9

March, 2010

testing experience

The Magazine for Professional Testers

printed in Germany

print version 8,00 €

free digital version

www.testingexperience.com

ISSN 1866-5705

© iStockphoto.com/NWphotoguy

Advanced Testing Techniques

iqnite GERMANY2010 Conferences Special

Test Techniques - Why bother?

by Erik van Veenendaal

Formal and informal test design techniques have been around for many years. Their uptake has, to say the least, not been the best. A recent survey revealed that a mere 50% of the test organizations actually use test design techniques in their projects, and only 25% used multiple test techniques, thereby being able to differentiate test coverage based on identified product risks. However, the recent growth in testing professionalism has brought many techniques to the foreground of the testing activity. There are many sources of information with respect to test techniques, their methods and coverage measure. (My favorite ones are listed at the bottom of this column). Let's look at a number of reasons why we should consider the use of test design techniques as part of our everyday testing activities, and who should be looking to use them. Test techniques come in many shapes and sizes, some formal and some not, some dynamic and some static. Almost inevitably, the focus will be drawn to the dynamic test design techniques and activities. However, static techniques must always be borne in mind, prevention will always be better than cure. Of course, there is not one complete answer to suit every situation, type of application or degree of quality required.

Are techniques for testers only?

Absolutely not! Everyone who is involved in product development, implementation and maintenance or use has a vested interest in establishing the quality of the product to be delivered. Specific techniques target one or more specific types of defect. Also, software engineers and business analysts should be taught techniques, including reviews, which reflect the development phase they are involved in. Everyone should be made aware of the type of tests they could carry out and the value they would add. If everyone in product development was aware of this issue and took the time to apply the basic aspects of test design techniques, then many, many, many defects would either not happen in the first place or be found before they could have an impact. The good news is that we at Improve Quality Services are providing more and more training and practical workshops regarding test design techniques to software engineers. Also, the current shift to agile development means testing is a team responsibility, and everyone should be involved in testing.

Why use testing techniques at all?

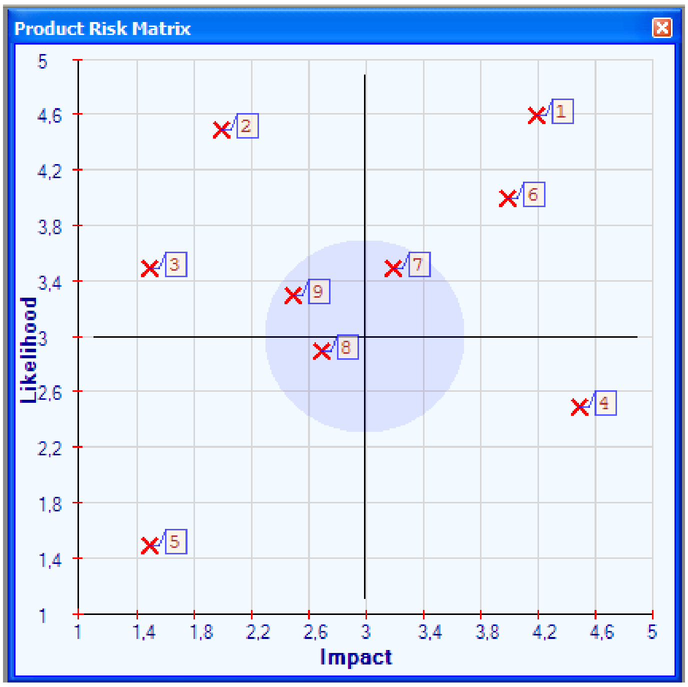
Why use test case design techniques at all? Maybe the following will convince you of their value:

- Objectivity; guarantees a certain level of coverage linked to an identifiable process of achieving it
- Increase in the defect finding capability (recent surveys indicate up to 25% more defects are found when testers are trained and use test design techniques)
- The ability to reproduce tests
- Building testware to support long-term system maintainability and test automation.

Test techniques provide an understanding of the complexities imposed by most systems. The use of techniques forces testers into thinking about what they test and why they are testing it. This is also for those who use exploratory testing; professional exploratory testers use the principles of test design techniques to draft their test ideas. In many cases, techniques identify a level of coverage that would otherwise be a mystery. Remember techniques will not provide a mechanism to exercise 100% test coverage. However, if information regarding the level of coverage to be achieved is available, then objective decisions can be made based on the risk of the system under test and on which tests to leave out.

