11

September 2010

testing experience

The Magazine for Professional Testers

Metrics

ISSN 1866-5705 www.testingexperience.com free digital version print version 8,00 € printed in Germany

© sandra zuerlein - Fotolia.com

The Value of Testing in 5 Dimensions

by Peter Zimmerer

Today most people in the software business agree that testing is important, but there is still a very diverse understanding of what testing is all about and what its value is. Unfortunately, there are still outdated definitions used for testing (see [IEEE 610.12-1990] or [IEEE 829-2008]) that limit testing just to "dynamically execute a piece of software to detect bugs". However, there is hope if we look at the definitions for testing as given in the ISTQB glossary (see [1])

The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.

or as given by Cem Kaner (see [2]):

Testing is an empirical technical investigation of the product / system / artifact / service under test conducted to provide stakeholders with information about the quality.

The question is: How can we put these definitions into practice in real project life? And how can we find a way to improve the communication of the value of testing to the different stakeholders like managers and developers? One answer could be to break testing down into 5 dimensions to better visualize its mission, motivations, activities, and corresponding values (see figure 1).

Dimensions of testing

Demonstrate, check, confirm, verify, validate

First of all testing is about demonstrating in a constructive way that something works, at least to some extent. Further it is about checking and confirming artifacts like requirements, features or use cases. And it is also about verifying ("Are you building the thing *right*?") and validating ("Are you building the *right* thing?").

Detect, search

In testing we try to detect bugs as early as possible in a destructive way, and we search for unknown and therefore unspecified behavior in the system under test.

Mitigate, reduce risks, investigate, explore

Any testing should be based (at least to some extent) on risks in the system under test, so this is also a very important dimension in testing. Thereby we also investigate and explore in order to look deep inside the system under test.

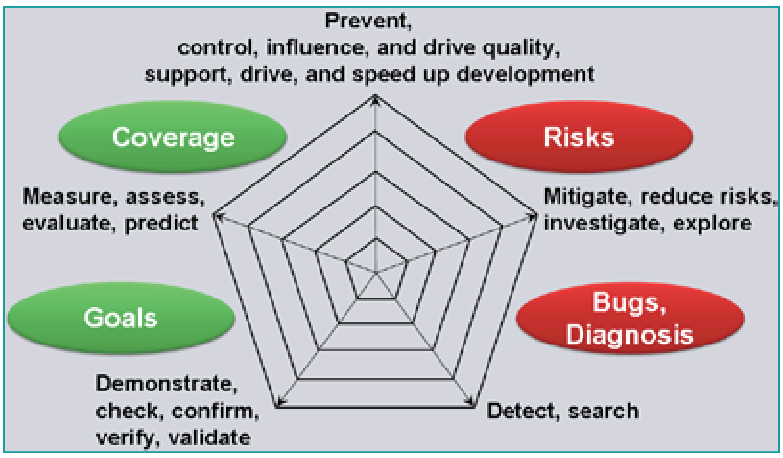

Figure 1: Dimensions of testing

Measure, assess, evaluate, predict

During testing we collect data and measure, for example to address quality attributes like performance, reliability or availability. Based on these data we know something about the current status, and we can hopefully make a good forecast for the future.

Prevent, control, influence, and drive quality, support, drive, and speed up development

Prevention of bugs should always be one direction in our testing activities. These could be approaches like systematic root cause analysis as well as early test case design by test-driven development – the idea is to use the information we get from testing directly to change or rather drive developments in the future. Furthermore, this testing dimension enables higher productivity and accelerated development by facilitating change, identifying any regressions as early as possible and decreasing maintenance effort.

All 5 dimensions are part of the ultimate mission of testing, which is to provide information and evidence about the quality of the product / system / artifact / service under test for the different stakeholders. If this information is effectively used, then we create real value for the business, i.e. the business value of testing lies in the savings that the organization can achieve from improvements based on the information that is provided by testing. These can be improvements

- in the product / system / artifact / service under test, for example by fixing a detected defect or removing an unspecified, unwanted behavior in the system
- in the decisions we make, for example by considering mitigated and residual risks in a release decision
- in the development approach (process) itself, for example by fostering prevention tasks.

