Issue 31

THE TESTER

March 2010 Issue

IN THIS ISSUE:

Thursday 11 March 2010

Next Conference: Lean and Mean Testing

Keynote Presentations

- **Lean Principles in Testing, IT and Life**
- **Advancing Testing using Axioms**

Track Sessions

- **Visual Approach to Risk Based Integration Testing**
- **Customer Exprerience Testing**
- **Introducing Testing into a Scientific Software Team**
- **Agile: A Development Model or a Religion?**

Interactive Workshops

- **Learning Lean Principles through Production Simulation**
- **The Elements of Test Strategy**
- **End to End Agile: Telling a Testing Story**
- **Don’t Shoot the Messenger**

bcs The Chartered Institute for IT

Software Testing Specialist Group

Please note that any views expressed in this Newsletter are not necessarily those of the BCS.

FROM THE EDITOR

Matt Archer, Editor

Welcome to the first Tester magazine of 2010.

Our opening article is by **Tim Hunter**. Tim presents his recipe for improving software quality through a blend of traditional and progressive techniques. His new approach, aptly known as Quality Driven Development (QDD), places software quality at the centre of the development lifecycle and is well worth considering if you are currently weighing up different processes for your team.

Our second article has been written by **Stephen Allott.** Stephen shares his knowledge of agile software development, specifically SCRUM, as he take us through what it means to follow the spirit of "being agile". If you are considering adopting a more agile approach to software development, Stephen's article will make a great introduction.

If you are inspired by reading the 2 fantastic articles in this edition and would like to become a published author in *The Tester* yourself, then please email me.

I look forward to seeing you all at the conference in March. In the mean-time, happy testing!

Matt Archer

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

SIGIST 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

WEBSITE LINKS

BCS SIGiST website:
www.SIGiST.org.uk

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

INTRODUCING QUALITY DRIVEN DEVELOPMENT

Quality IT is IT that we have confidence in. We can only arrive at Quality IT by using formal, structured testing methods before releasing software into live environments. The current debate around IT quality incorrectly assumes that the 'rules of measurement' can be made 'more forgiving', when it is, perhaps, the practice of the game of IT itself that needs to be improved.

Quality IT is not just desirable, it is essential if we are to persuade business users to invest further in IT. The problem is, how do we measure success? A lack of agreed measures of success, according to the National Audit office, is one of the 8 most common causes of project failure [1]. It is obviously important to get the right balance between stifling creativity and imposing necessary quality standards. However, we need to ask the following questions: Do we really want quality? Do people actually prefer partially developed 'frameworks' that provide them with something to do?

It sometimes seems that we, as consumers, are just on the end a conveyor belt of development where we just accept the latest fad. Then, development becomes just one endless process where we 'test' products into shape by 'fix on fail'. However, are we therefore, merely 'scheduling change' and accepting that we don't have to build and test products like other industries?

The problem is, there's not much incentive to develop quality IT systems. Poorly functioning systems keep people in a job. Users get paid premium rates for developing 'skills' in 'working around' bad systems. People make a fortune out of supporting bad systems. However, we should be striving to create usable, quality systems that enable users to get involved in higher-value work.

IT development is a serious business. Developments are often potentially very risky. They could cause financial loss in a company, cause instability, damage the reputation of customer and financial services in the UK, endanger investments and jobs or cause information security breaches. However, it sometimes seems that IT professionals are still able to brush aside the need for professionalism and quality in ways that their counterparts in other professions can only envy.

Quality is no longer being taken seriously. All forms of testing are basically being made into toothless, 'rubber stamping' exercises. Over the last 12 years we have seen a major shift away from traditional views of quality towards 'progressive' views. The 'progressive school' proposes we move away from formal objective testing towards a 'continual assessment' style testing system, which has led to a decline in standards. For instance, 'A' levels [2] are now two grades easier than 20 years ago. An additional feature of the progressive approach is to turn the teacher/pupil relationship around so we start to view the teacher as being responsible for the pupil's failings.

The problem with 'progressive' quality models is that they attempt to get the consumer to consume products as rapidly as they can be produced. Technology can undoubtedly be created quickly (although there are then doubts as to its quality). However, it is not widely appreciated that there is a limit to how quickly technology can be absorbed by users. Many people are still failing to use a fraction of the functionality available in, for instance, a PC software package.

However, we can protect our investment in IT by ensuring that the measures of success remain clear and firm. With costly IT project failures and IT disasters on the increase, this is an issue of great concern to many people, including business managers, politicians and tax-paying citizens. The point is, you can't improve quality by lowering standards.

Developers should not resist formal structure being imposed on them. If IT wants to be considered to be a true profession, then it needs to mature. Equally, testers must be realistic. They must appreciate that unless something is created, there will be nothing for them to test.

Quality Driven Development

My own methodology, Quality Driven Development (QDD), provides a compromise between progressive and traditional views of quality. It is a hybrid approach that builds flexibility into a strictly controlled quality process. QDD is a modified form of Waterfall, which provides for an iterative, transparent prototyping phase before the system reaches a stable state.

QDD is different to other methodologies in that it only sets the 'clock ticking' on testing time from the point that a stable release of software has been arrived at. 'Testing', up to that point, is merely developers using testers to develop the system for them. In QDD that is called 'Development By Test' (DBT) and it is chargeable to the development, not the test budget. QDD is a useful technique to manage situations where 'buggy' software is expected.
These are the 7 principles of Quality Driven Development:

1. **Q**uality can't be improved by lowering standards: The priority is to deliver a quality, reliable, fully functioning product that will enhance the prestige and reputation of the user and IT supplier, and that will enable the user to provide a high quality service to its customers.

2. **U**ser Acceptance of software based on conformance to requirements, specification and usability. Bad functionality cannot be accepted as part of a user's job.

3. **A**udit trails of documentation (business case, requirements, specifications and test plans, test cases and test results) must be maintained.

4. **L**ifecycle: There must be transparency to the user (and project management) of the impact on the project caused by changed requirements or any incompetence on behalf of the developers. This should be facilitated by change control, defect logging systems and other management reporting.

5. **I**ncremental Delivery must not be misused. If software is shown to the user it must be made clear how much tested functionality exists (and how much critical functionality is missing) before the user 'gets on-board'. It must also be made clear what kind of environment and context the software is running in. A façade cannot be passed off as real progress.

6. **T**esting must be done independently and objectively. Working software is defined as software that has been successfully and independently tested against requirements and specifications and in test environments which are as representative of live environments as possible. Entry and exit from these environments must be under the control of independent testers.

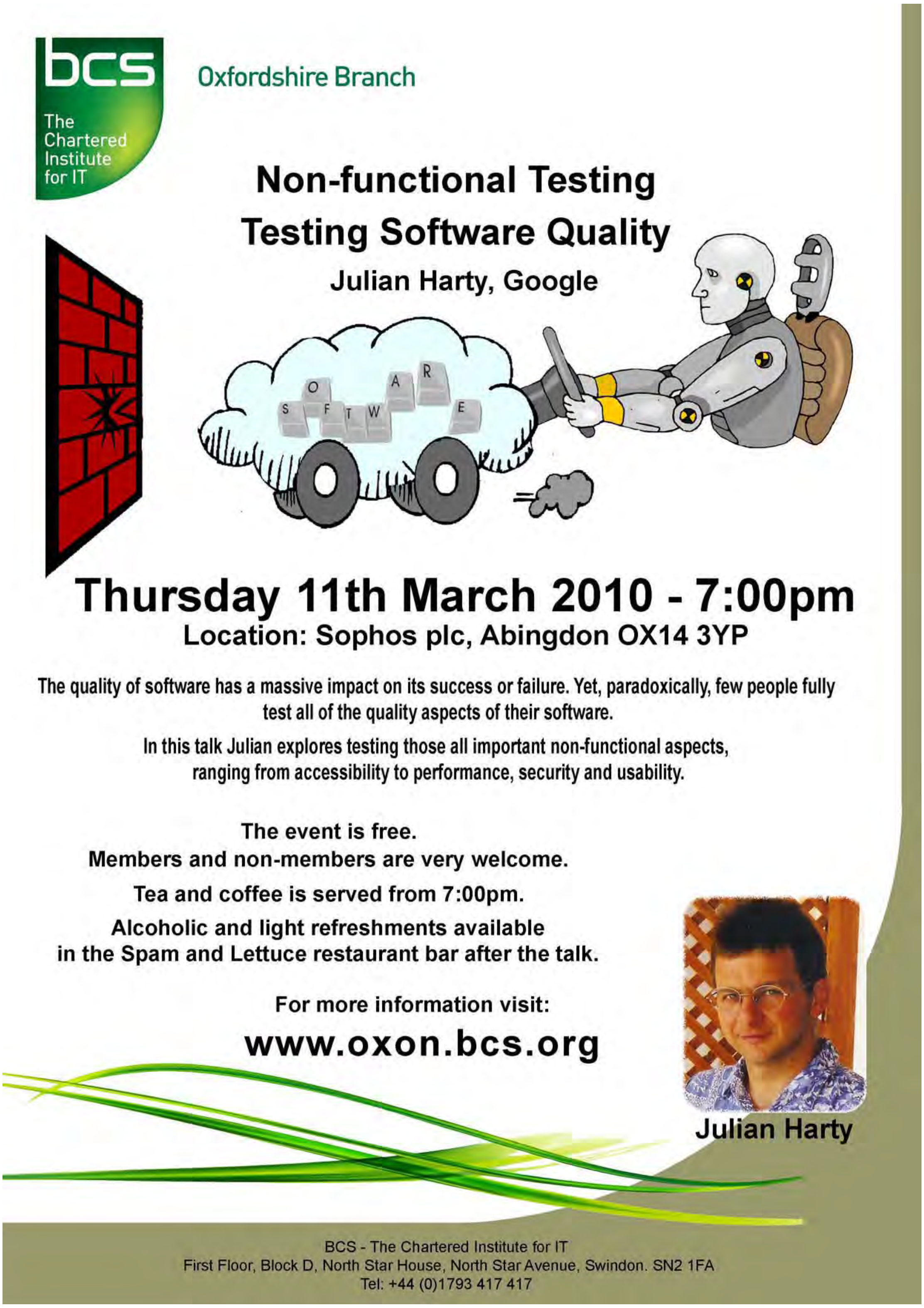
7. **Y**ield nothing re Quality. Business users have the responsibility to specify detailed requirements. Business users and developers must maintain a structured formal relationship throughout the project; otherwise the principle of independence and the contractual relationship may be compromised.

I have written a book about QDD called 'Prudent Pathways To Quality, which examines the conflict between the priorities of 'the Project' and the sometimes inconvenient truth revealed by formal testing. The book also examines how we measure success and is essential reading for anyone wanting to improve the quality of systems development.

[1] www.ogc.gov.uk, Project_Failure.pdf, Common Causes of Project Failure
[2] http://www.telegraph.co.uk/education/2540628/A-levels-now-two-grades-easier-than-20-years-ago.html. 'A level' is a final school examination, taken before entrance to University, in the UK.

Tim Hunter BA MBCS CITP PGD CCI (Open) has 30 years' experience of IT development, testing and technical support, gained in major companies throughout the UK and Europe. He has written many articles for newspapers and magazines about IT and politics. Tim Hunter's new book 'Prudent Pathways To Quality', is available now on www.lulu.com (and www.amazon.com), price £15.99 paperback, £7.99 download.

Publisher: Yorview
ISBN: 978-0-9562357-0-1
Paperback A5, 5.83" x 8.26", 237 pages



Oxfordshire Branch

Non-functional Testing

Testing Software Quality

Julian Harty, Google

Thursday 11th March 2010 - 7:00pm

Location: Sophos plc, Abingdon OX14 3YP

The quality of software has a massive impact on its success or failure. Yet, paradoxically, few people fully test all of the quality aspects of their software.

In this talk Julian explores testing those all important non-functional aspects, ranging from accessibility to performance, security and usability.

The event is free.
Members and non-members are very welcome.

Tea and coffee is served from 7:00pm.

Alcoholic and light refreshments available
in the Spam and Lettuce restaurant bar after the talk.

For more information visit:

www.oxon.bcs.org

Julian Harty

BCS - The Chartered Institute for IT
First Floor, Block D, North Star House, North Star Avenue, Swindon. SN2 1FA
Tel: +44 (0)1793 417 417

BOOK REVIEW:

HOW WE TEST AT MICROSOFT

Author: Alan Page, Ken Johnston, Bj Rollison
ISBN 978-0-7356-2425-2

It would be easy to be cynical about a book by Microsoft on testing software, but that would be a great mistake; this book describes convincingly how a large organisation improved the testing of its software, by changes to the organisational attitudes, responsibilities, and skills. It is also a valuable handbook of practical techniques and tools, described by people who actually use them in their real work. As testing continues to improve at Microsoft, cynicism becomes not simply inappropriate but also ostrich like – I can think of other organisations who need to learn the same lessons…

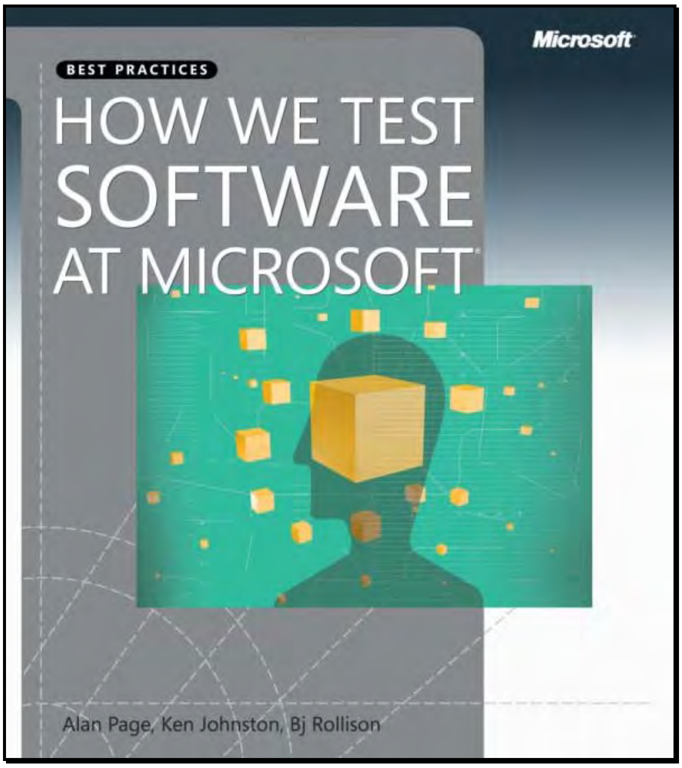
The book is divided into four main parts. The first sets the scene with the background of software engineering and software testing in Microsoft, describing the change in culture and attitude bringing the idea of a test engineer as opposed to a tester. This part is useful if you want to understand Microsoft's structure for testing, and perhaps to draw lessons for your own organisation's structure. The second part describes HOW TO test, looking at test case design techniques and the analysis of risk. Part 3 focuses on the management of the testing via tools and systems, but also includes a chapter on non-functional testing, which for me would sit more happily in section 2. The final part of the book discusses the future for testing at Microsoft.

My suggestion is that if you are doing testing or managing testing, start by reading part 2 and part 3 of the book. You will get some good insights into useful techniques and tools which you can apply to your own work, to help in designing, executing and controlling your testing, as well as in obtaining and using customer feedback. These chapters cover a wide range of techniques with good examples.

I found part 1 and part 4 less convincing on first read, but on returning to them after enjoying parts 2 and 3 realised that I was being somewhat British and reserved; the exuberance and enthusiasm that the authors display for their company and their work is genuine, heartfelt and deserved, as well as reflecting the culture of the company.

I recommend this book thoroughly; it will be a useful addition not just to your book shelf but to your work desk. It will help testers, test managers, developers and analysts to understand and apply improved test methods in their daily work.

Reviewed by: Isabel Evans, Testing Solutions Group Ltd, December 2009


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

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 SIGiST Library could help!

The SIGiST 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 SIGiST member.

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, please contact Sue Atkins on 01697 748 748 or email her at siglib@iotest.com. Happy Reading!

THE SPIRIT OF THE SPRINT:

AN INDEPENDENT VIEW OF THE IMPACT OF AGILE

Introduction

The goal of Agile is working software although as Sam Clarke pointed out at the recent SIGIST conference, “we’ve not got a great definition of what ***working software*** actually means”. SCRUM is just one of many ways, albeit a popular one, of managing an Agile software development project.

According to Martine Devos, independent consultant & certified SCRUM trainer, the spirit of SCRUM, why they did it like this, seems to have sometimes been lost in translation. Some trainers don’t always have the experience required to put across all of the nuances of this approach. This may explain why in some organisations you hear either the complaint that “SCRUM doesn’t work” or the more popular “we’re doing SCRUM”, when in fact they’re not really “doing” SCRUM. In this article, based on the teachings of Martine and my experience over the last 12 months of Agile projects, I’ve tried to explain SCRUM in terms of what I call “the spirit of the sprint”.

What is SCRUM?

The first thing to get your head around is that SCRUM is a project management framework not a development methodology. It does not tell you how to do your job. Some senior consultants and senior software testers involved in defining processes and methodologies find this odd; they are expecting SCRUM to tell you how to do this, how to do that; however this was never its original intention. It will not write the code, design the database or figure out the right user acceptance testing scenarios for your particular situation. You have to do that. Sorry.

In simple terms SCRUM asks you to develop software incrementally in small time periods known as **sprints**. These are typically from 2 to 4 weeks in duration. To build a large system you obviously therefore need to plan more than one sprint; hey this is not rocket science you know.

User stories are features of the system described concisely (usually on small post-it notes) and create what’s known as a “**Product Backlog List**” (PBL). The team selects items from the list during a planning meeting and works out what will be developed in the first sprint; you can’t have everything at once; Agile is not to be interpreted as just “faster development”; it is controlled, flexible to change and **disciplined**.

It’s important to understand the **estimating** process and the **planning poker** game. The general rule is to break the development into small pieces (**tasks**) that will take no more than 16 hours. Everyone gets a chance to vote on how long a particular task will take. Estimates are constantly revised and the **burndown** chart is used to help monitor progress (**velocity**) towards your end date and goal of “working software”.

At the end of the sprint a **retrospective** helps the team learn lessons and improve for next time.

How to get better at SCRUM

Here are just a few suggestions as to how you can get better at SCRUM:

1. The key is having “a hand picked team” of the right people with the appropriate technical skills, knowledge and personalities to succeed. The team will include testers as well as developers.

2. Over time the team is expected to be a **self organising** team.

3. The product owner (business analyst) should be a part of the team. Typically a SCRUM team will have between 5 and 9 people.

4. Only produce documents when needed

5. **Feedback** is a very important aspect of SCRUM.

