

THE TESTER

SIGiST Spring Conference Wednesday 4th March 2014

52

IN THIS ISSUE

Conference

Agenda2

Venue3

Abstracts.....4

Articles

Extended OATS
.....7

Regular Features

Testing Events
Listing 12

From the Editor

Welcome to the spring edition of The Tester. For the first SIGiST conference of 2015 we return to a previous SIGiST venue of the Royal College of Obstetricians and Gynaecologists (RCOG), near Regents Park in central London. See page 3 for details on how to get there.

As usual for the conference, we have a mixture of UK and international speakers, and the abstracts for their presentations are available from page 4. Again we have a “double-session” workshop, “Defect Measurement and Analysis”, run by the United Kingdom Software Metrics Association.

We are always looking for speakers / workshops for the conference, and articles for The Tester. If you want to speak check out the SIG website: <http://www.bcs.org/category/10880> or contact me if you want to become a published author. Check out the Extended OATS article relating to pair-wise interactions in this issue.

Phill Isles
The Tester Editor
phill.isles@bcs.org

Conference Booking Instructions

To register online, please use the link below, or scan the QR code with your smart device. Please note the BCS booking system accepts multiple and third party bookings.

<https://events.bcs.org/book/1277/>



Join our Linked-In Group:

<http://www.linkedin.com/groups?mostPopular=&gid=3466623>

Follow us @SIGiST



Conference Agenda

BCS SIGiST - Spring 2015 Conference			
Wednesday 4th March 2015			
Royal College of Obstetricians and Gynaecologists, 27 Sussex Place, Regent's Park, London, NW1 4RG.			
08:45	Coffee & Registration, Exhibition opens		
09:40	Introduction and Welcome Stuart Reid, SIGiST Chair		
09:45	Opening Keynote		
	RIP Testing - 2018 Chris Ambler, Head Of Testing, Capita Customer Management		
10:45	Open Microphone and Networking session		
11:00	Tea / coffee break		
11:30	<table border="1"> <tr> <td> Continuous Quality Engineering in a Digital World Paul Baker, Head of Quality and Test Engineering / Assurance for Digital Services, Visa Europe </td> <td> Morning Workshop Defect Measurement and Analysis United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i> </td> </tr> </table>	Continuous Quality Engineering in a Digital World Paul Baker, Head of Quality and Test Engineering / Assurance for Digital Services, Visa Europe	Morning Workshop Defect Measurement and Analysis United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>
Continuous Quality Engineering in a Digital World Paul Baker, Head of Quality and Test Engineering / Assurance for Digital Services, Visa Europe	Morning Workshop Defect Measurement and Analysis United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>		
12:30	Lunch break Opportunity to visit the Vendor Exhibition		
13:30	<table border="1"> <tr> <td> The Pillars of Agile Testing David Evans, Partner, Neuri Consulting </td> <td> Afternoon Workshop Defect Measurement and Analysis (continued) United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i> </td> </tr> </table>	The Pillars of Agile Testing David Evans, Partner, Neuri Consulting	Afternoon Workshop Defect Measurement and Analysis (continued) United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>
The Pillars of Agile Testing David Evans, Partner, Neuri Consulting	Afternoon Workshop Defect Measurement and Analysis (continued) United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>		
14:30	<table border="1"> <tr> <td> UAT - that's what you do with what's left of your project, right?! Roy Dalglish, TSG </td> <td> United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i> </td> </tr> </table>	UAT - that's what you do with what's left of your project, right?! Roy Dalglish, TSG	United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>
UAT - that's what you do with what's left of your project, right?! Roy Dalglish, TSG	United Kingdom Software Metrics Association (UKSMA) Kristina Rungano Masuwa-Morgan <i>et al.</i>		
15:30	Tea / coffee break		
16:00	Closing Keynote		
	Don't Get SMACked How Social, Mobile, Analytics and Cloud technologies are reshaping QA and testing John Fodeh, Practice Lead - Quality Engineering & Assurance - Nordics, Cognizant		
17:00	Closing Remarks Stuart Reid, SIGiST Chair		

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops will have limited places.

SIGiST Conference Venue



Royal College of
Obstetricians &
Gynaecologists



photo credit: RCOG

For the March 2015 conference, the SIGiST returns to a previous venue, the Royal College of Obstetricians and Gynaecologists, just off Regent's Park, London. With its excellent lecture theatre and facilities, RCOG is the perfect place for the SIGiST conference. Full details of how to get to RCOG can be found on their website: <https://www.rcog.org.uk/en/contact-us/directions/>.

Baker Street underground is the nearest station to RCOG. Exit the station and turn right on to Baker Street. Continue walking for 200m towards Regent's Park. Follow the road around to your left on to the Outer Circle. Walk past the first entrance to Sussex Place until you reach the next entrance. Turn left into Sussex Place and the College will be on your right.

Presentation Abstracts and Speaker Biographies

Chris Ambler

“RIP Testing - 2018”

Based around the future of testing and the future of testers. I will discuss my prediction that the testing role as we know it will no longer exist by 2018 but will 'morph' into a role of Business Transformist working in a different way to support the stakeholders with 'getting it right first time'. Looking at current methods against future methods', the Internet of things and what needs to be done to improve successful timely delivery.

Testing is dead. Long live testing!

Chris Ambler is Head of Testing at Capita Customer Management.

Paul Baker

“Continuous Quality Engineering in a Digital World”

Paul Baker is Head of Quality and Test Engineering / Assurance for Digital Services, Visa Europe.

UKSMA (Kristina Rungano Masuwa-Morgan *et al.*)

“Defect Measurement and Analysis”
[Workshop]

Promote defects measurement as part of United Kingdom Software Metrics Association's (UKSMA) mission of promoting software metrics and measurement.

Familiarise the software community with UKSMA's “Defect Measurement and Analysis Handbook”.

Engage with the professional community on the priorities for defects measurement.

This workshop and the UKSMA Defect Measurement and Analysis Handbook offer a unique opportunity to explore strategies and techniques for managing software defects and dealing with their impact on delivery costs and schedules.

David Evans

“The Pillars of Agile Testing”

The craft of agile testing is maturing, and we have plenty of skilled and gifted testers in the industry, honing their skills, sharing experiences and techniques. But do you find that you have trouble making the

success of individual testers scale from the few to the many? Does their great testing get diluted by the mediocrity of average testing in the organisation?

For your organisation to succeed with agile testing, you have to understand these things:

- What goals you are trying to achieve
- The key measures or indicators for those goals
- The prerequisites for success
- The dependencies and connections between all of the above.

In this talk I will share my “Pillars of Testing” model, as featured in Lisa Crispin and Janet Gregory’s new book “More Agile Testing”. The model sets out the structures and dependencies that influence how effective our overall test effort can be in the organisation as a whole.

David Evans is an experienced agile consultant, coach and trainer with over 25 years of IT experience. A thought-leader in the field of agile quality, he has provided training and consultancy for clients worldwide. A regular speaker at events and conferences across Europe, David was voted Best Keynote Speaker at Agile Testing Days 2013. He is co-author of the best-selling book “50 Quick Ideas to Improve your User Stories”, and has also had several papers published in international IT journals. He currently lives and works in the UK, where he is a partner along with Gojko Adzic in Neuri Consulting LLP. He can be reached at david.evans@neuri.co.uk on email and @DavidEvans66 on Twitter.

and yet is also arguably the most significant testing stage in driving the secure and successful implementation of your system, while safeguarding the operability of your business.

With this in mind, blended from more than seventeen years’ experience working in UAT, I will be offering some insights to the testing we actually do at the end of projects, that would be far safer and far more effective done somewhere earlier in the life cycle. We’ll also take a look at some testing activities you may never have considered, that really should be part of your UAT. And we’ll even take a look at how to sell your plan to senior management along the way. That way, you can turn your UAT from an exercise on closing the gate on an already bolted horse, into a key, valuable protection for your business operability in the face of changing systems.

Roy Dalgleish. After a promising start as a psychopharmacologist, I realised that my bedside manner was better suited to working with computer systems than actual live patients. Since making the career change, I’ve spent more than half my working life testing systems, designing how to test systems, teaching other people how to test systems, and explaining to companies why their systems needed testing. Much of that time, I’ve spent delivering efficient, accessible airline baggage and passenger management systems, driving safety and security for passengers and top class business processes for airlines. Lately, though, I’ve been working in banking regulation systems. It turns out that it’s all much the same when you get down to it - making sure that folks lives and money are kept safe.

Roy Dalgleish

“UAT - that's what you do with what's left of your project, right?!”

UAT is probably one of the most abused and most misunderstood stages of testing,

John Fodeh

“Don’t Get SMACkEd - How Social, Mobile, Analytics and Cloud technologies are reshaping QA and testing”

With a revolution witnessed in social media, mobile communications, cloud solutions as well as the additional capability to process information through analytics, our markets and business models are being impacted in many different ways. As a result, IT organizations are now expected to deliver changes to the implemented as well as new functionality, fast - sometime within days or even hours.

What does this mean for us as testers? Can we adjust our existing models to cope with that or does it require us to rethink our models?

This presentation addresses the challenges and constraints imposed by SMAC as well as the opportunities it brings. It will also describe the concept of "Code Halo", the digital identity we produce with every digital click, swipe, "like", buy, comment and search. Code Halos are becoming increasingly vital to the success of businesses and are calling for new ways for doing our testing.

- The technology shift – what does it mean to the business? What does it mean to testing?
- The need for a well-orchestrated strategy
- The role of the tester in the new world

***John Fodeh** John Fodeh has more than 18 years of experience in software testing with extensive expertise in quality management, process improvement, testing tools, and strategy. At Cognizant, he heads the Quality Engineering and Assurance practice in the Nordics and is a part of company's global QE&A Think Tank team. John has written a number of articles and book contributions. He is an active member of special interest groups in software testing, chairman of the Danish Software Testing Board and speaker at various testing conferences.*

SIGiST White Paper Scheme

We have set up a new area on the BCS website of a searchable repository for white papers and articles on testing and we are looking for contributors. That means you!

Do you have an existing paper you would like to repurpose and make more widely available through the SIGiST website?

- Then please send us the paper with three keywords for searching.

Would you like to write a new paper?

- Please send us the title and abstract together with the three keywords (or phrases)
- We will review the proposal and guide you through the authoring process
- For those who are thinking of speaking at SIGiST then this might be a good way to prepare a talk and get some useful feedback

If you have been thinking of writing or publicising an existing paper then this is the ideal opportunity. Please email your existing paper (with keywords) or your proposal to The Tester Editor, phill.isles@bcs.org

Past articles from The Tester will slowly be added to the repository as well.

Follow this link to the repository: <http://www.bcs.org/category/18128>

Extended OATS

Optimum test coverage and increase defect removal efficiency

Shajahan Pulikkal, UST Global

Abstract

The Orthogonal Array Testing Strategy (OATS) is a systematic, statistical way of testing pair-wise interactions, reducing the number of combinations and providing maximum coverage with a minimum number of test cases. This paper introduces OATS, and provides an analysis on the scope of OATS in the current QA world. It then proposes an automated testing tool 'Extended OATS' to overcome the limitations of OATS and to resolve the most common challenge faced in the testing community – ensuring 'optimum test coverage and increase defect removal efficiency'.

Introduction

Problem Statement

The most common challenge faced in the testing community is to ensure for optimum test coverage and increase defect removal efficiency.

Pairwise Testing

Pairwise testing aims to catch potential bugs that are present when specific pairs of items cause an issue, e.g. A web form may work fine using Firefox and the form may work fine if the user selects England as the location. But it may have an error if both Firefox is used and England is selected. This pair causes an error where neither alone causes an error.

The OATS Theory ⁽¹⁾

The Orthogonal Array Testing Strategy (OATS) is a systematic, statistical way of testing pairwise interactions. OATS can be used to reduce the number of combinations

and provide maximum coverage with a minimum number of test cases. OATS is an array of values in which each column represents a variable / factor that can take a certain set of values called levels. Each row represents a test case. In OATS, the factors are combined pairwise rather than representing all possible combinations of factors and levels. Orthogonal arrays are two dimensional arrays of numbers which possess the interesting quality that by choosing any two columns in the array you receive an even distribution of all the pairwise combinations of values in the array.

This technique is very useful for finding a small set of tests (from a large number of possibilities) that exercises key combinations. Orthogonal array-based solutions aim for the same coverage goal as pairwise solutions. OATS also provides a uniform distribution throughout the domain.

OATS in Practice

The below example describes a practical scenario using OATS.

Consider three parameters A, B and C.

Parameters A, B and C each have positive values 1, 2, and 3. Testing all combinations of the three parameters would involve executing a total of 27 (3x3x3) test cases.

With the way programming works, a fault will most likely depend on the values of two parameters, not three. The fault might occur for each of these three test cases:

A=1, B=1, C=1, A=1, B=1, C=2, and A=1, B=1, C=3.

All possible pair-wise combinations between parameters A and B, B and C, and C and A can find the fault among them. It is not necessary to run all 27 scenarios, 9 test cases would be sufficient.

Limitations of OATS

- OATS has limitations when the factors are dependent. OATS specifies that the factors have to be independent.
- OATS does not have an option to exclude invalid combinations based on project requirements.
- Arrays can be difficult to construct and it can be overwhelming to use in manual testing.

Approach

OATS needs to be redefined to overcome the limitations on factor dependency and to generate optimum test coverage by excluding all invalid combinations. Optimum pairwise combinations need to be created based on the factors and their values. Creation of pairwise combinations will be difficult for high numbers of factors and a huge amount of manual effort will be required for this.

An automation script needs to be implemented to replace the manual creation of pairwise combinations. 'Extended OATS' generated pairwise combinations will be in a test case skeleton format. Test cases can then be easily defined from the 'Extended OATS' generated test case skeleton.

Proposed Solution – 'Extended OATS'

A practical testing scenario is used to describe the proposed solution.

Test Requirement

We have three factors (Browser, Region and Device) to be tested and each factor has three different values.

Browser	Region	Device
IE	.CO.UK	DESKTOP
Firefox	.COM	IPAD
Google Chrome	.CO.IN	GALAXY TAB

Figure 1 – Test requirement factors

For 100% test coverage, normal system testing would be performed with all possible combinations of factors and values. For the above test requirement a total of 27 test cases needs to be executed to make sure 100% test coverage. Figure 2 lists the normal test case skeleton for all possible 27 test cases.

	Browser	Region	Device
1	IE	.CO.UK	DESKTOP
2	IE	.CO.UK	IPAD
3	IE	.CO.UK	GALAXY TAB
4	IE	.COM	DESKTOP
5	IE	.COM	IPAD
6	IE	.COM	GALAXY TAB
7	IE	.CO.IN	DESKTOP
8	IE	.CO.IN	IPAD
9	IE	.CO.IN	GALAXY TAB
10	Firefox	.CO.UK	DESKTOP
11	Firefox	.CO.UK	IPAD
12	Firefox	.CO.UK	GALAXY TAB
13	Firefox	.COM	DESKTOP
14	Firefox	.COM	IPAD
15	Firefox	.COM	GALAXY TAB
16	Firefox	.CO.IN	DESKTOP
17	Firefox	.CO.IN	IPAD
18	Firefox	.CO.IN	GALAXY TAB
19	Google Chrome	.CO.UK	DESKTOP
20	Google Chrome	.CO.UK	IPAD
21	Google Chrome	.CO.UK	GALAXY TAB
22	Google Chrome	.COM	DESKTOP
23	Google Chrome	.COM	IPAD
24	Google Chrome	.COM	GALAXY TAB
25	Google Chrome	.CO.IN	DESKTOP
26	Google Chrome	.CO.IN	IPAD
27	Google Chrome	.CO.IN	GALAXY TAB

Figure 2 – Test case skeleton with 100% test coverage

How 'Extended OATS' works

The steps below describe the implementation of 'Extended OATS'. Firstly all possible combination of two pair values will be created among different test factor values. This will be a reasonable number of records, since all the combinations will be on a two factor level.

In the next step, we need to identify the mappings between each record on different two pair combinations. Each record in the below pair table (Figure 3) will have a mapping record in another pair table. The

correlated records will be combined to make the final pairwise combination on all the factors. All two pair combination should be unique on the final result. e.g. 2 pair combination IE + .CO.UK, IE + DESKTOP are listed only once in the final result (Figure 4).

This will reduce the test case count from 27(3x3x3) to 9 (3x3).

All possible pairwise combinations between parameters 'Browser' and 'Region', 'Region' and 'Device', and 'Device' and 'Browser' are identified.

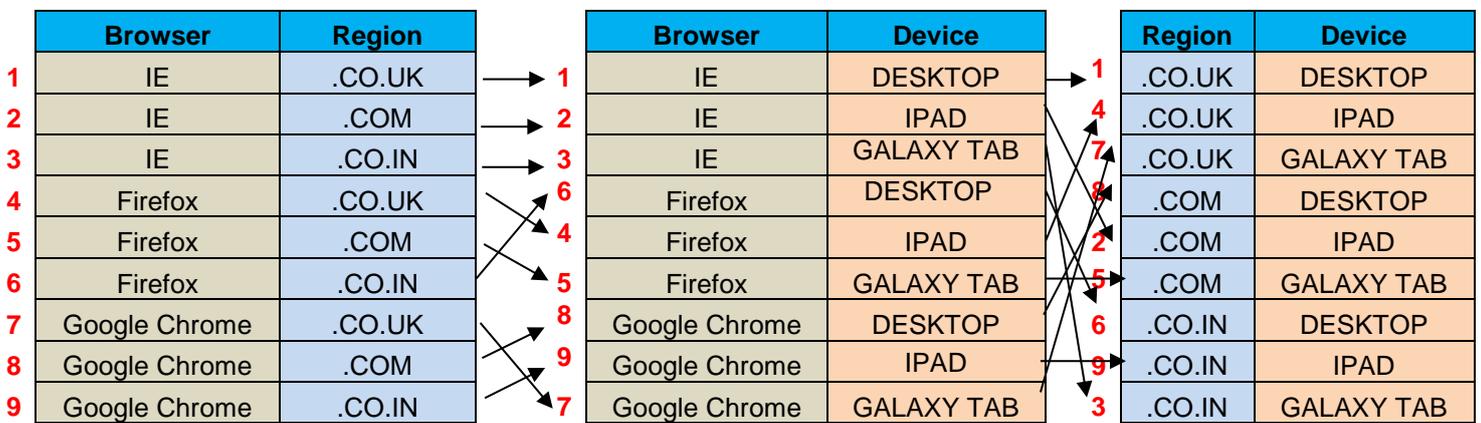


Figure 3 – Extended OATS Process flow diagram

	Browser	Region	Device
1	IE	.CO.UK	DESKTOP
2	IE	.COM	IPAD
3	IE	.CO.IN	GALAXY TAB
4	Firefox	.CO.UK	IPAD
5	Firefox	.COM	GALAXY TAB
6	Firefox	.CO.IN	DESKTOP
7	Google Chrome	.CO.UK	GALAXY TAB
8	Google Chrome	.COM	DESKTOP
9	Google Chrome	.CO.IN	IPAD

Figure 4 – Extended OATS result – Optimum coverage test case skeleton

The 9 test cases in figure 4 are the most prioritized among all possible 27 test cases. This will provide almost a similar coverage as executing all 27 scenarios.

The rest of 18 test cases needs be executed only if the testing time line permits.

This approach can select 16 test cases with a similar coverage to 64 test cases, for a set of 4 variables with 4 levels of values.

This same concept can be applied to more complex scenarios where testing an application might require 1,000+ test cases. Utilizing Extended-OATS, it can be reduced significantly, down to 100 or less test cases to execute.

How to remove factor dependency in Extended OATS?

Resolve factor dependencies

Invalid combination of factor values can be declared initially and those pairs will be removed while constructing the final pairwise combinations.

Different levels of factor values

‘Extended OATS’ test coverage matrix can be created even when the factor value levels are different.

Example:

We can consider the similar example which we used in Figure 1. Here we have the same three factors and only two values for the factor ‘Device’.

Browser	Region	Device
IE	.CO.UK	DESKTOP
Firefox	.COM	IPAD
Google Chrome	.CO.IN	

Figure 5 – Test requirement with different level factors

1	IE	.CO.UK	1,3	IE	DESKTOP	1,4	.CO.UK	DESKTOP
2	IE	.COM	2	IE	IPAD	7	.CO.UK	IPAD
3	IE	.CO.IN	4	Firefox	DESKTOP	8	.COM	DESKTOP
4	Firefox	.CO.UK	5,6	Firefox	IPAD	2,5	.COM	IPAD
5	Firefox	.COM	8	Google Chrome	DESKTOP	3	.CO.IN	DESKTOP
6	Firefox	.CO.IN	7,9	Google Chrome	IPAD	6,9	.CO.IN	IPAD
7	Google Chrome	.CO.UK						
8	Google Chrome	.COM						
9	Google Chrome	.CO.IN						

Figure 6 – Extended OATS Process flow diagram for different level factors

The test case skeleton can be created in the same way as mentioned in the Figure 3. But few pairs will be repeated to accomplish all possible pairwise combinations between parameters. All test cases will be different from each other even though we used few pairs are repeated.

Test case skeleton by Extended-OATS automation tool

Here few pairs are repeated to accomplish all possible pairwise combinations between parameters. But all test cases are different from each other.

	Browser	Region	Device
1	IE	.CO.UK	DESKTOP
2	IE	.COM	IPAD
3	IE	.CO.IN	DESKTOP
4	Firefox	.CO.UK	DESKTOP
5	Firefox	.COM	IPAD
6	Firefox	.CO.IN	IPAD
7	Google Chrome	.CO.UK	IPAD
8	Google Chrome	.COM	DESKTOP
9	Google Chrome	.CO.IN	IPAD

Figure 7 – Extended OATS result for different level factors

Key benefits of Extended OATS

- **Cost Effective:** To reduce manual effort generally automation would be preferred, but it's a bit costly to implement. Cost of Extended-OATS would be very less
- **Efficiency:** An Efficient application for test case generation that works on any system
- **Increased Productivity:** Reduce manual effort and ensures Quality deliverable
- **Test Coverage:** Guarantees up to 100 % test coverage with minimal test cases.
- **Quality:** When test coverage is at its maximum, Quality of the product is ensured.
- **Time:** Maximum defect removal in less time.

References

(1) Twin Cities Quality Assurance Association (TCQAA); <http://www.tcqaa.org/>



Shajahan Pulikkal has been a test professional for 8+ years and has worked in a number of different sectors ranging from Automation Testing, to SOA & POS Testing. Shajahan is currently working in client site, UK. He provides support for different accounts within UST Global as a SOA testing consultant. Shajahan has developed numerous in-house tools to make testing more fast and accurate with maximum test coverage.

Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on phil.isles@bcs.org

Event Listings 2015

If you would like your event listed here, please contact the Editor on phill.isles@bcs.org

March

SIGiST

4 March 2015
London, UK
<http://www.bcs.org/server.php?show=nav.9264>

Test Bash 2015

27 March 2015
Brighton, UK
<http://www.ministryoftesting.com/training-events/testbash-2015/>

April

The Test Management Summit

28 - 29 April 2015
London, UK
<http://uktmf.com/>

May

STAREAST

3 - 8 May 2015
Orlando, US
<http://stareast.techwell.com/>

Belgium Testing Days

18 - 21 May 2015
Brussels, Belgium
<http://btdconf.com/>

June

SIGiST

5 June 2015
London, UK
<http://www.bcs.org/server.php?show=nav.9264>

September

SIGiST

15 September 2015
London, UK
<http://www.bcs.org/server.php?show=nav.9264>

November

EuroSTAR

2 - 5 November 2015
Maastricht, Netherlands
<http://www.eurostarconferences.com/>

Agile Testing Days

9 - 12 November 2015
Potsdam / Berlin, Germany
<http://www.agiletestingdays.com/>

December

SIGiST

2 December 2015
London, UK
<http://www.bcs.org/server.php?show=nav.9264>





The image shows the BCS Personal Development Plan (PDP) interface on three devices: Desktop/Laptop, Tablet, and Smartphone. The Desktop/Laptop view shows a detailed dashboard with a navigation menu, a main content area with a list of activities, and a sidebar with a search bar. The Tablet view shows a simplified version of the dashboard with a pie chart and a list of activities. The Smartphone view shows a mobile-optimized version of the dashboard with a list of activities and a search bar. A yellow banner in the top right corner says "NEW".

bcS The Chartered Institute for IT

Personal Development Plan

Your portable record of professional aspirations and activities

NEW

Desktop/Laptop

Tablet

Smartphone

Did you get your Personal Development Plan email with suggested potential CPD activities?

The BCS Personal Development Plan (PDP) uptake is going well, with over 1,000 users already actively recording their CPD Development Goals, Activities and preferences. It's not just about recording details though, as there is a Resources section that shows live feeds of potential CPD activities, and a tailored email is sent every 2 months with details of the latest videos, articles, blogs, books and research in your specified field of interest. If you haven't registered yet, you can see the content from the latest PDP bulletin for topics relating to solution development and implementation here <http://www.bcs.org/content/ConWebDoc/50854> or by going to the CPD Portal at: <http://www.bcs.org/pdp/> and selecting the "Give me ideas" link.

The BCS Personal Development Plan is free to use; BCS members can use their Member Secure Area login and password to access it at <https://pdp.bcs.org/>, and non-members can use most of the facilities (using the same link) and registering to create their own user name and password. You can use it on a PC / laptop or compatible tablet PC or smartphone.

THE TESTER

SIGiST Summer Conference Friday 5th June 2015

53

IN THIS ISSUE

Conference

Agenda2

Venue3

Abstracts.....4

Articles

A New Model for
Testing 13

Regular Features

Testing Events
Listing8

EuroSTAR discount

Additional 10%
conference
discount for
SIGiST members
.....9

From the Editor

Welcome to the summer edition of The Tester. As I write this the UK weather is its usual mixture of dry one day and wet the next. Hopefully it will remain dry for the rest of the summer. For the second SIGiST conference of 2015 we move to a new venue, the BCS Offices in central London. The SIGiST conferences will remain at the BCS Office venue for the rest of 2015. See page 3 for details on how to get there.

We have two stunning key-note speakers for the conference: opening with James Lyndsay; and closing with Paul Gerrard. With our recent trend for ‘double-session’ workshops, we have “Using Influence Diagrams to understand testing”, presented by Stuart Ried and Isabel Evans. Places are limited so sign up for the workshop now!

We are always looking for speakers / workshops for the conference, and articles for The Tester. If you want to speak check out the SIG website: <http://www.bcs.org/category/10880> or contact me if you want to become a published author.

Phill Isles
The Tester Editor
phill.isles@bcs.org

Conference Booking Instructions

To register online, please use the link below, or scan the QR code with your smart device. Please note the BCS booking system accepts multiple and third party bookings.

<https://events.bcs.org/book/1278/>



Join our Linked-In Group:

http://www.linkedin.com/groups?mos_tPopular=&qid=3466623



Conference Agenda

BCS SIGiST - Summer 2015 Conference				
Friday 5th June 2015				
BCS, The Davidson Building, 5 Southampton Street, London, WC2E 7HA.				
08:45	Coffee & Registration			
09:40	Introduction and Welcome Stuart Reid, SIGiST Chair			
09:45	Opening Keynote			
	A Nest of Tests James Lyndsay, Workroom Productions			
10:45	Open Microphone and Networking session			
11:00	Tea / coffee break			
11:30	<table border="1"> <tr> <td>Testing business logic... begins in Requirements! Paul Vincent, Sapiens</td> <td>Morning Workshop Using Influence Diagrams to Understand Testing Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)</td> </tr> </table>	Testing business logic... begins in Requirements! Paul Vincent, Sapiens	Morning Workshop Using Influence Diagrams to Understand Testing Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)	
Testing business logic... begins in Requirements! Paul Vincent, Sapiens	Morning Workshop Using Influence Diagrams to Understand Testing Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)			
12:30	Lunch break			
13:30	<table border="1"> <tr> <td>Principles for testing? <i>Presentation and Discussion</i> James Christie, Claro testing</td> <td rowspan="2">Afternoon Workshop Using Influence Diagrams to Understand Testing (continued) Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)</td> </tr> <tr> <td>Testing as a Service: Models Jonathon Wright, Director Testing, Quality & Assurance of Hitachi Consulting</td> </tr> </table>	Principles for testing? <i>Presentation and Discussion</i> James Christie, Claro testing	Afternoon Workshop Using Influence Diagrams to Understand Testing (continued) Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)	Testing as a Service: Models Jonathon Wright, Director Testing, Quality & Assurance of Hitachi Consulting
Principles for testing? <i>Presentation and Discussion</i> James Christie, Claro testing	Afternoon Workshop Using Influence Diagrams to Understand Testing (continued) Stuart Reid (Independent Consultant) and Isabel Evans (Dolphin Computer Access)			
Testing as a Service: Models Jonathon Wright, Director Testing, Quality & Assurance of Hitachi Consulting				
14:30				
15:30	Tea / coffee break			
16:00	Closing Keynote			
	A New Model for Testing Paul Gerrard, Principal, Gerrard Consulting			
17:00	Closing Remarks Stuart Reid, SIGiST Chair			

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops will have limited places.

SIGiST Conference Venue

For the June 2015 conference, and for the rest of 2015, the SIGiST conference moves to the BCS London office. Travel details and location below.

London office guide

How to get to the BCS London office

First Floor
The Davidson Building
5 Southampton Street
London WC2E 7HA
Telephone 01793 417666

These area and inset maps have been simplified in the interests of ease of understanding. Not all roads are shown. The inset map below is more accurate.

Access by car is very difficult due to the local one-way system. There are no car parking facilities at BCS London. The nearest car park is located on Drury Lane, Parker Street, Parker Mews, London, WC2B 5NT.

The rear door in Exeter Street is to be used for deliveries only and is normally locked.

The main entrance is fully accessible to wheelchair users and should be used by all staff and visitors.

On arrival, report to the Davidson Building Reception who will direct you to the first floor.

