WHITE PAPER | MAY 2016

Models at Heart

In a series of articles Paul Gerrard, a testing guru and consultant, discusses a range of testing topics. Test models are fundamental to testing and, in this article, Paul talks about the art of creating and using models. If testers are "shifting left," pairing with developers or at least working more closely with developers, testers (and developers) need to be able to create models, learn how to articulate and share them, and support better collaboration.

Paul Gerrard
Gerrard Consulting

Sponsored by
ca technologies

Test Design is Based on Models

In this article, I want to discuss the most important concept in testing—the art (or is it a science?) of creating and using models. Boris Beizer said in 1990[1]:

> "Testing is a process in which we create mental models of the environment, the program, human nature, and the tests themselves. Each model is used either until we accept the behaviour is correct or until the model is no longer sufficient for the purpose."

Test design is the process by which we select, from the infinite number possible, the tests that we believe will be most valuable to us and our stakeholders. Our test model helps us select tests in a systematic way. Test models are fundamental to testing and the rest of this article discusses their importance and use.

Fundamental Test Process

In the New Model for Testing[2], I suggest that, at the most fundamental level, all testing can be described thus:

1. We identify and explore sources of knowledge to build test models.
2. We use these models to challenge and validate the sources.
3. We use these models to inform (development and) testing.

I make a distinction between exploration and testing. The main difference from the common testing view is that I will use the term exploration to mean the elicitation of knowledge about the system to be tested from sources of knowledge.

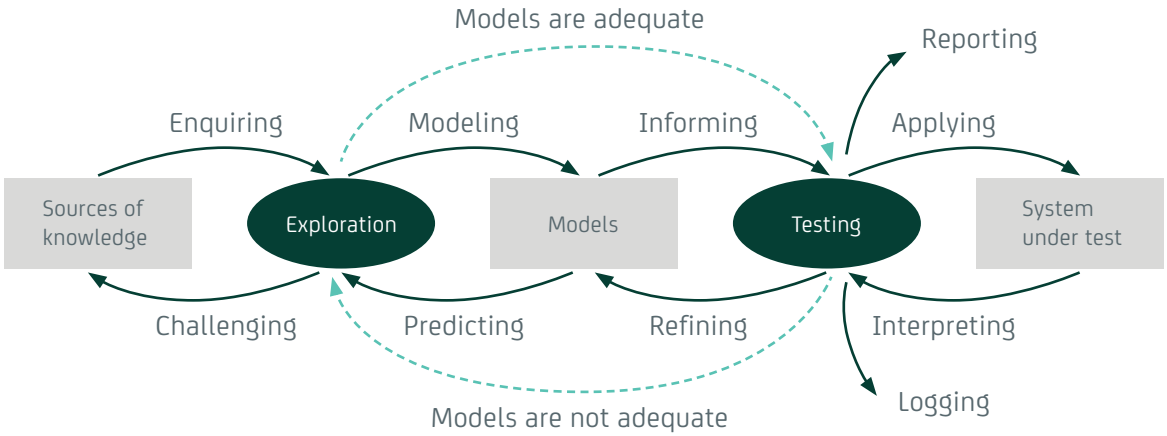

Figure A.

New Model for Testing: Models at the heart of the tester's thinking processes

Sources of Knowledge

We build our models from information that we elicit from sources of knowledge. Given a mission for testing, our first task is to identify these sources. These sources of knowledge might be:

- **Documentation**: specifications, designs, requirements, standards, guidelines and so on
- **People**: stakeholders, users, analysts, designers and developers and others
- **Experience**: your own knowledge and experience of similar (or dissimilar) systems, your preferences, prejudices, guesses, hunches, beliefs and biases
- **System**: the system under test, if it exists, is available and is accessible

We gather information from our sources of knowledge to derive models that we use to challenge our sources and design and/or test our systems.

All of our sources of knowledge are fallible and incomplete and so are our models. Testers use experience, skill and judgement to sift through these sources, to compare and contrast them, to challenge them, to arrive at consensus. These capabilities are normally associated with systems or business analysts, of course.

What is a Test Model?

A test model might be a checklist or set of criteria; it could be a diagram derived from a design document or an analysis of narrative text. Many test models are never committed to paper—they can be mental models constructed specifically to guide the tester whilst they explore the system under test.

We use test models to:

- Simplify the context of the test. Irrelevant or negligible details are ignored in the model.
- Focus attention on a particular aspect of the behaviour of the system. These might be critical or risky features, technical aspects or user operations of interest, or particular aspects of the construction or architecture of the system.
- Generate a set of unique (within the context of the model) tests that are diverse (with respect to that model).
- Enable the testing to be estimated, planned, monitored and evaluated for its completeness (coverage).

From the tester's point of view, a model helps us to recognise particular aspects of the system that could be the subject of a test. The model focuses attention on areas of the system that are of interest.

An Example

Suppose we want to test how a car (an automatic gearshift model) accelerates from rest to its top speed and check that it meets our performance objective (e.g., from a standing start to 60 mph in 8 seconds). We might model this system as:

1. A mass (of the whole vehicle and driver) acting at a defined centre of gravity—which accelerates according to Newton's second law
4. A power source (the engine) having a power output varying from a minimum to a maximum value dependent on the gas pedal position
5. A gas pedal or accelerator that can have a variable position
6. Formulae that relate the gas pedal position, power output and acceleration

We can probably obtain all the information we need for our model from the design document for the car.

