

Issue  
**15**

# THE **TESTER**

**March 2006 Issue**

## **NEXT CONFERENCE**

**Wednesday  
15 March 2006**

## **A Test in Time Saves Nine**

- Developing Testers: What can we learn from athletes?
- Testing Large Software Systems
- Intelligent Test Strategy
- Are you Ready for Model-Based Testing?
- Process Improvement at Marks and Spencer
- Test Planning Optimisation Through Metrics analysis
- Making the Most of What You've Got
- The Pain and Gain of Test Automation
- Achievable Futures

## **IN THIS ISSUE:**

**FROM THE EDITOR**

**FUTURE SIGIST CONFERENCE DATES**

**NEXT MEETING – PROGRAMME**

**SPEAKER BIOS AND ABSTRACTS**

**GO WITH THE BUG FLOW**

**TESTING FOR PROFIT**



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

### FROM THE EDITOR

I hope you've got our 2006 conference dates noted in your new diaries! The next date is Wednesday 15 March when we have another good programme outlined for you.

Those who attended the last conference were informed that Mark Fewster, who has done a superb job in the role of Programme Secretary for a number of years, feels that it is now time for him to pass the role on to someone else. We will therefore be looking for volunteers at the AGM (June conference), and ideally before this so that the role can be discussed with interested parties. Mark plans ahead for speakers at future events so there will not be a totally blank sheet on taking over and Mark will also be happy to lend a helping hand during a transition period. The role may be undertaken by more than one individual, as long as there is just one person with ultimate responsibility for arranging speakers at our conferences. If you would like to know more, please contact Mark at [mark@grove.co.uk](mailto:mark@grove.co.uk)

Please note that the UML Testers' Forum is now affiliated as a sub-group of the SiGIST, and there is an announcement of a meeting in this issue.

Don't forget to tell your testing colleagues of the SiGIST conference on 15 March and we look forward to seeing you there!

Pam Frederiksen  
Communications Secretary  
Tel: 01483 881188 (Leysen Associates)  
Fax: 01483 881189  
Email: [pam@leysen.com](mailto:pam@leysen.com)

**BCS SIGIST website:** [www.sigist.org.uk](http://www.sigist.org.uk)

**SIGIST Standards Working Party:** [www.testingstandards.co.uk](http://www.testingstandards.co.uk)

**SIGIST UML Testers Forum:** [www.umltesters.org](http://www.umltesters.org) \* **NEW**

### FUTURE SIGIST CONFERENCE DATES

15 June 2006

19 September 2006

14 December 2006

### BOOKING INSTRUCTIONS

1. Download a booking form from:

<http://www.sigist.org.uk/bookingForm.pdf>

2. Complete and fax to:

Colin Chivers  
01793 480270

OR Post to:

Colin Chivers  
Specialist Groups & Branches Accounts  
1 Sanford Street  
Swindon  
SN1 1HJ

## NEXT MEETING – PROGRAMME

### BCS SIGIST – A Test in Time Saves Nine

Wednesday 15 March 2006

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

08:30	Coffee & Registration, Exhibition opens		
09:25	Introduction and Welcome – Stuart Reid, SIGIST Chairman		
09:30	<b>Featured Speaker</b>		
	<b>Developing Testers – What Can We Learn from Athletes?</b> <i>Paul Gerrard, Systeme Evolutif Ltd.</i>		
10:30	SIGIST Best Presentation Award for 2005		
10:35	Networking session and commercial break		
10:50	Coffee & opportunity to visit the exhibition		
11:20	<b>Testing Large Software Systems</b> <i>Hans Bohlbro, Systematic Software Engineering A/S</i>	<b>Featured Speaker</b> <b>From a Sow's Ear to a Silk Purse: Making the Most of What You've Got in Software Testing</b> <i>James Lyndsay, Workroom Productions Ltd.</i>	<b>Featured Speaker</b> <b>Intelligent Test Strategy</b> <i>Paul Gerrard, Systeme Evolutif Ltd.</i>
12:05	<b>Are You Ready for Model-Based Testing?</b> <i>Richard Warden, Software Futures Ltd.</i>		
12:50	Lunch & opportunity to visit the exhibition		
13:50	<b>Book Review</b>	<b>From a Sow's Ear to a Silk Purse: Making the Most of What You've Got in Software Testing (continued...)</b>	<b>Intelligent Test Strategy (continued)</b> <i>Paul Gerrard, Systeme Evolutif Ltd.</i>
14:00	<b>Process Improvement at Marks &amp; Spencer</b> <i>Andrew Goslin, Marks &amp; Spencer</i>		
14:45	Tea & opportunity to visit the exhibition		
15:15	<b>Test Planning Optimisation Through Metrics Analysis</b> <i>Francis Miles, Nokia</i>	<b>The Pain and Gain of Test Automation – The Early Days</b> <b>Andy Redwood, Portman Building Society</b>	<b>Intelligent Test Strategy (continued)</b> <i>Paul Gerrard, Systeme Evolutif Ltd.</i>
16:00	<b>Featured Speaker</b>		
	<b>Achievable Futures</b> <i>James Lyndsay, Workroom Productions Ltd.</i>		
17:00	Closing Remarks		

The SIGIST committee reserves the right to amend the programme if circumstances deem it necessary.

### PARALLEL SESSIONS

There are three parallel sessions today offering alternative presentations to those held in the main lecture theatre. These are held in a separate room that restricts the number of attendees. Places will be available on a first-come, first-served basis on the day. There is no advanced booking and no additional fee.

## Announcements

### **SIGIST Library**

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, Techniques, Test Process Improvement

If you would like to know more about the library and books available, or for any queries, please contact Julie Gardiner on 07974 141436 or email her at [gardinerjulie@yahoo.co.uk](mailto:gardinerjulie@yahoo.co.uk). Alternatively, download the book loan form on the SIGIST website [www.sigist.org.uk](http://www.sigist.org.uk). Happy Reading!

### **UML Testers Forum**

The UML Testers' Forum is now affiliated as a sub-group of the SIGIST which means the two can work very closely together, share resources and explore the world of model-based testing. Forum chairman Richard Warden is speaking on UML Testing at the 15th March meeting, see programme in this issue.

The Forum is holding a meeting on Wednesday 29th March at Testing Solutions' offices, Walbrook, central London and we invite anyone with an interest in UML and model-based testing to attend. For details of the meeting and the Forum please visit our website [www.umltesters.org](http://www.umltesters.org)

## SPEAKER BIOS AND ABSTRACTS

### Featured Speaker:

#### **Paul Gerrard**

*Systeme Evolutif Ltd.*

---

**Co-author (with Neil Thompson):**  
***“Risk-Based E-Business Testing”***

### **Developing Testers - What Can We Learn from Athletes?**

#### **Abstract:**

When it comes to improving the capabilities of our testers, if you believe the training providers brochures, you might think that a few days training in a classroom is enough to give a tester all the skills required to succeed. But it is obvious that to achieve mastery, it can take months or years to acquire the full range of technical and inter-personal skills required.

Based on the presenters experience as a rowing coach, this keynote describes how an athletic training programme is run and compares that with the way most testers are developed. An athlete will have a different training regime for the different periods of the year and coaching, mentoring, inspiring and testing are all key activities of the coach. Training consists of technical drills, strength, speed, endurance and team work. Of course a tester must spend most of their time actually doing their job, but there are many opportunities for evaluation and training to occur even in a busy schedule.

Developing tester capability requires a methodical, humane approach with realistic goals, focused training, regular evaluation, feedback and coaching as well as on-the-job experience.

#### **Biography:**

Paul is the Technical Director and principal consultant for Systeme Evolutif, an influential Testing Services Company. He has conducted assignments in all aspects of Software Testing and Quality Assurance. Previously, he has worked as a developer, designer, project manager and consultant for small and large developments using all major technologies and is currently the webmaster for the Evolutif and other websites.

Paul has degrees from the Universities of Oxford and London, is Web Secretary for the BCS SIG in Software Testing (SIGIST) and former Chairman of the Certification Board for the ISEB Tester Qualification whose aim is to create a training and qualification scheme for professional testers. He is a regular speaker at seminars and conferences in the UK, continental Europe and the USA and was recently awarded the “Best Presentation of the Year” prize by the BCS SIGIST.

Paul has written many papers and articles, most of which are on the Evolutif website. With Neil Thompson, Paul wrote “Risk-Based E-Business Testing” – the standard text for risk-based testing.

In his spare time, Paul is a coach for Maidenhead Rowing club women’s squad.



### **Hans Bohlbro**

*Systematic Software Engineering A/S*

---



## **Testing Large Software Systems**

### **Abstract:**

Independent test teams in large software development projects are faced with a unique set of challenges in relation to product complexity, coordination and cooperation with many development teams, and a development period often running over several years. This presentation describes a number of techniques applied to the incremental development model that have proven to be very effective in tackling these challenges. The result is a project where the development and testing process are integrated in such a way, that they are truly parallel and supportive of each other thereby achieving a significant improvement in the overall product quality.

The presentation is based on the collective experience and solutions of several large projects at Systematic Software Engineering A/S.

### **Key Points:**

In this session attendees will learn:

1. Practical approaches to making incremental development, test and delivery work in large projects.
2. Practical approaches to creating a common understanding of what to test.
3. Practical approaches to improving corporation of test and development and thereby achieving an improvement in the overall product quality.

### **Biography:**

Hans Jørgen Bohlbro is a Chief Program Manager at Systematic Software Engineering A/S in Denmark – a CMMI Level 5 software company. He is currently program manager for the company Business Process Improvement division and is also head of the internal Knowledge Leadership for Software Test. He is an active member of a Danish special interest group in software testing and has previously given presentations at a number of seminars and conferences on the subjects of software testing and CMMI implementation. He has 9 years of experience in software test and project management. He holds a BSc. from the University College of Aarhus and is also PMP and holds an ISEB Practitioner Certificate in Software Testing.

## Featured Speaker:

### **Paul Gerrard**

*Systeme Evolutif Ltd.*

---

**Co-author (with Neil Thompson):**  
***“Risk-Based E-Business Testing”***

### **Tutorial: Intelligent Test Strategy**

#### **Abstract:**

Successful project managers manage achievements, not activities. What this means is that the readiness and acceptability of project deliverables are what must be monitored, rather than the completion status of activities. But most managers track time, cost and effort only because they are easy to measure.

The decision making in projects is critically dependent on accurate information and this tutorial describes how QA and testing is the main source of Project Intelligence (PI). Delegates will see how PI can be gathered, analysed and presented to senior management to support the crunch decisions that often go unmade because of lack of information.

This tutorial offers a new approach to test strategy, using techniques based on the emerging discipline called Benefits Realisation Management. Results Chain models will be used to show the fundamental relationships between project goals, risks, project activities and dependencies being used as a framework for test strategy. PI promotes better communication within projects and supports the need of senior management to make better decisions when setbacks occur early, when change becomes inevitable and testing is squeezed.

The techniques of PI will be illustrated and practiced using several exercises during the day. A case study will be used to explain how PI can work in a realistic setting and the ways that PI can be used to support decision making are discussed.

The PI approach transforms management's perception of testing, promotes closer collaboration between development and test and better informs management decision making.

#### **Biography:**

Paul is the Technical Director and principal consultant for Systeme Evolutif, an influential Testing Services Company. He has conducted assignments in all aspects of Software Testing and Quality Assurance. Previously, he has worked as a developer, designer, project manager and consultant for small and large developments using all major technologies and is currently the webmaster for the Evolutif and other websites.

Paul has degrees from the Universities of Oxford and London, is Web Secretary for the BCS SIG in Software Testing (SIGIST) and former Chairman of the Certification Board for the ISEB Tester Qualification whose aim is to create a training and qualification scheme for professional testers. He is a regular speaker at seminars and conferences in the UK, continental Europe and the USA and was recently awarded the “Best Presentation of the Year” prize by the BCS SIGIST.

Paul has written many papers and articles, most of which are on the Evolutif website. With Neil Thompson, Paul wrote “Risk-Based E-Business Testing” – the standard text for risk-based testing.

In his spare time, Paul is a coach for Maidenhead Rowing club women's squad.



### Featured Speaker:

#### **James Lyndsay**

*Workroom Productions Ltd.*

---

### **From a Sow's Ear to a Silk Purse: Making the Most of What You've Got in Software Testing**

#### **Abstract:**

The test team has a greater influence on the success of testing than any single process, tool or technique. Yet, under-resourced and over-stretched, it can be a source of weakness.

James Lyndsay leads this double-length workshop, looking at effective techniques to help get the best out of your team, without increasing your budget or finding fewer bugs. You'll look at ways of finding nuggets of time within your existing processes, and discuss how to use this time for better testing, rather than more testing. With an eye on motivation and morale, the workshop will give an overview of:

- Really cheap tools;
- Communication within and around the team;
- How to encourage learning, improvement, and knowledge transfer;
- Automated metrics and reporting;
- Reducing your tests without reducing your coverage.

#### **Biography:**

James Lyndsay is an independent Test Strategist, based in London. He's spent well over ten years in software testing, and has been the principal consultant at Workroom Productions since its formation 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 enterprises. James is an invited attendee of LAWST and WHET, and is a regular and popular speaker at international test conferences, delivering keynote talks at STAREast and ASIARstar in 2003. He received "Best Paper" at STARWest 2002 and at EuroSTAR 2002 for "Adventures in Session-Based Testing". James is also a director of The Manual (<http://www.the-manual.org/>), a not-for-profit organisation whose aim is to gather and publish basic skills.





**Richard Warden**

*Software Futures Ltd.*

---

### Are You Ready for Model-Based Testing?

**Abstract:**

Don't panic if you are not. This presentation will explain the development and testing activities in Use Case driven development using the Unified Modelling Language. Starting with the initial requirements model we look at the progression in developing an architectural model. Different UML models are introduced at various stages and we show how and where testers are involved.

UML provides a variety of model types that testers use initially during reviews – a very effective means of early error detection. Each model, such as Use Case, Sequence, Activity and State, can then be used during test analysis and design to build a comprehensive test suite. For example we can start with the functional tests based on the Use Case view and then increase our coverage by developing tests from other views, e.g. state-based testing and sequence-based testing.

The presentation will include examples so that even if you are new to UML you can understand the way in which model-based testing works and compare it with your current testing methods. Although UML presents a technical challenge in terms of understanding the language and the underlying object-orientation concepts, many traditional skills are needed. So, all is not lost!

**Biography:**

Richard Warden is founder/chairman of the UML Testers' Forum that recently affiliated to the BCS SIGIST. He wrote and tested his first computer program on 12th May 1970 and has never looked forward! He was a development analyst, designer, programmer and tester on RAF defence support systems, followed by work on business and CAD systems for RACAL Electronics in management roles of testing, development and quality assurance. He then worked for the K3 Group as product and research manager before gaining his freedom.

Richard has been an independent IT consultant for 15 years ([www.softwarefutures.co.uk](http://www.softwarefutures.co.uk)). For the last eight he has worked extensively with UML; testing systems, developing techniques and courses and providing training and consultancy. For nearly two years he was a test consultant to the Swiss Exchange, working on UML-based trading systems. Following that he was the principal test consultant to Semaphore Europe, an object-technology company. In 2004 he co-founded the Forum with Testing Solutions Group ([www.testing-solutions.com](http://www.testing-solutions.com)). Apart from model-based testing Richard enjoys tutoring ISEB Foundation Certificate courses to help encourage the next generation of testers.



