Software Testing Specialist Group

Issue 36

# THE TESTER

March 2011 Issue

## Next Conference: Wednesday 16th March 2011

**Bob van de Burgt**

**Mike Bartley**

**Alan Richardson**

**Stevan Zivanovic**

**Henrik Andersson**

**Phil Stead**

**Anna and Linda Hoff**

# ‘Times they are a changing!’

## Conference presentations

To open the day **Bob van de Burgt**, of the Netherlands, will give us ‘***An Introduction to Lean Test Management***’. A vital part of today’s test management toolkit. Bob is a great enthusiast for change, and has been a leader of the Dutch testing community for many years.

This is followed by two track sessions, firstly **Mike Bartley**, who broadens the theme and asks us the question ‘***Have you noticed, even the hardware is changing?***’ and looks at the effects this will have on how we both write and test software.

Secondly, to whet our appetites before lunch, **Alan Richardson** is going to talk to us about ‘***Learning, Developing, Evolving: The Path of the Technical Tester’.*** Alan in his inimitable style is going to reveal his secrets of Technical Testing. A definite must.

The afternoon will kick-off with a short bonus session from Stevan Zivanovic, the theme of which is ***Social Networking for Testers***. An informative session with a real frisson of danger as we will aim to Tweet, Live, and see what people are saying about the day!

This is followed by two track sessions which give us interesting and complementary views of change. The first of which is presented by **Henrik Andersson**, from Sweden (apologies I got this wrong in the Early Program announcement), who will tell us his story of ‘***Introducing Exploratory Testing Champions***’.

And this is followed by Phil Stead, who will explain that ‘***Changing Testing is about Changing People***’, a view developed from his experience delivering large change programs.

To close the day we have a special performance of ‘***THE SUPER TESTERS - A slightly True Story***’, given by Anna and Linda Hof. This is probably the most entertaining testing play you will ever see!

## In this issue

### CONFERENCE PROGRAM

### HOW TO

### ARTICLES

### TESTING EVENTS

# FROM THE EDITOR

**Matt Archer, Editor**

Our first article this month comes from **Graham Parsons** - CEO at Reflective Solutions and visionary behind StressTester™**.** In his article, '**Performance Testing in an Agile Environment: Making the impossible possible**', Graham described how it is possible to undertake performance testing during every sprint or iteration through the project, rather than just at the end (arguably, when it is too late to respond).

Our second article has been written by **Andrew Gibbons.** Andrew took advantage of the open networking session at the last SIGiST conference to briefly talk about apprenticeships for testers. In his article, '**Software Test Analyst Apprenticeships: Developing the Test Managers of Tomorrow**', Andrew presents the many benefits of apprenticeships to both employers and tester alike. A must read for anybody thinking of expanding their team with junior members.

Our third article has been written by **Mike Bartley** of Test and Verification Solutions. In his article, '**Testing Concurrent Software**', Mike explores the rising trend of parallel computing and the impact it has on software testing.

In our final article, '**EuroSTAR is over --- I look forward to EuroSTAR**', **Peter Morgan** explains why this year's EuroSTAR conference should be bigger and better than ever. If after reading Peter's article, you feel compelled to attend, don't miss the 10% discount on page 17.

If you have been inspired by any of the articles in this edition and would like to write an article for *The Tester* yourself, then please feel free to email me.

Matt Archer

The Tester Editor
BCS Specialist Group in Software Testing
matthewjarcher@googlemail.com

# WEB LINKS

**SIGiST conference website:**
**www.SIGiST.org.uk**

**Standards Working Party:**
**www.testingstandards.co.uk**

**LinkedIn Page:**
**http://www.linkedin.com/groups?mostPopular=&gid=3466623**

# LINKEDIN AND TWITTER

The BCS Software Testing Specialist Group is now using social media platforms to improve communications both to members and between members.

Our LinkedIn Group (link below) will carry details of our conferences as they become available. It will also provide a place where people can discuss testing topics, make requests about future conferences, find employment opportunities (there are a few jobs advertised already) and generally keep up to date with our chosen industry. If you are already a member of LinkedIn then simply visit the group and make a request to join.

If you're not a member then go to http://www.linkedin.com/ to create an account.

If you use Twitter you can follow us @SIGiST.

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

# CONFERENCE BOOKING INSTRUCTIONS

**If you would like to pay online, you can use our new online booking and payment system.**

**www.bcs.org/events/registration**

**If you would like to pay by cheque, you can download a booking form.**

**www.bcs.org/upload/pdf/sigist-bookingform.pdf**

**If you have a query relating to making a booking, please contact Gemma Stanley-Evill, Specialist Groups' Officer.**

**Tel: (01793) 417656**

**gemma.stanley-evill@hq.bcs.org.uk**

# CONFERENCE AGENDA

# ‘Times they are a changing!’

**Wednesday 16 March 2011**
**Royal College of Obstetricians and Gynaecologists**
**27 Sussex Place, Regent's Park, London NW1**

| Time | Session | Length |
|---|---|---|
| **08:30** | **Registration open, Coffee and Exhibition Hall** | |
| **09:25** | **Welcome & Introduction**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |
| **09:30** | **An introduction to Lean Test Management**<br>**Bob van de Burgt –** Professional Testing | **60** |
| **10:30** | **Networking Session** | **15** |
| **10:45** | **Tea / Coffee Break Exhibition Hall and Networking** | **30** |
| **11:15** | **Have you noticed, even the hardware is changing?**<br>**Mike Bartley -** Test and Verification Solutions | **45** |
| **12:00** | **Learning, Developing, Evolving: The Path of the Technical Tester**<br>**Alan Richardson -** LMAX | **45** |
| **12:45** | **Buffet lunch, Exhibition Hall and Networking** | **75** |
| **14:00** | **Social Networking for Testers**<br>**Stevan Zivanovic -** Experimentus | **15** |
| **14:15** | **Introducing Exploratory Testing Champions**<br>**Henrik Andersson** – House of Test Consulting | **45** |
| **15:00** | **Changing Testing is about Changing People**<br>**Phil Stead** - IBM | **45** |
| **15:45** | **Tea / Coffee Break, Exhibition Hall and Networking** | **30** |
| **16:15** | **THE SUPER TESTERS - A slightly True Story**<br>**Linda Hoff -** KnowIT Göteborg<br>**Anna Hoff -** KnowIT Create | **45** |
| **17:00** | **Closing Remarks**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |

# ABSTRACTS AND BIOGRAPHIES

## An introduction to Lean Test Management

### Bob van de Burgt, Professional Testing

Cost reductions and the quest for more efficiency are more evident in today's business world. It also follows that our testing processes will ultimately be affected.

Many management theories speak about "Lean" as being one of the solutions. One of the key steps in using "Lean" is the identification of which steps add value and which do not. This presentation will explore the use of "Lean" within testing and more specifically within test management.

As a guideline, the presenter will follow the "Lean manufacturing process": the generic process management philosophy derived from the Toyota Production System. It is renowned for its focus on reduction of "seven wastes" in order to improve overall customer value. Everything not adding value to the customer is considered to be waste. Examples are: rework, waiting and poor information.

This presentation will focus on the various elements as mentioned above. Also Six Sigma as being one of the more popular theories that introduces the concept of "Lean" will be explained.

This presentation is especially of interest to business managers, IT managers, QA managers and test managers that are involved in improving the quality of test management processes.

***Bob van de Burgt** is test consultant at Professional Testing bv. He contributed to the development of the testing method TestFrame® and the test management approach of Logica for which he also was co-author of the published books. Bob has given many testing courses (including ISTQB) and is a frequent speaker at (inter)national congresses. He has been  the chairman of the Dutch Special Interest Group in Software Testing TestNet for many years and was EuroSTAR Programme Chair in 2008.*

## Have you noticed, Even the hardware is changing?

### Mike Bartley, Test and Verification Solutions

This session explains why hardware manufacturers are putting multiple processors on their chips and why this matters to you as a software tester.

The highly practical session will look at the effects that multiple cores will have on how we both write and test software. We will consider real concurrent software issues such as non-determinism, race conditions and deadlocks. Similar issues already exist in distributed computer systems and we will discuss how we, as testers, can cope with these challenges.

By the end of the session you will have a thorough understanding of concurrent software issues, the challenges they pose to us as testers and the techniques we can employ to cope with them.

***Mike** has been involved in software testing and hardware verification for over 20 years. He started his career in testing of military software and safety-related aerospace applications using formal mathematical methods. He then moved into commercial hardware verification of a 64-bit MPEG4 chip at ST Microelectronics. From there he moved to Verification Manager at Infineon building up a team of over 35 verification engineers using state-of-the-art verification technology to verify numerous chips and design IP ranging from secure chip cards, through automotive applications to mobile phones. Mike then moved to start working with start-up companies in charge of both the testing of software products (tool chain, run-time libraries, applications, etc) and the verification of the hardware products - firstly at Elixent (now Panasonic) and then ClearSpeed. In these roles he established software testing and hardware verification teams (including offshore resources), flows and processes which were used to sign off numerous hardware/software products and are still in place today.*

*Mike now runs his own software testing and hardware verification consultancy helping with company strategies, benchmarking, project plan and execution reviews, training and providing teams both on-site and offshore. His customers' software and hardware products cover a wide variety of application domains: safety-related automotive; mobile phones; low-power controllers; Ethernet routers; high-definition televisions.*

*Mike gained a PhD in Mathematical Logic from Bristol University. He has since obtained an MSc in Software Engineering and an MBA through the Open University. Mike has had numerous papers published and presented at a number of conferences.*

## Learning, Developing, Evolving:<br>The Path of the Technical Tester

### Alan Richardson, LMAX

Can testers still coast by with a few techniques, and a few buzzwords? Can testers still survive by producing endless pages of tests without executing them and relying on the project getting canned after a few years? In some companies they possibly still can, but not for very much longer, Alan Richardson will explain why.

Software Development changes constantly. Software Testing needs to change too. The 'Test' no longer remains the sole property of the 'tester'. Developers and business analysts can write automated requirement tests numbered in the thousands without any help from testers. Testers need to evolve their role and approach. By adopting more technical approaches testers can add value that no-one else on the development team does.

The 'tester' needs to work quickly, to learn to approach the system technically as well as from the requirements domain. The 'technical tester' comes armed with tools, and the ability to construct their own tools. Able to strip the application down to its basics. Able to hone in on technical risk and exploit it. Ah, you ask, is it really possible to learn this dark art? Are these terrible secrets available for anyone to use? No, we say, not just anyone. For, only to those that attend this talk will Alan explain the approaches he uses to do this.

Of course, Alan will describe some of the free and open source tools he currently uses to approach testing technically. But this is no mere "Bluffers guide to Technical Testing". You will get more than just a simple list of tools and tool types.

More importantly will Alan will explain some of the mind sets, models, and techniques that form his general technical testing model, including:

- Learn the traditional testing concepts that stop you testing technically

- How to get more from your testing books than they ever intended to tell you
- How to build test techniques from first principles
- How to quickly build a map
- How to use the difference between map and territory to improve your testing
- The Ying and Yang of technical testing: Observe and Manipulate

Attendees will leave with an understanding of the need for, and the pragmatics of, technical testing . And a solid set of next actions and research topics leading them towards the path of increasing technical skill.

***Alan** has been involved with testing and development for his entire professional career. He had the advantage of growing up in the 80s when hair was large and computers were affordable, thus allowing him to learn to code badly at an early age and experiment with embarrassing hairstyles when they first came into fashion. This early technical start left him with a habit of learning new programming languages and technical techniques; and a desire to hack and tinker with technologies that he didn't fully understand. He has always approached testing with a technical perspective.*

*Despite acting in a management role he remains hands on and keeps his technical skills up to date. His recently published book on test automation "Selenium Simplified" is designed to help testers with no programming or technical expertise learn to construct production ready automation scripts in Java and Selenium 1.0*

*Most of his writing about testing can be found at compendiumdev.co.uk and eviltester.com*

*Alan Richardson is currently Head of Testing at LMAX, using Agile methods to help build the world's highest performance financial trading exchange.*

## Social Networking for Testers

### Stevan Zivanovic, Experimentus

Love it or hate it, “Social Media” is now an everyday tool. In true Tweet style we will briefly investigate what is meant by Social Media, e.g. Twitter, Linked-In and blogs, the typical tools that are commonly used and why we as testers may be interested – but in more than 140 characters. We will also do a live feedback session using twitter. If you are on Twitter, please post your comments on the SIGiST hash tag (@SIGiST) before and during the session and we can all see what you are thinking – scary? I hope so.

***Stevan** has been actively involved in the Quality arena for over 18 years, where he has worked in a wide variety of businesses, from Safety Critical, to financial, to “Dot Coms”. He has a pragmatic approach and has a proven track record of implementing realistic, workable, real world changes for large, multinational teams. Stevan has presented at EuroSTAR, BCS, UK TMF and Agile Business Conference, where his presentations were well received. He enjoys speaking and training people to motivate them in testing and quality.*

## Introducing Exploratory Testing Champions

### Henrik Andersson – House of Test Consulting

This is the story of how I introduced Exploratory Testing Champions at a Swedish telecoms company, and of how we met the astonishing request to implement ET in only 8 days to an organization with 80 testers! How we gathered the most passionate testers and tutored them to become champions. And how we reached out to many testers quickly, and also boosted the passionate testers further, recognizing them as skilled and special.

Firstly we ran "ET pilot days" for 6 teams to evaluate the fitness of ET (comments will be presented). A decision to roll out Exploratory testing in the organization was made with a huge constraint, we only had 8 days at our disposal!

Undeterred, we then assembled 9 passionate and dedicated testers to work with, rather than immediately reaching out to all 80 testers. We called this group our "Exploratory Testing Champions", with the aim that they would be capable of teaching and implementing ET in the test teams.
I will go through the content, purpose and outcome of each workshop that we had with this group. We accelerated the passion and skills of an exclusive group of testers with the mission to enthuse and develop a larger set of testers. Today our ET Champions are teaching exploratory testing with great success.
ET is now part of the working procedures since management has allocated separate time for ET sessions.

The objective of this presentation is to give you a road map and a tool kit to enable you to set up a similar project.
I will present a background to Exploratory testing, scope, contents and results of the workshops , responsibilities and expectations of the Exploratory testing Champions. I will also present real facts from this live case study, the schedule, comments from participants and where we are today.

***Henrik Andersson** is consultant and founder of House of Test, consultancy and outsourcing based in Sweden and China. He helps international companies increase their efficiency and reconstructing their testing. He provides leadership and management consulting for managers and leads. He tests, coaches, consults, speaks, writes, manages and thinks about software testing and problem solving.*

*He is a believer and member of the school of Context-Driven Testing. His way of testing is influenced, developed and inspired from and by James Bach, Michael Bolton, Cem Kaner, Scott Barber, Jerry Weinberg, and others.*

## Changing Testing is about Changing People

### Phil Stead, IBM

Testing is one of the most exciting places to be right now and this is driven by the dynamic nature of change being introduced into the industry. In this session, Phil will answer a number of key questions such as 'How does he innovate to provide value in delivering testing to ever-demanding clients?' What changes has he embraced and driven into the market to move the perception of testing from a back-end, commodity service, to that of a high-value, pervasive activity that drives benefit into business change programmes? And most importantly, what is the impact of this change on our testers, what new skills and ways of working are we embracing and developing? Phil will use real examples and case studies to share his experience and predict where the testing industry is going next.

***Phil** is an Associate Partner within IBM's Global Business Consulting Services (GBS) and leads IBM's Test Consulting Practice for UK & Ireland, which now totals over 50 consultants. Phil's career in IT spans 29 years and includes implementation, support, project management and consulting. For the last 12 years he has specialised in testing and has led some of IBM's largest test transformation programmes within the UK. Phil is responsible for growing IBM's testing business in the UK and is a member of IBM Global Test Leadership team with responsibility for turning test innovation into value for IBM's UK & Ireland clients.*

## THE SUPER TESTERS - A slightly True Story

**Linda Hoff, KnowIT Göteborg**
**Anna Hoff, KnowIT Create**

“The Supertester - A slightly true story" is a theater performance where we try to mix serious test issues and common prejudices with laughs. We will take the roles and personalities of project managers, testers and super heroes to a new level. We will use prejudice and radical opinions in our effort to entertain and inspire.

You will meet the project manager who blame the test department for everything that goes wrong. The project manager loves all the new and cool development methods. He is not making it easy for the poor shy little company tester who lost the last shred of self confidence the first day at work. Can she be saved by the magnificent Supertester and once again be restored to a strong tester who stand up for her profession? Will the powers of the Supertester be enough to save the organization?

Join us in a twisted but still slightly true world of testing.

Key Points:

- Have fun and be inspired to become a Supertester yourself!
- Extraordinary testing is a mindset.
- There are no golden solutions (ET, ISTQB, Scrum) only a perfect mix.

***Anna** and **Linda Hoff** are two sisters active within the testing profession. Both are ISTQB certified and have 14 years of testing and test leading experience together. Linda is also a certified TPI assessor and Scrum Master. Anna is experienced in exploratory testing. During their University studies the Hoff sisters enjoyed working with student farce theatre. Through the “Supertester” fun session they will use their creativity and passion for testing to promote the profession.*

# PERFORMANCE TESTING IN AN AGILE ENVIRONMENT: MAKING THE IMPOSSIBLE POSSIBLE

**Graham Parsons, CEO at Reflective Solutions and visionary behind StressTester™**

### The Agile approach

The adoption of Agile practices during the development of a web application can deliver significant benefits, both in terms of an application's quality and the enhanced productivity that generally results from the Agile development process.