A fundamental understanding of these dimensions is very useful in practice to effectively address typical questions in testing, as the following examples illustrate.

Developer: *My unit testing does not detect any important new bugs. Why should I do unit testing in the future?*

-> Do not miss the other 4 dimensions in understanding the real value of unit testing besides detecting bugs. Unit testing provides value within all 5 dimensions of testing.

Manager: *You have this big regression test suite. Why do we need to spend so much effort on just repeating these tests again and again?*

-> Map the value of the regression test suite to the different dimensions to emphasize its need and impact on the business to management. Often different parts of the regression test suite address the 5 dimensions to a different extent – that can be a starting point to prioritize the regression test cases based on the overall testing mission.

Project Lead: *You have all this great test automation in place, but the system still crashes quite often. What shall we do?*

-> This can be an indicator that by doing a lot of test automation the focus is currently too much on the "check" dimension only. Strengthen the testing on the "detect" and on the "mitigate risks" dimensions (including investigation and exploration) to expand and vary the scope of testing.

Frequently asked questions

Why are there just 5 dimensions and not more or less?

First of all, it should be easy to remember and easy to read. Based on psychological studies we know that a human being can remember 5±2 things at a point of time. And the number of different dimensions is always a trade-off between abstraction and detailing.

Don't some dimensions contain too many different things?

Again the intention is to avoid too many dimensions, i.e. to avoid fragmentation. Each dimension includes several activities with corresponding values that are related and are driven by the same motivation, which is why they have been clustered together into one dimension.

Are the different dimensions really disjoint?

No, not completely – but are totally disjoint dimensions possible at all? This means that dependent on the interpretation there is some overlap. For example "check" can also include "checking for some risk" and "investigate" can also mean to "measure some resource consumption". The important thing is to understand that every dimension visualizes a different, complementary view and perspective of the testing approach. Good testing always requires identifying and selecting an adequate mixture of these different dimensions to adequately cover the testing space.

The figure looks like a kiviat diagram. Is that intended?

Yes, definitely. You can think of an increasing intensity for every dimension from the center point to the edge. However, for perfect testing that does not mean that you should always be on the edges only, in fact that depends on the context. For example, if you develop a prototype for a tradeshow, the focus could be only on "demonstrate and check". If you develop a safety-critical system, you should also take "mitigate risks" and "measure" into account. So, for different projects the covered area in the diagram will be different. It is precisely the area covered which describes and visualizes the testing mission for a specific project.

Is there any semantics in the ordering of the dimensions in the figure?

Yes, there are some semantics concerning "bottom – top" and "left – right".

At the bottom are the more classical activities like "check" and "detect". As you go to the top of the diagram, maturity increases. That means more advanced testing approaches and processes as well as more experienced and professional people (!) are needed here, especially for the "prevent".

On the left side of the diagram are the more positive, constructive activities, which provide information on the achieved goals and coverage; on the right side are the more negative, destructive activities, which address things like bugs and risks.

The next step

Having this clear picture of the testing mission is very helpful to guide the overall testing approach and to define the testing strategy. That also includes the determination of the test design methods to be applied to create the right test cases for effective and efficient testing. So, an interesting question which comes up now is: "How do test design techniques support and foster these 5 dimensions of testing?"

To answer this question, we now cluster test design techniques into 5 related categories as well (see figure 2). Although there is no strict biunique mapping between all the corresponding edges of the diagrams in figures 1 and 2, there is a sound correlation between them. That means the 5 categories of test design techniques may contribute to several of the 5 dimensions of testing, but will do so with a focus on the corresponding edge as explained next.

Black-box -> Demonstrate, check, confirm, verify, validate

Demonstrating and checking is initially based on a black-box view of the system under test. So, the correlation between these edges is strong. Besides this, black-box test design techniques also contribute to other dimensions, e.g. to the "detect" and to the "measure" dimensions.

White-box -> Detect, search

Typically, a white-box view is needed (and beneficial!) to detect bugs and to identify strange, unknown behavior in the system under test. So, the correlation between these edges is also strong.