**Andrew Goslin**

*Marks & Spencer*

---



## Process Improvement at Marks & Spencer

### Abstract:

Why is process improvement so hard? Why do changes take so long to implement? Why is process improvement associated with high costs? Is there any benefit? Does anybody really think it is worthwhile?

Marks & Spencer have established an ongoing roadmap to implement a number of process improvements using TMM and CMMI. These process improvements have been delivered in the context of moving development and testing activities off-shore. Process health-checks have been used to establish baselines for process improvement and enable progress to be measured. The health-checks were vital to enable Marks & Spencer to identify and share existing best practices, but also to address areas of weakness.

No process improvement activity can take place without understanding the drivers and needs of the organisation as a whole. The business challenges and objectives are of particular importance. However, there are many facets and viewpoints to consider, each placing demands on scarce resources. Furthermore, long running process improvement programmes require careful management of conflicting priorities to ensure the successful delivery.

Conflicting priorities are not the only or perhaps the most significant issue to consider. The organisational culture and the way the "change game" is played will have a significant impact on the ability to deliver process improvements consistently across the organisation. The process improvement programme requires effective preparation, planning, and communication to enable the implementation of changes. The implementation will require monitoring and ultimately some form of evaluation or review to ensure that the desired goals have been achieved. But, this is not enough! The willingness of the people in the organisation to accept and embrace the changes can make or break the success of the programme. Different individuals and teams will perceive the changes in different ways and will adopt changes at different rates. People will play games, political games. There are many forms of resistance. Overcoming the resistance is crucial. Understanding how and why people resist the process improvement changes is important to understand in order to develop and maintain momentum.

Successful process improvement relies on robust reference models and objective analysis and measurement tools. It is important to understand both the needs of sponsors and stakeholders in order to select and use tools appropriately. At Marks & Spencer we have learnt many and sometimes hard lessons along the way. It is important to learn from your mistakes early and to recognise the need to make improvements in the delivery of the process improvement programme itself.

Process improvement programmes can be successful and very effective in delivering lasting organisation changes that benefit the organisation!

### Biography:

Andrew Goslin is responsible for Test Strategy at Marks & Spencer. The role encompasses policy, strategy and practices integrated with project delivery process and supporting test tools. He has over 10 years IT experience in various roles and has worked in Retail and Financial Services. He has a BSc in Computer Science, is a certified ISEB Test Practitioner and is a trained CMMI assessor. He is active in promoting professional testing, speaking regularly at conferences, working with ISEB and the TMMi Foundation.

### **Francis Miles**

*Nokia*

---



## **SW Test planning optimisation within a complex Mobile phone project, through metrics analysis techniques**

### **Abstract:**

Mobile Phone projects are complex. They comprise hundreds of deliverables (components) throughout the entire lifecycle of the project (hopefully maturing incrementally), and all needing to be accurately integrated and built, and of course tested. Phone operating systems these days are incredibly complex, resembling early MS Windows operating systems.

This presentation focuses on the methods employed to improve SW testing coverage, by adopting various metrics analysis techniques to help identify risk areas within our products. Once we are aware of persistent regression and unreliable areas, taking into account lessons learned from previous efforts, we can pinpoint software test resource better, and focus on testing areas that really need to be debugged and developed further in time for the product launch date. Less risky areas, which prove through test analysis to consistently behave, will have less effort exerted on them.

The presentation will describe briefly how a typical project lifecycle evolves, and what types of testing is deployed for different stages. It will also describe some of the problems observed over time, and show how the adoption of a standard top level incremental software test strategy rarely works well within this scope. It will go through various examples of test models, and test models within test models, to cope with the development curves of all required deliverables. Finally, a description of some of the techniques evolved to generate good test metrics, and all the hidden secrets they contain will be given.

Over the project development period, massive amounts of test data is generated, and it is this historical data that contains clues as to how specific components are maturing, how effective and efficient the supplier processes are at developing and releasing deliverables, and how overall project regression can be observed, tracked and managed to acceptable levels within the project. Evidence of how we analyse test results to obtain automatic risk weighting to focus our test planning, resulting in faster focussed testing and better failure trapping will be shown, as well as how all this clicks together to demonstrate pictorially risk and maturity throughout the various stages of the project.

### **Biography:**

I started out my testing life during my University Electronic Engineering Sandwich Placement year, as an Automated Test Development Engineer with Siemens Nixdorf in Germany. Here I helped to develop automated test solutions for fast mainframe de-skew asics. Automated in-circuit and functional test software development, mostly within production environments, formed the basis of my career up until 3 years ago, when I started focussing on software test management.

I started on the Nokia 3650 project, and was responsible for the mostly black-box aspects of software test. From there, and through successive phone programs, I developed a deep understanding of our software development processes, and tailored our software test strategies to suite. The main problem was how to do as much quality testing as possible with limited resource, and to tight timescales. This is where we started to look at the past test histories of projects, and understand what we could do better, with more focus. A lot of metric crunching later, some analysis and planning techniques have been adopted, that help us focus resource to deal with the truly important testing areas, quickly.

### **Andy Redwood**

*Portman Building Society*

---

## **The Pain and Gain of Test Automation – The Early Days**

### **Abstract:**

Portman has embarked on a programme of Test Process improvement incorporating test automation using Mercury toolsets. We are 6+ months into improvements, incorporating updates as part of project and change programmes. We started from a position of pure manual testing and we are still evolving.

What have we done to date? Why have we done it? How did we decide what would work best for us? How are we fitting it around the BAU and Change Portfolio? What's gone well and what has not?

Where do we hope to be and how will we know when we have got there?

I'll tell you as much as I can in the time available and hopefully this will help those of you who also wish to embark along this path.

### **Biography:**

Andy made a natural progression from the technical side of the Film Industry to Software programming in 1985. Since this time he has held positions from Business Analyst to Programme Manager, Quality Assurance Manager, Development Manager and Head of Testing Services, predominately in the City in the last 4 years.

Andy has a reputation for strategic vision and delivering global business solutions reporting directly to the Board, liaising on £10M+ programmes since 1995 and has been regularly personally responsible for £5M quality assurance and testing of business and technical solutions, managing teams of between 5 and 60+ people, including external suppliers and offshore parties across three continents.

Andy demonstrates enthusiasm, parallel thinking and team creativity to drive through time to market business solutions whilst maintaining costs.

Andy is a frequent speaker at international conference. He was Chair of the ISEB International Panel in 2003/4, the UK representative to ISTQB in 2003 and founder Chair of the ISEB UK Executive Committee in 2004.

Andy was awarded the EuroSTAR Award for outstanding contribution to the Testing Industry in Europe, in December 2005 following a previous nomination in 2003.

Andy is the interim Group Business Manager for Testing at the Portman Building Society.



### Featured Speaker:

## **James Lyndsay**

*Workroom Productions Ltd.*

---

### **Achievable Futures**

**Abstract:**

Over the last ten years, we've seen huge changes in the ease and ubiquity of our access to information, and have vastly more power at our fingertips. Software testing has failed to keep up with the times. This talk highlights important trends and describes how we can rise to their challenge, and use our existing skills, approaches and tools to achieve a bright and rather different future.

**Biography:**

James Lyndsay is an independent Test Strategist, based in London. He's spent well over ten years in software testing, and has been the principal consultant at Workroom Productions since its formation 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 enterprises. James is an invited attendee of LAWST and WHET, and is a regular and popular speaker at international test conferences, delivering keynote talks at STAREast and ASIARstar in 2003. He received "Best Paper" at STARWest 2002 and at EuroSTAR 2002 for "Adventures in Session-Based Testing". James is also a director of The Manual (<http://www.the-manual.org/>), a not-for-profit organisation whose aim is to gather and publish basic skills.



### GO WITH THE BUG FLOW

#### A Seven-Step Process for Establishing an Effective Bug Triage System

An article by Robert Sabourin

One of the fundamental questions in software engineering is “How do we know when we’re finished?” Whether you are just putting bug tracking in place or are experiencing the mid-project requirements turbulence of a dynamic business context, an effective bug triage process can help you answer with confidence: “We have completed the development and testing we set out to complete. We have explored any new risks we discovered, and the bugs that are left are bugs that are acceptable to us... at least for now!”

One of the most effective bug triage systems is a bug flow. What follows is a seven-step process to establish a bug flow.

#### Step One. Identify key stakeholders.

Stakeholders can be found in every nook and cranny of a software development project. Never be too certain that your list is complete. For instance, in one dynamic e-commerce company, a senior executive I had never met decided to heroically micromanage a software project. He had a bug list from his discussions with executives at the client. We were not allowed to work on anything unless it was to correct a bug in his list. We had to abandon some excellent software practices to follow his directives.

The mistake made was failing to identify this executive as a stakeholder. Had he been identified earlier, he could have influenced the bug decision-making process before it was in use, rather than as a crisis late in the project.

Therefore, find all of your stakeholders. Talk to project managers, accountants, and business unit managers. Involve customer support, product management, sales, operations logistics and information technology. Investigate who key stakeholders were on past projects.

Keep in mind that stakeholders can be working in the interest of your business, your client, yourself, or some other group. If you find many differing and conflicting interests, consider resolving them early through a little informal arbitration over lunch. If things block up, escalate the problem to management—you have found an important project obstacle early!

#### Step Two. Learn how decisions are made.

Take a look at how decisions are made. Was there a formal change control board defined for the last project? Ask specific questions about the past: What happened last time? Was this the right thing? How should it have happened? Ask probing questions about the future. What if this type of problem happens? Who should we call? Who makes the final decision? If that person makes the final decision, will someone else come around and try to undo the decision? A bug flow is efficient when key stakeholders buy into the process, key decision makers buy into how key decisions are made, and decision making is delegated to people who can make timely decisions based on relevant business context.

#### Step Three. Define a bug priority and severity scheme.

Now you are ready to define a relevant priority and severity scheme for the project. Each and every bug we find and triage is assigned a priority and severity.

**Bug Priority** answers the questions “Should we fix this?” and “By when?” There are dozens of possible prioritization schemes. I encourage teams to implement as simple a priority scheme as possible. (For some sample prioritization schemes, see the accompanying sidebar.)

A common mistake is to add “conditionals” to the priority ratings. For example, “Px - Fix if possible” is quite common. My experience with “fix if” priorities is that early in the testing cycle a lot of bugs are assigned a “fix if” priority. Then as time runs out, they are reclassified as “fix later.” The decision to fix a bug is indeed a business decision. By including conditional priority, the decision is now being delegated to someone who will define what “possible” means. Is this the intent? To paraphrase Yoda speaking to Luke Skywalker in the *Return of the Jedi* “No! Try not. Do or do not. There is no try.”

## The Tester

Step		Who	Description
1	Isolate Bug	Bug Finder	Attempt to repeat bug in a small number of steps
2	Enter Bug	Bug Finder	Enter bug description including detailed description, severity, consequences, how to repeat, version, build, platform, date, time, operating details, as required using bug tracking system
3	Review Bug	Bug Finder / Tester	Have peer or test lead review bug description for clarity and completeness
4	Prioritize Bug	Bug Review Committee	During bug review meeting assign a priority to the bug (several bugs will be reviewed in one meeting)
5	Assign Bug	Development Lead	As priority dictates assign bug to a developer for correction and rework
6	Fix Bug	Developer	Assigned developer corrects bug
7	Confirm Bug Fix	Tester	In next build, including bug fix, the test team confirms the bug is corrected without unexpected side effects
8	Close Bug	Tester	Closed confirmed bugs in defect tracking system
9	Reopen Bug	Tester / Test Lead	If bug is not confirmed, it is re-open for subsequent re-prioritization, and if necessary, re-assignment

**Table 1: An example of a typical bug flow, showing steps and responsibilities**

A more effective prioritization scheme would encourage making a decision about bug priority, while also allowing for future changes to the decision if the business or technical contexts change. If you must put conditional decisions in the prioritization scheme, then every effort should be made to minimize the use of it! I have seen projects where over 80% of the bugs are classified "Fix if possible"; this is a sign of weak, ineffective decision making.

**Bug Severity** defines how much damage a bug can cause. Bug severity is a very important input to the prioritization decision. Severity alone cannot be used to decide bug priority; decision makers should also take care to review the consequences of not fixing the bug, the technical risk of fixing the bug, the possible side effects of fixing the bug, and the relative cost of fixing the bug now or later.

Bug severity is often indicated by a simple numeric scheme with a reasonable scale that is well understood by all team members. When determining the bug severity scheme, it is a good idea to find a relevant example for each severity level from past projects. Past examples can also be used as a teaching aid. (See the accompanying sidebar for a sample scheme.)

If you are comparing bug data between different projects, be sure that the severity schemes are similar. This allows apples-to-apples comparisons. Make sure that all testers are taught how to correctly classify severity. It is a good idea to collect representative bugs of software failures of all possible severities. These can later be used to train testers and to teach stakeholders to better understand what bug data really means. *(They will have a better mental image what a bug count of 27 S2 bugs means if they know what an example S2 bug is!)*

### **Create A Common Understanding**

It is a good idea to ensure team members share a common definition of what a bug is. In some organizations, a bug is any concern that could impact the success of the project; in others, a bug is a behavior which does not conform to requirements. No matter what your definition, ensure that all team members are in harmony. Generally it is better to report a bug than to risk not reporting it. If someone ever hovers on the boundary of whether a systems behavior is in fact a bug, they should be encouraged to report the bug. Another gem of folk wisdom—"It is better to have a redundant bug than to not have it in the list at all"—should be drummed into all team members. Encourage everyone to report the bugs they observe and to properly follow the bug reporting steps.

## Step Four. Define steps to take when a bug is found.

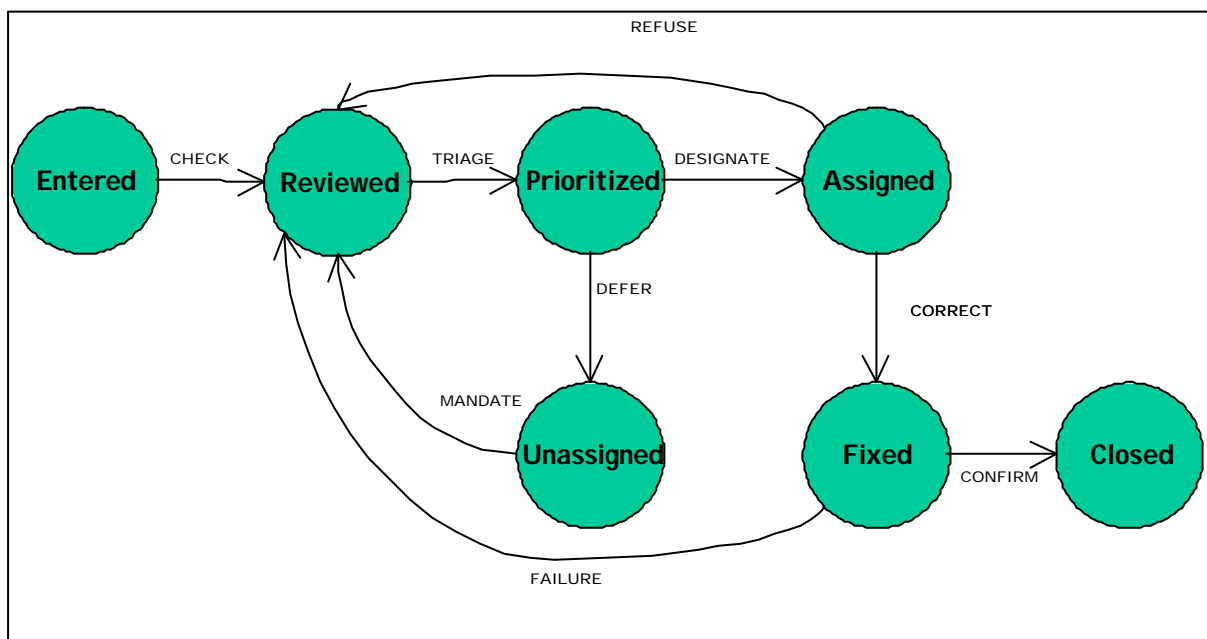
The credo "Never Lose a Bug!" should echo through the halls! All team members, testers, developers, writers, managers, analysts, and subject matter experts must know what to do if and when they spot a bug. A simple list of step-by-step instructions will suffice. Table 1 shows the steps used on a real project.

