) the additional effort of the defect taxonomy based review. A pragmatic approach to analyze or estimate the costs and benefits is a comparison of the overall preparation and validation time with and without defect taxonomies in a specific project. We conducted this analysis procedure in project A and found that the break-even of defect taxonomy based requirements validation was reached already after the first iteration. With a growing number of iterations the time difference to the standard validation process without taxonomies and thus also the benefit of our defect taxonomy based validation approach increases further.

ACKNOWLEDGMENT
This work was supported by the project “QE LaB – Living Models for Open Systems (FFG 822740)”.

REFERENCES