6. The certified SCRUM master:

    - Has had appropriate training
    - Facilitates the daily meeting
    - Removes any impediments to progress
    - Ensures that the right pace is followed by the team

What about testing?

There are many books, conferences, tools and debates about how to test within the SCRUM however these all seem to focus at the moment on unit testing. Very little thought has been given to end to end integration testing, user acceptance testing or non-functional aspects such as performance and security. Various models have been proposed to deal with this problem and as each client situation is unique the method adopted is unlikely to follow a "one size fits all" approach and so some tailoring of the test process within an Agile development shop will be necessary.

Use of Test Tools?

Opensource tools such as Fitnesse, Selenium and WebDriver are becoming popular with the Agile development community and the major test tool players such as HP (and some of the smaller ones like T-Plan) are gradually modifying their tool sets to work within an Agile environment. One thing everyone seems to be agreed on is that you cannot do regression testing without some form of test automation suite; there simply isn't time to run all the tests manually. True, there are script maintenance issues however using frameworks (such as AXE) and a data driven approach to decouple the tests from the tool automation technology will obviously help and pay back your investment in tools many times over.

Summary and Conclusions

Each organisation needs to weigh up the advantages and disadvantages of using an Agile approach on their software development projects. It's not necessarily going to benefit all projects in all situations.

If you decide to "go Agile", it's worthwhile taking time to understand the meaning behind the manifesto, read some of the history and adopt all of the Agile principles; I don't believe you can cut corners and miss out some of the steps just because someone said "oh that won't work here".

SCRUM can deliver obvious benefits to organisations in that useful working software is delivered on a regular and predictable basis. SCRUM helps you learn as a team however problems seem to appear faster. It will very quickly identify weak players in your team and they must be prepared to develop themselves in order to keep pace with the best of the best. SCRUM cannot solve every problem in your organisation so please don't raise your expectations too high. Yes you still need capable people to analyse the business problem and come up with good testable requirements. And yes you still need good systems architecture, a solid testing infrastructure and environment, strong management and effective testers.

We'll talk about scalability and the Scrum of Scrums in the next article. Remember, you can't do everything at once.

Stephen Allott is an older, experienced, some might even say wiser, consultant within the software testing and quality assurance space. He was recently elected as a Fellow of BCS – The Chartered Institute for IT and is programme secretary for the BCS Specialist Group in Software Testing.

Stephen K. Allott FBCS CITP
Programme Secretary
BCS Specialist Group in Software Testing
stephen.allott@electromind.com
www.electromind.com
http://rapidsoftwaretesting.eventbrite.com
0773 476 1373

EuroSTAR 2010 call for papers is open!

Hardly has the dust settled from EuroSTAR 2009 when the call for submissions for the 2010 version of this conference is made. Why is this so early? The call is made to give prospective presenters time to get their ideas together, and enable a quality and varied program to be assembled. This year the Program Chair is John Fodeh, assisted on the Program Committee by a Swede (Rikard Edgren), a Dutch lady (Nathalie van Delft) and myself from the UK. John's bold aim is to make the 2010 EuroSTAR conference the best ever, and as there have some very good previous conferences, that is a tall order. However, the quality of the conference will be determined to a large extent by the calibre of the submissions. The theme for EuroSTAR 2010 is "Sharing the Passion".

Not all of those who submit an outline are accomplished speakers, and the review committee are looking for a mixture of theory and practice, advanced and introductory session to enliven the proceeding in Copenhagen around the end of November. Perhaps you have presented a session at SIGiST or in your workplace in the recent past, and your talk was well received. Then try for a larger audience. All conference attendees need a varied program incorporating key aspects of our profession, but we also need the unusual. Few who heard the session 'Testing on the Toilet' from Google co-presenters at EuroSTAR 2008 will forget the impact of the talk, but it was not a run-of-the-mill presentation. It should be noted that talks based upon experience, what I call 'war stories', are always popular and well received.

Get your thoughts on paper, and review, rework and refine the ideas. Aim to make 3 or 4 points, as it is better to have a few well thought-out ideas, rather than trying to pack in too much. Ask a trusted friend or colleague to give you their (honest) opinion. Spell check your outline, taking note of the information on length and the guidelines on the EuroSTAR website (http://www.eurostarconferences.com/). You may like to use something like WORD to format your submission, and then use cut-and-paste to submit via the web site.

You cannot be accepted for the free EuroSTAR conference slot allocated to speakers unless you submit a proposal (which is stating the blindingly obvious). The bad news is that if you are accepted, the hard work really starts. But that is into the future. Here are the key dates to note:

Deadline for submissions:	Friday 05 March, 2010
Notification of acceptance:	Friday 16 April, 2010
Finalised slide material due	Not yet finalised, but "end of August" as a guideline
Conference dates:	29 Nov - 02 Dec, 2010

Do consider submitting an outline by 5th March. Your story could make the conference extra special for a number of attendees. At the very least, it will make sure that the program review committee have something to do in March, when something like 480 submissions will be whittled down to 65 sessions for what is hoped will be the best EuroSTAR conference to date. Brits are usually well represented in EuroSTAR speakers. YOU can make sure that the speaking record is continued.

***Peter Morgan**, EuroSTAR 2010 Programme Committee Member*

CONFERENCE PROGRAMME
11TH MARCH 2010

LEAN AND MEAN TESTING

Our programme secretary, Steve Allott, is unable to write the conference summary this time round due to work pressures, so I am filling in for him. As Steve's deputy, I have been involved in pulling together the conference programmes for the last year and I feel we have a really exciting programme lined up for March.

The theme this time is **"Lean and Mean Testing"** which is very topical in these cost conscious times. We start with a keynote from Wayne Mallinson entitled '**Lean Principles in Testing, IT and Life**'. Personally, I am intrigued and cannot wait to hear Wayne's views on how to keep your life lean.

After coffee we have a choice of workshops. Paul Gerrard, who is always a popular speaker at the SIGIST, will be running one workshop session on **"The elements of Test Strategy"** and will be looking at test strategies as a journey and providing some useful guidelines of what to include. The second workshop is going to provide an opportunity to delve a little deeper into lean techniques, as Wayne is going to be leading a practical session on **"Learning Lean Principles through Production Simulation"**.

In the main auditorium and keeping to our theme of 'Lean' we have a session on a **"Visual Approach to Risk Based Integration Testing"** from Neil Pandit. Risk based testing can be a cause for conflict between project stakeholders and test teams and Neil will be presenting some real examples of test reporting to highlight risks to the project.

My colleague, Lucy Heenan, and I will be presenting a session on **"Customer Experience Testing"**, what it is all about and some thoughts on getting started in this relatively new area.

After lunch we have the 15 minute Sharepoint slot. This time Tonnvane Wiswell will be presenting a book review on "Managing the Test People" by Judy McKay.

We follow the sharepoint in the main auditorium with **"Introducing Testing into a Scientific Software Team"**. Part of our ongoing theme of getting testers and test managers to present their real experiences, in this session Chris Morris is going to recount how he managed to get a bunch of academics to become testers. Agile is still the watchword of many IT and testing discussions so Pablo Garcia Munas's discussion **"Agile: A development model or a religion?"** will really strike a chord with many of you. Pablo will be telling us about his own experience of agile projects and some key approaches for success.

Keeping to the lean and agile theme we have the first of the afternoon workshops entitled **"End to End Agile: Telling a Testing Story"** by Ant Gardiner and Tom Quinn, who will be illustrating their session with help from live demonstrations and real world stories. The other workshop will be presented by Susan Windsor, who is going to present **"Don't Shoot the Messenger"**. Susan will be discussing some useful techniques for communicating our message and will then give attendees the opportunity to practise some of them.

After the tea break, we reconvene in the main hall for the closing keynote from Paul Gerrard, who will be presenting his testing axioms or rules for testing **"Advancing Testing using Axioms"**. This will be based on his Testers Handbook which many of you will have come across and if we are lucky he will bring a few copies with him.

And finally, we will have our sponsors exhibiting in the atrium and presenting a couple of lunchtime vendor talks. Please do take the opportunity of the coffee, lunch and tea breaks to have a chat with them and with your fellow conference attendees. The SIGIST conferences are a real opportunity to network with your fellow testing professionals and share ideas and issues.

Mo Shannon
Deputy Programme Secretary

Some dates for your diary . . .

Please make a note of the dates of our forthcoming conferences as attendance is on the increase and so we'd really like you to book your place early to avoid disappointment.

Upcoming Conference Dates

Tuesday 29th June 2010
"Automation and tools"
Keynote: Mark Fewster, Grove Consultants

Thursday 16th September 2010

Wednesday 8th December 2010

Please enjoy the conference talks and workshops and please remember to make the most of the networking sessions and the exhibition.

MARCH 2010 CONFERENCE PROGRAMME

BCS SIGiST – **Lean and Mean Testing**

11th March 2010
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

08:30	Coffee & Registration, Tools & Services Exhibition opens		
09:15	**Introduction and Welcome** Geoff Thompson, SIGiST Vice Chairman		
09:30	Opening Keynote **Lean Principles in Testing, IT and Life** Wayne Mallinson, Test and Data Services		
10:30	**Networking session and commercial break**		
10:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition		
11:15	**Visual Approach to Risk Based Integration Testing** Neil Pandit, Sopra	**Workshop M1** **Learning Lean Principles through Production Simulation** Wayne Mallinson Test and Data Services	**Workshop M2** **The Elements of Test Strategy** Paul Gerrard, Gerrard Consulting
12:00	**Customer Experience Testing** Mo Shannon and Lucy Heenan, BT		
12:45	Lunch break Opportunity to visit the Tools & Services Exhibition (13.00) Lunch time vendor talks		
14:00	**The Share Point:** Book Review: "Managing the Test People" by Judy McKay Tonnvane Wiswell, The Post Office		
14:15	**Introducing Testing into a Scientific Software Team** Chris Morris, Daresbury Lab	**Workshop A1** **End to End Agile: Telling a Testing Story** Ant Gardiner & Tom Quinn, iMeta Technologies Ltd	**Workshop A2** **Don't Shoot the Messenger** Susan Windsor, WMLH Consulting
15:00	**Agile: A Development Model or a Religion?** Pablo Garcia Munas, Know-IT		
15:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition		
16:15	Closing Keynote **Advancing Testing using Axioms** Paul Gerrard, Gerrard Consulting		
17:00	Closing Remarks		

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary. Workshops will have limited places, to avoid disappointment these must be booked in advance.

ABSTRACTS AND BIOGRAPHIES

Opening Keynote: Lean Principles in Testing, IT and Life

Wayne Mallinson, Test and Data Services

LEAN manufacturing has surpassed traditional mass production techniques with higher quality, reduced costs and faster time to market. LEAN started in the automobile industry with Toyota spearheading the techniques and specific company behaviours required. LEAN has more recently shown similar positive results in other manufacturing organisations, retailers and service organisations and is now enjoying great interest in some IT-serviced companies as it improves software development and testing activities.

LEAN principles have commonalities with Agile approaches. Learning more about LEAN techniques, culture and practice can add further value to your Agile or Waterfall software development approaches.

Like many great truths LEAN principles can be used to advantage both at work and at home to help you reduce waste by converting it into shorter cycle times, higher quality products and much less stress. To do all this successfully you will need the buy-in of everyone at work as LEAN will change everything given the effort.

***Wayne Mallinson** has moved through the profession of Mining Geology to Software Testing, which he has practised for the past 22 years and still is learning more each day. He is a qualified natural scientist and testing Practitioner and is now studying for his Executive MBA degree in a quest to learn how a little bit of management science can raise the bar in software quality and productivity.*

He and his wife Jenny, have three grown up children.

Visual Approach to Risk Based Integration Testing

Neil Pandit, Sopra

The objectives of risk based testing and the interpretation of them by Project Stakeholders and testing teams can sometimes cause confusion and division. Often, Project Stakeholders are keen to adopt a process of risk based testing as the benefits are easily promoted: reduced testing costs through targeted testing and reduced risk of failure in production. However, from a test team perspective, it is seen as an opportunity for management to do less testing and thus be perceived as increasing the overall risk in production.

As a result, in projects where significant change is required, integration testing can be particularly difficult and Test Managers are presented with immense challenges. Not least of these is the visibility around how much / little testing to do in order to satisfy all parties. In addition, in an environment where progress reporting still favours the traditional “progress against planned”, neither Stakeholders nor the test team are being fully informed of the risks that have been mitigated.

Using real examples this presentation highlights a possible solution via a practical approach to risk based testing and reporting for complex networks of systems and interfaces. It provides an objective and visual representation of potential risks through the use of an annotated system architecture diagram. The process considers both the business impacts and technical complexities of the systems, with the architecture diagram being mutually agreed by both Project Stakeholders and the test team.

This system architecture diagram will not only recommend the order in which systems and interfaces are to be tested, but provide an objective priority of fixing new and existing defects. Finally, this presentation will show how testing progress and the associated residual risks can be reported visually, thereby satisfying both Project Stakeholders’ needs and addressing the test team’s concerns.

Neil Pandit *is currently a Senior Test Consultant at Sopra Group, providing consultancy and thought leadership across all industry sectors. With 14 years IT experience, the last 8 years he has specialised in Test Management and Consultancy. Starting his career in development, Neil rapidly progressed from managing testing teams to Senior Test Management Roles for major financial institutions and system integrators. Neil is currently involved in the practical application of risk based testing.*

Customer Experience Testing

Mo Shannon and Lucy Heenan, BT

Customer Experience is a relatively new area of business focus and trying to test it is even newer. This presentation will explain what we mean by the customer experience and some of the problems of trying to define it, design it and test it. We will look at the differences between a traditional test approach and one where we focus on the Customer Experience. We will also demonstrate how different customers can need a different customer experience. Finally we will consider how we can get our traditional testers to accept and adopt this radically different approach.

Mo *has been working in BT since she graduated. Mo spent the first 9 years as a junior manager working in planning and customer service before returning to her degree subject in IT. Mo spent 20 years in IT and 13 of those have been as a test specialist. She has accreditations in Software Testing from ISEB and in Project Management from the APM. Mo has played an active part in the formation of the E2E Testing Professional Community and regularly represents BT at external events and conferences. She is a member of the BCS and is currently the Deputy Programme Secretary for the Specialist Group in Software Testing.*

Introducing Testing into a Scientific Software Team

Chris Morris, Daresbury Lab

The Protein Information Management System is being developed by an inter-disciplinary team that is scattered among five academic institutions. It must be highly reliable, and usable. Over time, the team have become effective at quality assurance, using techniques including error seeding, static analysis, review, and test driven development. The presentation will discuss the culture change process necessary to introduce testing practices.

Chris Morris *is project manager for a grant funded project to develop a laboratory information management system for protein scientists. He began his working life as a programmer working in assembly language, spent fifteen years working in other industries, then returned to programming in 2001. He soon realised that the hardest part of the job was not the coding, and began to study how to for manage software projects.*

Agile: a Development Model or a Religion?

Pablo Garcia Munas, Know-IT

Agile and Scrum has extended over the world and many developers and testers accept it like the “only truth” and a religion.

This presentation highlights first the problem that we have many young fanatics in development that would go very far to defend their beliefs.

The second section talks about the speakers experience of many successful projects that have been/are using parts of Agile and Scrum as a project and development model. The most common solutions are using Agile in combination of a more traditional model, some of them are explained.

The most common mistakes and the keys for success are presented at the end.

Pablo Garcia Munos *has been in the testing industry since 1996 and is a known profile in the Swedish testing industry. He sits in the board for SAST (Swedish Sigist) and has done so for the last six years. Pablo is an accredited ISTQB Foundation teaches and gives courses on a weekly basis. He is currently employed at*

Knowit AB and works partly as a consultant and partly as competence responsible. He likes to speak about real things.

Advancing Testing Using Axioms

Paul Gerrard, Gerrard Consulting

Test Axioms have been formulated as a context-neutral set of rules for testing systems. The Axioms evolved from a series of blogs posted by Paul in 2008, eventually being documented in “The Tester’s Pocketbook”, published in October 2009. The sixteen Axioms provide a framework for all systems testing and represent the critical thinking processes required to test any system. There are obvious opportunities for further research and potential to advance the practice of testing using them:

- The Axioms enumerate the key areas of test strategy and provide a checklist of concerns to be addressed in any test approach.
- Any company can use the Axioms as the basis of context-neutral testing assessment and to identify areas requiring improvement without using artificial, inappropriate maturity levels.
- The Axioms provide a framework for organising testing and teams in new ways, in Agile and other environments.
- Axioms define sixteen skills areas required by testers, and could form the basis of a tester development framework and possibly a certification regime that has meaning to practitioners.

This talk introduces the Axioms, the thinking behind them and how they can be used to reframe and potentially solve the most urgent problems in our discipline.

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

Paul is Principal of Gerrard Consulting Limited, a Director of Aqastra Limited and is the host of the UK Test Management Forum.

Workshop M1: Learning Lean Principles through Production Simulation

Wayne Mallinson, Test and Data Services

Lean principles were originally crafted on the factory floor. It is therefore fitting that these principles are taught in the context of a production environment albeit a simulated one.

In this workshop, delegates will learn about flow, value stream mapping, value analysis, resource levelling, the elimination of waste and the use of kanban. Lean measures of cycle time, cycle efficiency and percentage complete and finished will also be taught in the context of the simulation exercises.

At the end of the session a discussion will be held to draw the similarities and differences between Agile projects and the Lean approach.

***Wayne Mallinson** has moved through the profession of Mining Geology to Software Testing, which he has practised for the past 22 years and still is learning more each day. He is a qualified natural scientist and testing Practitioner and is now studying for his Executive MBA degree in a quest to learn how a little bit of management science can raise the bar in software quality and productivity.*

He and his wife Jenny, have three grown up children.

Workshop M2: The Elements of Test Strategy

Paul Gerrard, Gerrard Consulting

Test Strategy is a much misunderstood concept. When talking about a strategy for testing, most professionals think of a document, perhaps structured in accordance with IEEE 829 – the Standard for Test Documentation. But test strategy isn't a document, to be produced by rote, to be distributed, reviewed, approved – and then ignored. A better way of looking at test strategy is as a thought-process and as a journey.

Test strategy is a thought-process because if every project is unique, then every strategy is unique also. Using off-the-shelf solutions or previous test strategies, i.e. someone else's thinking, will simply not do. You have to think through, often from first-principles, exactly what you are trying to achieve with testing and why, before you can figure out how.

Test strategy is a journey because not every question of strategy can be answered early on. Much of the information required to complete a strategy may not be available until testing (whatever that is) begins. Agile approaches seem to leave little room for test strategy (and test management in general), but the goals of testing are unchanged. **Test strategy is appropriate for all contexts and approaches** because it helps stakeholder to understand what is and is not being achieved on their behalf.