This simple process will take place for all bugs reported. Bug Isolation and Bug Reviewing should be performed with support of the more experienced team members. Prioritization and Assignment are done at a bug review meeting. Tight coordination between developer and testers takes place between when a bug is prioritized and when a bug fix is confirmed. The table shows main responsibilities—there may be many more interactions than indicated.

## Step 5. Build a state model of the bug flow.

Bug flows can be described using state transition diagrams. A state transition diagram uses circles to represent system states, and lines to represent how states can change. The life of a bug can be viewed as its transition through all of its different states of existence.

The state diagram in Figure 1 represents a simple bug flow. A bug can be in any of the seven states. These states are defined by reviewing all possible prioritization states of a bug. In our example, once a bug is entered into the defect tracking system a peer or test lead checks it. After a successful review, the bug is in the reviewed state. A bug triage meeting determines whether the bug should be fixed or not. If a bug should be fixed, it is assigned to a developer, confirmed to ensure it is fixed, and subsequently closed. If the developer cannot reproduce the bug or for some reason rejects the bug (perhaps too much technical risk), then the REFUSE transition moves the bug from the assigned state back to the reviewed state, from which it will be triaged again!



**Figure 1: State Transition Diagram provides an unambiguous description of the bug flow. This diagram shows all the possible dispositions of a bug, the flow between states, and the actions that drive state changes**

You must first identify all possible states of a bug. Consider starting before bug prioritization and ending with bug fix being confirmed. Draw arrows indicating all possible state transitions. States are labeled with the name of a state and arrows are labeled with the action (generally a verb) which transforms the bug from one state to the next state. Take care to use unique names to define each state and each state transition. At any time, every bug is in one and only one state. Project management can track the number of bugs in every state verses time, which leads to a traditional bug graph representation. Note that during different development phases, different bug flows may be required, so you should create a separate diagram for each case - i.e. system testing, integration testing, acceptance testing, and live operation. (For more on the different kinds of bug flows, see this issue's StickyNotes at [www.stqemagazine.com](http://www.stqemagazine.com).)



## Working with Priority and Severity

The simplest prioritization scheme I have successfully used in small commercial projects (5-10 team members) is this:

- P1 (High): Fix bug before commercial release**
- P2 (Medium): Fix bug in future commercial release**
- P3 (Low): Do not fix bug**

However, more granularity is often required. A more typical example would be:

- P1: Fix bug immediately**
- P2: Fix bug before next build**
- P3: Fix bug sometime before commercial release**
- P4: Fix bug in a future commercial release**
- P5: Do not fix bug**

One of my favorite severity schemes defines four levels of severity based on whether a client can work around or avoid the problem:

- S1 (Showstopper): All of product inoperable, crashed, data destroyed**
- S2 (Major): Feature is inoperable—no work around**
- S3 (Minor): Feature is inoperable—work around exists**
- S4 (Cosmetic): Cosmetic—does not impact operation**

Bug quadrants are an important tool to help focus testing teams on finding bugs that matter.

QI HIGH PRIORITY SEVERE	QII HIGH PRIORITY NOT SEVERE
QIII LOW PRIORITY SEVERE	QIV LOW PRIORITY NOT SEVERE

QI bugs are bugs that are both high priority and severe. Testers should be encouraged to try to identify as many QI bugs as possible. These are the bugs that will affect the bottom line for the product. QII bugs are less severe than QI bugs but are high priority. Perhaps the priority of the bug is dictated by a special need of the client—For example a spelling mistake is usually low severity, but on sensitive legal text it may be a very high priority.

Bugs in QIII are of a special class; these are severe bugs that we do not have to fix right away. Perhaps we can sell the product to customers not requiring features with QIII bugs for the first commercial drop of the product. Perhaps we can exclude problematic platforms for the sale—if the application crashes on a specific version of Windows, we can simply not allow it to be used on that version.

Bugs in QIV are low priority and low severity. A team should avoid having too many bugs in this quadrant. If there are large volumes of low severity bugs, they may have a very negative impact on the project and should thus be assigned a higher priority. Testers should report all QIV bugs observed, but test plans should not focus testing effort on finding this class of bugs.

I encourage testers to be aware of which quadrant the bugs they are finding belong to and to focus on identifying QI and QII bugs whenever possible.

A bug can jump between quadrants as a result of changes to the business or technical context of a project. I worked on a project where over 170 High Priority bugs were reprioritized to Low Priority (Fix later) because of a business change for the product's main customer. They decided to support a completely different database platform that was much better integrated with our code base. Thus all database bugs from the previous vendor were reassigned to a new priority.

You should continuously monitor the important business drivers on the project. Whenever an important business driver changes, you should review recent relevant bug prioritization decisions. In light of the new business conditions, should we still be fixing these bugs? Are there bugs that we decided not to fix which must now be addressed?

It is possible for the same exact technical bug to be in any of the four quadrants as a result of changing business conditions! One example is a bug that results in a system crash when a certain function F is used. The bug is reported as High Severity and assigned to QI. The client decides that the module with the bug will be replaced by a third party product, so the bug is downgraded to QII. Our product management decides to deprecate the feature, so the bug shifts all the way to QIV! Then our company merges with the third party product the client wanted to use. Now we will have to integrate their code into our function F in some future release. The bug jumps back to QIII.

### **Step 6. Ensure key stakeholders buy into bug flow.**

Now that a bug flow has been described and diagrammed, it is important to explain it to all stakeholders and to find out if everyone accepts the process and is willing to follow it. Review the priority and severity definitions, steps to handle a bug, and bug flow state transition diagram with all stakeholders. You may need to perform a group presentation. Bring examples to demonstrate severity levels clearly. Make clear who will be delegated the power of assigning priority to individual bugs. As required, you may need to add or remove states in the bug flow. Business sensitive stakeholders will value simplifications and state reductions—you must keep the system simple but meaningful!

One problem you may run into is that two stakeholders request opposite changes to the bug flow. For example, one stakeholder may want to review bug priorities before another and vice versa. When stuck with this political problem, reflect upon how great it is that you are resolving the issue before you have even processed your first bug! Confusion and power struggles hurt in-progress projects, but also force tough decisions to be made up front. You do not have to give in to either side - just be tactful and bring both opposing stakeholders into a joint meeting. Work from the common goals of the parties to isolate specific problem areas, always demonstrating their importance to the business success of the project. If the parties cannot resolve the issue, then they are clearly blocking your project. You are well within your authority to escalate the issue. Usually the conflicted stakeholders will renegotiate a reasonable solution that can be built into the workflow definitions! (Do this as early as possible.)

Some technically-oriented testers may be shocked at my suggestion that business people and external stakeholders should have a say in how bugs are prioritized! I must emphasize that the decision of which bug to fix and which bug to keep are among the most important factors to deciding go – or no go on shipping a software product. Bug workflow and triage is where such key decisions are made. If they are made with the full awareness of key stakeholders, they will better reflect the business goals and values. The most effective workflows I have used involve decisions made with three people at the table: a product manager representing the customer and business goals; a development manager representing the technological aspects of the project; and a representative from the testing team with a massive dose of objective input! (The CFO does not have to be in the triage meeting, but the CFO should buy into and support the business process.)

Share the state model and prioritization/severity schemes with all team members. Make sure that every single person on the team is aware of how bugs will be tracked, prioritized, reviewed, triaged, assigned and confirmed. Make sure staff knows that not all bugs will be fixed, but all bugs will be dealt with fairly and professionally throughout the project. Explain with examples the meaning of all severities and encourage staff to bring new instructive examples to your attention whenever they are uncovered during the project.

Make sure you remember to teach new team members the bug flow when they join the team.

### **Step 7. Adapt bug flow as required to reflect business and organizational changes.**

If you have identified the key stakeholders and adapted the bug flow based on stakeholder feedback you are ready to use the bug flow! That doesn't mean you're done. Be prepared to change it as time goes on. Many software development organizations are merging or restructuring to reduce cost and increase competitiveness. Whenever an organizational change takes place that influences your project, even indirectly, it is a good idea to review the bug flow. Are you considering the input of a stakeholder who is no longer relevant due to new responsibilities? Are you missing input from new stakeholders recently assigned to your business area of importance? Are you going to adopt new development methods as a result of a merger, or will you be using different release and deployment strategies as part of a business re-orientation? When the world changes around your project, take the time to make sure that the bug flow is still rational, reasonable and relevant!

#### ***A Warning about Tools***

A warning about bug track tools: I recommend that you define the bug flow first then choose a tool that can effectively implement it in your organization. Before you buy any defect tracking tool (or software development tool for that matter), make sure it can implement your workflow and actually work in your environment.

### **The moral of the story**

Bugs are not good or bad. Some bugs are important and have a high priority; some bugs are dangerous and have a high severity. Setting the priority and severity of a bug is a business decision. Always review previous decisions in light of changing business context and remember to ensure staff assigning priority and severity are aware of all relevant business drivers for the project.

A prioritized bug flow that everyone agrees on can allow you one day to say with confidence, “We are ready to ship. The bugs that are left are bugs we can live with—at least for now.”

---

*Robert Sabourin (rsabourin@amibug.com) is an author, lecturer and president of AmiBug.Com, Inc., a firm specializing in software management consulting, teaching and professional development. Robert is the author of the popular children's book I am a Bug (ISBN 0-9685774-0-7), which explains what SQA folks really do at work.*

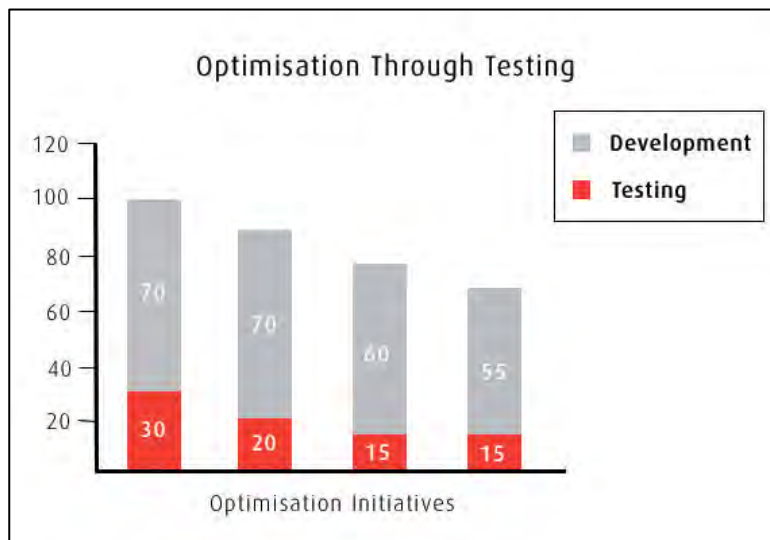
## TESTING FOR PROFIT

*An article by Paul Bevan, Business Development Manager – Testing, Newell and Budge*

As the focus on reducing IT development costs intensifies, businesses are increasingly questioning the “cost of quality” with regard to Testing, there is a common perception that Testing is costing too much.

In recent years, testing only at the end of the development lifecycle, when something has already been built, has been replaced with a growth in static testing techniques. This is in recognition of the benefits and value determined by early detection. However, there now seems to be a growing perception that the pendulum has swung too far the other way. A process driven, one size fits all, test approach that tests everything to the nth degree and may not be appropriate to the particular project or application. To combat this, testers need to be empowered and reminded that they are the experts, and as such should be applying the best fit or most appropriate testing model depending on the project scenarios and business drivers.

By applying appropriate testing techniques which optimise the accuracy of development efforts it is possible to make significant time and money savings throughout the span of the project, effectively meaning you are testing for free! Using a simple model to demonstrate this, take an organisation with a total IT spend of £100m split 70:30 between development and testing activity. The model shows that by employing staged testing initiatives the total cost has reduced from £100m to £70m hence testing, relatively speaking, costs nothing.



Testing teams should now strive to become much more business focussed so that the value is realised and supported at a senior level within organisations. In order to articulate the current issues it may be necessary to change the way in which the test team communicate their findings. For example multiple pages of defect reports are not always as relevant as risk or financial language, in other words the testing data needs to be turned into relevant information for presentation to the decision makers. One way of achieving this is to conduct a study on how much system faults and failures are costing your organisation. This provides a tangible value which gives the test team a way of measuring and communicating the success of improvement programs as well as justifying spend for further testing programs.

We have conducted studies that have shown testing efforts achieve staggering return on investment figures, making a straightforward business case for justifying testing spend.

It is a very well known fact that the further along the development cycle that issues are discovered the more costly they become to fix – this can not only be in terms of cost but also in terms of company reputation and, most importantly, meeting time to market commitments. Static testing techniques such as requirements and traceability testing can significantly reduce the opportunity for misinterpretation, especially when multiple external suppliers are used for software development; this reduces downstream defect rates and in turn reduces time to market - getting IT right first time.

For a full copy of the Testing for Profit White paper send an e-mail to: [testinginfo@sopranewellandbudge.com](mailto:testinginfo@sopranewellandbudge.com)

Issue  
**16** **THE TESTER**  
**June 2006 Issue**

**NEXT CONFERENCE**

**Thursday  
15 June 2006**

**A Every Test Has  
a Silver Lining**

- Are you Hiring Yesterday's Testers?
- AGM
- We Tested It - So Why Doesn't It Work In The WAN?
- The Blind Men Meet The Quality Elephant: How What They See Affects Testers
- How to Become a Sought-After Tester or Test Manager
- The Problem with Agile...
- A Balanced Scorecard Approach for Assessing Test Value and Success
- Seven Key Measures For Software Testing
- Successful Software Management: 15 Lessons Learned

**IN THIS ISSUE:**

FROM THE EDITOR

FUTURE SIGIST CONFERENCE DATES

NEXT MEETING – PROGRAMME

BCS SIGIST – EVERY TEST HAS A SILVER LINING

ANNOUNCEMENTS

BCS SIGIST PROGRAMME SECRETARY ROLE

SPEAKER BIOS AND ABSTRACTS

ARTICLE: COMPETITIVE TESTING – MARTIN JAMIESON, BT



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

## FROM THE EDITOR

We're looking forward to seeing you at our next Conference on Thursday 15 June, so do make sure that you let us know in good time that you wish to attend the event. Our Key Speaker will be Johanna Rothman.

We had some disappointed attendees at the March Conference who hadn't been able to attend some of the parallel sessions out of the main lecture theatre. These sessions are restricted in numbers, due to the available space in the break-out rooms, and therefore we have to apply a first come first served basis of admission.

