Test Manager: The Test Automation Component for the Maintenance of Large-Scale Systems

Michelangelo Interesse
IT Telecom - TelecomItalia Group, Netsiel S.p.A., via S. Dioguardi 1
Bari, BA 70124, Italy
web: http://www.netsiel.it/

Abstract

Nowadays, the information systems of large companies are quickly growing in complexity. In fact, tough new software development technologies and technical infrastructures are ever more powerful and allow higher degrees of flexibility, on the other hand this translates in a more complex solution for the final software systems and their maintainability results in a very challenging task. This is further emphasized in the cases, actually very common, where modern systems need to coexist with legacy ones. In this scenario, the testing of new functions or changes to existing ones has become a very demanding task.

Netsiel has developed, and it is continuously improving, a specific methodology (SAMMIT – Software Application Maintenance Methodology and Industrial Techniques) in order to properly define the processes (tasks, roles, products, etc.) covering the entire “maintenance life-cycle” and to aid software engineers in performing all the maintenance tasks required by large and complex information systems. Moreover, the methodology itself is complemented and fully supported by a proprietary software tool, the SAMMIT WorkBench, assisting software engineers in their day-by-day maintenance operations.

One of the key components of the tool is the Test Manager, whose aim is to support the “maintenance teams” in testing new functions or changes to existing ones in large-scale systems. The Test Manager is well suited to automate most of the testing tasks, ranging from Test Suite planning and design to Test Cases execution and Rework Issues management. Moreover, it supports testing at several levels (unit, functional, integration, system, regression, performance, acceptance, etc.) and it allows for direct relationships between Test Suites and software system elements, at any hierarchical level (entire system, application component or area, function, software object). Each Test Suite is made up of several Test Cases, grouped by one or more levels, arbitrarily defined for each software element type of any specific system. Moreover the versioning is handled at the Test Suite level and automatic inheritance mechanisms are handled either between subsequent versions at the same test level or between the corresponding versions of the Test Suite at different test levels. On the other hand, each project can include several Test Plans, each of them links to one or more Test Suites among those defined for the software system associated with the project itself.

The Test Manager handles a locking schema in order to prevent any possible conflicts among Test Plans, possibly belonging to different projects. Moreover, it handles roles at the user and group level, in addition to a notification mechanism of state changes and single task completion, which coordinates operations among maintenance team members.

A complete tracking of Test Plan execution is handled at the level of Test Cases and Rework Issues. Such a tracking system, together with notification features, statistics and process metrics and the web browser user interface, provide full process control and overhead minimization of intra/inter-team communication, while making it a reality the collaborative sharing of the test process among geographically distributed teams (or remote-working).

The Test Manager is founded on a J2EE-based multi-layered web architecture, which together with leading-edge application frameworks (i.e.: STRUTS) and design-patterns (i.e.: MVC) contribute to the high levels of scalability and security of the whole platform.