One of the key purposes of Agile is to ensure that the current version of software functions as expected, no matter what stage the development is at. To this end, every iteration of an Agile project includes functional testing in the form of unit and acceptance testing. The early discovery and resolution of defects written into the code can help to guard against major unforeseen problems arising near project completion. After all, who wants to find a fault with the application weeks or even days before launch? This could severely compromise an application's launch date (and delivery budget) and could cost a company dearly, in terms of overspend, lost revenue or business opportunities. There is even potential reputational damage to consider…..

So, with a firm case in mind for functional testing during every iteration of an Agile project, together with an acknowledgement that this approach has become industry best practice, why do most developers apply a different logic when it comes to performance testing?

### The risk of performance testing at the end of an Agile project

While the mindset is slowly shifting, in line with the introduction of a new breed of quick to set-up performance testing tools, a large number of Agile organisations are still waiting until their project is nearing completion to performance test their applications. The inherent danger with this approach is just as real as it would be if functional testing was left until development was all but wrapped up.

When performance testing is left until near the end of the project, time and financial pressures often lead to testing shortcuts being taken, in order to bring the project in on time and on budget. This can result in incorrect and unrealistic testing practices which, for example, might fail to test the full range of typical user journeys , therefore not placing the application under realistic load with the result of probably not uncovering all the performance defects and bottlenecks.

A business would be taking a very big risk if it launched an application that does not perform and scale at times of peak user traffic, especially if it was business critical. However, leaving testing until late in the project might result in the testing team identifying performance and scalability issues just weeks or days before the application is launched. This could result in a potentially disastrous overrun on development time and budget.

### Traditional barriers to performance testing in an Agile project… and a solution

Performance testing tools have traditionally been based on complex, bespoke scripts which need to be crafted by specialist tool experts. The process of writing, testing, correcting and re-testing the script code for the tool itself can take weeks and all of this needs to take place before any performance testing can be carried out on the actual application.

The timescales, costs and complexity associated with traditional performance testing tools, have led to a commonly held belief among IT managers and testers that the only place for performance testing within an Agile project is at the end of the development phase. While this belief is true when applied to the traditional tools which are still prevalent in today's marketplace, a new breed of quick to use testing tool has been specifically designed to enable correct and realistic testing during every iteration of an Agile project.

The simplicity of this new generation of tools enables configuration and testing timescales to be reduced from weeks to days. Overall testing costs can also be significantly lowered as the services of specialist script writers are not required and testing can be carried out in a much shorter timeframe, by any member of the IT team. When you consider that these tools do not compromise on the quality of testing, allowing the delivery of correct and realistic testing in a drastically reduced timeframe and at a fraction of the cost of traditional tools, it almost seems to be too good to be true. It isn't however and these tools, with the following common features, exist today:

- Easy to learn
- Full correct and realistic testing
- Quick to configure and use (including updates to test assets in a matter of hours)
- Minimal or zero-scripting
- Immediate diagnosis of problems

The availability of performance testing tools which require minimal scripting and even zero-scripting, offers a paradigm shift in performance testing capability to the Agile development community. For the first time, application performance defects can be quickly and easily identified and resolved as they are implemented, rather than at the end of the development process. The resulting efficiencies in project management, costs and timescales are substantial when compared to traditional performance testing tools and practices.

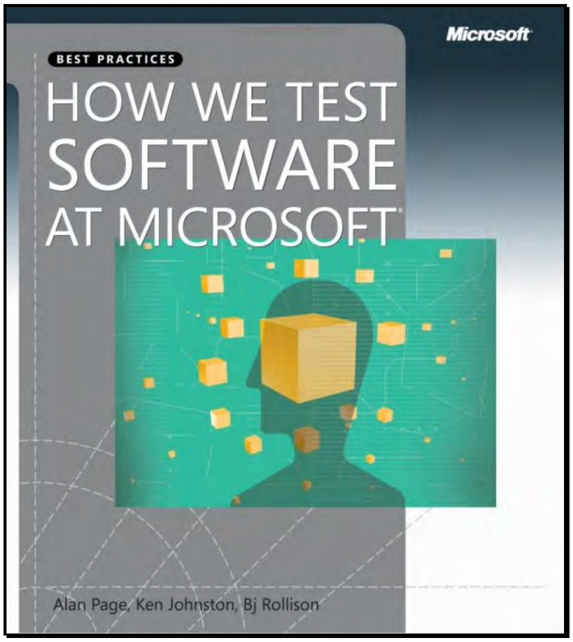
In fact, some of the tools are so easy and quick to use that they are being given to the developers so that they can performance test their code in their own environment before passing it over to the test team – after all no developer wants to deliver code containing defects – functional or non-functional.

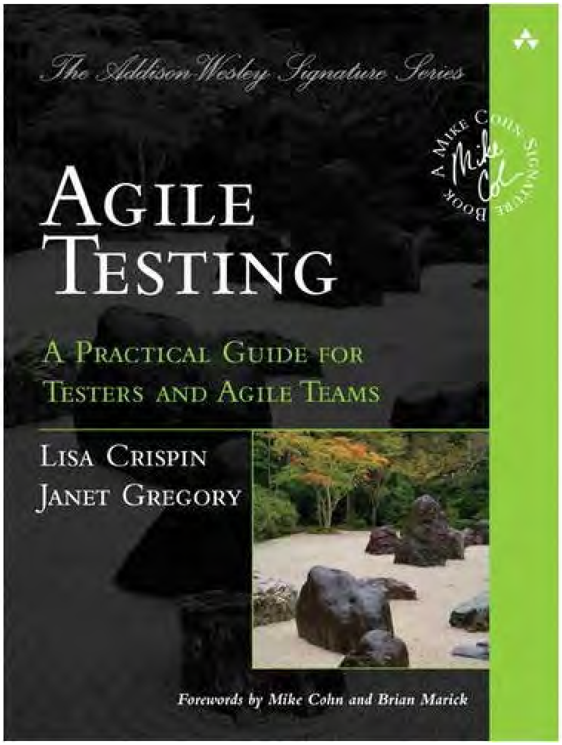
***Graham Parsons** is CEO of Reflective Solutions, a leading provider of performance testing and monitoring tools and services.(www.stresstester.com)*

# BORROWING A LIBRARY BOOK

Looking for a testing book but not sure which topics are covered? Or are you trying to decide which testing book to buy? Or do you simply want to increase your testing knowledge? If the answer to any of these questions is 'yes' then the BCS Software Testing Specialist Group Library could help!

The Library has lots of testing books covering a variety of topics and they are available to borrow for a period of 4 weeks - free of charge. Extended loans are allowed as long as the book has not been requested by another member.


Microsoft
BEST PRACTICES
HOW WE TEST SOFTWARE AT MICROSOFT
Alan Page, Ken Johnston, Bj Rollison


The Addison-Wesley Signature Series
A MIKE COHN SIGNATURE BOOK
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY
Forewords by Mike Cohn and Brian Marick

Topics include (amongst others) Requirements testing, Reviews/Inspections, Test Management, Test Techniques and Test Process Improvement.

If you would like to know more about the library and books available, or for any queries, visit…

http://www.bcs.org/server.php?show=ConWebDoc.11675

# SOFTWARE TEST ANALYST APPRENTICESHIPS: DEVELOPING THE TEST MANAGERS OF TOMORROW

**Andrew Gibbons, Profile Development and Training**

Speaking at the December meeting of the BCS Specialist Group in Software Testing, Les Hatton stated that he believed there are still too few testers in the software industry, and many others agreed. With the amount of software in existence increasing rapidly, and the complexity of the testing process increasing with the need to support multiple versions of products running on multiple platforms, the need for more testers is likely to increase further with time.

I believe that Software Test Analyst Apprenticeships offer a new way for organisations to develop testing resources. They provide a way in which test teams can “grow their own” test analysts, in their own image, and in a very cost-effective manner. I attended the last meeting of the BCS Specialist Group in Software Testing to ask whether professional testers and test managers felt that there is a role for a Software Test Analyst Apprenticeship in producing the test managers of tomorrow, and providing an additional testing resource today. I was pleased and encouraged by the positive response I got from many of the attendees. I felt that an article for The Tester, where I could give more details of the idea, was warranted.

The expected effect of rising university fees on the number of graduates of all kinds, including those with software-related degrees, is that there will be fewer in the coming years. Apprenticeships give an alternative entry point for young people with a desire to enter the Software Industry. They also allow them to earn as they are being trained.

Apprenticeships offer employers a highly cost-effective route to recruiting and training the next generation of employees – while apprentices get the balance of skills, knowledge and workplace experience they need to become competent IT professionals.

As employees, apprentices earn a wage and work alongside experienced staff to gain job-specific skills. Off the job, apprentices receive training to work towards nationally recognised qualifications. This training is delivered by a Training Provider such as a work-based training company or college of further education. The employer and the training provider work together to devise an apprenticeship programme that meets the employer’s needs and will provide a positive learning experience for the apprentice.

There is Government funding available to cover the full or part of the training costs, depending on eligibility. Anyone living in England, over 16 and not in full-time education can apply. Existing employees can also be put onto apprenticeship programmes.

Apprenticeships can take between one and four years to complete depending on the level of Apprenticeship, the apprentices’ ability and the industry sector. The minimum salary is £2.50 per hour; though many apprentices earn significantly more.

If you have a test team or department that would benefit from additional resources, but have limited funding to make this happen, I urge you to consider taking on an apprentice or two.  For further details, please contact apprenticeships@e-skills.com.

*After more than a decade working in software testing, **Andrew Gibbons** MSc CITP MBCS (andy.gibbons@profiledt.co.uk), is now a director of a work-based training company who deliver IT Professional Apprenticeships, amongst other things.*

# TESTING EVENTS CALENDAR 2011

**Belgium Testing Days**
http://www.belgiumtestingdays.com/
14 – 17 February 2011
Brussels, Belgium

**Czech Test**
http://www.czechtest.com/
9 - 10 March 2011
Prague, Czech Republic

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
16 March 2011
London, UK

**IEEE International Conference on Software Testing, Verification and Validation**
http://sites.google.com/site/icst2011/
21 – 25 March 2011, Berlin, Germany

**TAICPART**
http://www.taicpart.org/
25 March 2011
Berlin, Germany

**UK Test Management Forum**
http://uktmf.com/
27 April 2011
London, UK

**STAREAST**
http://www.sqe.com/stareast/
1 – 6 May 2011
Orlando, US

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
21 June 2011
London, UK

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
16 September 2011
London, UK

**STARWEST**
http://www.sqe.com/StarWest/
2 - 7 October, 2011
Anaheim, US

**EuroSTAR 2011**
http://www.eurostarconferences.com/
21 - 24 November
Manchester, UK

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
13 December 2011
London, UK

**BCS Scottish Testing Group**
http://www.bcs.org/server.php?show=nav.9729/
Spring / Autumn
Edinburgh or Glasgow, UK

# TESTING CONCURRENT SOFTWARE

**Mike Bartley, Test and Verification Solutions**

Hardware verification engineers have always faced the complexity of concurrent execution and temporal considerations when verifying hardware designs. However, silicon manufacturers are now moving to multicore designs (i.e. multiple CPU cores on a single chip) to achieve the relentless drive for improved performance at lower power now demanded by consumers. This pushes the responsibility for realizing that performance to the software community. Rather than simply relying on clock frequency and CPU design improvements to achieve their performance increase, software engineers will need to write their code to take advantage of the additional cores. Testing such software is fraught with potential issues and we will investigate just a few of them at the next SIGiST conference in March.

So why is testing concurrent software hard? The main issue is that concurrency introduces non-determinism so that running the same concurrent software twice is not guaranteed to give the same result each time due to different inter-leaving of threads of execution. For example, in Figure 1 below two threads share the variable “num”.

```
static int num = 0;

thread1 () {
 int val = num; // step 1
 num = val + 1; // step 3
}

thread2 () {
 int val = num; // step 2
 num = val + 1; // step 4
}
```

```
static int num = 0;

thread1 () {
 int val = num; // step 1
 num = val + 1; // step 2
}

thread2 () {
 int val = num; // step 3
 num = val + 1; // step 4
}
```

***Figure 1: Threads accessing shared code***

The two boxes demonstrate two different inter-leavings of those threads which lead to different values in “num”. You may feel that the thread should have been written as a “num++” but remember that will each get translated into three CPU instructions on the hardware: fetch to a register; increment that register; and then write the register value back to memory. This non-determinism can often mean that a test may fail on one run but then pass on the next - often referred to as a “heisenbug” (after the Heisenberg Uncertainty Principle). This makes finding bugs and then debugging even more difficult.

Race hazards also mean that if we execute the code once and it passes then it does not mean our job is necessarily done. We have now tested one particular interleaving of the threads and this interleaving worked. There may still be potential interleavings that fail and we simply haven’t run them yet. This means that our traditional models for measuring how well we have tested our code no longer work.

There are techniques to avoid the “race conditions” observed when running the code shown in Figure 1. The main technique is to make operations on the data “atomic”. This works by taking a “lock” on the data before operating on it. This means that no other thread can also work on the data in parallel and thus avoids the race conditions described above. Unfortunately however, this approach introduces its own new problems. For example, processes can deadlock if they are trying to both lock the same items of data.

```
Code for Process P      Code for Process Q
Lock(M)                 Lock(N)
Lock(N)                 Lock(M)
  Critical Section        Critical Section
Unlock(N)               Unlock(M)
Unlock(M)               Unlock(N)
```

***Figure 2: Potential for deadlock***

Remembering that the code in Figure 2 can potentially have different interleaving of execution so that process P could lock variable “M” and then process Q could lock variable “N”. They are then waiting on each other to release their locks to continue execution and no progress can be made. The conference will consider the conditions for creating deadlock and the other types of bugs that commonly when using locks on data. Using locks to make access to shared data atomic is not the only technique programmers use to share data. We will consider other mechanisms and the issues with using these at the conference too.

There have been very few studies into the numbers of bugs that concurrent programming introduces. However, one study indicated that about 10% of all the bugs written into the programs are caused by concurrency issues. So, working as a tester if you have just found 90 sequential bugs in the code then you still have another 10 concurrency bugs to find. And these could be the hardest 10% too! Why? Because in order to find these bugs you probably need to find ways to provoke different execution interleavings and there is no obvious way to do this. There are tools being developed and beginning to appear on the market to help this. These tools try to put different loads on the various CPU cores running the code to produce various execution traces. However, the major issue is that we do not know the loading profile of our user base. It is not unfamiliar for the code to be “exhaustively” tested only to hit a new bug during its first run on the target user system. Very embarrassing!

So, in this article we have very briefly seen that testing concurrent software will provide a number of issues for testers used to sequential models of execution. We will look at these issues in detail at the conference and also how testers can overcome them.

***Mike** has been involved in software testing and hardware verification for over 20 years. He started his career in testing of military software and safety-related aerospace applications using formal mathematical methods. He then moved into commercial hardware verification of a 64-bit MPEG4 chip at ST Microelectronics. From there he moved to Verification Manager at Infineon building up a team of over 35 verification engineers using state-of-the-art verification technology to verify numerous chips and design IP ranging from secure chip cards, through automotive applications to mobile phones. Mike then moved to start working with start-up companies in charge of both the testing of software products (tool chain, run-time libraries, applications, etc) and the verification of the hardware products - firstly at Elixent (now Panasonic) and then ClearSpeed. In these roles he established software testing and hardware verification teams (including offshore resources), flows and processes which were used to sign off numerous hardware/software products and are still in place today.*

*Mike now runs his own software testing and hardware verification consultancy helping with company strategies, benchmarking, project plan and execution reviews, training and providing teams both on-site and offshore. His customers’ software and hardware products cover a wide variety of application domains: safety-related automotive; mobile phones; low-power controllers; Ethernet routers; high-definition televisions.*

# EUROSTAR IS OVER ---
# I LOOK FORWARD TO EUROSTAR

**Peter Morgan, Software Tester**

On 26th January 1936, King George V died, and the traditional announcement was made: "The King is dead ...... Long Live the King!" The person had passed away but the position was immediately filled by the new holder. As with the changing monarch in the UK, so it is with the EuroSTAR conference. The EuroSTAR (2010) conference was hardly finished when attention moved on to EuroSTAR (the 2011 incarnation).

This year, SIGiST attendees not only have the conference on home soil (Manchester, 21st → 24th November 2011), but two of our 'own' on the Program Committee. Program Chair Geoff Thompson is joined by Graham Thomas (both BCS Software Testing Specialist Group committee members), Dutchman Derk-Jan de Grood and Morten Hougaard from Denmark. These four have the unenviable task of assembling a program that is suitable for beginner and experts alike, trying to balance subject matter, countries and companies, whilst including so-called hot topics.

Being in Manchester, there should be more Brits attending than when overseas travel is involved. In the UK we have traditionally provided a high proportion of EuroSTAR speakers (typically 20% - 25%), but not necessarily as high a proportion of general attendees. Perhaps the attendance trend will be reversed in 2011. There were over 900 people at EuroSTAR in Copenhagen, with over 150 Danes, enjoying their status as 'the home team'. Snow meant a late arrival home for me amongst others, but there will be no snow in Manchester for the 2011 event (Geoff Thompson guaranteed that on the main stage in Copenhagen). Personally, I am not sure how he can possibly know that!

Hardly had the delegates returned home from Copenhagen when the call for papers for Manchester was released. The conference theme is 'In Pursuit of Quality', with a closing date for submissions of 25th February, now not far away. One thing is certain; there will be a large number of submissions, with some from regulars (Brett Gonzales for sure) and others from first timers. One pointer for a good conference is to have lots of items submitted for consideration. And just because the Program Chair is a Brit does not mean that the program will be packed with Geoff's mates – the selection process is both clearly defined and transparent. It is very possible that the record for EuroSTAR conference proposals (standing at 432 for an individual conference) will be broken this year, all vying for "about 65" speaking slots.