You will notice from the Programme that the AGM will take place at the next Conference. As stated in the previous Tester we are seeking a new Programme Secretary to take over from Mark Fewster, who has shaped the format of our Conferences very successfully in the past. He would now like to pass the role on to someone else to give time for other activities, and so we have included a description of what the role entails in this edition. If you would like to discuss this in more detail, please contact Mark at [mark@grove.co.uk](mailto:mark@grove.co.uk) and he will be happy to answer any queries you may have.

*If you would like to be considered for this role or any other role on the SIGiST committee please forward a manifesto of approximately 500 words stating what you would intend to achieve in the role to Geoff Thompson our Vice Chairman ([geoff.thompson@experimentus.com](mailto:geoff.thompson@experimentus.com)). We must receive this by Friday 26 May. Each candidate must also provide the names and contact details of a proposer and a seconder. It will not be possible to accept applications for any positions on the day of the AGM. Where there is more than one candidate we will ask each one to present their manifesto at the AGM, and then a vote will be taken.*

We look forward to seeing you in June!

Pam Frederiksen  
Communications Secretary  
Tel: 01483 881188 (Leysen Associates)  
Fax: 01483 881189  
Email: [pam@leysen.com](mailto:pam@leysen.com)

BCS SIGIST website: [www.sigist.org.uk](http://www.sigist.org.uk)

SIGIST Standards Working Party: [www.testingstandards.co.uk](http://www.testingstandards.co.uk)

SIGIST UML Testers Forum: [www.umltesters.org](http://www.umltesters.org)

## FUTURE SIGIST CONFERENCE DATES

19 September 2006

14 December 2006

### BOOKING INSTRUCTIONS

1. Download a booking form from:  
<http://www.sigist.org.uk/bookingForm.pdf>

A. Complete and fax to:	OR B. Post to:
Phil Dyson 01793 417444	Phil Dyson Specialist Groups Officer First Floor, Block D North Star House North Star Avenue Swindon SN2 1FA

## NEXT MEETING – PROGRAMME

### BCS SIGIST – Every Test Has a Silver Lining

**Thursday 15 June 2006**

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

<b>08:30</b>	<b>Coffee &amp; Registration, Exhibition opens</b>	
<b>09:25</b>	<b>Introduction and Welcome – Stuart Reid, SIGIST Chairman</b>	
<b>09:30</b>	<b>Featured Speaker</b>	
	<b>Are you Hiring Yesterday's Testers?</b> <i>Johanna Rothman, Rothman Consulting Group, Inc.</i>	
<b>10:30</b>	<b>AGM</b>	
<b>10:40</b>	<b>SiGiST Best Presentation Award</b>	
<b>10:45</b>	<b>Networking session and commercial break</b>	
<b>11:00</b>	Coffee & opportunity to visit the exhibition	
<b>11:30</b>	<b>We Tested It - So Why Doesn't It Work In The WAN?</b> <i>Frank Puranik, Itheon Networks</i>	<b>Featured Speaker</b>
<b>12:10</b>	<b>The Blind Men Meet The Quality Elephant: How What They See Affects Testers</b> <i>Isabel Evans, TSG Ltd.</i>	<b>How to Become a Sought-After Tester or Test Manager</b> <i>Johanna Rothman, Rothman Consulting Group, Inc.</i>
<b>13:00</b>	Lunch & opportunity to visit the exhibition	
<b>14:00</b>	<b>Book Review</b>	<b>A Balanced Scorecard Approach for Assessing Test Value and Success</b>
<b>14:15</b>	<b>The Problem with Agile...</b> <i>Ken Brennock, Insight Test Services Ltd.</i>	<i>Isabel Evans, TSG Ltd.</i>
<b>15:00</b>	Tea & opportunity to visit the exhibition	
<b>15:30</b>	<b>Seven Key Measures For Software Testing</b> <i>Graham Thomas, Independent Consultant</i>	
<b>16:00</b>	<b>Featured Speaker</b>	
	<b>Successful Software Management: 15 Lessons Learned</b> <i>Johanna Rothman, Rothman Consulting Group, Inc.</i>	
<b>17:00</b>	Closing Remarks	

The SiGiST committee reserves the right to amend the programme if circumstances deem it necessary.

## ANNOUNCEMENT

### **SIGIST Library**

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, Techniques, Test Process Improvement

If you would like to know more about the library and books available, or for any queries, please contact Julie Gardiner on 07974 141436 or email her at [gardinerjulie@yahoo.co.uk](mailto:gardinerjulie@yahoo.co.uk). Alternatively, download the book loan form on the SIGIST website [www.sigist.org.uk](http://www.sigist.org.uk). Happy Reading!

### **BCS SIGIST PROGRAMME SECRETARY ROLE**

The role of Programme Secretary for the BCS SiGiST is to plan and prepare the programme for each meeting. This comprises a number of responsibilities as described below.

#### **Planning**

- Decide on the format of the programme (number and type of presentations).
- Identify and invite speakers.
- Agree fees and expenses as appropriate.
- Review any unsolicited presentation proposals received.
- Prepare programme.
- Notify committee of programme.
- Notify SiGiST Administration of speaker details.

#### **Documentation**

- Prepare programme, abstracts and biographies.
- Check submitted presentations for readability, correct any problems.
- Prepare delegate handouts.
- Forward delegate handouts to SiGiST Administration for printing.
- Prepare introduction slides for presentation on the day.

#### **On the Day**

- Introduce speakers.
- Liaise with venue AV personnel.

Much of the work involves finding and conversing with speakers and obtaining the required information from them (title, abstract, biography, photo and handout materials). Fortunately this work can be spread over several weeks. Compiling the delegate handout materials is also a significant amount of work and some of this is more time critical.





**SPECIALIST GROUP IN SOFTWARE TESTING**

## **Notice of Annual General Meeting**

Notice is hereby given that the Annual General Meeting of the British Computer Society Specialist Group in Software Testing (SIGIST) will be held on Thursday 15<sup>th</sup> June 2006. The venue for this meeting will be the June 2006 SIGIST meeting held at the Royal College of Obstetricians and Gynaecologists – RCOG.

### **Agenda**

1. Minutes of Previous AGM and Matters Arising
2. Reports
  - Chair
  - Treasurer
  - Standards committee
3. Constitutional changes
4. Elections
5. To consider any nominated business

Items to be considered for inclusion on the AGM agenda should be emailed to [julie@gstc.co.uk](mailto:julie@gstc.co.uk). Additions to the agenda must be received no less than fourteen days prior to the meeting.

## SPEAKER BIOS AND ABSTRACTS

### Featured Speaker:

### **Johanna Rothman**

*Rothman Consulting Group, Inc.*

---

**Author: "Hiring the Best Knowledge Workers, Techies & Nerds"**

**Co-author: "Corrective Action for the Software Industry" & "Behind Closed Doors, Secrets of Great Management"**

### **Are you Hiring Yesterday's Testers?**

Abstract:

Today's test teams must provide information about a product throughout its lifecycle. Have you adapted to these needs? Or are you still playing by yesterday's rules?

Testing a complex system is just as difficult as creating a complex system. Just creating unit tests that test each path or object individually is not sufficient testing for a complex system. And just creating expert user based tests is insufficient. The more complex the system, the more varied the required tester skills.

There's a difference between first-class and second-class testers, and it's not hard to tell the difference once you know what to look for. Johanna will discuss how to recognize second-class testers, how to articulate the value of your test group, how to define what first-class testers know and what a first-class group can do, how to recognize first-class testers when you hire, and where to consider training for testers with fewer skills.

Biography:

Johanna Rothman of Rothman Consulting Group, Inc., consults, speaks, and writes on managing high-technology product development. During her decade-long consulting career, she has assisted managers, teams, and organizations become more effective by applying her pragmatic approaches to the issues of project management, risk management, and people management. She's helped Engineering organizations, IT organizations, and start-ups hire technical people, manage projects, and release successful products faster. Her action-based assessment reports have helped managers and teams improve their projects, products, and financial results. She is a sought-after speaker and teacher in the areas of project management, people management, and problem-solving.

Johanna has written over 100 articles and papers, maintains two blogs, and is a frequent contributor for Fast Company's online career center, Software Development, Better Software, Computerworld.com, and StickyMinds.com. Johanna is a co-author (with Esther Derby) of the popular and pragmatic "Behind Closed Doors, Secrets of Great Management". She is also the author of the highly acclaimed "Hiring the Best Knowledge Workers, Techies & Nerds", and is a co-author (with Denise Robitaille) of "Corrective Action for the Software Industry". Johanna is also a host and session leader at the Amplifying Your Effectiveness (AYE) conference.



## **Frank Puranik**

*Itheon Networks*

---

### **We Tested It - So Why Doesn't It Work In The WAN?**

Abstract:

Increasingly, applications are expected to work over Wide Area Networks (WANs), wireless LANs, GPRS, 3G or satellite networks. But software performance testing is still frequently only conducted over the fast and reliable Local Area Networks (LAN) of the test lab. Satisfactory application performance in these conditions is no guarantee of acceptable performance in non-LAN networks. This presentation will explain the reasons why applications can behave very differently in the WAN/Wireless networks and why testing in live network environments, even in so-called "off-peak times", is not really an option. The presentation will go on to explore the methods that are available to software testers to carry out safe but realistic performance testing in the WAN environment.

The ability to conduct accurate WAN/Wireless network performance testing will help to increase confidence that the application is going to work successfully when finally rolled out into the production environment and enhances the perceived value of the software tester in the process.

What the audience will learn:

- Why applications that perform satisfactorily in Local Area Networks often behave so poorly when placed in Wide Area or Wireless Networks.
- The importance of testing new software that is expected to run over WAN or Wireless networks in realistic network conditions and the best ways to do this.
- How the ability to test in WAN and Wireless Network conditions will enhance the value of the software tester

Biography:

Frank co-founded Itheon in 1987 and is responsible for the Application Testing and Network Performance division which specialises in the delivery of application performance testing and network emulation technology. Frank has a Pure Mathematics degree from Sheffield University and over 20 years experience of working in a wide variety of roles within the software industry including software development, sales, technical support and design.



## **Isabel Evans**

*Testing Solutions Group Ltd.*

---

**Author: “Achieving Software Quality Through Teamwork”**

### **The Blind Men Meet The Quality Elephant: How What They See Affects Testers**



Abstract:

- By moving our key measures within the test department closer to our customers' corporate measures we can ensure we meet their strategy for quality and governance
- Simple alignments between a well founded organisational excellence model and the work of the test team to help measurement and improvement, in order to relate to the customer
- In the end the risks we take with testing are risks that affect our customer, our colleagues and ourselves. Our testing may enable or prevent our customer meeting goals, fulfilling strategy, or even achieving proper governance.

In the test team, and within IT, we concentrate on measuring such items as defect density and product attributes. We see testing as an activity which is designed to help mitigate risk. Yet testing can increase risk; for the test team, for the project and for the customer organisation. Sometimes we believe we are working towards providing better solutions for our customers yet they reject our efforts. Why is this?

*“Some blind men found an elephant. “What is this thing?” they cried.*

*“It’s a snake” said Tom...*

*“No - It’s a wall” said Joe...*

*Harry laughed: “You fools...”*

*It’s plain to see – it’s a fan moving side to side.”*

Quality is an elephant found by blind men who think they can see; it can be experienced in many ways. Different people within an organization may hold different opinions about quality, depending on their roles and viewpoint. If testers focus on one definition (their own) they may not communicate well with others in the organization who use a different definition. Testers need to listen and understand the quality viewpoints of others, if they are to deliver what is required.

In the end the risks we take with testing are risks that affect our customer, our colleagues and ourselves. Our testing may enable or prevent our customer meeting goals, fulfilling strategy, or even achieving proper governance.

If we examine the framework for excellence used by our customer organisations we will often find ways to measure our success in relation to their corporate goals and strategy. This enables us to understand how our work helps the customers' strategy for quality and governance.

**Biography:**

Isabel has 20 years experience in the IT industry, mainly in quality management, testing, training and documentation. She has helped organisations in development of procedures, standards and methods to aid testing of software during development and maintenance projects. She has managed test groups, and performed testing design and development for acceptance and system testing of packages and bespoke systems. She has also provided Quality Assurance Support, Release Management, and Customer Support for IT organisations. Most of her work has been with clients in the financial, communications and software sectors.

As well as presenting seminars and training courses to clients, Isabel has spoken on software quality, testing and test management at conferences in the UK, Europe and the USA including EuroSTAR, PSST, Quality Forum, BCS SIGIST, and the “Year 2000 and EURO Summit”. She regularly attends conferences, courses and meetings in her interest areas.

Isabel has been a member of various working parties and groups to contribute to improvement in software quality and testing, including the Quality Forum Testing Metrics Forum, the Customer Satisfaction Measurement working party, and the BCS SIGIST Test Standards Working Party, currently developing the non-functional testing standards.

Her book, “Achieving Software Quality Through Teamwork” was published by Artech in May 2004.

## Featured Speaker:

Johanna Rothman

*Rothman Consulting Group, Inc.*

---

**Author: "Hiring the Best Knowledge Workers, Techies & Nerds"**

**Co-author: "Corrective Action for the Software Industry" & "Behind Closed Doors, Secrets of Great Management"**

### **Workshop: How to Become a Sought-After Tester or Test Manager**

Abstract:

No one wants to be yesterday's tester or test manager. But if you haven't assessed your skills in a while, you could be yesterday's catch, not today's. In this interactive workshop, Johanna will lead you through an assessment of your skills—both technical and non-technical skills—to see where you are.

Once you know where you are, you can develop an action plan to move to your next stag. And, if it fits, you'll have an opportunity to build a support network.

Bring your pencil and an open mind. If you have a resume, bring that too. If you can, bring a list of your accomplishments for the past year.

Outline:

1. Technical skill assessment (for both managers and technical staff)
2. Non-technical skill assessment (for both managers and technical staff)
3. Small group work to discuss where you are and where you want to go
4. Action planning
5. Support group development (if we have time)
6. Debrief

Biography:

Johanna Rothman of Rothman Consulting Group, Inc., consults, speaks, and writes on managing high-technology product development. During her decade-long consulting career, she has assisted managers, teams, and organizations become more effective by applying her pragmatic approaches to the issues of project management, risk management, and people management. She's helped Engineering organizations, IT organizations, and start-ups hire technical people, manage projects, and release successful products faster. Her action-based assessment reports have helped managers and teams improve their projects, products, and financial results. She is a sought-after speaker and teacher in the areas of project management, people management, and problem-solving.

Johanna has written over 100 articles and papers, maintains two blogs, and is a frequent contributor for Fast Company's online career center, Software Development, Better Software, Computerworld.com, and StickyMinds.com. Johanna is a co-author (with Esther Derby) of the popular and pragmatic "Behind Closed Doors, Secrets of Great Management". She is also the author of the highly acclaimed "Hiring the Best Knowledge Workers, Techies & Nerds", and is a co-author (with Denise Robitaille) of "Corrective Action for the Software Industry". Johanna is also a host and session leader at the Amplifying Your Effectiveness (AYE) conference.



## **Ken Brennock**

*Insight Test Services Ltd.*

---

### **The Problem with Agile ...**

Abstract:

While agile has provided the solutions for many software development problems, it is not a silver bullet and has still to over-come issues that more traditional project development lifecycles handle. Testing is an important part of how agile succeeds. This presentation asks the question, can more traditional development lifecycles adopt these testing activities, to improve the quality of software in context of your environment or business domain?