This workshop considers what test strategy really is. The session promotes the Test Axioms from "The Tester's Pocketbook", as a checklist of items to be covered by a strategy and provides a set of questions to focus your thinking on each. A case study will be used to answer the questions and provide a basis for discussing what strategy should be used.

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

Paul is Principal of Gerrard Consulting Limited, a Director of Aqastra Limited and is the host of the UK Test Management Forum.

Workshop A1: End to End Agile: Telling a Testing Story

Ant Gardiner & Tom Quinn, iMeta Technologies Ltd

This presentation will demonstrate the entire lifecycle involved in delivering a story within a sprint. From the definition of the story to the delivery and sign off they will explore the whole process with live demonstrations, detailed explanations and real world stories.

The presentation will flow as follows:

- Introduction to StoryTeller
- Creation of a story within the sprint along with acceptance criteria
- Creation of story acceptance criteria and associated tests
- Test Driven Development of the unit tests before the code is written
- Running of tests to prove failure
- Creation of the code
- Re-running of the test to show passes (both unit and acceptance tests)
- Potential refactoring
- Creation of manual test cases
- Build and release to test environment
- Running of test cases
- Running of exploratory testing

- Closing of story as done

All of the above will be done as a live demo.

The presentation is intended to show how a typical agile project could run but also show how the programmer and tester are merging closer together to deliver working software at the end of a sprint. As a side effect of this the presentation will also show how the traditional tester will need to move to a technical angle, yet at the same time showing that bridging the gap between technical and none technical needn't be so daunting.

The StoryTeller software is relatively new to the market and this presentation is a good opportunity to show this tool to the agile community.

The presentation is delivered by two very experienced people who both approach agile projects from different backgrounds, mindsets and skills sets but work together to deliver consistently good quality software. This gives the presentation a balanced view and also shows how the two disciplines can work together to solve common agile development problems.

The demonstration will be live adding an extra dimension to the topic rather than a series of slides and theory.

Tom Quinn *works as a Technical Architect for iMeta Technologies in Southampton. Tom started out as a C++ programmer, specialising in business to business e-Commerce applications.*

Workshop A2:
Don't Shoot the Messenger

Susan Windsor, WMLH Consulting

How many times have we taken criticism for being the bearer of bad news? Did we create the defects (of course not!); did we do our best in the time available (of course we did!). So, how can we move forward from here? Why are we misunderstood and not appreciated?

Getting our message over effectively requires others to receive it in a way they can understand. So, one key area we all strive to improve in the testing profession is in the area of communications. Being able to communicate effectively can enhance your career in your current organisation, improve your value as potential employee and provide you with greater self-confidence.

With the industry emphasis moving to goal based testing (as an enhancement to risk based testing) it's critical that we fully engage with stakeholders to appreciate their goals, include them in our test approach and perhaps most importantly, to provide them with sufficient and appropriate information to take critical management decisions.

This workshop will take two elements of effective communication:

- The science of persuasion
- The art of story telling

Individually, each of these techniques can significantly improve your communications skills. This workshop will introduce each element, provide reference material for further learning, and allow attendees to practice them all using practical exercises in the safe environment of the workshop. The workshop goals are to:

- Help you to increase stakeholder's confidence by learning how to present your judgement in a more appropriate way

- Learn techniques to help you obtain the information you need to create a successful test approach

- Improve your career prospects by adopting more effective communications techniques

***Susan** is the Managing Director of WMHL Consulting Ltd. in the UK, delivering strategic testing consulting services and a Director of Aqastra, retraining business administration staff to become acceptance testers. Prior to setting up her own business, Susan managed the Testing Service business for IBM. Susan has spoken at many industry conferences in the past, including EuroSTAR, SQSTest in London, Softest in Ireland, SIGIST in London and Expo08 in Madrid.*

Issue 33

THE TESTER

June 2010 Issue

IN THIS ISSUE:

Tuesday 29th June 2010

Next Conference: Testing Then and Now

Keynote Presentations

- **"Testing is ..."**
- **Holistic Testing after the Fact**

Track Sessions

- **The Future of Testing**
- **Dev and Test, Better together**
- **A Brief History of Test**
- **Alternative Testing: Do We Have to Test Like We Always Have?**

Interactive Workshops

- **EuroSTAR Testlab**
- **Maximising your Career Potential in Testing/QA**

bcs The Chartered Institute for IT

Software Testing Specialist Group

Please note that any views expressed in this Newsletter are not necessarily those of the BCS.

FROM THE EDITOR

Matt Archer, Editor

Welcome to the June edition of The Tester.

Our article this month has been written by **John Reber**, an independent test consultant from the UK. In his article, '**Test Driven Development – do testers still own the test tools?'** John explores the relationship between testers and developers on an agile project, including the division of testing related duties and the tools used to support them.

If you are inspired by reading John's article and would like to become a published author in *The Tester* yourself, then please email me.

I look forward to seeing you all at the conference in June. In the mean-time, happy testing!

Matt Archer

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

SIGIST 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

WEBSITE LINKS

BCS SIGiST website:
www.SIGiST.org.uk

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

TEST DRIVEN DEVELOPMENT – DO TESTERS STILL OWN THE TEST TOOLS?

Test Driven Development (TDD) is a practice associated with a number of Agile flavours. It has become an increasingly popular method for integrating quality into the process of software development at an early stage of the project lifecycle. In this article I ask the question which members of the project team should own the tools that are used to drive the automated testing process within TDD? Are they now solely developer tools or should the testers be the owners?

This article is based on a presentation given by John Reber at the London Tester Gathering in Jan 2010.

Agile, TDD and the tester

Testers have always advocated early involvement in the lifecycle of a project through the likes of the V model. However, the V model cannot be successfully applied without buy-in from the entire project team - invariably this just isn't the case. Ask the average developer or project manager what the V model recommends and you are likely to be left with a number of quizzical looks.

The Agile approach on the other hand can be viewed as an answer to a QA analysts long held dream of being involved in the project at an early stage - the methodology is practically demanding it. Testers can benefit from the fact that Agile is a widely supported method which actively promotes involvement from all project members at most stages of the iterative process whether it is at project planning, estimating, participating in story huddles or attending retrospectives.

Testers not only attend these sessions but they are expected to actively contribute. In an Agile environment the tester is fully expected to utilize all the skills they have traditionally performed in testing plus a whole lot more. Not only does the tester have to rise to the challenge of these additional demands but the collaborative nature of Agile means that roles on the project can blur.

Nowhere is this truer than in the case of Test Driven Development (TDD), a method of development practiced in a number of Agile flavours such as eXtreme Programming (XP). TDD gives the programming team and the tester the assurance that potentially all written code is covered by a test, many defects are caught early and by extension a greater level of confidence in the code that is being delivered.

Increasingly TDD is being extended to the point that not only are programmers writing unit tests prior to the coding effort but functional acceptance tests are also being integrated into the process. This has been termed Acceptance Test Driven Development (ATDD). For some projects this additional effort is crucial to truly satisfying the requirements specified by the customer.

TDD and specifically ATDD raise a number of questions for the tester. At what point should the tester be involved in creating the automated tests for the story in play? Whose responsibility is it to code the automated functional tests? And who owns the test tools that drive the process?

TDD and the test tools

Whilst Agile often maintains there are no specialists, in the traditional waterfall approach the project roles tend to be clearly defined. In the case of test automation members of the QA team use tools, usually of the record and playback variety, to replace copious time consuming manual regression test suites. Because of the test effort required to create and maintain these suites the automated testing process and its respective tools are almost exclusively owned by the test team. Developers generally have minimal input into the process.

On Agile projects the approach to testing and test tools differs. There are a plethora of open source tools that support the TDD process. Programmers have the xUnit family to create unit tests as well as a multitude of tools to create mocks and stubs for integration tests. Business analysts and even Product Owners may be involved in creating functional tests using tools such as Fitnesse. Then there are those tools used by both testers and developers, such as Selenium and Webdriver, which add further sophistication to the creation of automated functional tests.

The increasingly common use of these open source automated testing frameworks has added new complexities for testers. Testers have to consider some of the following questions when utilising these tools:

- What technical knowledge is required to use these tools up?
- Where should the tools various property files, logs, user extensions etc be located in the project architecture?
- Who needs to be involved? Developers, system architects, system administrators, build/configuration administrators?
- How do we integrate the tests into some sort of Continuous Integration process?

With these questions in mind it's clear that the use of test tools within the Agile practices of TDD and Continuous Integration (CI) is not a solitary activity conducted by the test team. This being the case let us look at the high level process of TDD and see where the tester can add value.

TDD – the process

The developer(s) has a story that has been planned in for the iteration/sprint and is ready for play. The functional acceptance criteria have been added by the Business Analyst or Product Owner, and from this the tester has extrapolated the test scenarios perhaps in conjunction with the Product Owner/BA.

The developers will use a unit testing framework to unit test their individual lines of code. They add their assertions, verifies etc. They may also write some integration tests. As an aside, there may be some value in the testers viewing these unit tests prior to code check in or at least stepping through the code with the developer. This exercise will often provide a quality audit of the unit tests and allows the coder to explain to the tester what the test coverage is up to this point.

Whilst viewing unit tests may not be mandatory for a tester it is essential that the tester is the guiding force when it comes to functional test scenarios. So let us assume that for this story all our test scenarios are candidates for functional test automation. We want these tests to run as part of CI giving us a higher degree of quality and confidence prior to us perhaps doing any further exploratory or manual testing. But who should script and code these tests?

I would argue that the tester should if they have the technical know-how. The UI presentation layer may not have been finished, all the values and targets to enter into our script may not yet been known, the developers probably haven't confirmed most of the implementation yet, but the onus is on the QA team to at least determine the structure of the automated testing. This being the case, if its not possible to write the complete functional test up front then the tester should write a draft structure and then collaborate with the developer to flesh out the test as the coding of the story progresses.

To summarise the involvement of the QA in TDD:

1. A story is ready to be played. The developer writes unit/integration tests. Tester codes or drafts the functional automated test.
2. As the story is coded the developers and tester collaborate to flesh out the functional test.
3. On completion of code effort the developer then demo's the working tests on their local environment to the tester prior to checking in their code and tests.
4. All being well the tests pass the CI build. Testers may now do further manual and/or exploratory testing as they see fit.
5. The QA team may also use some or all of the functional automated test to integrate into a larger regression script which can be run on a standalone CI server or perhaps on a separate QA environment.

Blockers and impediments to the process

For various reasons the process does not always work this smoothly. Some QA teams may simply not have the manpower to manage all the demands of an agile project. Or there may be a skill deficit in the QA team with no one confident enough in using the test tool of choice. If preparing automated test scripts up front is a problem then perhaps provide the developers written test scenarios from which they can write their own test scripts. However, it is then even more essential that the tester sits with the developer prior to completion of the story to ensure they are satisfied with the automated implementation of their written test scenario.

Here is a selection of some of the other issues testers may need to consider and some potential solutions:

The CI build may eventually have a large and unwieldy number of functional acceptance tests which really slows the build down?
The testers should help the team provide a solution. Perhaps the functional tests can be run on a separate CI box. Perhaps some of the low priority tests can run less frequently? Some tests may no longer be valid and can be removed from the build.

How much technical & coding expertise does today's agile QA need? What if a decision is made to use a different tool?
One solution is to have a technical tester on each team. The preferred solution would be for all agile testers to get skilled up.

A 'green' build does not mean your test has been run.
Make sure you know that the tests are indeed running and not being ignored. Perhaps some sets of tests have been deleted or disabled during code refactoring or bug fixing. There are many ways for a test to be ignored, set to pending or pulled out and it's up to the tester to know when this happens. The test team should have an awareness of the entire test coverage.

Cry wolf syndrome.
There is a risk that badly written tests that regularly generate false failures will be ignored, so that when a real failure occurs it may not be detected. Likewise a high number of passing unit tests may bring a false sense of security and a danger of complacency, resulting in less additional QA activities.

'Testers drive the testing process'

Perhaps asking the question who owns the test tools is misleading. It is clear that the sphere of testing no longer belongs exclusively to the QA team and by extension the test tools now belong to the entire project.

Yet whilst the testing of an Agile project is a team exercise there are inherent risks around diminishing levels of quality when developers are expected to write their own functional acceptance tests. Testers, acting as pre-sign off stewards of quality, should be instrumental in writing tests based on the customer requirements, with or without the tools.

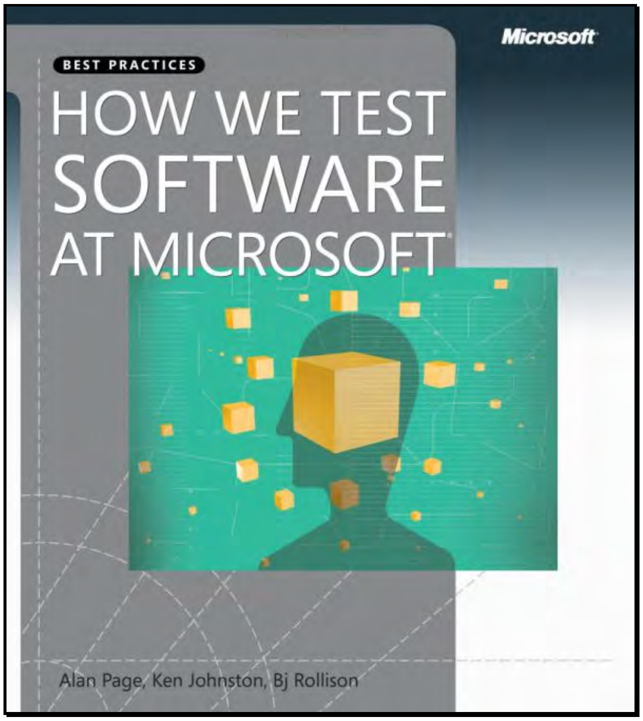
Ultimately it pays to have the test experts provide the guardianship of the testing process. This is far more important than disputing the ownership of the test tools themselves.

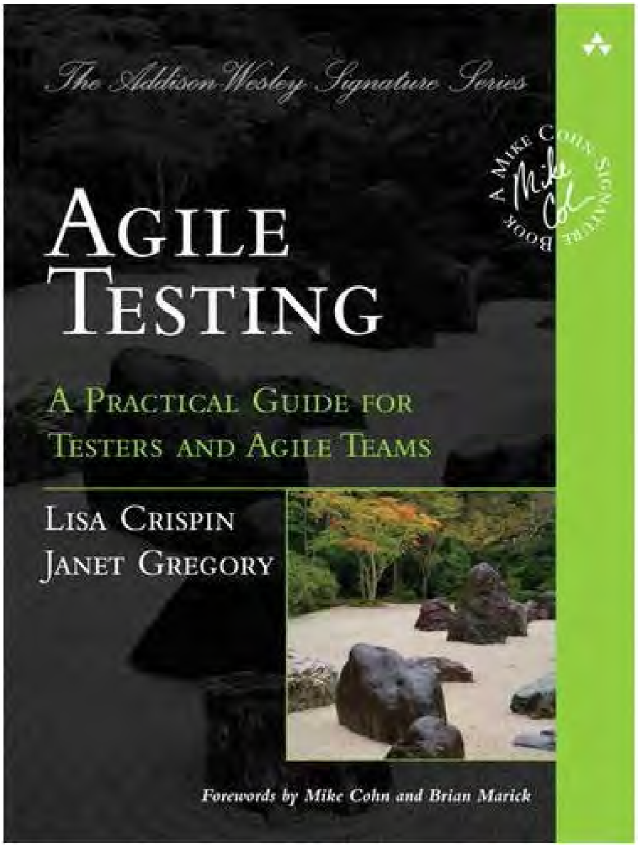
John Reber is an independent test consultant currently working on a large distributed agile project. He also provides a one day practical agile QA course.

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 SIGiST Library could help!

The SIGiST 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 SIGiST 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, please contact Sue Atkins on 01697 748 748 or email her at siglib@iotest.com. Happy Reading!

JUNE 2010 CONFERENCE PROGRAMME

BCS SIGiST – **Testing Then & Now**

29 June 2010
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

08:30	Coffee & Registration, Tools & Services Exhibition opens		
09:15	**Introduction and Welcome** Stuart Reid, SIGiST Chairman		
09:30	Opening Keynote "**Testing is ...**" Mark Fewster, Grove Consultants		
10:30	**Networking session and commercial break**		
10:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition		
11:15	**The Future of Testing** Bart Knaack, Logica	**Workshop MA1** *(Full day, both morning & afternoon sessions)* **EuroSTAR Testlab** James Lyndsay, Workroom Productions	n/a
12:00	**Dev and Test, Better together** Kevin d'Souza & Duncan Millard, Microsoft		
12:45	Lunch break Opportunity to visit the Tools & Services Exhibition. 13:00 - Lunch time vendor talks.		
14:00	**The Share Point:** Book Review by Steve Allott, ElectroMind **'User Acceptance Testing - A Practical Approach' -** James Windle		
14:15	**A Brief History of Test** Dorothy Graham, Independent	**Workshop MA1** *continued* **EuroSTAR Testlab** Bart Knaack, Logica & James Lyndsay, Workroom Productions	**Workshop A2** **Maximising your Career Potential in Testing/QA** Jennifer Lumley, e-Assurance & Jon Tyler, E-Resourcing
15:00	**Alternative Testing: Do We Have to Test Like We Always Have?** Julian Harty, Google		
15:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition		
16:15	Closing Keynote **Holistic Testing after the Fact** Clive King, Sun Microsystems		
17:00	**Then and Now - Three Generations of Testing** Dorothy Graham, Independent, Julian Harty, Google & Mike Davis, Google - Closing Remarks -		

17.15 – 20.00: Evening drinks & nibbles to celebrate the BCS Software Testing SG's 21st Birthday

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops places will be allocated on a first come first served basis on the day.

Opening Keynote: "Testing Is…"

Mark Fewster, Grove Consultants

Software testing is often seen as a searching activity in which teams of testers pit their wits against the bugs in a glorified game of hide and seek. It can also been seen as a lone occupation: one tester searching for the proverbial needle in a hay stack. Besides searching, testing can be seen as an activity that measures the software, testers striving to assign numbers or qualitative measures that convey useful information. Another view has us seeing testing as sampling since we cannot test everything but must be satisfied with a fairly small subset of things from an unbounded number of possibilities. Yet other views of testing see us checking and assessing the software. Checking software against some known or understood benchmark or assessing software using our own individual or collective judgement.

Each of these views of testing open a window on other worlds in which the activities are also common but are somehow perceived differently because software is not involved. By looking at how these same activities are performed in other worlds we can gain some fresh insights into ways of doing our testing. We can gain a deeper understanding of what we do and glean ideas to help us improve our testing activities.

***Mark Fewster** has nearly 30 years of experience in the software industry taking on roles from programmer to development manager, tester to test consultant and trainer.*

With Grove Consultants Mark specialises in providing consultancy and training in software testing. He has published papers and co-authored a book with Dorothy Graham, "Software Test Automation".