So I first of all want to encourage you to ATTEND the conference in Manchester, and secondly to CONSIDER submitting an idea for a conference presentation. Full details of the submission process are given on the EuroSTAR website (http://www.eurostarconferences.com/speaker-zone/call-for-submissions-2011.aspx), together with hints and tips. If you have not spoken at EuroSTAR before, you could aim for a 20 minutes session (a 'mini-track'), which is limited to first time speakers at EuroSTAR. However, that does not necessarily mean that the competition for mini-track speaking slots is any less intense!

EuroSTAR embraces new things, both in terms of presentation ideas and other conference activities. EuroSTAR 2010 was the first conference I have attended with a significant on-line presence. The change was even more marked for me, as I had not attended the 2009 event. Not only were there plenty of people blogging and tweeting, but there were also several webinars output from the conference venue itself, and the Lee Copeland facilitated panel session was broadcast as it happened. On-line activity can be great for getting messages out quickly; the mere mention of something to say had Blog STAR Ajay Balamurugadas typing away; very useful when sessions had to be altered at short notice (two speakers snow-bound at Edinburgh airport).

*__Peter Morgan__ passed the first ISEB Practitioner Certificate software testing exam, in 2002. He worked for a testing consultancy and wrote their (old-style) 'Practitioner' course, teaching the Foundation level. A member of the ISEB accreditation panel, Peter is a hands-on tester, working on a free-lance basis within the UK. He has presented at EuroSTAR conferences, attending and occasionally speaking at the London SIGiST.*

Issue 37 THE TESTER

June 2011 Issue

## Next Conference: Tuesday 21st June 2011

Nathalie Van Delft

Neil Thompson

Andy Glover

Chris Ambler

Dot Graham

Stevan Zivanovic

Stuart Reid

# ‘What is testing?’

## Conference presentations

I was delighted to be asked to assume the role of programme secretary to the SIG, so I very much hope you enjoy the programme that has been arranged for you today. I would, however, like to express my thanks to the outgoing secretary, Graham Thomas. Graham has done a sterling job and left a legacy that I hope to match. ***Bernard Melson***.

We kick-off today’s ‘What is Testing’ programme with a keynote talk called ‘Adding Value’ from ***Chris Ambler***. In today’s market our stakeholders expect more from us as testers, so Chris’s talk about adding value through quality is topical.

We move on to a short networking session, following which we have two track sessions, the first from ***Neil Thompson*** on ‘The Science of Software Testing’ and the second from ***Natalie van Delft*** on ‘The Testing Ethics Debate’.

The afternoon starts on a light-hearted note with a short presentation called ‘A picture is worth a thousand words’ from ***Andy Glover***. Renowned for his use of cartoons to illustrate testing, we will have to wait and see what he gives us.

We move on to a rapid-fire set of four presentations on the ‘What is’ theme, with:

- ***Dot Graham*** and ‘What is Coverage’?
- ***Neil Thompson*** and ‘What is Risk’?
- ***Natalie van Delft*** and ‘What More is Testing’?
- ***Stuart Reid*** and ‘What is a Testing Professional’.

Keeping with the current trend and interest in Agile, ***Stevan Zivanovic*** will be presenting his views on ‘Leadership in an Agile Context’. Finally, we close the day with ***Dot Graham’s*** eye-opening talk about ‘What test managers think they know about test automation - but don't’. Should be interesting!

## In this issue

### CONFERENCE PROGRAM

### HOW TO

### ARTICLES

### TESTING EVENTS

# From The Editor

**Matt Archer, Editor**

Our first article this month comes from **Chris Ambler** – Managing Director of Testing Stuff**.** In his article, '**Adding value**', Chris explores the skills a tester needs to add value to a project, including the ability to innovate, solve problems and be the guardian of quality. As you read Chris' article, ask yourself, "how well do 'I' add value by exhibiting those skills?"

Our second article has been written by **Stevan Zivanovic.** In his article, '**Leadership in an agile context**', Stevan highlights the opportunities testers have within an agile project to lead, and offers some good advice to anyone who finds themselves in a leadership position.

Our third article has been written by **Neil Thompson** of Thompson Information Systems Consulting Ltd. In his article, '**The science of software testing**', Neil provides us with an insight into how testers all over the world have been using existing sciences as the basis to forward their thinking about testing. Neil will be presenting on the same topic at our upcoming conference. For me, it's one not to be missed.

If you have been inspired by any of the articles in this edition and would like to write an article for *The Tester* yourself, then please feel free to email me.

Matt Archer

The Tester Editor<br>
BCS Specialist Group in Software Testing<br>
matthewjarcher@googlemail.com

# LinkedIn and Twitter

The BCS Software Testing Specialist Group is now using social media platforms to improve communications both to members and between members.

Our LinkedIn Group (link below) will carry details of our conferences as they become available. It will also provide a place where people can discuss testing topics, make requests about future conferences, find employment opportunities (there are a few jobs advertised already) and generally keep up to date with our chosen industry. If you are already a member of LinkedIn then simply visit the group and make a request to join.

If you're not a member then go to http://www.linkedin.com/ to create an account.

If you use Twitter you can follow us @SIGiST.

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

# Web Links

**SIGiST conference website:**
**www.SIGiST.org.uk**

**Standards Working Party:**
**www.testingstandards.co.uk**

**LinkedIn Page:**
**http://www.linkedin.com/groups?mostPopular=&gid=3466623**

# Conference Booking Instructions

**To register and pay online, please use the link below.**

**The new BCS booking system accepts multiple and third party bookings:**

**https://events.bcs.org/book/8/**

**If you have a query relating to making a booking, please contact Gemma Stanley-Evill, Specialist Groups' Officer.**

**Tel: (01793) 417656**

**gemma.stanley-evill@hq.bcs.org.uk**

# CONFERENCE AGENDA

## 'What is testing?'

Tuesday 21st June 2011
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

| Time | Session | Length |
|---|---|---|
| **08:30** | **Registration open, Coffee and Exhibition Hall** | |
| **09:25** | **Welcome & Introduction**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |
| **09:30** | **OPENING KEYNOTE: Adding Value**<br>**Chris Ambler** | **60** |
| **10:30** | **Networking Session** | **15** |
| **10:45** | **Tea / Coffee Break Exhibition Hall and Networking** | **30** |
| **11:15** | **The Science of Software Testing**<br>**Neil Thompson** | **45** |
| **12:00** | **The Testing Ethics Debate**<br>**Nathalie Van Delft** | **45** |
| **12:45** | **Buffet lunch, Exhibition Hall and Networking** | **75** |
| **14:00** | **A picture is worth a thousand words**<br>**Andy Glover** | **15** |
| **14:15** | **Lightning Talks - What is Software Testing?**<br>• **Dot Graham** - What is Coverage?<br>• **Neil Thompson** - What is Risk?<br>• **Natalie Van Delft** - What More is Testing?<br>• **Stuart Reid** - What is a Testing Professional? | **45** |
| **15:00** | **Leadership in an Agile Context**<br>**Stevan Zivanovic** | **45** |
| **15:45** | **Tea / Coffee Break, Exhibition Hall and Networking** | **30** |
| **16:15** | **What test managers think they know about test automation - but don't**<br>**Dot Graham** | **45** |
| **17:00** | **Closing Remarks**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |

# ABSTRACTS AND BIOGRAPHIES

## OPENING KEYNOTE: Adding Value

### Chris Ambler, Testing Stuff

Quality is expected but no one totally understands how or why or even what quality really is. It's also safe to say that everyone is a tester in their everyday lives so does testing actually exist as a profession? This presentation explores what a tester really is and how they add value. Are we there to innovate, report on the findings or interpret the results? Are we really the guardians of quality or are we there to just find bugs? Also, the presentation explains what sort of person makes a good tester and what attributes are necessary to carry out the necessary tasks to add business value. This presentation explores these questions and comes up with an answer to the legendary questions - 'what is quality?' and 'what do we as testers really do to influence it?'

*As Managing Director of Testing Stuff* ***Chris*** *has a testing career spanning over 28 years in defence, testing combat systems, test management and consultancy working with various banks, insurance companies and the likes of Seeboard, TFL, Lombard, Home Office, Sky Television and Maersk. Spending over 7 years in the video games industry as the European QA Director for Electronic Arts and as the European Test Manager for Microsoft Game Studios he believes in testing at the right price and leveraging the latest techniques. Chris regularly speaks at conferences worldwide.*

## The Science of Software Testing

### Neil Thompson, Thompson information Systems Consulting Ltd

The "Art" of Software Testing is a book, and there are two with "Craft" in the title, but when will we achieve Science? This presentation builds on the few blog posts etc which (so far) mention analogies between testing and scientific philosophy & methods. By understanding conjectures, hypotheses, models & theories we should test better, using heuristics, patterns & techniques for appropriate coverage and to mix pre-designed with exploratory, and mix tests with "checks". It's not just "experiments" to test correctness, it's also experiments to test bugginess and to uncover quality information.

2nd, I suggest new insights based on evolution: not just Darwinian-biological but Dennett's notion that all science is evolving – from physics through inorganic then organic chemistry, thence through biology to social sciences. An analogy of software lifecycles (lean or otherwise) with this "value flow" can inspire more creative and focussed unit, integration, system & acceptance testing, as part of a broader quality view. I recap a simple table-diagram thinking tool called a Value Flow ScoreCard (very briefly, as this was already presented at SIGiST Sep 2008). This incorporates Goldratt and Systems Thinking.

3rd, by combining principles of "emergence" with evolution, I see quality improvement in terms of increasing sophistication plotted against increasing diversity – this is Value Flow Science. Software technology & development have progressed in both these dimensions faster than has testing, so far. Arguably emergence proceeds along a path balanced between too much order and too much chaos (Kauffman). So, as software becomes increasingly pervasive & evolves towards greater "artificial intelligence", testing & quality need to innovate appropriately. The philosophy of science has itself already evolved beyond the cliché of Popper's falsifiability

4th, I offer another useful analogy from evolutions: as genes control biological evolution, “memes” arguably control mental, social & cultural evolution. Memes are more powerful than genes, leveraging language, mathematics & other formal sciences, then technology. These constitute platforms which facilitate step-change innovation. For example, no wheeled animals have yet evolved, but humans have evolved to invent inorganic wheels. In testing & quality, we have a number of existing concepts – test levels, types, techniques etc – but how do we know we are not missing something useful? I offer some suggestions to steer testing in appropriate future directions, eg things we could adapt from other disciplines such as insurance and AI.

***Neil Thompson*** *has 33 years' experience so far in information systems, currently specialising in testing and quality consultancy and management (since 1992). Earlier roles included programmer/tester, analyst, project manager and management consultant. Has worked for a hardware manufacturer, software houses, user organisations and consultancies. Co-author with Paul Gerrard of the book Risk-Based E-Business Testing (2002). University degree is in Natural Sciences.*

## The Testing Ethics Debate

### Nathalie Van Delft, Capgemini

Testing is more than just the activity of testing, it’s also all the things around testing, like behaviour and professional ethics.  This session is a hosted “House of Commons” debate on the topic ‘Ethics within the testing profession’. In this 45 minute session several stimulating statements are given which is guaranteed to lead to a meaty discussion and push (y)our tester’s morale to its limits!

The hosted ‘House of Commons’ debate is derived from a popular Dutch television show which goal it was to bring the actualities within politics to the common’s living room. The statements are introduced by the hosts. One host will be in favor of the statement and the other host will be against the statement. The audience will then have the opportunity to react on the made pleas. The host will encourage the audience to take their side and philosophize on the chosen point of view. At the end of the session a ‘best debater’ will be appointed!

Do you already want to think about your points of view or get in the mood for this discussion? Here are some prepared statements:

- You can break the law to meet your test goals
- You must always tell the truth
- You must always be able to use Privacy sensitive data to test
- You can drop test cases (based on your own insight) if necessary, to be ready on time
- As a tester, you should not have a professional code of conduct (or ethics), because it hinders good testing
- Testing and ethics don’t mix

Do YOU dare to let go of your normal standards and philosophize what could be when one is not so honest? Are some statements really that black and white? And what are those nuances to be applied? Share your opinion and perhaps we will arrive at the basis of the “Testers code of conduct”!

***Nathalie*** *Rooseboom de Vries van Delft is Expert Group Leader of the cluster testing technologies and processes and CoP Testing Lead at Capgemini, responsible for thought leadership and testing competence development.  She fulfills the roles of test manager and –advisor with various clients. She speaks on national and international test events on regular basis, writes in specialist publications and participates in the Dutch Standardization Body (NEN) workgroup* *for Software- and System development. She is very passionate about (software) testing in general, but the subjects Data Warehouse Testing, Chain testing, Standardization, Ethics/ Philosophy and Test Architecture (Framework) are most favorite.*

## A picture is worth a thousand words

### Andy Glover, Cartoon Tester

Describing testing to a project manager or user can be difficult at times, but have you ever tried to explain software testing with a painting, a photo, or even a cartoon?

In this fun and interactive presentation, Andy will try to describe software testing with simple drawings and images. By using lateral thinking techniques and a bucket load of creativity (arguably essential characteristics for a software tester) you will come out of the presentation with a picture worth a thousand words!

***Andy** is a Test Manager at a small development house developing mobile phone applications for clinical trials. He's been testing around 9 years mostly in the pharmaceutical industry. In his spare time, Andy designs and draws cartoons about software testing which are printed in The Testing Planet and are posted on his blog: cartoontester.blogspot.com.*

## Leadership in an Agile Context

### Stevan Zivanovic, Experimentus

*The critical and key differentiator with Agile as a methodology over other development methodologies is the primary focus on people – your colleagues, your customers and yourself. Just look at the manifesto and principles. You are empowered and expected to take individual responsibility for your actions, behaviour and skills and contribute actively to the team. The actions, behaviours and skills include "soft" skills as well as technical.*

*At various points in a project, you will need to take the lead. This presentation will focus on what actions you can take to provide good leadership. It will prompt you to identify what your core experience is and how you can apply this in the context of your work. By demonstration leadership at the appropriate time and taking individual responsibility to be the best leader you can be, you can influence change for the better.*

***Stevan** has been actively involved in the Quality arena for over 18 years, where he has worked in a wide variety of businesses, from Safety Critical, to financial, to "Dot Coms". He has a pragmatic approach and has a proven track record of implementing realistic, workable, real world changes for large, multinational teams. Stevan has presented at EuroSTAR, BCS, UK Test Managers Forum and Agile Business Conference, were his presentations were well received. He enjoys speaking and training people to motivate them in testing and quality.*

# What test managers think they know about test automation - but don't

## Dot Graham, software testing consultant

Recent research has highlighted the critical role that managers play in the success or failure of test automation.

Without adequate management support, well-intentioned initiatives are doomed to be short-lived and guaranteed to waste time and money.

But even extensive management support, without sufficient knowledge of the issues, may fail to produce results with lasting and growing benefits to the organization. It is a serious mistake to think that automation is just a technical issue that doesn't need management involvement.

In this talk, Dorothy Graham outlines five common management misconceptions about automation and how to ensure that you have realistic expectations. Understand why objectives and measurement are critical for test automation and how automation objectives area different to test objectives, how to resource test automation efforts, how to measure Return on Investment, and more. Know what issues to look out for as early indications of problems, and how to get back on track to successful automation.

- understand management's role in successful automation and how to guide for success
- avoid common mistakes and common misconceptions about test automation
- know what to monitor and how to get back on track

***Dorothy Graham** has been in testing for over 30 years, and is co-author of 3 books (Software Inspection, Software Test Automation and Foundations of Software Testing). Her latest book, Experiences of Test Automation, co-authored with Mark Fewster, contains case studies of successful (and some unsuccessful) test automation in practice, and will (hopefully) be published in 2011.*

*Dot was the program chair for the EuroSTAR Conference in 1993 and 2009. She has been on the boards of conferences and publications in software testing. She was a founder member of the ISEB Software Testing Board and was a member of the working party that developed the first ISTQB Foundation Syllabus.*

*She is a popular speaker at conferences and seminars world-wide and holds the European Excellence Award in Software Testing.*

## Adding Value

'Quality' is expected by everyone but no one really understands what it is or how we really achieve it. Everyone is constantly testing and looking for quality but doesn't realise it. My definition of quality is something like:

> *'A tangible (or non-tangible) measure that needs to be objectively and pragmatically made to determine the personal and/or business fulfillment of an outcome using personal experience and if possible, the experience and feedback of others'.*

So what are testers and do they actually exist? Asking 20 testers what they do would give you 20 different answers so let's look at the sorts of things testers might do:

***Are we innovators***? Innovation is a very important part of being a tester. We need to create and design testing processes and structures to ensure that our products meet the specifications of our customers. Only about 16% of the population are described as innovators or early adopters which are traits that are necessary in a tester. Along with innovation around process and structure, we also create the necessary safeguards for keeping the customers happy once the system is in use and continuously developed.

***Are we reporters***? We need to provide information based on collected data that is relevant to the recipient. As testers we need to be aware of what our customers need to be successful. It is important that we can deliver this information on progress, statistics and bug information in a formal and unemotional way. We should view this as collecting *data*, turning the data into *information*, giving the information to the customer to create *knowledge* that allows them to make a *decision*.

***Are we interpreters***? Everyone is working towards a common goal. Our testing stakeholders include business people, developers, customers, managers and technical people. Our job as testers is to interpret everyone's version of the common goal and try to create and report on a jointly developed thread.

**Are we problem solvers**? the perception is that this is the main role of a tester. Analysis of issues is an important part of what we do as testers but you could argue that that we are problem finders - not problem solvers.

**Are we the guardians of quality**? Everyone should care about and impacts quality. All stakeholders have a responsibility to ensure quality is achieved but maybe testers hold the keys to ensure it is objectively measured and delivered.