Travel tips from major London stations

Charing Cross – 6 minutes walk

Waterloo – 12 minutes walk across Waterloo Bridge, or buses 139 or 176 to Stop ①

London Bridge – onward rail link to Charing Cross

Kings Cross or St Pancras – Piccadilly Line to Covent Garden tube, or bus 91 to Stop ②

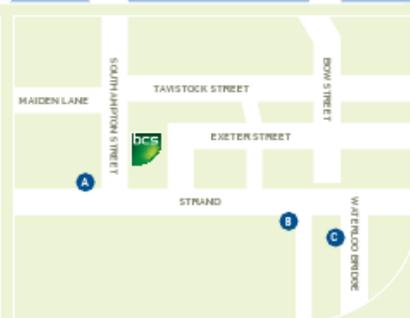
Euston – West End Branch of Northern Line to Charing Cross, or bus 91 to Stop ③

Victoria (rail and coach stations) – Circle Line to Embankment, but the most direct journey is via bus 11 to Stop ④

Paddington – Circle Line to Embankment or Temple, Bakerloo Line to Charing Cross or buses 15 or 23 to Stop ⑤

Liverpool St – Circle Line to Embankment or Temple, or buses 11 or 23 westbound

Fenchurch St – Walk to Tower Hill, then District or Circle to Embankment



Presentation Abstracts and Speaker Biographies

James Lyndsay

“A Nest of Tests”

Dull testing is easy, and easy jobs should be automated. James Lyndsay will show you how he uses simple datasets and tools to generate thousands of stupid tests, and how he aggregates those measurements to arrive at something more revealing.

We'll explore ways that you can adjust your design to investigate what you've found, and we'll talk about the ways you can use the approach, and its weaknesses. We've got (helpful) software to test, so if you happen to have a laptop with you, there is the option of doing some testing in parallel with James - at generating and executing bulk tests, visualising the output, and considering what to investigate next.

Play with it in advance at

<http://sigist20150605.workroomprds.com/>

James Lyndsay has been testing since 1986, and has worked independently since setting up Workroom Productions in 1994. Consulting since 2000ish, he's worked to technical requirements for companies that make and sell software, to commercial requirements for companies that buy and use software, and to unexpected requirements everywhere. He's been the test strategist on huge and critical projects, the sole tester in tiny startups, and all points inbetween. In the community, James is known for writing the Black Box puzzles, for keynotes and tutorials (and the TestLab) at many international conferences, as the facilitator of LEWT (the London Exploratory Workshop in Testing) and for his hands-on public and corporate testing workshops.

Paul Vincent

“Testing business logic... begins in Requirements!”

Development approaches such as Agile Scrum mean faster iterations for requirements, development and test. It has always been the case that earlier testing means faster issue detection and faster resolutions leading to reduced risk.

Today, the Business Analysts involved in user stories and requirements gathering are also involved in testing. They are using standardised model notations like BPMN in wider numbers – and many BPMN tools allow for testing aspects like path verification. But a major area of untested requirements remain in the business logic embedded within process tasks.

Helping to solve this problem is the new OMG standard DMN, for Decision Model and Notation. This is the equivalent of BPMN but for decision logic. DMN is encouraging a new generation of Business Analyst tooling that can provide business rule verification and validation. The result is that the Business Analyst can produce verifiable business logic for the developers, but also test cases. Some tooling also enables code generation, simplifying the development task as one of service configuration (and test setup!).

This session gives a quick introduction to DMN, has a quick look at one decision model methodology, TDM (The Decision Model), that can be described in DMN, and demonstrates verification and validation of some TDM decision models.

Paul Vincent BSc MSc MBCS is a long-time proponent of business rule and decision technologies. He is currently working at a large UK bank as Decision Architect.

Stuart Reid and Isabel Evans

“Using Influence Diagrams to Understand Testing” [Workshop]

Influence diagrams provide a simple-to-create and easy-to-understand approach to addressing the complexities of real-life problems. For instance, as testers we may want to find more bugs, but what is the knock-on effect of this on the developers and the business? Developers now have more defects to debug and it's likely that the business has lost confidence in meeting their delivery date (and that's just at the top level!). Influence diagrams provide a means of understanding and managing the complexities of those key interactions between testers, developers and the business.

Over the last few years, we have been using them in real organisations to help the introduction of testing and agile approaches as a practical tool to:

- Analyse the causes of problems
- Help identify potential solutions & improvements
- Predict and track the outcome of changes in organizations

In this practical workshop, Stuart and Isabel will, using hands-on activities, help you construct and interpret influence diagrams of increasing complexity. You will build diagrams to illustrate typical problems and solutions both in testing organizations and projects, and in the interactions between the business, testers and developers.

Using simply pens, paper and plenty of discussion we will analyse several typical testing problems and identify potential solutions. In this short workshop, we will start with simpler problems and diagrams, but also demonstrate how to analyse complex ('wicked') real-life situations (the original purpose of influence diagrams). By the end of the workshop you will have been introduced to and started to use influence diagrams. We will also provide you with the take-away of example influence diagrams that illustrate a number of different IT delivery and testing situations.

Stuart Reid is CTO with STA Consulting and has over 30 years' experience in the IT industry, working in development, testing and education. Application areas range from safety-critical to financial and media. Stuart supports the worldwide testing community in a number of roles. He is convener of the ISO Software Testing Working Group, which is developing the new ISO 29119 Software Testing standards and the new Reviews standard, chairs the BCS specialist Group in Software Testing, and founded the International Software Testing Qualifications Board (ISTQB) to promote software testing qualifications globally.

Isabel Evans has nearly thirty years of experience in IT, mainly in quality management and testing. Since the mid-1980s, her work has focused on encouraging IT teams and customers to work together, focusing on results, flexibility, risk and test-driven approaches. Isabel is a popular speaker at software quality conferences worldwide and has been a member of several working groups for industry improvement. She is a published author on software quality, a Chartered IT Professional and a Fellow of the BCS. For four years, she has been quality manager at Dolphin Computer Access.

James Christie

“Principles for testing?” [Presentation and Discussion]

There has been much debate in recent years about the balance between principles and rules when regulation is framed. Software development and testing are complex activities that do not lend themselves to fixed rules or prescriptive “best practice”.

If stakeholders are to be confident that testers will provide value then perhaps we need clear principles against which testing can be evaluated. Testing lacks principles intended to guide and shape behaviour. I will show how this has contributed to some of the confusion and disagreement arising from ISO 29119 and the Stop 29119 campaign. I will also argue that we can learn from the “rules based versus principles based” debate and I will initiate a discussion about what form principles might take for testing, and where we should look for sources for these principles.

James Christie is a self-employed testing consultant with 32 years IT experience. Before moving into testing he spent six years as an IT auditor, so he has experience on both sides of the fence. He has also worked in information security management, project management, business analysis and development. His experience helps him understand the relationships between different specialisms.

He is particularly interested in links between testing, auditing, governance and compliance. He spent 14 years working for a large UK insurance company, then nine years with a big IT services supplier working with large clients in the UK and Finland. He has been self-employed for the last eight years.

Jonathon Wright

“Testing as a Service: Models”

The Testing landscape is changing forever, the traditional approach to providing business value through testing is constantly being challenged. So businesses need to constantly re-examine the real value of testing services as an integral part of their overall delivery capabilities. The tendency has been to rely on complex hybrid resourcing models made up of internal resourcing and/or external resourcing (near shore, mid shore and off shore), to strive for Testing Centres of Excellence (TCOE).

The question is: are traditional Testing as a Function (TaaF) as part of the Software Development Lifecycle (SDLC) model still valid? Or whether Testing as an Activity (TaaA) needs to develop a Solution Delivery Lifecycle Integration (SDLCi) model to provide business value as part of a Global Testing Marketplace?

Jonathon Wright has over 15 years of international automation experience with a number of global organizations; including Deutsche Bank, Lehman Brothers, Hitachi Consulting, Thomson Reuters, Xerox, New Zealand Lotteries Commission, Unisys and Siemens. He's a serial blogger on Test Automation as a Service (TaaaS.net).

Jonathon also contributed to the best-selling book Experiences of Test Automation: Case Studies of Software Test Automation, Dorothy Graham & Mark Fewster, and a number of eBook's on Testing as a Service models (epistemic & systemic entropy), Advanced UFT 12 for Test Engineers Cookbook (Testing-store.com) and API testing in the cloud (service & network virtualisation).

He is the Director Testing, Quality & Assurance of Hitachi Consulting as well as presenting at various international testing conferences, such as Gartner (London), STARWest (California), Fusion (Sydney), ANZTB (Melbourne), EuroSTAR (Gothenburg and Dublin), BCS SIGIST (London).

Paul Gerrard

“A New Model for Testing”

This talk proposes a model of the thought processes that every tester uses. In a sentence, what we do is this: "we explore sources of knowledge to build test models that inform our testing". The model identifies two modes of thinking – exploration and testing – and we use judgement to decide when to flip from one to the other.

Exploration has four activities: enquiring, modelling, predicting and challenging. Testing has six activities: informing, applying, interpreting, logging, reporting and refining.

Separating out these ten activities clarifies what testers do. It identifies the capabilities and skills that all tester need to acquire, to practice and excel in.

This is the final keynote of the June 2015 SIGiST, and during his talk Paul will reflect

on how the presentations and discussions during the day map to the model of testing.

Paul Gerrard is a consultant, teacher, author, webmaster, programmer, tester, conference speaker, rowing coach and 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, he is a Principal of Gerrard Consulting Limited, the host of the UK Test Management Forum and was the Programme Chair for the 2014 EuroSTAR testing conference.

In 2010 he won the EuroSTAR Testing Excellence Award and in 2013 he won the inaugural TESTA Lifetime Achievement Award.

He's been programming since the mid-1970s and loves using the Python programming language.

Check out Paul's accompanying paper, added at the end of The Tester.

SIGiST White Paper Scheme

We have set up a new area on the BCS website of a searchable repository for white papers and articles on testing and we are looking for contributors. That means you!

Do you have an existing paper you would like to repurpose and make more widely available through the SIGiST website?

- Then please send us the paper with three keywords for searching.

Would you like to write a new paper?

- Please send us the title and abstract together with the three keywords (or phrases)
- We will review the proposal and guide you through the authoring process
- For those who are thinking of speaking at SIGiST then this might be a good way to prepare a talk and get some useful feedback

If you have been thinking of writing or publicising an existing paper then this is the ideal opportunity. Please email your existing paper (with keywords) or your proposal to The Tester Editor, phill.isles@bcs.org

Past articles from The Tester will slowly be added to the repository as well.

Follow this link to the repository: <http://www.bcs.org/category/18128>

Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on phill.isles@bcs.org

Event Listings 2015

If you would like your event listed here, please contact the Editor phill.isles@bcs.org

May

STAREAST

3 - 8 May 2015
Orlando, US

<http://stareast.techwell.com/>

Belgium Testing Days

18 - 21 May 2015
Brussels, Belgium

<http://btdconf.com/>

June

SIGiST

5 June 2015
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

September

SIGiST

15 September 2015

London, UK

<http://www.bcs.org/server.php?show=nav.9264>

November

EuroSTAR

2 – 5 November 2015
Maastricht, Netherlands

<http://www.eurostarconferences.com/>

Agile Testing Days

9 – 12 November 2015
Potsdam / Berlin, Germany

<http://www.agiletestingdays.com/>

December

SIGiST

2 December 2015
London, UK

<http://www.bcs.org/server.php?show=nav.9264>



The EuroSTAR Conference Super Early Bird discount deadline is fast approaching! Don't forget to take advantage of the additional discount allocated to **BCS SIGiST** members.

Don't know EuroSTAR?

Never been to the EuroSTAR Conference? Here's a taste of what you can expect: [Trailer](#)

Calling all BCS SIGiST Members! – book early for EuroSTAR to save over 25% on registration, or over 35% for groups of 5.

As a member of BCS SIGiST you can avail of a further 10% off your conference registration. This 10% discount code can be used at any time when registering but if you register on or before May 29th you could save over €600 – over 25% off!! Get over 35% off group registrations (of 5 or more) where every fifth attendee goes for free!

Register using the discount code: **SOBC10**

Are you looking to be inspired in 2015? EuroSTAR is the place for you!

Long established as Europe's #1 software testing conference, EuroSTAR features keynotes and tutorials from global thought-leaders, dozens of insightful track sessions and the opportunity to network with hundreds of software testing professionals. Be there in Maastricht to celebrate everything that's great about your profession over four intensive days of knowledge sharing in a vibrant atmosphere! This year the conference takes place at the MECC Maastricht from November 2nd – 5th.

The 2015 Programme offers practical advice, real-life experience stories and thought-leading insights into DevOps, Mobile, Management, Communication, Agile, Test Automation and more – presented by a mix of experienced speakers and newcomers to EuroSTAR. You can see the full programme [here](#).

[Register Now](#)

Or if you have any questions about registration / pricing please contact siobhan@eurostarconferences.com

Mobile Deep Dive 2015

EuroSTAR have added a brand new bonus event on mobile testing – EuroSTAR Mobile Deep Dive, which takes place on Friday 6th November.

Tickets to Mobile Deep Dive cost €850 (normal group rates apply) and if you register to attend the full EuroSTAR Conference (any Tues-Thurs inclusive ticket) you can also get a 50% discount on a ticket to the Mobile Deep Dive Event. [View Mobile Programme](#)

Be there in Maastricht!

Looking for a new role? Try BCSrecruit.



If you are Job Seeker, or a Job Recruiter check out BCSrecruit <http://www.bcsrecruit.com/>

Here's a current Software Test Engineer vacancy from the site.

<http://www.bcsrecruit.com/job/software-test-engineer-953290023?src=search&tmpl=sem&sctr=IT>



We're looking for an ambitious and talented test engineer to join our team of developers managing significant annual growth and tackling tough online booking, search and geo challenges. We're all about collaboration - you'll take part in regular brainstorms as the company grows rapidly, and gain direct experience of the rapid evolution of a high-profile growth company, working alongside experts in an apolitical, friendly environment.

Cool stuff we've done recently include launching our new mobile site and app and being cited at the recent W3C/ODI/OKF Open Data on the Web event for our use of public transport and attraction data. Behind the scenes we've built some pretty nice features to help campsite owners manage their vacancies and encourage them to sign up.

We're based in the award-winning Barley Mow Centre off the Chiswick High Road (nearest tubes Turnham Green, Gunnersbury) and are a lively friendly team with bags of experience. We want someone similar to join us - is that you?

Responsibilities:

- assuming responsibility for the software development testing cycle, defining and implementing the testing process, and ensuring testing is undertaken
- ensuring high quality releases throughout our web and mobile platforms
- defining tests in conjunction with our development team
- creating test scripts, and designing and writing test report templates
- tracking bugs in our ticketing system
- automating where appropriate

What you might be working on:

- adding geo features such as new points of interest and time-based search group bookings functionality to determine the optimal combination of pitches offered
- growing our euro-denominated business to capitalise on the European market
- enabling local-market versions of our site
- personalising our site
- optimising our marketing channels
- maximising speed and stability
- launching our mobile apps

You are:

- a graduate in a technology discipline, or someone with equivalent experience in the software testing field
- experienced in testing both web and mobile applications
- experienced with Selenium or equivalent tools
- BCS / ISTQB Foundation or Advanced Level qualification (or equivalent)
- meticulous and detail-obsessed, with outstanding communication skills
- keen to get involved in a hands-on way, suggesting new ways forward for the site and enjoying coming up with new feature ideas
- a fan of agile working practices - you're proactive and get things done
- a quick-on-the-uptake problem solver who takes ownership of problems and projects

Some remote working is possible, and some evening and weekend may be required.

About Pitchup.com

Pitchup.com is the market leading online booking site for campsites, glampsites and caravan parks, with over 1,200 sites available to book in 14 European countries.

Brits spend 25% more nights camping and caravanning than staying in hotels, and the market is booming as Brits switch to holidays at home. Now, we're expanding into Europe and further afield.

Judged Best UK Travel Site at the British Travel Press Awards, Pitchup.com receives up to 70,000 visits per day to its website and acclaimed mobile platforms ([URL removed]) The company is profitable and was founded in 2009 by former lastminute.com staff with a background in the holiday park sector.

Don't take our word for it...

- * Winner, 'Travel / leisure / sports mobile strategy / campaign' and runner-up, 'Mobile optimised / responsive website', Marketing on Mobile Awards 2014
- * Runner-up, West London Business of the Year, West London Business Awards 2013
- * Best UK Travel Website, British Travel Press Awards, 2011
- * Winner, Future 50 - Red Bull / Real Business, 2011
- * 'Best travel websites ever' - Guardian, 2011
- * Winner, Best Travel Directory - TravelMole awards, 2011
- * Winner, Best Domestic Strategic PR campaign, Travel Marketing awards, 2011
- * Winner, Best UK Travel Information Site, Travolution awards, 2010
- * Featured throughout the national press including the BBC

We're a lively, friendly team with bags of experience, based at the award-winning Barley Mow Centre with around 70 other small companies, off Chiswick High Road (nearest tubes Turnham Green, Gunnersbury). Now we're building our team for further expansion both within the UK and abroad. We want someone similar to join us - is that you? Come and join us and help us make Pitchup.com an international success! Check out <https://www.pitchup.com/jobs/>





The image shows the BCS Personal Development Plan (PDP) interface on three devices: a Desktop/Laptop, a Tablet, and a Smartphone. The Desktop/Laptop view shows a dashboard with a navigation menu, a main content area with a list of activities, and a sidebar. The Tablet view shows a similar dashboard with a pie chart for 'Activities Report: CPD Activity Type Analysis' and a list of activities. The Smartphone view shows a mobile-optimized version of the dashboard. A yellow banner in the top right corner says 'NEW'. The BCS logo is in the top left corner.

bcS The Chartered Institute for IT

Personal Development Plan

Your portable record of professional aspirations and activities

Desktop/Laptop

Tablet

Smartphone

Did you get your Personal Development Plan email with suggested potential CPD activities?

The BCS Personal Development Plan (PDP) uptake is going well, with over 1,000 users already actively recording their CPD Development Goals, Activities and preferences. It's not just about recording details though, as there is a Resources section that shows live feeds of potential CPD activities, and a tailored email is sent every 2 months with details of the latest videos, articles, blogs, books and research in your specified field of interest. If you haven't registered yet, you can see the content from the latest PDP bulletin for topics relating to solution development and implementation here <http://www.bcs.org/content/ConWebDoc/50854> or by going to the CPD Portal at: <http://www.bcs.org/pdp/> and selecting the "Give me ideas" link.

The BCS Personal Development Plan is free to use; BCS members can use their Member Secure Area login and password to access it at <https://pdp.bcs.org/>, and non-members can use most of the facilities (using the same link) and registering to create their own user name and password. You can use it on a PC / laptop or compatible tablet PC or smartphone.

A New Model for Testing

Discussion Paper

Paul Gerrard
Gerrard Consulting
e: paul@gerrardconsulting.com
w: gerrardconsulting.com
t: @paul_gerrard

Version: Initial Draft 0.1 for discussion
Published: 29 June 2014

Table of Contents

Table of Contents	i
1 Why a New Model?	1
2 Fundamentals	1
2.1 Base definition of test	1
2.2 Thought Processes, Not Test Logistics	2
2.3 What Stakeholders Want	2
2.4 Fundamental Test Process	2
2.5 Our Brains are Designed to Build Models of Our World	3
2.6 Developers and Testers Create Models	3
2.7 Test Design is Based on Models	4
2.8 Sources of Knowledge	6
3 Exploration and Testing – Two Modes of Thinking	7
4 The Exploration Process	8
4.1 The Goal of Exploration	8
4.2 The Four Activities of Exploration	8
4.3 Enquiring	9
4.4 Modelling	9
4.5 Predicting	11
4.6 Challenging	11
4.7 Exploring the System Under Test	12
5 The Testing Process	12
5.1 The Six Activities of Testing	12
5.2 Informing	13
5.3 Applying	13
5.4 Interpreting	14
5.5 Logging	14
5.6 Reporting	14
5.7 Refining	15
5.8 New Model Testing	15
6 Discussion	17
6.1 Where do the Test Axioms fit?	17
6.2 Why not separate design and delivery in the New Model?	18
6.3 Where does human judgement fit in the model?	18
6.4 How does the New Model relate to TDD, BDD and similar approaches?	18
6.5 Does the New Model describe Non-Functional as well as Functional Testing?	20
6.6 How does the New Model relate to the “checking v testing” dichotomy?	20
6.7 How does the New Model relate to ‘exploratory testing’?	21
6.8 Does the New Model change thinking about test automation?	22
6.9 What about developer testing?	22
6.10 Testing Skills – a different perspective?	23
6.11 Does the New Model affect certification?	25
7 Conclusion	25
8 References	26

1 Why a New Model?

Don't we know everything there is to know about testing? The response from pretty well everyone who knows anything about testing is 'certainly not'. I am proposing a New Model because I believe that the testing world is getting shaken up quite dramatically. The current confused state of affairs [1,2] could mean that some testers will lose their jobs and be reassigned to do other things, and some of the value that testers add (but few can articulate, by the way) will be lost. The software industry will be the poorer for it.

The current perspectives, styles or schools of testing will not accommodate emerging approaches to software development such as continuous delivery and, for example new technologies such as Big Data, the Internet of Things and pervasive computing. These approaches require new test strategies, approaches and thinking. Our existing models of testing (staged, scripted, exploratory, agile, interventionist) are mostly implementations of testing in specific contexts.

I believe there is an underlying model of testing that is context-neutral and I have tried to shed some light on what this might be by postulating the Test Axioms, for example [3]. The Axioms are an attempt to identify a set of rules or principles that govern all testing. Some people, who have used them think they work well. They don't change the world, they just represent a set of things to think about – that's all. But, if you choose them to be true, then it becomes possible to avoid the quagmire of debates about scripted versus unscripted testing or the merits and demerits of (current) certifications, the value of testing and so on.

The model of testing presented in this paper is an extension to this thinking. The model represents the thought-processes that I believe are going on in my own head when I explore and test. You might recognise them and by doing so, gain a better insight into how you test. I hope so. As George Box said, 'essentially, all models are wrong, but some are useful'. This model might be wrong, but you might find it useful. If you do find it useful, let me know. If you think it's wrong, please let me know how I might improve it.

This paper presents an alternative view of the core activities of testing and a New Model of it. The aim of the paper is to make this model available and through discussion and challenge – to improve it or kill it. It is a straw man. It is a model. It is wrong. It might be useful.

2 Fundamentals

I will use my selected definition of testing and suggest a model based on a belief that ALL testing is exploratory [4].

2.1 Base definition of test

There are some core definitions that need to be stated at the outset. I will continue to use my selected definitions, sourced from a dictionary [5] and used in [3, 6]:

Test: (noun) a procedure for critical evaluation; a means of determining the presence, quality, or truth of something; a trial

Test: (verb) to critically evaluate; to determine the presence, quality, or truth of something; to conduct a trial

Testing: (noun) doing some or all of the above.

2.2 Thought Processes, Not Test Logistics

When tests are performed on-the-fly, based on mental models, the thought processes are not visible to others; the thinking might take seconds or minutes. At the other extreme, complex systems might have thousands of things to test in precise sequence, in complicated, expensive, distributed technical environments with the collaboration of many testers, technicians and tool-support, taking weeks or months to plan and apply.

Depending on the approach used, very little might be written down or large volumes of documentation might be created. I'll call the environmental challenges and documentary aspect 'test logistics'. The environmental situation and documentation approach is a logistical, not a testing challenge. The scale and complexity of test logistics can vary dramatically. But the essential thought processes of testing are the same in all environments.

So, for the purpose of the model, I am going to ignore test logistics. Imagine, that the tester has a perfect memory and can perform all of the design and preparation in their head. Assume that all of the necessary environmental and data preparations for testing have been done, magically. Now, we can focus on the core thought processes and activities of testing.

The model assumes an idealised situation (like all models do), but it enables us to think more clearly about what testers need to do.

2.3 What Stakeholders Want

Testing is an information activity performed on behalf of (people who I will call) testing stakeholders. The manager who asked you to test could be your most important stakeholder but project stakeholders, the 'business', users, developers and others are all testing stakeholders because at some time or another, they will be very interested in the outcome of testing.

If you are testing the products of your own efforts (for example, you are a developer), *you could be your own stakeholder*. Your approach to testing your own products or systems will be focused on what *you and others*, as stakeholders, want to learn about those products or systems. Usually, the outcomes and interpretations of testing help stakeholders to make a decision – to accept, to reject, to delay, to stop, to investigate further, to re-think, to fix a defect, and so on.

The testers' mission is usually determined before the tester starts these tasks. Testers need to understand who their stakeholders are, what goals and risks concern them, what decisions they need to make, why, when and how. But they also need to know the scope of the investigation, the effort and elapsed time available to perform the testing and the nature of the information that testing must provide.

The information that stakeholders need from test reporting strongly influences what models or modelling approach will be most meaningful to them.

2.4 Fundamental Test Process

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*.

2.5 Our Brains are Designed to Build Models of Our World

Our brains are capable of modelling and remodelling our surroundings. These models allow our brain to send impulses to our muscles in highly complex patterns that enable us to move and reach our destinations without bumping into things. Our brain recalculates and recalibrates these models second by second. Modelling and visualisation are essential skills that play an important role 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.

2.6 Developers and Testers Create Models

I have hinted that perhaps, the modelling activity that testers do is the same as the modelling that developers do. Now, this is a quite speculative suggestion. Because I am a developer and because I test, I find that my mind constantly flip-flops between a melange of mental questions that refuse to go away: “What is required”, “How do I know?”, “How will it be used?”, “How shall I build it?”, “How will I know I’m done?”, “How shall I test it?”, “Is it finished, safe, ready?” These questions dominate my thinking whether I am writing code or testing it.

The questions that pester the developer and tester trigger the need to model the requirements, the solution and the tests. So let me suggest that perhaps, developer and tester exploration and modelling really are quite similar. Am I really suggesting that developers and testers explore in the same way? Well, yes, sort of – but it can’t be as simple as that, can it?

Developers explore their sources of knowledge to understand what to build and potentially, how to ‘grow’ their code into useful features. Whether the developer is using a test-driven approach or they adopt the ‘code a bit, test a bit’ approach, each test confirms that the incremental code the developer has just written matches their mental model.

Sometimes, just like a tester, the developer will try things out, not knowing what the outcome might be and then decide what to do next.

The developer might take the outcome of a test and change or refine their model. Or they might ask for more information. Or they might fix some anomalous behaviour. At this time, the developer would not call these anomalies bugs because they occur as part of the learning process. As they model, they compare the software with their models. Through exploration and experimentation, the model and the software evolve towards an alignment that could provide value to a stakeholder.

A tester has similar choices. The anomalous behaviour might expose a flaw in the requirement, the model or the system, and judgement is required. The tester might decide to explore further, refine the model, or log a problem report.

2.7 Test Design is Based on Models

Most of the text of this section has been extracted from the Tester's Pocketbook, pages 25-30.

Boris Beizer said in 1990 [7]:

'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 to select tests in a systematic way. Test models are fundamental to testing and the remainder of this section discuss them.

2.7.1 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.

We usually base models on one of the following sources:

- The test basis – text or diagrams or information that describe required behaviour.
- The architecture of the system – we identify testable items in its user-interface, structure or internal design.
- Modes of failure – patterns of failure of concern from our experience or historical record.
- Usage patterns – the way the system will be used, operated and interacted with in a business context.

2.7.2 An example of a test model

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

We can extract 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. If the system under test (the car) does not behave correctly according to our model we either change the car, or we change the model (our interpretation of the car’s behaviour).

Everything looks fine – doesn’t it?

2.7.3 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?¹

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’ [8]). 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

¹ Even applying an efficiency rating would be a gross over-simplification. Typically, 80% of the power generated by burning gasoline is wasted heating the car and atmosphere, overcoming friction of car components, tyre wear and wind resistance.

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.

2.7.4 Formal test models

Formal models tend to be derived from analyses of design or requirements text, tables or diagrams or are derived from the architecture or structure of the system itself. These models are often specified (and sometimes mandated) in standards or development methodologies. They are intended to be systematic methods that, when properly used by testers, deliver equivalent sets of tests from the same test basis and context. In principle, a quantitative coverage measure can be obtained from a formal test model².

2.7.5 Informal test models

Test models don't necessarily have to be diagrammatic or represent the requirements or design of the system. For example, some models are just lists of modes of failure, risks or vulnerabilities.

Security hackers and criminals adopt patterns of attack to undermine or breach the security of software systems. To verify that systems are not vulnerable, security testers use lists of vulnerabilities as a model to trigger ideas for tests.

Other informal test models include navigation paths through business processes or the system itself, quality criteria, user roles and behaviours or scenarios that stakeholders believe are relevant to the use of the system in the real world.

Informal models cannot be used to define quantitative coverage measures.

2.7.6 Ad-Hoc test models

Some models can be ad-hoc, invented by the tester just before or even during testing. If, while testing, a tester sees an opportunity to explore a particular aspect of a system, he might use his experience to think up some interesting situations on-the-fly. Nothing may be written down at the time, but the tester is using a mental model to generate tests and speculate how the system should behave.

Stakeholders may not tell testers to use specific test models, but where documentation written by, on behalf of or approved by stakeholders is used to derive tests (the Test Basis), the stakeholders ought to be aware of what models are adopted and how tests are being derived. The stakeholders may be of the opinion that the model generates too few (or too many) tests to be meaningful or economic.

2.8 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 accessible.

² See the section on Coverage in [2] (p 35)

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.

3 Exploration and Testing – Two Modes of Thinking

There are two modes of thinking in our test approach – exploration and testing – that have distinctly different goals. By separating the two, we allow our minds to focus on the different goals at hand. Our thinking is clearer because our judgement on whether a source is reliable is not clouded by whether (or not) we have found a good test of the system (and vice versa). This is not an argument for staged testing. Rather, I make the case for clear thinking, depending on what your goal is at the time – creating good models from trusted sources or creating and applying effective tests.

We start by exploring our sources – we formulate models; we use models to challenge our sources through example to improve our sources and our models. When we are satisfied that a model is adequate, we use the model to inform our testing. I use the term ‘inform’ deliberately. The model may be formulated in such a way that test cases are readily obtained. Some models, for example state diagrams, boundary values or decision tables expose test cases readily. Other models such as check lists of risks or design heuristics require further thinking. For example, ‘which tests will best demonstrate whether a mode of failure is possible or likely?’