Mark has served on the committee of BCS SiGiST and is a contributor to both the ISEB and ISTQB software testing certification schemes.

The Future of Testing

Bart Knaack, Test Lab

One of the big sources for the future of testing is academia. New developments being initialized in universities and research centers, find their way into the state of business within time. A major topic in software testing (or quality improvement in general) in academia at this moment are model based techniques. These techniques are currently starting to emerge in industry albeit reluctantly. However , since software systems are getting increasingly more complex, these techniques will become vital to ensure that the overall quality of these systems can be validated.

In this talk Bart Knaack discusses his personal experiences with model based techniques for quality control, with their advantages and disadvantages. The outcome of these techniques can be used to improve the quality of the software. He gives insight in the hurdles to be taken, before these techniques can be applied in a broader perspective.

During his talk the following techniques are presented together with examples of their usage:

- Model Based requirements validation: By modeling requirements, we can use model validation techniques to find errors in the requirements at an early stage.
- Model Based code correlation: Once the requirements of a system have been modeled we can use code correlation techniques to inspect the structure of the code, with respect to the models.
- Model Based Testing: Setting up tests based on (formal) models of the system. Distinction can be made between:

    - Model Driven Testing (online): The model will directly be connected to the system under test to automatically execute the tests.
    - Model Based Test generation (offline): Tests are generated from a model to be executed at a later stage.
- Model Checking for software validity: Model checking techniques can also be directly applied to 'real-life' software to validate the software.

At the end some insight or given in current joint efforts with academia to come to a broader application of these techniques in the near future.

Bart Knaack *is senior test advisor at LogicaCMG. Before this he has done research on model checking at the University of Eindhoven and on Model Based Testing at Lucent Technologies. Currently he is involved in the research on Model Based Quality Improvements at LogicaCMG in collaboration with the Technical Universities in the Netherlands, participating in the D-Mint project. He is a frequent guest lecturer at high schools and universities in the Netherlands.*

Dev and Test, Better Together

Kevin d'Souza & Duncan Millard, Microsoft

Developers and Testers seem to live on different planets with no shared philosophy, understanding or tools for a common approach to building quality software. Without a shared philosophy, the partnership of developers and testers can deteriorate into "sides", which causes software quality to degrade. Absence of shared understanding has enabled testers to create a world in which developers have little or no space. Testers don't see a bug in the same way a developer does and are not empowered enough to share this data across the teams.

For the longest time we've continued to contribute to this divide and live in a world that separates developers and testers. And yet, the trend in software is towards those that require more frequent and sophisticated interaction between developers and testers.

In this session a Dev lead and a Test lead will look back at a journey that brought to life this difference in thinking, for them. Together, they will delve into how best practices, toolsets and techniques they put in place, helped form the basis for these distinct perspectives to co-exist and enhance the quality of software. They will also introduce some of the Visual Studio 2010 offerings that are designed to bring a software delivery team a step closer to the "Dev and Test, better together" experience.

Kevin d'Souza, Microsoft *currently works in the capacity of senior software consultant at Microsoft. With 11 years of IT experience, the last 5 years he has specialized in Test Automation, Management and Consultancy. Starting his career in development, Kevin rapidly became closely involved with test teams, and progressed from test automation, to managing test teams to senior consultant. He is a strong advocate for building a test team that strongly complements a development team.*

He and his wife have two toddlers, both girls and are expecting their third, a boy, in August.

Duncan Millard *has been working in the IT industry for 12 years and is a senior development consultant with Microsoft. He is currently development lead on a large, complex project on which a really strong test team plays a crucial part in ensuring the quality of our delivery. Outside of work, he is interested in the latest technologies and gadgets, as well as being a huge sports fan.*

A Brief History of Test

Dot Graham, Independent

How long have you been in software testing? What was testing like when you first encountered it? How did the discipline of software testing get to where it is today, and where is it headed in the future?

In this presentation, Dot Graham takes a look back at software testing from a personal perspective as well as a general view. Each decade has had its own “hot topics” in computing in general and in testing. These concerns had a significant impact on the testing at that time, and sometimes still influence testing today. In this 21st anniversary year, we will look particularly at what testing was like at the time this specialist interest group in software testing was started.

Some things have changed dramatically over the past 40 to 50 years, but interestingly, some things have not changed – why is this, and should this be a cause for concern?

We will finish by looking at what the future might bring for software testing.

Dot Graham *has been in software testing for over 30 years, and is co-author of 3 books (Software Inspection, Software Test Automation and Foundations of Software Testing). She is currently working with Mark Fewster on a new book: Experiences in Test Automation.*

She founded Grove Consultants in 1989 but in 2008 returned to being an independent consultant (semi-retired). She is a popular speaker at conferences, seminars and company events. Dot helped to start testing qualifications in the UK and helped develop the ISTQB Foundation Syllabus. She was Programme Chair for EuroSTAR in 1993 (the first) and 2009.

She holds the European Excellence Award in Software Testing, and her main hobby is choral singing.

Alternative Testing:
Do We Have to Test Like We Always Have?

Julian Harty, Google

Are the “old ways” always the “best ways” to test? Julian Harty shares his thought-provoking ideas on when traditional testing is—and is not—appropriate and poses alternatives for us to consider. For example, what might happen if we choose not to test a product at all? Perhaps the benefits of earlier delivery would outweigh the cost and delay that testing imposes. If a key goal of testing is to provide answers to quality-related questions about a product, are there alternative information sources for answers—say, from live experiments in production? How do you know whether your testing approach is really efficient and effective, especially if you already consider yourself a testing expert? Can your testing knowledge and experience blind you to alternative strategies? One option is to put yourself to the test. For instance, you could more objectively evaluate your skills by working on a crowd-sourced test project. Come, listen, join in, and leave invigorated with a fresh perspective on how you can become a better, more aware, and more astute tester.

Julian Harty *is passionate about finding ways to get software and technology to work for users, whatever their needs and capabilities e.g. to use mobile phones as extensions of a blind person's senses. As part of his work he's been actively involved in software testing since 1999 to help improve the quality of the software he's involved in. Until recently he was the first Test Engineer in Europe for Google, where he worked for 4 years.*

Over the years he's learnt plenty about testing techniques that work, some that didn't, and ways that automation can help. In the process he started writing and sharing material with other people who were interested in testing software. That work has been shared at workshops and tutorials around the world, and his work on test automation techniques for mobile phone applications ended up being published as a book, and online on http://www.stickyminds.com and at http://sites.google.com/site/mobilewirelesstestautomation/ .

He's an active contributor to several open-source testing and development projects http://code.google.com/u/julianharty/ and on software testing e.g. he was on the examination panel for ISEB for 3 years and has co-written over 100 pages of material on non-functional testing available at http://www.commercetest.com/

Closing Keynote:
Holistic Testing after the Fact

Clive King, Sun Microsystems

Most definitions and preconceptions about testing are focused prior to the release of a product. Issues rooted in the interaction between multiple components often only surface in production as the customer stresses the wider system in ways which were not envisaged. Modern Enterprise class solution stacks are built from in excess of 100 millions lines of code from more sources than you can count on both hands. When the solution fails or fails to perform, business need often dictates that recovery takes priority over the need to establish root cause which is subsequently deferred to pre-production or lab environments.

This talk examines the issues around reproducing complex problems rooted in complex multi-vendor solutions outside the production environment. How can we improve our chances of reproducing the same issue and what lessons can be integrated into pre-released lifecycle?

Dr. Clive King *is a Senior Staff Engineer at Sun Microsystems working in the support organisation. His core role is the diagnosis of performance, availability and data integrity issues for high end customers.*

He is a B.C.S. Fellow and a member of the B.C.S. Academic Panel.

Workshop MA1:
EuroSTAR Testlab

Bart Knaack, Test Lab

The Test Lab is an exciting, interactive new addition to the Sigist conference, a replica of the new addition to Eurostar 2009. A specially set up, dedicated environment, the Test Lab provides a real system to test, and the kit needed to test it. Open all day this means that, you'll be able to get direct hands-on experience at the conference itself. On top of this, the Test Lab will host a sparkling combination of open-space sessions, guided experiences and charters to give you the opportunity to try out new ideas with colleagues, experts and tools. We hope the Test Lab will be a fascinating and challenging environment – and we'll be keeping a public bug log, test repository and real-time metrics throughout to help you learn from each other.

We are James Lyndsay and Bart Knaack, and we're dedicating our time at SIGIST to the Test Lab. At the EuroSTAR 2008 closing party, we found ourselves talking about the successes and highlights of the just-gone conference. We both wanted to get back to our workplaces to try out the ideas – but became fascinated by the idea of having some sort of a shared workspace at the conference itself. From then onward the testlab has become a success by itself!

Come and bring your laptop to the testlab and you will have access to the code and to the live bug log. We'll give the lab over to anyone with something to do or show in Open-Space sessions, and we know that you will build on each other's work to reach extraordinary insights, but we will also provide charters, assignments and material to work on in open sessions.

PLEASE NOTE: The test lab is a "hands on" practical workshop and you will be expected do some testing exercises throughout the day. If you wish to sign up for this workshop on the day please bring along your laptop or any other suitable device (e.g. iPhone, iPad) that is able to run a web browser. Places are strictly limited and will be allocated on a first come first served basis.

Bart Knaack *is senior test advisor at LogicaCMG. Before this he has done research on model checking at the University of Eindhoven and on Model Based Testing at Lucent Technologies. Currently he is involved in the research on Model Based Quality Improvements at LogicaCMG in collaboration with the Technical Universities in the Netherlands, participating in the D-Mint project. He is a frequent guest lecturer at high schools and universities in the Netherlands.*

Workshop A2:
Maximising your Career Potential in Testing/QA

Jennifer Lumley, E-Assurance & Jon Tyler, E-Resourcing

Being a competent tester/test manager/consultant is only part of the skill set needed to ensure you maximise your potential. Managing your own career and building a successful future does not 'just happen'. Planning, learning new skills, building relationships and managing career progression are all key in ensuring you keep yourself in the best possible position. You can be the best tester/manager/consultant but still be overlooked.

This workshop examines all aspects of career management and maximising opportunities through skills management, representing yourself for promotion/career move and differentiating yourself. We also investigate the market place, the different means of identifying new career opportunities and income.

Jennifer Lumley *heads up E-Assurance who specialise in careers in Testing/QA, providing advice and guidance on employment, training and self improvement.*
She has 12 years experience working in software testing within recruitment, consultancy and advising on skills development. Prior to this she spent over a decade working in more generalist IT recruitment.

Jon Tyler *is joint founder and owner of E-Resourcing Ltd of which E-Assurance is a trading division. Jon has over 20 years IT recruitment industry experience, before that he worked for Unisys for 11 years. Jon was MD of Volt Europe which he grew from £6M to c£70M and then 5 years ago jointly set up E-Resourcing aiming to set a higher level of personal service to clients and applicants/contractors in the IT skills sector. The success of the company has been recognised by fantastic testimonials from clients, applicants and contractors alike and was also named as the 24th fastest growing UK private company in the latest Sunday Times Virgin Fast Track List. Jon still enjoys being at the coal face of the business.*

Issue 34

THE TESTER

September 2010 Issue

IN THIS ISSUE:

Thursday 16th September 2010

Next Conference: A Testing Toolbox

Keynote Presentations

- **The Irrational Tester**
- **UAT - A Game for Three Players**

Track Sessions

- **Application Security Awareness**
- **Delight your Organisation, Increase your Job Satisfaction & Maximise your Potential**
- **The New Role of The Tester - Becoming Agile**
- **How To Suspend Testing And Still Succeed - A True Story**

Interactive Workshops

- **The Tester's Toolbox - Seven Powerful Cognitive Techniques**
- **If Testing is a Wicked Problem, how can we cope?**

bcs The Chartered Institute for IT

Software Testing Specialist Group

Please note that any views expressed in this Newsletter are not necessarily those of the BCS.

FROM THE EDITOR

Matt Archer, Editor

Welcome to the September edition of The Tester.

Our article this month has been written by **Giles Davies**, a technical specialist from Microsoft. In his article, **'Improving collaboration between developers and testers with Visual Studio 2010'** Giles examines the latest version of Visual Studio, focusing on those features that bring developers and testers closer together. If you are thinking about a change in toolset or an upgrade to your existing Microsoft offering then this article is for you.

If you are inspired by reading Giles' article and would like to become a published author in *The Tester* yourself, then please email me.

I look forward to seeing you all at the conference in September. In the mean-time, happy testing!

Matt Archer

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

SIGIST 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

WEBSITE LINKS

BCS SIGiST website:
www.SIGiST.org.uk

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

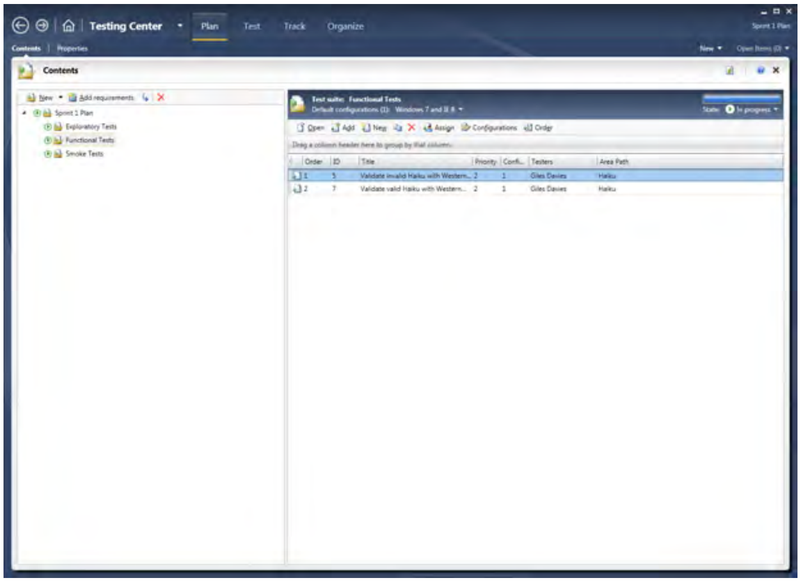
IMPROVING COLLABORATION BETWEEN DEVELOPERS AND TESTERS WITH VISUAL STUDIO 2010

Microsoft Visual Studio is best known as providing an environment for developers, but one of the core new areas for Visual Studio 2010 is a focus on supporting testers with new dedicated testing tools. A major theme behind the introduction of the new testing support is to help bring developers and testers closer together.

When a tester reports a bug the ideal situation is that the developers see the bug details and can immediately start working towards a resolution. In reality there may well be a lot discussion between the testers and the developers over the details of the bug before the developers feel that there is enough information to start working on a fix. We can break the reason for this into two broad areas:

1. Insufficient information to reproduce the bug
2. Environmental differences between the test and development environment

The key new concept in Visual Studio 2010 is that of an actionable bug. This can be defined as a bug report that contains sufficient information for the developer to immediately be able to take action to resolve the bug, addressing the first area above. The challenge is to provide an actionable bug without imposing a burden on the tester (the tester wants to get on with testing, not filing bug reports). To illustrate what an actionable bug is let's start by walking through the execution of a manual test. The tester will be using the new Microsoft Test Manager. This is a new, dedicated tool for testers that integrates into Microsoft Team Foundation Server. Test Manager means that testers don't need to have or use Visual Studio, but have access to the same environment that the development teams are using.



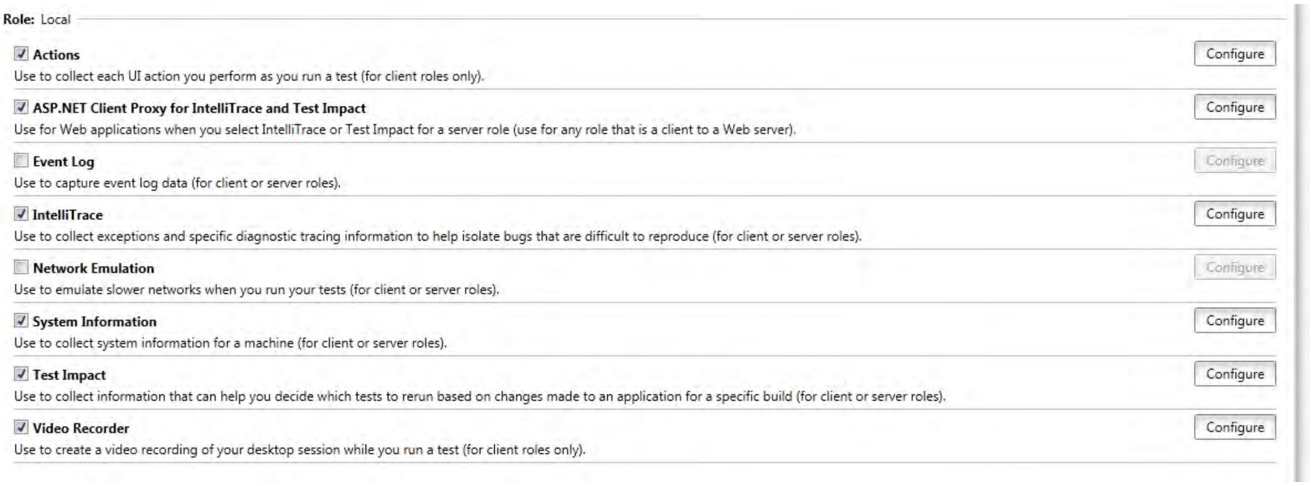
Test suite: Functional Tests
Default configurations (1): Windows 7 and IE 8
State: In progress

Drag a column header here to group by that column.

Order	ID	Title	Priority	Confi...	Testers	Area Path
1	5	Validate invalid Haiku with Western...	2	1	Giles Davies	Haiku
2	7	Validate valid Haiku with Western...	2	1	Giles Davies	Haiku

Fig 1. Microsoft Test Manager.

In this simple example I've defined a single test plan with three test suites. In the test plan I've configured a number of diagnostic data adapters. These collect data on behalf of the tester so that the developer is provided with as much useful information as possible whilst trying to minimize the input required from the tester.



Role: Local

☑ **Actions**
Use to collect each UI action you perform as you run a test (for client roles only).

☑ **ASP.NET Client Proxy for IntelliTrace and Test Impact**
Use for Web applications when you select IntelliTrace or Test Impact for a server role (use for any role that is a client to a Web server).

☐ **Event Log**
Use to capture event log data (for client or server roles).

☑ **IntelliTrace**
Use to collect exceptions and specific diagnostic tracing information to help isolate bugs that are difficult to reproduce (for client or server roles).

☐ **Network Emulation**
Use to emulate slower networks when you run your tests (for client or server roles).

