DETAILED PROGRAM
1st International Workshop on Requirements and Testing (RET '14)

http://webhotel.bth.se/re14/ret/

August 26th, 2014
Karlskrona, Sweden

Co-located with IEEE International Requirements Engineering Conference (RE 2014)

08:30 **Welcome and Introductions** — Michael Felderer

08:45 **Keynote** — Magnus Ohlsson

The Agile Hangover – Handling Testable Agile Requirements

Abstract: Agile ways of working have more or less become a de facto standard the last couple of years. Still quite many of the organisations we meet still struggle with a number of problems. Instead of taking care of the real problems they switch development process. For example, they tried to run Scrum but it did not work so they turned to Kanban but that neither worked so now it is some kind of "Kanbanish". Often the organisations fail to handle the changing requirements and we also see that they miss to capture important non-functional requirements or start to test them at a very late stage. Problems with requirements management will naturally lead to test related problems. In an agile world new ways of working and new test strategies are necessary because you cannot rely on the old test levels in the V-model. One of our most important tools is communication. It is the glue that will keep your lifecycle together. If we do not understand the customers' needs, how can we solve their problems? During this presentation we will focus on how you can ease the pain of requirements elicitation through different techniques. Further the test related problems will be addressed and provide insight what makes a Scrum team able to deliver high quality. It will be about discovering testable agile requirement. Involve the customer or fail!

09:45 **Sponsor presentation** — EASE — Björn Regnell

COFFEE BREAK (10:00-10:30)

10:30 **RET Challenges and Practices** — Session chair: Elizabeth Bjarnason

10:30 *Revisiting the Challenges in Aligning RE and V&V: Experiences from the Public Sector*

Jacob Larsson and Markus Borg

Abstract: Successful coordination of Requirements Engineering and Testing (RET) is crucial in large-scale software engineering. If the activities involved in RET are not aligned, effort is inevitably wasted, and the probability of delivering high quality software products in time decreases. Previous work has identified sixteen challenges in aligning RET in a case study of six companies. However, all six case companies selected for the study are active in proprietary software engineering. In this experience report, we discuss to what extent the identified RET alignment challenges apply to the development of a large information system for managing grants from the European Union. We confirm that most of the findings from previous work also apply to the public sector, including the challenges of aligning goals within an organization, specifying high-quality requirements, and verifying quality aspects. Furthermore, we emphasize that the public sector might be impacted by shifting political power, and that several RET alignment challenges are amplified in multi-project environments.

10:40 *On the Delicate Balance between RE and Testing: Experiences from a Large Company*

Krzysztof Wnuk, Linus Ahlberg, and Johannes Persson

Abstract: Requirements Engineering and Testing are considered inseparable and together enable successful projects. One could make a statement that serious negligence of any of the two will bring catastrophic consequences for the project, delays, failures and unhappy customers. Thus, alignment and cooperation between the two are not only recommended but highly required for contemporary complex software and system development. In this paper, we share some experiences from a large company that significantly downplayed requirements engineering activities while shifting the focus to testing and QA activities. Despite that, the company's products are well received by its customers and the company remains profitable and significantly growing. We share our experiences and discuss how this unbalance may actually be a recipe for success and what contextual factors enable using this approach.

10:50 *Testers Learning Requirements*

Hans Hartmann

Abstract: When software testers talk about their job, they generally state that their task is to find defects as well as to ensure that no (major) defects exist in the main paths of business processes. They often add the comment they are doing this by checking the software (application under test) against existing requirements. Its common sense that testers should be familiar with understanding and interpreting requirements. But when it comes to creation and review of requirements testers are usually not involved. Now, taking into consideration that many errors in software development are due to improperly defined requirements, one might wonder if integrating testers directly into the phase of requirements creation would be a promising option.

11:00 *A/B Testing: A Promising Tool for Customer Value Evaluation*

Peitsa Hynninen and Marjo Kauppinen

Abstract: This paper aims to describe the use of A/B testing, a method used in web-development and position it in the context of the requirements engineering process and customer value evaluation. We propose that A/B testing can complement qualitative user research and offer a potential way to validate the value which system improvements bring to customers. Finally, we discuss our research plan for investigating the use of A/B testing in company contexts.

11:10 Panel Discussion

Quality Requirements

Session chair: Michael Unterkalmsteiner

11:35 *Position on Metrics for Security in Requirements Engineering*

Mahwish Kundi and Ruzanna Chitchyan

Abstract: A number of well-established software quality metrics are in use in code testing. It is our position that for many code-testing metrics for security equivalent requirements level metrics should be defined. Such requirements-level security metrics should be used in evaluating the quality of software security early on, in order to ensure that the resultant software system possesses the required security characteristics and quality.