Some (perhaps most) mental models cannot easily be described. They could be based on our experience, imagination, prejudices or biases. They might exist only in our subconscious mind. There may be several or several thousand different visualisations, patterns or forms that our models might take. The workings of our brains is still a mystery. There might be a more satisfactory description of how brains work in the future but right now, for the purpose of this paper, we need only believe that models are formulated in the brain of the tester.

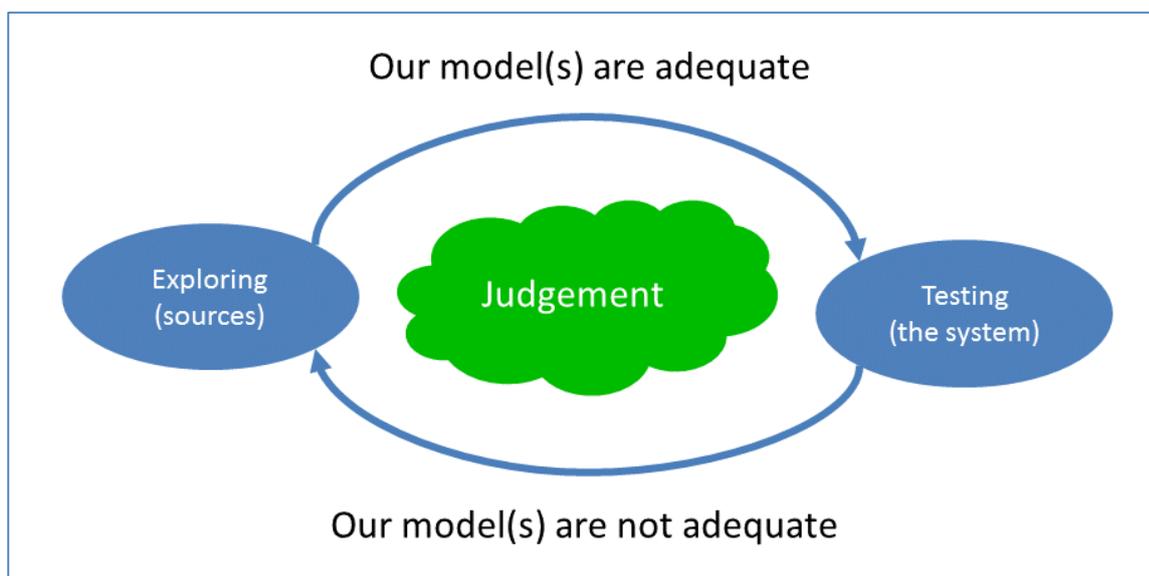


Figure 1 Exploration, Testing and Judgement

Figure 1 Exploration, Testing and Judgement illustrates the two modes of thought – exploration and testing, governed by the mission of the tester. Judgement is required when moving from one mode to the other. The two modes of thought represent two different processes followed by testers. I’ll

describe each in more detail. The model is introduced in two halves which are then combined into a single schematic later.

4 The Exploration Process

4.1 The Goal of Exploration

We use the exploration process to derive a model or more usually, set of models, from the source(s) of knowledge. In deriving these models, the tester will detect gaps and inconsistencies in the sources. There are two main activities:

- We elicit information from our sources of knowledge to derive and improve useful models (for testing)
- We make predictions from our models, challenge our sources of knowledge through example and (perhaps) improve them.

These mutually supporting activities drive the tester's behaviour when in exploration mode. The goal of improving our sources is optional in some circumstances so we can offer two formulations of the outcomes (or outputs) of exploration:

1. Trusted models of the system from sources of knowledge or
2. Trusted models of the system, consistent with trusted sources of knowledge.

The first is easy to understand – we might leave behind our fallible sources. The second hints at a more involved process – including the improvement of the sources. These might be revised documentation or clarifications of thinking. The process might be very formal or informal. For the purpose of this paper, we'll rest at knowing that the exploration process provides the opportunity to improve the sources. But what do we mean by 'trusted'?

In the Business Story Pocketbook [9], we describe a trusted requirement thus:

"A trusted requirement is one that, at this moment in time, is believed to accurately represent the users' need and is sufficiently detailed to be developed and tested."

And we argue that:

"... a trusted requirement has both a requirement definition and a set of illustrative examples to fully describe a feature. It is easy to demonstrate that this is the case. Give a developer a set of examples only, and they can create code that passes all tests based on those examples. But their code might simply implement a series of choices of inputs and pre-defined outputs without providing a generalised solution. It's comparable to teaching a child their 'times table' without them understanding the general rules of multiplication. Asking 'what is 7 times 9?' may result in a correct answer. Asking what is 6.345 times 9.321 won't.

Trusted requirements usually require both a generalised statement and examples."

4.2 The Four Activities of Exploration

There are four activities of exploration that need explanation:

- Enquiring
- Modelling
- Predicting
- Challenging

These activities take place in sequence for some aspects of exploration, but exploration is an extremely dynamic and unpredictable process that mostly happens in the mind of the tester. Most people who are exploring are juggling multiple sources of and perspectives on information that may or may not be germane to the task. It is likely that different parts of the brain are doing these things in parallel. Our model is a crude representation of the reality. The figure below shows the process.

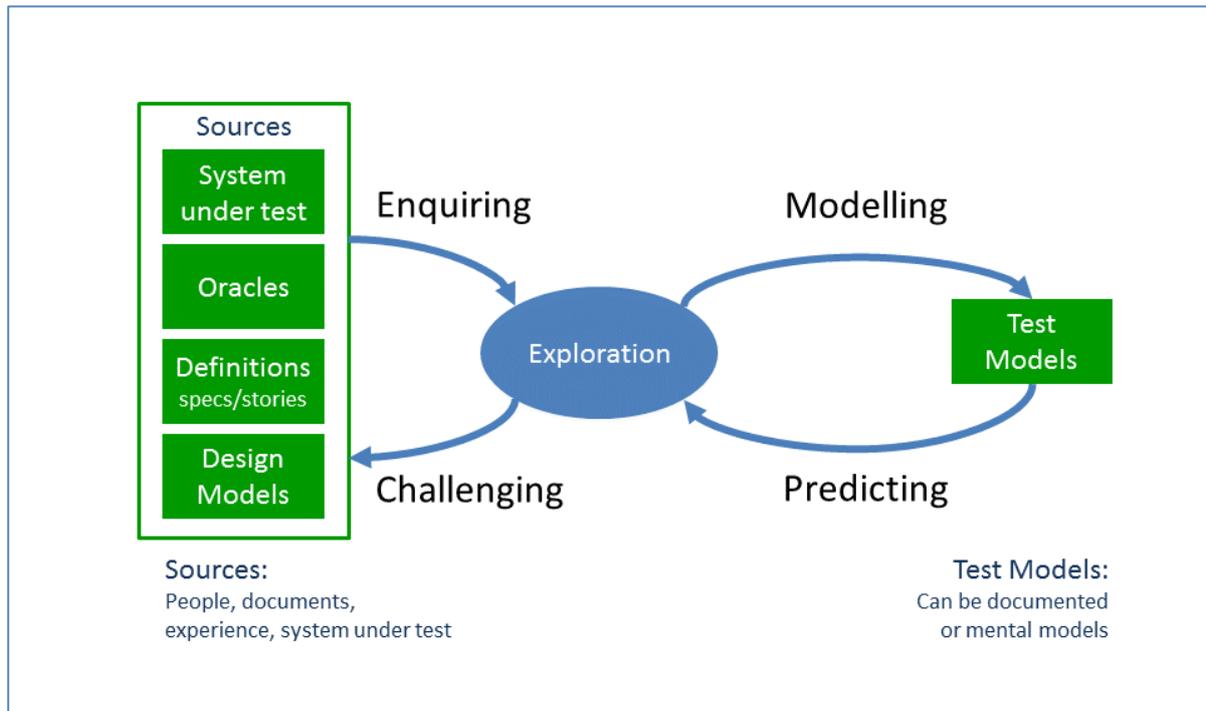


Figure 2 the Exploration Process

4.3 Enquiring

We need to elicit information from our sources of knowledge and we do this by enquiry. Because our sources vary considerably, our enquiry technique will vary. With a document, we read, assimilate, analyse and question. Typically, we will ask the author for clarifications as we learn more. We might ask open questions to get the 'big picture' initially, but as we learn more, we focus more and more on detail.

As our models evolve, we see gaps in our model and seek clarification, detail and confirmation. We distinguish between what the system must cope with and how it is expected to behave. All the time we are also looking to identify trustworthy oracles.

4.4 Modelling

As we acquire information from our sources and assimilate it, our mind attempts to structure disparate items of data into some kind of order or perspective. Modelling the data is how our brains make sense of it. I have said earlier that the process of modelling varies from drafting graphical representations (directed graphs, mind-maps and so on), to making check lists or tables or formulating mental models that may take nebulous forms that are never documented or shared with others.

Models take various forms. Here are some popular types of model:

Checklists and Inventories

A model can comprise a list of functional areas, features or issues to cover in testing. Pre-existing checklists of standard problems encountered in the past might be used and lists of quality criteria, non-functional requirements are common. A list of features or a table of contents lifted from a document might be used. A common approach is to compile a risk register that captures the modes of failure of most concern that can be used to direct the test effort and prioritise tests. Of course, in a similar vein, a list of goals or key processes that must work might also be on your list.

Some models are specifically tabulated (rather than simple lists). Obvious examples would be decision-tables and state tables (which are really a transformation of state transition diagrams).

Graphical models

In many situations, a picture is worth a thousand words, as they say. Graphical models are most often presented as *directed graphs*. A directed graph is a diagram that consists of nodes and links that connect them. Nodes are usually rectangles, ovals or circles. Links are usually lines with arrows indicating a permitted direction or transition between nodes.

Examples of directed graphs are state-transition models, control-flow graphs, flowcharts, sequence, swim-lane, collaboration and class-diagrams. These kinds of models are often 'given' as part of requirements or design documents. But the same 'blobs and lines' convention can be used by testers to capture any kind of structure that a system or business process implements.

In Agile teams, it is common for business processes to be 'story-mapped' using cards on a board or, increasingly, software tools designed for the purpose. These provide obvious opportunities for testers – to cover features and paths through the business process.

Data Domains

Many of the variables that represent data in use in systems follow rules of validity or processing. Domain-testing depends on identifying these variables, understanding their ranges or rules of validity and characterising them. These models can be analysed to identify test values that are of interest. Domain Testing is comprehensively described in [10].

Business Stories

The concept of user stories has wide acceptance in Agile teams. Story summaries or feature titles are usually augmented by acceptance criteria. These acceptance criteria are basic checks on what the feature must do and can be used as test ideas.

In Behaviour-Driven Development (BDD), stories are captured in a Domain Specific Language as structured text. These structured stories can be interpreted by software tools to generate test code, but more importantly, acceptance criteria or scenarios are used to clarify requirements through collaboration and discussion. Naturally, the scenarios can drive feature-based test automation and be used as the basis of system or acceptance tests [9].

Doodles and Mental Models

The first thing to say about mental models, is of course they could be 'any of the above' in principle. Personally, I have a lousy short term memory, so anything large enough to be called a state-diagram or set of story scenarios would have to be written down somewhere for me to use them. However, all models start their life as a mental model and the choice that the tester has is whether the doodle that you might start with needs to be refined into something more elaborate or documented.

Some people find it useful to doodle their models. An obvious example here is a mind map. Mind-maps are hierarchies that can be used to decompose systems to sub-systems to menu options to features and so on. They can help people to visualise the structure of the system under test. But

mind-maps are also useful to take a list, such as a list of failure modes and break these down to lower levels of detail to identify scenarios that might trigger the failures and capture ideas for tests to 'cover' the risks.

When exploring the system under test and especially when the scope of exploration is well understood, the paths one takes through the system can usually be memorised and used as a mental-map of the product. As a tester navigates through features, mental models pop-up in one's mind, and if the model triggers interesting tests then usually the tester will spend time applying tests that 'cover' the mental model.

4.5 Predicting

A significant aspect of our exploration is the identification and use of oracles. In [2], I define a test oracle this way:

"A Test Oracle is the source, or sources, of knowledge that enable us to predict the outcome of any test. In effect, an oracle tells us what a system does in all circumstances and situations. If an oracle does this, it is perfect. Our oracle might be derived from the same sources as our models. Whatever our sources of knowledge are, like models, they are fallible."

Sometimes, our models will explicitly state outcomes. For example, a formally described state model will show initial state, input(s), final state and output(s). On other occasions, outcomes will be implicit ("failure is not an option") or be calculable from a stated formula, algorithm or table or we can refer to our trusted sources as oracles.

At any rate, when we are modelling, we can derive examples of a system in operation, suggested by our model and use these examples to post 'what if?' challenges to our sources.

4.6 Challenging

We pose challenges to our sources and we do this by example. Typically, we use our knowledge of the requirement and our model to posit interesting questions on the behaviour of our system. These questions take the form of scenarios that we believe the system must deal with. These scenarios or examples can be used to confirm our understanding, to point out gaps or inconsistencies in our sources. The DeFOSPAM technique, described in the Business Story Pocketbook [9] sets out one method for doing this. Other, alternative techniques are offered in [10, 11, 12, 13].

This 'challenge by example' can be seen to work nicely with sources that are people or documents. The challenge is much like a question that leads to a specific answer or outcome. Perhaps, like a lawyer or Socrates, a good tester should know pretty much what the answer is before they ask the question. Some typical answers from stakeholders would include:

"I never thought of that – let me ask someone and get back to you"

"That's unlikely. But I need to write a new requirement for that situation"

"That's inconsistent, the requirement needs a re-think"

"Stop. We need to re-write that requirement before we go any further"

"No, the behaviour is correct, but the requirement is unclear in this area"

The next action varies depending on these outcomes of course. But what if our source of knowledge is the system itself? This 'special source' needs special treatment.

4.7 Exploring the System Under Test

For testers, the system under test can itself be a source of knowledge and might be our main source if, for example, the documentation is poor or non-existent. We will explore the software and build our models. As we see examples of the system in use, we build a picture – a model – of its behaviour. Initially, we will guess, we will speculate, and posit heuristics to build a richer picture and try them out. Often, our heuristics will fail. So we try others.

As we explore, our models are informed by our knowledge and experience, our heuristics and the behaviours exhibited by the system. Over time, often in just a few minutes, our models coalesce and take form. In our minds, our model matures into something we have some confidence in. As we explore further, we begin to trust that our model can predict the behaviour of our system. When we trust our models, we believe that we can derive meaningful tests of the system.

At this point, we move into the testing process.

5 The Testing Process

The exploration process generates sets of models, from the source(s) of knowledge. When we test, we use these models to inform our selection of tests to learn how the system under test behaves and use that knowledge to inform stakeholders.

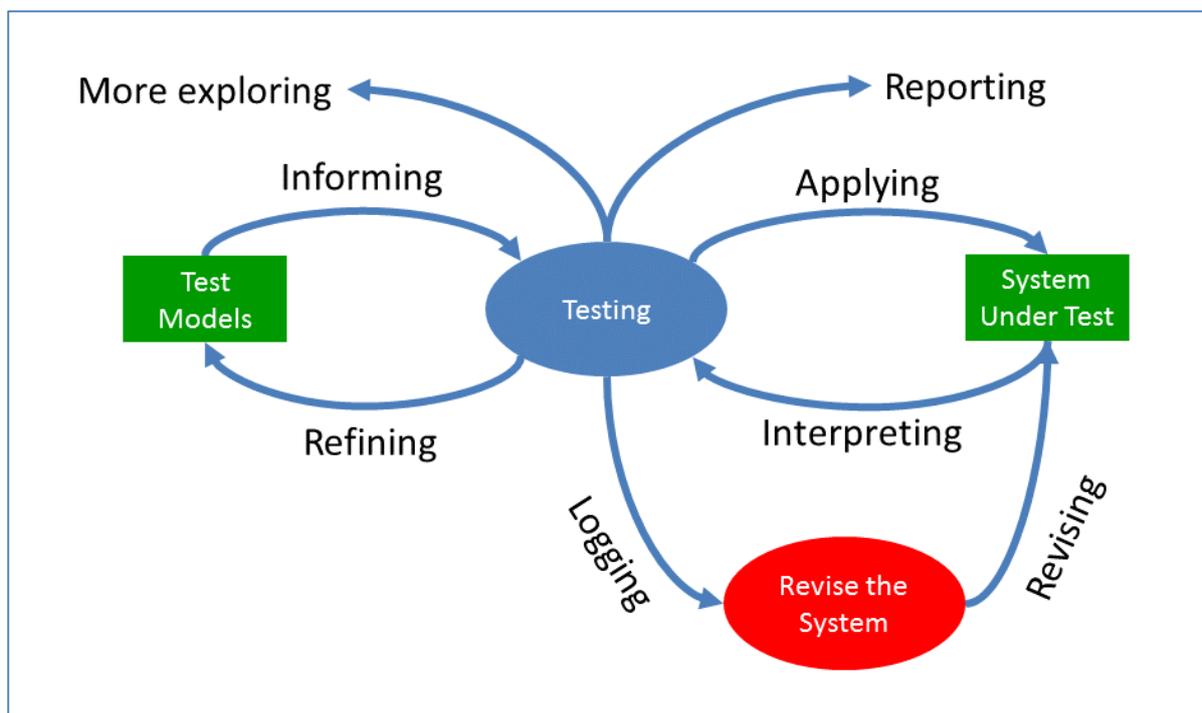


Figure 3 the Testing Process

5.1 The Six Activities of Testing

Figure 3 the Testing Process, is a schematic showing the flows of the thought process. There are six aspects of testing that need explanation:

- Informing
- Applying
- Interpreting

- Logging
- Reporting
- Refining

They take place in sequence for some aspects of the testing, but testing, by its nature, has uncertain outcomes so in most projects, like exploration, testing is a dynamic and unpredictable process.

5.2 Informing

Section 2.7 sets out how models are used – we use models to identify things to test. Some people call these things to test ‘test ideas’. One can also call them (test) coverage items and, if the model makes the calculation meaningful, calculate coverage measures.

In section 4.4, I set out the main types of models. The particular process of selecting tests using a model varies with the model of course, and in the case of test design techniques, several excellent books have been written describing the process. These are the standard patterns:

Checklists and Inventories

Tests are derived from a list of areas to cover. These might be risks, or modes of failure or design heuristics to consider and so on. In this case, the choice and number of tests is based on the judgement of the tester.

Graphical models

In the case of graphical models, the most common type is a directed graph. Think of them as diagrams comprising blobs and arrows. Flowcharts, sequence diagrams, swim-lanes and so on. Tests are often selected to cover all blobs or all arrows or selected sequences of them. Overall, *tests are implemented as paths through the directed graph.*

Data Domains

The Domain testing Workbook [10] has expanded the rather basic equivalence partitioning and boundary-value test design technique to a very general test design approach. I refer you to this reference.

Business Stories

Stories and scenarios written in a structured format are increasingly popular in Agile environments through the take up of BDD and associated tools. But scenarios are described more generally in [9, 13, 14]. The Gherkin format [15] makes stories and scenarios a convenient alternative to traditional test cases.

Doodles and Mental Models

As suggested earlier, these styles of model can be of any form so typically, one of the approaches above is appropriate.

5.3 Applying

I used the term ‘applying’ to indicate the actual execution of a test. Humans might apply tests through the user interface (UI) or some kind of proxy arrangement. Tests can also be applied using automated tools driving a technical or programmer’s interface (the API), or the UI. Automated tests require some form of script or set of instructions. Humans may, or may not, require scripts. Scripted tests usually focus on particular expected outputs or can instruct the tester to make broader observations.

Human testers apply the tests, observe outputs and judge whether an output matches an expectation (scripted or not) or judges whether the outcome of the test is anomalous in some way.

Tools can apply prepared tests and capture outputs and make comparisons with prepared expected results but tools are unsophisticated ‘observers’. Tools can only be used for basic checking activities in this respect. Obviously, some tests such as high volume tests through component APIs cannot be applied except by using tools. Tests of components that don’t have a user interface must involve some technology and might be applied by tools or people supported by tools.

Whether a human or automated tool applies the test and performs a comparison, *only a human can interpret the outcomes of tests.*

5.4 Interpreting

When a test is applied, the output might be as expected or not or the outcome might be anomalous in some way. There are a huge number of possible outcomes that could never be predicted (and documented) in advance, so testers often observe anomalous behaviour or inconsistencies that are not envisaged by the tester and certainly, not specified on scripts. Sometimes the tester sees things that just seem not to look ‘quite right’. It’s an unpredictable and potentially complicated situation. What interpretations on these observations are possible?

- The system behaves as expected (or not) in one situation (but may or may not behave correctly in other, similar, important, common situations) – perhaps more tests are required
- The system may behave (or misbehave) in ways that were not modelled, identified or predicted by a test model – perhaps the model needs refinement or new models are required
- The system may behave (or misbehave) in ways that were not recognised or even seen by the tester (whether scripted or exploratory) or tool (scripted) – testers and tools are fallible
- The system may fail in ways that are not acceptable to stakeholders – these need diagnosis and correction
- The system cannot be made to fail in ways that stakeholders are concerned with – perhaps stakeholder concerns are addressed or perhaps our tests are poor
- The patterns of behaviour observed may support a conclusion that the system is acceptable (or not)
- The patterns of behaviour (or misbehaviour) give rise to new concerns, new risks and more tests.

This is a (non-definitive) list of possible interpretations. Tests might have been applied by humans or by tools but these interpretations are made by testers and/or stakeholders and require domain and system knowledge, awareness of risk and the application of human judgement.

5.5 Logging

When the interpretation of an outcome is that the system is exhibiting anomalous behaviour, then the conclusion might be that further investigation is required. The logging of anomalies (also known as failures, bugs, defects, errors, problems, issues etc.) may follow a formal process, involving reviews and/or triage processes, change control and re-testing. But logging might also be informal whereby the tester talks to the developer and agrees a position. The tester might refine their model and carry on testing or the developer might fix the defect and notify the tester when the fix is implemented and released.

5.6 Reporting

Reporting is the process whereby the tester provides meaningful feedback to stakeholders. Partly, this will relate to the completion status or coverage of tests to give an indication of progress, but it will also give an indication of completeness or thoroughness. The status of individual tests is of

interest, but it is more the patterns that emerge from the interpretations of these tests that inform the decision-making of stakeholders.

Underpinning all good test reporting to stakeholders is to use models that are meaningful to stakeholders in your testing. For example, if your models describe business processes supported by the system then reporting can be framed by those models. If your models relate tests to business processes or goals, stakeholders can put the outcomes of your tests into context and relate them to their own experience. If you are the developer or your stakeholder is, then your tests might reasonably relate to the coverage of code, decisions, control flows and so on.

5.7 Refining

As tests are applied and outcomes interpreted, it is almost inevitable that your models may turn out to be incomplete, inconsistent or incorrect. A sensible test strategy must allow the testers to reflect, re-explore, re-think or make adjustments to align their models with the sources of knowledge and the knowledge captured in the design of the system itself.

Sometimes developers have, through their discussion and collaboration, acquired deeper insights into the stakeholders' needs than the tester. The developers' knowledge becomes embedded in the system itself and may never be apparent until the tester tries to test it. Only then might it appear that the tester's knowledge is lacking. Perhaps the tester needs to consult the developers, stakeholders, or users to acquire the same insight.

When the tester does understand things better, it might emerge that the developers made some poor design choices and it is their design that needs to change to align with requirements. You never know how these things might turn out.

5.8 New Model Testing

On the following page is the full model with both the exploration and testing process combined. The 'refine the system' element has been removed for clarity.

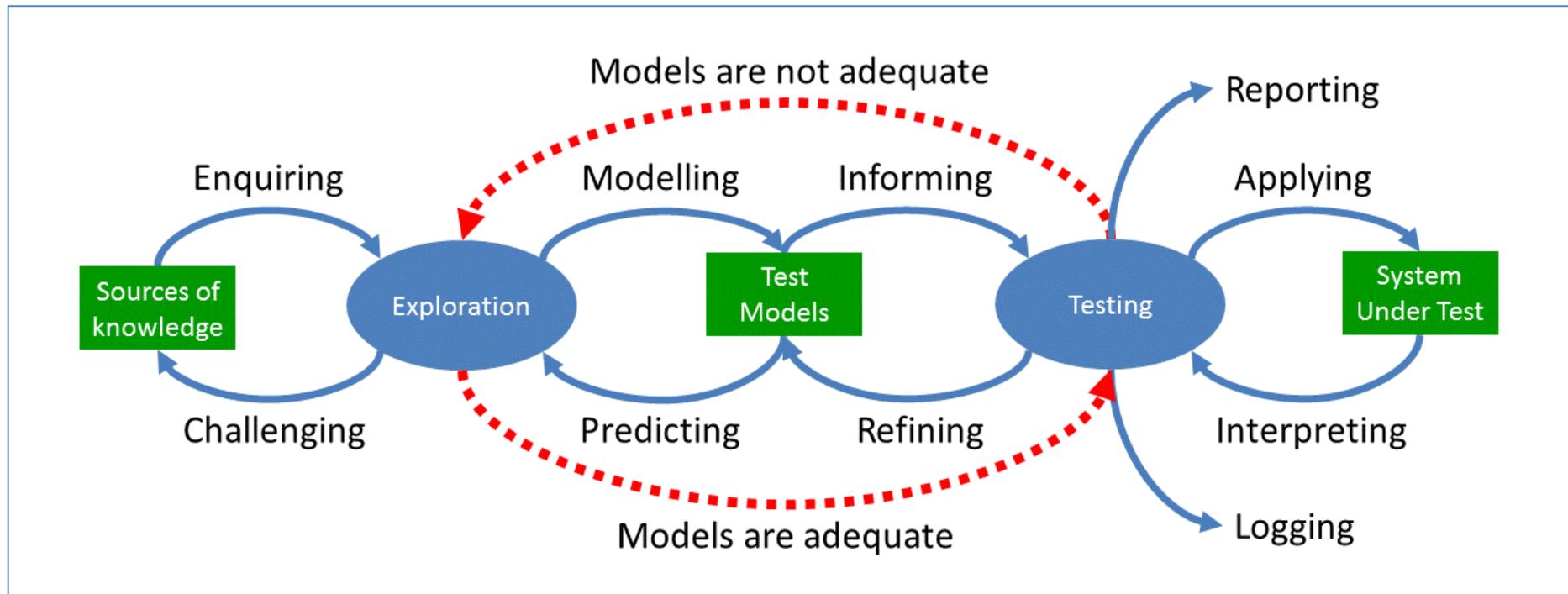


Figure 4 New Model Testing (with system revision removed).

6 Discussion

The New Model offers a new perspective on testing but might raise some concerns. In this section I'll try and pre-empt some more prominent concerns by asking and answering some obvious questions that arise.

6.1 Where do the Test Axioms fit?

The Tester's Pocketbook sets out sixteen areas which are germane to test strategy. I have been teaching test strategy workshops using the axioms since 2009 and, excepting context-specific issues, they provide a workable framework for the creation of a test strategy. The axioms can be viewed at the Test Axioms website [16].

The categorisation I applied in the original book aligns with the New Model, but not exactly. Perhaps I should revise the Pocketbook accordingly. At any rate, here is my suggested alignment of the three Axiom categories, and the New Model.

Stakeholder Axioms

- Testing needs stakeholders
- The value of evidence is for the stakeholder to decide
- If we don't manage scope we may never meet stakeholder expectations
- The scope of testing and acceptance are always compromises

New Model

- Overarching strategy/principle
- Overarching strategy/principle
- Overarching strategy/principle
- Overarching strategy/principle

Design Axioms

- Test design is based on models
- Testers need sources of knowledge to select things to test
- Testers need sources of knowledge to evaluate actual outcomes or behaviours
- Testing needs a mechanism for ordering tests by value
- Testing needs a test coverage model or models
- Our sources of knowledge are fallible and complete

- Exploration Process
- Exploration Process
- Overarching strategy/principle
- Overarching strategy/principle
- Exploration Process
- Exploration Process

Test Delivery Axioms

- The value of testing is measured by the confidence of stakeholder decision-making
- Some repeated tested are inevitable
- Run our most valuable tests first – we may not have the time to run them later
- Test execution requires a known, controlled environment
- Testing never goes as planned; evidence arrives in discrete quanta
- Testing never finishes; it stops

- Overarching strategy/principle
- Testing Process

6.2 Why not separate design and delivery in the New Model?

If the Testing Process consists of Design and Delivery activities, why not separate them in the New Model? In principle, the design and delivery could be separated. The rather complicated Testing Process could then perhaps have been simplified. There are arguments for and against this change.

If the New Model had three thinking modes, exploration, design and delivery then it might align more closely with the traditional, staged or waterfall view of testing. But this staged approach is not a good fit with 'exploratory testing'. The exploratory tester does not design all their tests and then apply them. Tests are applied soon after they are thought of – there is little or no delay between these two activities. In a waterfall project, there could easily be delays of weeks or months between design (and documentation) of tests and test execution.

However, if we take seriously the idea that documentation of tests is a logistical, and not a testing activity, then these delays do not exist and thought processes of staged and exploratory testing are the same. So the 'no delays allowed' argument of exploratory testers fades.

You might now ask, if you don't separate design and delivery, why did you separate the exploration process from the test process? My argument here is that all testers explore to build their models and then transition to the testing process. Exploratory testers are no different – they explore a feature in scope and then they make a judgement that their model(s) can be trusted to derive some tests from. Then they proceed to select and apply those tests.

The decision to transition from exploration to testing is a rather subtle judgement, whereas the transition between design and application of those tests is more mechanical.

6.3 Where does human judgement fit in the model?

The New Model, as presented is a set of thought processes, so human judgement cannot be separated from the many, many micro-decisions made in the exploration and testing processes. However, there are two points in the model where human judgement is critical and explicitly applied.

The decision to transition from exploration to testing is one of judgement as to whether the models being developed are sufficiently detailed and reliable to derive tests from. I have no guidelines to suggest yet, except that you must use your judgement. I am hopeful that a set of criteria for making the transition (in both directions) could be derived, but I have a suspicion that the dilemma to be resolved is akin to the 'who tests the testers?' quandary. More work is required here.