Grey-box -> Measure, assess, evaluate, predict

Perhaps here the direct relation between the two diagrams is not so strong and seems a little artificial, but a grey-box view is very helpful (if not needed!) for example in adequately measuring performance data or so.

Fault-based -> Mitigate, reduce risks, investigate, explore

Fault-based test design techniques directly address different types of risk, and they are a rich source to guide investigation and exploration, so again here is a strong correlation.

Regression -> Prevent, control, influence, and drive quality, support, drive, and speed up development

First of all the direct relation perhaps sounds a bit weak. However, if you have a regression test suite in place (preferable already from the beginning by doing test-driven development!), then you have a safety net that helps you to effectively prevent many defects in future and that enables you to control (and drive!) refactoring activities within development. Having these practices in place is also the starting point and a precondition for really improving productivity and accelerating development and maintenance.

For every category of the test design techniques given in figure 2, there is a number of specific test design techniques. A more detailed overview on these test design techniques can be found in [3].

Conclusion

Good testing requires many skills, capabilities, knowledge, experience, and creativeness. Therefore the diagrams as given in figures 1 and 2 should never be used too restrictively, but rather as a map and a general direction for the testing mission. Identifying the required intensity for each dimension and selecting an adequate mixture of them to adequately cover the testing space are the key for effective and efficient testing.

A good understanding of what testing is all about is only the first step. More difficult but even more important is to communicate this testing mission and its value in an appropriate manner to the different stakeholders like managers, project leads, developers, and even to other testers. For this purpose, characterizing testing in 5 dimensions to visualize its mission, motivations, activities, and corresponding values can be useful. The better we get at doing this, the better we will collaborate with the other stakeholders, and it will help us also to strengthen and sustain our own testing territory.

[1] ISTQB Standard Glossary of Terms used in Software Testing V.2.0, December 2007, http://www.istqb.org/downloads/glossary-current.pdf

[2] http://www.kaner.com/pdfs/ETatQAI.pdf

[3] P. Zimmerer, Systematic Test Design...All on One Page, STAREAST Conference 2008, http://www.stickyminds.com/s.asp?F=S14140_CP_2

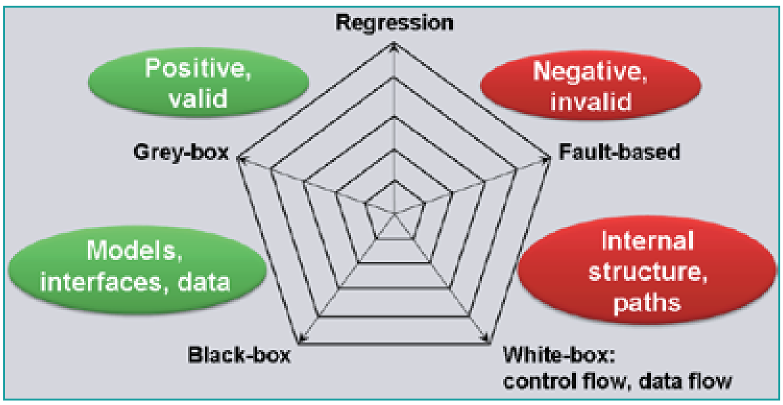

Figure 2: Categories of test design techniques

Biography

Peter Zimmerer is a Principal Engineer at Siemens AG, Corporate Technology, in Munich, Germany. He studied Computer Science at the University of Stuttgart, Germany and is an ISTQB® Certified Tester Full Advanced Level. For more than 19 years he has been working in the field of software testing and quality engineering for object-oriented, distributed, component-based, service-oriented, and embedded software. He was also involved in the design and development of several Siemens in-house testing tools for component and integration testing. At Siemens he performs consulting and training on test management and test engineering practices, including testing strategies, testing techniques, testing processes, test automation, and testing tools in real-world projects and drives research and innovation in this area. He is author of several journal and conference contributions and regular speaker at international conferences in Europe, Canada, and USA.

Subscribe at

testing experience

www.testingexperience.com