I think it's safe to say that we do all of these things but I'm not sure we are recognised for these abilities. For other people to understand what testers do - its really important that WE know what we do so make sure you tell everyone!!

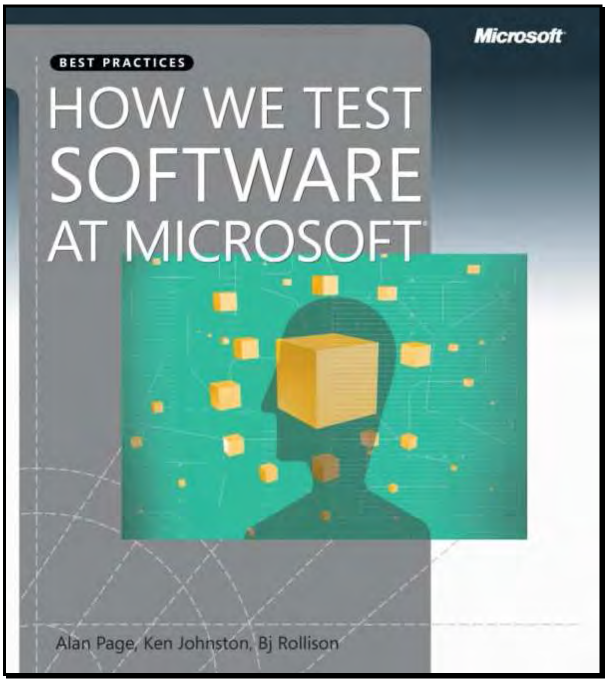
Ted says:
*'Quality is in the eye of the beholder. But the beholder is really important!!'*

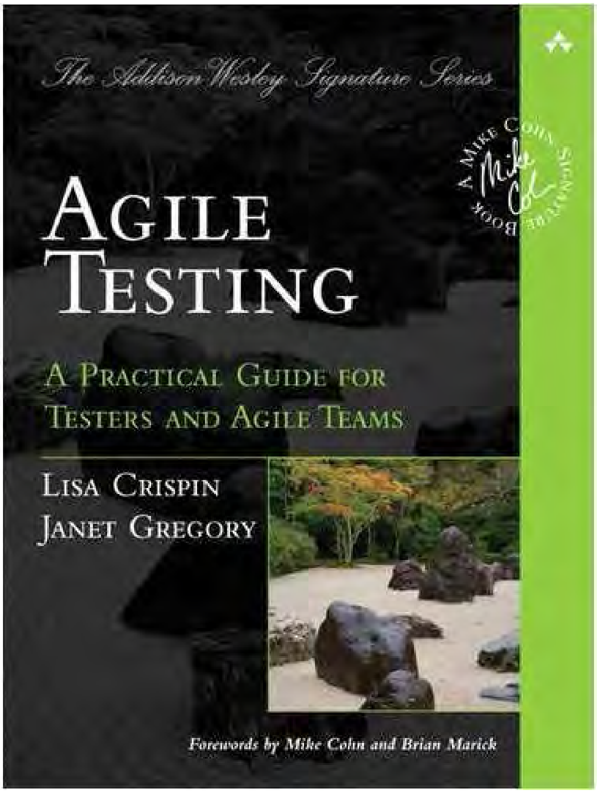
As Managing Director of Testing Stuff, Chris has specialised in software testing for nearly 30 years. His experience spans a huge variety of projects in many industries including banking, insurance, defence, telecoms and transport. Most recently he set up Microsoft Game Studio's European video games testing division and prior to that he was the European QA Director at Electronic Arts. Chris is a highly regarded thought leader in the industry and is a regular keynote speaker at international conferences in software testing and video games development. Committed to quality at the right price, Chris is dedicated to creating a professional approach to video games testing.

# Borrowing A Library Book

Looking for a testing book but not sure which topics are covered? Or are you trying to decide which testing book to buy? Or do you simply want to increase your testing knowledge? If the answer to any of these questions is 'yes' then the BCS Software Testing Specialist Group Library could help!

The Library has lots of testing books covering a variety of topics and they are available to borrow for a period of 4 weeks - free of charge. Extended loans are allowed as long as the book has not been requested by another member.


Microsoft
BEST PRACTICES
HOW WE TEST SOFTWARE AT MICROSOFT
Alan Page, Ken Johnston, Bj Rollison


The Addison Wesley Signature Series
A Mike Cohn Signature Book
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY
Forewords by Mike Cohn and Brian Marick

Topics include (amongst others) Requirements testing, Reviews/Inspections, Test Management, Test Techniques and Test Process Improvement.

If you would like to know more about the library and books available, or for any queries, visit…

http://www.bcs.org/server.php?show=ConWebDoc.11675

# LEADERSHIP IN AN AGILE CONTEXT

**Stevan Zivanovic, Experimentus**

The critical and key differentiator with Agile as a methodology over other development methodologies is the primary focus on people – your colleagues, your customers and yourself. Just look at the manifesto and principles.  You are empowered to take individual responsibility for your actions, behaviour and skills and contribute actively to the team. The actions, behaviours and skills are more than just technical. Your skills may be in; being able to resolve conflicts of interest, listen critically, re-iterate ideas in a different fashion. It is not just your ability to create a $10^{10}$ "all pairs" matrix and rationalise it into a prioritised set of test cases.

Within this context it may seem strange to talk about leadership. You may be asking "Surely as a team, everyone is equal?" The answer to this that everyone has an equal right to be there, to participate, to be heard and to be valued. However each team member will be bringing in their unique and highly valued skills. When these skills are demanded, it is up to that individual to lead. At this point it is worth identifying what I mean by leadership. From the Oxford English dictionary, the key, relevant and primary definition of the verb "to lead" is:

> *"To cause (a person …) to go with one by holding them by the hand (…etc.) while moving forward"* {http://www.oxforddictionaries.com/definition/lead?rskey=nlDegY&result=1#m_en_gb0461360}

The analogy to an Agile project is so obvious, however I think it only presents part of the answer. How does this relate to me as an individual in an Agile project? The following is taken from the pre-course notes for a leadership course at MIT University (July 2010):

> "*Being a leader and the exercise of leadership is all about realising a future that wasn't going to happen anyway*."

In the context as a specialist with a testing role in an Agile team, this could translate as: "As a leader in testing within this team, I will realise a future outcome where the most efficient and effective tests are applied in the appropriate time by the team."

The skills in being an effective leader stem from you. Leadership as a skill can be taught and even the greatest leaders that you can think off have learnt these skills. The key element is that you want to and believe that you deserve to lead. Based on this, there are three areas that you need to investigate, namely; your integrity, your authenticity and that your cause is larger than just your own aims. Note that none of this about others giving you a role, job specification, title. This inferred authority actually makes it harder to lead (see J Burger, American Psychologist, Vol 64(1), Jan 2009, 1-11), as it can produce the action of obedience from others. Obedience is not what an Agile team is about – an Agile team needs active and equal cooperation by all participants.

The skills in leadership are varied and there is significant reference material available. Just a search on Amazon will show you. The request from this article and the presentation is that you take individual responsibility to be the best leader you can be and to actively practice it within your teams.

*__Stevan__ has been actively involved in the Quality arena for over 18 years, where he has worked in a wide variety of businesses, from Safety Critical, to financial, to "Dot Coms". He has a pragmatic approach and has a proven track record of implementing realistic, workable, real world changes for large, multinational teams. Stevan has presented at EuroSTAR, BCS, UK Test Managers Forum and Agile Business Conference, were his presentations were well received. He enjoys speaking and training people to motivate them in testing and quality.*

# TESTING EVENTS CALENDAR 2011

**Software Testing Club Meetup**
http://www.softwaretestingclub.com/
1 June 2011
Bristol, UK

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
21 June 2011
London, UK

**UK Test Management Forum**
http://uktmf.com/
27 July 2011
London, UK

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
16 September 2011
London, UK

**STARWEST**
http://www.sqe.com/StarWest/
2 - 7 October, 2011
Anaheim, US

**UK Test Management Forum**
http://uktmf.com/
26 October 2011
London, UK

**EuroSTAR 2011**
http://www.eurostarconferences.com/
21 - 24 November
Manchester, UK

**SIGiST**
http://www.bcs.org/server.php?show=nav.9264
13 December 2011
London, UK

**BCS Scottish Testing Group**
http://www.bcs.org/server.php?show=nav.9729/
Spring / Autumn
Edinburgh or Glasgow, UK

# THE SCIENCE OF SOFTWARE TESTING:

## EXPERIMENTS, EVOLUTION & EMERGENCE VIA VALUE FLOW

**Neil Thompson, Thompson information Systems Consulting Ltd**

The “Art” of Software Testing is a book, and there are two with “Craft” in the title, but when will we achieve Science? This article (see associated presentation for more details) seeks to move forward from those few sources I have been able to find which (so far) mention analogies between software testing and scientific philosophy & methods.

The subject gets a passing mention in some textbooks: Boris Beizer referred to (1984) an experimental process in performance testing and (1995) falsifiability of statements & objects. Rick Craig & Stefan Jaskiel described (2002) black-box science & art and “white-box” science (no art in white-box?). Marnie Hutcheson examined (2003) some “myths” about art, science & software, before dissecting the engineering approach.

Individually, and collectively [1], Kaner, Bach and Pettichord have made several points on this:

- “that the software works” can be treated as a theory, and testers design experiments to falsify that;
- testers should behave empirically and think sceptically, recognising the limitations of “knowledge”;
- testing is grounded in cognitive psychology, and needs inference, conjecture & refutation, not just comparison of output to expected results;
- all testing is based on models;
- more recently, [2], testing as a “social science”: software is made for people to use within societies, so metrics should be aimed at stakeholders comprehensively; whereas bad models and reliability theories give blind spots and impede trade-offs. Verification & validation need to be clearly distinguished.

And in articles and blogs by other authors:

- Paul Carvalho has added that testing skills include learning or relearning scientific method (emphasis on multiple sources) and knowledge of probability and statistics.
- Randy Rice [3] recognised that some recent scientists have “less rigorous” methods, but chose to stick with the rigorous view that the scientific method has steps which are analogous to those of software testing, and he examined the associated terms – observation, experiment, hypothesis, assumption, fact, and law. He also highlighted the “longer view”, ie process improvement.
- David Coutts [4], while acknowledging openness to context-driven ideas, took as his basic position that both science & software testing have right or wrong answers, so a test either passes or fails based on its requirements. He did however go beyond falsificationism, quoting extra criteria for science listed by Lewis Wolpert: economy (few theories/laws explaining many observations), consistency (both internal & external), mathematical foundation, and independent verification. He also distinguished retesting as a different theory, and noted references to evolution and memes (see my paragraphs below).
- BJ Rollison [5] started with falsificationism but objected that there are too many hypotheses to test in a reasonable time. He brought in the concept of risk, and also applied science to debugging.

However, I wish to re-evaluate these positions and hopefully proceed further – notably to question what kind of experiments testers are / should really be running, and to acknowledge that “the scientific method” is itself a controversial area and has already evolved beyond the Karl Popper view that several testers have quoted. Software testing should also look more to the future than it arguably has to date.

### Experiments

By understanding conjectures, hypotheses, models & theories we should become able to test better, using heuristics, patterns & techniques for appropriate coverage and to mix pre-designed with exploratory, and mix tests with “checks” (see http://www.developsense.com/blog/). It’s not just “experiments” to falsify correctness, it should also be experiments to test bugginess in different specific ways (starting with existing techniques but going on to combine and innovate techniques); plus experiments to derive information about quality.

### Evolution and Value Flow ScoreCards

I suggest new insights based on evolution: not just Darwinian-biological but Dennett’s [6] notion (originated by David Hull?) that all science is evolving – from physics through inorganic then organic chemistry, thence through biology to social sciences. An analogy of software lifecycles (lean or otherwise) with this “value flow” can inspire more creative and focussed unit, integration, system & acceptance testing, as part of a broader quality view. I developed with Mike Smith a simple table-diagram thinking tool called a Value Flow ScoreCard which can be used in many ways, eg balancing multi-dimensional test coverage, setting test policy/strategy, and process improvement/definition. The latter uses incorporate Goldratt and Systems Thinking.

### Emergence and Value Flow Science

By combining principles of “emergence” with evolution, I see quality improvement in terms of increasing sophistication plotted against increasing diversity – this is Value Flow Science. Software technology & development have progressed in both these dimensions faster than has testing, so far. The “punctuated equilibria” phenomenon of the fossil record has parallels elsewhere. Arguably emergence proceeds along a path balanced between too much order and too much chaos [7]. So, as software becomes increasingly pervasive & evolves towards greater “artificial intelligence”, testing & quality need to innovate appropriately. Just as the philosophy of science has evolved to embrace Bayesian principles [8], so should software testing.

### Platforms and Cranes – Genes to Memes

I offer another useful analogy from evolutions [6 again]: as genes control biological evolution, “memes” arguably control mental, social and cultural evolution. Memes are more powerful than genes, leveraging language, mathematics & other formal sciences, then technology. These constitute platforms which facilitate step-change innovation. For example, no wheeled animals have yet evolved, but humans have evolved to invent inorganic wheels. In testing & quality, we have a number of existing concepts – test levels, types, techniques etc – but how do we know we are not missing something useful? There are currently schisms between (claimed) different “schools” of software testing. I offer some suggestions to steer testing in appropriate future directions, eg things we could adapt from other disciplines:

- The insurance industry is founded on risk management, and they have specialists who manage the financial impact of risk (and uncertainty!) – they are called actuaries. Testers should learn from what actuaries think and do.
- A few years ago, we heard about genetic algorithms in IT, but that seems to have gone quiet – these now deserve a renaissance in the broader evolutionary context, and are of particular interest to testing if it is ever to be able to keep up with the ongoing progress of information systems towards AI.

### Messages to take away (these are just a selection)

When you are strategising, planning and doing testing:

- test according to your scale, using analogies from different sciences to help “frame” [9] your tests.

When considering your position and future in the testing industry:

- try to understand apparent values and behaviours of others in terms not just of “what they have been taught about testing”, but also their personalities and what their bosses say they want (or should want!);
- think of innovations as potential “platforms” from which new methods may become feasible, eg using automation in an exploratory rather than merely confirmatory way.

In your wider life:

- when reading new material, consider reading two authors at once (or maybe three) – either different representations of similar opinions, or apparently opposing opinions. This uses the idea that much innovation comes from synthesising the “adjacent possible” [7 again].

## Key References

[1] Cem Kaner, James Bach & Bret Pettichord: **Lessons Learned in Software Testing** – a Context-Driven Approach (Wiley 2002)
[2] Cem Kaner, **Software Testing as a Social Science** (http://www.kaner.com/pdfs/KanerSocialScienceSTEP.pdf 2004 & later)
[3] Randall W. Rice: **The Science of Software Testing** (http://www.riceconsulting.com/articles/science-of-software-testing.htm )
[4] David Coutts: **The Test Case as a Scientific Experiment**

(http://www.stickyminds.com/sitewide.asp?ObjectId=8965&Function=edetail&ObjectType=ART 2005)
[5] B.J. Rollison: **The Science of Software Testing** (http://blogs.msdn.com/b/imtesty/archive/2007/01/15/the-science-of-software-testing.aspx 2007)
[6] Daniel C. Dennett: **Darwin’s Dangerous Idea** – Evolution and the Meanings of Life (Simon & Schuster 1995)
[7] Stuart Kauffman: **Investigations** (Oxford University Press 2000)
[8] Peter Godfrey-Smith: **Theory and Reality** – an Introduction to the Philosophy of Science (University of Chicago Press 2003)
[9] Michael Bolton & James Bach: **Test Framing** (http://www.developsense.com/resources/TestFraming.pdf 2010)

***Neil Thompson*** *is Director of Thompson information Systems Consulting Ltd, which currently specialises in testing/quality strategy, management and process improvement. Neil has 33 years' experience so far in information systems, roles including programmer/tester, analyst, project manager and management consultant. He has worked for a hardware manufacturer, software houses, user organisations and business consultancies. Co-author with Paul Gerrard of the book Risk-Based E-Business Testing (2002). University degree is in Natural Sciences. Neil has spoken at various testing conferences since the first EuroSTAR in 1993.*

September 2011, Edition 38

Specialist Group in Software Testing

# THE TESTER

## IN THIS ISSUE

## Next Conference: Friday 16th September 2011

**Fred Beringer**

**Ian Gilchrist**

**Mark Fewster**

**Derk-Jan de Grood**

**Paul Gerrard**

**Peter Morgan**

After the obligatory AGM session at the start of the day, the real business launches with the keynote session from Paul Gerrard about “Using Business Stories to Test Requirements and Systems”. Paul will look at how human beings have used stories for millennia and how they are a natural and easy way to share understanding of requirements. They are also a simple method for walking through scenarios so that anomalies and gaps stand out and this method can be used for requirements review.

After coffee break, Mark Fewster will talk about “Experience Driven Test Automation”, using case studies from his new book, co-written with Dot Graham, which is to be published later this year. Mark will illustrate some common themes on what works and doesn’t work with approaches to test automation. Next, Fred Beringer looks at how we can use cloud computing to help us test performance in a live-like environment, with realistic conditions such as firewalls and network latency.

Our short session after lunch will be presented by Peter Morgan who will review ‘Agile Testing, A Practical Approach”, written by Lisa Crispin and Janet Gregory.

The afternoon tracks start with Mieke Gevers’ “Ever Been Fooled by Performance Testing Results?” Mieke will look at some of the problems with evaluating performance testing results, and using some case studies, presents some tips on how to spot patterns in your results.

Next “Test Automation: how far should it go?” Has the drive for automation gone too far? Have we lost sight of the purpose of testing? Should we ever aim for ‘Full Automation’? These are some of the key questions that Ian Gilchrist will be discussing in his session.