Biography:

Ken Brennock is the Technical Director, and a co-founder of Insight Test Services Ltd a sister company of Insight Consulting Ltd. Insight Test Services provides training, consultancy and resources in the testing domain.

Ken's experience spans over 14 years, and he has worked in many companies in that time, both large multinationals and small indigenous companies, including Motorola, Iona, Accuris and Macalla Software. He has been involved in testing his entire career, including hardware and software.

In Motorola Ken developed and tested software in a CMM level 3 environment, while also performing the role of Configuration Manager for the department. Since then he has set-up and managed a number of test teams and developed test processes for a number of companies. Focusing on processes that match the business needs. More recently Ken consults with Insight Test Services clients on how to improve their test and development process. Ken is an ISEB accredited trainer and develops and presents a number of testing courses. He has worked using ISO and CMM and using many methodologies, Waterfall, RUP, XP and Agile.



## **Isabel Evans**

*Testing Solutions Group Ltd.*

---

**Author: “Achieving Software Quality Through Teamwork”**

### **A Balanced Scorecard Approach for Assessing Test Value and Success**

Abstract:

Internal test metrics such as test progress, defect density, and TPI/TMM measures on process improvement do not reveal the complete picture of test value and success. By comparing common test metrics with those found in the Balanced Business Scorecard—financial, customer, internal, and learning/innovation metrics—we see the need to also report financial and customer measures, some of which are quantitative such as profits and others of which are more qualitative such as customer satisfaction. Learn to measure the financial impact of testing through productivity metrics and measures of how testing affects the total cost of quality. Include in your reporting qualitative assessments such as the customers' perception of the usefulness of testing, the visibility of testing on projects, acceptability measures, and estimation accuracy.

- Setting measures for all viewpoints of testing's value and success
- Data collection needed to support these metrics
- Make reports to colleagues and customers add value for decision making

Biography:

Isabel has 20 years experience in the IT industry, mainly in quality management, testing, training and documentation. She has helped organisations in development of procedures, standards and methods to aid testing of software during development and maintenance projects. She has managed test groups, and performed testing design and development for acceptance and system testing of packages and bespoke systems. She has also provided Quality Assurance Support, Release Management, and Customer Support for IT organisations. Most of her work has been with clients in the financial, communications and software sectors.

As well as presenting seminars and training courses to clients, Isabel has spoken on software quality, testing and test management at conferences in the UK, Europe and the USA including EuroSTAR, PSST, Quality Forum, BCS SIGIST, and the “Year 2000 and EURO Summit”. She regularly attends conferences, courses and meetings in her interest areas.

Isabel has been a member of various working parties and groups to contribute to improvement in software quality and testing, including the Quality Forum Testing Metrics Forum, the Customer Satisfaction Measurement working party, and the BCS SIGIST Test Standards Working Party, currently developing the non-functional testing standards.

Her book, “Achieving Software Quality Through Teamwork” was published by Artech in May 2004.



## **Graham Thomas**

*Independent Consultant*

---

### **Seven Key Measures for Software Testing**

Abstract:

Time and again I keep on encountering bad measurement, metrics and reporting on software testing projects.

Recently I came across the worst example that I had seen in over 20 year's experience of IT. And it got me thinking that surely it doesn't have to be like this?

Spurred into action I spent some time thinking about where the problem with bad software testing measurement and reporting originated and how to correct the problem.

Part of the problem comes from the fact that there isn't a standard set of measures, so should we actually get upset when software testers measure the wrong thing, in the wrong way, and then report it badly? Actually no! Not until there is a standard definition for software test measurement and reporting.

So there is the challenge for this presentation. To present a standard set of measures, metrics and reports for software testing so that there can no longer be any excuse.

This presentation proposes 7 key measures across the software testing lifecycle, covering; Planning, Risk, Test Preparation, Test Execution and Defect analysis. The presentation will also identify effective ways to present the 7 key measures in the form of a practical model.

Biography:

Graham works as an independent consultant with over 25 years experience of IT and has specialised in software testing since the early 1990's. Latterly his focus has been on program test management and quality assurance.

He has also worked as a developer, analyst and project manager.

He is an active member of the BCS SIGiST Standards Working Party, acting as secretary since 2001. He has presented papers at conferences, seminars and testing groups since 1995.





## Featured Speaker:

### **Johanna Rothman**

*Rothman Consulting Group, Inc.*

---

**Author: "Hiring the Best Knowledge Workers, Techies & Nerds"**

**Co-author: "Corrective Action for the Software Industry" & "Behind Closed Doors, Secrets of Great Management"**

### **Successful Software Management: 15 Lessons Learned**



Abstract:

Many software managers came to management through the technical ranks. Although they may have had plenty of technical training and mentoring, they frequently have to learn management skills the hard way, through trial and error. Johanna will describe some technical management tips and tricks learned through trial and error, focusing on technical managers and their particular issues.

You'll learn about a manager's job, how to create an effective work environment, and how you can help people do their best work.

Biography:

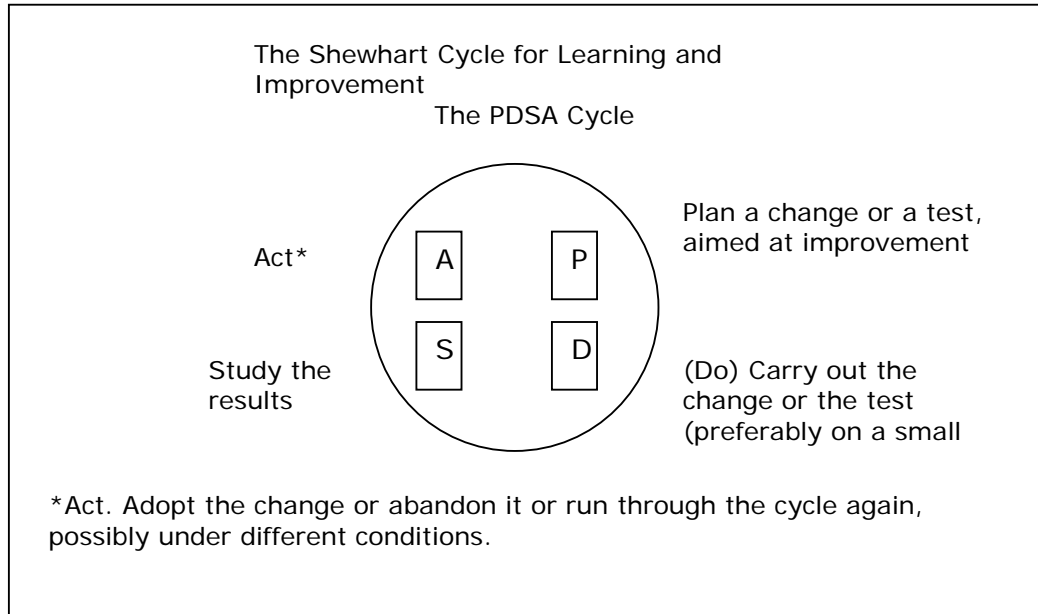
Johanna Rothman of Rothman Consulting Group, Inc., consults, speaks, and writes on managing high-technology product development. During her decade-long consulting career, she has assisted managers, teams, and organizations become more effective by applying her pragmatic approaches to the issues of project management, risk management, and people management. She's helped Engineering organizations, IT organizations, and start-ups hire technical people, manage projects, and release successful products faster. Her action-based assessment reports have helped managers and teams improve their projects, products, and financial results. She is a sought-after speaker and teacher in the areas of project management, people management, and problem-solving.

Johanna has written over 100 articles and papers, maintains two blogs, and is a frequent contributor for Fast Company's online career center, Software Development, Better Software, Computerworld.com, and StickyMinds.com. Johanna is a co-author (with Esther Derby) of the popular and pragmatic "Behind Closed Doors, Secrets of Great Management". She is also the author of the highly acclaimed "Hiring the Best Knowledge Workers, Techies & Nerds", and is a co-author (with Denise Robitaille) of "Corrective Action for the Software Industry". Johanna is also a host and session leader at the Amplifying Your Effectiveness (AYE) conference.

# ARTICLE: COMPETITIVE TESTING

Martin Jamieson, BT

Traditionally testers have followed a model in which the requirements are agreed and signed off at an early stage in the project lifecycle. The requirements have been split into functional and non-functional requirements and testers have divided testing into functional and non-functional testing. Non-functional testing has been defined as performance, security, usability etc. and the adherence to constraints e.g. the Data Protection Act.



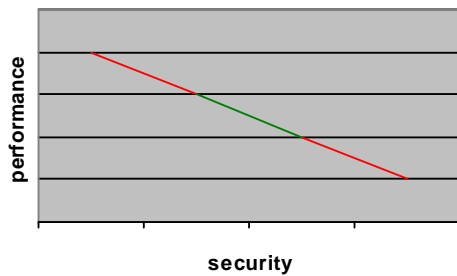
An evolutionary approach is now increasingly being used to manage projects. This recognises the role that feedback from the project stakeholders plays in refining and prioritising the key requirements. The idea that requirements can be frozen is regarded as anathema. Requirements will always change because we live in a changing world. We need to adapt to survive and manage requirements flexibly. An iterative project lifecycle with incremental releases into test is likely to become the norm. By delivering early and in small increments we quickly resolve ambiguities and reduce the cost of failure. This is essential if we are to remain competitive in a capitalistic market place. Minimising risk is maximising profit and profit based testing will test the most profitable requirements first. Testing is an intelligence gathering process. It is better to test something rather than nothing. It is better to test a partial solution and get advance warning of poor quality rather than wait for a full solution and find out when it is too late. This approach has the additional advantage of spotting "feature creep" at an early stage. Design features which cannot be traced to requirements need to be removed – they create unwanted complexity.

We should now consider which requirements are likely to change and whether it is useful to classify testing as functional and non-functional. Pure functions are essential necessities. They are what a system does and are not likely to change. As Tom Gilb has pointed out they are binary in nature – they are either present or not. The same also largely applies to the constraints. It is the attributes of these functional requirements which testers are likely to find the most interesting and challenging. It is where these attributes are scalar in nature that testers are most challenged. Often these requirements are not specified in a testable form. They are expressed in qualitative rather than quantitative terminology. Testers need to verify that the requirements are testable. We need to define a scale (unit of measure) and a meter (method for measuring). We also need to have a clearly defined goal.



The three measuring devices can be calibrated to the same scale but have very different practical uses.

Where the requirement is complex we probably need to define the sub requirements and test against these. It is the successful management and prioritisation of the scalar requirements which will give us competitive advantage. Arguably systems engineering is held in poor esteem today because these requirements have not been defined adequately and therefore not tested properly. Testers need to ask – what kind of requirement is this and how am I going to test it? If a requirement is not binary in nature then it must be quantified in order to be tested. It is also important to recognise that scalar requirements can intersect; for instance there is a trade-off between security and performance.



Requirements need to be considered as a whole. The red lines reflect unacceptable minimum levels.

It is worth noting that requirements which have traditionally been classified as non-functional do not neatly fit this label. E.g. access control is essential and fundamental to many systems. It is either a functional requirement or a design constraint. Design ideas often become functional requirements over time. In the UK an automatic gearbox in a car is still very much an optional extra; in some countries it has evolved into the standard. A design idea at inception can become a required function from the viewpoint of the developer. As testers the distinction between functional and non-functional is largely irrelevant. We have a whole set of requirements which must all be tested. E.g. some security requirements will be essential functional requirements, some might be constraints. Penetration testing must achieve a certain level of accreditation but the penetration testing itself will consist of defined scales and meters. It makes more sense to classify tests according to the aspects of the system we are testing. To define tests as functional or non-functional is often an arbitrary choice dependent on our perspective. From a dictionary point of view the term non functional implies the opposite to functional, i.e. redundant. It confuses our business partners who are not familiar with such an anachronistic term.



# THE TESTER

**September 2006**

**NEXT CONFERENCE**

**Tuesday  
19 September 2006**

## **Too Many Bugs Spoil the Broth**

- **Proving Our Worth: Quantifying the Value of Testing**
- **Lord of all Things**
- **XPeriencing Change: From Driven-Tester to Test-Driven**
- **Measurement Abuse in Software Testing Environments**
- **Keyword Driven Test Automation Illuminated**
- **The Nine Forgettings**

**IN THIS ISSUE:**

**FROM THE EDITOR**

**FUTURE SIGIST CONFERENCE DATES**

**NEXT MEETING – PROGRAMME**

**BCS SIGIST – EVERY TEST HAS A SILVER LINING**

**ANNOUNCEMENT**

**SPEAKER BIOS AND ABSTRACTS**

**ARTICLE: SUCCESSFUL SOFTWARE MANAGEMENT - FOURTEEN LESSONS LEARNED**



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

## FROM THE EDITOR

The June Conference was a resounding success – judging by the conversations I had with attendees and also borne out by the evaluation forms. A good mix of informative presentations and good humour!

Johanna Rothman, our Keynote Speaker, has kindly provided an article for inclusion in The Tester which you will find in this edition.

This was the last full programme created by our out-going Programme Secretary, Mark Fewster. Mark was thanked at the conference for his dedication to ensuring the SIGiST conference successes over the last seven years and was presented with gifts to mark his time with us. However, he's not going away – so we look forward to seeing a relaxed Mark at our future meetings!

Lloyd Roden has stepped into Mark's shoes and as you will see has created a very interesting programme for the September conference. We welcome Lloyd, who has been well known within the SIGiST for a number of years.

Please let your colleagues know of the date and content of our next conference on Tuesday 19 September – why not pin the conference programme on your notice board at work? Also, if you are wishing to attend a parallel session please note that there is restricted seating in these break out rooms.

We look forward to a bumper attendance in September!

Pam Frederiksen  
Communications Secretary  
Tel: 01483 881188 (Leysen Associates)  
Fax: 01483 881189  
Email: [pam@leysen.com](mailto:pam@leysen.com)

BCS SIGIST website: [www.SIGiST.org.uk](http://www.SIGiST.org.uk)

SIGIST Standards Working Party: [www.testingstandards.co.uk](http://www.testingstandards.co.uk)

SIGIST UML Testers Forum: [www.umltesters.org](http://www.umltesters.org)

## FUTURE SIGIST CONFERENCE DATES

14 December 2006

14 December 2006

13 March 2007

13 June 2007

## BOOKING INSTRUCTIONS

1. Download a booking form from:  
<http://www.SIGiST.org.uk/bookingForm.pdf>

A. Complete and fax to:	OR B. Post to:
Phil Dyson 01793 417444	Phil Dyson Specialist Groups Officer First Floor, Block D North Star House North Star Avenue Swindon SN2 1FA

## NEXT MEETING – PROGRAMME

### BCS SIGIST – Every Test Has a Silver Lining

**Tuesday 19 September 2006**

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

08:30	Coffee & Registration, Exhibition opens	
09:25	Introduction and Welcome – Stuart Reid, SIGIST Chairman	
<b>09:30</b>	<b>Featured Speaker</b>	
	<b>Proving Our Worth: Quantifying the Value of Testing</b> <i>Lee Copeland, Software Quality Engineering, Inc.</i>	
<b>10:30</b>	<b>Networking session and commercial break</b>	
10:45	Coffee & opportunity to visit the exhibition	
<b>11:15</b>	<b>Lord of all things</b> <i>Andy Redwood,</i> <i>Redwood Associates</i>	<b>Featured Speaker</b>
<b>12:00</b>	<b>XPeriencing Change: From driven-tester to test-driven</b> <i>Antony Marcano,</i> <i>testingReflections.com / etest associates (UK) Ltd</i>	<b>Workshop:</b> <b>Proving Our Worth: Quantifying the Value of Testing</b> <i>Lee Copeland, Software Quality Engineering, Inc.</i>
12:45	Lunch & opportunity to visit the exhibition	
<b>13:45</b>	<b>Measurement abuse in Software Testing Environments</b> <i>Les Hatton, Kingston University</i>	<b>Putting it into practice; XPeriencing Change: From driven-tester to test-driven</b> <i>Antony Marcano,</i> <i>testingReflections.com / etest associates (UK) Ltd</i>
14:45	Tea & opportunity to visit the exhibition	
<b>15:15</b>	<b>Keyword Driven Test Automation Illuminated</b> <i>Consultants</i> <span style="float: right;"><i>Mark Fewster, Grove</i></span>	
<b>16:00</b>	<b>Featured Speaker</b>	
	<b>The Nine Forgettings</b> <i>Lee Copeland, Software Quality Engineering, Inc.</i>	
17:00	Closing Remarks	

The SIGiST committee reserves the right to amend the programme if circumstances deem it necessary.