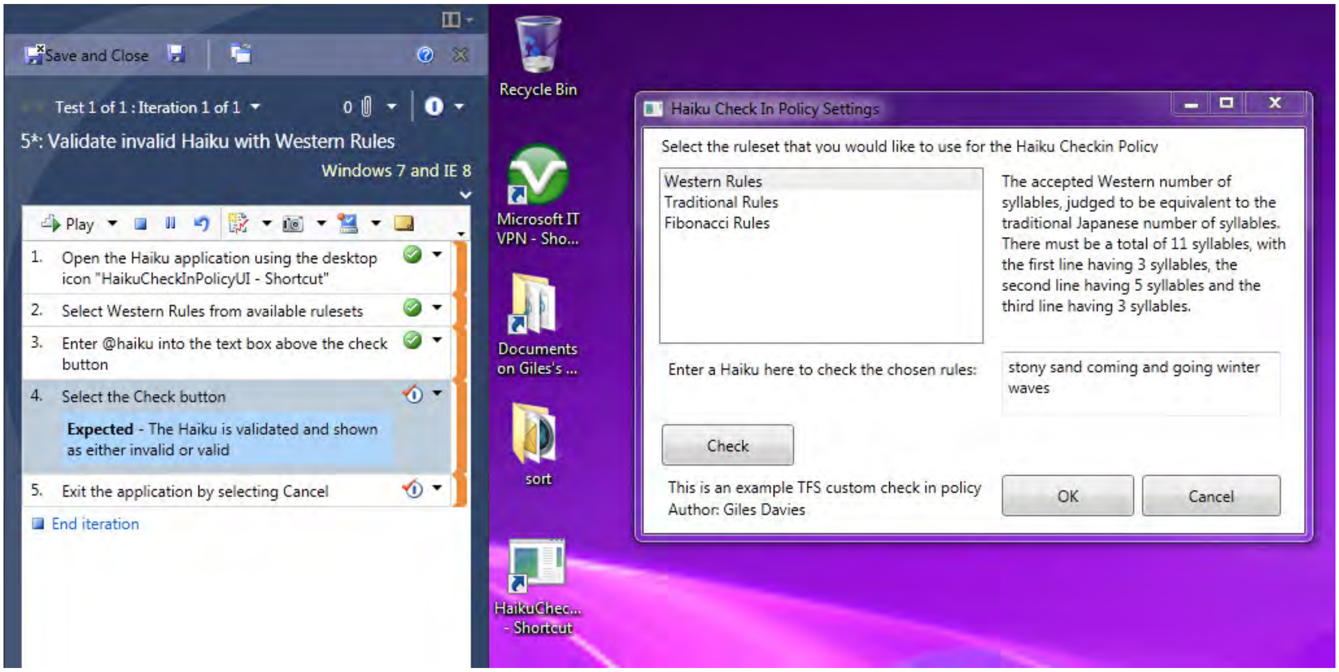
☑ **System Information**
Use to collect system information for a machine (for client or server roles).

☑ **Test Impact**
Use to collect information that can help you decide which tests to rerun based on changes made to an application for a specific build (for client or server roles).

☑ **Video Recorder**
Use to create a video recording of your desktop session while you run a test (for client roles only).

Fig 2. Microsoft Test Manager Diagnostic Data Adapter Configuration.

The tester chooses one or more tests to execute and during the execution of the test the specified data is captured and automatically included in any bug report. Here we can see the test being run just prior to a bug being found in step 4 of the test case:



Test 1 of 1 : Iteration 1 of 1

5*: Validate invalid Haiku with Western Rules

Windows 7 and IE 8

1. Open the Haiku application using the desktop icon "HaikuCheckInPolicyUI - Shortcut"
2. Select Western Rules from available rulesets
3. Enter @haiku into the text box above the check button
4. Select the Check button

   Expected - The Haiku is validated and shown as either invalid or valid
5. Exit the application by selecting Cancel

End iteration

Haiku Check In Policy Settings

Select the ruleset that you would like to use for the Haiku Checkin Policy

Western Rules
Traditional Rules
Fibonacci Rules

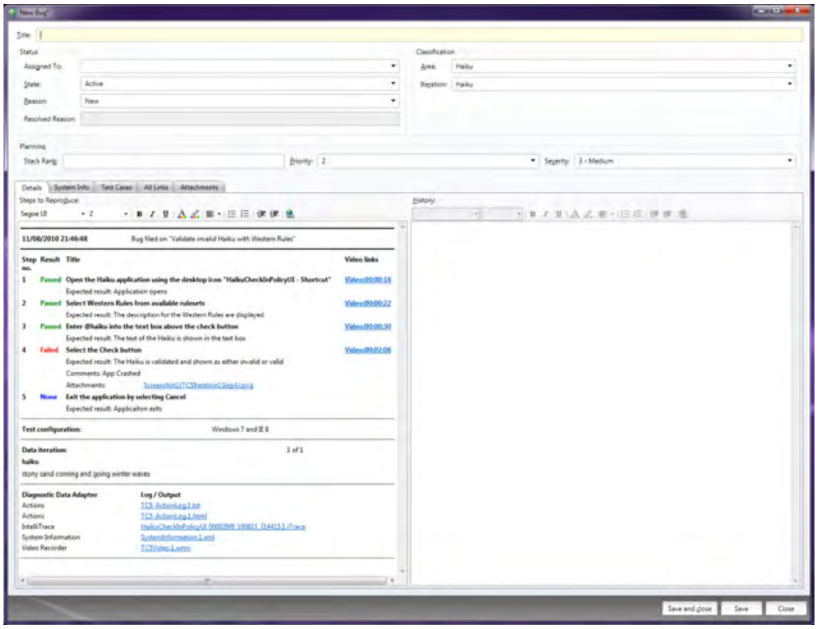
The accepted Western number of syllables, judged to be equivalent to the traditional Japanese number of syllables. There must be a total of 11 syllables, with the first line having 3 syllables, the second line having 5 syllables and the third line having 3 syllables.

Enter a Haiku here to check the chosen rules: stony sand coming and going winter waves

This is an example TFS custom check in policy
Author: Giles Davies

Fig 3. Microsoft Test Manager executing a manual test

On finding the bug the tester does two things; take a screenshot and create a new bug. The new bug report is shown below, but note that nothing has been entered manually:



New Bug*

Title:

Status
Assigned To:
State: Active
Reason: New
Resolved Reason:

Classification
Area: Haiku
Iteration: Haiku

Planning
Stack Rank:
Priority: 2
Severity: 3 - Medium

Steps to Reproduce:

11/08/2010 21:46:48 Bug filed on "Validate invalid Haiku with Western Rules"

Step no.	Result	Title	Video links
1	Passed	**Open the Haiku application using the desktop icon "HaikuCheckInPolicyUI - Shortcut"** Expected result: Application opens	Video:00:00:18
2	Passed	**Select Western Rules from available rulesets** Expected result: The description for the Western Rules are displayed	Video:00:00:22
3	Passed	**Enter @haiku into the text box above the check button** Expected result: The text of the Haiku is shown in the text box	Video:00:00:30
4	Failed	**Select the Check button** Expected result: The Haiku is validated and shown as either invalid or valid Comments: App Crashed Attachments: Screenshot1[TCSIteration1Step4].png	Video:00:02:06
5	None	**Exit the application by selecting Cancel** Expected result: Applicaiton exits	

Test configuration: Windows 7 and IE 8

Data iteration: 1 of 1

haiku

stony sand coming and going winter waves

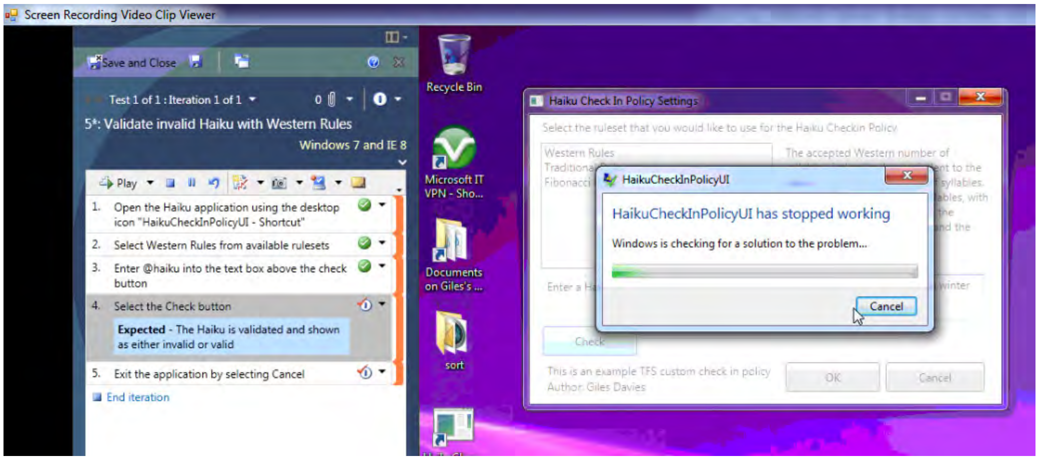
Diagnostic Data Adapter	Log / Output
Actions	TC5_ActionLog1.txt
Actions	TC5_ActionLog1.html
IntelliTrace	HaikuCheckInPolicyUI_00003f4ff_100811_214415.1.iTrace
System Information	SystemInformation1.xml
Video Recorder	TC5Video1.wmv

History:

Fig 4. A bug report as first generated by Microsoft Test Manager.

There are lots of details here but the most important points are:

- The test steps, and their outcome, have been added automatically. The tester does not need to type the test steps performed into the bug report, although they are editable.
- Each step has a link to jump into a screen recording of the test in action at that step. This is of huge benefit to developers who can see and understand what was happening in each step of the test.



Screen Recording Video Clip Viewer

Save and Close

Test 1 of 1 : Iteration 1 of 1

5*: Validate invalid Haiku with Western Rules

Windows 7 and IE 8

1. Open the Haiku application using the desktop icon "HaikuCheckInPolicyUI - Shortcut"
2. Select Western Rules from available rulesets
3. Enter @haiku into the text box above the check button
4. Select the Check button
   Expected - The Haiku is validated and shown as either invalid or valid
5. Exit the application by selecting Cancel

Haiku Check In Policy Settings

Select the ruleset that you would like to use for the Haiku Checkin Policy

Western Rules
Traditiona
Fibonacci

The accepted Western number of

HaikuCheckInPolicyUI

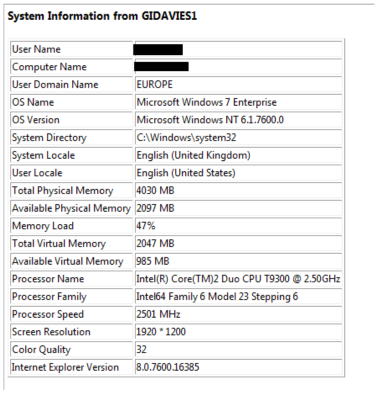
HaikuCheckInPolicyUI has stopped working

Windows is checking for a solution to the problem...

This is an example TFS custom check in policy
Author: Giles Davies

Fig 5. Video of the test being executed, showing the bug.

- The screenshot taken is included in the bug.
- An action log is recorded. This records the key strokes and mouse gestures performed during the test. This can be a useful supplement the test steps and show exactly what happened during the test. (The action log can also be used to later fast-forward through manual tests and form the basis of converting a manual test into an automated test but these are outside the scope of this article.)
- System information is recorded automatically to help avoid return questions (“what resolution were you running, how much memory, which operating system”):



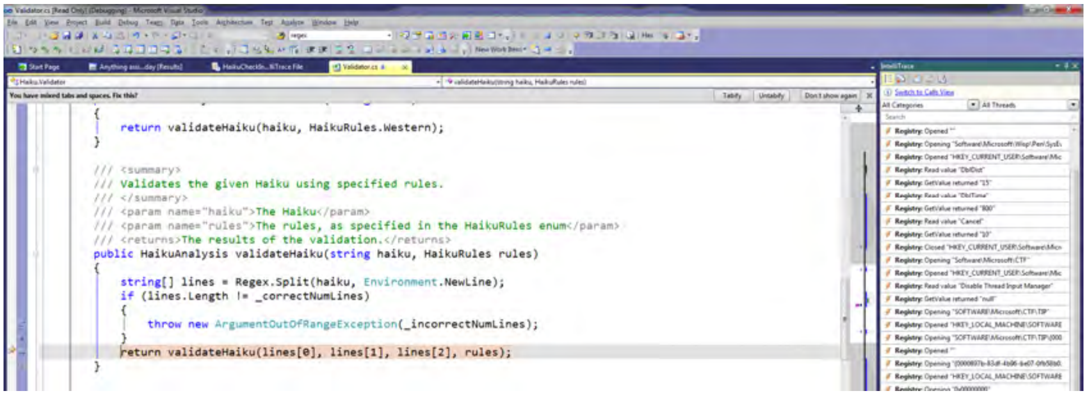
System Information from GIDAVIES1

User Name	[redacted]
Computer Name	[redacted]
User Domain Name	EUROPE
OS Name	Microsoft Windows 7 Enterprise
OS Version	Microsoft Windows NT 6.1.7600.0
System Directory	C:\Windows\system32
System Locale	English (United Kingdom)
User Locale	English (United States)
Total Physical Memory	4030 MB
Available Physical Memory	2097 MB
Memory Load	47%
Total Virtual Memory	2047 MB
Available Virtual Memory	985 MB
Processor Name	Intel(R) Core(TM)2 Duo CPU T9300 @ 2.50GHz
Processor Family	Intel64 Family 6 Model 23 Stepping 6
Processor Speed	2501 MHz
Screen Resolution	1920 * 1200
Color Quality	32
Internet Explorer Version	8.0.7600.16385

Fig 6. System information recorded at the time of the bug by Microsoft Test Manager.

- An IntelliTrace log is captured. This is a new technology that provides a log of the calls and events in the application under test that a developer can use to debug the application at a later point in time. This means that the developer can debug the steps that the tester took through the application, even though they are doing this after the event.
- The tester need only complete the title of the bug, the priority, the severity and decide who to route this to next. Anything else will be supplementary information over and above the collected data.
- In this example the application is a thick client but everything covered here also applies to a web application.

The developer can then find and open up the same bug report inside Visual Studio and access exactly the same information; test steps, user actions, video, screenshots, system information and the IntelliTrace log. Opening up the IntelliTrace log will take the developer to the line of code that threw the exception that caused the bug:



```
        {
            return validateHaiku(haiku, HaikuRules.Western);
        }

        /// <summary>
        /// Validates the given Haiku using specified rules.
        /// </summary>
        /// <param name="haiku">The Haiku</param>
        /// <param name="rules">The rules, as specified in the HaikuRules enum</param>
        /// <returns>The results of the validation.</returns>
        public HaikuAnalysis validateHaiku(string haiku, HaikuRules rules)
        {
            string[] lines = Regex.Split(haiku, Environment.NewLine);
            if (lines.Length != _correctNumLines)
            {
                throw new ArgumentOutOfRangeException(_incorrectNumLines);
            }
            return validateHaiku(lines[0], lines[1], lines[2], rules);
        }
```

```
Registry: Opened ""
Registry: Opening "Software\Microsoft\Wisp\Pen\SysE
Registry: Opened "HKEY_CURRENT_USER\Software\Mic
Registry: Read value "DblDist"
Registry: GetValue returned "15"
Registry: Read value "DblTime"
Registry: GetValue returned "800"
Registry: Read value "Cancel"
Registry: GetValue returned "10"
Registry: Closed "HKEY_CURRENT_USER\Software\Micr
Registry: Opening "Software\Microsoft\CTF"
Registry: Opened "HKEY_CURRENT_USER\Software\Mic
Registry: Read value "Disable Thread Input Manager"
Registry: GetValue returned "null"
Registry: Opening "SOFTWARE\Microsoft\CTF\TIP"
Registry: Opened "HKEY_LOCAL_MACHINE\SOFTWARE
Registry: Opening "SOFTWARE\Microsoft\CTF\TIP\{000
Registry: Opened ""
Registry: Opening "{0000897b-83df-4b96-be07-0fb58b0
Registry: Opened "HKEY_LOCAL_MACHINE\SOFTWARE
Registry: Opening "0x00000000"
```

Fig 7. IntelliTrace highlighting the line of code that caused the bug.

The developer now has a far richer set of information, but the tester has had to provide minimum direct input. The developer will have less questions for the tester, and the tester can move on quickly and with a good expectation that their bugs will be resolved faster and with less intervention.

But what about the second area – developers cannot reproduce the bug because of environmental differences between the test and development environments? This is where Visual Studio 2010 Lab Management comes in to play. Lab Management provides support for virtualized test environments and in the above example would provide an additional option; to take a snapshot of the test environment, which may consist of one or more virtual machines, and include a link to that snapshot in the bug report. The developer can then click on the link and instantiate a copy of the virtual test environment in the state is was in when the tester recorded the bug. This would then allow the developer to attach to the application under test for debugging and examine data or logs in the environment that the tester was using.

There is far more to the new testing tools in Visual Studio 2010 but hopefully this gives an indication of how integrated tooling can help developers and testers collaborate much more effectively.

***Giles Davies** works in the Developer and Platform Evangelism Group in Microsoft UK as a technical specialist covering development tools, specialising in the full Application Lifecycle Management tooling of Team Foundation Server and Visual Studio.*

Giles started his development career with Microsoft technologies in the days of client/server applications before becoming an early adopter for Java, working with CORBA and subsequently J2EE. Having worked in the Java space for a number of organisations including Borland and IBM Rational, Giles joined Microsoft in 2008, and is enjoying working with the Microsoft tools and frameworks. He has had various roles including that of developer, technical lead, software architect, consultant and project manager, and has used a range of development processes including formal methods, Rational Unified Process and Scrum.