11:45 *Verifying Security Requirements using Model Checking Technique for UML-Based Requirements Specification*

Yoshitaka Aoki and Saeko Matsuura

Abstract: Use case analysis is known to be an effective method to clarify functional requirements. Security requirements such as access or information control tend to increase the complexity of functional requirements, and therefore, need to be correctly implemented to minimize risks. However, general developers find it difficult to correctly specify adequate security requirements during the initial phases of the software development process. We propose a method to verify security requirements whose specifications are based on Unified Modeling Language (UML) using the model checking technique and Common Criteria security knowledge. Common Criteria assists in defining adequate security requirements in the form of a table. This helps developers verify whether UML-based requirements analysis models meet those requirements in the early stages of software development. The UML model and the table are transformed into a finite automaton in the UPPAAL model checking tool.

12:00 *Using Automated Tests for Communicating and Verifying Non-functional Requirements*

Robert Lagerstedt

Abstract: In software development the code often must comply to a number of non-functional requirements, like architectural requirements. These requirements are often communicated and verified by writing guidelines and creating reports of the non-compliance. This way of communicating and verifying non-functional requirements is very costly since all developers needs to understand all requirements. It is also very hard for a developer to remember all requirements and it is easy to make mistakes. In software development much of the work is done in a tool-chain. The tool-chain contains tools like text editors, compilers, linkers, static analysis tools, automatic test frameworks etc. An alternative way to communicate and verify non-functional requirements is to add them to the tool-chain as automated tests and checkers, so the developers get fast automated feedback when they do mistakes. I have worked many years as a software architect defining and writing architectural requirements and my observations and experiences shows that the productivity is increased and number of non-compliant non-functional requirements is lower using the tool-chain feedback instead of using guidelines and reports.

12:10 Panel Discussion

Lunch Break (12:30-14:00)

14:00

Formal Languages and Models Session chair: Markus Borg

14:00 *C&L: Generating Model Based Test Cases from Natural Language Requirements Descriptions*

Edgar Sarmiento, Julio Cesar Sampaio do Prado Leite, and Eduardo Almentero

Abstract: Software testing tasks are usually time-consuming, especially if one considers complex projects. Requirements engineering artifacts are a valuable starting point for the development of software products, and most of software requirements specifications are written in natural language. This paper presents a tool that implements an approach for generating test cases based on Natural Language (NL) requirements specifications. The C&L tool translates automatically NL requirements descriptions into behavioral models to support automated testing. Our approach is easy to use and at the same time it decreases the time and the effort with respect to test case generation. Demonstration of the feasibility of the proposed approach is based on a example of use that describes the operation of the C&L tool.

14:15 The Observer-Based Technique for Requirements Validation in Embedded Real-Time Systems

Jiale Zhou, Yue Lu, and Kristina Lundqvist

Abstract: Model-based requirements validation is an increasingly attractive approach to discovering hidden flaws in requirements in the early phases of systems development life cycle. The application of using traditional methods such as model checking for the validation purpose is limited by the growing complexity of embedded real-time systems (ERTS). The observer-based technique is a lightweight validation technique, which has shown its potential as a means of validating the correctness of model behaviors. In this paper, the novelty of our contributions is three-fold: 1) we formally define the observer constructs for our formal specification language namely the Timed Abstract State Machine (TASM) language and, 2) we propose the Events Monitoring Logic (EvML) to facilitate the observer specification and, 3) we show how to execute observers to validate the requirements describing the functional behaviors and non-functional properties (such as timing) of ERTS. We also illustrate the applicability of the extended TASM language through an industrial application of a Vehicle Locking-Unlocking system.

14:30 *Towards the Automated Generation of Abstract Test Cases from Requirements Models*

Maria Fernanda Granda, Nelly Condori-Fernández, Tanja E. J. Vos, and Oscar Pastor

Abstract: In a testing process, the design, selection, creation and execution of test cases is a very time-consuming and error-prone task when done manually, since suitable and effective test cases must be obtained from the requirements. This paper presents a model-driven testing approach for conceptual schemas that automatically generates a set of abstract test cases, from requirements models. In this way, tests and requirements are linked together to find defects as soon as possible, which can considerably reduce the risk of defects and project reworking. The authors propose a generation strategy which consists of: two meta-models, a set of transformations rules which are used to generate a Test Model, and the Abstract Test Cases from an existing approach to communication-oriented Requirements Engineering; and an algorithm based on Breadth-First Search. A practical application of our approach is included.

14:45 Panel Discussion

COFFEE BREAK (15:00-15:30)

15:30	**Mapping exercise**	Elizabeth Bjarnason
17:00	**Sum-up and Workshop Closing**	Michael Felderer