The other obvious place is the Interpreting activity. However a test is applied, by a human or by using a tool, the outcome requires human judgement to interpret. Automated tests, (whether just one or thousands are involved) that pass need minimal interpretation – most of the thinking was done before the tests were designed and applied. A single test failure or anomaly, however the test was applied, might take seconds or days to diagnose and understand – there is no telling how it might pan out.

6.4 How does the New Model relate to TDD, BDD and similar approaches?

Let's consider TDD first. It is best described in [17]. TDD is a development approach where program code is created in very small increments. Features are created through a number of iterations through the following sequence:

1. Decide on the feature or incremental functionality to add to a feature and create an automated test to cover it (in some way).
2. Run your tests. Watch the new test fail.

3. Write the minimum amount of code (or make the minimum change) to make the new test pass.
4. Repeat the test and if it fails go to 3 and try again.
5. Now all tests pass, consider refactoring the code. Refactoring might be a tidy-up, a minor or major re-design based on satisfaction with the current design. Go to 4.
6. If the feature is not finished, go to 1.

The TDD process requires that the developer has a clear understanding of the requirement, but not necessarily the design of the code. This iterative/incremental learning process has the following outcomes:

- All code that is written has at least one automated test to demonstrate its behaviour.
- All automated tests pass.
- The design of the code emerges from the process rather than the feature being fully-designed up front.
- If the process is followed and refactoring performed carefully, the design should be a good one.

How does the New Model relate to this approach? Well, the short answer is – it doesn't. The longer answer is that this process is not a test process, it is a design process. The developer will almost certainly explore sources of knowledge, scoping out one feature at a time, building up a model of how that feature should behave. But from then on, the key choice is, "What is the next incremental code addition or change required to advance the feature's development?" This is a program design choice even though creation of a test is the very next step.

The test that is created is chosen to show the incremental change is present and works at least once. The intent is to create scaffolding that provides confidence in the development process and a covering set of tests that allow the inevitable refactoring process to be done safely. Now, some developers might take care to add more tests than others to cover edge-cases or error handling, but this is down to the discretion of the developer. Perhaps it should be mandatory part of the TDD process.

There is a good case to be made for developers to learn the basic black-box and structural test-design techniques, but it remains that the core TDD approach does not map directly to the New Model.

TDD is also focused on components that can be tested with fast-running tests applied in a Continuous Integration regime, so components are normally isolated from databases, network messaging and even the user interface. TDD-developed code almost always requires additional testing through the user-interface, integration and system testing activities that do map to the New Model.

BDD is somewhat different [18]. In this case, specifications in the form of user stories written in a formal, domain-specific language, are used by programmers to generate their test code. From the test code, there follows a TDD-style approach to build up the required features, seeing tests fail, writing code to make tests pass, refactoring regularly.

Stories are written to capture requirements, or illustrate the behaviours of features by example, so that the three-amigos – BA, developer and tester – can evolve a common understanding of the features to be built. The examples or scenarios that illustrate features are written in language that

can be interpreted by tools such as Cucumber [19] or SpecFlow [20] to generate and execute test code that drives the TDD process.

There is direct parallel between the story writing process and the New Model. The collaborative specification is the exploration process. The output of the process is stories – the model, if you like. Scenarios or acceptance criteria in the stories are used to challenge the understanding of the requirement as well as illustrate it. The transition from specification (exploration) to coding requires the judgement and agreement of the participants. The tests derive directly from these scenarios.

However, the mapping to the testing process is less certain:

- BDD advocates and practitioners emphasise the value of collaborative specification over the value of the test automation, but also encourage practitioners to limit themselves to creating only ‘key examples’ (see for example, [21]) and not a covering set of tests. These test are regarded as ‘acceptance tests’. Test coverage may only be superficial – and other tests are likely to be required.
- Developers use the generated test code to drive their automated unit testing and are encouraged to write additional tests that use this test framework. Developers might not create tests that go beyond the need to create tests in the TDD style although it might be helpful if they did. Some more research and experience reports are required in this area.

Overall, BDD maps nicely to the exploration process, less well to the testing process.

6.5 Does the New Model describe Non-Functional as well as Functional Testing?

Modelling non-functional (NF) testing (including performance, security, usability, availability and so on) is a challenge because there are so many varied NF requirements and test approaches. It would be quite an achievement to derive a single, unified model that covers both functional and NF testing. The New Model intentionally focuses on functional testing but, in principle, could support the testing of some NF requirements.

The certification schemes separate functional and NF precisely because the logistics of each NF test type is distinctly different and as a consequence, they tend to offer only a superficial review of these techniques than a usable set off skills (of which more, later).

So is the New Model appropriate for NF testing? It remains to be seen, I think. Because the New Model de-scopes the logistics, it may be that it can be used to model most (if not all) of the NF testing approaches. Certainly NF testing has an ‘explore the sources of knowledge’ phase and modelling for security, performance, capacity/volume and availability for example is well-established.

The testing process – informing, applying, interpreting and logging activities seem to correspond with the more prominent dynamic tests like performance, reliability, availability and security. Usability and the ‘softer’ areas – it’s not clear.

Further work in this area needs to be done, and in particular, we need some experience reports.

6.6 How does the New Model relate to the “checking v testing” dichotomy?

Naming tests that can be scripted and performed by humans or tools as checks [22] does not inspire me, I have to say. Smart people I’ve discussed the idea with say it confuses rather than enlightens. Cem Kaner, in a blog here [23] ably criticises that ‘strange dichotomization of testing and checking’.

My criticism of the check v test definitions is that the definition of checking seems to describe evaluation of observations. Is this to be read as ‘performing a comparison’ or is it the interpretation of a positive, negative or near-miss comparison? The first can be automated, but the second cannot.

Does the definition of a check include its design *and* application? If so, then why doesn't the checking definition include 'questioning, study, modelling, observation and inference' – *these are all part of both testing and checking.*

If that isn't the intended interpretation, the evaluation by humans making a value judgement (testing) involves learning and checks performed using an 'algorithmic rule' do not. If a test does not cause learning – that is, if you learn nothing when a test is applied and interpreted – then I'd say that the test has no value.

A test that involves a comparison with an expectation (or a calculable outcome or oracle) seems not to be a test, but a check. *So is a test where the outcome cannot be compared with an oracle at all?* Are tests evaluations where the judgement is always subjective?

Confusing, isn't it?

An obvious challenge is to ask how, for example, Stephen Hawking tests. One of the smartest people on the planet cannot use their hands or voice, so would have to use proxy testers or tools following instructions to apply the professor's test. Hawking therefore cannot, by definition, test – he can only check. This makes no sense to me. The disabled professor can mentally model and describe a test for others to apply and for him to interpret. Is this test any less legitimate than the same test modelled and performed by an able-bodied professor?

I will continue to treat checks as 'tests that could be applied by humans or tools'. Only humans can interpret outcomes. Simple enough.

6.7 How does the New Model relate to 'exploratory testing'?

In the New Model, all testing involves exploration of sources of knowledge. But the popular view of exploratory testing [24] is as an unscripted/improvised activity performed only by humans where the *primary source of knowledge is the system itself.*

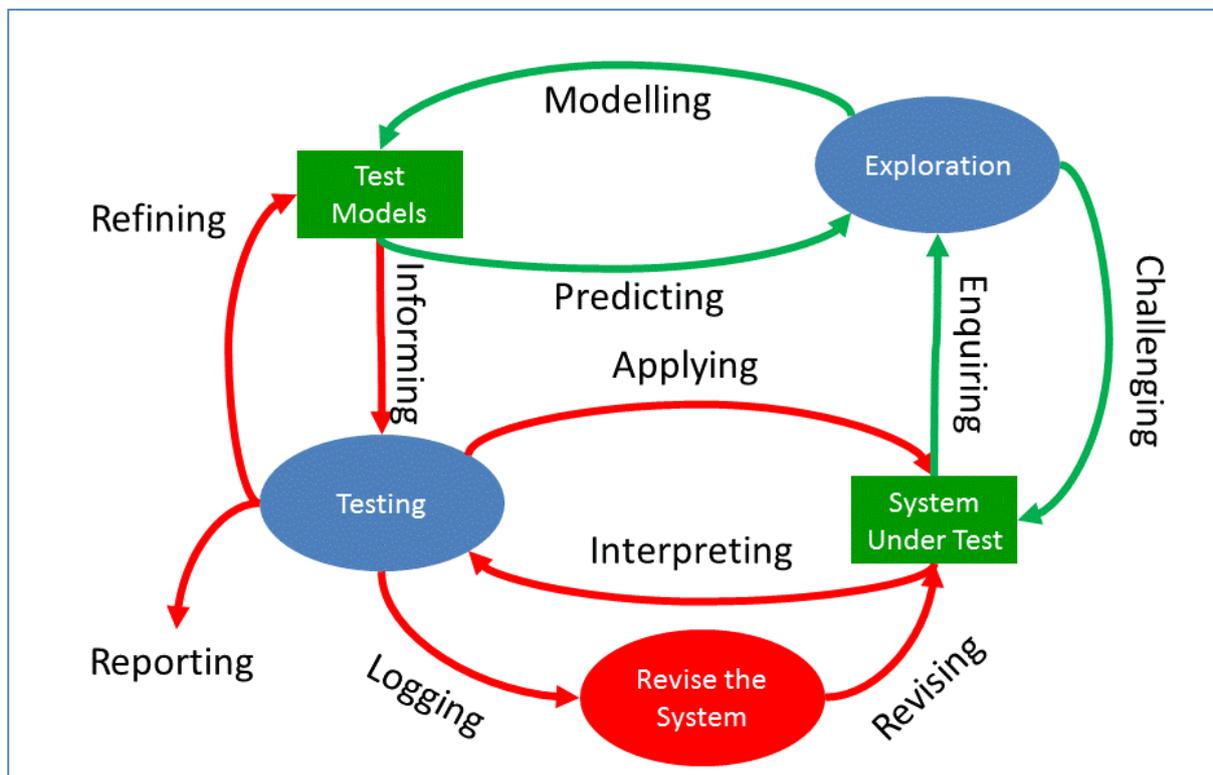


Figure 5 New Model and 'Exploratory' Testing

Let's consider a situation where the *only* source of knowledge is the system under test. In the New Model, the elements representing the source of knowledge and the system can now be merged into one. The model is basically 'wrapped around' and shown in the diagram above.

You can see that the topology of the New Model is unchanged. The same flows of thinking can be followed. The Judgement required to transition between the exploration process and the Testing Process is the same, although one would obviously expect the transition to be much more rapid than in a staged project.

If the system is the only source of knowledge, then the New Model stands up. But it is hard to imagine a situation where the system is the only source, even in a classroom exercise. For example, the tester always brings their experience and knowledge to their exploration. Stakeholders, users and developers are usually able to advise. In the case of an off-the-shelf package implementation, there is usually copious documentation, and so on.

To avoid confusion, we might need a new name for 'exploratory testing'? Perhaps something like 'Improvised Testing'?

6.8 Does the New Model change thinking about test automation?

We have already discussed the roles of developer unit-test automation in section 6.4. Does the model shed any new light on the use of automation at system level and/or using GUI test automation?

If we regard test record-keeping as logistics, we can ignore most 'test management' tools in this discussion.

The opportunities for automating the application of tests are unchanged by the New Model. The problems of test automation are not affected either. However, by separating the exploration and testing processes, it is obvious that whereas only humans model, the role and value of automation is clearly in the 'applying tests' activity. In Model-based testing (MBT), models are used explicitly to provide the data or at least a template for (usually) high volumes of tests. MBT fits the New Model.

The trend in the developer community, where TDD, BDD, Continuous Integration (CI) and DevOps cultures are increasingly popular is reducing the reliance on GUI test automation as an anti-regression measure. Developer tests, being maintained and managed in a CI regime provide most of the cover for regression problems, but the need for system-level regression testing remains.

At the system level, the motivation for *intentionally automated tests as a regression detection measure* is somewhat different from that of functional testing performed by humans. The information need is less to detect low-level differences in behaviour but rather that the core workflows in a system operate consistently across releases. I explore testing and other anti-regression approaches in a series of four papers [25].

There is no reason why the New Model should not be used to design a set of automated tests that have an anti-regression objective. It should be possible to create a model of the system to be instrumented for regression and use that to derive tests intended to be automated. More work is required in this area.

6.9 What about developer testing?

Most software (that is testable) does not have a UI. So developers have to use tools to apply sub-system tests whether they adopt TDD, BDD or not, but too often developers rely on late system testing to catch problems. The shift-left approach puts more emphasis on early, automated

developer testing than on late system testing. But really, the shift encourages teams to change their mind-set towards a test-early, test-often attitude. This is only to be encouraged.

Whether developers follow the test-first approach or not, the tools, mostly free to use, are widely available and proven. In the open source community, it is almost unacceptable to post code to public repositories without an accompanying automated build process and tests. If companies pursue the DevOps approach and/or eliminate independent test teams in favour of embedded testers, testing will be seen as an activity that is naturally part of software development rather than as a safety-net.

If developers and testers share some thought processes for their testing then this can only help to institutionalise testing in all development activities. Note that the model suggests similar processes are at work, not that developers and testers think the same.

6.10 Testing Skills – a different perspective?

I have hinted that by excluding the logistical activities from the New Model, then the processes can be both simplified and possibly regarded as universal. By this means, perhaps the core testing skills of developers and testers might coalesce. Testing logistics skills would naturally vary across organisations, but the core testing skills should be the same.

From the descriptions of the activities in the exploration and testing processes, it is clear that the skills required to perform them are somewhat different from the traditional view of testing as a staged activity performed exclusively by independent test teams. Perhaps the New Model suggests a different skills framework.

As a challenge to the status-quo, I have put together a *highly speculative list of skills that might be required*:

6.10.1 Exploration

- Analysis, enquiry and elicitation from varied sources of knowledge:
 - Documents: Business models, requirements, designs, standards, technical specifications
 - Systems: legacy systems, company portals, social media etc.
 - People: interviewing, listening, questioning, giving/receiving criticism
- Modelling
 - Use of existing standard model types, UML, financial models, business processes, workflows etc.
 - Creation of custom models, using heuristics, guesses, brainstorming, ideation, creative thinking and custom test design techniques
 - Comparison of models, value, advantages, disadvantages, compromises
- Prediction
 - Identification, validation and use of oracles
 - Predicate logic and proof
 - Hypothesis and inference
- Challenging
 - Requirements validation by example
 - The Socratic method

- Rapid Review and Inspection techniques e.g. DeFOSPAM, requirements animation, prototyping

6.10.2 Testing

- Informing
 - Test case design using the black-box, structural techniques
 - Test case design using ad-hoc models
 - Test models and the meaning of coverage
- Applying
 - Testing as controlled experiment
 - Scripted testing and observation
 - Exploratory/improvisational testing and observation
 - Note taking, recording
- Interpreting
 - Basic data analysis and statistics
 - Decision-making with incomplete data
 - Computer forensics
- Logging
 - Fault tree analysis
 - Failure diagnosis
 - Bug advocacy, triage processes and negotiation
- Reporting
 - Meaningful software and test metrics
 - Note taking, status reporting and feedback
 - Visual presentation of data
 - Reporting and presentation skills.

6.10.3 General

- Understanding stakeholders, their goals and concerns
- Deriving test ideas from goals and risks, designing feedback
- Production and test analytics
- Risk management, risk-based testing and decision-making
- Understanding compromise
- Critical Thinking
- Interpersonal skills
- Dealing with uncertainty/fallibility.

6.11 Does the New Model affect certification?

At the risk of kicking over a hornets nest, if the skills profile of testers changes there are some clear implications for certification and I should discuss them here. The certifications that exist are extremely popular but widely criticised. I don't want to go over the pros and cons of the existing schemes but I will summarise the situation as:

- Certifications are popular, the market for certified tester training is strong.
- The value of current certified training courses is doubted by many: The schemes promote best practices that are often inappropriate; they are stuck in the past; they promote testing-by-rote rather than creative thinking.
- Examinations are syllabus memory tests more than tests of capability.
- The schemes are evolving, but extremely slowly.
- There is suspicion of and dissatisfaction with the certification boards.

From my own experience of teaching certified courses, the most valuable aspects of these courses are the test-design techniques. Students certainly find these the most rewarding during the class. But even though many thousands have taken these classes and passed exams, the number of people who use the techniques directly is extremely small. I think this is because the techniques are taught as clerical methods. Classes do not teach modelling – they teach how to follow a selected model by rote. Until the schemes include modelling as a core skill, certified training will be limited in its value.

There is a place for certification as part of professional development schemes, but there is no clear or agreed definition of what our profession is. Professional certification ought to be a mix of training, evaluation, relevant and recent experience and peer review rather than training and multiple choice exams. Few companies have meaningful development schemes for testers because of the currently confusing situation.

The biggest problem with the existing schemes is that their syllabuses are dominated by what I have called testing logistics, and particularly on waterfall, staged or factory approaches. By removing the testing logistics, a New Model of testing emerges and perhaps better certification syllabuses can be designed.

I hope the New Model and speculative skills inventory trigger some debate and new thinking in this area. It is overdue.

7 Conclusion

I believe that our existing models of testing are not fit for purpose – they are inconsistent, controversial, partial, proprietary and stuck in the past. They are not going to support us in the rapidly emerging technologies and approaches.

In this paper, I have suggested a New Model of testing that might be a useful framework for thinking about testing and how testers think. I have tried to be consistent with the intent and content presented in the two pocketbooks [3, 9]. Some more obvious challenges to the model have been considered and discussed.

The certification schemes that should represent the interests and integrity of our profession don't, and we are left with schemes that are popular, but have low value, lower esteem and attract harsh criticism. My goal in proposing the New Model is to stimulate new thinking in this area.

The model has been presented to substantial conference audiences in Finland, Poland and the UK during April, May and June this year. It was challenged and debated by test managers, senior testers and consultants in a workshop at the Test Management Summit in April. The feedback and response has been notably positive in all cases. I am planning to present the New Model at several more public and company-internal conferences in the UK and elsewhere during 2014.

This is a work in progress. I am actively seeking feedback and guidance on the New Model and the narrative in this paper.

8 References

1. Testing is in a Mess, Paul Gerrard, <http://gerrardconsulting.com/?q=node/591>
2. Will the Test Leaders Stand Up?, Paul Gerrard, <http://gerrardconsulting.com/?q=node/621>
3. The Testers Pocketbook, Paul Gerrard, <http://testers-pocketbook.com/>
4. All Testing is Exploratory, Paul Gerrard, <http://gerrardconsulting.com/?q=node/588>
5. Definition of Test, American Heritage Dictionary, <http://www.ahdictionary.com/word/search.html?q=test&submit.x=37&submit.y=17>
6. Definition of Test, Testing, Paul Gerrard, <http://testaxioms.com/index.php?q=node/3>
7. Software Testing Techniques, Boris Beizer, 1990
8. Lessons Learned in Software Testing, Kaner, Bach, Pettichord, 2002
9. Business Story Pocketbook, Paul Gerrard & Susan Windsor
10. The Domain Testing Workbook, Kaner, Padmanabhan and Hoffman.
11. Exploring Requirements, Donald Gause & Gerald M. Weinberg
12. Testing Requirements, Paul Gerrard, <http://gerrardconsulting.com/?q=node/516>
13. Scenarios, Stories, Use-Cases, Ian F. alexander & Neil Maiden
14. Mastering the Requirements Process, Suzanne Robertson & James Robertson
15. Gherkin story format, <https://github.com/cucumber/cucumber/wiki/Gherkin>
16. Test Axioms website, <http://test-axioms.com>
17. Test-Driven Development by Example, Kent Beck.
18. Behaviour-Driven Development, https://en.wikipedia.org/wiki/Behavior-driven_development
19. Cucumber website, <http://cukes.info/>
20. SpecFlow website, <http://www.specflow.org/>
21. Focus on key examples, Gojko Adzic, <http://gojko.net/2014/05/05/focus-on-key-examples/>
22. Testing and Checking Refined, James Bach, <http://www.satisfice.com/blog/archives/856>
23. The Insapience of Anti-Automationism, Cem Kaner, <http://context-driven-testing.com/?p=69>
24. Exploratory Testing, Wikipedia Definition (with references), https://en.wikipedia.org/wiki/Exploratory_testing

25. Regression Testing and Automation, a series of four articles, Paul Gerrard,
<http://gerrardconsulting.com/?q=node/479>

THE TESTER

SIGiST Autumn Conference Tuesday 15th September 2015

54

IN THIS ISSUE

Conference

Agenda	2
AGM	3
Venue	5
Abstracts.....	6

Articles

May I get personal?.....	15
-----------------------------	----

Software Quality Beyond In-house Code	19
---	----

Is It Time for an ISTQB Software Localisation Testing Certificate?	28
---	----

Regular Features

Fifteen minutes with	24
-------------------------------	----

Testing Events Listing	14
---------------------------------	----

From the Editor

Welcome to the autumn edition of The Tester. For the third SIGiST conference of 2015 we return to the previous venue of the BCS Offices in central London. See page 5 for details on how to get there.

The September conference will host the SIGiST AGM. Taking place before our regular proceedings, there are a number of committee positions up for election in September. Check out the AGM notice on page 3 for full details.

What a conference programme we have lined up for September! Opening with Stevan Zivanovic and closing with Julian Harty - both keynotes are sure to be interesting. There are two workshops this time round, but places are limited so sign up now! Check out all the articles in this edition, one accompanies one of the workshops.

We are always looking for speakers / workshops for the conference, and articles for The Tester. If you want to speak check out the SIG website: <http://www.bcs.org/category/10880> or contact me if you want to become a published author.

Phill Isles
The Tester Editor
phill.isles@bcs.org

Conference Booking Instructions

To register online, please use the link below, or scan the QR code with your smart device. Please note the BCS booking system accepts multiple and third party bookings.

<https://events.bcs.org/book/1406/>



Join our Linked-In Group:

<http://www.linkedin.com/groups?mos tPopular=&qid=3466623>



Follow us @SIGiST



Conference Agenda

BCS SIGiST - Autumn 2015 Conference Tuesday 15th September 2015 BCS, First Floor, The Davidson Building, 5 Southampton Street, London, WC2E 7HA.		
08:45	Coffee & Registration	
09:10	BCS SIGiST Annual General Meeting 2015	
09:25	Introduction and Welcome Stuart Reid, Chair, SIGiST	
09:30	Opening Keynote	
	Adapting Agile Practice Stevan Zivanovic	
10:30	Open Microphone and Networking session Chaired by Jen Wheeler, Networking Secretary, SIGiST	
10:45	Coffee, Tea & Refreshments	
11:15	Using Industrial Placement Students as part of your Test Organisation Matthew Cardle & Alex Hill, IDBS	Morning Workshop The Consequences of your Unconscious Mind Gillian Arnold, Tectra
12:05	Performance Testing with JMeter at Skyscanner Indu Nair & Pete George	
12:50	Lunch	
13:50	All aboard for a Quality Journey Kiruba Vijayaraghavan, Sita Aero	Afternoon Workshop Stick, Carrot or a Softer Approach? Elaine Sullivan, Skybrook Consultants
14:45	Changing face of test management (in an agile world) Tom Roden, Neuri	
15:30	Coffee, Tea & Refreshments	
16:00	Closing Keynote	
	Software talks – are you listening? Julian Harty, Commercetest	
17:00	Closing Remarks Stuart Reid, Chair, SIGiST	

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops will have limited places.



Specialist Group in Software Testing

The BCS Specialist Group in Software Testing

Notice of Annual General Meeting

Notice is hereby given that the Annual General Meeting of the BCS Specialist Group in Software Testing (SIGiST) will be held on Tuesday 15th September 2015. The venue for this meeting will be the BCS, First Floor, The Davidson Building, 5 Southampton Street, London., WC2E 7HA.

Agenda

- ❖ Welcome and Introductions
- ❖ Apologies for absence
- ❖ Minutes of the 2014 AGM (and matters arising)
- ❖ Reports
 - Chair
 - Treasurer
 - Standards committee
- ❖ Committee elections
 - Treasurer
 - Programme Secretary
 - Sponsor / Vendor Co-Ordinator
 - Marketing Co-Ordinator
 - Tester Editor
 - BCS Liaison & Network Coordinator
- ❖ To consider any nominated business

Items for inclusion on the AGM agenda should be emailed to maureen.shannon@bcs.org. Additions to the agenda must be received no less than fourteen days prior to the meeting.

SIGiST Election process

Elections will normally take place at the SIGiST Annual General Meeting (AGM) in September. In extraordinary circumstances (e.g. early resignation) the SIGiST committee has the power to invite someone to take on any of the vacant roles until either the AGM or an Extraordinary Meeting when the role will be filled using the election process described here.

Elections are required in two sets of circumstances:

1. Automatically after a SIGiST Committee member(s) has held a position for 3 years.
2. If a SIGiST committee member resigns before the completion of their 3 year tenure.

The basic process to be adopted for any election follows:

Task	Timescales
When an election is to take place at an AGM the available positions should be announced. Otherwise, for an Extraordinary Meeting, an email will be sent to all registered email addresses on the SIGiST database announcing the election(s).	No later than 30 days prior to the election.
The name of any member accepting nomination for election or re-election as an Officer or as a Committee member should be submitted in writing to the Secretary, with an accompanying short manifesto (no more than a page of A4) describing what they expect to bring to the role, by two members of the Group and with the written consent of the nominee. See the Member Group Rules (http://www.volunteer.bcs.org/Rules) for further details.	At least 20 clear days prior to the election (after this point no more applications will be accepted).
A list of applicants for each job is released to the SIGiST members via email together with their manifestoes.	At least 10 days prior to election.
Election takes place during AGM or Extraordinary meeting.	At the AGM or Extraordinary Meeting.

Rules
1. Each candidate may stand for as many positions as they want (and can vote for every position available – subject to items 4 and 5 below), but may only hold one position. In the event that someone is elected to more than one position then they must immediately decide which position they wish to take up and vacate the other positions. The second-placed candidates for the vacated positions are then elected to those positions.
2. Should the nominations number equal to or less than the vacancies, the nominees will be deemed to have been duly elected without an election.
3. A simple majority is required to be elected to a position.
4. Only members as defined in http://www.volunteer.bcs.org/Rules may vote.
5. Voting is only allowed if the member is physically present at the AGM
6. The formal voting process will take place on the day of the meeting (a simple show of hands).

SIGiST Conference Venue

For the September 2015 conference, the SIGiST returns to the BCS London office. Travel details and location below.

London office guide

How to get to the BCS London office

First Floor
The Davidson Building
5 Southampton Street
London WC2E 7HA
Telephone 01793 417666

These area and inset maps have been simplified in the interests of ease of understanding. Not all roads are shown. The inset map below is more accurate.

Access by car is very difficult due to the local one-way system. There are no car parking facilities at BCS London. The nearest car park is located on Drury Lane, Parker Street, Parker Mews, London, WC2B 5NT.

The rear door in Exeter Street is to be used for deliveries only and is normally locked.

The main entrance is fully accessible to wheelchair users and should be used by all staff and visitors.

On arrival, report to the Davidson Building Reception who will direct you to the first floor.

Travel tips from major London stations

Charing Cross – 6 minutes walk

Waterloo – 12 minutes walk across Waterloo Bridge, or buses 139 or 176 to Stop **C**

London Bridge – onward rail link to Charing Cross

Kings Cross or St Pancras – Piccadilly Line to Covent Garden tube, or bus 91 to Stop **B**

Euston – West End Branch of Northern Line to Charing Cross, or bus 91 to Stop **B**

Victoria (rail and coach stations) – Circle Line to Embankment, but the most direct journey is via bus 11 to Stop **A**

Paddington – Circle Line to Embankment or Temple, Bakerloo Line to Charing Cross or buses 15 or 23 to Stop **A**

Liverpool St – Circle Line to Embankment or Temple, or buses 11 or 23 westbound

Fenchurch St – Walk to Tower Hill, then District or Circle to Embankment



Presentation Abstracts and Speaker Biographies

Opening Keynote

Stevan Zivanovic “Adapting Agile Practice”

I am an ardent fan of Agile. I have spoken on many occasions about the benefits and practices. However....

In the real world, with enterprise companies, you get:

- Some things take a while to build - months if not years.
- Projects have multiple vendors, all with their own approach and methodologies.
- Sequential project management is still easier in concept and some managers like it.
- Teams are distributed around the world, different times-zone, different cultures.
- Some senior stakeholders mistrust Agile.
- Aspirations to be Agile, but no knowledge of what it means

So how do you do "Agile" in this context?

This talk will focus on the methods I have employed in these types of organisations to implement practices that give the real benefits of Agile, without an organisation "going Agile".

Stevan Zivanovic has a passion to support people to achieve their best. His long career in the IT industry has taken him from a practitioner in the discipline of software testing to a management consultant; advising, supporting and enabling individuals and teams to change and deliver. He has used Agile practices and technical solutions to successfully deliver projects (large and small) and translated these experiences to facilitate others to achieve.

Closing Keynote

Julian Harty

“Software talks - are you listening?”

We have reached an age where running software can tell us much about how it's being used, in ways that extend, augment and enhance our software testing. Particularly with mobile apps, it's impractical to 'test' the software across the wide variety of conditions and on the galaxy of devices it may be used on. Instead, runtime information can help us discover potential problems and use the information to improve the app, and our development and testing practices. There are various sources of information, we will compare and contrast them. We'll also investigate how Mobile Analytics can help us improve how we test, what we test and when we test if we learn how to apply it effectively in our applications and our processes.

In this talk, Julian will introduce ways of using mobile analytics as an integral part of testing your mobile applications. Well-designed analytics can help assess various software qualities, including performance, reliability, and even usability. He will cover practical aspects, challenges, concerns, together with examples of how using mobile analytics can help listen more effectively and clearly to what our software is experiencing when used by people worldwide. Testers can, and should, play a pivotal role in the application of analytics. Come on, it's time to get involved!