UK Visual Studio Team Email: ukvsts@microsoft.com

UK Visual Studio Team Blog: https://blogs.msdn.com/ukvsts/

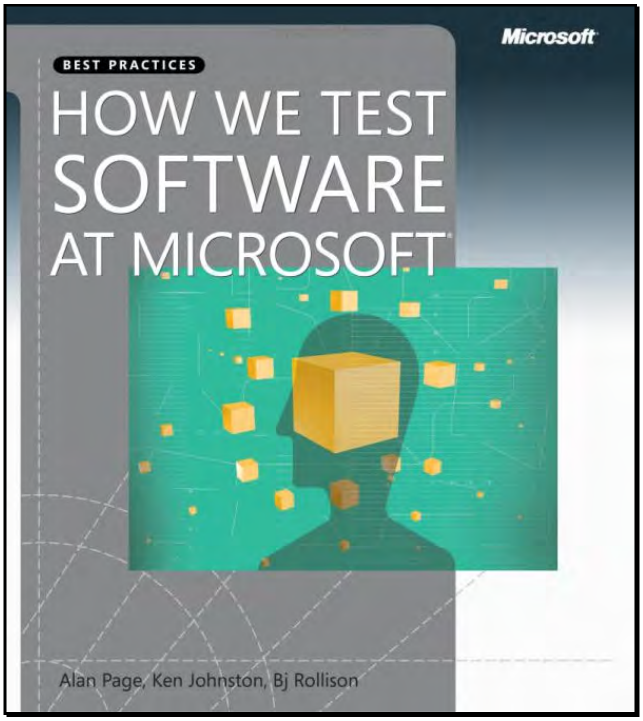
Visual Studio Test Tools Product Team Blog: http://blogs.msdn.com/b/vstsqualitytools/

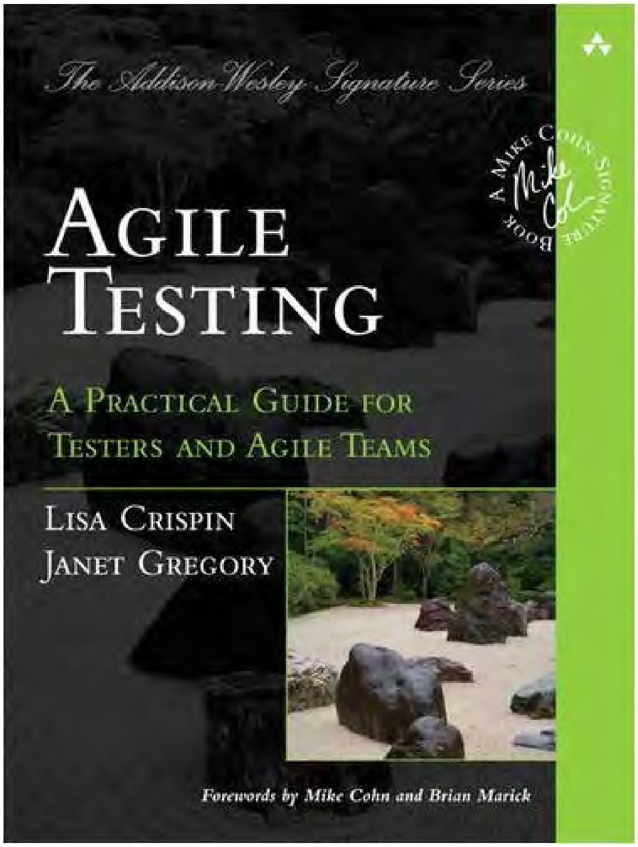
Visual Studio 2010 Lab Management: http://msdn.microsoft.com/en-gb/vstudio/ee712698.aspx

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 SIGiST Library could help!

The SIGiST 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 SIGiST 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, please contact me at matthewjarcher@googlemail.com.

2010 EVENTS CALENDAR

TAICPART
http://www.taicpart.org
3 - 5 September
Windsor, UK

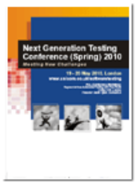
TestExpo
http://www.testexpo.co.uk
15 September
London, UK

CONQUEST
http://www.isqi.org/en/conferences/conquest/2010
20 - 22 September
Dresden, Germany

STARWEST
http://www.sqe.com/starwest
26 September - 1 October
San Diego, US

ignite UK
http://www.iqnite-conferences.com/uk/index.aspx
4 October
London, UK
(ignite wolrdwide http://www.iqnite-conferences.com/)

UK Test Management Forum
http://uktmf.com/
27 October
London, UK



Next Generation Testing Conference (Spring) 2010

Next Generation Testing Conference
http://www.unicom.co.uk/product_detail.asp?prdid=1620
3 - 4 November
London, UK

expo:QA
http://www.expoqa.com
15 - 18 November
Madrid, Spain

EuroSTAR
http://www.eurostarconferences.com
29 November - 02 December
Copenhagen, Denmark

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

CONFERENCE PROGRAMME

BCS SIGiST – A Testing Toolbox

16 September 2010
Royal College of Obstetricians and Gynaecologists
27 Sussex Place, Regent's Park, London NW1

08:30	Coffee & Registration, Tools & Services Exhibition opens	
09:10	**Introduction and Welcome** Stuart Reid, SIGiST Chairman	
09:15	**BCS SIGiST AGM**	
09:30	Opening Keynote **The Irrational Tester** James Lyndsay, Workroom Productions Ltd	
10:30	**Networking Session and Commercial Break**	
10:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition	
11:15	**Application Security Awareness** Martin Knobloch, OWASP.org	**Workshop M1** **The Tester's Toolbox - Seven Powerful Cognitive Techniques** Graham Thomas, Independent Software Testing Consultant
12:00	**Delight your Organisation, Increase your Job Satisfaction &** **Maximise your Potential** John Isgrove, Collaborative Consulting Ltd	
12:45	Lunch break Opportunity to visit the Tools & Services Exhibition (13.00) Lunch time vendor talks	
14:00	**The Share Point:** **Some Challenges Testing in a Data Warehouse** Peter Morgan, Nicemove Ltd	
14:15	**The New Role of The Tester - Becoming Agile** Stuart Taylor, Trader Media Group	**Workshop A1** **If Testing is a Wicked Problem, how can we cope?** James Lyndsay, Workroom Productions Ltd
15:00	**How To Suspend Testing And Still Succeed – A True Story** Graham Thomas, Independent Software Testing Consultant	
15:45	Tea/coffee break Opportunity to visit the Tools & Services Exhibition	
16:15	Closing Keynote **UAT - A Game for Three Players** James Windle	
17:00	Closing Remarks Stuart Reid, SIGiST Chairman	

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.
Workshops places will be allocated on a first come first served basis on the day.

ABSTRACTS AND BIOGRAPHIES

Opening Keynote:
The Irrational Tester

James Lyndsay, Workroom Productions Ltd

Ever wondered whether reason played any part in a decision? Sometimes decisions are influenced by far more - or far less - than a rational analysis. Some decisions are irrational, and testers make decisions that are just as irrational as anyone else. Understanding the patterns that underlie that irrationality will help us be better testers.

Come to this session to discover a testing angle on bias; why we so often labour under the illusion of control, how we lock onto the behaviours we're looking for, and why two people can use the same evidence to support opposing positions. James Lyndsay will will share real-life experiences of tester irrationality, to help you understand the ways in which these common biases affect our everyday work as testers. Discover why timeboxes work, why independence matters, and the subtle nudges that can help encourage us to stay on track. Be prepared to join discussions, engage with demonstrations, and challenge your preconceptions!

***James Lyndsay** has been testing since 1986, and has worked independently since setting up Workroom Productions in 1994. As a consultant, he's worked in a variety of businesses and project styles; from retail to telecommunications, from rapidly-evolving internet start-ups to more traditional large-scale enterprise. He's worked to technical requirements for companies that make and sell software, to commercial requirements for companies that buy and use software, and to unexpected requirements everywhere. He's been in and out of agile (and Agile) teams since 2002. James was an internal irritant to the ISEB exam process for five years, is a regular speaker and occasional teacher, runs LEWT (the London Exploratory Workshop in Testing) and has won prizes for his papers*.

Application Security Awareness

Martin Knobloch, OWASP.org

This presentation is about creating application security awareness within the audience. What is application security all about?

We go through life, doing risk judgments multiple times a day. Is it safe to cross the road? With application security, there is no difference. Who does not use online banking, hasn't booked a vacation online, has never entered banking or credit card details on a website. Have you ever searched yourself on the internet, surprised by the extensive results?

Martin explains what is sense and nonsense about application security. How to test and discover security flaws in web applications. Why is the internet not a safe place and can it ever be?

***Martin Knobloch** is employed as Senior Security Consultant at Sogeti Nederland B.V. He is founder and chair of the taskforce Proactive Security Strategy (PaSS), an integral concept for information security in organization, infrastructure and software.*

Martin is member of the Dutch OWASP Chapter Board. Next to this he contributes to several projects as the OWASP Boot Camp, OWASP Education Project, OWASP Speaker Project and is member of the OWASP Global Education Committee.

Delight your Organisation, Increase your Job Satisfaction & Maximise your Potential

John Isgrove, Collaborative Consulting Ltd

In the current economic climate it is more important than ever that organisations are agile; ensuring they work efficiently, maximising existing resources, getting more value from less. An agile approach enables organisations to adapt to every changing circumstance and quickly grasp opportunities when they are presented, delivering faster return on investment. The benefits of an effective, proven, enterprise scale agile approach are much more than this though.

Agile project delivery is something many organisations have heard about. Many have considered it but are not quite sure what is involved, others concerned about various stories they may have heard on the grapevine. During the presentation John Isgrove will explore what agile is not, what key issues it can address in an organisation and what key benefits it can deliver both for the organisation and importantly the people within.

John will develop your knowledge and understanding of a proven business focused agile framework that embraces an organisation's business and IT functions and tailored to the organisation or project. We will explore the impact, approach and importance of software testing during an agile project. We will learn how the framework provides processes and practices that encourage collaboration and innovation, which in turn helps to maximise project team members' potential, increase their business domain knowledge and their resulting value to the organisation. We will also cover how a major hurdle in many organisations, the gap between business and IT functions is removed whilst at the same time significantly increasing the recognition, value and job satisfaction of team members.

Following the presentation John would hope you are as energised, excited and confident as he is about the significant benefits an agile project approach can deliver.

***John Isgrove** has worked in the IT services industry for over 24 years encompassing technical and management roles in bespoke & packaged software development, ITservice management and consultancy.*

John is passionate about agile and became involved in the mid-90s. An accredited practitioner, John takes a pragmatic, business results focused approach to successfully implementing proven agile delivery practices. Key for him is strong end user involvement and empowered, collaborative team members from all functions across the software development lifecycle.

The New Role of the Tester – Becoming Agile

Stuart Taylor, Trader Media Group

Trader Media runs the autotrader.co.uk website, which sees 500,000 unique visitors a day generate 20,000,000 page impressions. When the business decided it was time to re-vamp the platform, they decided they would adopt an Agile delivery methodology to help deliver a huge project with higher quality than before, within budget and on time.

The software was to be crafted using Test Driven Development (TDD) and eXtreme Programming (XP), testing would be automated and run from within a Continuous Integration environment. With all this testing done by the development team, it was hard to see where the QA would fit into the new delivery methodology and how QA would add value. What we didn't know is that QA would have to work harder than ever before. New tools (selenium/web driver), new frameworks (FitNesse/twist) and new challenges (automated performance testing in a CI). This talk will let you learn from our mistakes, and dispel some Agile myths.

***Stuart Taylor** is an ISEB certified Test Practitioner and Agile Evangelist, a very technical Internet application tester and test team lead; educated to degree level in an electrical engineering discipline with over 10 years specialised Internet testing experience.*

As an early adopter Stuart has grown with the web, and is able to impart his knowledge describing very complicated systems in jargon free language using easily understood metaphors. Stuart has worked extensively with web and internet technologies (including digital TV), he is an ardent proponent of Automated Acceptance Testing and continuous integration development environments.

How to Suspend Testing and Still Succeed - A True Story

Graham Thomas, Independent Software Testing Consultant

This presentation covers a case study from a large testing program for a Member bank which was part of the UK Faster Payments Infrastructure.

Graham will tell the story of a testing program that was destined to fail, but ultimately succeeded. He will give practical details of what went wrong, explain why testing had to be suspended, and discuss how with no real hope of recovery the team managed to set and meet their resumption requirements, and ultimately complete their testing on time.

He will explain the background to the project, the testing strategy that was devised and the program organisational control structure. He will also tell the story of what happened during test execution. Identify where things started to go wrong, how this was identified, and what measures were taken to ensure a successful resolution.

He will go into the detail of the challenges that the testing team, and the program were daily presented with when testing was suspended. And tell how innovation, ingenuity and perseverance, against all the odds, won the day.

This is a real ‘war story’, from the testing front line, with valuable hard won experience, and is told in the very real hope that will benefit all who hear it.

***Graham Thomas** started programming in 1978, working as a programmer through the 1980s before discovering software testing in 1992. He has qualifications in development & testing, and wide ranging experience across the industry.*

Graham now works as a program test manager and testing change agent and has presented on testing since 1995. He won the 2006 BCS SIGiST Best Presentation award, and also gives time to conference programme committees, presentation review panels, and testing award deliberations.

Closing Keynote: UAT – A Game for Three Players

James Windle

The context of the presentation is User Acceptance Testing (UAT) of IT systems: the testing undertaken to verify that an IT system meets the end user’s actual requirements and does not contain any significant defects. The focus is the relationships between the three parties involved: the Customer, the User and the Supplier, all of whom will be fully defined. It will be shown that their interests are broadly aligned, especially and vitally it terms of the delivery of a usable system. However, there are inevitable divergences of interest and these will be covered. There are also discrepancies of perception/understanding. It will be shown how these potential conflicts can be addressed by co-operation based on good communication. This approach should be seen as a constructive alternative to the all too common atmosphere of confrontation between the parties, from which none of them benefits.

As a necessary background to all these considerations, the presentation will cover a formalised approach to Acceptance Testing, identifying those elements which are relevant to any combination of the three parties. This includes:

- the planning and design of User Acceptance Testing

- the structure and drivers of User Acceptance Testing
- the supplier's responsibilities during acceptance test
- the test team: its composition, responsibilities, skills, perceptions and availability
- the entry criteria for testing and during testing
- an understanding of the nature of defects and their management in a co-operative manner
- conditions for the suspension of testing
- other classes of testing, such as 'compliance testing' and their relationship to UAT
- the basis of final acceptance
- and last but not least, the scheduling of all the activities

The talk will demonstrate how the supplier can benefit from appreciating UAT from the customer's and user's perspective.

The presentation, based on my book, 'User Acceptance Testing - A Practical Approach' is a step-by-step distillation of my experiences with Acceptance Testing in a variety of industry sectors.

James Windle *worked for a Software House after graduating with a degree in Electrical Engineering. In 1983 he founded Firesoft, an IT consultancy initially specialising in business analysis. He gradually realised testing was the weak link in the IT system development process - unsystematic, under resourced and inefficient. Working with a UAT team on a large Customs & Excise project, James developed a simple defect management tool and wrote a guide to UAT.*

Workshop M1:
Seven Things that you Might Not Know
(but may find really useful)

Graham Thomas, Independent Software Testing Consultant

This workshop will take you on a journey through some very useful but mostly unknown tools for perception and comprehension which will aid the delegates in their daily testing lives.

Building on the presenter's previous work in the field, and his the enthusiasm for the subject, this workshop will take the attendees on a 90-minute journey of mind opening discovery, looking at 7 key but often overlooked tools.

The tools, and their techniques are easy to learn and very powerful to use. And they will help the delegates to master testing in the industry's currently very demanding transition from that of a structured V-model history to a leaner, more agile and exploratory approach.

The techniques that will be covered in the workshop are:

1. Gall-Peters Projection
   – a different but more accurate way to look at the world

2. Popper's Theory of Testability
   – a powerful tool to scope testing

3. Mind Control
   – finally proof that your mind is not you own!

4. The Stroop Effect
   – a powerful mechanisms that can control your behaviour

5. The Necker Cube
   – what you see is not what I see!

6. The Spinning Dancer

- the whole may look different to the detail

7. e-prime
   - how to communicate experience rather than judgement

The workshop will explain each technique through demonstration and interaction, followed by a discussion of the power of the technique and an insight into its most effective use.

The session will be highly interactive, directly involving the delegates in all of the exercises to give them a first-hand experience of each technique that they will be able to take back to their workplace.

The delegates will take away workable examples of each technique and an understanding of how to use each to best effect.

***Graham Thomas** started programming in 1978, working as a programmer through the 1980s before discovering software testing in 1992. He has qualifications in development & testing, and wide ranging experience across the industry.*

Graham now works as a program test manager and testing change agent and has presented on testing since 1995. He won the 2006 BCS SIGiST Best Presentation award, and also gives time to conference programme committees, presentation review panels, and testing award deliberations.

Workshop A2:
If Testing is a Wicked Problem, how can we cope?

James Lyndsay, Workroom Productions Ltd

Systems thinkers draw a contrast between tame and wicked problems. Wicked problems cannot be solved in the sense that a puzzle can be solved, or a game won.

In this workshop, we will explore this contrast, and consider which elements of testing look like wicked problems, and which look more tame. We will talk about the ways we cope with our wicked problems, and review how our approaches might be similar to those of other industries.

***James Lyndsay** has been testing since 1986, and has worked independently since setting up Workroom Productions in 1994. As a consultant, he's worked in a variety of businesses and project styles; from retail to telecommunications, from rapidly-evolving internet start-ups to more traditional large-scale enterprise. He's worked to technical requirements for companies that make and sell software, to commercial requirements for companies that buy and use software, and to unexpected requirements everywhere. He's been in and out of agile (and Agile) teams since 2002. James was an internal irritant to the ISEB exam process for five years, is a regular speaker and occasional teacher, runs LEWT (the London Exploratory Workshop in Testing) and has won prizes for his papers*.

Software Testing Specialist Group

BCS SOFTWARE TESTING SPECIALIST GROUP

NOTICE OF ANNUAL GENERAL MEETING

Notice is hereby given that the Annual General Meeting of the BCS Specialist Group in Software Testing (SIGiST) will be held on Thursday 16th September 2010. The venue for this meeting will be the Royal College of Obstetricians and Gynaecologists – RCOG.

Agenda

- ❖ Minutes of Previous AGM and Matters Arising
- ❖ Reports
  - Chair
  - Treasurer
  - Standards committee
- ❖ Constitutional changes
  - None raised
- ❖ Committee elections
  - Secretary
  - Programme Secretary
- ❖ To consider any nominated business

Items for inclusion on the AGM agenda should be emailed to jennifer.lumley@e-resourcing.co.uk. Additions to the agenda must be received no less than fourteen days prior to the meeting.

SIGIST ELECTION PROCESS

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

Elections are required in 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 (including a short job specification – 3 lines max.) should be announced prominently within 'The Tester' (normally in the edition advertising the AGM). Otherwise, for an Extraordinary Meeting, an email will be sent to all registered email addresses on the SIGiST database announcing the election(s).	Maximum 8 weeks prior to election.
Candidates must register their interest in standing for one of the positions with the SIGiST Secretary and provide an accompanying short manifesto (no more than a page of A4) describing what they expect to bring to the role. See section 4. of the SIGiST constitution for eligibility.	At least 4 weeks 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.	3 to 4 weeks 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 one 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. If the nominations number equal to or less than the vacancies, the nominees will be deemed to have been duly elected without an election.
3. Each candidate must create a short manifesto describing why they feel they are the right person for the role and send it to the Secretary of the SIGiST when they register their interest in standing for that role.
4. A simple majority is required to be elected to a position.
5. Only members as defined in section 4. of the SIGiST constitution may vote
6. Voting is only allowed if the member is physically present at the AGM
7. The formal voting process will take place on the day of the meeting (a simple show of hands).

