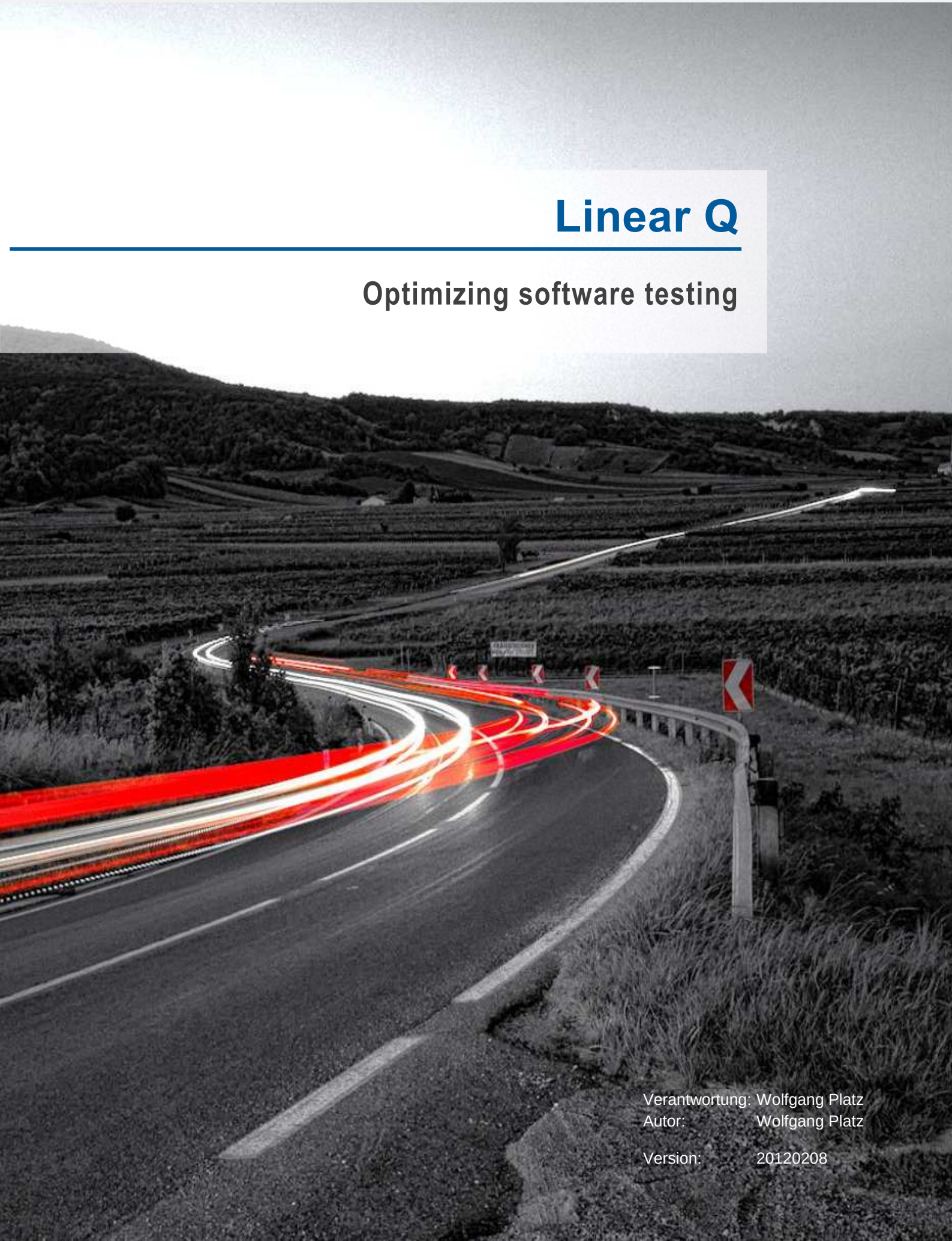
Linear Q

Optimizing software testing



Verantwortung: Wolfgang Platz
Autor: Wolfgang Platz

Version: 20120208

1 Executive Summary

The new *Linear Q* test method offers its users great potential for increasing the effectiveness and efficiency of their software testing. The benchmark for measuring these optimizations is the test coverage that a test can achieve during a specific run-time. Users of *Linear Q* can:

- significantly increase test coverage at the same cost
- and dramatically reduce costs at the same test coverage.