## ANNOUNCEMENT

### **SIGIST Library**

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, Techniques, Test Process Improvement

If you would like to know more about the library and books available, or for any queries, please contact Julie Gardiner on 07974 141436 or email her at [gardinerjulie@yahoo.co.uk](mailto:gardinerjulie@yahoo.co.uk). Alternatively, download the book loan form on the SIGIST website [www.SIGiST.org.uk](http://www.SIGiST.org.uk). Happy Reading!

## ABSTRACTS AND BIOGRAPHIES

### Featured Speaker:

### **Lee Copeland**

*SQE Inc.*

---

**Author:**

***“A Practitioner’s Guide to Software Test Design”***



### **Proving Our Worth: Quantifying the Value of Testing**

**Abstract:**

“Testers are such a nuisance. They waste time, create bugs, get in the way of shipping our products, lower the morale of our developers, and cost too much.” That’s how my interview with Rajan began. Rajan is a project manager for MegaCorp, my new consulting client. Later in the day, as I was talking with Rayanne, the testing manager, she countered, “Management doesn’t have a clue. They don’t see the value in our work. And why can’t they make up their minds — one day quality is the most important thing, the next day shipping the product is.”

I hear these verbal bombardments a lot. Perhaps you have heard them too. Perhaps they are flying overhead in your organization right now. How do these critical feelings arise? I believe that this antagonism often comes from a basic misunderstanding of the real purpose of testing.

Over the years writers have defined testing as a process of finding, a process of evaluating, a process of measuring, a process of improving. For a quarter of a century we as testers have been focused on the internal process of testing, while generally disregarding its real purpose.

The real purpose of testing is to create information. James Bach nailed it when he wrote, “The ultimate reason testers exist is to provide information that others on the project use to create things of value.” That is why testing exists — to provide information of value. So, when managers complain that testing “costs too much” perhaps they are really trying to say, “I’m not getting enough valuable information to justify the cost of testing.” When testers say “my management doesn’t see the value in our work” perhaps they are really trying to say, “My management doesn’t value the information I’m providing to them.” To prove our worth, to increase the value of testing, we must first focus on testing’s purpose — providing valuable information — not its process.

Join Lee as he discusses why quantifying the value of testing is difficult work — perhaps that’s why we concentrate so much on testing process—that’s much easier. But until we do this difficult work, until we prove our worth through quantifying our contribution, we should expect the bombardments to continue.

**Biography:**

Lee Copeland has over thirty years experience as an information systems professional. He has held a number of technical and managerial positions with commercial and non-profit organizations in the areas of applications development, software testing, and software development process improvement.

He began his career as a systems engineer for Data General Corporation providing pre-sales and post-sales technical support for their computers and software products. Later, he worked for the LDS Church managing software development and software process improvement. In the later position, he directed the creation of a testing organization using inspections, functional and structural testing, and metrics for the evaluation of the quality of both the development and testing processes. He and his staff worked to encourage and support software development teams in improving their own testing.



Now, as a consultant with Software Quality Engineering, Lee has developed and taught numerous training courses focusing on software development and testing based on his extensive experience. In addition, he provides consulting services to SQE's clients.

He is a well-known and highly regarded speaker at software conferences both in the United States and internationally. He currently serves as Program Chair for the STAR testing conferences. Lee is the author of *A Practitioner's Guide to Software Test Design*, a compendium of the most effective methods of test case design.

On a more personal note, Lee's turn-ons include long walks on the beach, waterfalls, orange juice (with pulp), yellow roses, a complex bass line, historical fiction, the smiles of babies, chocolate malts, Jules and Vincent, and backrubs with hot lotion.

## **Andy Redwood**

*Redwood Associates.*

---

### **Lord of all things**

Abstract:

As an independent Test Strategist I'm privileged to be able to work with many organisations – in fact some 15 in the last 10 years, some of them more than once.

The culture and size of an organisation can influence the project types, their size, the people who work on them and the quality they deliver. In fact most attributes are dependent on the Management style from Project Managers, through the Business areas, right up to the Board itself.

I've identified 3 philosophies that I believe are common in organisations. At times they exist in combination, driven by either a Business focus, or an IS focus. The balance can affect Business objectives, IS delivery, Capability Maturity and the corporate attitude to Testing.

This presentation is a bit of fun, but carries a serious message –

*Three Things for the Elven-kings under the sky,  
Seven for the Dwarf-lords in their Testing halls of stone,  
Nine for Mortal Project Managers doomed to die,  
One for the Dark Lord (or two) on his dark throne In the Land  
of the Two Ivory Towers where the Shadows lie.  
One Thing to rule them all, One Thing to find them,  
One Thing to bring them all and in the darkness bind them.....*

This presentation will examine the three things for the Elven Kings and I will share with you Gollum's observations.

Biography:

Andy was last year's winner of the EuroSTAR Testing Excellence Award.

He is a professional Test Manager working independently for a varied client base, usually in the financial services markets in the City delivering testing solutions on global projects.

Andy is a frequent speaker at International conferences and a regular guest at the SiGiST in London.



## Antony Marcano

etest associates (UK)Ltd.

---



### XPeriencing Change - From Driven-Tester to Test-Driven

Abstract:

More and more testing teams are faced with the prospect of their organisation adopting Test Driven Development (TDD). Some are making the transition smoothly whereas others are finding it a struggle, but why?

In this talk, Antony shares his experiences of TDD projects and the challenges faced by testing teams when adjusting to this seemingly counter-intuitive approach to software development. Much of the writing on TDD apparently targets developers, highlighting all the reasons why they'll never want to work in any other way, ever again. But, as testers, what's in it for us?

What if many of our shared frustrations with 'traditional' projects could be a thing of the past? What if someone could recount specific experiences showing that it is possible to work on a project where:

- You were involved in projects from the outset?
- Your tests weren't derived from a functional specification? They *were* the functional specification!
- Code was written in order to work with the automated tests rather than automated tests developed to work-around the code, resulting in highly-testable software?
- Developers couldn't say they were finished until they'd proven that the software works and that it works correctly (using tests you'd helped to design)?
- Regression is found and fixed so quickly that there's no time (or need) to log a bug?
- Developers frequently come to you for your expert advice in testing?
- Those apparently obvious and avoidable bugs rarely appear and it's actually a challenging to find things wrong with the system from day one?
- You knew within minutes that a Developer had checked in code that broke the system and you were alerted as soon as they'd checked in the code that fixed it?
- A build was almost always available for testing and could be deployed automatically at any time?

Sound too good to be true? So what's the catch? Change! For test teams, TDD isn't simply a case of changing when you are involved in the development cycle, it can involve a complete change in processes, tools and mind-set. In this talk, Antony will share his personal experience of these changes as well as his observations of others on their journey from being driven-testers to becoming test-driven.

Biography:

Antony Marcano of etest-associates (UK) Ltd. is an independent practitioner/consultant with experience on Agile projects spanning the last 6 years. He has worked in software testing for over 11 years across diverse sectors, covering numerous aspects of testing including test-management, test-automation, load & performance testing and security.

In his 'spare' time, he runs testingReflections.com, described by one well known software tester as "*the most influential software testing 'blog' site*". Antony's own weblog (<http://am.testingReflections.com>), listed on the 'blogroll' at TestDriven.com, is frequently updated with everything from links to tools, through to his own views and perspectives on software development and testing.

## Featured Speaker:

### **Lee Copeland**

SQE Inc.

---

**Author:**

**“A Practitioner’s Guide to  
Software Test Design”**

### **Workshop: Proving Our Worth: Quantifying the Value of Testing**

**Abstract:**

In the companion workshop to *Proving Our Worth: Quantifying The Value Of Testing*, designed for those who want to try the techniques Lee described, you can participate in an exercise designed to help you become proficient in “proving our worth.” We will divide into groups of four—two taking the role of clients and two taking the role of testers. Based on information supplied by the clients, testers will discover and define information that their clients would feel is valuable (worth the cost of the testing that creates it). Each group will present its findings and their work will be analyzed. Join us for an exciting learning experience.



**Biography:**

Lee Copeland has over thirty years experience as an information systems professional. He has held a number of technical and managerial positions with commercial and non-profit organizations in the areas of applications development, software testing, and software development process improvement.

He began his career as a systems engineer for Data General Corporation providing pre-sales and post-sales technical support for their computers and software products. Later, he worked for the LDS Church managing software development and software process improvement. In the later position, he directed the creation of a testing organization using inspections, functional and structural testing, and metrics for the evaluation of the quality of both the development and testing processes. He and his staff worked to encourage and support software development teams in improving their own testing.

Now, as a consultant with Software Quality Engineering, Lee has developed and taught numerous training courses focusing on software development and testing based on his extensive experience. In addition, he provides consulting services to SQE’s clients.

He is a well-known and highly regarded speaker at software conferences both in the United States and internationally. He currently serves as Program Chair for the STAR testing conferences. Lee is the author of *A Practitioner’s Guide to Software Test Design*, a compendium of the most effective methods of test case design.

On a more personal note, Lee’s turn-ons include long walks on the beach, waterfalls, orange juice (with pulp), yellow roses, a complex bass line, historical fiction, the smiles of babies, chocolate malts, Jules and Vincent, and backrubs with hot lotion.

***Les Hatton***

*Kingston University.*

---

## **Measurement abuse in Software**

### **Testing environments**

Abstract:

Unfortunately no abstract was available at the time.



## **Antony Marcano**

*etest associates (UK)Ltd.*

---



### **Workshop: Putting it into practice- From Driven Tester to Test Driven**

#### Abstract:

We expect that Antony's earlier talk will generate more questions that there will be time for him to answer. This session will be an open discussion where you can put these questions to him and he can elaborate on some of the points touched upon in his main talk, such as:

- Practical steps for smoothing the transition to Test Driven Development
- How Exploratory Testing fits into TDD projects
- Tools mentioned in the talk, such as:
  - FIT/FitNesse
  - xUnit
  - Cruise Control (Continuous Integration)
- Non-functional testing and TDD
- Gerard Meszaros' "Test Doubles" (as in "Stunt Doubles")
- Making the bug-tracking system redundant
- Skills-mix in an 'agile' testing team
- *Or any other point you want to focus on*

The shape that this session takes will be entirely up to you!

#### Biography:

Antony Marcano of etest-associates (UK) Ltd. is an independent practitioner/consultant with experience on Agile projects spanning the last 6 years. He has worked in software testing for over 11 years across diverse sectors, covering numerous aspects of testing including test-management, test-automation, load & performance testing and security.

Some of the companies that Antony has worked with over the years include the BBC, Vodafone, Barclays Bank, FT.com, BT and the law firm Freshfields Bruckhaus Deringer.

In his 'spare' time, he runs testingReflections.com, described by one well known software tester as "*the most influential software testing 'blog' site*". Antony's own weblog (<http://am.testingReflections.com>), listed on the 'blogroll' at TestDriven.com, is frequently updated with everything from links to tools, through to his own views and perspectives on software development and testing.

## **Mark Fewster**

*Grove Consultants*

---



**Author:**  
**“Software Test Automation”**

### **Keyword Driven Test Automation Illuminated**

Abstract:

Test Automation has come a long way over the past 20 years. IN that time many of today's most popular test execution automation tools have been brought into use and a range of implementation methods have been tried and tested. Some of these have been developed further in an attempt to drive down the build and maintenance costs of automated testing and to increase its benefits.

Many organisations that have been successful with test execution automation have employed a data-driven approach. This in turn has been developed further into more sophisticated approaches that are often termed keyword-driven (though other terms are used).

While a simple data-driven approach is easy to describe, keyword-driven is a little more difficult to pin down. There are now many implementations of keyword-driven test execution automation and some are hard to tell apart from the more advanced data-driven versions. So what is keyword-driven test automation? Perhaps we should ask “What isn't keyword-driven test automation?”

This presentation provides an objective analysis of keyword-driven test automation. By considering the various components of implementations, advantages and disadvantages of each the benefits and pitfalls within different contexts are identified. The presentation concludes by trying to answer the question “What is, and what is not, keyword-driven test automation?”

Anyone involved with test execution automation should find this presentation of interest. For those with little or no experience of keyword-driven approaches then the presentation will give a solid understanding of what can be achieved and in outline, how. For those already familiar with keyword-driven approaches then the presentation will bring greater clarity and possibly some new ideas for how they can improve their own keyword-driven approaches.

Biography:

Mark Fewster has 20 something years experience in the software industry, much of this specialising in testing. He held posts from software developer to development manager for a multi-platform graphical application vendor, where he implemented a testing improvement programme and successfully developed and implemented a testing tool that led to dramatic and lasting savings for the company.

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

Until recently Mark served as Programme Secretary for the British Computer Society's Specialist Group in Software Testing. He was also involved in the drafting of what became the Software Component Testing Standard BS7925 and in the development of the original ISEB software testing qualifications.

Mark has co-authored with Dorothy Graham the book "Software Test Automation" published by Addison-Wesley.

## Featured Speaker:

### **Lee Copeland**

SQE Inc.

---

**Author:**

***“A Practitioner’s Guide to  
Software Test Design”***

### **The Nine Forgettingings**

**Abstract:**

People forget things. Simple things like keys and passwords and the names of friends long ago. People forget more important things like passports and anniversaries and backing up data. But Lee is concerned with things that the testing community is forgetting — forgetting our beginnings, the grandfathers of formal testing and the contributions they made; forgetting organizational context, the reason we exist and where we fit in our company; forgetting to grow, to learn and practice the latest testing techniques. Join Lee for an explanation of these forgettingings and six others and the negative effects of each if we fail to remember.



**Biography:**

Lee Copeland has over thirty years experience as an information systems professional. He has held a number of technical and managerial positions with commercial and non-profit organizations in the areas of applications development, software testing, and software development process improvement.

He began his career as a systems engineer for Data General Corporation providing pre-sales and post-sales technical support for their computers and software products. Later, he worked for the LDS Church managing software development and software process improvement. In the later position, he directed the creation of a testing organization using inspections, functional and structural testing, and metrics for the evaluation of the quality of both the development and testing processes. He and his staff worked to encourage and support software development teams in improving their own testing.

