

# te testing experience

The Magazine for Professional Testers

printed in Germany

print version 8,00 €

free digital version

[www.testingexperience.com](http://www.testingexperience.com)

ISSN 1866-5705

© iStockphoto.com/abu



## Model-Based Testing

## Beware!... Model-based testing

by Erik van Veenendaal

### What it should not be!

Having been in the software testing profession for over 20 years, reading or hearing about model-based testing has always given me a sense of warning. Too often I have been disappointed over the years, great papers and enthusiastic test professionals (and experts) presenting the latest about model-based testing.... Often presented as a silver bullet, no more test design techniques and activities, test cases would be generated and of course automated. Great savings and benefits promised, the world would change and traditional methods were no longer needed. Practice was always very different from theory, and I hardly ever saw their ideas being applied in projects. It stems from these experiences in the recent and far past, that I'm always very cautious when I hear people presenting ideas for model-based testing.

A definition in line with the above:

***“Model-based testing (MBT) is the automatic generation of software test procedures, using models of system requirements and behavior. Although this type of testing requires some more up-front effort in building the model, it offers substantial advantages over traditional software testing methods.”***

Of course, the definition comes with all these great advantages that will be achieved:

- Project maintenance is lower. You do not need to write new tests for each new feature. Once you have a model, it is easier to generate and re-generate test cases than it is with hand-designed test cases.
- Design is fluid. When a new feature is added, a new test action is added to run in combination with existing test actions. A simple change can automatically ripple through the entire suite of test cases.
- High coverage. Tests continue to find bugs, not just regressions due to changes in the code path or dependencies.
- Model authoring is independent of implementation and actual testing, so that these activities can be carried out by different members of a team concurrently.

You may wonder why we are not all doing this. Just apply MBT and all testing problems will go away.

### What it should be!

I hope I did not offend too many test professionals with the first section. I hope, however, that I have alerted most of you. Of course, model-based testing is a great idea, but it is not a silver bullet (as nothing ever is) and some discussion is needed to put things into perspective.

A much better definition can be found on Wikipedia:

***“Model-based testing is the application of model-based design for designing and optionally executing the necessary artifacts to perform software testing. Models are used to represent the desired behavior of the System Under Test (SUT).”***

### Model-based testing ≠ Test automation

Note that in the second definition the word automation is not present. This puts things into perspective. Model-based is all about using (formal) models, e.g., UML based, to design test cases. The model is a tool to better understand the SUT, and will be used as a starting point to design test cases. Of course, some formal models allow for automation and generation of test cases, but this is not a mandatory part of model-based testing. And, as we all know, automation is never as easy as it looks in demos.

### All testing is model-based

If one takes a step back, isn't every requirements specification (whether formatted using formal models or just plain text) a model of the desired behavior of the SUT? When designing test cases, we use the requirement to establish our test conditions and test cases. So in fact requirements-based testing (as explained in many traditional methods) is also model-based testing. This puts things into perspective.

### Large up-front investment

Model-based testing requires a substantial larger up-front effort. Of course, we all know up-front investments will pay off later, but somehow this is always very difficult to sell to management. It is easy to say, but usually very difficult to implement. A parallel can be made with reviews. We all know they pay off later, but recent

studies show that after all these years only 25% of organizations practice reviews in a professional way. This puts things into perspective.

### **Exploratory testing is not covered**

Many model-testing “experts” claim it replaces all other testing. I believe that over the last decade we have learned that experienced-based testing, e.g., exploratory testing, is a very powerful tool to find defects. In some projects it is “just” an add-on to more formal test design techniques, and in some projects it is even the only testing strategy being applied. Model-based testing and exploratory testing can be combined, but come from a very different way of thinking. It’s not all about processes, people are a key to success as well.

There are of course many more issues to be discussed in the context of model-based testing, each with its advantages and disadvantages. I’m sure you will find many of these in this issue of Testing Experience.

Good luck with your model-based testing, but beware.....

## > biography



### **Erik van Veenendaal**

([www.erikvanveenendaal.nl](http://www.erikvanveenendaal.nl)) is a leading international consultant and trainer, and a widely recognized expert in the area of software testing and quality management with over 20 years of practical testing experiences. He is the founder of Improve Quality Services BV ([www.improveqs.nl](http://www.improveqs.nl)). At Euro-Star 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 various domains for more than 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, vice-president of the International Software Testing Qualifications Board (2005–2009) and currently vice chair of the TMMi Foundation.

## **Erik van Veenendaal**

### **Practical Risk-Based Testing The PRISMA<sup>®</sup> Approach**



**UTN**  
Publishers

## **NEW PUBLICATION**

**PRISMA** is an approach for identifying the areas that are most important to test, i.e., identifying the areas that have the highest level of business and/or technical risk. The PRISMA Approach provides a practical, realistic and effective way to introduce and maintain a testing approach that manages product risk, provides guidance to the testing effort and supports ongoing process improvement efforts. Risk identification, risk analysis, risk mitigation and risk monitoring (with a special focus on reporting) are explored and presented in this book.

**Erik van Veenendaal** is a widely-recognized expert in software testing and quality management, a leading international consultant and trainer with over 20 years of practical experiences, and the founder of Improve Quality Services BV.

ISBN 9789490986070  
pages: 136  
price € 19.90.  
Order at [www.utn.nl](http://www.utn.nl)

“As the title indicates, this is a practical and straightforward approach to risk-based testing that addresses the real world needs to efficiently apply a testing methodology that satisfies the needs of the stakeholders, creates an understandable road map and allows the testers to apply their unique knowledge and experience. The book is easy to read, provides clear examples and leaves you excited to apply what you’ve learned.”

**Judy McKay**