SIGIST ELECTION CANDIDATES

Graham Thomas, Programme Secretary

Background

I first attended the SIGiST in 1992, have frequently attended since, and from 1995 have become a regular speaker, winning the 2006 best presentation award for "***7 Key Measures for Software Testing***". I joined the BCS SIGiST Standards Working Party in 1999, working on non-functional software testing standards. Since 2001 I have served as secretary of the working party. In 2002 I served as Treasurer of the BCS SIGiST.

Eligibility for the role

In support of my nomination for Programme Secretary, which is the role of pulling together the program for the regular seminars, I think the following information is relevant:

- I am currently an independent software testing consultant. I have worked in IT since the early 1980's, and in software testing since the early 1990s. I have worked in various testing roles, from test analyst through to test manager. In the last decade I have specialised in programme test management and testing change roles.
- In attending conferences, seminars and testing groups, and through my standards working party work, I have met and developed a strong network of local and international contacts from which I will draw to build successful seminar programmes.
- I have been attending software testing conferences, seminars and working groups, in the UK, Europe and the USA, so am familiar with the current state of the testing industry, the marketplace, services & tools, future trends, and innovations that are relevant and interesting to the BCS SIGIST audience. I have come to know who the good speakers are and what goes down well with conference audiences.
- I have helped to review conference papers for major testing conferences, i.e. EuroSTAR and expo:QA. I have been a member of the EuroSTAR programme review committee since 2006 and this has helped me to understand the range, variety and quality of presentations and presenters currently out there.
- I have worked on the Conference Programme Committee for EuroSTAR (2008) and expo:QA (2010), so am familiar with, and been responsible for; creating a balanced, informative, interesting and captivating conference schedule.
- When I have been lucky enough to see a good presentation at a conference or seminar I have recommended that the speaker also considers the BCS SIGiST, and have passed on their details to the SIGiST.

Support

The following people have expressed their support for my nomination.

Isabel Evans -	CITP and Fellow of the BCS, author, conference speaker and Principal Consultant at Testing Solutions Group.
Geoff Thompson -	Fellow of the BCS, author, conference speaker, Director of TMMi Foundation, Director and Chairman of UKTB, and current BCS SIGiST Vice Chair.

Summary

In summary, I would like to say that I view the Programme Secretary role, should I be successful, as an honour and privilege enabling me to support what I consider the spiritual home, over the last 21 years, of software testing in the UK. I view the role as an exciting challenge at this pivotal turning point in the software testing industry, the transition from traditional structured testing to a leaner and more agile future.

Mohinder Khosla, Secretary

I have been a member of SIGIST for 21 and BCS member for 25 years. I have attended most of the events since SIGIST was launched. Recently I retired from my position at Steria plc as a result I have more time at hands to serve the specialist group. I have served other specialist groups such as Data Management and SPA as committee member in the last couple of years and arranged speakers and hosted evening events.

I have been working in the IT sector for more than 25 years in various roles as a developer, QA and tester analyst, application support, delivery manager and project manager with BT and with Steria plc lately. I have vast experience in testing and beyond. I feel I have the necessary experience to bring to the post.

Mike Bartley, Secretary

Background

Academic

I gained a PhD in Mathematics, and an MBA and MSc in SW Engineering from the Open University. I have passed both ISEB Practitioner Certificates (in Software Test Management and in Analysis). I have taught for the Open University and local colleges on Software Engineering and testing, and have developed course materials for the Open University.

Industrial

I have over 20 years of full life-cycle experience in software and hardware development for commercial, real-time, embedded and safety-related products. I have focused my career on software testing and hardware verification, and have worked both at a technical level and a management level (running teams of up to 40 people). I now run my own business providing consultancy and execution services in software testing and hardware verification.

I regularly write papers in SW testing (and HW verification) for publication or for delivery at conferences (Testing Experience Magazine, SIGiST, TMF, SQC, Testing & Finance, …).

Professional

I was Chairman of the Bristol branch of the BCS from 1991 to 2001. I started on the Bristol Branch committee again back in 2008 and I am standing for chair again in September 2010.

I regularly organise conferences and speakers for the BCS in Bristol, and with local Bristol organisations. For example, I am currently organising "Tech Startup School" (a series of 11 evenings to help budding entrepreneurs start their own technology startups), "The Multi-Core Challenge")a half day event looking at developing and testing code for multi-core platforms) and a quarterly national DVClub for the hardware verification community (which have a live link so that we get delegates from around the world attending virtually).

What I bring to SIGiST

I bring a wealth of academic, industrial and professional experience in software testing and an extremely wide network of contacts for potential SIGiST speakers. I have a vast amount of experience in organising a wide variety of conferences in various formats that I also bring to the committee.

Finally, I have served on a number of voluntary committees for professional organisations understand the ways such committees operate and the various roles on those committees , including that of Secretary. I would consider it an honor to work on the BCS SIGiST committee as Secretary and help support SIGiST in the fantastic work it does for the software testing community.

Issue 35

THE TESTER

December 2010 Issue

Next Conference: Wednesday 8th December 2010

Susan Windsor

Gojko Adzic

Geoff Thompson

Les Hatton

Fran O'Hara

Erkki Pöyhönen

John Kent

‘The Keynotes - 6 of the best!’

Conference presentations

We have a really strong and varied program for the December conference, entitled ‘**The Keynotes – 6 of the best**’. Every presentation will be of Keynote quality. We hope you will enjoy it.

To open the day **Les Hatton** will ask the question ‘*Would anybody notice if testers did something else in life?* ‘. I hope that the answer is yes. I am looking forward to his inimitable and unmissable style.

This is be followed by **Gojko Adzic** who is talking about ‘*Winning Big with Agile Acceptance Testing*‘. I have seen this presented as a workshop and I think it is a really valuable real-life experience.

The morning is rounded off by **John Kent** who is going to talk to about ‘*How to Improve Test Team Effectiveness using Test Entities*’. John has been working on Test Entities for several years. You may have seen his EuroSTAR presentation in The Hague.

After lunch **Geoff Thompson** will present a bonus session. The Theme is ‘*What Influences me in Software Testing*’. Expect some fun!

Erkki Pöyhönen, who has travelled from Finland, will start the afternoon with a talk entitled ‘Paradigm shifting without a clutch - What I'd do differently?’ He has taken his inspiration from the world of Dilbert, and reworked this classis lesson for software testing.

This is followed by **Fran O’Hara**, who has travelled from Ireland, to talk to us about ‘*Scrum – A Tester Perspective*’. I think this will have great relevance and help us understand the changing world of development.

The final speaker of the day is **Susan Windsor**, who is going to show us ‘How to Create Good Testers’. This is more challenging than it sounds and she will help us to first understand what good testers are.

In this issue

CONFERENCE PROGRAM

HOW TO

ARTICLES

TESTING EVENT CALENDAR

From The Editor

Matt Archer, Editor

Our first article this month comes from **Gojko Adzic** - an active member of the Agile Alliance Functional Test Tool (AA-FTT) programme. In his article, '**Lowering the barrier to entry for agile testing using patterns**', Gojko examines the barriers to adopting agile testing and the work the AA-FTT programme is doing to help remove those barriers. If you enjoy Gokjo's article, I can personally recommend his books ('Test Driven .NET Development with FitNesse' and 'Bridging the Communication Gap'). Both make a though provoking read for anybody working in an agile environment.

Our second article has been written by **Susan Windsor** of Gerrard Consulting. I really like Susan's article as it avoids championing a single process and instead addresses the most important question… what process is right for me? Read Susan's article, '**Agile or Waterfall - Can Project Profiling Help?**' to see how project profiling can help you decide.

In our final article, '**Ham, Eggs and SCRUMmy cakes**', **Peter Morgan** looks at the often overlooked and subtle factors that make SCRUM the success process that it is.

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 email me.

Matt Archer

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

Web Links

BCS SIGiST website:
www.SIGiST.org.uk

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

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

LinkedIn and Twitter

The SIGiST 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 SIGiST group and make a request to join. If you're not a member then go to http://www.linkedin.com/ to create an account first.

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

'The keynotes – 6 of the best!'

Wednesday 8 December 2010
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** **Stuart Reid –** Chair of the BCS SIGiST	**5**
09:30	**Would anybody notice if testers did something else in life?** **Les Hatton -** Professor of Forensic Software Engineering CISM, Kingston University	**60**
10:30	**Networking Session**	**15**
10:45	**Tea / Coffee Break and Exhibition Hall**	**30**
11:15	**Winning Big with Agile Acceptance Testing - Lessons Learned from 50 Successful Projects** **Gojko Adzic** - Neuri	**45**
12:00	**How to Improve Test Team Effectiveness using Test Entities** **John Kent** - Simply Testing	**45**
12:45	**Buffet lunch, Exhibition Hall and Networking**	**75**
14:00	**What Influences me in Software Testing?** **Geoff Thompson** - Experimentus	**15**
14:15	**Paradigm shifting without a clutch - What I'd do differently?** **Erkki Pöyhönen**- Teito	**45**
15:00	**Scrum - A Testers Perspective** **Fran O'Hara -** Sogeti	**45**
15:45	**Tea / Coffee Break, Exhibition Hall and networking**	**30**
16:15	**How to Create Good Testers** **Susan Windsor** – Gerrard Consulting	**45**
17:00	**Closing Remarks** **Stuart Reid – Chair of the BCS SIGiST**	**5**

ABSTRACTS AND BIOGRAPHIES

Would anybody notice if testers did something else in life?

Les Hatton, Professor of Forensic Software Engineering CISM, Kingston University

This talk is in two parts. First, the reason for the title is that given the appalling quality of many systems, (as I will illustrate here with copious and mostly recent tales of the front line), it seems fair to say that there simply aren't enough of us to make any significant difference. It is perfectly possible to produce excellent systems but far too often, we as a society simply choose not to. Some of the reasons why will become obvious from the examples given but one of the main ones is that systems design and implementation has been hi-jacked by management consultants and infested by nonsensical, ephemeral jargon at the expense of reality and the end user.

In the second half, I cease ranting for a moment and return to the scientific method. Testers have long used the knowledge that defects appear to cluster but have you ever wondered why this is so ? As I shall demonstrate, it turns out to be a beautiful property of a complex system related to power-law behaviour and completely independent of the technology used to build that system, so you can indeed depend on it to improve your testing efficiency.

Finally, is this talk fully compliant with all procedures and end to end mentored with regard for all modalities by an appropriately standardised milieu of consensual sub-committees engaging all stake-holders with a measure of gravitas although pushing the envelope outside the box to address all low-hanging fruit before close of play ? I really don't care, I'm a tester, I just break stuff.

***Prof. LES HATTON** is on his third scientific career. He started in 1973 at the Met Office with a shiny new PhD in computational fluid dynamics, but decided the scientific civil service was not for him and left in 1974 to 'see the world' as an exploration geophysicist based in Houston, Texas and then London. After bouncing around a bit on the world's oceans and visiting various generally unsavoury places, he co-founded a geophysical company in 1979 which was sold along with his soul in 1984. He received the 1987 Conrad Schlumberger Award for his work in computational geophysics. Shortly after, he switched careers again to pursue growing interests in software and systems failure, a nice indoor job with no heavy lifting based in his garden shed, shared with his eldest son's gerbils. He became Professor of Forensic Software Engineering at Kingston University in 2004. He has been voted in the "leading scholars of software systems engineering" by the prestigious US Journal of Systems and Software, and since December 2009, is on the editorial board of IEEE Software. In his spare time he is the guitarist and harp player in the Juniper Hill Blues Band.*

Winning Big with Agile Acceptance Testing – Lessons Learned from 50 Successful Projects

Gojko Adzic, Neuri

After an extensive research of agile acceptance testing implementations for his new book, Gojko Adzic presents great war stories that will inspire you to improve your software development process. Learn why and how teams all over the world succeed in bridging the communication gap between business stakeholders and implementation teams and how they got users, developers and testers to collaborate in defining great requirements and acceptance tests to produce software fit for purpose. Gojko will also present a summary of the most important success patterns for different contexts and talk about how to solve most common implementation issues with agile acceptance testing.

***Gojko Adzic** got bitten by the agile testing bug five years ago. Since then, he has helped numerous teams implement these practices, written two books on the subject (Test Driven .NET Development with FitNesse and Bridging the Communication Gap) and contributed to several opensource projects in the agile testing space. At the moment, Gojko is working on his third book, titled Specification by Example.*

Gojko is a frequent speaker at leading software development and testing conferences and runs the UK agile testing user group. His company Neuri Ltd helps ambitious teams from web startups to large financial institutions implement specification by example and agile testing practices.

How to Improve Test Team Effectiveness using Test Entities

John Kent, Simply Testing

Do you spend too much time writing detailed test scripts and too little time increasing test coverage by analysing the system design? Do you know the most efficient way to build tests? Do you consider how much detail should be written into the steps in a test script?   Do you know when you should build test scripts and when it would be better to stick with test cases or even just test conditions? Do you know the reasons for writing test scripts? This presentation will offer answers to these questions using an entity model of software testing as an oracle.

The most scarce commodity in software testing is time. It makes sense then, to be able to write and execute tests in the most efficient way possible. Very often however, no consideration is given to the way we build tests; rather testers just dive in and write scripts. The entity model of software testing has been developed to improve tester productivity and make testing more efficient. It attempts to fully define the test assets we create and the relationships between them. It maps out the entities we use from requirement specifications, test conditions and test cases to test results. An interesting theory with few practical uses? Not so, because the entity model demonstrates the relative benefits of different approaches to building and running tests and offers ways to improve. The presentation will enable testers to improve their testing by giving them a route map which will show the way to more efficient testing.

***John Kent,** a leading consultant, specialises in test automation and test management. He developed the Liberation automated test framework and was involved in the development of the T-Plan test management tool. He wrote the **The View From Kent** column in Professional Tester Magazine and is co-author of the **Official Netscape Guide to JavaScript1.2** (Netscape Press). John regularly presents to international audiences on software testing subjects including his own automation training course.*

What Influences me in Software Testing?

Geoff Thompson, Experimentus

Geoff has enjoyed working within the software testing discipline for 20 years now and in this short bonus session he will encapsulate why he started down that road and share what keeps him motivated today.

***Geoff** has a real passion for the right software testing. He is a founder member of the International Software Testing Qualification Board (ISTQB), is currently the Chairman of The UK Testing Board. He co-authored the BCS book Software Testing - An ISEB foundation. In 2008 Geoff won the European Testing Excellence Award. He is a popular speaker having recently delivered keynote presentations in India and Australia as well as being a regular speaker at EuroSTAR and the BCS SIGiST.*

Paradigm shifting without a clutch – What I'd do differently?

Erkki Pöyhönen, Teito

The Testing field is rich. So rich that normally we do not believe how rich it can be -- rich with different groups of people with varied needs, rich with development challenges and rich with solutions. The people with different background and different paradigms seldom communicate well. That means also different views about generic concepts, like what is considered a reasonable defect report, a good test case or even feasible testing!

I've had the pleasure to experience testing on several, very varied contexts; first product development in a small company, then large-scale development in formal ways, later IT, back to R&D, doing training, consulting, waterfall, Agile. All new situations bring fresh challenges but also a new and widening view to testing. Whenever I asked for advice from a consultant I got the stock answer "it depends", at some point I learned why? and what else to say.

In this talk you'll hear about the changes I had to adjust to, what I learned, how my testing thinking has changed along the years -- including what were my worst mistakes and also my best lessons in testing.

***Erkki** works as software testing consultant and test manager at the service company Tieto Oyj, having 1300 testers globally. Before testing services he has worked as a programmer, tester, line manager, management consultant, trainer and webmaster. As a keen facilitator of professional networks he has been a founding member of ISTQB, FiSTB (Finnish Software Testing Board) and FAST (Finnish Association of Software Testing) TTL (Finnish Information Processing Association, similar to BCS but smaller); while bootstrapping the FAST on 2001 he had SIGIST as a model and benchmark for success. He's a frequent speaker on testing events and promoter for testing profession in Finland. Professionally most memorable achievements so far include being voted by his peers as "Tester of the Year" in Finland on 2008 and programme chair role of EuroSTAR2004. He's proud father of 4 young adults, married, and enjoys photographing Finnish nature and singing in a choir on any available time.*

Scrum – A Testers Perspective

Fran O'Hara, Sogeti

Of all the agile methods, Scrum seems to be spreading the most. However, Scrum does not say much about testers and testing and, in many cases, organisations using agile struggle to implement effective approaches to testing and achieving the productivity enhancements with the required level of quality. This is because of issues such as:

- Implementing SCRUM as a effective work management approach but without designing/adopting appropriate development and test practices within the incremental framework
- Partial implementation of agile methods and agile mindset sometimes resulting in negative quality implications
- No clear Scrum test strategy

This presentation will:

- Explain the essence of Scrum, its roles, practices and the mindset changes required
- Define how the tester fits into the Scrum team
- Present the key learning points so far in Scrum from a test perspective including testing without detailed requirements, testing strategies in incremental/iterative environments and how to integrate agile practices such Test Driven Development (TDD) and exploratory testing in Scrum

***Fran O'Hara** is practice manager/principal consultant with Sogeti Ireland. Sogeti now incorporates Insight Test Services which Fran co-founded in 2003 to provide test consulting, training and managed test services. Fran is also a director of Inspire Quality Services. He specialises in pragmatic approaches to process improvement, agile and associated best practices. Fran is a regular speaker at process improvement and testing conferences. He is an ISEB/ISTQB tutor, a trained SEI CMM lead assessor, a certified ScrumMaster and TickIT auditor, a fellow of the Irish Computer Society and co-founder of the Irish SIG in Software Testing - SoftTest.*

How to Create Good Testers

Susan Windsor, Gerrard Consulting

During this presentation I will cover three key questions:

1. What is a good tester?
2. Why would you want to be one?
3. How can you create or become one?

Firstly, I don't believe there is an industry wide definition of "good tester" that fits every situation; but I do believe good testers exist, and that they are very special and valuable individuals! So, I'll share what I believe constitutes a good tester in some different situations and you can identify how this relates to your world. I'll also give my opinion on how this relates to our obsession for certification.

As for wanting to be a good tester, I will ask why you wouldn't! It's our work that provides the most valuable management information on the planet; gives confidence to stakeholders that their system is going to bring the benefits they want; and the skills we need cover the entire project life-cycle. If only we could unlock our minds from the tethers others try to constrain us with.

Finally, I will share my experiences of how to create good testers. For each of the situations I defined earlier, I'll expand on how your organisation can identify people to target for a career move into testing. Maybe you'll identify how to improve your own career prospects too!