Selected reference projects have shown a potential to save at least 50% of the costs compared to traditional approaches, when the test coverage should reach 50 %. And the potential of *Linear Q* continues to grow as test coverage increases.

Linear Q is a comprehensive test approach that begins with a risk-based functional structure and ends with a final result that shows the test results in relation to the risk-based structure. The key elements of *Linear Q* are:

- Methodical approach of *Linear Expansion*™ for defining optimized test cases
- Concept of *inner values* for determining each individual test case's contribution to the overall test
- Business-based test automation with minimal maintenance effort - implemented according to the Business Dynamic Steering concept
- Sourcing patterns
- Possibility of reducing the business unit's testing to less than 10 % of the total effort

The basic tool for *Linear Q* is the *TOSCA Testsuite*™. This is the only test solution available on the market that supports *Linear Q* throughout every stage of testing. *TOSCA Testsuite*™ allows users to get the most out of *Linear Q*.

Inhalt

2 Current situation and potentials

The current economic instability as well as changes in the financial market have led to enormous cost pressure on IT departments - and testing, too. Nevertheless, the quality of testing must not deteriorate.

Many companies have sought to solve the cost problem by outsourcing their testing activities or by using test automation. Many of our customers report very unfavorable experiences with sourcing and automation.

Linear Q is a method for optimizing the effectiveness and efficiency of software testing. It makes it possible to achieve maximum test quality at minimal cost. The following will explain the main features of this method and its USPs.

2.1 The dilemma: measuring test performance

Using the test coverage that can be achieved within a specific run-time is a reasonable way to estimate test performance. Test coverage shows how much of the overall functionality of the system under test (SuT) the test covers.

In practice, the different functions of the SuT vary in importance to users: generally, 80 % of the business is handled by 20 % of the functionality and vice versa (80:20 rule). Thus, meaningful test coverage begins with risk weighting of the functions under test (risk-based testing approach, relative test coverage). Test cases must be created for these functions and they must also be assessed according to the risk covered by them.

Before *Linear Q*, the testing industry didn't have a reliable tool for measuring test coverage. This explains why test managers have tried to use large numbers of tests to show test performance since they couldn't specify test coverage. This has also led to the development of large test case portfolios with very little effect.

2.2 Effectiveness and efficiency

Testing can be optimized by increasing the following parameters:

- Effectiveness: have the right test cases been created, that is, the test cases that make the biggest contribution to test coverage?
- Efficiency: have the test cases been executed as cost-effectively as possible and with minimum run-time?

The primary way most companies attempt to optimize test performance is by making test execution more cost-effective (= increased efficiency). But these efforts will not achieve much without also considering how to improve effectiveness.

Linear Q is designed to enhance both parameters.

3 Linear Q

3.1 Overview

Linear Q is a method for optimizing the effectiveness and efficiency of software testing. It covers the following areas:

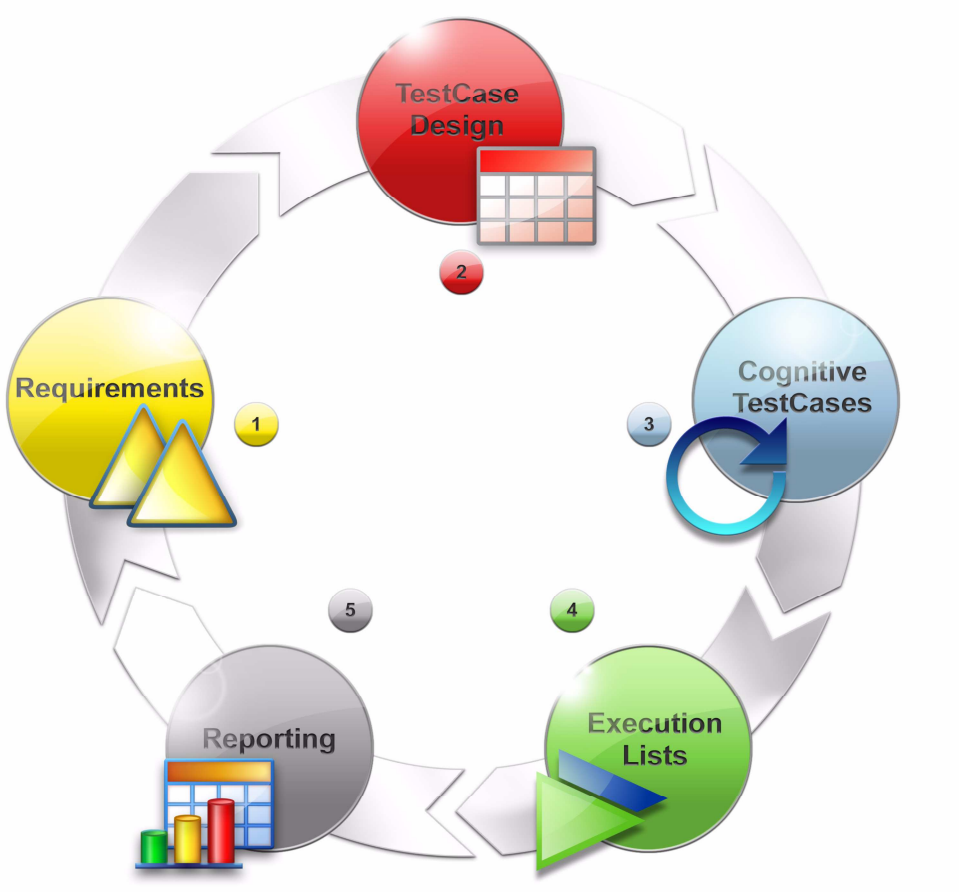


Illustration 1: Phase model of Linear Q

The *Linear Q* method covers the following phases:

Phase 1: Risk-based functional structuring on the basis of requirements

The functional and non-functional requirements of the SuT are defined in a tree structure. The impacts of an inoperable function are estimated from a business perspective and are recorded as the risk weight of the function.

This results in a list of all the functions, which can be prioritized according to weight and fed into *TestCase-Design*.

Phase 2: Methodical *TestCase-Design*

TestCase-Designs are created for the individual functions. These logical outlines of test cases are drafted quickly on the basis of highly innovative methodical concepts. In this way, the contribution of each test case to the overall test coverage can be determined.

Risk-based functional structuring and **methodical *TestCase-Design*** are the cornerstones of *Linear Q* as well as the key to **optimizing the effectiveness of software testing**.

Phase 3: TestCase specification and automation

The test case outlines are automated according to their regressive defect potential and their technical feasibility. The level of detail of the specification depends on the character of the test case (one-time test vs. regression test).

Linear Q ensures that the specification is detailed enough, but not too much, to keep costs under control.

Automated tests are profitable when they can be maintained efficiently. The basic tool of *Linear Q* is the *TOSCA Testsuite*™, which reduces the maintenance effort of automated testing to the absolute minimum *[TWPPS12008]]*.

***Phase 4*:** Test execution

Test execution is structured according to several points of view:

- Test stages (integration testing, user acceptance testing etc.)
- Chronological order of testing - under certain circumstances tests must run for several logical test days *[TWTDK12010]*
- Creating work packages for testers (manual testing) and test computers (automated testing)

Linear Q provides effective and efficient best practices for structuring test execution in *TOSCA* as well as for manual and automated testing.

Test automation with minimal maintenance and **structured test execution** are key elements of *Linear Q* for **optimizing test efficiency** and for reducing test run-times.

Linear Q is also suitable for **sourcing patterns**, which means that viable sourcing models can be executed.

Phase 5: Reporting and documentation of test results

Test results are projected back on to the risk-based structure (see Phase 1). Test performance is evaluated on the basis of various *key performance indicators (KPIs)*.

Linear Q provides **template reports** and **KPIs**, which can be used to represent and monitor test results and test performance.

3.2 Methodical TestCase-Design with Linear Q

Linear Q gets its name from its key feature: a unique *TestCase-Design* method. *TRICENTIS*® has developed the **Linear Expansion method** on the basis of established methodical concepts (creation of equivalence classes and boundary values, combinatorial principles).