Julian Harty has fun testing. Over the years he's worked for various organisations, including Google, eBay, Klarna, and Salesforce; run startups and companies; published books on testing mobile apps, and spoken at a bunch of conferences, including the SIGiST on several occasions. For fun, he's doing a PhD on Mobile Analytics and applying the research and concepts with several global companies. He contributes and publishes open materials, including Selenium, Robotium, and many others. He's also passionate about helping people live better lives through using [mobile] technologies. You can read more about his work online e.g. <http://kusaidiamwalimu.org> and blog.bettersoftwaretesting.com

Morning Workshop

Gillian Arnold

“The Consequences of your Unconscious Mind”

The BCS has been leading the charge on establishing great diversity policies in the Tech Sector. We recognise that where there are diverse teams there are happier staff, and where there is diversity at the highest levels of an organisation, innovation and profit is increased. We want to get the message out to all of our specialist groups and committees that diversity in decision making is good for the industry and good for us all. Gillian will help us to understand how our own unconscious processes can influence more than we would believe, and will show the impacts of implicit bias in the workplace and beyond.

***Gillian Arnold** has extensive experience in the IT industry and setup her own IT Services and Staffing company, Tectre, in late 2009. Tectre provides staffing, training, and consultancy to the IT Vendors, Distributors and Resellers and has recently been offering consulting on Energy Efficient Computing. Tectre has just completed a government sponsored research project into Energy Efficient Compute with the University of Huddersfield. Tectre provides diversity focussed recruitment for positive action campaigns, and Unconscious Bias training and development. The company is also supporting women returners to the STEM industries by providing focused training and career advice for returnees.*

Most of Gillian’s early career was spent working for IBM in the UK and she held customer facing training, technical, sales, business development, strategic marketing and consultancy roles. She proved her expertise in managing and establishing teams for new software and hardware products, building teams with cross-industry and cross platform experience across the UK and Europe.

Gillian has a long involvement with work for Women in IT, supporting both industry and academic institutions in their work on diversity. She currently sits on the board of directors for WISE, the UK organisation which supports women in Science, Engineering, Technology, Maths. Gillian is chair of the BCS Institute for IT group: BCSWomen and is driving the BCS initiatives in support of women in the Technology Sector. Gillian won the 2012 Cisco / Everywoman In Technology award for Technology Inspiration of the year.

Afternoon Workshop

Elaine Sullivan

“Stick, Carrot or a Softer Approach?”

Gone are the days where managers routinely dictate to their staff and even bonuses or promotion opportunities appear to have limited effect in the ongoing motivation of personnel.

Through use of practical exercises, case studies and discussion, this workshop promotes an alternate approach to the 'stick and carrot' and introduces a 'softer', more motivational method of working based upon personal commitment, group dynamics and common aims.

Workshop Purpose:

To provide you and/or your staff with a set of tools and strategies to:

- Understand and overcome key factors that hinder individuals from meeting their potential.
- Defeat procrastination 'The Thief of Time'.
- Rise above limiting beliefs to have more, be more and do more in less time.
- Set up a structure that empowers and galvanise individuals into action and thereby enhance productivity as a result.

***Elaine Sullivan's** career has been varied, from being the UK Network Manager for Compaq to undertaking roles such as Test Manager, QA Manager and Programme Manager for a number of high profile companies and public sector clients.*

Using the experience gained from these organisations, Elaine now offers Business and Personal Development Training to individuals and companies alike. By providing a 'softer' approach to motivation there is usually a significant and immediate impact on staff productivity which, in turn, benefits both the individual and business alike.

Don't miss Elaine's accompanying paper, later in The Tester.

Track Session

Matthew Cardle & Alex Hill

“Using Industrial Placement Students as part of your Test Organisation”

IDBS has been taking on industrial placement students as part of our Testing team annually for almost a decade. We have found that this has been beneficial both to us and to the students involved; often resulting in returning permanent staff. Generally during their placements students have worked well as valued members of the team and have helped make our department diverse and vibrant. They have also played their part in keeping our software industry leading.

In this talk we will discuss our experiences of employing students within Software Testing, the good, the bad and the drunken! Utilising real world examples - both from the perspective of a Test Manager and of an ex-student who has since joined the team full-time we aim to give you an insight into student industrial placements in software testing and whether or not they may be useful to you and your business.

***Matt Cardle** is Test Delivery Manager at IDBS, where he has been working since 2008. IDBS has given him not only the opportunity to work alongside a great team, but the perfect opportunity to marry his technology and scientific backgrounds. During his time at IDBS, Matt has been involved in recruitment and has used placement students within his department to great benefit.*

***Alex Hill** is a Test Analyst at IDBS, having entered into the world of software testing as a student tester on an industrial placement. After finishing her degree she returned as a permanent software tester. Since then she has been interested in highlighting the potential benefits industrial placements can offer to both Students and testing departments and how we can get more of the best student talent involved in, and excited about, software testing as a career path after their studies.*

Track Session

Indu Nair & Pete George

“Performance Testing with JMeter at Skyscanner”

This presentation provides an introduction to some of the key features of the open source testing tool, Apache JMeter, and how it has been used very effectively in performance testing at Skyscanner.

Indu Nair works as a QA Engineer at Skyscanner Limited. In her current role, she is involved in website backend testing and has been doing Performance testing as a part of this.

In the last 5+ years, she has been involved in all aspects of system development life cycle from requirement gathering to commercial launch.

Indu has been involved in testing applications across a wide range of devices including TV's, gaming consoles and mobile devices.

Pete George is an experienced test specialist and trainer. In 2014, he provided Skyscanner with JMeter training and, following their successful adoption of JMeter performance testing, suggested that this might be a good story for SIGiST.

Pete predominantly works in Agile contexts and has found JMeter to be a highly flexible tool for rapid, iterative development.

Track Session

Kiruba Vijayaraghavan

“All aboard for a Quality Journey”

SITA is a specialist in air transport communications and IT, an industry where software has transformed the customer experience, safety and security like no other. Quality is of paramount importance to ensure an ever increasing complexity in our products only impacts the 2800 plus customers positively. Our challenge through the years has been to consistently deliver the quality 'on which life depends' more cost efficiently.

This talk is about our journey to transforming a globally distributed QA team, supporting a variety of software delivery lifecycles and next generation technologies, from being a cost centre to a business value driven organisation. We bring you certain approaches that were innovative and yielded maximum returns:

- The organisation was re-aligned to a business demand driven service organisation.
- Our engagement with offshore partners shifted from being capacity based to a result oriented one in which they are rewarded on the basis of test output.
- A Just in time progressive automation approach was implemented, in which test automation scripting was based on the testing needs in future cycles. The return on investment was calculated for every 'automation story' and realised benefit was tracked on a monthly basis.
- Post the CMMi Level 3 Certification our strategy was to shift left - quality assurance at customer contract stages, Risk based testing, end to end traceability of customer requirements and static code analysis.

At the core of efficiency was our award winning test methodology for agile delivery by distributed teams including offshore locations for very large programme.

Kiruba Vijayaraghavan, Senior Manager at SITA has more than 15 years of IT testing experience across industry verticals and technology. He owns the central strategy for offshoring testing services and drives improvements in delivery & supporting tools. He specializes in Assessment and implementation of Test Centre of Excellence, and programme management of testing for large implementations. He has shaped Functional, Automation and Performance Testing, Data Warehousing & Business Intelligence Testing projects for Fortune 500 clients. He has defined the process and technology implementation of Test Management, Configuration Management and Project Management tools.

Track Session

Tom Roden

“The changing face of Test Management (in an agile world)”

Test management doesn't exist in the world of agile, or rather test managers don't, or do they? Agile methods such as Scrum have many traditional test management activities built in. With practices like self-organizing teams, role blurring, and skill diversification, the face of test management is changing. But is that a bad thing? Tom Roden explores the key tenets of test management in an agile context, the likely dispersal of traditional responsibilities, and the profound effect on teams and managers. Hear his first hand experiences, some new and radical ideas, and research from test management practitioners worldwide in organizations transforming to agile methods. As a test manager and leader, learn how to prepare yourself to adapt and thrive in a changing landscape. As an agile tester or team member, challenge yourself to answer questions about the maturity of your team's testing capability.

*A partner at Neuri Consulting, **Tom Roden** is a software delivery coach, consultant, and quality enthusiast, helping people make changes needed to thrive and adapt to their ever-changing environment. Influenced by agile and lean principles, Tom collaborates with teams intent on delivering high quality software with speed and predictability. He helps test managers and leaders establish testing capabilities that can deliver on the promise of agility. Learn more at www.neuri.co.uk. Follow Tom @TommRoden.*



Event Listings

If you would like your event listed here, please contact the Editor phill.isles@bcs.org

2015

September

SIGiST

15 September 2015
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

November

EuroSTAR

2 – 5 November 2015
Maastricht, Netherlands

<http://www.eurostarconferences.com/>

Agile Testing Days

9 – 12 November 2015
Potsdam / Berlin, Germany

<http://www.agiletestingdays.com/>

December

SIGiST

2 December 2015
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

2016

March

SIGiST

15 March 2016
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

May

STAREAST

1 - 6 May 2016
Orlando, US

<http://stareast.techwell.com/>

Belgium Testing Days

TBC May 2016
Brussels, Belgium

<http://btdconf.com/>

June

SIGiST

9 June 2016
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

September

SIGiST

TBC September 2016
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

December

SIGiST

TBC December 2016
London, UK

<http://www.bcs.org/server.php?show=nav.9264>

May I Get Personal?

Elaine Sullivan, Skybrook Consultants Limited

Do you see yourself as an 'Individual'...

... or are you merely a 'Resource'?

One of the greatest contributions of the Agile Manifesto and Extreme Programming was its emphasis on people however, somewhere over the years, businesses have moved from having a 'Personnel' Department to having a department that deals with Human 'Resource'.

Thankfully, in most companies we have stopped having to literally 'clock on' but, in return, we now have to adhere to many more documented processes, codes of conduct, checkbox activities and an increasingly automated environment that can seem to intrude into people's daily jobs. Through it all, the John Smiths and Jane Jones of this world seem to have transformed from being 'Personnel' that were once known throughout the company to simply being a 'Resource', someone who can be booked for a role in the same way as one can book an overhead projector for a meeting.

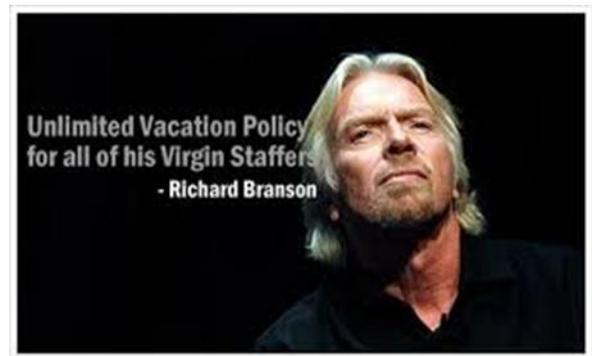
At the same time, they feel pressured to perform well, work hard, sometimes **too** hard but, as a result of the constraints placed upon them, feel disempowered and, sometimes, undervalued. They work for long hours, even when the company has embraced the Agile methodology with its main tenet to maintain a 'consistent and sustainable pace'. And I bet that if I could see the responses to the question "Who here reading this article, regularly does more than a 40 hour week?", I'd see many of you now sitting with your metaphorical hands up.

And what of those other companies that haven't adopted Agile, or indeed businesses

that are outside the software arena, can **they** still aspire to the benefits of a fully motivated, collaborative, productive team with individuals holding themselves personally responsible and accountable?

This mentality doesn't come about as a result of laying down more rules, adopting new guidelines or the threat of a stick and, increasingly, it cannot be fostered by offering the carrots of benefits, money or even **more** money.

The world of the employee and employer is changing, one of the world's most influential businessmen, Sir Richard Branson of Virgin fame, is a very vocal advocate of treating his staff as individuals.



Take Virgin's policy of allowing people to have unlimited leave providing they get their work done; this was discussed as part of a talk given externally to Virgin. Initially, during the discussion the idea didn't appear to receive a positive reaction, however on talking to the employees later they said that they were too frightened to clap as several of their bosses were in the room.

So ...

What would it take for you or your company to give your employees the freedom to make

their own decisions, to be able to express themselves and their thoughts and, as a result, feel more in control of themselves, their career and their own lives? And what might that do to their productivity and in turn the productivity of the team or the company?

Part of the empowerment of a company's staff can be achieved by cultivating a culture of leadership and collaboration rather than that of management and servitude, and by encouraging an environment where each and every person can believe in themselves and their ideas, where they can have the certainty that as individuals they, personally, **can** make a difference.

Of course someone needs to be at the helm, **of course** someone needs to make the tough decisions but, in general, with the right overarching theme for the business, with goals and tasks that are meaningful to the staff expected to undertake them, huge strides towards greater productivity can be made.

Consider Ryanair for example. Do you remember the headlines?

“Ryanair confirms it WILL bring in charges for on-board toilets”

Cited by The Daily Mail around the beginning of April in 2010, and reported across the media, Ryanair were said to be pushing forward with their plans to not only charge for the use of toilets but also reduce the number of toilets on the plane. This would then enable them to increase the number of seats on planes that were already cramped and offering little leg room, all this to make more money! Apparently, Michael O’Leary, Ryanair’s boss, made the comment in 2009 but seemingly then ‘backtracked’ on the idea.



Given the timing of the article, I believed it could have been an April Fools’ joke, but the thing was it was **plausible**, and, with a jaded opinion of Ryanair and their money making schemes, I, and probably many others too, believed this was ‘just the thing that they’d likely do’.

Customer opinion of Ryanair and their service was pretty much at an all-time low. If a bag was 5cm too large, the customer would be told it would have to go into the hold (at extra cost), if the bag was just a few hundred grams overweight they’d be told they’d have to pay extra for that too, if the customer was to arrive two minutes after the gate was closed that was it, they missed their flight and had to book another (again at further expense). The customers were frustrated, the interactions at the desk were fractious; whilst the staff may well have wished to deal with the customers’ frustration they had not been empowered to do so.

With this as the background, in 2014 the company announced a new ‘Customer-Friendly’ ethos, and at the same time, it would appear they realised that their front-line customer service team, the men and women on the desk talking directly to the customers, were probably best placed to make the decision as to whether something such as a slight contravention of the rules of size or weight or allowing a late arrival at the desk, would be the most pragmatic and customer friendly approach.



The net result of this 'Charm Offensive' as the Guardian called it in their article of late May, or in Michael O'Leary's words, the move to "stop unnecessarily pissing people off" is that passenger numbers over the year are up by 11%, significantly more than the projected target increase of 4%, and, whilst in part due to reduced fuel costs, profits have also jumped by two-thirds.

Whilst I can find no documented evidence to back up my belief, I would also like to think that this 'offensive' has lowered the staff's stress levels, raised their job satisfaction and enhanced the staff's belief in themselves and that they can make a difference. I also believe that working at Ryanair is probably a much happier place with more motivated staff and that they take pride in personal responsibility and now love to be accountable for their actions as they make a difference to their customers.

Perhaps Ryanair have also taken a leaf out of another airline's book; Sir Richard Branson famously said,

"Train people well enough so they can leave, treat them well enough so they don't want to".

Sir Richard Branson really understands that his staff want to be treated as 'Individuals' not as 'Resource' and that as a result of doing so, his staff are happier and, in turn, his companies benefit.

So ask yourself, could you or your company benefit from a similar ethos? Would you like to see how by adopting the same attitudes might be advantageous and how, by understanding what really motivates people, overcoming the mental barriers to success, and empowering ourselves and our staff can increase productivity? If so, come to the SIGIST Autumn 2015 Conference, on Tuesday 15 September 2015 where I will be conducting a workshop, 'Stick, Carrot or a Softer Approach' and we can explore these concepts together.

Elaine Sullivan's career has been varied, from being the UK Network Manager for Compaq to undertaking roles such as Test Manager, QA Manager and Programme Manager for a number of high profile companies and public sector clients.

Using the experience gained from these organisations, Elaine now offers Business and Personal Development Training to individuals and companies alike. By providing a 'softer' approach to motivation there is usually a significant and immediate impact on staff productivity which, in turn, benefits both the individual and business alike.





The EuroSTAR Conference Early Bird discount deadline is September 25th! Don't forget to take advantage of the additional discount allocated to **BCS SIGiST** members.

Don't know EuroSTAR?

Never been to the EuroSTAR Conference? Here's a taste of what you can expect: [Trailer](#)

Calling all BCS SIGiST Members! – don't forget your special discount.

As a member of BCS SIGiST you can avail of a further 10% off your conference registration. This 10% discount code can be used at any time when registering but if you register on or before September 25th you could save over €350 – over 19% off!! Make even more savings with group registrations (of 5 or more) where every fifth attendee goes for free!

Register using the discount code: **SOBC10**

Are you looking to be inspired in 2015? EuroSTAR is the place for you!

Long established as Europe's #1 software testing conference, EuroSTAR features keynotes and tutorials from global thought-leaders, dozens of insightful track sessions and the opportunity to network with hundreds of software testing professionals. Be there in Maastricht to celebrate everything that's great about your profession over four intensive days of knowledge sharing in a vibrant atmosphere! This year the conference takes place at the MECC Maastricht from November 2nd – 5th 2015.

The 2015 Programme offers practical advice, real-life experience stories and thought-leading insights into DevOps, Mobile, Management, Communication, Agile, Test Automation and more – presented by a mix of experienced speakers and newcomers to EuroSTAR. You can see the full programme [here](#).

[Register Now](#)

Or if you have any questions about registration / pricing please contact siobhan@eurostarconferences.com

Mobile Deep Dive 2015

EuroSTAR have added a brand new bonus event on mobile testing – EuroSTAR Mobile Deep Dive, which takes place on Friday 6th November.

Tickets to Mobile Deep Dive cost €850 (normal group rates apply) and if you register to attend the full EuroSTAR Conference (any Tues-Thurs inclusive ticket) you can also get a 50% discount on a ticket to the Mobile Deep Dive Event. [View Mobile Programme](#)

Be there in Maastricht!

Software Quality Beyond Testing In-house Code

Klaus Haller & Rudolf Grötz

Abstract

Crucial software fails and management needs someone to blame? Blame the testers! They should find bugs before they get into production! Yes, even testers make mistakes. They might even sign-off software they are not convinced of. More often, however, issues are not caused by bugs in the tested code, but by other factors. This article discusses three of these factors. First, there are configuration parameters. They impact test coverage and test processes. Second, relying on suppliers implies specific quality risks. Third, the business aims themselves can cause issues. The aim of this paper is to discuss these three points and to provide a solution by enhancing standard software change processes.

Configuration Parameters: When software suddenly turns mad!

Configuration parameters allow the adaption of software behaviour quickly, if business needs change. Also, they ensure repeatable installations. Repeatable means that when the installation is complete, it is in a carefully defined state; it can be reconstructed for future test cycles. Thus, all installation parameters such as paths, Java Virtual Machine settings, timeout periods *etc.* must be put into an installation parameters file. Based on this file, a batch job performs the actual installation. This is one core idea of DevOps [1]. When test and production systems are set up as similar as possible,

this reduces “production only bugs.” Such bugs appear in production only and do not appear in testing due to a different set-up. They are the fear of IT departments.

However, besides technical parameters, there are Application parameters. They impact the business logic. In a core-banking system, they define e.g. the limit for loans for which two credit officers have to approve the loan. Other parameters provide the files of the bank logo used for account statements. Parameters provide more flexibility since changing them is easier than changing code.

On the other side, such parameters have drawbacks. Test coverage can drop and they allow for bypassing the software change processes. Various software deployment tools install software in production only, if it is packaged and signed-off by testing. Neither developers (and certainly not users) can change the software behaviour without a sign-off from testing. However, this changes for GUI parameters. Power users might be able to change GUI parameters and, thereby, bypass the software change process (Figure 1).

An organizational solution is needed: First, restrict who has access to GUI parameters. Second, communicate that no change is allowed without testing. Third, make clear that sanctions for not following the rules are widely understood.

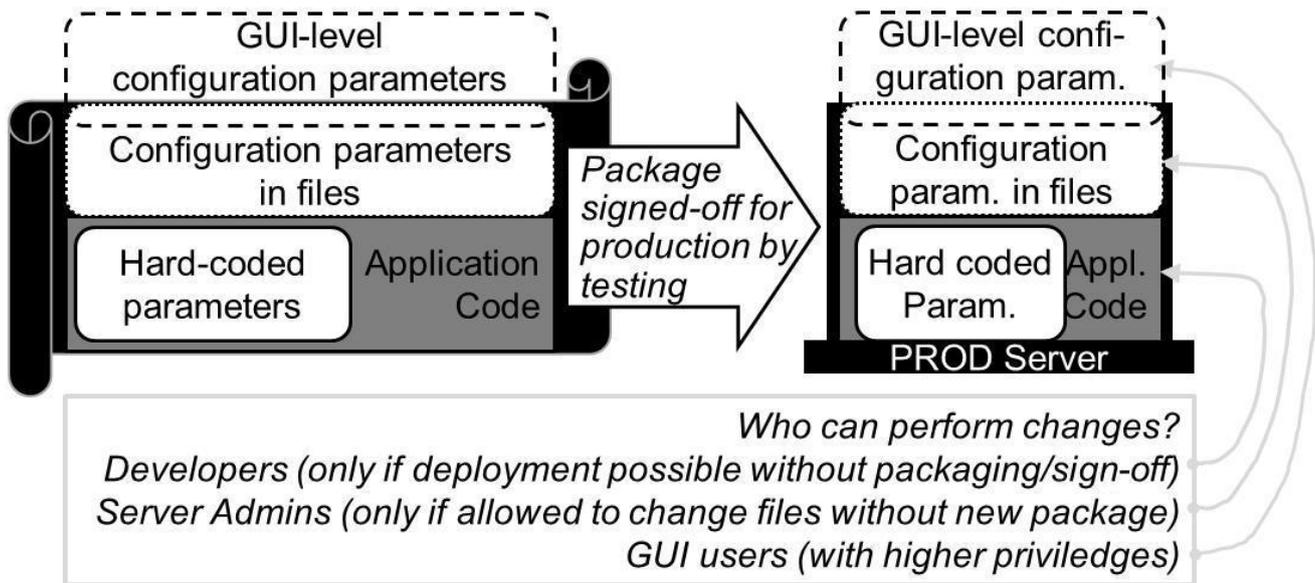


Figure 1: How configuration parameters undermine test and change processes

The second drawback is a drop in test coverage. The number of configuration options might explode due to the parameters. No test budget will grow at the same pace. We assume a system with five parameters: JVM memory settings, timeouts, maximum number of users, disk size and application server version. Each parameter can have one value out of four. The result is $4 \times 4 \times 4 \times 4 \times 4 = 1024$ configuration options. No IT department will pay for testing all options if the software is installed in three branches in Zürich, London, and Singapore only.

Thus, there is a risk that when changing parameters in production, the new configuration might not have been tested. The application usage can move out of the test scope (see Figure 2). It is not clear whether the software might crash or produce wrong results. To prevent this, changing parameters must trigger testing, even if there is no new package (Figure 3, Checkpoint A).

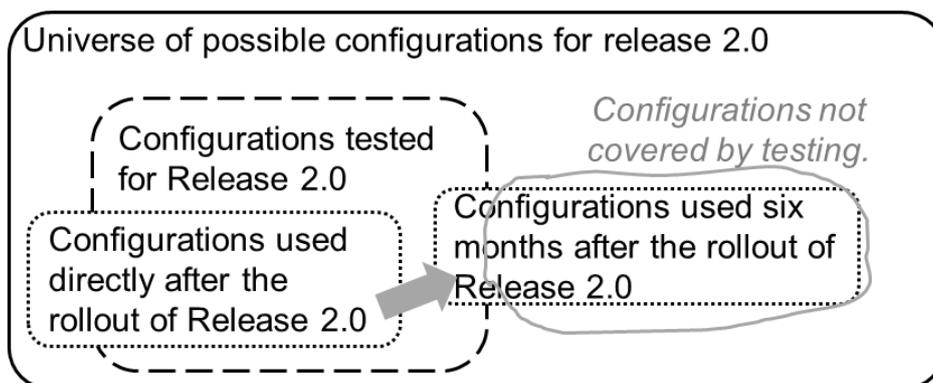


Figure 2: Test Coverage and Application Usage over time.

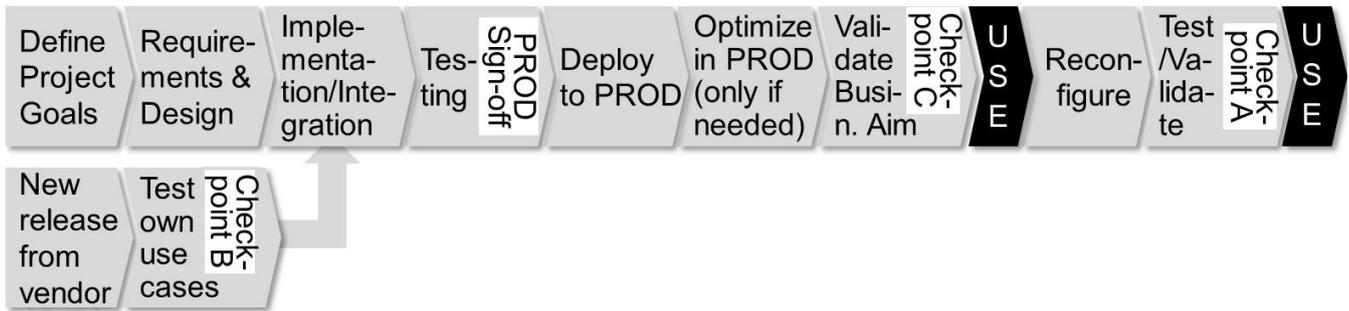


Figure 3: Unified software change process considering technical and business-focused quality assurance

Software Supply Channels – Stable as a House of Cards?

This section looks on the quality impact of 3rd party software components. Our example is a contract management solution of an insurance company. It enables insurance agents to print out contracts, which clients sign. It can scan contracts and store them in an archiving system. The solution incorporates three 3rd party software components: a reporting engine for rendering a PDF with the contract for printing it; a scanning solution with OCR; and a document archive (Figure 4).

All vendors have one dilemma in common. On the one hand, they need economies of scale. The software must meet the needs of many (potential) customers. On the other hand, software vendors make an implicit promise: the software works; it is (nearly) bug-free; you can start using it tomorrow. Obviously, the more configuration options software has, the less likely is that all options are tested in-depth and work as expected.

The dilemma of software vendors has implications for IT departments. First, the latter have to accept this reality. Vendors test a new release before rolling out software to their customers. Their test scope, however, is not guaranteed to match the exact usage scope of all customers. Second, IT departments must manage this quality risk. They could hope that there are no bugs or that those that are present are found in system integration testing. This is obviously late and risky. A better approach is to model test cases based on their own usage of the 3rd party software. The IT department tests based on them when the vendor rolls out a new release. This is a new quality gate (Figure 3, checkpoint B). Only if the new release works with the rest of the solution, is it incorporated in the customer’s IT landscape.

In the case of niche products, the most sustainable solution is to try to hand over the test cases to the vendor. Then, the vendor can add them to their regression test set.

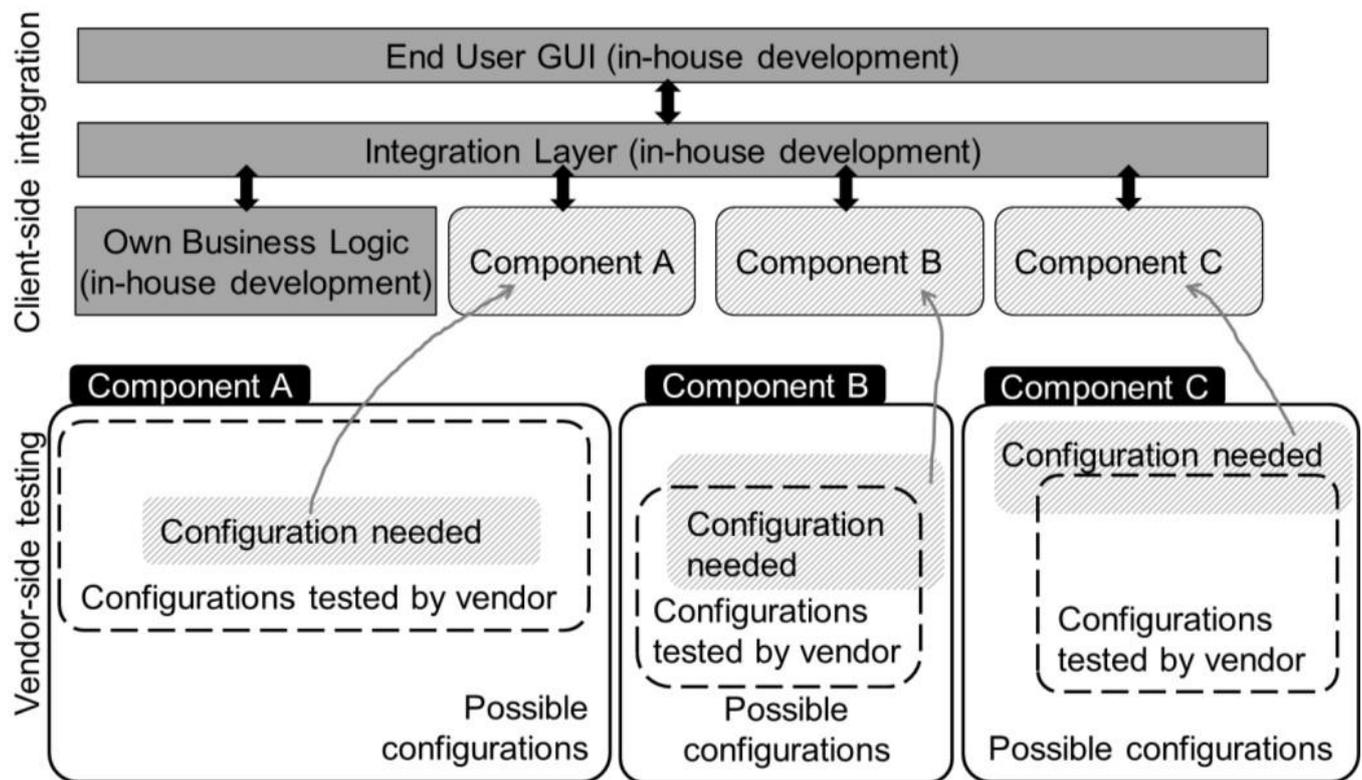


Figure 4: Quality risks in software supply chains

Business Focused Quality – or – Testing is neither Optimization nor Validation of Business Aims

The reason to start an IT project can be anything from a purely technical to a highly business-related aim. An example for a technical project is upgrading all Linux servers to a common patch level. The project succeeds if the technical goal is reached. Projects with a strong business focus differ. We use an investment fund as an example. The fund uses an automated trading system, which decides on a day-to-day basis when to buy and sell which stocks. Now a trader has an idea: If we introduce a new trading rule “sell stocks which gained 10% or more in a week”, the fund profit should rise by 1% per year.