Using the model, we could design a test like this: "From rest, set the pedal to maximum power for a period of ten seconds. Use our formulae to calculate a predicted speed for every second of acceleration. Compare the actual speed with predicted speed every second of the test."

When we conduct the test in a real car we compare its speed at every second to that predicted by the model. In this way, we could determine whether the car meets its performance objective. For example, if the system under test (the car) does not reach the required speed at each measurement point in the test, we either change the car, or we change the model (our interpretation of the car's design), or the design itself.

Everything looks fine—doesn't it?

Models Over-Simplify, So Use More Than One

But in the real test, our car may not behave as we expect because our model ignores several key aspects of the car's behaviour and context. We might reasonably ask:

- Would a real driver be as aggressive or gentler with the gas pedal?
- What is the wind speed and direction?
- What are the road conditions (wet, dry, tarmac, dirt, etc.)?
- What load is the car carrying, beyond the driver?
- Is the car on a level road, an uphill or downhill incline?
- What is the power efficiency of the system?

Our model is grossly simplified, incorporates many implicit assumptions and would need significant refinement to be an accurate representation of a real car under test. All models simplify the context of tests to varying extent, so we normally use several models to broaden our view and coverage (referred to as 'diverse half-measures'[3]). The challenges are to select models that are an accurate enough representation of our system under test and to interpret the test outcomes obtained with care.

In general all test models, even those proposed by textbooks, are heuristic in that they are useful in some situations but are always incomplete and fallible. Before we adopt a model, we need to know what aspects of the behaviour, design, modes of failure or usage patterns the model helps us to identify and what assumptions and simplifications it (explicitly or implicitly) includes.

Our Brains are Fantastic Modelling Engines

Our brains are capable of modelling and remodelling our surroundings. To achieve something as 'simple' as walking, our brain needs to understand, second by second (and faster than that) the configuration of the major bones in our body. It needs to understand the tensions in around one hundred muscles. To take a single step our brain needs to send impulses to all these muscles to work in highly complex patterns that enable us to move and reach our destinations without bumping into things. Our brain is simply not fast enough to process all this information. It uses models to simplify and manage this challenge. Modelling and visualisation are innate, essential skills required in all of our everyday lives. The BEST robots on the planet are still quite crude in comparison.

You must have seen golfers practicing their swing before they take a shot. They rehearse and visualise the shot, the trajectory of the ball and the target. In many sports, coaches film athletes and talk them through their movements in great detail helping them to visualise so they can control their movement, often under great physical stress. Athletes consciously model their world to achieve perfection or control and some call it "the zone."

But models don't only represent physical movement. For example, when Stephen Hawking lost physical capabilities, he invented a collection of powerful mental tools—models—that allowed him to carry on working with his physics, without using a blackboard or written formulae.

We use this same modelling skill to develop and test systems. Our brains are incredibly sophisticated and fast modelling engines, and mental modelling dominates our thinking.

Models at the Heart of Testing

I suggest[2] that developer and tester exploration and modelling really are quite similar. If testers are "shifting left," pairing with developers or at least working more closely with developers, testers (and developers) need to be able to create models, learn how to articulate and share them and support better collaboration.

In a blog[4], "Courage and Ambition in Teaching and Learning," I suggest that the way we teach test design needs a rethink. Test design techniques are most often taught as clerical procedures. We need to teach testers how to create models, to appreciate them, to select them, and discard them. We need to teach the models that underpin the techniques, not just the procedures to implement them.

Thinking and modelling are at the heart of what testers do. The scale, complexity and rates of change in Digital Systems and the Internet of Things are all on the rise. A new way of thinking and the acquisition and refinement of modelling skills must be our top priorities if we are to cope with these challenges.

About the Author

Paul Gerrard is a consultant, teacher, author, webmaster, developer, tester, conference speaker, rowing coach and a publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them.

Educated at the universities of Oxford and Imperial College London, in 2010, Paul won the Eurostar European Testing excellence Award and, in 2013, won The European Software Testing Awards (TESTA) Lifetime Achievement Award.

In 2002, Paul wrote "Risk-Based E-Business Testing" with Neil Thompson. Paul wrote "The Tester's Pocketbook" in 2009. Paul co-authored "The Business Story Pocketbook" with Susan Windsor in 2011 and wrote "Lean Python" in 2014.

In 2014, Paul was the Programme Chair for the EuroSTAR Conference in Dublin.

He is Principal of Gerrard Consulting Limited, Director of TestOpera Limited and is the host of the Test Management Forum.

Mail: paul@gerrardconsulting.com
Twitter: @paul_gerrard
Web: gerrardconsulting.com

For more information visit **Develop & Test** with CA Technologies.

ca technologies

Connect with CA Technologies at ca.com

CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate – across mobile, private and public cloud, distributed and mainframe environments. Learn more at **ca.com**.

References

1. Software Testing Techniques", Boris Beizer, 1990
2. "A New Model for Testing", Paul Gerrard, http://dev.sp.qa/download/newModel
3. "Lessons Learned in Software Testing", Kaner, Bach, Pettichord, 2002

CS200-200386_0516