Finally, in our closing keynote, Derk-Jan de Grood tells us how to “Go Sleuthing with the Right Test Techniques”. Derk-Jan shares his insights on test technique selection and bemoans the fact that so many books and testing courses teach the techniques but don’t give any guidance on how to choose the best technique for the job.

Mo Shannon, Deputy Programme Secretary

# Announcements

## SIGIST Conference Booking Instructions

To register online, please use the link below. Please note, the new BCS booking system accepts multiple and third party bookings:

https://events.bcs.org/book/8/

If you have a query relating to making a booking, please contact Gemma Stanley-Evill, Specialist Groups' Officer.

Tel: (01793) 417656

gemma.stanley-evill@hq.bcs.org.uk

## LinkedIn & Twitter

The BCS Software Testing Specialist Group is now using social media platforms to improve communications both to members and between members.

Our LinkedIn Group (link below) will carry details of our conferences as they become available. It will also provide a place where people can discuss testing topics, make requests about future conferences, find employment opportunities (there are a few jobs advertised already) and generally keep up to date with our chosen industry. If you are already a member of LinkedIn then simply visit the group and make a request to join.

If you're not a member then go to http://www.linkedin.com/ to create an account.

If you use Twitter you can follow us @SIGiST.

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

## Four Top Talks on Software Testing in Bristol

The BCS Bristol Branch, in collaboration with the Software Testing Club, have organised four top software testing speakers.

- Mon 19th Sept 2011, Effective Specifications & Tests for Agile Projects, Gojko Adzic
- Mon 5th Dec 2011, Using Business Stories to Test Requirements and Systems, Paul Gerrard
- Mon 13th Feb 2012, Lessons from Experience in test automation, Dot Graham
- Wed 9th May 2012, Agility and Risk : Challenges for your Agile QA Strategy, David Evans

**Talk will be held at City of Bristol College, Bristol BS1 5UA Refreshments are served at 7.00pm for a 7.30pm start**

**Gojko Adzic** is a strategic consultant who helps ambitious teams to improve the quality of their software products and processes. He discusses how to organise requirements, specifications and tests effectively to support an agile development process.

**Paul Gerrard** specialises in test assurance. Paul will show us how a disciplined approach to story-writing and requirements testing can improve requirements and the target solution dramatically.

**Dorothy Graham** has been in testing for 40 years, and is co-author of 4 books on test automation and will answer the question "What is happening now in test automation?"

**David Evans** is an independent agile consultant, coach and trainer with over 22 years of IT experience. He will take us on a tour through the key challenges for creating an agile quality strategy.

Speaker details and the abstracts for these talks can be found here;

www.tandvsolns.co.uk/files/SWTestTalksBristol.pdf

# Conference Agenda

Friday 16th September 2011
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

| Time | Session | Length |
|---|---|---|
| **08:30** | **Registration open, Coffee and Exhibition Hall** | |
| **09:10** | **Welcome & Introduction**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |
| **09:15** | **BCS SIGiST AGM** | **15** |
| **09:30** | **OPENING KEYNOTE:**<br>**Using Business Stories to Test Requirements and Systems**<br>**Paul Gerrard** | **60** |
| **10:30** | **Networking Session** | **15** |
| **10:45** | **Tea / Coffee Break Exhibition Hall and Networking** | **30** |
| **11:15** | **Experience Driven Test Automation**<br>**Mark Fewster** | **45** |
| **12:00** | **The Cloud: A Game-Changer for Web Performance Testing**<br>**Fred Beringer** | **45** |
| **12:45** | **Buffet lunch, Exhibition Hall and Networking** | **75** |
| **14:00** | **Book Review – Agile Testing, a Practical Guide**<br>**Peter Morgan** | **15** |
| **14:15** | **Ever Been Fooled by Performance Testing Results?**<br>**Mieke Gevers** | **45** |
| **15:00** | **Test Automation – How far should it go?**<br>**Ian Gilchrist** | **45** |
| **15:45** | **Break, Exhibition Hall and Networking** | **30** |
| **16:15** | **CLOSING KEYNOTE:**<br>**Go Sleuthing with the Right Test Technique**<br>**Derk-Jan de Grood** | **45** |
| **17:00** | **Closing Remarks**<br>**Stuart Reid –** Chair of the BCS Software Testing Specialist Group | **5** |

# Notice of AGM

Notice is hereby given that the Annual General Meeting of the BCS Specialist Group in Software Testing (SIGiST) will be held on Friday 16th September 2011. The venue for this meeting will be the Royal College of Obstetricians and Gynaecologists – RCOG.

## Agenda

- Welcome and Introductions
- Apologies for absence
- Reports
  - Chair
  - Treasurer
  - Standards committee
- Constitutional changes
  - Adoption of the BCS Member Rules document, replacing the SIGiST constitution
- Committee elections
  - Chairman
  - Vice Chairman
  - Treasurer
  - Marketing Secretary
  - Programme Secretary
- To consider any nominated business

Items for inclusion on the AGM agenda should be emailed to mohinder.khosla@talk21.com. Additions to the agenda must be received no less than three 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 2 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 15 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 section 4. of the SIGiST constitution for eligibility. | At least 5 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. | 5 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), but may only hold one position. In the event that someone is elected to more than one role then they must immediately decide which role they wish to take up and vacate the other positions. The second-placed candidates for the vacated positions are then elected to those roles. |
| 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 section 4. of the SIGiST constitution 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). |

# Testing Events Calendar

## August

**CAST - Conference of the Association for Software Testing**
8 - 10 August 2011
Seattle, USA
http://www.associationforsoftwaretesting.org/conference/

**Software Testing Club Meetup**
25 August 2011
Cardiff, UK
http://www.softwaretestingclub.com/
http://www.meetup.com/SoftwareTestingClub/events/21481961/

## September

**Software Testing Club Meetup**
15 September 2011
Cambridge, UK
http://www.softwaretestingclub.com/
http://www.meetup.com/SoftwareTestingClub/events/17566221/

**SIGiST**
16 September 2011
London, UK
http://www.bcs.org/server.php?show=nav.9264

## October

**STARWEST**
2 - 7 October, 2011
Anaheim, USA
http://www.sqe.com/StarWest/

**BCS Scottish Testing Group**
w/c 24 October 2011
Edinburgh, UK
http://www.bcs.org/server.php?show=nav.9729/

**UK Test Management Forum**
26 October 2011
London, UK
http://uktmf.com/

## November

**EuroSTAR 2011**
21 - 24 November 2011
Manchester, UK
http://www.eurostarconferences.com/

## December

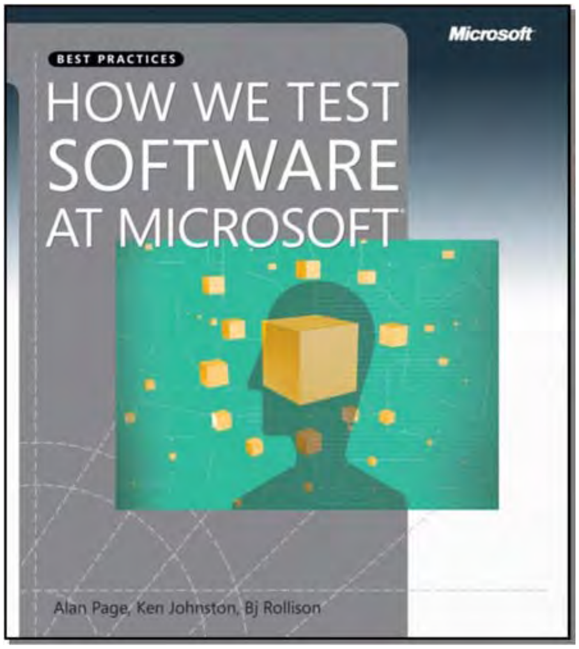
**SIGiST**
13 December 2011
London, UK
http://www.bcs.org/server.php?show=nav.9264

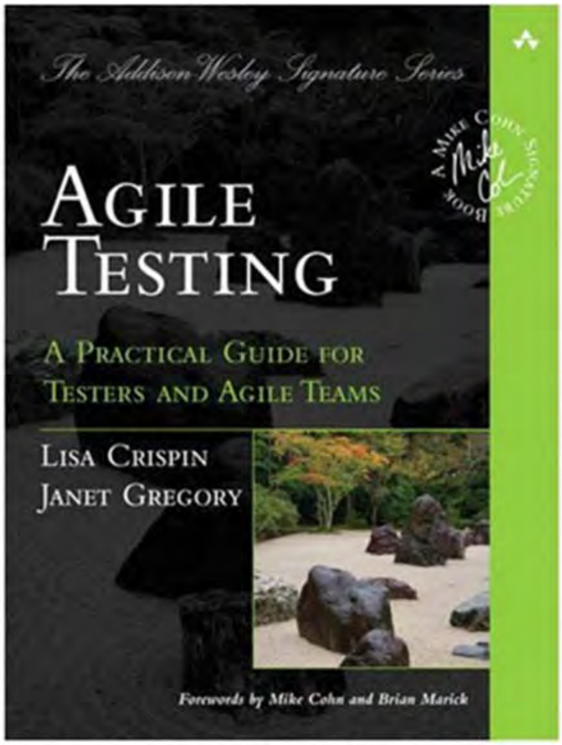
# Specialist Group Library

## Borrowing a book

Looking for a testing book but not sure which topics are covered? Or are you trying to decide which testing book to buy? Or do you simply want to increase your testing knowledge? If the answer to any of these questions is ‘yes’ then the BCS Software Testing Specialist Group Library could help!

The Library has lots of testing books covering a variety of topics and they are available to borrow for a period of 4 weeks - free of charge. Extended loans are allowed as long as the book has not been requested by another member.


Microsoft
BEST PRACTICES
HOW WE TEST SOFTWARE AT MICROSOFT
Alan Page, Ken Johnston, Bj Rollison


The Addison-Wesley Signature Series
A Mike Cohn Signature Book
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY
Forewords by Mike Cohn and Brian Marick

Topics include (amongst others) Requirements testing, Reviews/Inspections, Test Management, Test Techniques and Test Process Improvement.

If you would like to know more about the library and books available, or for any queries, visit… http://www.bcs.org/server.php?show=ConWebDoc.11675

# Testing Automation… How far should it go?

Ian Gilchrist, IPL

Do you recognize this person? "***There is no way that testing should be automated in terms of deriving scripts, input or expected output values, or external call simulations from the code under test. The only purpose of testing is to demonstrate that code does what it should do, not that it does what it does.***" That person could be anyone who comes from the 'old-fashioned' school of development testing, and until fairly recently it certainly included me. I have 25 years experience in software testing, going back to the days of Assembler and Fortran.

However, I have mellowed in recent years and would like to share some of the reasons why I now welcome some of the advances made in software testing technology and in particular the opportunities raised by automation.

> **"…we routinely used the metric: 3 days testing per 1 day of coding. Modern testing tools can reduce this to 1 day of testing per 1 day of coding…"**

The first thing to note is that software testing used to be very labour-intensive. In my experience we routinely used the metric: 3 days testing per 1 day of coding. Modern testing tools can reduce this to 1 day of testing per 1 day of coding, without any significant diminution of the value of the work even by the highest standards. This has been achieved largely by generating test templates from code modules, but still requiring the tester to complete the process by supplying input values and expected outputs.

Secondly, and as a complement to functional testing, it is now possible to auto-generate code 'robustness' tests which simply fire large volumes of input value at the code under test to see if it can survive without crashing. Crashes would typically indicate the existence of divide-by-zeros, overflows, and other weaknesses which are typically not exposed by normal functional testing. These robustness tests can be made fairly rigorous by asking the test tool to auto-select 'limit' values for inputs based on their data types.

Lastly, and most recently there are now options to generate a baseline of tests from code which is considered 'trusted' but for which no unit tests were ever produced (in a repeatable form). These tests auto-select input values based on data types and a knowledge of the values needed to force decision branches within the code. Checks on outputs will auto-select for expected values that pass. Thus, at the end of the process we have a unit test which shows that the code does what it does. The real value now comes when we want to modify the code for new functionality, because we now have a baseline against which we can check that only the changes we want have been made and that there are no unwanted side-effects.

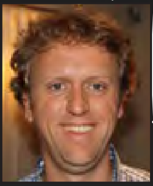
EuroSTAR 2011
Virtual Conference
13th September 2011

# EuroSTAR is delighted to invite you to our first ever FREE Virtual Conference!

The EuroSTAR Virtual Conference will be taking place on Tuesday, 13th September at 9:00am London Local Time. It promises to be an engaging and inspiring virtual event with 8 world class speakers, who will also be available for a live Q&A session after their presentations, interactive discussions in our networking lounge, a Test-Tools Virtual Expo and a test-related resource centre.

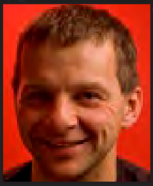
This is a wonderful opportunity to join a truly innovative and insightful conference from the comfort of your own desk, and it's totally FREE! So make sure you register and we'll 'see' you on the 13th September 2011!

## EuroSTAR Virtual Conference Speakers

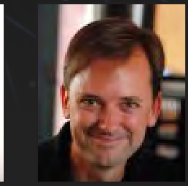

Erik Boelen

Fiona Charles

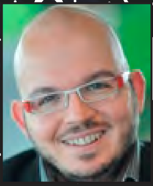
Mark Fewster


Julian Harty

Bart Knaack


Scott Rich

Ruud Teunissen


Anko Tijman

## Register here for the first ever EuroSTAR Virtual Conference

http://www.eurostarconferences.com/content/eurostar-virtual-conference.aspx

## EuroSTAR 2011 Supporting Organisations Competition

**4 Free Conference Places up for Grabs!**

As a member of the BCS SIGIST, a EuroSTAR 2011 Supporting Organisation, you can enter our exclusive Supporting Organisation competition where we are giving away 4 Free Conference Places to EuroSTAR 2011 in Manchester! All you have to do is follow this link and answer the question, http://www.eurostarconferences.com/competitions/supporting-organisation-competition.aspx, Good Luck!

EuroSTAR 2011 Early Bird booking deadline is Friday, 23rd September

Join the EuroSTAR Community @ www.eurostarconferences.com

EuroSTAR 2011 Linked in The EuroSTAR Blog Written by Testers for Testers Webinar Archive eBook Library

# Cloud Testing: a revolution to test the performance of web and mobile apps

Fred Beringer, SOASTA

The web has revolutionized the way we conduct business, consume information and socialize. For some businesses it has become the primary source of revenue and the main customer-facing outlet for advertising and brand management. Applications are not accessed through a fat client anymore but through a web browser, a mobile or a tablet. Applications are now an aggregation of contents, information, data, and media. Application architecture gets more complex and rely more and more on external third-party. The web has forever changed the application landscape.

Performance engineering has had trouble to adapt to this new paradigm which require scale, speed and of course affordability. Performance testing has remained in the lab, behind the firewall while the need for realistic and external testing is more important than ever before.

Unfortunately, most applications are not tested efficiently today, due to expensive hardware investment they're still tested at a small percent of expected traffic and then extrapolated to get an unreliable estimation for performance.

Cloud Testing was born in 2006 when cloud computing took off. Cloud computing provides a simple web service interface in order to obtain and configure capacity with minimal friction. It reduces the time requires to obtain and boot new server instances to minutes, allowing to quickly scale capacity, both up and down, as computing requirements change. Cloud Computing changes the economics of computing by allowing to pay only for capacity needed.

> **"Unfortunately, most applications are not [performance] tested efficiently today, due to expensive hardware investment they're still tested at a small percent of expected traffic…"**

Cloud testing was born based on cloud computing's promises and offer today a fast, scalable and affordable approach to test web application.

Specifically, leveraging the Cloud to do this type of testing yields a number of benefits:

- Tests can be achieved at level observed on production systems, but also at unexpected level. Typically companies would want to test at 100% of typical traffic level but cloud testing allows them to test at 150%, 300%, 500%! They're able to generate from hundreds to millions of users.

- It is possible to generate a realistic and geographically dispersed load. Today's online world requires real-world traffic.

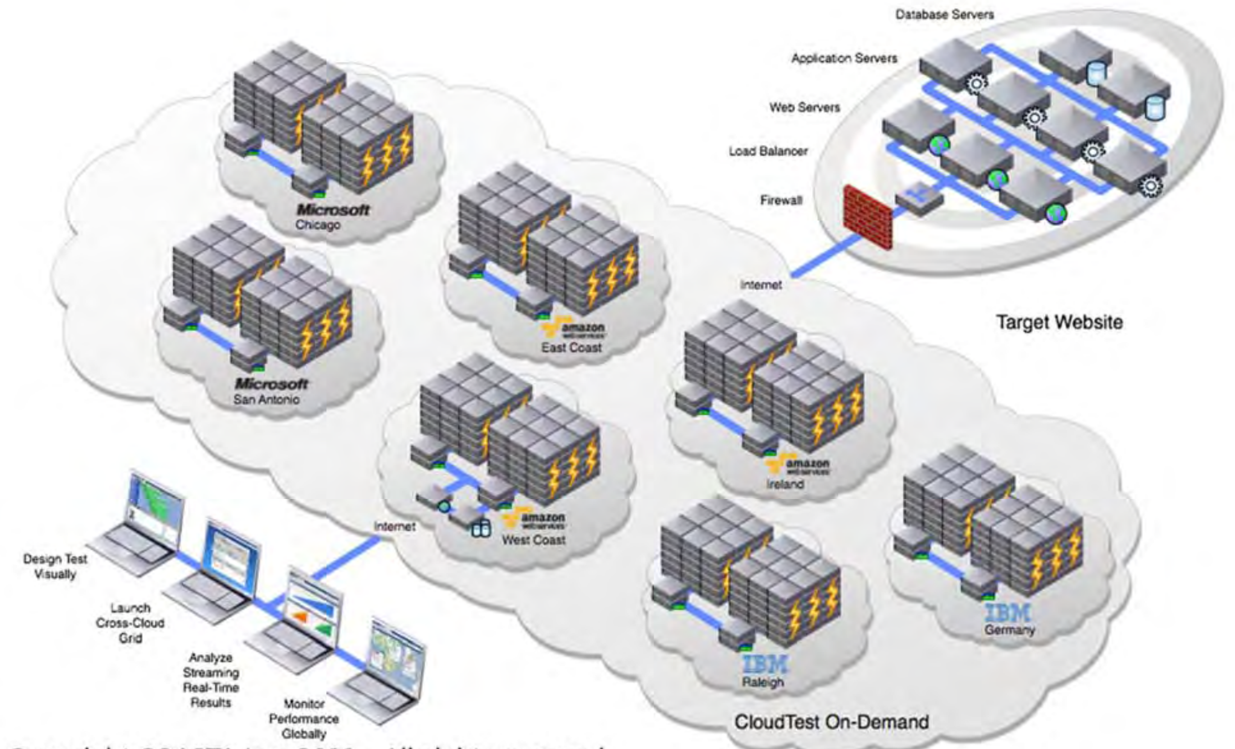
- It is possible to test both inside and outside the firewall, and find all potential performance problems i.e.