Now, as a consultant with Software Quality Engineering, Lee has developed and taught numerous training courses focusing on software development and testing based on his extensive experience. In addition, he provides consulting services to SQE’s clients.

He is a well-known and highly regarded speaker at software conferences both in the United States and internationally. He currently serves as Program Chair for the STAR testing conferences. Lee is the author of *A Practitioner’s Guide to Software Test Design*, a compendium of the most effective methods of test case design.

On a more personal note, Lee’s turn-ons include long walks on the beach, waterfalls, orange juice (with pulp), yellow roses, a complex bass line, historical fiction, the smiles of babies, chocolate malts, Jules and Vincent, and backrubs with hot lotion.



# Article: Successful Software Management - Fourteen Lessons Learned

**Johanna Rothman**

Shortly after I became a manager, I dragged myself home from work, flopped on the couch, and sighed to my husband, "This management stuff is hard. Nothing I learned in school prepared me for this people stuff. And that "management training" —that was just form-filling-out nonsense. The soft skills—dealing with people—is the hardest." My husband chuckled and commiserated.

If you are like me, and you started your professional career as a technical person, this "management stuff" is hard to do. Not the forms, although the forms can be irritating, but knowing how to deal with people and completing the work your organization expects of you is difficult. I have now had over fifteen years of management experience and I have learned a number of lessons about managing people.

## Define the Manager's Role

When you become a manager, your role is to organize purposefully [1]. For me, that means creating an environment in which people can perform their best work. As a software manager, that means I work to create business value by balancing the needs of the business, the employees, and the environment. There is no One Right Way to do this; every organization is different. However, these lessons have served me well in numerous organizations.

## Know What They Pay You to Do

I have been a manager of developers, testers, and support staff. You would think it would be easy to know what the company paid me to do. However, my mission as a test manager, to report on the state of the software is sometimes different from what my organizations desired: find the Big Bad Bugs before the customer does; or bless this software. Even my mission as a development manager, develop the team members as much as the software, can be different from what one organization desired: create software just good enough that we can be bought out.

My mission does not have to be the same as yours, and you may modify your mission as your organization changes. However, delivering on your mission as a manager is what your organization pays you to do. What is important is to notice when your title, your mission, and what the company pays you to do are not synchronized.

One QA Manager said it this way, "My management only wants to me to manage the testing, not raise risks, look for process improvement opportunities, or even gather and report on what I think are standard metrics. My manager and I are both frustrated. Focusing on *just* the testing is wrong." This QA manager has at least one alternative—change his title so that he and the organization both know that he is not attempting to perform organization-wide process improvement, to clarify expectations in the organization.

Doing what the organization pays you to do, and *not* doing what they do not pay you to do makes a huge difference in how successfully you and your group can accomplish your mission. Make sure you clarify your mission at your organization, so you can create a to-do and not-to-do list. These lists help you plan the work—for you and your group.

One development manager who temporarily took over installations from the tech support people realized that he no longer had a development team, but an installation support team. The development manager put installations on his not-to-do list and developed a plan to move installations back to tech support.

When you align yourself with your manager's priorities, you do the work they pay you to do.

## Plan the Work: Portfolio Management

It is easy to be reactive at work, and feel buffeted by the requested changes of your group. It is harder and necessary to be proactive and plan your group's work, even if that work changes every week. For me, planning includes these activities: identifying the project portfolio (new work, ongoing work, periodic work, *ad hoc* work), developing strategies for managing the work for each project, and knowing what done means for each project. One of the questions I like to ask is "How little can we

do?" I do not want to shortchange any project, so by asking about the minimum requirements, I can accommodate more projects successfully.

Part of planning the work is assigning the people to projects. I assign people to one important project, and allow them to take on little bits and pieces of much less important work when they need a break or are stuck on the important project. I avoid context switching (moving from one unrelated task to another) as much as possible.

### **Accept only one #1 priority at one time**

I have worked for many managers who demanded my staff and I work on several top priority projects simultaneously.

Senior managers perform different work than first-line and middle managers. It is not possible for senior managers to work on more than one top priority task at one time, but because they tend to have more wait states in their work, these senior managers are under the illusion they are working on several top priority projects at the same time.

Middle and first-line managers can only work on one #1 priority task at one time. However, sometimes we confuse urgency and importance [2]. At one organization, I would arrive at work in the morning, check my voicemail, and respond to all the voicemail requests. That took me until noon. I would check my email and voicemail after lunch, and run around responding to those urgent requests. After a week of this, I realized I wasn't performing any of the important work, such as planning for the group and lab, reviewing critical development plans, or planning my hiring strategy. And, I would notice that although people marked their emails and voicemails *high priority*, they didn't utilize the information I had given them at the time I responded.

I stopped immediate response to urgent requests, and replanned my days. I still checked voicemail and email, but I tended to ask more questions about the deadlines for these requests. Prioritizing requests helped me manage my management time.

I still had the problem of too many high priority projects coming into my group, so I asked my manager these questions:

- If you could have one project first, which one would it be?
- What are the consequences if we release any of these projects late?

We talked and negotiated which projects had to be completed when and why. When I understood the tradeoffs between projects, I was able to manage the work coming into my group.

### **Commit to Projects After Checking With Your Staff**

Business needs change, and sometimes, your manager will grab you in the hall, and say, "Hey, can you do this project now, and finish it in two months?" Or, a senior management planning committee will call you into their meeting, and say, "We need this project now. Can you commit to it?"

It is very tempting to say yes. And saying yes is exactly the wrong thing to do. You can say, "Let me check to see if my previous estimate is still accurate, and I will get back to you before 5pm today."

If you say yes, you are training your senior management to ask you for answers when you do not know the answers, and you have committed your staff to a project that may not be the same scope you originally estimated.

### **Hire the Best People for the Job**

Especially if you manage many projects, your greatest leverage point is in hiring appropriate staff for the jobs you need filled. Too often, we hire people who have similar technical skills and personalities as the people already in our groups. Hiring people who are *just like the ones we have now* are not always the best people for the job.

When you hire people your staff thinks are great, you increase morale in the group, and you increase your group's capacity over time. I recommend you develop a hiring strategy, so you know the kinds of technical and soft skills you are looking for, and that you choose a variety of techniques for interviewing.

I have found auditions [3,4,5] to be an essential technique for interviewing technical staff. I normally create auditions of 30-45 minutes duration, so I can see how a person works in a particular setting. Auditions help candidates show what they can do. If you organize a congruent audition, you do not trip people up on esoteric ideas or jargon; you create a simplified situation that the candidate could encounter at work. Watching the candidate, or having the candidate explain their answers/results is a powerful interview technique.

You can create auditions for any position, including project managers, developers, testers, writers, support staff, analysts, systems engineers, product managers, program managers, and people managers. Define the behaviors you require in a position, and then create an audition, using your products or open source products to see the person at work. Create auditions that are 30 minutes long to start. If you are having trouble deciding between multiple candidates, define another audition that is one hour long, and invite the candidates back to see how they manage that audition. Auditions show you how the person works at work, priceless information.

I also recommend behavior-description interview questions [5,6], to understand how a candidate has performed in previous jobs. Behavior-description questions are open-ended, and ask the candidate to tell you the story of previous work. For example, if You would like to understand how a project manager deals with a project team who has not yet met a schedule, you could ask this series of questions: "Have you ever managed a project where the team had trouble meeting the schedule?" If the answer is no, you can decide if the project manager has enough experience to manage your team. If the answer is yes ask, "What did you do? What actions did you take on that project to help the project team meet the schedule?" The answers you hear will help you assess that candidate's ability to work in your organization.

### **Preserve Good Teams**

Part of my hiring strategy is to hire people who fit into my already-existing team, but sometimes you inherit teams, or a project has completed and a team is ready to move on. When a team is successful, I try to keep the team together, so they can continue working well together. I may bring more people into the team, one at a time, especially if the team has been highly productive. But I do not scatter the productive team and hope they will form more productive teams. That just reduces the productive people's productivity.

Teams can overcome bad management and bad process, but they cannot overcome a team un-jeller. A team un-jeller is the person who walks into the lunchroom, and suddenly everyone else leaves. Or, the un-jeller creates an argument out of every conversation.

If you have a team un-jeller, find another place for that person to work, preferably at your competitor.

### **Avoid Micromanaging or Inflicting Help**

Many of us were software developers, testers, analysts, or some other technical role before we became managers. When we were technical contributors, we knew how to perform the technical jobs. However, once you have been a manager for a while, you probably do not know precisely how to perform the employee's job.

I once had a boss who liked to creep into my office, stare over my shoulder, and say, "On line 16, shouldn't that be a..." By the time he had reached the 16, I jumped out of my chair, flustered, with my concentration gone. Micromanagement neither gets the job done faster, nor does inflicting advice or help.

On the other hand, you and the employee both need to know that the employee is progressing. I ask my staff to decide when they have been stuck for too long (time-box the work). Some tasks require weeks of study, but most tasks require days or hours. If the employee spends more than the agreed-upon time on the task, their job is to ask for help. As the manager, your job is to find them help, not necessarily inflict your help.

### **Treat People Individually and With Respect**

Buckingham and Coffman [7] claim that each employee's relationship with their manager is key to that employee's success and long-term happiness in the organization. That means we need to treat people fairly, but uniquely, so that we build and maintain the best possible relationships with each employee.

Everyone has their own preferences, especially in their communications patterns, and how they organize their thoughts about their work. Some people prefer email communications; some prefer in-person discussions. Some people want to understand all the reasons behind your requests, and others will take the request at face value. Some people need to gather data to make decisions; others will develop a model about how the situation and make a decision based on their model.

It does not matter if people work top-down or bottom-up, or if they want to talk in person or by email. What matters is that you, within reason, accommodate everyone's uniqueness.

I once managed two very talented developers. They shared a large office. Begrudgingly, they allowed me to have 20-minute one-on-ones with each of them every two weeks. In between, if I wanted to talk to either of them, I had to email them first—dropping in was not allowed. I treated them differently than the other people in my group, but fairly, taking their preferences into account.

They frequently worked on the same software. They never spoke to each other aloud, they only communicated via email, even though they shared an office. Because they were so successful at their work together, and even mentoring others in the organization by email, their communications preferences were a bit odd, but acceptable. If I had tried to change them, to meet my needs and work with them the same way I worked with the other people, none of us would have been happy.

### **Meet Weekly With Each Person**

Even if you have hired stars, you still need to know each person's progress on their tasks, and how the project as a whole is progressing. I use one-on-ones weekly to meet with each person. We discuss the employee's progress on their tasks. Sometimes, tasks are amorphous and difficult to know when to stop or if the employee needs help. I ask each employee to show me visible progress on each task: drafts of plans, multiple designs, prototype test results, anything that shows me the employee is making progress and is not stuck. If the employee needs help to complete the task, we discuss which kinds of help are appropriate.

I receive many benefits from weekly one-on-ones with my staff. I learn weekly what everyone is doing, and I can track that in my notebook, so it is easy to write up useful performance evaluations, including examples of successful and not so successful actions the employee has taken over the year. And, because we meet weekly, I can give feedback each week, not when we make time. I also reduce the number of staff interruptions, because everyone knows they can ask me non-urgent questions in the one-on-one. I can perform weekly career development and learn if my staff has personal issues affecting their ability to do their jobs.

If I am managing more than eight people, I meet biweekly with more senior staff, because they need less direct supervision.

Some of you are probably thinking you do not have time to meet with everyone once a week. However, if you do not set up specific times to meet with everyone, you tend to either not know what people are doing, or you are interrupted frequently by your staff with questions.

### **Plan Training Time Each Week**

Technical work is constantly changing, and most of the technical people I know enjoy learning new things. If you have a budget for formal training, that is great. Even if you do not have a budget, plan training time each week, in the form of brown-bag lunches, presentations from other groups in your organization, an internal user-group meeting of one of your tools, or presentations from people in your group about their successes or difficulties.

I use the weekly group meeting as a time to deliver the training. When I managed development groups, I organized this internal training: technical leads of other sub-projects to explain their architecture and API to other groups, testers to explain patterns of defects they found, different techniques for peer review, or discussion of a particularly interesting article in one of the technical magazines someone had read.

### **Fire People Who Cannot Perform the Work**

Even when you meet regularly with your staff, encourage your staff, and acquire help when they need it, some people in your group may not be able to perform to the level that you require. First, make sure you have been specific and given feedback to the employee, with examples of inadequate

behavior. If the employee understands the lack of performance, you can choose whether to coach the person, or perform a get-well plan, or in radical circumstances, escort the employee out the door.

Keeping non-productive employees has direct and indirect costs. The direct costs are easier to define: you are paying a salary and benefits and not receiving the expected work. The indirect costs are much more subtle and more damaging.

When you continue employing an inadequate employee, the morale of the entire workgroup declines. If morale declines enough, your best people will leave. Not only do you have someone in your group who is not successful, that person has driven away the people who are the most successful.

In addition to low morale, you and your group accomplish less than you expected. You are not just accomplishing less because of the one employee who cannot work at the level you require; that person probably has handoffs to others in your group, and those other people will be delayed by the inadequate work.

I once inherited a group where the previous management had "spared" an employee from previous layoffs, because he was having personal problems. Those personal problems affected his work—he did not always come to work, he was late on every deliverable, and he was unable to perform most of his work. In my one-on-ones with the employee, I gave him examples of his work and asked if he was able to work. He said yes. (If he had said no, we would have put him on short-term or long-term disability.) We chose to perform a get-well plan, which the employee stopped after a week. After the employee left, the morale in the group jumped dramatically, and we were able to accomplish more work.

### **Emphasize Results, Not Time**

I have worked for senior managers who rewarded individuals on the basis of their work hours—who started early and stayed late. Unfortunately, these managers had no ability to understand the results the long-working employees imposed on the rest of the organization: buggy code, inadequate designs, and tests that did not find obvious problems. When people work long hours, their productivity decreases, not increases [DeMarco, Peopleware]. In [Slack \[8\]](#), Tom DeMarco says, "Extended overtime is a productivity–reduction technique." The longer people stay at work, the less work they do. Instead they perform the life activities they are not performing outside of work.

Make it possible for people to only work 40 hours a week. The less overtime people put in, the better their work will be.

If people tell you they are working long hours because they cannot accomplish anything in their regular work weeks, ask your staff where they spend their time. Look for patterns such as multi-tasking, or meetings that do not have any productive output. Use your management power to discover and remove the obstacles preventing people from working a 40-hour week.

### **Admit Your Mistakes**

Sometimes, those obstacles to people completing their work successfully in 40 hours arise from your management mistakes. It is difficult, and sometimes embarrassing to have to admit you have made a mistake. In my experience, when I have admitted mistakes to my staff, they've respected me more for it.

### **Recognize and Reward Good Work**

Money is not an adequate reward for many technical people. If people think they are paid fairly, more money is not enough of a reward. Recognition of good work and the opportunity to perform meaningful work [Kohn] is much more important than monetary rewards. Lack of money can be a demotivator, but only money is not sufficient for a significant reward.

Kohn [9] says, "[Rewards] motivate people to get rewards." If your organization has trained employees to expect money as a reward, this appreciation technique may seem small. Try it anyway.

When I use appreciations as a recognition technique I say, "I appreciate you Jim, for your work on the blatz module and API definition. Your work made it possible for Joe to write great tests and for me to predict the project's progress." Appreciations between peers could mean even more than money from you. When you appreciate a person for good work and you explain what the work meant to you, you are motivating the person to continue performing similar work.