Selecting techniques

The decision on which test design techniques are to be used and when is largely dependent on the understanding of the risk and in which component parts of the system it lies. Risk identification, assessment, management and mitigation are a fundamental part of the testing function. The identification and assessment of risk provides a focus to the testing, the management of risk is included as a part of the test process controls, and mitigation is achieved when the correct coverage of tests is performed. Often the test items are visualized in a risk matrix that identifies the most critical test items either from an impact (business) or likelihood (technical) perspective.



Example of a product risk matrix (from the PRISMA tool)

The choice of which test techniques to use also depends on a number of other factors, including the type of system, regulatory standards, customer or contractual requirements, test objective, documentation available, knowledge of the testers, time and budget, development life cycle, use case models and previous experience of types of defects found. Some techniques are more applicable to certain situations and test levels; others are applicable to all test levels.

Perhaps the single most important thing to understand is that the best testing technique is no single testing technique! Because each testing technique is good at finding one specific class of defect, using just one technique will help ensure that many (perhaps most, but not all) defects of that particular class are found. Unfortunately, it may also help to ensure that many defects of other classes are missed! Using a variety of techniques will therefore help ensure that a variety of defects are found, resulting in more effective testing.

Test modeling

It is readily agreed that by painting a picture or drawing a diagram thought processes can be explained to others, and perhaps ourselves, more clearly. The vast majority of formal dynamic test techniques identified in BS7925-2 utilize one method or other to model the system component under test. This is not a coincidence. Modeling software is often considered the most complete method of establishing paths, decisions, branches and statements. The objective of formal test techniques is not just to provide an objective coverage measure, but also to document the decision making process, whilst making the component clear to both technical and non-technical staff. Testers must ensure they know what they are going to test from the possible options, and perhaps more importantly they must be able to identify which situations are not going to be executed. The level and detail of the modeling is dictated by the technique chosen. Testers must ensure they use these models to the best advantage to achieve a deeper understanding of how the system works.

Advantages / Disadvantages

So should testers use formal and informal test case design techniques? The decision to use or not use test case design techniques is directly related to the risk of the products being tested. Initially, a basic view of both the advantages and disadvantages must be taken [5].

Advantages	Disadvantages
Objectivity	Require training to some degree
Formal coverage measures	Time to implement – culture change
Early defect finding	Buy-in needed for everyone
Traceability	Not seen as useful for all types of application
Coverage independent of the tester	Takes more time than less formal test design
Way to differentiate test depth based on risks using different techniques	Does not cover all situations (error guessing will still be useful)
High level of re-use (re-usable testware)	Little use of domain knowledge of tester
Repeatability and reproduceability	
Audit trails	
Higher defect finding capability	

Whether testers like it or not, test case design techniques, whether formal or not, are an integral part of everyday testing life. If testing is to keep up in these ever-changing times, the testing discipline must ensure the validity of its function by continuing to be objective. In order to remain objective, the level of testing must be measurable. The only measure available is coverage, and only formal test case techniques and static testing methods provide this framework. If testing is to be taken seriously, the testing function must take itself seriously and ensure its objective is to make any testing objective. So should formal and informal test case design techniques be used? In one word: "YES".

Recommend literature

[1] Beizer, B. (1990), Software Testing Techniques, 2nd edition, Van Nostrand Reinhold, ISBN 1-850-32880-3

[2] BS 7925-2 (1998), Software Component Testing, British Standards Institution

[3] Copeland, L. (2003), A Practitioner's Guide to Software Test Design, Artech House Publishers, ISBN 1-58053-791-X

[4] Pol, M., R. Teunissen and E. van Veenendaal (2002), Software Testing; A Guide to the TMap approach, Addison-Wesley, ISBN 0-201-74571-2

[5] Veenendaal, E. van (2005), The Testing Practitioner, UTN-Publishing, ISBN 90-72194-65-9

Erik van Veenendaal is a leading international consultant and trainer, and recognized expert is the area of software testing and quality management. He is the director of Improve Quality Services BV. At EuroStar 1999, 2002 and 2005, he was awarded the best tutorial presentation. In 2007 he received the European Testing Excellence Award for his contribution to the testing profession over the years. He has been working as a test manager and consultant in software quality for almost 20 years.
He has written numerous papers and a number of books, including "The Testing Practitioner", "ISTQB Foundations of Software Testing" and "Testing according to TMap". Erik is also a former part-time senior lecturer at the Eindhoven University of Technology, the vice-president of the International Software Testing Qualifications Board and the vice chair of the TMMi Foundation.