Inherent to the application itself but also to its entire ecosystem.

- The cost of tests is definitely lower as organization rents hardware and pay-as-they-use.
- It allows testers to respond to fast development cycle time by making agile performance testing a realistic alternative.
- For the first time, performance tests can be run in production. This is the only way to gain full confidence in the application.

The figure below describes a fairly common setup for cloud testing. Load injectors are deployed in public clouds and are able to target a website whether it is located in a data center or in the cloud. The Cloud is also used to gather the terrabytes of performance data coming back from the application and its underlying infrastructure. Performance analytics can then be aggrgated, combined and correlated in a central location.

Cloud testing is a "game changer" for organization seeking greater levels of web reliability. It enables a new, leading edge, agile and cost-effective performance and load testing approach. With cloud testing, no organization has any excuses not to do performance testing the way it is supposed to be done. The scalability challenges are long gone. The speed and agility is definitely built-in. Best of all cloud testing is a fraction of the cost of traditional testing in a lab. A true revolution!



Copyright SOASTA Inc. 2011 - All right reserved

# 6 October 2011, London

**iqnite 2011 United Kingdom – The Conference for Software Quality**

## Conference Agenda

Special offer:

**15 % discount for members**

Promotion code: BCSNews

| | |
|---|---|
| 9.30 - 9.45 | CONFERENCE OPENING |
| 9.45 - 10.30 | KEYNOTE<br>**Individual Success is not always about Individuals**<br>Baroness Tanni Grey-Thompson DBE |
| 10.30 - 11.00 | EXHIBITION & COFFEE BREAK |

| | |
|---|---|
| **Innovation**<br>Chair: Helen Willington, Sodexo | **Industries**<br>Chair: Michiel van der Voort, BCS, The Chartered Institute for IT |
| 11.00 – 11.40<br>**CERN Opens a New Dimension of Software Quality: Static Analysis Meets 50 MLOC**<br>Axel Naumann, CERN | 11.00 – 11.40<br>**Uniting the Tribes: Bringing IT and Business together through Testing**<br>Den Fitzpatrick, Dixons Group / Deri Jones, SciVisum |
| 11.45 – 12.30<br>**The Gaps in the Integration of Security Testing in Quality Frameworks**<br>Bola Rotibi, (ISC)$^2$ Application Security Advisory Board (ASAB) | 11.45 – 12.30<br>**The CFS Approach to a One-Test Service Transformation**<br>Paul Shatwell, Cooperative Financial Services / Kiruba Vijayaraghavan, Infosys |

| | |
|---|---|
| 12.30 – 1.30 | EXHIBITION & LUNCH BREAK |
| 1.30 – 2.15 | PANEL DISCUSSION<br>**Quality Assurance vs Testing – Perception and Reality**<br>Programme Committee of iqnite 2011 United Kingdom |
| 2.15 – 2.45 | EXHIBITION & COFFEE BREAK |

| | |
|---|---|
| **Agile**<br>Chair: Thomas Spielmann, Centrica | **Metrics**<br>Chair: Mark Mitton MBE, Deutsche Bank |
| 2.45 – 3.25<br>**No Test Levels needed in Agile Software Development!**<br>Leo van der Aalst, Fontys University of Applied Sciences | 2.45 – 3.25<br>**The Value of QA Metrics at Allan Gray**<br>Margarethe de Cafmeyer, Allan Gray |
| 3.30 – 4.15<br>**Top 10 Quality Tips to Agile**<br>Stevan Zivanovic, BJSS | 3.30 – 4.15<br>**Making Problem Analysis & Resolution a Success**<br>Jan van Moll, Philips Healthcare |

| | |
|---|---|
| 4.15 – 4.45 | EXHIBITION & COFFEE BREAK |
| 4.45 – 5.30 | KEYNOTE<br>**Managed Testing Service at Specsavers – The Way to and the Benefits from**<br>Jason Taylor, Specsavers |
| 5.30 – 6.30 | NETWORKING SESSION |

**Please visit www.iqnite-conferences.com/uk for more details on this year's iqnite in London, United Kingdom.**

# “TBYDWTFIP” Go sleuthing with the right test design techniques

Derk-Jan de Grood, Valori

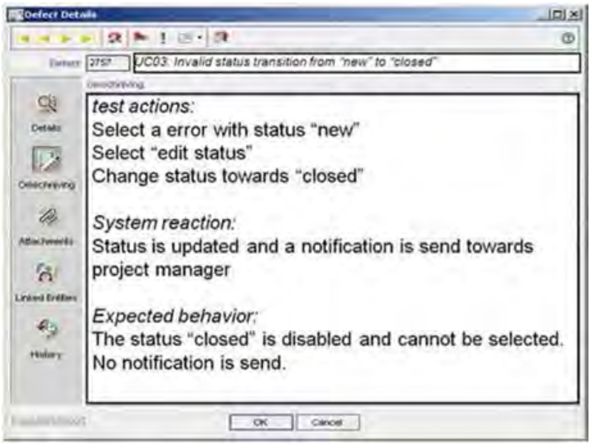
This September I will be talking on the BCS SIGIST about test design techniques. I always find this a rewarding but risky topic. Rewarding, because the discussions about techniques never sees to come to an end, and both on technical and management level discussions are challenging.  How should we use techniques, what are the differences between the techniques, they are a curse or a blessing? Each tester has his own experience and his own ideas. Because of this, the topic is also a risky one. There are testers that take theory very seriously, and if you say something wrong during a presentation, references to the literature are quoted heartedly.  Personally, I think you need to know your techniques well, but you should not be limited by the theory. Use them to your advantage. I you want to bend the rules, fine with me. Anything goes! , but I advise you to use them.

How we can use the techniques effectively? One acronym plays a center role is TBYDWTFIP. Experienced testers might recognize this right away as the acronym for "The Bugs You Don’t Want To Find In Production".

Testing can be seen as an activity that tries to find errors before they occur in production. But do we know where we need to focus our search? I get the impression we rely on the specifications too often and forget to ask to our users and operational managers what problems or errors they do not want to find during operations.

Test techniques to help us to look at the application with a aimed focus. Each technique is specialized in finding certain types of errors. Principally, you will not find errors in the state transitions with boundary value analysis. State transition testing will not help you in finding errors in the input validation. If we know what our errors our stakeholders hate to find in production, we know what to do. Find them before we go live! By choosing the right test design techniques we will improve our chances of success in finding them.

In order to select the most efficient  test techniques is important have a understanding of relation between the techniques and errors they help detecting. During training sessions I like to show the participants problem reports and ask them: “What test technique could have been used to find the bug?”. Experience learns that the question is more difficult than you might expect.  Below I show you one of those bug reports. You’ll do it better?



UC03: Invalid status transition from "new" to "closed"

*test actions:*
Select a error with status “new”
Select “edit status”
Change status towards “closed”

*System reaction:*
Status is updated and a notification is send towards project manager

*Expected behavior:*
The status “closed” is disabled and cannot be selected.
No notification is send.

# Presentation Abstracts and Speaker Biographies

## Paul Gerrard

### "Using Business Stories to Test Requirements and Systems"

The use of stories to communicate dates back 30,000 years to the time when cave paintings recorded daily experience of people living as hunter-gatherers. If a software team uses a whiteboard to capture and talk about user stories to scope the next phase of development, they are drawing on an instinctive need to use examples, to criticise, discuss and refine them to arrive at a shared understanding.

Stories worked for cavemen, they work for agile teams, and they'll work for you too because they are universal.

Stories derived from written requirements can be used to walk-through business scenarios and when users see the proposed system 'in action', requirements anomalies stand out and trigger informed discussions of situations, variations and outcomes. A disciplined approach to story-writing and requirements testing can improve requirements and the target solution dramatically. 'Business Stories' can be shared as examples for developers to see what was intended to help their understanding, and of course, they also provide the basis for later acceptance tests.

Up-front requirements testing doesn't require extra effort - much of this analysis work would be done during acceptance test preparation anyway. This approach provides a step-up with business impact analysis, regression testing, and even test automation.

***Paul Gerrard** is a consultant, teacher, author, webmaster, programmer, tester, conference speaker, rowing coach and most recently, a publisher. He has conducted consulting assignments in all aspects of software testing and quality assurance, specialising in test assurance. He has presented keynote talks and tutorials at testing conferences across Europe, the USA, Australia, South Africa and occasionally won awards for them. In 2010, Paul won the Eurostar European Testing Excellence award.*

*Educated at the universities of Oxford and Imperial College London, Paul was the founding chair of the BCS ISEB Testing Certificate Scheme and a member of the Working Party that produced BS 7925 – the Component Test Standard.*

## Mark Fewster

### "Experience Driven Test Automation"

Oh no! Is this yet another approach to Test Automation? Actually, no it isn't. This is about what other peoples' experience with test automation can teach us - how it can help us capitalise on good ideas and avoid potentially useless ones.

A new book by Dorothy Graham and Mark Fewster "Experiences of Test Automation" due to be published this autumn describes 29 case histories of test automation across a rich variety of application domains, environments and organisations. The book includes success stories, failure stories, and a few so-far-so-good stories.

While every story is different, there are some surprisingly common elements running through these case studies. In this presentation Mark highlights some of the common themes that span both management and technical issues. For example, the influence that managers have over test automation success and failure, the importance of keeping management informed and involved, and the need to match an appropriate level of investment with the desired objectives. Some of the technical issues include attention to testware architecture at an early stage, consistency of working methods to encourage reuse and reduce maintenance costs, and the quality of scripting.

By studying the experience of others we can start or progress our own test automation with a deeper understanding of the important issues, mitigate risks and capitalise on opportunities.

*Since joining Grove Consultants in 1993, **Mark** has provided consultancy and training in software testing, particularly in the application of testing techniques and test automation. He has published papers in respected journals and is a popular speaker at national and international conferences and seminars.*

*Mark has served as Programme Secretary on the committee of British Computer Society's Specialist Interest Group in Software Testing (BCS SIGiST) and has also served on the Information Systems Examination Board (ISEB). He is currently helping the ISTQB in defining the expert level certification for test automation.Mark has co-authored a book with Dorothy Graham, "Software Test Automation" published by Addison-Wesley. Mark and Dot are currently working a new book "Experiences of Test Automation" to be published in 2011.*

# Fred Beringer

## "The Cloud: A Game-Changer for Web Performance Testing"

Today in retail, financial services, media, telecommunications and a host of other industries, more and more business is transacted through consumer web sites and mobile applications. With new channels creating spikes in traffic, highly complex system architectures, and internet-savvy customers, websites and web applications must be tested *at scale* to maximize business results and avoid a catastrophic crash. However, whether due to time or cost or other reasons, upwards of 90 percent of web applications are not fully tested before launching. If testing is done, many times it's with a small percent of expected traffic, which is then extrapolated for an estimation of performance.

Cloud computing is changing the game for testing web applications. Cloud testing enables, for the first time, performance testing that complements the lab and accounts for the conditions in a production environment, such as traffic spikes, network latency, firewalls, and other factors. And it can be done far more affordably than traditional testing methods, as part of agile development cycles, and without an army of highly skilled performance engineers.

Through customer examples Fred will explain why performance testing is more important than ever before, the fundamental of cloud computing and its application to full-scale performance validation. Fred will describe a methodology for iteratively testing all levels of infrastructure and software that uncovers issues in real time, and ultimately tests true production environments at scale.

***Fred Beringer** is Vice President Business Development at SOASTA. He has 15 years of software development and testing experience, managing large organizations responsible to develop software and application for large financial services and telecommunications customers.*

*Mr. Beringer was Software Testing and QA director for Experian Decision Analytics where he built and led a worldwide organization* *responsible to test all existing and new products within the Decision Analytics portfolio. During his 4 years at Experian, Mr. Beringer introduced Cloud Testing in his team while helping the overall development organization transition to agile methodologies.*

# Peter Morgan

## “Book Review - Agile Testing, a Practical Guide”

The short session will give insight into a very practical manual. The authors (Lisa Crispin and Janet Gregory) deliver useful tips and experience to make a real difference to those engaged in delivering working software in an Agile development environment. The book is not a “story”, and most may well not read it in strictly sequential order, but there are sufficient forward and backward pointers to add significant value even if a time-pressed individual only initially intended to read just one chapter or part thereof.

This is nothing like a complete guide to testing in an Agile development life cycle model. It is better than that, in that there are no ready-made answers. This may enable YOU to write your own “complete guide to testing in an Agile development life cycle model”, tailored to the needs of your company and the individuals involved. Lisa and Janet ask the questions. It is up to you to answer them.

***Peter Morgan** is a freelance testing professional with more than 30 years experience in the ICT industry. His time has sometimes moved from testing to ‘development’, but he would add “always using the mindset of a tester”. An enthusiastic speaker and author, Peter tries to base his output on hands-on experience, attempting to relate fine sounding ideas back to how it will affect Joe or Jane Tester in their everyday working lives.*

# Mieke Gevers

## “Ever Been Fooled by Performance Testing Results?”

Have you ever been in a situation where Performance testing results could fool you? What is meant by "data"? Is it possible to get control of it? This session consists of presenting tips on to how to get control of the performance testing  results and its evaluation process. Bringing a real live experience to the audience, several case studies will be shown so the participants can help to evaluate the presented data.

***Mieke Gevers** has been in the IT industry for more than twenty years. She has developed a special interest in the techniques and processes relating to performance management and automated testing. Mieke has been a speaker at various conferences throughout the world including STAREAST, STARWEST, EuroSTAR, AsiaSTAR and Qsit (India) as well as Special Interest Groups in software testing in several countries. She served on the Program Committee for the EuroSTAR conference in 2007 and 2009 and was Program Chair of the Belgium Testing Days 2011 and 2012. A co-founder of the Belgian Testers Organization, Mieke has been a board member of KVIV and the ISTQB-affiliate Belgian Software Testing Qualifications Board.*

# Ian Gilchrist

## “Test Automation… How far should it go?”

The automation of software testing has come a long way in 30 years, from a manual and non-repeatable process to fully automated and repeatable. However the question can be asked whether we have gone too far? In the haste to create tests that ‘Pass’ have we lost track of the main purpose of testing, which is to show that software does what it should do? What, if anything, is the role of full automation? Illustrated with examples from C-code modules.

***Ian** has nearly 30 years in the software industry, mainly in 'high-integrity' work, using languages such as Assembler, Fortran, Ada and C. Has focussed on software testing tools for the last 15+ years.*

# Derk-Jan de Grood

## "Go Sleuthing with the Right Test Technique"

Although much information is available on test design techniques, very little is written on how to select which techniques to use for the job at hand. Derk-Jan de Grood believes that many testers find it difficult to select the right techniques and very often use a technique simply because they know it. Instead, the best reason is that the technique is likely to discover the most important errors quickly. Derk-Jan shares his insights on test technique selection and poses three questions you should ask yourself when selecting a technique: What types of errors do I want to find? What impact do these errors have in production? Is the needed information to perform these tests available? He then lays out a list of common test techniques and discusses which error types they are most likely to discover. Take back a new understanding of test technique choice and selection to become a better software defect sleuth.

We testers set a great store to test design techniques. TMap Next dedicates about 130 pages to test design techniques. This is about 17% of the whole book. "Foundations of software testing" by Graham et al., dedicates about 20% of its content to explaining the functioning of different techniques. Strangely enough, how to select a technique is hardly discussed. In my opinion that is a shame, because selecting the right test technique is important for the efficiency of testing. My experience is that many testers find it difficult to select the right techniques.

Selecting the techniques you are going to use is a careful process. Selecting the wrong technique may cost you a lot of time without finding many errors or useful information. Failing to select the right technique may lead to mayor defects not being found. By using the right techniques, the tester executes the right tests and is accountable for his actions and the quality of his advice. Since the techniques define the tests that are executed, they also determine the information that can be given to the stakeholders.

Many times test techniques are applied because the tester is familiar with them and the test base supports the technique. I'll explain the decision model from my book, and will inverse the problem. We do not use techniques because we can, but because they help us to provide the right information. Unlike many methods we start with the errors and select the techniques that help finding them.