This method resolves what appears contradictory at first glance: the requirement of minimizing the number of test cases while at the same time ensuring efficient defect localization and maximum test coverage.

This method is also the basis of determining the *inner value* of test cases, that is, the contribution each test case makes to the overall test coverage.

Based on the cornerstones of **Linear Expansion** and the **concept of *inner values*** *Linear Q* is the key to determining how much test coverage has been achieved.

At the same time, methodical *TestCase-Design* with *Linear Q* provides the added benefit of supplying the data objects which are needed as the test data for complex processing operations *[TWTDI22010]]*.

3.3 Potentials by the numbers

In comparison with traditional approaches, test projects with *Linear Q* can significantly increase test coverage at the same cost and dramatically reduce costs at the same test coverage.

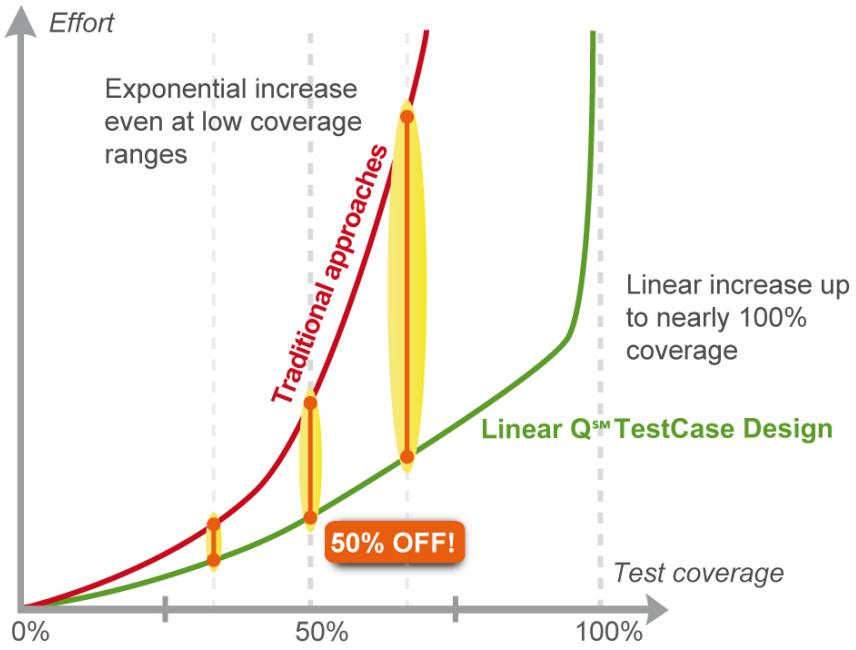


Illustration 2: Effort and costs with Linear Q

The higher the targeted test performance is, the greater the potential savings with *Linear Q* will be:

- The savings are usually already well above 50 % at a test coverage of 50 %.
- In practice, *Linear Q* is the only approach to offer more than 90 % test coverage for complex systems at a reasonable expense.

3.4 Organizational potentials

Linear Q uses a workshop format to create *TestCase-Designs*: *subject matter experts* (SME), such as business analysts from the different business units, create test cases together with a moderator and a test expert.

These workshops not only lead to very effective test cases, but also serve as the most efficient way imaginable to transfer know-how to the test experts. These experts can then do most of the testing themselves without the further assistance of the SMEs.

