



MicroDoc Computersysteme GmbH

Author: Christine Mitterbauer

Date: Okt. 04, 2006

Position statement:

Test driven development approach demands sophisticated tool support for being accepted and successful

Topic outline:

Implementing embedded software using the test-driven approach sounds very interesting: first describing the requirements and functionality of the desired product in form of test cases, then implementing according to these requirement descriptions and after that refining and refactoring the code always being able to test the current implementation versus the requirement specification. In addition with a test coverage of almost 100 % of the code, maintenance and enhancement life cycles should in theory become faster, less error-prone and predictable. To achieve these goals sophisticated tools are needed supporting every team member in defining, running and evaluating all product requirements as test cases during the whole life cycle of the software. The acceptance of such an approach depends directly on the facilities of the testing framework tool.

Key features are:

- Intuitional and easy to use for all team members
- multi-user-support
- comparability and reproducibility of tests and results including
 - storage of previous tests and their results
 - tools to compare different test runs
 - support for test documentation
- automation
 - of test case creation (autogeneration)
 - of test execution during development and build process including test documentation
- support for different testing aspects
 - customer tests (acceptance tests)
 - developer tests (unit tests, module integration tests)
- support for different testing environments
 - within the development system (e.g. via using simulations)
 - remote on the device (with remote debugging capabilities)
 - test result processing local or remote
- support for configurable tests
 - testing in different threads, at the same time or in sequence
 - specifying various sequence szenarios of single tests

- declaration of several test input parameters and the appropriate expected test results
- configurable result processing
 - how: persisted or not
 - location: local or remote
- support for different J2ME configurations
 - CLDC
 - CDC
 - MIDP

As we are actually implementing embedded software with eclipse using OSGi, we face a lot of problems with OSGi service dependencies when testing. The dynamic nature of OSGi services leads to the need for a testing tool which is fully integrated in the eclipse framework (equinox) considering these service dependencies and the life-cycle of managed OSGi software components.