***Derk-Jan de Grood** has broad, hands on experience as test engineer, test manager and adviser in a large range of industries. As a manager of several test departments he has learned how to implement test methods the practical way. He gives lectures at various Dutch universities and is author of the first educational book specially written for teaching software testing at Dutch universities. Derk-Jan is also author of TestGoal, the result-driven test philosophy and recently published "the hero that guards my nightly rest".*

*As an ISTQB full advanced certified test manager he provides training sessions on a regular basis. These training session vary from the standard introduction into result driven testing to custom made trainings that tune in on specific needs of the client. Besides that, he is a passionate, inspiring speaker at major testing conferences all over the world such as the STAR conferences in the USA and Europe.*

*See also my linkedIn profile*

December 2011, Edition 39

Specialist Group in Software Testing

# THE TESTER

## Next Conference: Tuesday 13th December 2011

IN THIS ISSUE

## SIGIST Conference Booking Instructions

To register online, please use the link below. Please note, the new BCS booking system accepts multiple and third party bookings:

https://events.bcs.org/book/127

If you have a query relating to making a booking, please contact Gemma Stanley-Evill, Specialist Groups' Officer.

Tel: (01793) 417656

gemma.stanley-evill@hq.bcs.org.uk

## LinkedIn & Twitter

The BCS Software Testing Specialist Group is now using social media platforms to improve communications both to members and between members.

Our LinkedIn Group (link below) will carry details of our conferences as they become available. It will also provide a place where people can discuss testing topics, make requests about future conferences, find employment opportunities (there are a few jobs advertised already) and generally keep up to date with our chosen industry. If you are already a member of LinkedIn then simply visit the group and make a request to join.

If you're not a member then go to http://www.linkedin.com/ to create an account.

If you use Twitter you can follow us @SIGiST.

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

# Conference Agenda

Tuesday 13th December 2011
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

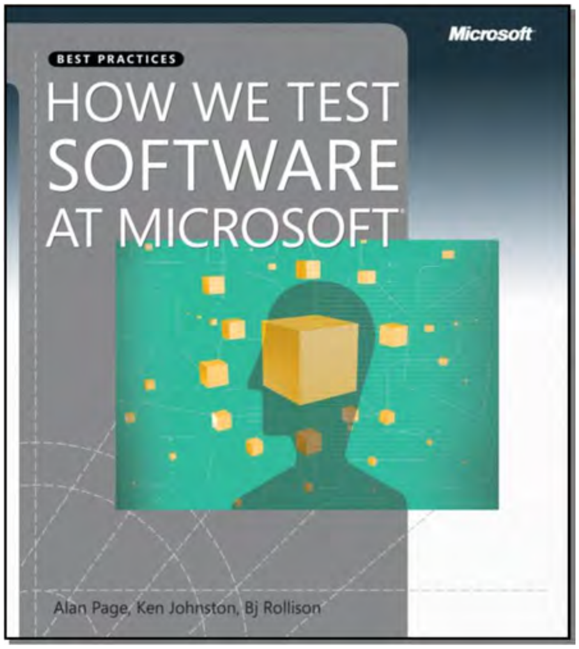
| Time | Session | | Length |
|---|---|---|---|
| 08:30 | **Registration open, Coffee and Exhibition Hall** | | |
| 09:25 | **Welcome & Introduction**<br>**Geoff Thompson –** Vice Chair of the BCS Software Testing Specialist Group | | 5 |
| 09:30 | **OPENING KEYNOTE: Test Techniques**<br>**Anne Mette Hass** | | 60 |
| 10:30 | **Networking Session** | | 15 |
| 10:45 | **Tea / Coffee Break Exhibition Hall and Networking** | | 30 |
| 11.15 | **What does Lean mean for Software Testing**<br>**Mark Robinson** | **MORNING SESSION WORKSHOP**<br>**The Testoff**<br>**TCL** | 45 |
| 12.00 | **Aligning Correct and Realistic Performance Testing with the Agile Development Process**<br>**Graham Parsons** | | 45 |
| 12:45 | **Buffet lunch, Exhibition Hall and Networking** | | 75 |
| 14:00 | **A Brief on ISTQB Expert**<br>**Paul Weymouth** | | 15 |
| 14:15 | **Can scrum prevent defect ping pong**<br>**Patrice Willemot** | | 45 |
| 15:00 | **Tips for the Programme Test Manager**<br>**Lucinda Casey** | | 45 |
| 15:45 | **Break, Exhibition Hall and Networking** | | 30 |
| 16:15 | **CLOSING KEYNOTE: Testing as a Service**<br>**Jonathon Wright** | | 45 |
| 17:00 | **Closing Remarks**<br>**Geoff Thompson –** Vice Chair of the BCS Software Testing Specialist Group | | 5 |

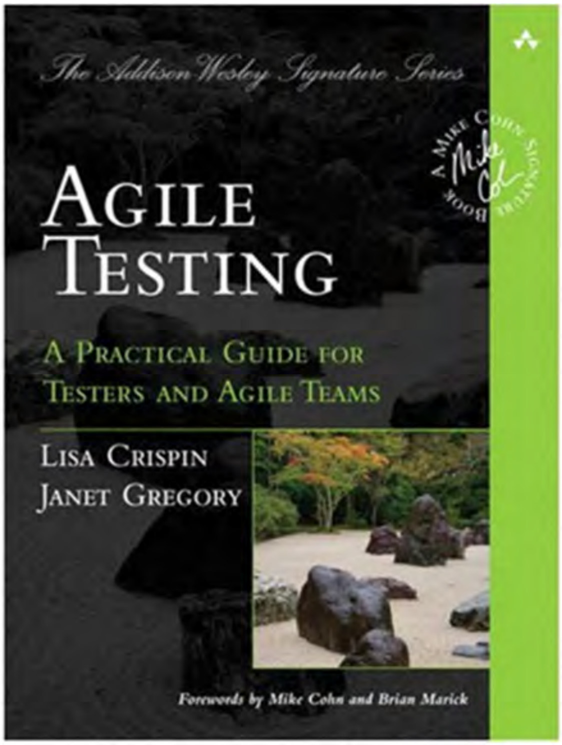
# Specialist Group Library

## Borrowing a book

Looking for a testing book but not sure which topics are covered? Or are you trying to decide which testing book to buy? Or do you simply want to increase your testing knowledge? If the answer to any of these questions is 'yes' then the BCS Software Testing Specialist Group Library could help!

The Library has lots of testing books covering a variety of topics and they are available to borrow for a period of 4 weeks - free of charge. Extended loans are allowed as long as the book has not been requested by another member.


Microsoft
BEST PRACTICES
HOW WE TEST SOFTWARE AT MICROSOFT
Alan Page, Ken Johnston, Bj Rollison


The Addison Wesley Signature Series
A Mike Cohn Signature Book
AGILE TESTING
A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS
LISA CRISPIN
JANET GREGORY
Forewords by Mike Cohn and Brian Marick

Topics include (amongst others) Requirements testing, Reviews/Inspections, Test Management, Test Techniques and Test Process Improvement.

If you would like to know more about the library and books available, or for any queries, visit… http://www.bcs.org/server.php?show=ConWebDoc.11675

# Aligning Correct and Realistic Performance Testing with the Agile Development Process

## Graham Parsons, Reflective Solutions

One major drawback of the Agile development process is that performance testing is normally left until the end of the project, which can lead to projects being delivered late and/or over budget if problems are discovered. This does not need to be the case. New performance testing tools and approaches now mean that it is possible to verify application performance at the end of all iterations. Graham Parsons from Reflective Solutions will speak at the SIGIST event in December to show how application performance testing within sprints is now achievable.

The later a defect is found, the more costly – in terms of both time and money – it is to fix. If the problem is a deep-rooted application code issue, it is not unknown for projects to be delayed by many weeks or even months. The ideal solution to this would be if performance testing could be executed at the end of every iteration, to enable the early identification of performance defects, which will significantly reduce the time and cost of fixing any defects.

However, there are a number of commonly stated objections as to why performance testing cannot be integrated with the Agile development process, including time and effort requirements; the need for specialist resources; scheduling issues; and costly tools.

There is now a new breed of performance testing tool which allow the above objections to be overcome. These new tools are quick and simple to learn and use, significantly lowering the barriers to entry for performance testing.

The reliance on specialist, expert resources can now be removed, meaning that performance testing can be undertaken in timescales that are suitable for Agile project developments. This simplicity enables anyone in a development team to learn how to use a tool in just a few days and, once trained, start configuring and executing performance tests.

Performance testing throughout Agile projects is now truly possible. Test at the end of each sprint and you remove the risk of a problem being identified at the end of development and delaying project delivery.

# What does Lean mean for Software Testing?

Mark Robinson

Lean is all about speed: how can we get the solution to the customer as fast as possible? To increase speed, we need to stop doing activities that do not add any value to the final product - which Lean defines as waste. Things like:

- having lists of work in progress (large defect databases, anyone?),
- adding features nobody will use (and, as testers, having to test them all for each release),
- multiple departments all solving the same problem, all "re-inventing the wheel" without effectively communicating lessons learnt,
- people all working on a part of the solution, playing a "Chinese whispers" game as the requirements move down the value chain and tacit information is lost,
- working on multiple tasks simultaneously, in which time drains away as people mentally switch between jobs
- waiting for people and test systems to become available or decisions to be made and
- making and solving defects, which requires a "hardening" of software via multiple test runs.

Lean helps us by providing tools to identify and reduce waste. One very powerful tool is the Value Stream Map, where a process of delivering value to the customer (feature or bug fix) is mapped out and the lead time is plotted against the effort time. How efficient are our processes, really? Another is Kanban, which forces us to work within the constraints of our available resources and ditch the "wishful thinking" which so often charcterises our planning.

As testers, I see our role changing dramatically, from being people who test after the software has been written to people who are actively up-front *building quality in*, together with developers.

I will discuss this in more detail on the 13 December at the SIGIST conference, together with solutions, demonstrations and stories.

# Presentation Abstracts and Speaker Biographies

## Anne Mette Hass

### "Test Techniques"

One of the main objectives for a good tester is to produce good test cases. The talk will explain what a good test case is, and how testers can benefit from using specific techniques to produce test cases. The talk will give an overview of some of the most used techniques and demonstrate how some of them can be used.

***Anne Mette Hass** has worked in IT since 1980. She has been involved in all areas of software development, but specialized early in requirements, quality assurance, and configuration management.*

*Anne Mette Hass is technology manager at Devoteam Consulting A/S in Denmark. She is a very experienced consultant in requirements engineering and management and software testing*

*Anne Mette Hass is also a very experienced teacher end often take the role as mentor and sparring partner.*

*Anne Mette Hass has been employed in different lines of business, including health care, oil industry, telecommunications, Digital Equipment Corporation (DEC), and as a supplier for the space industry in Denmark, Norway, England, France and Italy.*

## Mark Robinson

### "What does Lean mean for Software Testing"

Traditionally, software testing occurs after the software is written. From a Lean perspective, this is waste because the software needs to be reworked, retested and often reworked again. Instead, Lean says that quality should be built-in to the way of working.

This presentation will explain how the seven wastes of software development (partially done work, extra features, relearning, hand-offs, task switching, delays and defects) apply specifically to software testing. It will then focus on how the last two can be prevented; delays by using Kanban and Value Stream Mapping (using a specific example of a defect found by a customer) and defects by regular automated smoke testing and Test Driven Development. Use will be made of entertaining, interactive examples and stories from real life experience.

The conclusion is that each found defect should be translated into an automated test case to ensure it cannot re-occur: "mistake-proofing".

***Mark Robinson** passed ISEB Practitioner in Software Testing with Distinction and has had software testing and test team lead roles at various companies. He has also coached junior and senior testers and written software testing articles for Bits & Chips and Computable. Mark is strongly interested in making business processes, especially testing, more effective by using various techniques like Test Process*

*Improvement, Lean Software Development and wikis to improve corporate communication.*

# Graham Parsons

## "Aligning Correct and Realistic Performance Testing with the Agile Development Process"

One major drawback of the Agile development process is that performance testing is normally left until the end of the project, which can lead to projects being delivered late and/or over budget if problems are discovered. This does not need to be the case. New performance testing tools and approaches now mean that it is possible to verify application performance at the end of all iterations. Join this session to discover how application performance testing within sprints is now achievable.

*A recognised expert in the field of web application performance and load testing, **Graham Parsons** is co-founder and CEO at Reflective Solutions.*

*Graham has played a key role in improving the efficiency and effectiveness of performance testing tools, through the advancements of Reflective Solutions' industry-leading, StressTester™.*

*Trained as a specialist in Java architecture, Graham co-founded Reflective Solutions in 1998 to offer high level strategic consultancy to adopters of the programming language and computing platform. In the subsequent years the company developed specialist knowledge in the area of enterprise Java performance and created sophisticated products to test the scalability of this technology.*

# TCL (Andy Dowson, Paul Darby, James Brooks)

## "Workshop: The Testoff"

The Testoff is back at the SIGIST by popular demand!

Run by TCL, a specialist consultancy in software testing, this is your chance to learn some hands-on testing skills by taking part in this competitive session to test applications.

Find as many bugs as you can as quickly as possible to beat the clock and the other teams!

The Testoff is a workshop open to 16 delegates who will get the opportunity to work in teams to test an application and learn practical and applicable testing skills along the way.

Testing will run for an hour with an introduction and the winning team announced at the end of the session.

***James Brooks, Consultancy Partner, TCL***
*James has been an IT consultant since 2000. Having joined TCL in 2001 as a Graduate Test Analyst he then worked his way up through the ranks to his current position, where he is responsible for service delivery and the management of client relationships for a number of key TCL customers including Global telecoms companies and the World leading weather service.*

***Paul Darby, Account Manager Designate, TCL***
*Paul has over 10 years of software test experience within Public Service, Pre-Press and defence industries. Paul's testing career has seen him testing on Unix, Windows and Mac environments, working with teams in the UK and America. He started at TCL in 2007*

*and has been working on the FiReControl project for EADS.*

***Andy Dowson, Project Manager, uTest***
*Andy is the Project Manager for uTest, the world's largest marketplace for software testing services. The company provides real-world testing services through its community of 40,000 + professional testers from 175+ countries around the world.*

# Patrice Willemot

## "Can scrum prevent defect ping pong"

Defect ping pong is a process where a defect is driven back and forth between several players. Those players are often identified as a tester and a developer. Is scrum able to solve this process? Can an agile approach where communication and team spirit are important prevent this process? I am sure of it and during this presentation I will prove it.

***Patrice** started his testing career as a WinRunner test automation engineer and moved to test lead and test manager to become a process test consultant at CTG nv Belgium. He has been in the field of Software Testing since 2004 testing several disciplines such as banking, healthcare and automotive. He used different test approaches and test types for testing GUI applications, web applications and interfaces. For the past 1.5 years he has been interested in the agile world. He is very curious about how different test approaches can be used within agile environments.*

# Lucinda Casey

## "Tips for the Programme Test Manager"

Many Programme Test Managers today operate in a world of mass outsourcing where multiple incumbent suppliers own the many different components that make up the organisations systems and processes. So how does the PTM establish and run a test function to encompass the supplier differences in areas such as delivery of code, testware and environment preparation, and then manage sufficient test execution and efficient defect turnaround whilst also managing stakeholders throughout the lifecycle?

***Lucinda** is a youngish female test manager - freelance. I have 10 + years' experience in software testing for major blue chip companies in pharmaceuticals, banking and retail.*

# Jonathon Wright

## "Testing as a Service"

The current economic climate is making companies review their approach to IT even more closely. From a recent Computer Weekly article: "... the prediction that 60% of the average enterprise will have 60% of its applications in the cloud." (Karl Flinders, 19/10/11). However, the same vision could also extend to IT services and particularly testing. Existing technologies of virtualisation, business process modelling, cloud based test automation tools and rapid and easy internet access allow for the development of approaches that allow companies to order testing as a service and pay only for what they use. There is no need to spend large sums on test environments and data, test tool selection and maintenance. The use of cloud based services mean you can select the right level of service at a time when you need it and at the volumes you need it whether it is one or one thousand testers on one or many environments, located locally or around the world. The use of service allows dynamic scaling within minutes. This presentation introduces the approach that we are taking to evolve this service to our existing clients.

***Jonathon Wright** has over 10 years of commercial automation framework experience with a number of global organisations including Lehman Brothers, New Zealand Lotteries and PlanIT (based in Sydney). He is an active blogger on "Test Automation as a Service" (TaaaS.net) as well as presenting at various international testing conferences. Currently working leading the 'Test Innovation and Automation' team at BJSS based in central London and is also contributing towards a number of upcoming books on test automation and testing in the cloud.*

# Common procedures for testing web applications

Kevin C. Martin, Alver Services
07825 538260
8MARTINK@uk.ibm.com
kevin@alverservices.com
Bsc Hons, MBCS, Msc-IT

## Abstract

This case study discusses the complexity of testing web based applications and some of the more common manual methods employed for the task. Here we discuss how the test team can get the best out of these methods and how the rest of the development team can help in this process, software development is after all a team process. On a future paper I also intend to discuss the merits and disadvantages of automated testing techniques.

## Background

As I have stated in a previous paper (K Martin 2010); software development has changed dramatically over the past fifty years. In the early days of software development system memory was at a premium, coding was achieved in machine code and most programs were small and simple. As a result testing was considered a minor task and a hindrance by most programmers.

As software development tools evolved we entered the era of the DOS programmer. These programmers used development tools such as C, Ada and Pascal to write their programs. They still considered testing a hindrance and tried to avoid it at any cost, it was a job for lesser mortals. Most software companies spent as little investment as possible on testing and most bugs tended to be reported by end users.

With the advent of Windows a new level of complexity and confusion was added to the software mix. Suddenly not only did software development companies have to worry about their own bugs but there was also the added concerns of just how stable was the Windows operating systems their clients were now using.

Windows and other GUI's have evolved tremendously over the past decade, some versions have been much better than others, a prime example of a quickly dropped but never to be forgotten version will always be the infamous Windows ME.

