

Titel

Requirement-basierter Test in der Praxis

Referent(en)

Kemminger, Karl / Austrian Testing Board, Österreich

An wen richtet sich der Beitrag?

Tester, Testmanager, Qualitätsmanager, Anforderungsmanager, Softwareentwickler

Keywords / Stichwörter

Anforderungsmanagement, Testmanagement, Requirements

Abstract / Zusammenfassung

This presentation describes practical experience with test management for various innovative projects at Siemens Austria which were carried out in the past few years. The projects deal with telecommunication software and embedded software on specific devices.

The test cases are based on user requirements, and so efficient requirements engineering is essential for project success.

History and motivation

Originally no requirements engineering was established and the requirements were described in Word documents only. There was no systematic tracking of test cases to requirements, and checking to see if all requirements were covered was done on a review basis only. Software was developed using the waterfall model or V-model.

Two objectives led to the introduction of tool-based requirements engineering:

Design-to-cost (to avoid implementing unnecessary features)

Quality improvement (to reach higher CMMI level)

The new tool was installed and requirements imported from Word documents. Test cases were stored in the same systems as special type and traced to requirements.

Whereas the new tooling was accepted quickly by the test team, the system was seen in the development team as an overhead with no usage for the development.

To improve the mindset of the project team and to ensure project success, additional features and improvements were implemented

Integration into existing environment (eclipse)

Support for automated unit tests

Support of agile and iterative methods (more and more projects used such methods)

These improvements are described in detail below.

Integration into existing environment

All functions were integrated into the development environment Eclipse which had already been used by development and therefore the look and feel was well-known. All functions were implemented as Eclipse plugins which allow for easy installation and access.

The unique user interface allows access to all configuration objects such as requirements, source code, test cases, bug reports, etc.

The window arrangement of the user interface can be configured for different functions such as project lead, developer, test manager, tester, scrum master etc. Each team member therefore has comfortable access to those objects and views which are of interest to him.

With these functions, the tool becomes a central collaboration platform for the whole team.

Main functions of the Eclipse plugins are

Creation, update and deletion of configuration objects (requirements, test cases, etc.) in Eclipse

Connect/derive test cases to/from requirements

History of all configuration objects

Marking of derived objects as „Suspect“ following changes

Searching and filtering

Visualization of results and statistics

This requirements engineering tooling covers also the most important tools for test management

Requirements Management

Test Management

Incident Management

Testing is supported in various ways, for example

Support of all test phases

Test cases derivable from requirements, source code, ...

Coverage check

Label derived objects from changed objects as „suspect“

Review support

Automated test document creation, e.g. in Wordformat

Support for automated unit tests

The tooling allows for the insertion of JUnit test suites which are carried out for each build.

Results are stored automatically in tool. This provides good documentation of unit test results which have not been documented before, and allows the test team to also evaluate the results and gives objective criteria for release of software for the next test level.

Support of agile and iterative methods

Several functions were implemented to support the new iterative and agile methods which were used in many projects last few years.

The system allows each team member to edit open tasks on a daily basis (e.g. estimated, previous and left hours), and creates automatic graphical visualizations (e.g. Burndown graph).

Systemtest experience

Furthermore, practical experiences and hints for the different phases of the fundamental test process according to ISTQB are described and some examples are given.

These hints are the result of 30 years of practical experience as a tester and test manager. It does not cover the theory, but describes how one should handle particular situations which are common in testing.

Biografie

1975-1979 Studium Technische Mathematik an TU Wien

1980-2009 Testmanager und Dienststellenleiter bei Siemens AG Österreich, PSE

Seit 2009 Mitarbeit im Austrian Testing Board

Einen ausführlichen CV als .pdf kann ich jederzeit gerne zur Verfügung stellen.

Kontaktinformationen

Kemminger, Karl

Austrian Testting Board

1200 Wien

Austria