Three dimensions describe the success of the project:

(1) Technical correctness: Is the business rule implemented as specified? Are stocks sold if they gain 10% or more in a week, but not if they raise only 5% or drop by 15%?

(2) Achievement of business aim: Does the new rule increase profits by 1%?

(3) Optimization question: Is “10% gain within a week” the best configuration? Could the profit be increased by changing the rule to “sell stock if it gained 8% within three days”?

Testers sign-off the technical correctness of the software after testing (1). They do not and cannot check whether and how efficient software helps achieving business aims (2 and 3). For the latter, often the software has to be in production for days or weeks to see the effects. This requires rethinking root-causes for rolling back to an old release or deploying emergency fixes to production.

IT problems (buggy software, which get into production, aka a testing disaster) are only one root-cause for emergency fixes. Wrong assumptions by the business are a second option (e.g. the trading rule was not a good idea). The software change process has to reflect them as well. Besides a sign off from testers, a sign-off against business aims and

optimization goals is needed. This requires adding a new checkpoint to the software change process *after* the deployment to production (Figure 4, checkpoint C).

Conclusion

Software quality is more than testing internally developed code. Configuration

parameters, 3rd party software components and business (optimization) aims pose new challenges for software testing and change. To overcome this, this article elaborated how to enhance software testing and change management processes to uniformly assure technical and business-focused software quality.

[1] M. Loukides: What is DevOps? <http://radar.oreilly.com/2012/06/what-is-devops.html>, last retrieved July 26th, 2015



Klaus Haller is an IT consultant with Swisscom Enterprise Customers in Zurich. Since 2005, he has worked mainly in the Swiss banking sector. His areas of expertise are testing and test centre organization, test data management, compliance testing and IT risk. He publishes frequently in magazines and speaks on conferences. More about him on his webpage <http://www.klaushaller.net>



Rudolf Grötz is an ISTQB Certified Full Advance Tester. He heads the QA Division of Jumio Inc. in Vienna. Since he got in contact with agile ideas in 2008, he is convinced that “agile” is like a poison. It works in the right doses, but too much is deadly. Thus, he continuously aims to find the right does to make requirements engineering and test automation a success. He publishes frequently in magazines and speaks on conferences. More about him on XING: https://www.xing.com/profile/Rudolf_Groetz

The opinions expressed in this article are the authors' own and do not necessarily represent the views of the companies they are working for.

Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on phill.isles@bcs.org

15 minutes with

Jim Thomas of Test and Verification Solutions Limited



JW5 Associates

Jennifer Wheeler of JW5 Associates spent 15 minutes with **Jim Thomas**, Director of Software Testing, Europe at Test and Verification Solutions Limited (TVS Ltd) talking about his career in testing so far.

How did you get into testing in the first place?

I came from a signal processing and hydrodynamics research background that involved quite a bit of maths and computing when I decided that I wanted to move more into software development and joined IPL, at the time a startup software consultancy in Bath. IPL's founders met at the European Space Agency on Darmstadt and had brought with them a highly disciplined approach to software development. Joining IPL opened my eyes to the difference between programming and software engineering. There was a structured approach to creating a highly modular software design where all developers coded and unit tested using project-specific test harnesses. We all undertook unit testing and got involved in integration and systems testing. That's was really my introduction to testing software properly, initially developing automated unit test scripts.

What do you think helped you develop your career the most?

I think I benefitted from being able to work on lots of different applications with diverse technologies - this really helped me develop my skill set. Projects at IPL were customer driven with different clients using different tools and products so I learnt how to build and test software from different perspectives. This helped me develop a logical problem solving approach which has been very valuable and transferable to any situation.

I feel I was fortunate to be at working in a boom time for the software industry. IPL was growing and I was given responsibility for leading projects and people quite early on.

IPL's Managing Director had a strong technical background. He was very understanding and supportive. All projects have their challenges as we know but as long as the team were doing their jobs properly he was happy. There was only ever a problem if it was found that projects had had corners cut in order to deliver, for example cutting back on testing – if that was found to be the case then those responsible didn't tend to stay with the company for long! The pressure was on quality and not delivery which instilled a great approach and attitude early in my career.

I enjoyed working with customers so gravitated to a responsible position. I also like helping people develop their skills so moving to a leadership role came naturally.

What are your feelings on qualifications in testing?

Overall I feel they are a good thing. Gaining formal accreditation demonstrates you have a level of understanding of testing. However qualifications alone doesn't make someone a good test engineer, and equally there are good test engineers without qualifications.

Qualifications can however offer advantages such as strengthening CVs and demonstrating a commitment to testing.

You have been in testing for 32 years, what advice would you offer people at the beginning of their testing careers who are keen to develop their careers along this path?

Learn development/programming skills – this will be invaluable in today's testing arena.

Broaden your experience – do as much testing on different systems with different technologies as possible.

It is absolutely key to seek out feedback from more experienced people regularly – be proactive in asking for help and learn from others, if possible seek out a mentor.

Get active in the testing community, whether through the BCS SIGiST membership or other testing groups - get involved. Staying on top of industry developments is important and exchanging ideas and information with others really helps. With this in mind TVS recently established a software testing conference for the South West called Intelligent Testing as a setting for test professionals in the area to meet.

Keeping up to date with new approaches to testing, tools, processes *etc.* can be a full time job in itself. How do you stay in touch with what is happening in the market?

It is a challenge with so much out there these days.

- I go to as many conferences and talks as possible, in fact TVS hosts quite a few conferences – inviting leading people from the industry and tool vendor companies to talk about challenges, innovation and new approaches to testing.
- I monitor web content like professional tester, computer weekly *etc.*
- LinkedIn – I am a member of a quite a few test groups.
- Staying social – networking with other people in the testing world is a great way to broaden your knowledge without the marketing spin.
- I also keep in touch with different tool vendors to monitor what new offerings and technologies are coming to the marketplace.

Can you tell me a bit about TVS?

TVS is a specialist testing and verification company based in Bristol and with test centres in India. We provide services for both software and hardware (ASIC, FPGA, *etc.*) testing which is quite unusual.

Our services are flexible and can be tailored to meet clients' needs from a purely onshore solution to offshore testing which is managed and delivered by our team here in the UK. We think the combination of onshore expertise with an integrated offshore capability is an attractive proposition for many organisations that want to outsource their testing or need access to expert test resources they can rely on.

We focus on automation, helping companies improve quality, save time and reduce costs. We aim to deliver applications and systems that are not just reliable but also safe and secure. Many systems have safety requirements and we have the verification and testing expertise to

help customers with that, and increasingly there is a security angle that we help customers address in build and test.

How has testing changed over the last 5 years?

There has been a real increase in test automation and unit testing due to the adoption of an Agile approach by organisations. That's probably been the biggest change I've seen. I started my software career automating unit tests and now that, and the use of application-level test automation tools, are really mainstream.

We are now seeing test automation move to the next level with service virtualisation tools. With technology landscapes becoming more interconnected it is important that tools and test environments meet this challenge.

How do you approach supplying testing services to companies at the moment?

Our business comes from people getting to know us through our speaking at conferences, networking events and hearing about our work. For us it is all about building relationships and building long term partnerships with our customers.

What projects are you working on now?

We have a number safety related test and verification projects, primarily in the automotive, medical and avionic industries. With automotive systems and medical devices becoming increasingly reliant on software these two industry domains in particular are growth areas for more formalised verification.

We were recently awarded a place on the government's G-Cloud framework, for both general testing services and security/pen testing. We are now involved in a number of public sector projects using more mainstream test automation technologies such as Selenium.

What do you enjoy most about heading up a testing practice?

Helping customers scope solutions to problems they are experiencing.

Then overseeing projects and delivering solutions which meet or exceed expectations.

And finally helping people develop their skills and ability, that's what I really enjoy.



For more information on TVS and their services please contact Jim at jim.t@testandverification.com

JW5 Associates

For help and assistance in any aspect of Testing career planning or recruitment please contact Jennifer Wheeler at Jennifer@JW5.co.uk or call on 07733 121897

bcS
The Chartered Institute for IT

Personal Development Plan

Your portable record of professional aspirations and activities

NEW

Desktop/Laptop

Tablet

Smartphone

Did you get your Personal Development Plan email with suggested potential CPD activities?

The BCS Personal Development Plan (PDP) uptake is going well, with over 1,000 users already actively recording their CPD Development Goals, Activities and preferences. It's not just about recording details though, as there is a Resources section that shows live feeds of potential CPD activities, and a tailored email is sent every 2 months with details of the latest videos, articles, blogs, books and research in your specified field of interest. If you haven't registered yet, you can see the content from the latest PDP bulletin for topics relating to solution development and implementation here <http://www.bcs.org/content/ConWebDoc/50854> or by going to the CPD Portal at: <http://www.bcs.org/pdp/> and selecting the "Give me ideas" link.

The BCS Personal Development Plan is free to use; BCS members can use their Member Secure Area login and password to access it at <https://pdp.bcs.org/>, and non-members can use most of the facilities (using the same link) and registering to create their own user name and password. You can use it on a PC / laptop or compatible tablet PC or smartphone.

Is It Time for an ISTQB Software Localisation Testing Certificate?

Mark Rice

Abstract

The International Software Testing Qualifications Board (ISTQB) is a non-profit organisation geared towards providing a globally standardised set of qualifications for software testers. It is the most successful scheme of its kind, having issued over 300,000 certificates worldwide. Yet, this paper argues that existing ISTQB syllabi do not adequately discuss the important concept of software localisation or the localisation testing processes which are required to ensure a high quality international software product. It is contended that not only should software localisation be included in the ISTQB Foundation syllabus, but it is also deserving of an extension certificate of its own. Beginning by exploring the current state of education and certification in the software development and software localisation arenas, this paper argues that the growing recognition of the importance of software localisation as a sub-discipline of software development needs to be extended to other software development sub-disciplines, namely software testing. The existing ISTQB syllabi and certificate structure are examined, and, using video games as a motif, key localisation concepts which, it is suggested, should be included in future ISTQB literature are set out. This is followed by the presentation of a modified ISTQB syllabi/certificate structure which overlays potential localisation ISTQB certificates on top of the existing structure. Finally, a corollary of the localisation exploration is proposed: that video games are as equally deserving of ISTQB recognition as software localisation.

Introduction

Numerous training and certification schemes exist in the fields of software and video game development; schemes which continue to mature and grow in number every year. At the university level for example, virtually every major UK university has software and video game development courses or modules at bachelor's or master's levels¹. In terms of industry qualifications, many schemes, applicable to software development in particular, but not necessarily *exclusive* to software or gaming, also exist, including Certified ScrumMaster², PRINCE2³, ITIL⁴ and Six Sigma⁵.

Ignoring for a moment any incidental localisation content within said software and gaming education, an analogue of this situation can be argued to exist with localisation in a software

¹ <http://www.prospects.ac.uk>, 18.02.2015.

² <https://www.scrumalliance.org>, 18.02.2015.

³ <http://www.apmg-international.com/en/qualifications/prince2/prince2.aspx>, 18.02.2015.

⁴ <http://www.apmg-international.com/en/qualifications/itil/itil.aspx>, 18.02.2015.

⁵ <http://www.sixsigmaonline.org/index.html>, 18.02.2015.

and gaming capacity, though conceivably to a less-developed degree. Bernal-Merino⁶ principally discusses the training opportunities available in the video game localisation milieu, though the majority of his exploration could reasonably be applied to the wider arena of *software* localisation too. In terms of non-university education for instance, Bernal-Merino discusses⁷ webinars, online resources, summer schools, EU-funded projects and The Institute of Localisation Professionals (TILP)⁸, which offers the Certified Localisation Professional (CLP) certificate⁹.

Yet Bernal-Merino raises concerns about localisation education in a video game capacity, many of which are, again, also relevant to software localisation and indeed software and video game development. For example, he argues that while the video game industry is fast-moving, professional practice and formal education are both “unavoidably slow”¹⁰ in adapting to change. This is compounded by the reality that video game localisation requires additional skills over *generic* localisation, including video game lore knowledge, technical (IT) knowledge and familiarity with the wide-ranging narrative styles of video game content¹¹. Additionally, Bernal-Merino contends that the “varied array of entertainment software products, together with the apparent lack of standards and the different requirements imposed by the various gaming platforms, make the translation of video games a difficult area to systemise for training purposes”¹². Crucially, he recognises that some education is unregulated or self-regulated, with certificates of attendance often replacing assessment and certificates of proficiency. In some cases, well-respected companies lend unwarranted kudos to certifying bodies; the prestige of the consumer, rather than the skills of the provider, is used as a yardstick. The university-level courses also suffer from these problems, though these are assuaged somewhat by the compulsory peer-reviewed validation cycles and external marking by tutors, ensuring a minimum guaranteed level of quality¹³.

Nevertheless, training and particularly certification has its benefits. Despite some quality issues, it could be argued that, on the whole, certification in the software and video game, and software and video game *localisation*, industries promotes industry-wide adoption of tried-and-tested methods, that is to say, *standardisation* and *best (or ‘good’) practice*. It also provides some assurance to employers that employees with a given certificate possess the desired level of knowledge of a subject and are able to perform at an acceptable level of proficiency.

At some level, certification is also demonstrative of *recognition* and *legitimacy*, insofar as the subject matter, be it software, video games, localisation or otherwise, is of enough value to society, and has enough proponents, that it requires – and has organically grown to be worthy of – certification. An extension to this argument is that legitimacy also extends to *sub-disciplines*, ad infinitum, of already-accepted concepts. For instance, software development is an umbrella term, containing a number of sub-disciplines, each of which has a differing degree of acceptance, development and even *respect* within the collective umbrella society.

As software and video game *localisation* is being recognised as a fundamental part of software *development*, it is now time for this recognition to trickle down further into other sub-disciplines

⁶ M. A. Bernal-Merino, *Translation and Localisation in Video Games: Making Entertainment Software Global*, (New York, USA: Routledge, 2015), pp. 223-247.

⁷ Ibid., pp. 226-228.

⁸ <http://www.tilponline.net>, 18.02.2015.

⁹ <http://www.tilponline.net/page-984459>, 18.02.2015.

¹⁰ M. A. Bernal-Merino, 2015, op. cit., p. 223.

¹¹ Ibid., p. 228.

¹² Loc. cit.

¹³ For a further overview of Bernal-Merino’s concerns, see: M. A. Bernal-Merino, ‘Training Translators for the Video Game Industry’, in J. D. Cintas (Ed.), *The Didactics of Audiovisual Translation*, (Amsterdam, Holland: John Benjamins Publishing Co., 2008), pp. 141-156.

of software development. Bearing this in mind, let us now consider the sub-discipline of *software testing* in an education and certification context.

The ISTQB

Notwithstanding incidental software testing theory within university-level software development degrees, it is fair to argue that the provision of software testing education and certification is significantly more developed at the occupational/industry level than at the university level. The ISTQB is the epitome of this argument. The ISTQB, founded in 2002, is a non-profit organisation geared towards providing a globally standardised set of qualifications for software testers. It is the most successful and recognised scheme of its kind, having issued over 300,000 certificates worldwide¹⁴. It would not be an exaggeration to say that ISTQB certification is the preferred software qualification of the majority of software testing companies.

The certificates are organised into three tiers: Foundation (CTFL), Advanced (CTAL) and Expert (CTEL). Candidates must achieve lower tier certificates before they are permitted to attempt more advanced certificates. Between the CTFL and CTAL tiers, an optional, *Intermediate*, certificate, organised by the Information Systems Examination Board/British Computer Society (ISEB/BCS), may be attempted.

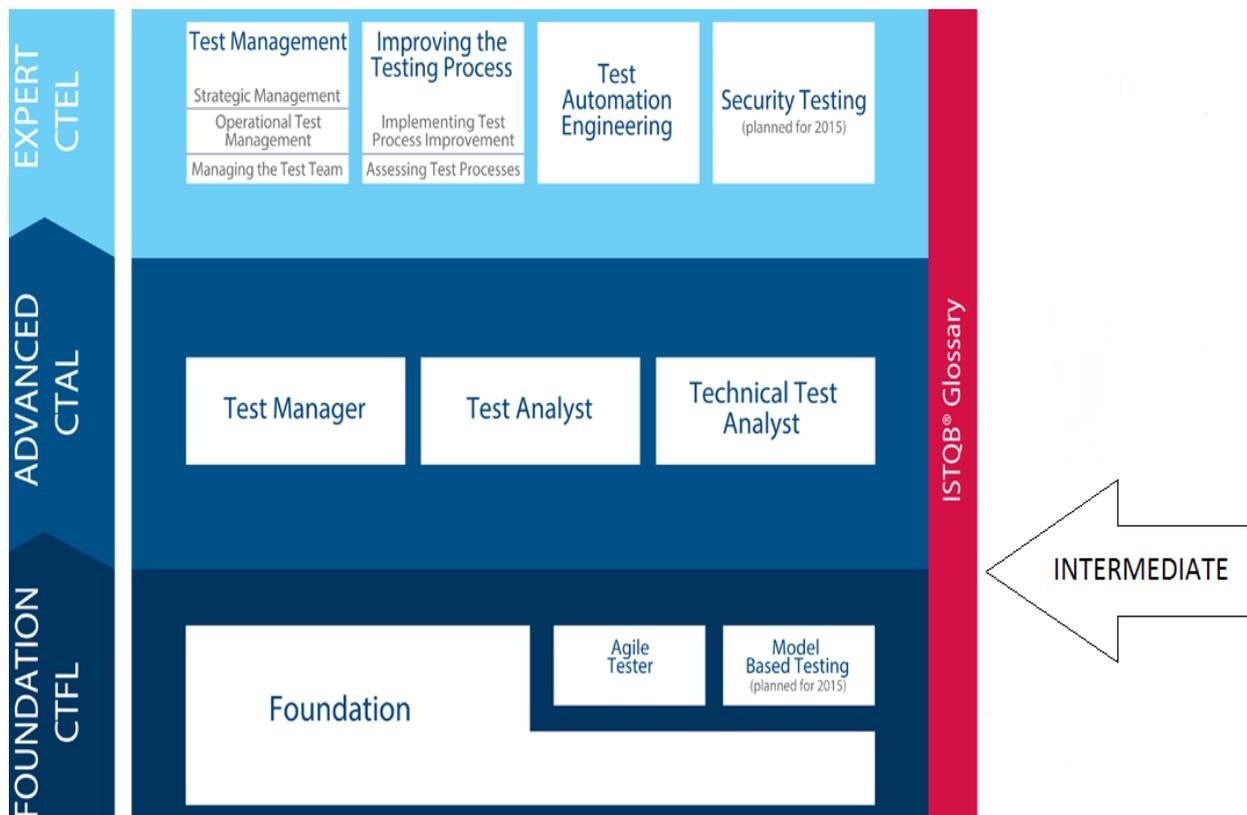


Figure 1: The current ISTQB certificate structure, adapted to show the ISEB/BCS Intermediate certificate. Source: <http://www.istqb.org>, 18.02.2015.

¹⁴ <http://www.istqb.org>, 18.02.2015, cf. CSTE at <http://www.softwarecertifications.org>, 11.03.2015.

From Figure 1, it is evident that the tiers run horizontally as well as vertically. Within the Foundation level for instance, in addition to the main Foundation certificate, there are the Agile Tester and Model Based Testing ‘extension’ certificates. The candidate must have achieved the main Foundation certificate in order to be eligible to attempt these. This is not true of the Advanced and Expert levels. The horizontal certificates in *these* levels are not interdependent and may be taken individually, with the only prerequisites being vertical ones.

Intuitively, as with software *development*, software testing is a broad church; as varied as the organisations which use software and IT Systems to achieve their goals. The syllabi reflect this by adopting a coverall approach. The guide to the Foundation certificate¹⁵, for instance, which is based heavily on the Foundation syllabus, discusses the Ariane 5 rocket failure on the same page as the UK online tax returns project¹⁶. In addition, the guide discusses concepts which apply to virtually all software quality assurance (QA) activities, such as the resources triangle¹⁷, testing principles¹⁸, the fundamental test process¹⁹, lifecycles²⁰, static testing²¹, test design techniques²², test management²³ and tool support²⁴. The guides and syllabi of the other ISTQB certificates follow a similar pattern, but to a deeper level of complexity and understanding. For instance, the Agile Tester extension certificate focusses on Agile software development models such as Scrum, Kanban and XP²⁵ while maintaining the ‘universal applicability’ approach.

However, software localisation *testing* is also an important software testing concept, yet this is not discussed in any great detail in any of the ISTQB syllabi. Indeed, it is ironic that the ISTQB glossary states²⁶ that it may be localised as required by country- or region-specific ISTQB sub-groups, but does not recognise that the *process* of software localisation must also be tested.

The statement that *software localisation testing is important* cannot be justified merely by pointing to the increased *legitimacy* value being placed on software and video game localisation, discussed at the start of this paper. The activities of software localisation need to be succinctly defined.

Defining Localisation

Software localisation though, like localisation in general, is also a broad and somewhat disputed term. Bernal-Merino²⁷ comprehensively discusses the meanings of localisation, its disputed elements and its related terms in a 2006 edition of the Journal of Specialised Translation (JoSTrans), but, for the limited purposes of this paper, it is argued that:

¹⁵ B. Hambling (Ed.), Software Testing: An ISTQB-ISEB Foundation Guide, Revised Second Edition, (Swindon, United Kingdom: BISL/BCS, 2011).

¹⁶ *Ibid.*, p. 10.

¹⁷ *Ibid.*, p. 13.

¹⁸ *Ibid.*, pp. 16-20.

¹⁹ *Ibid.*, pp. 20-25.

²⁰ *Ibid.*, pp. 34-56.

²¹ *Ibid.*, pp. 57-73.

²² *Ibid.*, pp. 74-128.

²³ *Ibid.*, pp. 129-166.

²⁴ *Ibid.*, pp. 167-214.

²⁵ R. Black et al., Certified Tester Foundation Level Extension Syllabus Agile Tester, Version 2014, (Brussels, Belgium: ISTQB, 2014), pp. 11-17.

²⁶ A. Beer et al., Glossary: Standard Glossary of Terms Used in Software Testing, Version 2.4, (Brussels, Belgium: ISTQB, 2014), p. 7.

²⁷ M. A. Bernal-Merino, ‘On the Translation of Video Games’ The Journal of Specialised Translation, Vol. N/A, No. 6, 2006, pp. 22-36.

Software localisation is more than the translation of software text and audio for international audiences; it also involves hardware considerations, packaging design, store/currency alteration, the tailoring of content to cultural sensitivities and transcreation: the act of maintaining a uniform message, mood, intent and emotional response for every language, country and region.

Thus, the effective *testing* of these factors is intuitively a crucial part of the overall software testing armoury, and is key to delivering a high quality international product. Video games are an obvious example of where software localisation, and thus software localisation *testing*, may occur. For instance, historical references, flags, maps, character names, character stereotypes and disputed borders are all inspected when a game is being tested. This is in addition to the alteration of button layouts, the tailoring of in-game stores and the checking of spelling, punctuation and grammar (SPG) of newly translated text. It is recognised that video games are not the only software products in need of localisation and localisation testing but they are, arguably, the most prominent and as such, video games are used as a motif throughout this paper.

In addition to contending that localisation is a vital software testing concept and thus worthy of discussion in – in the first instance – the ISTQB Foundation syllabus, the intent of this paper is also to demonstrate that software localisation testing should have an ISTQB extension certificate of its own. In order to facilitate these arguments further, the *key* concepts of software localisation, extracted from the bespoke definition discussed earlier, will now be summarised, albeit superficially. Then, the same key concepts *in a software testing context* will be explored. Examples from video games, old and new will be used to demonstrate the importance of these concepts, and the problems which may arise if localisation testing, and localisation, are not effective. Not only will this exploration again demonstrate the importance of software localisation testing, it is hoped it will also act as a guide for the ISTQB, if it chooses to include localisation content in its syllabi in the future.

Regarding the examples used in this paper, *Golden Age* (defined here as late 1970s to early 1990s) arcade video game developers in particular often did not have the budget or access to skilled localisation staff to perform dedicated localisation or localisation testing. Localisation was typically limited to translation – usually Japanese to English – and often performed in-house by the developers themselves. Obvious, and amusing, poorly localised content – affectionately known as ‘Engrish’ – was often the result. *Golden Age* titles thus demonstrate most clearly what, and how, things can go wrong during localisation, and what today’s localisation testers find. Yet, while such examples are not exclusive to *Golden Age* games by any means, today, mistakes are likely to be fewer and less severe prior to testing, as well as being much more likely to be caught and fixed by testers prior to the product reaching the customer. The heavy use of *Golden Age* examples is testament to today’s relatively high quality of software localisation and software localisation testing.

Key Localisation Concepts in an ISTQB Context

The Localisation Process. A detailed description of what constitutes software localisation and software localisation testing would need to be present in the Foundation syllabus, coupled with a working definition within the ISTQB glossary²⁸. Localisation in terms of *language*, *country* and *region* would need to be discussed. Many software items, especially video games, are localised at the regional level; represented by different stock keeping units (SKUs). For instance, a game may have a Latin American (LATAM) SKU, a European SKU and a North

²⁸ A. Beer et al., op. cit.

American SKU. Each regional SKU covers a number of countries, as well as the languages deemed appropriate to those countries, though some games will include all languages in every SKU. Thus, a video game will likely differ between regional SKUs in terms of language availability and substantial content. Content may subtly change within a single SKU too, based on country/language selection. It is more common however that, where a country has particular and/or wide-ranging demands which would compromise the content of a video game for other countries in the same region, then a *country-specific* SKU would be made. This is a nuanced area of localisation, and software on some platforms (e.g. mobile phones) may adhere less to this tradition. Yet, while it is intuitive to talk of localisation in terms of *language* – indeed, localisation *testing* is often organised in terms of the *languages* under test, while taking into account *countries* and the *SKUs* being tested – the typical level is *region*, of which *language* is but a constituent.

The positions of both localisation and localisation testing in the Software Development Lifecycle (SDLC) and the relationship between software localisation testing and other forms of software testing (e.g. functionality, performance or security) would also need to be discussed. A fundamental requirement here would be that the ISTQB syllabus discusses *where*, in the two most commonly used software development models – the basic V-model and the generic iterative/incremental model – localisation and localisation testing occur. Please see Figures 2 and 3 below.

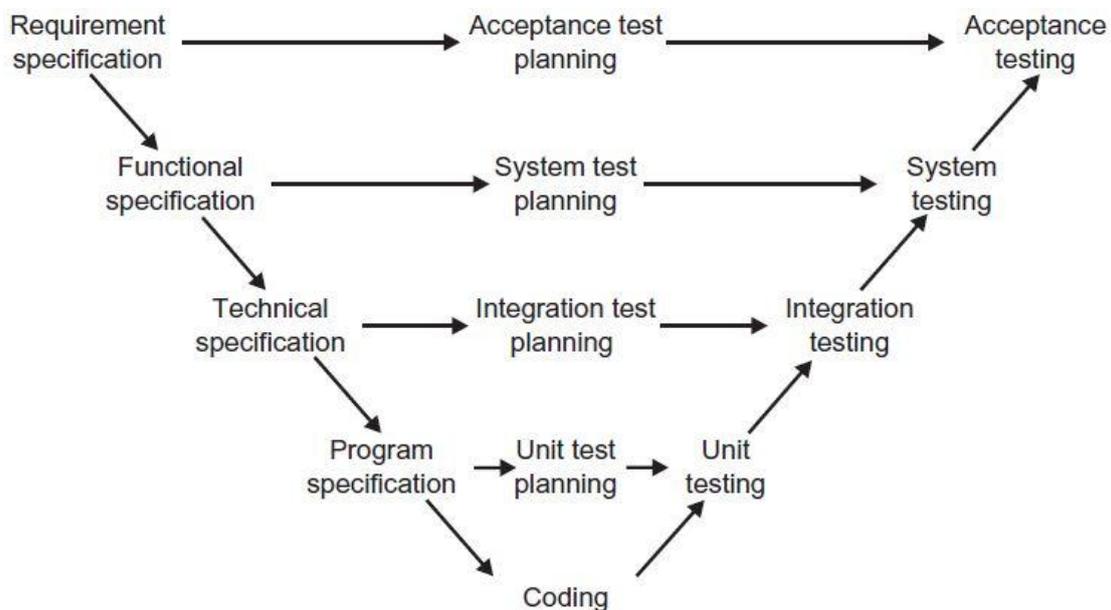


Figure 2: The basic V-model of software development and testing.

Source: B. Hambling (Ed.), *Software Testing: An ISTQB-ISEB Foundation Guide, Revised Second Edition*, (Swindon, United Kingdom: BISL/BCS, 2011), p. 38.

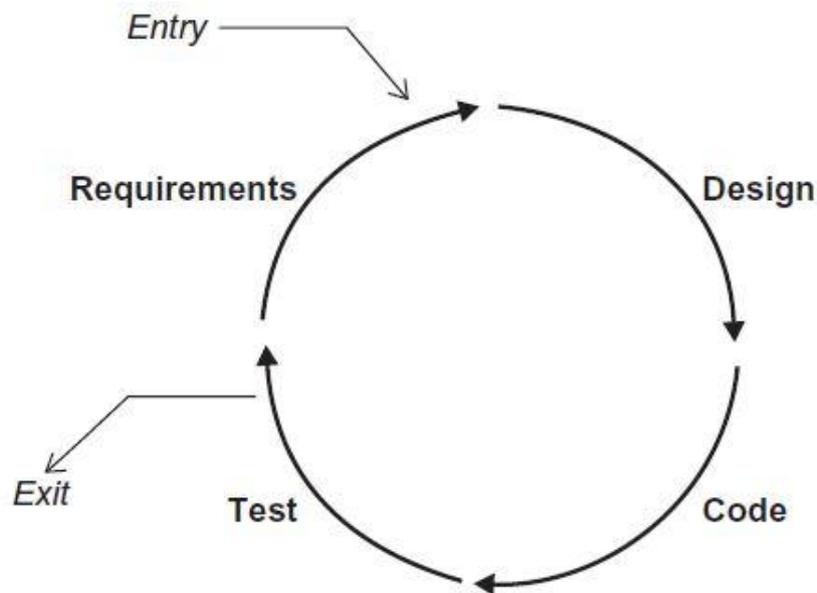


Figure 3: The basic iterative model of software development and testing.