All of these concerns have awoken most software houses to the real importance of good software testing methods and good software testers.

There are many good (and some not so good) software testing methods available. Some or more suited to a particular development project then others and a good software development team will only select methods that are applicable to the project at hand. Flexibility is a crucial skill in the development process and any team that has become inflexible and rigid in their approach will soon encounter major problems as they tackle new and varied projects.

In this paper I am discussing a small selection of some of the more common methods of manual software testing. Not all of these are used in every development project you may encounter but they will be seen in many. The first and most common is Functional Testing.

## Functional Testing

Functional testing (n.d.) will nearly always be required during a software test phase. There are essentially two types of functional testing; these are Full program testing and Change testing.

Full program testing is most commonly used when a web application is being tested prior to its first release into production. The test document created will be a complete step by step test of the web application. Every process should be carefully tested and the results analysed and fully documented.

No test of a new web application should be started until the Project Leader has held a development meeting with the assigned tester(s) and programmer(s). During this meeting the functionality and design of the new web application should be discussed in detail. Any specific requirements for the test schedule should also be highlighted during this meeting. The tester should also be presented with a copy of the original Project Initiation Document (PID) and at least one Test Case document (hopefully a lot more).

A Test Case document is a document which details one of more Test Cases, commonly referred to as a test suite. The Test Case itself has components that describe an input, action or event as well as an expected response. The purpose of each Test Case is to determine if a feature with the application is working correctly.

Test cases should be written by project team’s members who have a good knowledge of the systems functionalities as well as a good understanding of the client’s business processes. Therefore depending on your project team structure this is most commonly the Project Leader, although less commonly a developer or senior tester will undertake this task.

With this form of testing all other sections of the agreed test procedure as detailed below should also be undertaken with the exception of Regression Testing.

In contrast change testing is used when a previously published web application has undergone changes, bug fixes or enhancements. These changes may be very small or very significant. No matter what the scope of change the same processes and attention to detail should always apply. To successfully complete this type of testing the test team will require a copy of the Project Initiation Document (PID) which relates to this change and at least one Test Case document. To ensure that the test team fully understands what is required within the actual test a meeting should also be held by the Project Leader to discuss the changes and test requirements. With this form of testing all other sections of the test procedure as detailed below should also be undertaken.

Depending on the type of application under test it may be required to carry out functional testing on different browsers. If the web application is for in-house use only then this may well only require testing on company approved browsers. If the application is for use by a customer's work force then they may well also enforce a list of approved web browsers, good practice dictates your testing is conducted on all of these. If the application under test is available to the general public or there is no approved browser policy then it would be prudent to test all functional testing on all of the most common browsers, examples being Firefox, Chrome, Safari and Internet Explorer.

All complete retest on each browser should not be required. If certain pages are read only information pages then a general sampling should be sufficient. If however a web page is used to input or edit data such as a logon page order an online order page then these sections should be tested in all selected browsers to ensure sensitive data is not compromised and user action such as pressing the back button does not duplicate records.

The browsers and versions used should also be documented in the test document. This will ensure that anyone reading the document in the future will be fully aware of which browsers and versions were used during the test. The tester should fully detail the functional test by outlining each process with as much detail as is required to convey the thoroughness of the testing. Screenshots should also be included, these provide and very informative graphic view which backs up the textual description of the test. The order in which the test is completed is not important as long as every aspect is covered.

While completing the functional testing the test team should also be aware of the requirements of the other sections in the test document. For example any delays, lags should be noted for inclusion in the Non Functional Testing section. Also the test team should test access functionality before and after access has been removed from certain application functions. This will also include testing by pasting the URL into the browser address bar as per Logical Access Testing. The results of such tests should be recorded for inclusion in the logical testing section.

An important consideration during functional testing is 'where is the data coming from?' and 'is it correct?' For example all list boxes that are pre-filled with data, is it the correct data? It is crucial to ensure that the correct data is available in these lists and this may well vary depending on the user's access rights or their geographical location. If required the programming team should supply information which will explain how certain data is returned to the web application and under what circumstances.

Where applicable all calculations (decimal, percentage etc) will need to be carefully checked and verified before passing the functional test stage. Decimal values should also be validated as being correct in different locales as detailed in the locale testing section. This is part of the functional test and should be recorded for documentation in the locale section of the test document.

As you will appreciate Functional Testing is a very important type of testing, very few projects can be completed with this type of testing. Another common type of testing is Non functional testing.

## Non Functional Testing

Non-functional (n.d.) testing refers to aspects of the application under test such as usability or performance. Non-functional testing tends to answer such questions as ‘how well does the system perform when I save a new record?’

Non-functional testing can include (but is not limited to): Usability, Robustness, Compatibility, Performance, Load, Stress, Endurance, Stability, Accessibility, Extensibility, Scalability and Portability. Generally speaking, it is the testing of 'how well' the system works under normal usage.

For this test, the tester must consider the performance and system speed of each operation they test. From practical experience gained by years of software development and testing I have concluded that one of the best ways to achieve this is to keep asking these questions while performing the functional test. If the test team at any time were to notice a system delay or significant performance drop during functional testing they should note it and fully document their findings when they get to the Non Functional section of the test documentation.

Non functional testing and functional testing are usually completed at the same time. Some development teams will document them separately while others will merge both types into the same part of the document. Neither method is actually wrong as long as the testing is complete and correctly recorded.

## Regression Testing

Regression testing (n.d.) is any type of software testing that seeks to uncover new errors, or *regressions*, in existing functionality after changes have been made to the software, such as functional enhancements, bug fixes or configuration changes.

The intent of regression testing is to assure that a change, such as a bug fix, did not introduce new bugs and that the base functionality of the application has not been broken.

This form of testing is not required if the web application under test is a new application that has not been previously released, however existing applications will need regression testing. The most common method of regression testing is rerunning previously run tests. The tester should locate the most recent complete test document for the web application as well as the last three change test documents (less if only one or two previous documents exist). These documents should be stored in your company's document repository.

Regression testing involves retesting the unchanged parts of the web application and to achieve this goal the tester should step carefully through the previous test documents checking that the results are still the same. The steps need to be carefully mirrored and should be completed in Internet Explorer (IE), Firefox and other popular browsers. Which browsers were used should also be documented.

Any crashes, unexpected results or strange responses should be reported to the programmer via your company's usual bug reporting facility, my personal preference is Bugzilla but this is simply one of many options. When a detected bug is to be fixed and the new code released will often depend on the severity of the issue and how it is seen to affect the system as a whole.

Generally if the bug is a minor issue that does not affect further testing then holding the release to a point where a number of bug fixes are released should be considered. This causes less disruption by not having to stop testers while the server is updated. If the revised code is minor fixes then the testing should continue, however once the testing has reached the agreed end point a decision has then to be made in regard to the revisions and should a new full regression test be undertaken. Unless we are talking about really minor changes such as single line code changes or html test changes then the answer will nearly always be yes.

If, however the bug is a major problem that will affect or even stop further testing then the fix should be released as soon as it is available. If this is the cause of action then as soon as the revised code has been uploaded then a new full regression test should then started to ensure that the new fix has not introduced yet more new errors. This process should continue until a complete successful regression test is recorded.

Another common type of testing for web based applications is logical access testing. This type of testing is becoming more common and is discussed next.

## Logical Access Testing

When testing web based applications logical access is a vitally important part of the test regime. Testing should be broken into two distinct sections; the first section is Zero Access.

Zero access is the operation of testing that a URL can only be reached after a user has successfully logged into the application using a valid User ID and a valid password, otherwise they should be returned to the login page or a pre-defined warning page. An effective method for preparing for this test is to record all URL's that the tester encounters during the Functional test stage. It is vital that every possible URL is gathered during the functional test and then tested during this stage and this method helps reduce the number of missed URL's.

The method most often used is to log out of the application and then to paste each URL into the address bar and check the response. Your desired response will probably be for the user to be sent to a log in screen or a pre-defined error page. What you do not want is for the user to be allowed system access after they have logged out of the system.

The second section of logical access testing is Profile Access. This section is more complex than Zero Access and requires more thought while testing. In this section you can assume the user has logged in correctly and you should use the same URL's noted in functional testing as you used in the Zero Access section, In this section the questions to be answered are:

1. Does the system allow the user access to parts of the application that their profile states they should have access to? This should be tested by logging into the system and navigating to the required section.
2. This test should then be carried out by adding the URL's for certain areas and locations directly. If access to a section of the application is attempted that the logged in user does not have access to they should be returned to a logical position such as the systems main menu page.
3. The next step in this testing sequence is to test how the application handles URL's that are mistyped while being typed. The question here being how does the application handle and bad URL, hopefully the user will be returned to a logical position such as the systems main menu page.
4. Next these tests should be undertaken on a user profile that does not have access to the same areas as the previous user. The test should be undertaken in the same manner as before and all tests with results should be fully documented in the test document.

How far you take profile testing will of course depend on the complexity of your system. If multiple sections with the web application are dependant of profile access then each section will need testing. If different users have different access rights to certain actions such as create records, deleting records and running reports then these will all have to be tested. Also if user profile's also defines a user's geographical access then this must also be tested, as demonstrated next.

Profile testing access for a particular Country/Region. Take for example the URL below.

*http://www.yourapplication.com/customeraccounts/welcome.do?cty=de*

This address is for a customer accounts program on a test server and the Country is Germany (de). Testing should be undertaken on this address with user profiles that allow and do not allow access to the given Country. To confirm this procedure another Country should also be tested in the same way and all of the results should be fully documented in the test document.

These steps are very important. They confirm to the Project Leader and the customer that profile access is secure at different program levels and a Geographical level.

Another important type of test for web based applications is locale testing. This is another type of test that is becoming more common web based applications grow in number.

## Multiple Locale Acceptance Testing

Web based applications tend to be global products. The company I currently work for provide web based solutions to a global company which has business offices and data centres in over sixty countries. There are also many examples of global products that members of the public have access to, these include EBay, Amazon and Wikipedia. Your company's products may well also have such a global or at least continental reach. As a result locale testing is vitally important. Most of your web application will have a least one page which will show Date values and/or numeric values, when these pages are encountered during functional testing locale should also be checked at this point. The results should be noted and screen shots taking for the functional test area and the overall results should be recorded in this section.

It is also important to record the fact that the tests were undertaken in Internet Explorer (IE), Firefox, Chrome and other popular browsers. Also all country locales your product is likely to be used in should be tested very carefully.

## Field and Data Validation

The most common web application security weakness is the failure to properly validate input coming from the client or environment before using it. This weakness leads to almost all of the major vulnerabilities in web applications, such as cross site scripting, SQL injection, interpreter injection, locale/Unicode attacks, file system attacks, and buffer overflows.

Field and Data (n.d.) Validation is the process of applying certain rules to data within a field.

This may be a requirement or limitation on the number of characters that can are entered, such as with a password, or the assurance that the data entered falls within a certain range, such as a date.

Shown below are examples of each type of data you are likely to come across while doing Field Validation. This grid covers both field and data validation.

| Table Name | Column Name | Type | Length/Range | Field Title | Page Data Type | Result |
|---|---|---|---|---|---|---|
| tblTable | AppTitle | nvarchar | 256 | Application | Textbox | Pass/FAIL |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | Title | or List Box | |
| tblTable | JobDate | DateTime | N/A | Job Date | Date Picker | Pass/FAIL |
| tblTable | JobInt | int | Range | Record Number | Textbox or List Box | Pass/FAIL |
| tblTable | JobFloat | float | Range | Total Cost | Textbox or List Box | Pass/FAIL |
| tblTable | JobYesNo | bit | N/A | Validated | Checkbox | Pass/FAIL |
| tblTable | JobOldString | varchar | 200 | Old Text Box | Textbox or List Box | Pass/FAIL |
| tblTable | JobText | text | N/A | Full Description | Textbox or List Box | Pass/FAIL |

Data validation is confirmed by the ‘Page Data Type’ column and also by the text you are able to add to a record. If alphanumeric values are added to a textbox which is designed to accept numeric or decimal values then the program should capture this and display a meaningful and well placed error message. Error messages are covered in the next section.

Field length constraints should also be handled, an important part of the functional test is to ensure that these errors and handled and meaningful error messages are displayed.

The tables and fields used by each web application should be obtained from your company’s programming team or your company’s database administrator. With this information Field and Data Validation should then be completed, below are some examples of what should be checked, please also note that this type of testing is not just for web applications and is in fact applicable to many windows applications as well:

- Field types of Nvarchar should allow any character; the question here is the field length. What happens if the string length is longer than the actual field length? How does the web application handle this? Another important consideration with text based fields is vulnerability from Cross-site scripting (XSS). A cross-site scripting vulnerability may be used by attackers to bypass yours or your customers access controls, the affects of such attacks could result in a significant security risk if the data being exposed is highly sensitive.

  There are many different approaches that can be used when testing for XSS vulnerability and different software companies employ their own policy on the matter, below are a few examples commonly used:

  One such test for XSS vulnerabilities is to verify whether an application or web server will respond to requests containing simple scripts with an HTTP response that could be executed by a browser, for example:
  http://server/cgi-bin/testcgi.exe?<SCRIPT>alert(“Cookie”+document.cookie)</SCRIPT>
  In this example if the script was executed then this would be considered a fail.

XSS vulnerabilities also exist when the Web application under test accepts user input through HTTP requests such as a GET or a POST. The result of this request then redisplays the input somewhere in the output HTML code. Below is a simple example:
First the web request looks like;
GET http://www.web.com/page.asp?id=20&lang=en&title=Section%20Title

From this the HTML returned by the server after making this request includes;
<h1>Section Title</h1>

This means that the user input passed to the 'title' query string parameter was very probably placed in a string variable and inserted by the Web application into an <h1> tag. By providing the input, the attacker is able to control the HTML. Unfortunately if the site is not filtering input server-side a malicious user could abuse this in many ways, a good example being the attacker could inject code by breaking out of the <h1> as shown below:

http://www.web.com/page.asp?id=44&lang=en&title=Section%20Title</h1><script>alert('XSS%20attack')</script>

Such threats are serious and testing against them should always be part of the testing procedure.

- DateTime, these fields should only accept valid dates or times, anything else should be caught and cleanly handled. See error handling.

- Int, these fields should only accept whole numbers (127 but not 1234.11), anything else should be caught and cleanly handled. See error handling.

- Float, these fields should only accept whole or decimal numbers (127 or 1234.11), anything else should be caught and cleanly handled. See error handling.

- Bit, these fields should relate to Yes/No options such as tick boxes, they only accept 1 (Yes, true) or 0 (No, false).

- Text fields are multi line text fields which can take large amounts of data. These fields are used to stored data items such as descriptions and they can accept any form of keyboard character.

A consideration which the tester should consider is correct data type usage. For example if the field name if a field is DateCreated and the type is NVarChar you may think to yourself should this not be a DateTime field. Questions like this should be raised with the project leader. If they agree with you then they will raise or ask you to raise the issue with the application programmer.

The validity of each field should be verified in conjunction with Data Validation as detailed below; any errors should be recorded in your company's standard manner as soon as they are discovered.

## Error Messaging

This section involves the process of proving the application correctly displays error messages. Here we are not concerned with logic errors and java panic messages. What is of concern in this section is how good the built in error messages are? These error messages should be helpfully worded, friendly and make sense. They should also be appropriate to the error and spelling/grammar should also be good.

While functional testing the tester should attempt to invoke error messages wherever possible by entering invalid data into input boxes to check at (1) The program can handle the bad data and (2) the program responds with a well designed and helpful error message. They should also note the non appearance of an error message at a point where one would be expected.

Examples of such messages can be:

Please enter a valid date into Start Date.

Only 256 characters allowed in the name field.

## In Conclusion

In conclusion testing web based applications is a far more complex task than testing windows base programs. With web based software testing not only do you have to consider the application under task and the operating system but you also have to consider the behaviour and security issues of web browsers as well. A typical example of how these differences can affect testing considerations is data entry screens. In a winforms environment it is possible to force an edit screen open in model view, this means the user has to complete the form and either save of cancel before they can start a new operation. This means more control can be exercised over the user. In a web based application this is not possible, users are able to click the browser back button before completing the task and even attempt to open a second browser to edit the same record, stranger things have happened. In web based application you have less control over user actions therefore this fact as well as the potential impacts need to be considered when designing a test regime for a web based application.

As a result the testing regime has to be a lot more thorough and the methods employed carefully implemented. The methods discussed here are not the only ones available to software testers, indeed this is just a small selection, they are however some of the more common. These methods may not be all suitable for every test team but I suspect at least two or three of these methods are used by all good software testing teams.

Test teams have a lot to consider when testing web based applications and these do not simply include the software being developed, they also have further considerations. A major consideration is web browsers. Where relevant testing should always be mirrored in both Internet Explorer and Firefox, but what about the lesser used browsers such as Google Chrome and Safari. A small proportion of the user base might prefer using these browsers; under these circumstances those browsers require testing as well.

These and other considerations mean that software testing is a much more demanding and an increasingly more important part of the software development process. As a result the software testing regime is now a very important part of the development process. Not only should these methods be well defined and implemented, they should also be used by a well trained and intelligent team of software testers who are able to play a much more interactive role in the software development process.

## References

Functional testing (n.d.), http://en.wikipedia.org/wiki/Software_testing

Non-functional testing (n.d.), http://en.wikipedia.org/wiki/Software_testing

Regression testing (n.d.), http://en.wikipedia.org/wiki/Software_testing

Data validation (n.d.), http://en.wikipedia.org/wiki/Data_validation

K Martin (2010) “Testing web-based software - challenges and considerations”
http://www.computerweekly.com/Articles/2010/11/22/244056/Case-study-Testing-web-based-software-challenges-and.htm