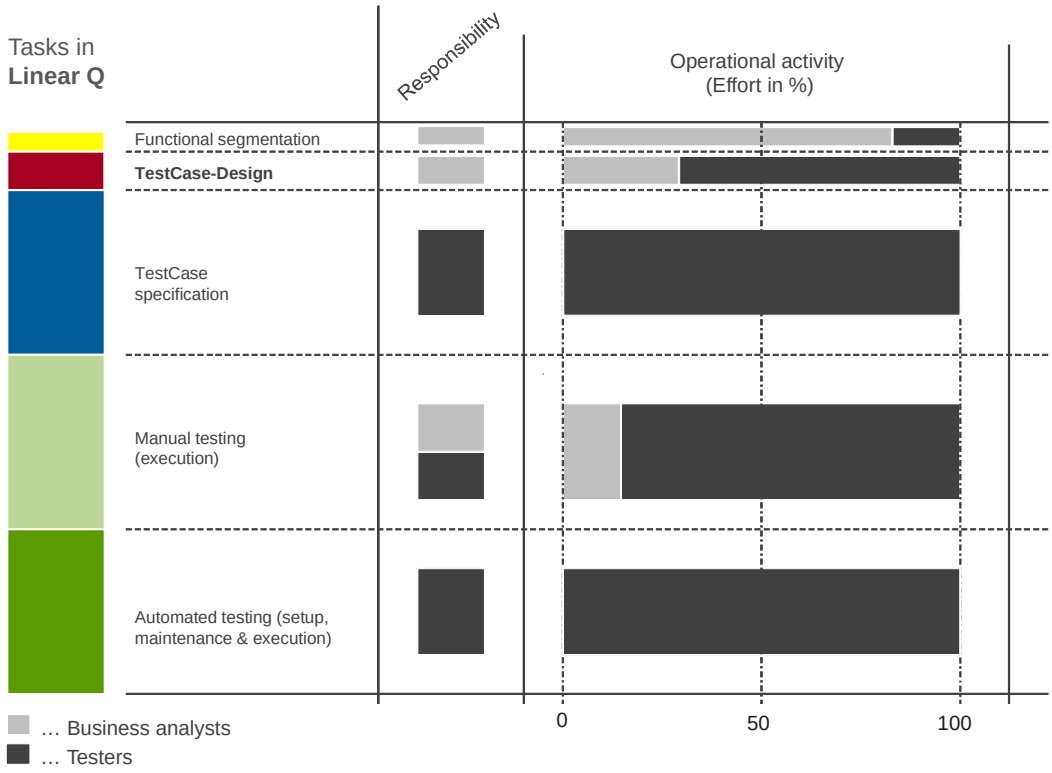


Illustration 3: Distribution of responsibilities and effort in Linear Q

- The phases for increasing effectiveness (Overview) are the responsibility of the SMEs, but the test experts already make a decisive contribution to structuring the test with their domain knowledge.
- Using *Linear Q* test experts can do most of the specification, maintenance and execution of tests on their own, but they need the SMEs help for the manual *user acceptance testing*.
- All in all, considering the amount of effort needed for each phase, the operational testing activity of the SMEs can be reduced to well below 10% of the total effort.

4 TOSCA Testsuite™ - The basic tool for Linear Q

TOSCA Testsuite™ provides ideal support for the *Linear Q* method:

- The risk-based functional structure is built in the Requirements AddIn. *TOSCA* offers a tool-supported, modern method for assessing the risk of each requirement.
- The *TestCase-Design* AddIn provides automated combinatorial generation, including the Linear Expansion method.
- *TOSCA* uses the concept of *Business Dynamic Steering* to solve the maintenance issue of automated software testing.
- *TOSCA* makes defect tracking easy with integrations to a wide variety of defect tracking tools.
- Projecting the test results back on to the Requirements structure produces transparent reports based on test coverage.

5 Linear Q - Benefits and USPs

5.1 USPs

- Methodical approach of Linear Expansion for defining optimized test cases
- The concept of *inner values* for determining the contribution of each test case to the overall test coverage
- Business-based test automation with minimal maintenance effort - implemented according to the Business Dynamic Steering concept
- *Linear Q* provides template reports and KPIs, which can be used to represent and monitor test results and test performance.

5.2 Benefits

- The savings are usually already well above 50 % at a test coverage of 50 %.
- All in all, considering the amount of effort needed for each phase, the operational testing activity of the SMEs can be reduced to well below 10 % of the total effort.

6 References and sources

[TWPPS12008] Paradigm shift in test automation, Wolfgang Platz & Michael Hentze, *TRICENTIS*® white paper, March 2008

[TWTDK12010] TOSCA@data, The Solution to the Test Data Problem, Wolfgang Platz, *TRICENTIS*® white paper, March 2010

[TWTDI22010] TOSCA@data Implementation, Wolfgang Platz, *TRICENTIS*® white paper, March 2010

***