Source: B. Hambling (Ed.), *Software Testing: An ISTQB-ISEB Foundation Guide, Revised Second Edition*, (Swindon, United Kingdom: BISL/BCS, 2011), p. 40.

Text and Audio Translation. These processes are the building blocks, and the most-recognised aspects, of software localisation, although the methods and tools used to achieve them are often complex. Translations may occur on an in-house, distributed, insourced or outsourced basis²⁹, or as a combination of these approaches. Typically, translations are conducted by third-party vendors who translate on a cost-per-word basis. These translations are then stored in a database and incorporated into software builds at regular intervals. Legal text and copyright screens are included in this activity; this content in particular must be correctly localised. The principal testing process, then, would be, first, examining the text and audio strings in the database for SPG errors; this would be *static*³⁰ in nature and could be termed a *review*. Following this, the localised content would be tested, *dynamically*³¹, using the program within which the content will ultimately be used. In the context of a video game, the localisation tester would perform actions such as playing the game, checking that all text and audio were correct, accurate and consistent and ensuring that they appeared at the desired points in the game. The tester would also check that text was not overlapping or ‘clipped’ in different screen formats (including mobile phone screens, if applicable). There would also need to be some agreement amongst the testers that the *meaning* of each string – and, to some extent, each graphic – was the same across languages, countries and regions; part of *transcreation*. Localisation testing of software packaging would not typically be executed by software localisation testers; product managers or their equivalent would perform this task. Let us look at some examples of what can happen when localisation efforts are not effectively tested. The first example, Aero Fighters 2³² (Figure 4) is a typical instance of ‘Engrish’: the transposition of ‘l’ and ‘r’ in Japanese to English translations.

²⁹ R. Black et al., *Certified Tester Advanced Level Syllabus Test Manager, Version 2012*, (Brussels, Belgium: ISTQB, 2012), p. 43.

³⁰ A. Beer et al., op. cit., p. 39.

³¹ Ibid., p. 19.

³² Aero-Fighters 2, Video System, 1994.



Figure 4: Frying over a jungle. Source: Aero-Fighters 2, Video System, 1994.

As stated earlier, localisation and localisation testing have improved dramatically since the Golden Age, yet mistakes do still slip through and affect the perception of a video game's quality, especially when such mistakes are broadcast on YouTube or other social media; hazards which were not present in the Golden Age. *Ape Escape 3*³³ for instance, originally a Japanese game, spells the world 'shield' incorrectly, and this mistake was broadcast on YouTube, along with criticism of Sony's localisation efforts³⁴.

Peripherals and Button Functionality. Some countries and regions, particularly Japan and Asia, use atypical controller button setups for common actions (e.g. performing jumps). The localisation tester would need to ensure that the buttons function as expected *and* that button icons and text hints convey the correct purpose of these buttons. This is also part of functional testing.

Cultural Sensitivity. This too is a functional testing task, but it could be argued that testers native to each language, country or region would be more informed of cultural hazards than their functional tester counterparts. In this process, within a gaming context, testers would be looking for references to, or incorrect usage of, flags, historical names, events, religions and disputed regions, in addition to graphics and words which could be considered insulting to some cultures. Some content may not be culturally sensitive per se, yet it may still be banned in some countries or regions owing to state censorship; this also has to be taken into account.

Ratings. Related to cultural sensitivity, it is important to note that different countries and regions use different ratings systems when classifying software (overwhelmingly *video games*). Europe, for example, uses Pan European Game Information (PEGI)³⁵, while Japan uses the Computer Entertainment Rating Organisation (CERO)³⁶. Again, functional and localisation testers are expected to flag content which may nudge a game into a higher rating than that

³³ *Ape Escape 3*, Sony Computer Entertainment, 2006.

³⁴ <https://www.youtube.com/watch?v=dH-cOoH5ZCg>, 22.02.2015.

³⁵ <http://www.pegi.info/en/index>, 20.02.2015.

³⁶ <http://www.cero.gr.jp/e/rating.html>, 20.02.2015.

which is claimed by its developers and producers, but localisation testers may have greater knowledge of their countries' and/or regions' rating expectations.

Localisation Tools. The ISTQB syllabi already include detailed analyses of tools used in a testing context³⁷ (colloquially known as *testware*³⁸). These analyses would need to be extended to include tools used *specifically* for software *localisation* testing, including text and audio databases³⁹, and *common* testware, such as bug reporting tools⁴⁰, applied in a localisation context.

Online Stores. Functional testing tests online store functionality, but it is important to remember that content (particularly text) is also localised in online store locations and thus would need to be tested in a localisation context. It is important also to test that the correct currency is used per country or region. This activity may be a 'store-side' responsibility and thus beyond the remit of software localisation testers who have been employed specifically to test only a piece of software's *non-store* localisation. However, localisation testing will occur one way or another.

Transcreation. Transcreation is sometimes used as a synonym of localisation. Yet, this paper prefers to consider it as a synecdoche, in that while transcreation refers to localisation, it is also used in the discrete sense of the degree to which the *overall* localisation process has resulted in a product that conveys the same brand, mood, message, intent and user emotional response in every language, country and region it has been localised to. There should be some mention in the ISTQB Foundation syllabus of the importance of looking at the success of localisation efforts in the round as well as per process, with an emphasis on 'emotional uniformity'. Localisation testers could, for instance, have periodic group walkthrough reviews of each version of a piece of software, to ensure the same message is being transmitted. It could be argued that a transcreation shortcoming played a part in a famous Golden Age urban myth. Street Fighter 2⁴¹ has the following scene:



Figure 5: 'Sheng Long'. Source: Street Fighter 2, Capcom, 1991.

³⁷ T. Muller et al., *Certified Tester Foundation Level Syllabus, Released Version 2011*, (Brussels, Belgium: ISTQB, 2011), pp. 57-64.

³⁸ Testware also consists of testing work products, such as test plans.

³⁹ The Localised Asset Management System (LAMS) is a commonly used tool by Sony Computer Entertainment Europe.

⁴⁰ DevTrack is one tool used in both a functional and localisation context. <http://techexcel.com/products/devtrack>, 22.02.2015.

⁴¹ Street Fighter 2, Capcom, 1991.

Shōryūken (昇龍拳), was the name of Ryu's special uppercut move (Ryu is the character on the right). The pinyin (i.e. the transcription of Mandarin pronunciations of Chinese characters to the Latin alphabet) of *Shōryū* (昇龍, *rising dragon*) is *shēnglóng*, and this word found its way into the English version of the game. The original Japanese quote should have been translated as 'If you cannot overcome the Rising Dragon Fist, you cannot win!'. This confusion spawned urban myths regarding the character of Sheng Long and of 'him' being a secret character in the game⁴².

The Ideal Software Localisation Tester. The ISTQB Foundation and Advanced syllabi discuss⁴³ the qualities of an ideal software tester. The software *localisation* tester must also possess these qualities. However, for localisation testers, emphasis should be placed on testers who are native to the language, country and/or region of the title they are testing, preferably multilingual and who possess excellent SPG skills. *Discretion* is also an important facet. Testers are sometimes required to shorten, delete, rephrase or add words on-the-fly with little guidance. In addition, some software (again, principally video games) purposely includes partially obscured text or foreign languages as content, which should therefore not be bugged, such as graffiti in Italian on a wall in the game *rain*⁴⁴, a Japanese game localised to English (Figure 6).

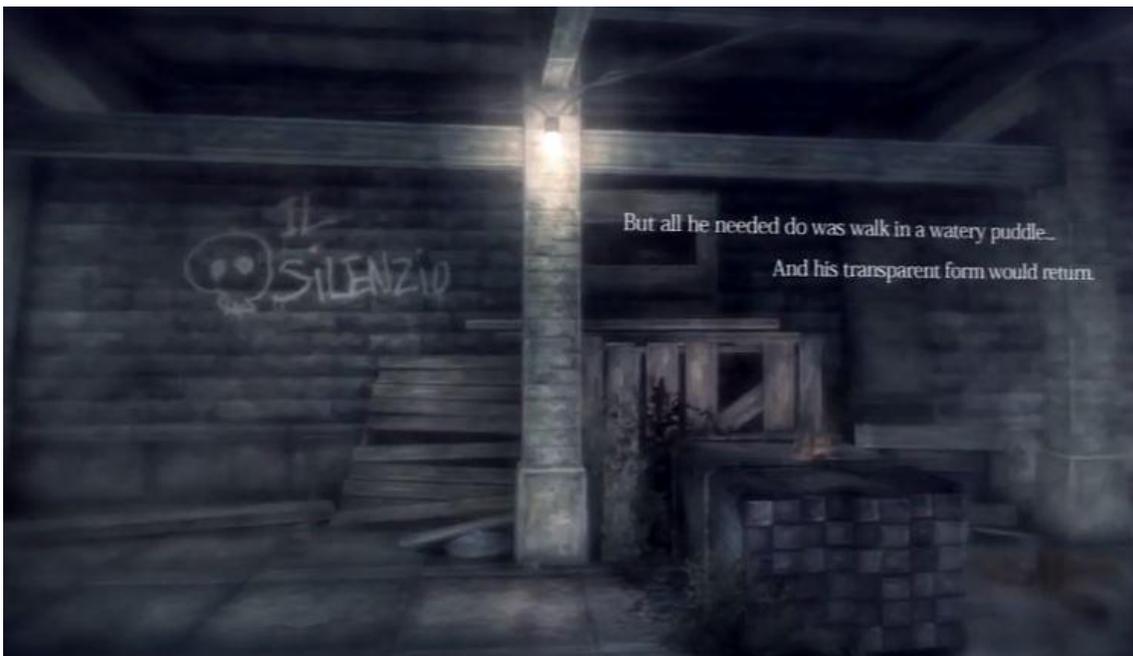


Figure 6: Italian graffiti on a wall in the game *rain*, a game localised from Japanese to English. Source: Rain, Sony Computer Entertainment, 2013.

Based on the author's experience, it is suggested that software localisation testing detects many localisation issues which would otherwise be seen by the customer. Localisation testing also compensates, to some extent, for initial poor or limited scope localisation, thus acting a 'safety net'. However, testers are not infallible; the better the initial localisation effort, the better the final result.

⁴² <http://www.capcom-fc.com/sf4/2008/04/41.html>, 23.02.2015, http://streetfighter.wikia.com/wiki/Sheng_Long, 19.03.2015.

⁴³ T. Muller et al., op. cit., pp. 18-19 and R. Black et al., 2012, op. cit., pp. 71-76.

⁴⁴ Rain, Sony Computer Entertainment, 2013.

The elements of software localisation testing which have been discussed are argued to be the *fundamental* facets which would need to be included in any localisation section of the ISTQB Foundation syllabus. The list is not exhaustive, and the reader may disagree with the inclusion/exclusion of some concepts. As with other concepts of the ISTQB, these elements would be discussed at a basic level in the Foundation syllabus but then explored in more detail in advanced/expert syllabi, where appropriate, and in a localisation extension syllabus.

This begs the question, ‘where should the extension certificate sit in relation to the other ISTQB qualifications?’ It is suggested here that there should be an extension certificate at the Foundation level, termed CTFL-L, which could only be achieved once the candidate had passed the standard CTFL certificate. Moreover, such is the scope of software localisation testing, that there could also be an Advanced (CTAL-L) certificate, indeed, perhaps even an Expert (CTEL-L) certificate, on the subject. Thus, Figure 1 would now look like this:

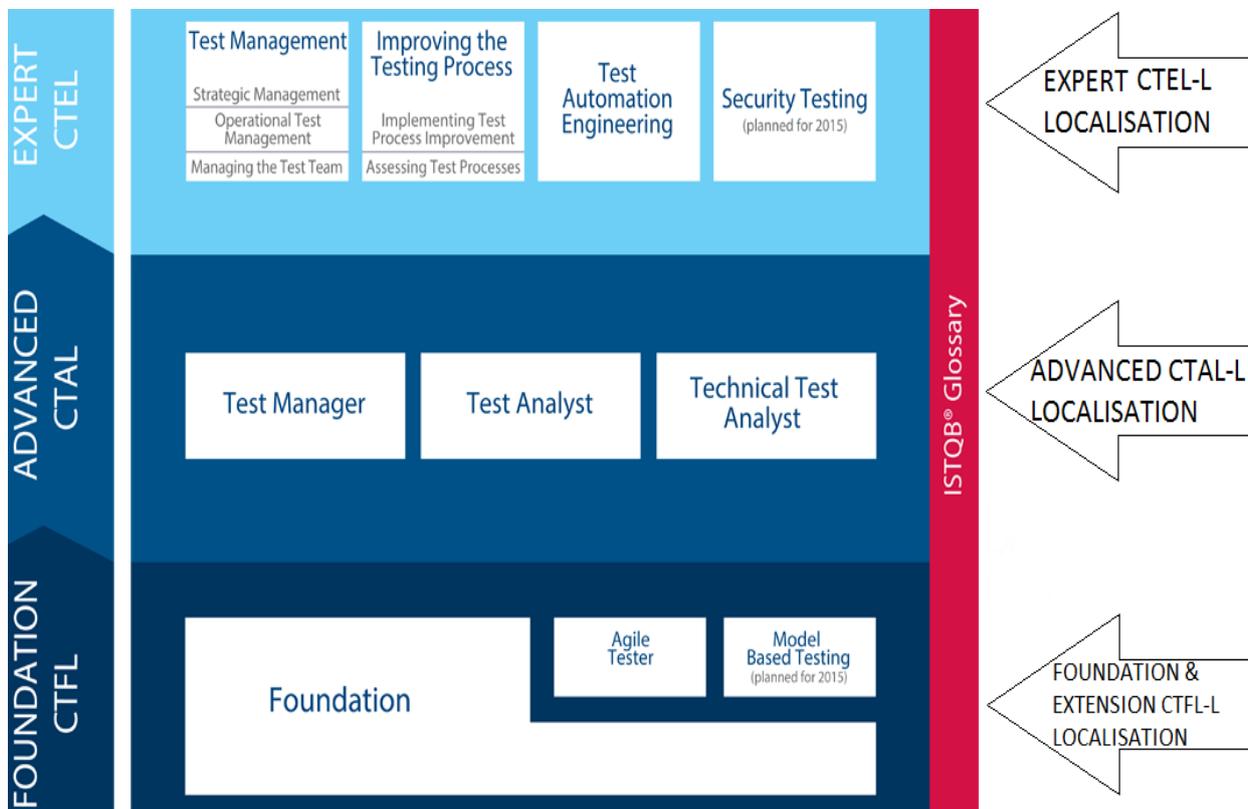


Figure 7: The ISTQB certificate structure with suggested localisation certificate locations.

A closing observation of this paper is this: this paper has focussed on software localisation in an ISTQB context, yet has relied heavily on *video game* localisation to demonstrate its points. So, despite the risk of conflating *software localisation* and *gaming software*, it is suggested that the equally absent *video game testing* concept deserves the same ISTQB inclusion in its own right as what, it is hoped, will be delivered for software localisation testing. A corollary of this argument is that *video game-specific* localisation testing factors should be discussed in both the software localisation testing and video game testing areas of the ISTQB syllabi. This video game testing debate is as equally urgent as that which is argued for software localisation testing.

In conclusion, it has not been the purpose of this paper to be prescriptive in its recommendations or to deal in absolutes of any kind; *exploration of possibilities* has been the priority. It is hoped that the purposely rudimentary nature of this paper will promote academic discourse and debate in both software localisation and ISTQB circles, with a view to encouraging syncretism between what are, after all, two *complementary* schools of study. If such debate is indeed embraced, then it is expected that what is currently an embryonic argument will be advanced and refined. Ultimately it is anticipated that once this paper's recommendations have been made more sophisticated, we will see the adoption of software localisation testing in an ISTQB context.

References

Aero-Fighters 2, Video System, 1994.

Ape Escape 3, Sony Computer Entertainment, 2006.

Beer, A. et al., Glossary: Standard Glossary of Terms Used in Software Testing, Version 2.4, (Brussels, Belgium: ISTQB, 2014), p. 7.

Bernal-Merino, M. A., 'On the Translation of Video Games' The Journal of Specialised Translation, Vol. N/A, No. 6, 2006, pp. 22-36.

Bernal-Merino, M. A., 'Training Translators for the Video Game Industry', in J. D. Cintas (Ed.), The Didactics of Audiovisual Translation, (Amsterdam, Holland: John Benjamins Publishing Co., 2008), pp. 141-156.

Bernal-Merino, M. A., Translation and Localisation in Video Games: Making Entertainment Software Global, (New York, USA: Routledge, 2015), pp. 223-247.

Black, R. et al., Certified Tester Advanced Level Syllabus Test Manager, Version 2012, (Brussels, Belgium: ISTQB, 2012), p. 43.

Black, R. et al., Certified Tester Foundation Level Extension Syllabus Agile Tester, Version 2014, (Brussels, Belgium: ISTQB, 2014), pp. 11-17.

Hambling, B. (Ed.), Software Testing: An ISTQB-ISEB Foundation Guide, Revised Second Edition, (Swindon, United Kingdom: BSL/BCS, 2011).

http://streetfighter.wikia.com/wiki/Sheng_Long, 19.03.2015.

<https://www.scrumalliance.org>, 18.02.2015.

<https://www.youtube.com/watch?v=dH-cOoH5ZCg>, 22.02.2015.

<http://techexcel.com/products/devtrack>, 22.02.2015.

<http://www.apmg-international.com/en/qualifications/itil/itil.aspx>, 18.02.2015.

<http://www.apmg-international.com/en/qualifications/prince2/prince2.aspx>, 18.02.2015.

<http://www.capcom-fc.com/sf4/2008/04/41.html>, 23.02.2015.

<http://www.cero.gr.jp/e/rating.html>, 20.02.2015.

<http://www.istqb.org>, 18.02.2015.

<http://www.pegi.info/en/index>, 20.02.2015.

<http://www.prospects.ac.uk>, 18.02.2015.

<http://www.sixsigmaonline.org/index.html>, 18.02.2015.

<http://www.softwarecertifications.org>, 11.03.2015.

<http://www.tilponline.net>, 18.02.2015.

<http://www.tilponline.net/page-984459>, 18.02.2015.

Muller, T. et al., Certified Tester Foundation Level Syllabus, Released Version 2011, (Brussels, Belgium: ISTQB, 2011), pp. 57-64.

Rain, Sony Computer Entertainment, 2013.

Street Fighter 2, Capcom, 1991.

Mark Rice is a software tester and project administrator for Sony Computer Entertainment Europe (SCEE). He also volunteers as a glossary reviewer for the ISTQB.

Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on phil.isles@bcs.org

SIGiST future dates for 2015

15 September 2015 (see Agenda in this edition of The Tester, page 2)

2 December 2015 - planned speakers include:
Mark Fewster, Graham Thomas, David Oxley

2016 SIGiST dates for your diary

15 March 2016
9 June 2016
TBC September 2016
TBC December 2016

Specialist Group in Software Testing

THE TESTER

SIGiST Winter Conference Wednesday 2nd December 2015

55

IN THIS ISSUE

Conference

Agenda2

Venue3

Abstracts.....4

Articles

Webdriver
integration with
JMeter for
simulating browser
based virtual users
.....12

Why I changed my
mind about UX
Certification....16

Regular Features

Fifteen minutes
with19

Testing Events
Listing22

From the Editor

Welcome to the last edition of The Tester for 2015.

Our programme secretary has done a great job again organising a thrilling programme for the SIGiST on Wednesday 2nd December. Subjects covered include Test Automation and Tester skills - including skills for Test Management and Programme Test Management. Opening with Mark Fewster, and closing with Graham Thomas - don't miss both these keynotes. The workshop in December is back by popular demand, on "Defect Measurement and Analysis", presented by the UK Software Metrics Association. Places are limited so sign up now!

Check out the articles in this edition of The Tester. One on Webdriver integration with JMeter, and one on User Experience (UX) certification.

We are always looking for speakers / workshops for the conference, and articles for The Tester. If you want to speak check out the SIG website: <http://www.bcs.org/category/10880> or contact me if you want to become a published author.

Phill Isles
The Tester Editor
phill.isles@bcs.org

Conference Booking Instructions

To register online, please use the link below, or scan the QR code with your smart device. Please note the BCS booking system accepts multiple and third party bookings.

<https://events.bcs.org/book/1407/>



Join our Linked-In Group:

<http://www.linkedin.com/groups?mostPopular=&gid=3466623>



Follow us @SIGiST



Conference Agenda

BCS SIGiST – Winter 2015 Conference – Wednesday 2 December 2015 BCS 1st Floor, Davidson Building 5 Southampton Street London WC2E 7HA		
Time	Session	
09:25	<i>Welcome – Stuart Reid, Chair, SIGiST</i>	
09:30	Keynote <i>Mark Fewster, Grove Consultants</i> “How Healthy is Your Test Automation?”	
10:30	<i>Networking Session – Jen Wheeler, Networking Secretary, SIGiST</i>	
10:45	Coffee , Tea & Refreshments	
Morning	Presentations	Workshop
11:15	<i>David Oxley, Intel Security</i> “Top Trumps – My skill is better than yours!”	<i>Kristina Masuwa-Morgan</i> UKSMA “Defect Measurement and Analysis” Part 1 11:15 to 12:50
12:00	Changeover	
12:05	<i>Chris Comey</i> “Key attributes and responsibilities of a Test Manager”	
12:50	Lunch	
Afternoon	Presentations	Workshop
13:50	Quick Talk - <i>Damon Rands, Wolfberry</i> “Digital foot printing - Building an attack”	<i>Kristina Masuwa-Morgan</i> UKSMA “Defect Measurement and Analysis” Part 2 13:50 to 15:30
14:15	Changeover	
14:20	Quick Talk - <i>Mark Rice</i> “Software Localisation and Software Localisation Testing: An Overview and Case Study”	
14:40	Changeover	
14:45	<i>Prakash Iijra, HCL</i> “Digitisation of a test organisation”	
15:30	Coffee , Tea & Refreshments	
16:00	Keynote <i>Graham Thomas</i> “Becoming a programme test manager”	
17:00	<i>Close - Stuart Reid, Chair, SIGiST</i>	

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops will have limited places.

SIGiST Conference Venue

For the last conference of 2015, the SIGiST returns to the BCS London office. Travel details and location below.

London office guide

How to get to the BCS London office

First Floor
The Davidson Building
5 Southampton Street
London WC2E 7HA
Telephone 01793 417666

These area and inset maps have been simplified in the interests of ease of understanding. Not all roads are shown. The inset map below is more accurate.

Access by car is very difficult due to the local one-way system. There are no car parking facilities at BCS London. The nearest car park is located on Drury Lane, Parker Street, Parker Mews, London, WC2B 5NT.

The rear door in Exeter Street is to be used for deliveries only and is normally locked.

The main entrance is fully accessible to wheelchair users and should be used by all staff and visitors.

On arrival, report to the Davidson Building Reception who will direct you to the first floor.

Travel tips from major London stations

Charing Cross – 6 minutes walk

Waterloo – 12 minutes walk across Waterloo Bridge, or buses 139 or 176 to Stop **C**

London Bridge – onward rail link to Charing Cross

Kings Cross or St Pancras – Piccadilly Line to Covent Garden tube, or bus 91 to Stop **B**

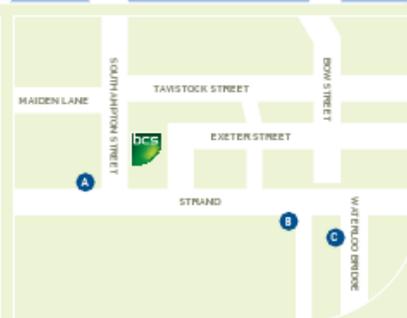
Euston – West End Branch of Northern Line to Charing Cross, or bus 91 to Stop **B**

Victoria (rail and coach stations) – Circle Line to Embankment, but the most direct journey is via bus 11 to Stop **A**

Paddington – Circle Line to Embankment or Temple, Bakerloo Line to Charing Cross or buses 15 or 23 to Stop **A**

Liverpool St – Circle Line to Embankment or Temple, or buses 11 or 23 westbound

Fenchurch St – Walk to Tower Hill, then District or Circle to Embankment



Presentation Abstracts and Speaker Biographies

Opening Keynote

**Mark Fewster,
Grove Consultants**

“How Healthy is Your Test Automation?”

Are you benefiting from test automation? Are you sure this will continue? Do you worry that your automated testing will falter and fail? Find out what the signs are of test automation that is destined to go wrong. Learn to distinguish between automation that is healthy and automation that has problems. If you already know your test automation is not in tip top form, discover some of the most common reasons for poor automation health and pick up ideas for what can be done to combat them.

In this presentation, Mark will offer a fitness regime for test automation, identifying some key ideas to help restore ailing test automation to good health and to keep it there well into the future. There will also be some advice for how to bring test automation back from the undead!

Three key points:

- Assess the health of your test automation.
- Identify good and bad habits.
- Develop a fitness regime to keep your test automation healthy.

Mark has over 30 years of industrial experience in software testing ranging from test management to test techniques and test automation. In the last two decades Mark has provided consultancy and training in software testing, published papers and co-authored two books with Dorothy Graham, "Software Test Automation" and "Experiences of Test Automation". He has spoken at numerous national and international conferences and seminars, and has won the Mercury BTO Innovation in Quality Award.

Mark has served on the committee of the British Computer Society's Specialist Interest Group in Software Testing (BCS SIGiST) and on the Information Systems Examination Board (ISEB). He is currently helping ISTQB define specialist certification for test automation.

Closing Keynote

Graham Thomas, Independent Consultant Practitioner

“Becoming a Program Test Manager”

After working for a while as a successful test manager the next forward step to take is into program test management. Many think a Program Test Manager is just some super test manager, or in a lot of cases, a ‘shouty’ test manager. In fact it isn't. You are transitioning into an oversight role, where others do the testing, and you are setting the direction, giving guidance, and having oversight.

This is quite a step up and suddenly requires a set of skills that successful test management does not develop. The scale has changed, you may be looking after multiple projects and multiple teams. You are now operating at the organisational level, working with other members of the program management team.

In this session we will look at a range of newly required skills; Leadership, Accountability and Responsibility, Oversight and Awareness, Stakeholder Management, Communication, Influencing and Negotiation. We will work through some useful models so that you can take away a kitbag of tools and techniques to use back in the office. We will also look at how to stay relevant to the testing operation, and retain value-add for your role whilst now working at the organisational level, and delivering through others.

And even if you aren't working as a Program Test Manager yet, the skills and techniques we look at in this session will be invaluable today, to start using, developing and refining.

Three key points:

- Tools and techniques for successful program test management.
- Useful models for influencing and negotiating.
- How to add value and stay relevant whilst performing a ‘delivery through others’ role.

Graham first learnt to program at college in 1978, worked as a developer through the 1980s, until in 1992 when he took his first steps in software testing. He has diverse IT and testing experience from end users and consultancies, in the public sector, retail, finance, banking, insurance and treasury. He now works as a program test manager or implementing testing change.

Track Session

Dave Oxley, Intel

“Top Trumps – My skill is better than yours!”

How many times have you heard something like ‘I’m better than you at that so leave the task for when I’m free’, or ‘I don’t know that as well as you so I’ll leave it for you’? These comments result in broken engineering teams and reduce the velocity at which the team work. So what do you do to overcome these problems as a tester?

Three key points:

- Give it a go – what’s the worst that can happen?
- Share knowledge and skills as much as possible.
- Step on toes if that will help your team.

Dave is a Quality & Security Champion for Intel Security. It’s a wonderful job of talking to people around the world on a regular basis, occasionally being forced to get on a plane and meet face to face. The focus is on ensuring each release is of a higher quality than the previous one through improving engineering working practices. Part of the process involves coaching teams as they move to Agile.

Track Session

Chris Comey

“Key attributes & responsibilities of a Test Manager”

The elements required to make a successful Test Manager are many and varied. You move from a testing role to a management role. You might still be doing some testing, but it is likely you will be focusing on planning, monitoring, controlling and reporting on testing. There are new aspects to your work required such as analysis, scoping and planning skills, communication skills and the ability to manage your team and stakeholder expectations. For a Test Manager, leadership, people management skills, motivation and issue resolution are essential. You have to consider many factors that may vary with context such as team location and culture, industry sector, deadlines and available resources, external change or regulatory change. Quite a lot to think about!

You need to understand the fundamentals of testing but as a Test Manager you must be prepared to shape the less ‘standard’ areas of the work in order to integrate with the particular situation in which you find yourself. Create and publish your plan within that context and prepare in advance for change and the need to adjust the plan as information becomes available or circumstances change.

This presentation will discuss the attributes, experience and awareness that a Test Manager needs to possess in order to succeed. Most of all it is when things are going wrong that a good test manager makes a key contribution to keep everyone calm, objective and focussed.

Three key points:

- The activities, attributes and experience required to be a “good” Test Manager who “keeps it real” will be presented for consideration.
- The key testing activities that require test management will be highlighted and discussed.
- The importance of tracking and reporting key information and managing stakeholder’s expectations, especially what is needed when things “go wrong” will be discussed.

Chris has been testing for 34 years during which time he has performed roles from Test Analyst through to Test Consultant and Trainer. With 15 years in total spent in the Test Manager role, Chris has spent the last 8 years managing test teams in the finance and legal sector and was recognised in 2013 winning the UKTB Test Manager of the Year at the TESTA awards ceremony. Having presented and conducted workshops at several testing conferences over the years, including EuroSTAR on 3 occasions, Chris is an experienced presenter who knows what the key aspects of test management are, and the importance of being 'ready' for disruption in all aspects of the process.