***Susan** has recently merged her business with Paul Gerrard and is now a Principal with Gerrard Consulting, responsible for the provision and delivery of testing consultancy. Prior to running her own company (for 5 years), Susan managed the Testing Service's for IBM. Overall, she has 35 years in IT, the last 15 years dedicated to testing. Susan has spoken at many industry conferences in the past, including EuroSTAR, SQSTest, Softest in Ireland, Unicom, ExpoQA in Madrid and at the BCS SIG.*

LOWERING THE BARRIER TO ENTRY FOR AGILE TESTING USING PATTERNS

Gojko Adzic, Neuri

Agile acceptance testing, acceptance test driven development, behaviour driven development and specification by example are great ways to build quality into products in short iterations or flow-style development. I'd go as far as saying that they are an essential part of a successful testing strategy with Scrum, Extreme Programming, Kanban and related processes. Although the basic ideas behind all these processes have been around since the eighties, the adoption in the community has been less than stellar. Editors of InfoQ, one of the leading agile media sites, suggested that the whole thing is just theoreticall. [1] A new effort by the Agile Alliance Functional Testing Tools (AA-FTT) programme promises to lower the barrier to entry and help teams get started with all these ideas easier.

While researching for my upcoming book Specification by Example, I collected more than 50 case studies of how successful teams approached testing and specifications in short iterations or flow-style development. Once I tried to put that into a consistent story, the level of confusion caused by inconsistent naming suddenly became apparent. Many teams had very similar elements in their processes, but used different names for them. All the successful teams used a set of examples as a target for development, acceptance criteria when it is done and regression check later. Some teams called this set of examples an “acceptance test”, some called it “feature specification”, and some focused on a tool or a mechanism to capture the examples, such as “feature file” or “fitnesse page”. Some names, such as “continuous integration” create a confusion because in this context they do not apply to integration tests but to functional checks. Some very important ideas, such as using a set of examples to aid change impact analysis and support, do not really have a name at all yet.

My angry blog post about this[2] caused a lengthy discussion on the AA-FTT mailing list.[3] This resulted in an effort by the AA-FTT programme to finally define a set of process patterns and a consistent language for the practices associated with agile acceptance testing, acceptance test driven development, behaviour driven development and specification by example.

AA-FTT to the rescue

AA-FTT[4] is a programme of the Agile Alliance, created in 2007 to build awareness and community around functional testing tools applied on agile projects. Although the programme name contains the word tools, the focus of the people involved in it has shifted towards processes and practices. Process consolidation was the key topic at the workshop in Orlando during Agile 2010. Following the discussion on the AA-FTT mailing list after the workshop, we decided to attempt to consolidate the practices and the language around the processes.

Patterns in software development were very popular in early nineties,[5] and although the interest in them has waned over the last decade they are a great way to organise process ideas and practices, so we decided to try out patterns workshops.

Jennitta Andrea, Elisabeth Hendrickson and I organised an initial patterns writing workshop in London in early October,[6] closely followed up by another workshop in Berlin during Agile Testing Days. We invited industry experts coming from different backgrounds, contexts and working with different tools to join us and kick start the process. Linda Rising, with more than 15 years of experience in patterns workshops, luckily agreed to facilitate the initial two workshops and help us get started.

The initial pattern ideas workshop produced around 100 interrelated themes,[7] on everything from project management over collaboration to test automation. This struck me as overly complex considering that the Gang of Four book, probably the best pattern language example in software, captured the key ideas of object oriented design in only two dozen patterns. Linda Rising suggested that we will see several layers of patterns evolving, with higher lever patterns capturing a whole range of ideas.

We then ran writers workshops in London and Berlin to start nailing down what these patterns actually mean. Working in groups of six, we reviewed suggested patterns and proposed refinements. This demonstrated how the same ideas can be implemented completely differently and that “best practices” really do not exist. We often got caught up in discussions on the limits of applicability of particular ideas which helped us better define generalised patterns and the contexts in which they apply.

Conclusions

The first thing that really became obvious to is that the community has to stop thinking about the tools and step back to look at a wider picture of the processes that have evolved over the last decade. Tools are there to assist us with a process, so focusing on tools and not on the process is just causing problems in adoption for new teams.

From the discussions during the workshops and my research, it is also obvious that this thing, whatever it ends up being called, is much bigger and more important than just functional testing. It involves collaboration, specification and long term product change management. I guess this will lead to renaming the programme from Functional Testing Tools to something more process oriented.

Inconsistent and illogical nomenclature only creates confusion and a barrier to entry. As the patterns workshops help us refine the process patterns and create the pattern language, I expect this barrier to entry to be significantly lowered. Patterns should help teams at least take the first step easier and overcome the initial problems, which are in my experience often caused by a gross misunderstanding.

Next steps

Follow-up workshops have already been scheduled in the US and there are talks of workshops in Germany and Finland. I plan to run at least one writers workshop next year in London, so if you are interested in attending get in touch by e-mail to gojko@neuri.co.uk,

We all agreed that the content will be published under the creative commons license, so I expect the initial set of patterns to show up soon on a public collaborative editing platform. At the time when I wrote this, we still did not have a specific URL for it but monitor the AA-FTT mailing list[3] to get notified when the web site comes up.

[1] http://www.infoq.com/news/2009/06/automated-acceptance-tests
[2] http://gojko.net/2010/08/04/lets-change-the-tune
[3] http://tech.groups.yahoo.com/group/aa-ftt/
[4] http://www.agilealliance.org/programs/agile-alliance-functional-test-tool-program-aaftt/
[5] http://c2.com/cgi/wiki?HistoryOfPatterns
[6] http://aaftt-2010patternworkshop.crowdvine.com
[7] http://specificationbyexample.com/patterns.php

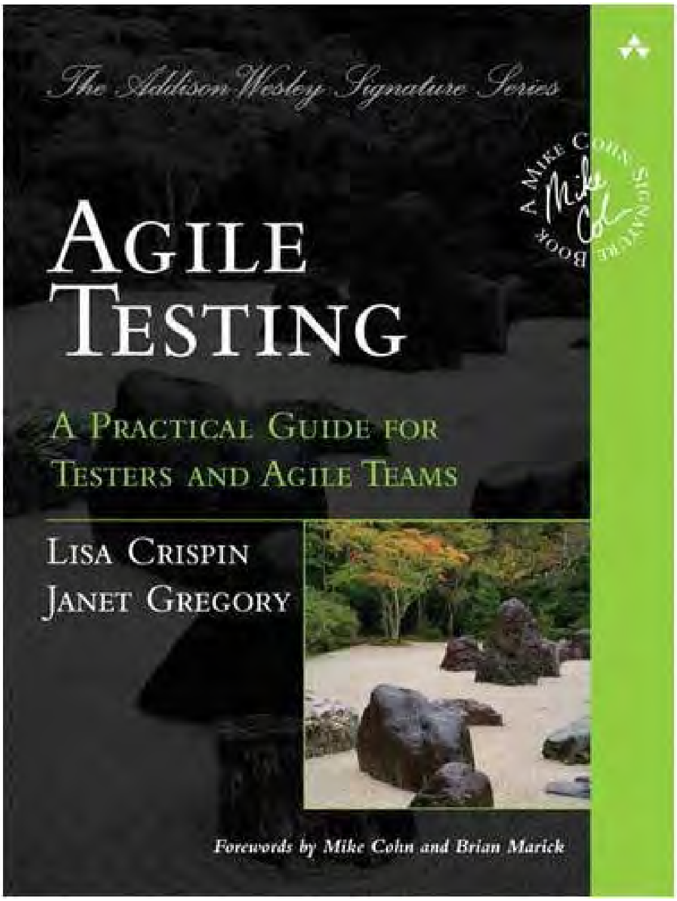
***Gojko Adzic** got bitten by the agile testing bug five years ago. Since then, he has helped numerous teams implement these practices, written two books on the subject (Test Driven .NET Development with FitNesse and Bridging the Communication Gap) and contributed to several opensource projects in the agile testing space. At the moment, Gojko is working on his third book, titled Specification by Example.*

Gojko is a frequent speaker at leading software development and testing conferences and runs the UK agile testing user group. His company Neuri Ltd helps ambitious teams from web startups to large financial institutions implement specification by example and agile testing practices.

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 SIGiST Library could help!

The SIGiST 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 SIGiST member.


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

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

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

AGILE OR WATERFALL - CAN PROJECT PROFILING HELP?

Susan Windsor, Gerrard Consulting

In IT, we all strive to meet the challenge of improving the success of system delivery. The priority areas for me are:

- Delivering what is actually required by the business
- Reducing the cost of re-work
- Increasing confidence in timelines

After so many years it's surprising, and perhaps disappointing, that we're still struggling. However there has been a great deal of effort gone into meeting these challenges.

As testers, we've set up independent test functions; strived to create a profession; become specialists in different disciplines; and introduced new techniques such as risk based testing.

The development approach itself has migrated from formal, highly structured waterfall, through to iterative prototyping and more recently to agile development. All have their strengths and weaknesses in different situations and yet so many organisations appear to be making the decision to switch to agile as a knee jerk reaction, without understanding what it actually means. Some, having switched to agile are reverting to waterfall because agile didn't work for them – hardly surprising is it!

Then there are projects that can't adopt all of the agile principles (such as test driven development and continuous user involvement) but still want to benefit from some of the activities such as defining requirements by examples (stories and scenarios), having short iterations of development, and so on. The advocates of each development approach become very competitive, vying for our attention and assuring us that if we don't do it "their perfect way" we're somehow failures.

In reality, we're all after the same thing and different projects have different needs. Some lend themselves more to agile, some to waterfall and some to somewhere in between – iterative development perhaps? How can this decision be taken, how does that impact the testing approach; and what can we do as testers to contribute to the decision making process?

In a testing improvement project we worked on earlier this year, something our client found to be very helpful was a project profiler and associated test framework. It's a simple concept that was both pragmatic and flexible – and therefore successful. Designed as a tool for project and test managers it helps to identify levels of formality and risk. As testers yourselves, I suggest you look into doing something similar within your organisation. It helps to ensure you've involved at the start of the project (which we always want to achieve) by having something valuable to support the project manager and sponsors.

The first thing we did was remove all the development process "labels" so we didn't talk waterfall and agile; that removed the emotion from the process! We then developed a framework of "Project Aspects" and defined a set of model answers for each of them. You can then quickly circle the model answers that represent the closest fit to your project. Inevitably, all the model answers won't fit neatly into a single column but you're very likely to get a bias towards one of them. In addition, the thought process you go through is very valuable. Let's assume you've got a bias towards the column that represents agile development, you can take the test framework associated with agile and use that to develop a specific test approach for your project. Then, for Project Aspects that don't fit into the agile framework, seek additional activities from the relevant test framework to build into your project test approach.

Our project profiler and test frameworks are too large to fit into this article; even if they did it would make sense for you to develop your own so that it's specific to your organisation; your decision making process; your skills;

and so on. However, there are some examples provided below of the Project Aspects and Generic Risks with their respective model answers. Can you identify which is the agile column? Let me give you a clue; it isn't column D, that represents having insufficient information to make a decision!

Project Aspects	**A**	**B**	**C**	**D**
Sources of Knowledge	New system replaces a well understood existing system; users have a clear vision of system goals and prefer to document their requirements up front	Users want to collaborate to jointly define requirements and meet them incrementally	Users put the onus of requirements elicitation on the project; requirements and the solution will evolve	Inexperienced users who are unable or unwilling to collaborate with requirements gathering
Requirements Stability	New system is a functional replacement of an existing system or a well-defined process(requirements can be fixed early on)	New system replaces an existing system with enhancements or an established (but not necessarily documented) process	New system supports a new business need; business process exists but will change/evolve; users have experience of requirements	New system supports a new business need; business process is not yet known; users have no experience or requirements
Responsibility for Acceptance	Users will take responsibility for UAT and have UAT experience	Users will be responsible for UAT but have no test experience	Users will take part in UAT or witness tests at critical periods, and will review the outcome	Users are unwilling/unable to take part in UAT; reluctant to make the acceptance decision or not known
External Dependencies	More than one or new external suppliers responsible for development; and supplier testing	Single, known supplier responsible for development and supplier testing	In-house development, no external dependencies	Dependencies on external suppliers, their responsibilities or competence not yet known
Visibility, Formality	High visibility/risk to general public; formal progress reporting required at board level; fixed scope and deliverables; formal approvals and sign-offs	High visibility/risk to business; formal progress reporting required; some defined deliverables, some deliverables will emerge/evolve; some approvals and sign-offs	Relatively low business-risk; informal progress reporting is acceptable; partial solution may suffice, incremental/iterative delivery	Potentially high visibility, high risk project; uncertain impact on the business
Generic Risks	**High Exposure**	**Medium Exposure**	**Low Exposure**	**Unknown**
Business Engagement	Available ad-hoc only; nothing significant	Available part time (1-2 days a week) on the project	Dedicated full time on the project	Not known
Project Complexity	Complex and/or critical business processes; conflicting goals/requirements	Stakeholders agreed on goals and architectural vision	Clear well-understood goals, low architectural and system complexity	Business goals and system complexity are not well defined or understood
Risk of System Failure	Failure would cause major/widespread disruption to the business or public embarrassment	Failure would cause widespread/minor or local/major temporary business disruption	Failure would cause temporary, minor, local disruption to the business	Impact of failure is not known

To complete the profiler, include all project aspects; generic risks and product risks that are relevant to your organisation. If you're using risk based testing, you will already have relevant product risks.
I strongly recommend that you try to construct a relevant profiler for yourselves. The real value I found was not only in being able to identify the most appropriate development method and test approach, but in the thought processes required to complete the profiler and how much the outputs helped have conversations with others about the rationale behind the thinking.

Finally, don't get hung up on labels and trends. Construct a test approach that is valid for each project you're involved in and use "examples" where ever possible to aid communication. Good luck! I'll be really interested to hear how you get on.

__Susan__ has recently merged her business with Paul Gerrard and is now a Principal with Gerrard Consulting, responsible for the provision and delivery of testing consultancy. Prior to running her own company (for 5 years), Susan managed the Testing Service's for IBM. Overall, she has 35 years in IT, the last 15 years dedicated to testing. Susan has spoken at many industry conferences in the past, including EuroSTAR, SQSTest, Softest in Ireland, Unicom, ExpoQA in Madrid and at the BCS SIG.

TESTING EVENTS CALENDAR 2010 / 2011

expo:QA
http://www.expoqa.com/
15 - 18 November 2010
Madrid, Spain

EuroSTAR
http://www.eurostarconferences.com/
29 November - 02 December 2010
Copenhagen, Denmark

TestExpo
http://www.testexpo.co.uk/
7 December 2010
London, UK

SIGiST
http://www.bcs.org/server.php?show=nav.9264
8 December 2010
London, UK

UK Test Management Annual Forum
http://uktmf.com/
26 January 2011
London, UK

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

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

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

ignite UK
http://www.iqnite-conferences.com/uk/index.aspx

(ignite wolrdwide http://www.iqnite-conferences.com/)

HAM, EGGS AND SCRUMMY CAKES

Peter Morgan, Software Tester

There are two types of people that can attend our daily development SCRUM meetings: pigs and chickens. Why these are called ‘pigs’ and ‘chickens’ is a well-known story, but in case anyone has not heard it, here it is, courtesy of Wikipedia:

> A pig and a chicken are walking down a road. The Chicken looks at the pig and says "Hey, why don't we open a restaurant?" The pig looks back at the chicken and says "Good idea, what do you want to call it?" The chicken thinks about it and says "Why don't we call it 'Ham and Eggs'?" "I don't think so" says the pig, "I'd be committed but you'd only be involved.”

So, ‘pigs’ have a voice at the SCRUM because they are committed, but ‘chickens’ are only involved, so for the most part stay silent. On our project, there was a mixture of Agile and traditional approaches; I was the tester in the only Agile team, a team of three, geographically removed from the rest of the project. For that reason, the daily stand-up SCRUM was actually a telephone conference call. These two matters, (Agile and traditional [almost Waterfall] within the same project and telephone conference call SCRUM meetings) presented some interesting challenges for the whole team and particularly the 3-man Agile cell, but that is not the purpose of this article. As the tester, I was very much the link between our development team and the rest of the project, and towards the proposed implementation date, spent 2 or 3 days a week with the majority of the team, in Central London. Some of the overall project team in London were regular or occasional participants in the daily SCRUM, as ‘chickens’.

The SCRUM meeting is actually a very good discipline, with its three focus points for each ‘pig’ in the 15 minute session:

1) What did you do yesterday?
2) What are your targets for today?
3) Any blockers?

It holds everyone to account, and disseminates information, so that all (including the ‘chickens’) know what is going on, and where we expect to be at the same time the next working day. However, we have a particular rule that also applies to SCRUM meetings, whether these are true stand up sessions, or like ours, a telephone conference call. Anyone who is late buys cakes!

If it is truly a stand up session, provided that some of your SCRUM colleagues are in close proximity, or the meeting area is within view, making the SCRUM meeting on time is not normally a problem. However, conference calls are different, and one that has cost me dearly, both through the pocket and around the waist. There I am, grappling with the latest release, seeing if my ‘observations’ have been addressed, and what effect this release has on the embryonic User Guide when bingo! My development colleagues sitting no more than 2 metres away have picked up their phones, connected and are awaiting my participation, already thinking “cakes”.

Excuses for non attendance are permitted, provided that these are given at the previous SCRUM, or are allowable (my 4 nights’ unexpected stay in hospital was fortunately an acceptable excuse). However, the vagrancies of London Transport are not valid. “You’re late, and we want our cakes!” Recently I had a real problem when I had two consecutive days in London. On the first, there was a problem with the Circle Line, so I was 6 minutes late. My fault, I should have left enough time to walk the whole way (Oh, he is a hard SCRUM master!), so cakes it would be. The SECOND day, the tube was slightly delayed, so I was faced with buying cakes on the way to the office, and being late again (meaning cakes the day after as well), or turning up with no cakes. I chose the latter, and asked those on the conference call whether I should buy cakes for the London participants or the West Country team, with one of the London ‘chickens’ to decide. Problem solved, cakes were bought in London and available that afternoon.

Nobody (neither ‘pigs’ or ‘chickens’) likes to buy cakes for being late. It is not the cost, or even the inconvenience (£6.00 for a very good selection, and 6 minutes maximum). No, it is the ignominy of having been found out. The starting time constraint of the SCRUM meeting is an example of the positive power of peer pressure. Avoiding the disapproval of one’s colleagues can be a tremendously motivating force – try turning up at a document review meeting without having even opened the document to see this disapproval with all its brutal force. Avoiding disapproval can be the factor that keeps the Agile development on track, that maintains the velocity of the development or even increases it. The thought of explaining to colleagues why you have not done that which you contracted to do (at the previous day’s SCRUM) is a big motivator. It is almost as much a motivator as the idea of (not) buying cakes. Almost, but not quite!

***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.*