

**“Beware of bugs in the above code; I have only proved it correct, not tried it.”
 Donald E. Knuth**

**“If anything can go wrong, it will.”
 Edward A. Murphy**

Abstract:

Project managers recognize that difficult-to-use processes in software development hamper their teams in fulfilling the needed tasks within the planned time. Often the resulting output does not meet the customers needs. On the other hand the business requirements of customers change rapidly and all involved stake-holders should respond by adapting the project goals.

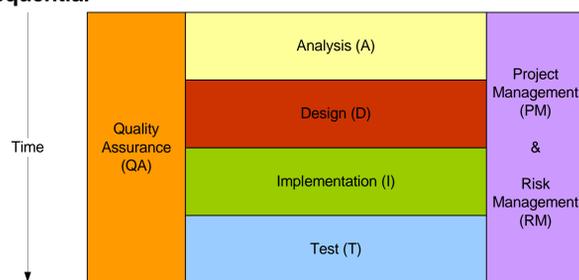
Thus they are looking for processes which are more light-weight so they are able to react in shorter time for changed needs. Furthermore project managers want to produce high-quality software with the smallest possible effort to obtain the most profit. In this thesis we introduce a package of verification and validation (V&V) techniques in combination with agile processes and show how they work and which problems could be solved by them. We focus on different methods for defect findings relating on different artifacts in different phases of software development. We introduce a technique that is called Usage-based Implementing (UBI) which is based on PairProgramming. Furthermore we replicate the method Usage-based Testing (UBT) which is conducted as desk-testing (UBT-i). Usage-based Reading (UBR) is a well known technique that we use to set Usage-based Testing and Usage-based Implementing in relation which each other.

The results of our pilot study show that a combination of *UBI* and *UBT-i* or *UBR* in an agile process, which we introduce as the Agile Inspection and Implementation Process (AIIP), is a good approach to produce reliable software and to support project teams with errorless artifacts so costs do not exceed expectation.

Theory:

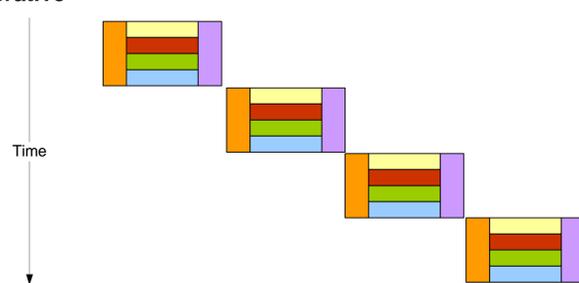
Several development processes exist to support the work of software developers, for example sequential variants like the waterfall model.

Sequential



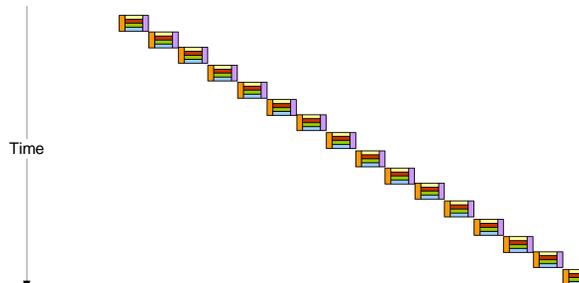
Rapid prototyping is a representative of the iterative software development methods. In every iteration the engineers refine the artifacts which have been produced prior.

Iterative



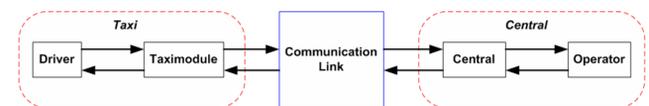
Agile processes like eXtremeProgramming or SCRUM are based on lots of iterations which are short. Hence the effort for planning and performing them, supported by agile methods, for example, PairProgramming, is smaller. The major rule for team leaders who conduct agile processes is that the customers should be involved by getting feedback in short intervals.

Agile



Pilot Study Setup – Taxi Management System:

The basis of our pilot study is a follow-up to several experiments which were performed at Vienna University of Technology, Austria and Lund University, Sweden.



There are several artifacts related with the Taxi Management System. The requirements, which are assumed to be correct, use-cases which were prioritized by experts and the design documents. We refactored all documents to match the standard of UML 2.0 and implemented the whole system in Java. After that we seeded 60 defects into the documents, 33 in the source-code and 27 in the design document. We classified all defects into the categories critical defects (A), major defects (B) and minor defects (C).

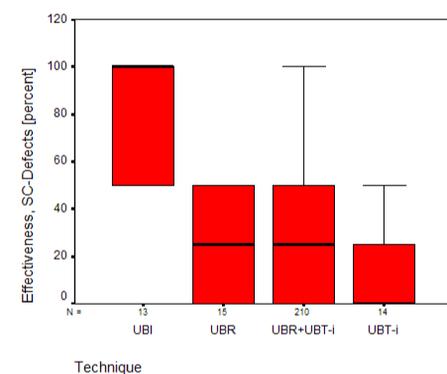
Techniques:

We used the following techniques to measure the difference in defect finding between an inspection, a testing and an agile implementation technique.

- Usage-based Reading, **UBR** (Biffl et al., Thelin et al., Winkler et al.)
 - reading through artifacts
- Usage-based Testing, **UBT-i**
 - replicated version of UBT (Andersson et al.)
 - finding matching equivalence class
 - create test-cases for inspection
- Usage-based Implementing, **UBI**
 - PairProgramming (Williams et al., Cockburn et al.)
 - detecting faults while implementing

Results:

We found out that there was no statistical significant difference comparing the different techniques when having a focus on the effectiveness in the design document. On the other hand inspectors that conducted UBI were more effective with finding defects in the source code than UBR and UBT-I, but spent the four-fold effort because of the additional implementation tasks and the fact that two persons are working concurrently.



Follow-Up – Agile Inspection and Implementation Process:

The well-known problem that Boehm stated is that the later an error is found the higher are the costs to fix them and the more rework effort will they produce. With UBI many errors can be removed while implementing but quality assurance with for example UBR or UBT-i is still needed. The Agile Inspection and Implementation Process is an extension of the SCRUM Process (Schwaber) to support producing high quality software.