Workshop

Kristina Rungano Masuwa-Morgan, UKSMA

“Defect Measurement and Analysis”

Back by popular demand!

The aims of the workshop are to:

- Promote defects measurement as part of UKSMA's mission of promoting software metrics and measurement.
- Familiarise the software community with UKSMA's “Defect Measurement and Analysis Handbook”.
- Engage with the professional community on the priorities for defect measurement.

This workshop and the UKSMA Defect Measurement and Analysis Handbook offer a unique opportunity to explore strategies and techniques for managing software defects and dealing with their impact on delivery costs and schedules especially in relation to contractual obligations. It covers defect measurement at the various stages of the Software Life Cycle and the Defect Life Cycle and explores the relative efficacy of different quantitative and qualitative measurement techniques and models. It also provides an opportunity for delegates to share experiences and challenges of using some of the contemporary defect logging and measurement tools and provides opportunity to explore together the essential requirements of a defect log for effective measurement and control.

Three key points:

- Measurement and Analysis of defects.
- Defect logs, and Defect logging and measurement tools.
- Contractual and cost issues for defect measurement.

Dr Masuwa-Morgan is on the UK Software Metrics Association (UKSMA) board and is also the Faculty Director of Learning, Teaching and Assessment (FDLT) at Canterbury Christ Church University. She has had the privilege of working in the Computing industry since the 80s, as Operations Manager and then as an analyst/programmer, logistics manager and then software manager before joining Higher Education as a lecturer in Computer Science, Digital Marketing, Digital Business, Business Information Systems and Information Systems/eBusiness strategy and leadership.

She has programmed and taught a host of programming languages including old school languages like COBOL, FORTRAN, PASCAL and so on right up to modern OO platforms such as Java, C and now mainly web development platforms and scripting languages. Her specialist interest areas are in human factors aspects of computing which she sees as key facets of Quality control. She has published and developed work in the area of accessibility requirements management. Her interest in developing tangible usability measurements resulted in her joining the UKSMA which shares the goal of promoting and improving software measurement and metrics within the software community.

Dr Masuwa-Morgan has also worked as Faculty Quality Officer, and Technology Enhanced Learning and Teaching Co-Ordinator. This, in addition to her work as Faculty Director for Learning and Teaching, gives her great insights into metrics and Quality Assurance. She also continues to provide consultancy services mainly in the areas of accessibility, web development and IS strategy. She is also a creative writer (published under the name Kristina Rungano) and she pairs this with her work in Computer Science by making links between literature, especially poetry, and Knowledge Management.

Quick Talks

Damon Rands, Wolfberry

“Digital foot printing - Building an attack”

Almost everyone at one time or another has been guilty of sharing too much information: from job specifications to social media the explicit and implicit trail left behind provides a wealth of valuable information. Understanding how to best protect your clients' intellectual property during testing and the impact of your online activity is essential in today's world.

Three key points:

- Understanding the impact of your company's digital footprint.
- How an attack is built and deployed.
- How to mitigate the threat by testing and during testing.

Damon is an experienced cyber security expert, with over 25 years working in the Information technology industry. Having worked in development, support and installation, he has helped companies across the United Kingdom build secure systems and implement best practises in order to protect their intellectual property and private information. He specialises in:

- *Digital Foot Print analysis.*
- *Pen Testing.*
- *Business Continuity.*
- *Cyber Essentials Certification.*
- *IASME Certification.*
- *Security Awareness Training*

Mark Rice

“Software Localisation and Software Localisation Testing: An Overview and Case Study”

This paper explores the high-level processes of software localisation and software localisation testing, in addition to discussing the importance of these processes and some of the challenges facing the industry today. While software localisation occurs for many forms of software, the motif of this paper is video games, and my experiences as a software tester and project manager inform the article.

Three key points:

- Software localisation and software localisation testing are vital concepts, particularly for video games.
- Numerous challenges face software localisation and software localisation testing, many of which were not present in the Golden Age of video games, such as social media criticism.
- In particular, there is a lack of software localisation testing certification.

Mark is a functional & localisation software tester and project administrator. He has previously worked for the Release Management, Localisation Services and Creative Services departments of his organisation and is currently seconded to the Business Operations department. Mark has a PhD in psychology and is qualified in Advanced ISTQB (Test Manager/Agile), Scrum, ITIL, PRINCE2, TMMi and 6 Sigma. He also volunteers as a glossary reviewer for the ISTQB.

Track Session

Prakash Ijral, HCL

“Digitization of a testing practice”

Without most of us taking notice, our world has gradually been digitalized during the last 20 years. Digital products and services can be found in almost all areas of our life. Executives in all industries are using digital advances such as analytics, mobility, social media and smart embedded devices – and improving their use of traditional technologies such as ERP – to change customer relationships, internal processes, and value propositions. A very interesting aspect is also how companies are seeing new options in the way they interact with customers and develop and release products while they are empowered by constant connectivity, the rise of social networks, and an increasing amount of software in products. They are speeding up cycle times and shortening learning curves by testing new products or ideas with consumers using mock-ups, computer-generated virtual products, and simulations. This paper explores the significance for organizations to excel in the digital industrial economy in order to become more technologically capable and sophisticated and also the changed role testing specifically test automation will play in the era of digitalization.

Key highlights of our research paper include:

- Why digitalization should be taken seriously? – Consumer, Businesses, Information Technology and Testing viewpoints included.
- Changing testing role in Digital Era.
- Analysis around key market forces for QA decision makers to consider.
- Study on how radical shifts in market forces (SMAC , IoT , 3D printing , E commerce and user experience) translate to radical shifts in business models. It also covers the key components of an appropriate testing innovation strategy that ties in with the corporate vision and company capabilities and helps in determining the best ways of fostering and sustaining organic innovation.
- Suggested long term and short term strategy.

We believe the audience will gain a perspective of how a testing organization (People, technology and Tools) require meeting the customer digital needs. As organizations move from non-digital or digital to a “REBORN digital “phase due to the huge demand from customer end, testing functions need to tighten their seat belts to enable themselves to help deliver world class services.

Prakash is responsible for defining and implementing the TCoE roadmap across the organization and has over 17 years’ extensive experience of working globally and implementing Centres of Excellence (Test Factory) across multiple clients in USA and Europe. Expertise in spearheading complex Software Testing Projects and played roles of Consultant/Test Architect for various Global Operation Centre’s and for many Clients in the USA, Europe and ANZ Geo. He possesses extensive Software development and testing experience spanning across Energy Utility, Manufacturing, Financial Risk Management, Retail, CRM, Business Intelligence, Health Care, Publication, Vendor Management and Data centre management areas. Prakash is trained at IIM Bangalore on Leadership and an Engineering graduate in Electronics and Telecommunications with Post Graduate Diploma in Advance Software Design and Development. He is a Microsoft Certified Solution Developer and has also played the role of Defect Prevention Council Head for HCL’s CMMI5 initiative.

SIGiST White Paper Scheme

We have set up an area on the BCS website of a searchable repository for white papers and articles on testing and we are looking for contributors. That means you!

Do you have an existing paper you would like to repurpose and make more widely available through the SIGiST website?

- Then please send us the paper with three keywords for searching.

Would you like to write a new paper?

- Please send us the title and abstract together with the three keywords (or phrases)
- We will review the proposal and guide you through the authoring process
- For those who are thinking of speaking at SIGiST then this might be a good way to prepare a talk and get some useful feedback

If you have been thinking of writing or publicising an existing paper then this is the ideal opportunity. Please email your existing paper (with keywords) or your proposal to The Tester Editor, phill.isles@bcs.org

Past articles from The Tester will slowly be added to the repository as well.

Follow this link to the repository: <http://www.bcs.org/category/18128>

Write an article

We are always on the lookout for new content, so if you have a testing story you would like to share, a test technique you would like to evangelise or testing research you would like to publish, then The Tester is the place to do it. Simply email the Editor on phill.isles@bcs.org

Webdriver integration with JMeter for simulating browser based virtual users

Praveena Sridhar, Parikshit Chakraborty & Dinesh Kaarthick,
SSF-Engineering Cisco Systems

Abstract

This is an automation framework developed using Google's Webdriver and JMeter to do Load testing of complex AJAX applications. The framework has been designed using JMeter's abstract implementation -

AbstractJavaSamplerClient. This framework leverages page object model of Webdriver based automation framework for feeding JMeter to solve the performance benchmarking of AJAX based applications.

We have many tools to perform server side performance testing, like Loadrunner, SilkPerformer, NeoLoad, *etc.*, some of which also provides options for client side performance testing. But, the choice is very limited for open source tools to perform client side performance testing. This paper aims to explain one such tool which helps in enabling a client side performance testing for modern day complex web applications.

Index Terms - JMeter, Performance testing, Webdriver, AbstractJavaSamplerClient.

I. INTRODUCTION

API stands for Application Programming Interface which specifies how one component should interact with other. It consists of a set of routines, protocols and tools for building the software applications. The API Testing is performed for the system, which has a collection of API that ought to be tested.

1.1 Problems

1. Traditional open source tools for Load testing can't handle complex Web2.0 features like AJAX.
2. No open source solutions available to handle real browser based load testing.
3. JMeter aggregates the response time for the HTTP calls and doesn't differentiate the sequential vs parallel calls made using AJAX.
4. Multiple browser based load testing not possible through open source tools.
5. The parameterized inputs for JMeter are static per test execution.

1.2 Solution

1. Since Webdriver based test framework classes are being invoked the AJAX based HTTP requests are taken care of from browser, for the AJAX calls depending on the application and use cases, custom Webdriver API can be called to confirm that the responses related to AJAX have been addressed for parallel calls response calculation.
2. Since it is Webdriver based load tests we can have multiple browsers across virtual users during the execution.
3. JMeter agents will just leverage the executables / class files of Webdriver based framework on the hub.
4. The TestNG framework integrated with Webdriver based automation framework gives the ability to parameterize the inputs for the performance calls on a method to method basis at runtime.
5. JavaSamplerClient interface defines the interactions between JavaSampler and any external java program that is to be executed.
6. Implement the runTest() method of JavaSamplerClient to pass the inputs from JMeter to Webdriver scripts and invoke the scripts.
7. The Webdriver based scripts are invoked sequentially in the order of the execution of a test case.

1.3 Example Test case:

- LoginPage.login()
- LandingPage.action_1 ()
- LandingPage.action_2()
- FunctionalPage1.action_1()
- FunctionalPage2.action_2()
- LogoutPage.logout();()

The order of the test execution can be ensured by passing the Session ID /

Browser handler to the next page object method call in the test case.

1.4 Example Webdriver Grid2 - JMeter Integration

```
public class InvokeWebdriver extends
AbstractJavaSamplerClient implements Serializable
{

@Override
public Arguments getDefaultParameters() {
    Arguments params = new Arguments();
    return params;
}

private void listParameters(JavaSamplerContext
context)
{
String name;
for (Iterator argslt =
context.getParameterNamesIterator();
argslt.hasNext();
System.out.println(name + "=" +
context.getParameter(name)))
{
    name = (String) argslt.next();
}
}

public void setupTest(JavaSamplerContext
context)
{
}

public void teardownTest(JavaSamplerContext
context)
{
}

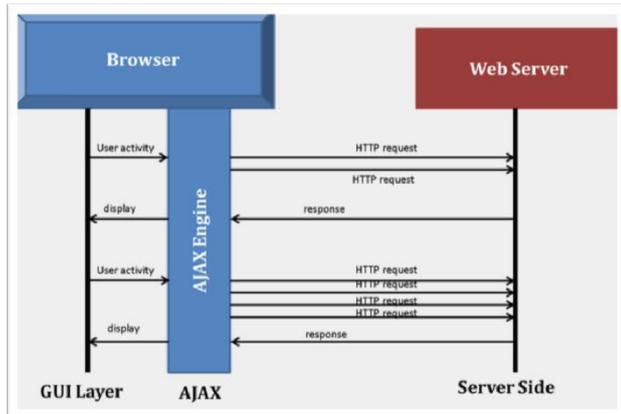
@Override
public SampleResult runTest(JavaSamplerContext
context)
{
    SampleResult results = new SampleResult();

    //Invoke Webdriver Classs here

    return results;
}
}
```

- Once the Integration code is written it has to be converted into a Jar file and then placed inside the \$JMETER_HOME/lib/ext directory.
- Now we can open JMeter and a Thread Group - Java Sampler and choose the InvokeWebdriver class from the dropdown.

II. AJAX BASED WEB APPLICATION



In the traditional web application the communication between the browser and the server happens directly and whenever the user requests for a page the server responds with the actual data. Whereas in AJAX based applications the AJAX Engine acts as an intermediary and takes care of the communication with the server in the background and the user is displayed with the available content. Also there can be multiple calls going to the server at the same time. In this scenario JMeter will invoke those requests sequentially and also show the aggregated response time for those calls. By combining the Webdriver scripts with JMeter we can get the real-time response time since the browsers by default handle the asynchronous / parallel calls.

II. FRAMEWORK IN NUT-SHELL

1. Parameterization

- By using the Java Sampler to integrate Webdriver scripts all the variables from JMeter script such as No. of Threads, Username, File inputs etc. can be passed on to the Webdriver script.
- By using the Webdriver TestNG based data provider

we can parameterize the attributes per method execution.

2. Invocation

The various actions that are performed in a web page have a corresponding method associated in the Webdriver class and a separate JMeter custom sampler can be created for each action.

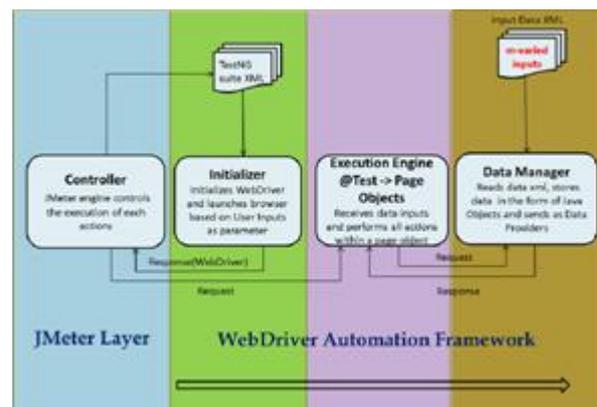
3. Response Assertion

Custom Webdriver API have to written to ensure that the page is completely loaded to take care of the AJAX response assertion.

4. Results

Results calculation can be easily performed because the browser takes care of the AJAX requests and we will be able to calibrate performance data per method in a page

5. Framework Architecture



6. Advantages of the Webdriver & JMeter Integration:

- The benefit of using Webdriver automation scripts in JMeter is that it provides the ability to utilize the same scripts for Load testing.

- Using this approach we can easily integrate the Functional & Performance test tools, which is Webdriver with JMeter.
- Load test results can project the real end user browsing experience by taking the browser rendering time also into account.

REFERENCES

[1]

http://www.tutorialspoint.com/software_testing_dictionary/performance_testing.htm

[2]

<https://www.safaribooksonline.com/library/view/selenium-testing-tools/9781849515740/ch08.html>



Praveena Sridhar (Quality Champion and Security Advocate)
SSF-Engineering Cisco Systems, SAN JOSE, United States
pravsid@cisco.com

With 16 yrs experience in S/w Testing across companies like Microsoft/BEA/BMC/Intuit/Cisco Systems. Expertise in domains of automation frameworks development in UI, API and security testing. Expertise in performance testing, High Availability testing and E2E tests for large scale enterprise environments. Presented papers in next gen testing conferences in Kerala and Sri Lanka, Bilbao-Spain, Brussels-Belgium, New York. Qualification: Bachelor of Engg from BITS-PILANI India and Software Engg Course from Stanford Palo Alto, Pursuing Advanced computer Security Certification from Stanford.



Parikshit Chakraborty (Tata Consultancy Services @ Cisco)
SSF-Engineering Cisco Systems, Bangalore, India
parichak@cisco.com

Have over 7 years of experience in software testing across companies like PointCross, Ciber Inc., GT Nexus and Tata Consultancy Services. Have expertise in the domains of automation frameworks development in UI, API and E2E tests for large scale enterprise environments. Have expertise in tool development for automating security testing. Presented papers at Next Gen Testing Conferences, India Chapter at Trivandrum, Kerala and SriLanka Chapter at Colombo. Apart from that white papers got selected for conferences at Bilbao-Spain, ICTSS 2015 New York. Also, got best white paper award at BTD 2015 conference at Brussels-Belgium. Like to pursue anything around automation and framework development. Also, like to learn new trends and technologies and would like to work on new tools, if any, around automation.



Dinesh Kaarthick (Performance Test Lead)
SSF-Engineering Cisco Systems, Bangalore, India
dkaarthi@cisco.com

Have almost 12 years of experience in development and testing projects across companies like Tata Consultancy Services, Cisco Systems. Have expertise in the domain of software performance testing using tools like HP Loadrunner, JMeter etc. Have expertise in software project development using Java. I am very passionate about Performance engineering and an aspiring Performance Architect and wanted to do large scale computer performance design, analysis, and tuning.

Why I changed my mind about UX Certification

David Travis, Userfocus Ltd

I'll admit it: when I used to hear people advocate professional certification in user experience, I was dismissive. Since we can't even agree on what "UX" is, how can we certify it? I wondered. I saw certification as a way of creating a closed shop to exclude dissenting voices. This is the story of why I changed my mind.

The arguments against certification

I found the "closed shop" argument a persuasive one. Did I want an elite squadron of user experience professionals defining user experience to be the narrow area of work that they did day-to-day? And how would this select cadre decide if someone was suitable to join their club: a multiple choice test couldn't possibly demonstrate the real-world expertise that people need to carry out user experience work. I was concerned that certification was simply a scheme — some might even say a scam — drawn up by training companies to boost delegates.

Back in July last year, I attended a panel held at the UXPA Annual Conference in London where they discussed this very issue. Nothing I heard changed my mind. In fact, I became more convinced of my view. A comment from one of the panellists at that meeting resonated with me. The panellist said that job candidates who put vendor-based UX certifications on their CV are automatically excluded from interview in his firm — because including the certification demonstrates their **lack** of knowledge, not its presence.

Since that time, four events changed my mind.

The arguments for certification

First, I worked with a client that asked for certification. This client (part of a large Government department) wanted to train and develop some of its employees so that they could do fundamental user experience activities, like user research and usability testing. I pointed out that they could achieve this without certification — but for this organisation, certification mattered. This is because the organisation saw certification as establishing a development path for employees and specifying a minimum level of competence. In fact, the organisation was astonished that there was no industry-wide scheme in place, since certification schemes exist in many other areas of IT, such as in business analysis.

Second, I realised that we need more foot soldiers. Without a certification scheme, we fall back on a secret society where only people who have worked in the field for many years know the conventions, language and culture of user experience. This creates a barrier to entry for novices — and simultaneously creates a culture where we laud “rock star” UX designers who know the secret handshake. But right now, looking at the state of product design, UX rock stars aren’t sufficient. If they were, there wouldn’t be so many poor user interfaces around. What the user experience field needs is more foot soldiers: people whose role is simply to convince their project teams to focus on users and their tasks, design iteratively and run usability tests. If more development teams worked that way, we would transform the lives of ordinary people — our users — overnight.

Third, whether or not I think certification is a good idea is irrelevant. The genie is out of the bottle. I run some online courses on UX that have around 6000 students and on a daily basis I receive emails from students asking about certification. I see students discussing the merits of the different vendor-based certification schemes and deciding to take one or other exam. However, as much as I respect the various training organisations that offer these certificates, it can’t be a good idea for the organisation that does the training to also be the organisation that does the certifying. There is an obvious conflict of interest which needs to be addressed, even if the genie is out of the bottle.

The fourth and final part of the jigsaw for me was realising that the argument, “We don’t know what UX is, so how can we certify it?” is nonsense. It’s true that there are many conflicting viewpoints about good practice in user experience. But we have a standard — [ISO 9241-210](#) — that has been debated and assembled by a team of international experts and that sets out the fundamental competencies of our field. Standards, especially International Standards, provide independent and authoritative guidance. The discipline of having to achieve consensus to create an international standard helps moderate some of the wilder claims of user experience enthusiasts and helps ensure that the resulting standard represents good practice.

Towards an independent UX certification scheme

So I’m delighted that since having my road to Damascus moment, I’ve been working with BCS — a well respected, independent certifying organisation — to create a suitable scheme. Although still in its beta phase, there will be two levels of certification: Foundation and Practitioner.

[The Foundation certificate](#) aims to create more foot soldiers who speak the language of user experience. Because the syllabus is based around the ISO 9241-210 standard, we can be sure the certificate encapsulates best practice. To pass, candidates will need to take a 1-hour multiple choice examination. The good news is that you do not need to attend a training course to take the examination because certification has at last been decoupled from training providers.

[The Practitioner certificate](#) will be partly examination based and partly based on a portfolio review. This certificate sets a much higher bar than the Foundation certificate. Since UX

professionals are now accustomed to submitting a portfolio as part of their job application process, it shouldn't take a great deal of extra effort to apply for Practitioner certification as part of the business-as-usual process of updating a portfolio. What differentiates this process for Practitioner certification is that candidates can't simply regurgitate what they learn on a course: a portfolio demands evidence of real-world practice.

My hope is that, with a heavyweight organisation like BCS behind it, we may at last end up with a UX certification scheme to be proud of. If you want to obtain the certificate, or train people to take the certificate in UX, you can [register your interest in the scheme at the BCS web site](#).



Dr. David Travis (@userfocus on Twitter) holds a BSc and a PhD in Psychology and he is a Chartered Psychologist. He has worked in the fields of human factors, usability and user experience since 1989 and has published two books on usability. David helps both large firms and start ups connect with their customers and bring business ideas to market. If you like his articles, you'll love his online user experience training course.

Further details can be found at <http://userfocus.co.uk/> and <http://uxtraining.net/>

This article first appeared on the Userfocus website in May of this year.



15 minutes with

Phill Isles of HSBC Private Bank



JW5 Associates

Jennifer Wheeler of JW5 Associates spent 15 minutes with **Dr Phill Isles**, Head of Testing for HSBC Private Bank (UK), Limited, talking about his career in testing so far.

How did you get into testing in the first place?

Well that's a bit of a long story, where do I start. It was not planned at all. My personal circumstances changed and I needed to find a job close to home. A friend of mine who worked as a management consultant offered to review my CV. His wife worked for a Software Testing Consultancy and mentioned her company were recruiting Testers. I went through quite a rigorous interview process (three interviews and a test) and was then offered a Test Analyst role which was great. Up to this point I had been a scientist working as an Electron Microscopist. (I have a PhD in Clastic Sedimentology!)

What do you think helped you develop your career the most?

My background in science has certainly helped with the analytical approach ("What happens if I do this?"), however I think sitting the ISEB Practitioner exam helped. The company I was working for at the time supported a number of us to attend a training course and sit the exam. Because I passed with Distinction I was invited to apply for a place on the ISEB Software Testing Accreditation panel which was brilliant. I got to meet experienced people in the testing community and as a result got involved in the SIGiST (Specialist Interest Group in Software Testing) community. Being around people with different experiences and sharing knowledge has been very helpful.

I see you have taken the ISEB Foundation and the ISEB Practitioner Certificate in Software Testing. Why did you decide to take these qualifications and do you think having them has helped in your career development and if so in what way?

I sat the Foundation exam because everyone who worked for the first Testing company I joined (Imago QA Ltd) was required to, even the receptionist! I remember I joined the company on the Monday and sat the exam on the Friday! I passed and at that point was a Certified Tester, however I was far from qualified and knew very little. What was really good was that it gave me the language needed to communicate with Testers and other software development people, and so the ability to learn.

The Practitioner exam. I was working for a different company who made some funds available for training and a group of us chose to do the Practitioner. We had formal training from an accredited provider but we also organised our own discussion sessions to review sections of the syllabus. I found the training and exam very helpful from a hands-on testing perspective, introducing me to many more testing approaches and methods.

In the Software Testing industry, there is a continual discussion for and against certification, I am definitely in the 'for' camp. However certification doesn't make you a qualified tester in my opinion, it is just a good place to start.

You have worked in software testing for 15 years. What advice would you offer people at the beginning of their testing careers who are keen to travel down a similar path from 'hands on' Testing to Test Management?

Don't just work for one organisation. Also, get out and go to conferences and events to hear different views; speak with people who have different experiences of testing and swap ideas. Some companies are restricted in their testing processes. If you go to events and pick up just one or two small ideas to help make improvements back in the office, then it was worth going.

Getting away from the day job and hearing people speak from very different organisations and companies can give a fresh perspective.

Staying up to date with new approaches to testing, tools etc. can be a full time job in itself. How do you stay in touch with what is happening in the market without getting swamped?

There is a lot of information and content on the web, so as well as going to events and conferences you can take part in webinars which are often free. Some conferences live stream their keynote speakers. If your company won't fund you to attend or allow you to take three / four days 'off' for a conference, look out for the free streamed presentations. There is a lot out there, so picking well is key. Perhaps start by picking high profile speakers from a range of testing industries, including traditional testing and Agile topics.

It may be helpful to set a target of attending one conference a quarter and taking part in one webinar every two months or so.

Communal Testing blogs are also a good source of information, such as LinkedIn. I find individual blogs can be restricted in topics covered.

You have been the Test Manager at HSBC Private Bank since 2007, can you tell us a little about your role and key responsibilities?

I must start by saying I am not a licensed spokesperson for HSBC so everything in this article is my personal view. I manage the testing of all the applications used by the UK Private Bank. This means I am responsible for everything from Test Policy to methods and approaches used.

How has your approach to testing at HSBC changed since you joined the company 8 years ago?

When I first joined IT projects followed the HSBC Group approach (to an extent) and I took over with the same remit. I am now encouraging more flexibility and changing the Testing approach to meet the context of the work being undertaken. I guess you could say our projects are Tested with a context driven approach, using both traditional and Agile methods.

What plans do you have to change anything around testing?

As I mentioned HSBC Group has a fairly fixed process regarding testing. I am actively involved with the in-house Testing community though and try and make changes where I can.

It is not easy or a quick process to get a consensus to change across such a large group. However we are making inroads towards improving testing practices. Pilot projects on various approaches are a useful way to see if a new approach or tool will be beneficial.

What do you most enjoy about heading up a testing practice?

The variety and scale of work is really interesting. One day we will be testing a single Word template for a specific letter, and the next we may be putting a new CRM system through its paces. Covering one organisation means there are lots of different applications to work on, and I find the variety stimulating because they can all harbour different types of errors.

Working in a Private Bank is interesting by itself. We don't see any client data, but we have to test with large numbers which I find fascinating. For example, when testing systems that deal with foreign currency exchanges, we have to enter large numbers. So, years ago when I caused a system to fail because I entered 99,999,999.99 into a numerical field and the developer said "don't be silly, no one will ever enter such a large number", they actually might in the Private Bank world.

Is there any other advice you would pass on to the testing community?

Volunteer for a role somewhere within the industry.

I have been going to conferences and events for years and recently I thought it was about time I started giving something back. With the help and encouragement of a good friend I have now presented at a number of conferences. The advice I would give people is although you might not be ready to present straight away, it is really helpful to start making notes on things you experience in your work. Maybe a project you worked on where you learnt something new about testing; a good story about something which went well or testing tools you evaluated; a new approach you tried that went well or not so well. People are very interested to hear about the good, the bad and the ugly of Testing! Keeping a log is both helpful to yourself and also may be useful to others if you choose to present in the future.

For more information on writing an article for The Tester or presenting at a SIGiST conference, please contact Phill at phill.isles@bcs.org



For help and assistance in any aspect of Testing career planning or recruitment please contact Jennifer Wheeler at Jennifer@JW5.co.uk or call on 07733 121897

JW5 Associates

Event Listings

If you would like your event listed here, please contact the Editor phill.isles@bcs.org

2015

November

EuroSTAR

2 – 5 November 2015

Maastricht, Netherlands

<http://www.eurostarconferences.com/>

Agile Testing Days

9 – 12 November 2015

Potsdam / Berlin, Germany

<http://www.agiletestingdays.com/>

December

SIGiST

2 December 2015

London, UK

<http://www.bcs.org/server.php?show=nav.9264>

2016

March

SIGiST

15 March 2016

London, UK

<http://www.bcs.org/server.php?show=nav.9264>

May

STAREAST

1 - 6 May 2016

Orlando, US

<http://stareast.techwell.com/>

June

Belgium Testing Days

13 – 16 June 2016

Brussels, Belgium

<http://btdconf.com/>

SIGiST

9 June 2016

London, UK

<http://www.bcs.org/server.php?show=nav.9264>

September

SIGiST

15 September 2016

London, UK

<http://www.bcs.org/server.php?show=nav.9264>

STARWEST

TBC 2016

Anaheim, US

<http://starwest.techwell.com/>

December

SIGiST

7 December 2016

London, UK

<http://www.bcs.org/server.php?show=nav.9264>





Personal Development Plan

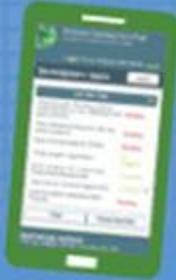
Your portable record of professional aspirations and activities



Desktop/Laptop



Tablet



Smartphone

Did you get your Personal Development Plan email with suggested potential CPD activities?

The BCS Personal Development Plan (PDP) uptake is going well, with thousands of registered users already actively recording their CPD Development Goals, Activities and preferences. It's not just about recording details though, as there is a Resources section that shows live feeds of potential CPD activities, and a tailored email is sent every 2 months with details of the latest videos, articles, blogs, books and research in your specified field of interest. If you haven't registered yet, you can see the content from the latest PDP bulletin for topics relating to solution development and implementation here <http://www.bcs.org/content/ConWebDoc/50854> or by going to the CPD Portal at: <http://www.bcs.org/pdp/>.

The BCS Personal Development Plan is free to use; BCS members can use their Member Secure Area login and password to access it at <https://pdp.bcs.org/>, and non-members can use most of the facilities (using the same link) and registering to create their own user name and password. You can use it on a PC / laptop or compatible tablet PC or smartphone.