

Mind the gap between Test Strategy and Execution

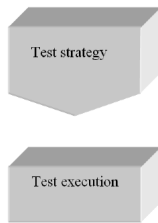
by Erik Runhaar

“We want 100% coverage of all functions. This means 100% service readiness, overall”. This was a statement I heard an airplane-factory manager say to the R&D department of an ERP software house. “This is what we do within our airplanes and this is what we expect from you as well”. It looks like a reasonable acceptance criterion and, especially within an airplane-factory, we believe people accept this responsibility. It is a logical statement at an employee meeting where pep talk is the main goal. But if we analyze what this woman really tried to make clear, we cannot distil anything useful for testing out of this statement.

This is exactly the problem we are facing when dealing with the issue: what do you mean when you are talking about coverage? And what is the link between this coverage and test execution and strategy?

First of all I will illustrate the problem with the following example, which might be recognizable.

The project has started and Jan is invited to the Project Start Up meeting. In the organization Jan is working for, testing has the right place early in the project. Directly at the start of the project, Testing is involved and represented by the overall test manager (Jan). After the meeting Jan starts to collect product (and project) risks and priorities in a stakeholder meeting. He invites development as well as support, the customer, testing and other groups on the impact side and the likelihood of failure side. The meeting results in a test strategy in which the most important risks that were raised in the meeting are covered. Jan finishes his Master Test Plan. The document is approved by the customer. So far this project runs in a way many test managers could only dream of. Then the test analysts are involved. They start



reading the functional design and create their test scripts (test procedures). They do their utmost to cover as much as possible of the Functional Design. They even use the two test techniques (equivalence partitioning and error guessing) they learned during a course. In some cases they switch to the method of the tracer, marking all important parts in the design. If you were able to talk to them, and ask how they know what to test (in other words: how they implemented the strategy), their answer would probably be: “Oh, if time wasn’t that scarce...”

This example illustrates the challenge many companies sooner or later have to face. There is a gap between the strategy and what the testers do in script preparation and execution. In my experience, the main reason for this is that the testers in most cases are not involved in the strategy meeting. They are not aware of a strategy at all defined by the stakeholders in the company, including the customer. In other cases, there is a strategy and the test manager and the testers are aware of this document, but none of them link this strategy to the way of testing, or are aware that that strategy should determine test execution. The following question arises: what do you report in a situation, where the test strategy reflects the customer’s requirements for quality, but the test scripts just aim to cover as much as possible, because you never know...

Issue or not

Today IT and Testing is becoming more and more “visible”. On the one hand customers

want the software to be ‘defect-free’ (a persistent misunderstanding). On the other hand, the customer wants the same software preferably yesterday and for a limited budget. That is the problem IT faces nowadays. As a reaction to this, agile testing methods and exploratory techniques arise as the solution: shorter time frames, shorter lines between customer and IT and Testing, no bureaucracy. No worries about issues like test strategy and test execution or even a relation between the one and the other.

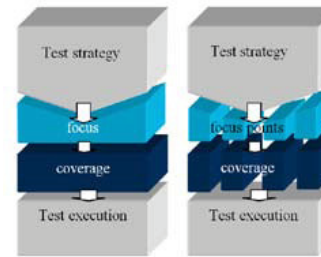
I notice in my daily work as a professional tester, that many companies do not even recognize this as an issue. In practice, at some point, it will hit the organization in the face. Traceability of the result and risks covered into test execution is not possible.

Another category of person is convinced this problem is unsolvable: “We have to live with it, we have to work in short time spans and with low budgets. So what choice do we have?” Especially in the latter situation, the answer is that the indication of scarce time and money turns out to be the key to solve the problem.

The clue

Time and money, these are the levers to link the customer’s requirements and the results of test execution.

A concept is needed to get the best testing done (cost) in the available time (time). This concept deals with what to test (focus). This focus must be determined by the customer of the product. This customer has a long list of

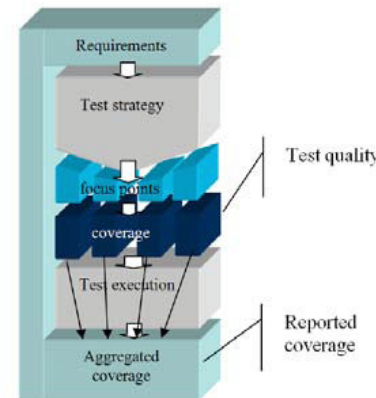


wishes and requirements towards the product, but also accepts the fact that “testing everything is impossible”. Therefore we need to know what aspects of the software system are really necessary to undergo a test. These strategic aspects need to be translated into measurable focus. Focus like: “Calculations need to be correct for all combinations of pricing-aspects in the billing module” or the user-process of the Call Centre needs to run without blocking problems.

What concept can fill this gap between the focus points, translated from the customer requirements, and the tests prepared and executed by the tester? Especially in a short period of time! This concept is called ‘Coverage’. Not coverage like: zero defects or service ready. The coverage needs to be related to the focus. With a good test script the tester will be able to cover 75% of the combinations of pricing aspects by his test. This is the answer to the customer’s question. The customer will gain confidence if IT and Testing can explain *what* is covered. So its not about: “how much did you test?”

Customers and coverage

Based on this, we can answer the real question of the customer which is translated into the test strategy. We are also able to report accordingly. But all of a sudden the customer is not interested in this information. Even if we split it up into focus points or different perspectives, even if we link the coverage one-on-one to these focus points, the customer appears not to be interested. It is often too detailed and does not answer the real question: “does the system work for me?”



What we discover here is the difference between having confidence in the testing department (IT) or process, and having confidence in the quality of the software. Up to this moment we translated the customer’s requirements into a strategy, but at the end of the day, the only ‘one’ who is interested in this strategy is the testing process. The conclusion is that we have only discussed the coverage towards ‘test quality’.

We must conclude that the customer takes this

‘test quality’ for granted, Testing gets paid to do high-quality work. The customer wants to have insight into which parts of his wishes and requirements are tested. The test report needs to be a high-level coverage picture: a mix of testing quality coverage aggregated to requirements coverage.

At home I tried the following on my wife. She wants to buy a bike. I asked her what her quality requirements were for that bike. She had only one single requirement: that bike needs to have the quality of a bike, with wheels, a handlebar and pedals etc. It should not break at the first hole in the road.

This is the customer; not interested whether the pedals are tested in all possible positions, the only and main interest of the customer is whether the bike rides like one can expect from a real bike.

If testing takes care of filling the gap between Strategy and Execution with a clear coverage picture, testing will be *the added value* to an organization and its customers.



Biography

In 1996 Erik started his career at an ERP company as a tester for a distribution and workflow application. Over the years he moved on to test management and test process improvement both in the Netherlands and abroad. He has gathered a lot of experience in several organizations testing ERP in amongst others insurance, utilities, and the banking industry. He has managed test projects for newly built software, maintenance projects and more technical, migration projects.

Erik gradually moved from test management to training and coaching the test managers and testers and to set up or improve the processes within these companies. Erik is an accredited trainer for ISTQB Foundation and an ISEB Practitioner as well as TMap Next Foundation and Advanced.

Erik is currently Test Consultant and Trainer at Polteq International Testing Services BV in the Netherlands.