In addition, consider time off, group activities, movie tickets, or funny awards, such as *best recursion of the week* as recognition techniques.

The most important part of rewards is to make sure the recognition and/or reward is congruent with each person's performance. Your staff knows who is performing well and who is coasting. If you recognize and reward evenly, you are not differentiating between outstanding performance and adequate performance. Make sure you reward a person's entire contribution (the entire work product, including how good the work product is, the timeliness of the deliverable, and the person's ability to work with others, whatever else is important to you), not just the size or quality of the work.

### Summary

Managers exist to help people do their best work to serve the business of the organization. Technical people can make great managers, as long as they understand people and to want to succeed at working with them. Many successful technical managers took the time to learn about management, putting as much effort (if not more) than the effort they took to learn the necessary technical background for the technical jobs. Managers do not have to be perfect; they have to be good enough to create a working environment for their employees to deliver great work.

© 2003 Johanna Rothman

### Acknowledgements

I thank Dwayne Phillips and the Crosstalk reviewers for their review of this article.

### References

1. Magretta, Joan. *What Management Is: How it Works and Why It is Everyone's Business*. The Free Press, New York. 2002
2. Covey, Stephen R. *The Seven Habits of Highly Effective People*. Simon & Schuster, New York. 1989.
3. DeMarco, Tom and Tim Lister. *Peopleware: Productive Projects and Teams*, 2<sup>nd</sup> edition. Dorset House, New York. 1999.
4. Weinberg, Gerald M. *Congruent Interviewing by Audition*, in *Amplifying Your Effectiveness, Collected Essays*. Dorset House. New York. 2000.
5. Rothman, Johanna. *Hiring Technical People*, Dorset House, New York, 2003.
6. Janz, Tom et al. *Behavior Description Interviewing*, Prentice Hall, Englewood Cliffs, NJ. 1986.
7. Buckingham, Marcus and Curt Coffman. *First, Break All the Rules: What the World's Greatest Managers Do Differently*. Simon & Schuster. 1999.
8. DeMarco, Tom. *Slack*. Broadway Books, New York, 2001.
9. Kohn, Alfie. *Punished by Rewards*, Houghton-Mifflin, New York, 1993.

Issue  
**18** **THE TESTER**  
*December 2006*

**NEXT CONFERENCE**

**Thursday  
14 December 2006**

**Time And Test  
Wait For No Man**

- Preventative Testing
- Formal Reviews – improving results
- Using Orthogonal Arrays
- Steps towards quality governance
- Web technology for testers
- Improving Test Quality
- Testing as measurement activity
- Test automation model
- Communication, leadership and innovation.

**IN THIS ISSUE:**

FROM THE EDITOR

FUTURE SIGIST CONFERENCE DATES

NEXT MEETING – PROGRAMME

BCS SIGIST – TIME AND TEST WAIT FOR NO MAN

ANNOUNCEMENTS

ABSTRACTS AND BIOGRAPHIES

ARTICLE: GONZO QA: FEAR AND LOATHING IN THE MORNING

ARTICLE: HORSES FOR COURSES



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

## FROM THE EDITOR

Here we are near the end of another year! Our next conference is on Thursday 14 December at which we welcome Rick Craig as our Keynote Speaker and a packed programme of interesting speakers to fill the programme for the day.

This time Lloyd Roden, our Programme Secretary, has introduced some mini-tracks of just 20 minutes. He plans to include mini-tracks at future conferences, so if you have thought of presenting a paper but found the prospect of 40 minutes a bit daunting then please contact Lloyd or me.

As our December conference is in the festive season we have a guest presenter with a difference to bring the day to a close. Neil Mullarkey from the Comedy Store will provide a presentation about communication, leadership and innovation in the workplace and we hope you will find this to be an entertaining end to the day.

I am pleased to include articles in The Tester this month from two members Kevin Buchta and Martin Cunnington. My thanks to both. If you would like to include an article in future issues then my contact details are below!

The December conference is likely to be very well attended, so please book your place early! See you there!

Pam Frederiksen  
Communications Secretary  
Tel: 01483 881188 (Leysen Associates)  
Fax: 01483 881189  
Email: [pam@leysen.com](mailto:pam@leysen.com)

**BCS SIGIST website:** [www.SIGiST.org.uk](http://www.SIGiST.org.uk)

**SIGIST Standards Working Party:** [www.testingstandards.co.uk](http://www.testingstandards.co.uk)

**SIGIST UML Testers Forum:** [www.umltesters.org](http://www.umltesters.org)

### FUTURE SIGIST CONFERENCE DATES

**13 March 2007**

**13 June 2007**

### BOOKING INSTRUCTIONS

1. Download a booking form from:  
<http://www.SIGiST.org.uk/bookingForm.pdf>

A. Complete and fax to:	OR B. Post to:
Phil Dyson 01793 417444	Phil Dyson Specialist Groups Officer First Floor, Block D North Star House North Star Avenue Swindon SN2 1FA



## NEXT MEETING – PROGRAMME

<b>BCS SIGIST – Time and Test Wait for No Man</b> Thursday 14 December 2006 Royal College of Obstetricians and Gynaecologists, 27 Sussex Place, Regent's Park, London NW1			
08:30	Coffee & Registration, Exhibition opens		
09:25	Introduction and Welcome – Stuart Reid, SIGIST Chairman		
09:30	Featured Speaker		
	<b>Preventative Testing</b> <i>Rick Craig, Software Quality Engineering, Inc.</i>		
10:30	Networking session and commercial break		
10:45	Coffee & opportunity to visit the exhibition		
11:15	<b>Formal Reviews – how to radically improve your results</b> <i>Fran O'Hara,</i> <i>Insight Test Services</i>	Featured Speaker	<b>Workshop: Web technology for testers; testing beneath the GUI</b> <i>Paul Gerrard, Gerrard Consulting</i> <b>Please bring a wireless enabled laptop</b>
12:00	<b>Steps towards achieving Quality Governance™</b> <i>Sarah Saltzman,</i> <i>Compuware</i>	<b>Workshop:</b> <b>Using Orthogonal Arrays and All Pairs in Test Design</b> <i>Rick Craig, Software Quality Engineering, Inc.</i>	
12:45	Lunch & opportunity to visit the exhibition		
13.45	<b>Mini-Track</b> <b>Improving 'Test Quality' through 'Quality Stubs'</b> <i>Lee Clifford,</i> <i>Virgin Mobile</i>	<b>Mini-Track</b> <b>Test Automation Model</b> <i>Nitesh Shah,</i> <i>CapGemini</i>	<b>Workshop: Web technology for testers; testing beneath the GUI</b> <i>Paul Gerrard, Gerrard Consulting</i> <b>Please bring a wireless enabled laptop</b>
14:05	Featured Speaker		
	<b>Testing as a Measurement Activity</b> <i>Rick Craig, Software Quality Engineering, Inc.</i>		
15:00	Tea & opportunity to visit the exhibition		
15:30	Featured Speaker		
	<b>Improvisation</b> <i>Neil Mullarkey, The Comedy Store.</i>		
17:00	Closing Remarks		

The SIGIST committee reserves the right to amend the programme if circumstances deem it necessary.

## ANNOUNCEMENTS

### **SIGIST Library**

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, Techniques, Test Process Improvement

If you would like to know more about the library and books available, or for any queries, please contact Julie Gardiner on 07974 141436 or email her at [gardinerjulie@yahoo.co.uk](mailto:gardinerjulie@yahoo.co.uk). Alternatively, download the book loan form on the SIGIST website [www.SIGiST.org.uk](http://www.SIGiST.org.uk). Happy Reading!

### **UML Testers' Forum**

Are you interested in model-based development and testing? If so, we invite you to the next UML Testers' Forum meeting on Monday 18th December at the BCS's London offices, Davidson Building, 5 Southampton Street, London WC2E 7HA.

The main presentation is entitled 'From Use Case to Test Case' and describes a UML development project from a tester's perspective. The meeting is from 1300-1600hrs and is preceded by a light buffet lunch at 1200hrs. For full details of the programme and to register for a (free) place please visit the Forum's website at [www.umltesters.org](http://www.umltesters.org)."

## ABSTRACTS AND BIOGRAPHIES

### Featured Speaker:

#### **Rick Craig, SQE Inc.**

#### **Author: "Systematic Software Testing" Preventative Testing**

##### Abstract:

It is common knowledge that the longer it takes to find and fix a defect the more it costs.....Several effective techniques such as inspections and reviews have been used for years to facilitate the early detection of defects. There are, however, other lesser known techniques that have been successfully employed by progressive engineers for an equally long period of time. We call these techniques, simply preventive testing. In a nutshell, preventive testing is the concept of using test cases as a way to model the requirements specifications before the design and coding of the system begin. When testers create test cases, they are focusing not only on whether the system works correctly, but on how it might fail. Most testers have written test cases that they knew would fail before they ever even executed them!! If those tests were written early, many defects could be prevented all together.



We do concede, however, that often there are no viable requirements specifications, which makes preventive testing problematic. In this instance, we employ a preventive technique known as "inventories" as a way of determining the functionality of the system from a testing perspective. Together these techniques are an effective addition to the techniques available to testers.

##### Learn:

- How testers can improve the quality of requirements specifications
- How testers can discover defects prior to the design and coding of the system
- What to do when adequate requirements specifications do not exist
- Shortfalls of preventive techniques.

##### Biography:

Rick Craig is a frequent speaker at testing conferences and is well-received worldwide as a test and evaluation instructor. He has implemented and managed testing efforts on large-scale, traditional, and embedded systems, and co-authored a study that benchmarked industry-wide processes. Rick is also co-author of *Systematic Software Testing*.

## ***Fran O'Hara, Insight Test Services Formal Reviews – how to radically improve your results***

Abstract:

We all know formal document and code reviews are supposed to be a great way to find major problems/defects early in the lifecycle. Reviews should improve product quality and help reduce timescales and rework costs as well as being a good way to learn and build capability in teams.

Most of us are doing reviews in some form or another so everything is OK, right? WRONG! Most formal reviews are poorly planned and managed and are not delivering anything near their effectiveness potential.

Also, testers need knowledge to test – reviews are a practical way to gain much of that requirements/system knowledge. Testers are excellent at finding documentation faults so their involvement adds considerable value. Key documents that benefit significantly from collaborative review involving developers, business analysts, users and testers include User Requirements and Functional Specifications as well as Test Strategies and Plans

This presentation will take the main industry reported pitfalls as well as participants issues with reviews and present an approach to formal reviews that

- addresses issues such as poor buy-in from reviewers, poor planning, lack of preparation, inefficient meetings, etc.
- is practical, efficient and, most importantly, effective (finds a high percentage of major faults)

A light weight definition of the peer review process with supporting templates and review checklists will be made available.

Biography:

Fran is (co)founder and director of

- Insight Test Services (2003), providing test consulting, training and managed test services,
- Insight Consulting (1996), providing process improvement services, and
- RPI Alliance (2002), advancing process improvement know-how.

He specialises in pragmatic approaches to process improvement and associated best practices with a particular emphasis on people issues.

Fran is a regular speaker at process improvement and testing conferences and seminars, including ICSTest, SofTest, BCS SIGIST, EuroSTAR, European SEPG and the U.S. SEPG. He is an ISEB/ISTQB tutor, a trained SEI CMM lead assessor and TickIT auditor, a fellow of the Irish Computer Society and co-founder of the Irish SIG in Software Testing - SoftTest.

Previously he worked as development manager in the area of safety-critical software for implantable defibrillators in Australia and in solid-state physics research in the Netherlands.



## **Sarah Saltzman, Compuware**

### **Steps towards achieving Quality Governance™**



#### Abstract:

Today's IT organisations face many challenges, from cutting costs to enabling business innovation. IT is an integral part of business today, which means that a focus on quality is a high priority. Quality must be built in to every phase of the development lifecycle; from requirements gathering through to testing.

In conjunction with Forrester, Compuware commissioned a study on quality initiatives within large IT organisations to determine how quality is perceived and what steps are being taken to ensure that Quality becomes an integral part of the business of IT. In this presentation Sarah will share the results of the survey and introduce Compuware's Quality Maturity Model, which can be used to determine your level of quality maturity within your own organisation. Sarah will then break down each phase of the model giving insights in to approaches that can be taken to improve software quality at each phase of the model as we move from Quality Control to Quality Governance™

#### Biography:

With more than 21 years experience within the IT industry, Sarah has a thorough technical and business level understanding of user and management requirements from mission-critical software applications. Sarah joined Compuware in 1995 and is responsible for innovating solutions that meet specific Quality and Governance requirements for key Compuware clients.

As a frequent speaker at trade shows, seminars and user groups etc., Sarah is Compuware's European spokesperson on all quality, risk management, project management and testing issues. Sarah is also involved in Compuware's business development and partner liaison within the European software quality market and is a key influencer to Compuware's product management division regarding the future direction of our software quality solutions.

Prior to Compuware, Sarah held various pre-sales, consultancy and database/programming positions at Sequel Ltd (latterly Platinum Technology), Safeway plc and Dixons plc. Sarah holds a Diploma in Management from the Institute of Leadership & Management.

## Featured Speaker:

### **Rick Craig, SQE Inc.**

**Author: "Systematic Software Testing"**

### **Workshop: Using Orthogonal Arrays and All Pairs in Test Design**



Abstract:

Trying to determine what to test is always a challenge, but in some systems the sheer number of possible combinations of parameters seems overwhelming. For example, if a system had 13 different parameters and each one could take on 3 different values, the number of combinations would exceed 1.5 million. Using a technique like all pairs it is possible to test all pairwise input combinations with just 15 test cases!! All pairs and orthogonal arrays are very useful tools when used under the right set of circumstances. Like all techniques, they involve compromise and a certain degree of risk. In this tutorial you will learn the basics of orthogonal arrays and all pairs and learn when these techniques may (or may not) be useful. All attendees will do a manual mapping of a problem on to an orthogonal array.

Learn:

- How orthogonal arrays can greatly reduce the number of tests required
- Why these techniques may cause black box testers to overtest.....
- Why it is important to understand the dependencies of the parameters
- Why it is important to conduct risk analysis and/or user profiles in addition to orthogonal arrays/all pairs
- Whether it matters if you choose orthogonal arrays or all pairs?
- How to use one of the available free tools.

Biography:

Rick Craig is a frequent speaker at testing conferences and is well-received worldwide as a test and evaluation instructor. He has implemented and managed testing efforts on large-scale, traditional, and embedded systems, and co-authored a study that benchmarked industry-wide processes. Rick is also co-author of *Systematic Software Testing*.

## **Paul Gerrard, Gerrard Consulting**

### **Author: "Risk-Based E-Business Testing" Workshop: Web Technology for Testers; testing beneath the GUI**



Abstract:

- How do the web and web pages work?
- How can free tools be used to test websites?
- When can I start testing my own website with tools?

Most system and acceptance testing of web and internet applications is still done manually. The test automation tools that do exist are all GUI-based, proprietary and expensive. GUI test tools are incredibly sophisticated and usually require programming skills to operate. Most of the complexity is required to deal with the vagaries of the GUI, not the essential tests themselves.

Testing Frameworks are emerging as the required 'front-end- to' test execution tools. But what is happening here? The complexity of the GUI is managed by two test tools and the browser. We aren't testing those, are we?

For the purpose of most functional testing is to execute transactions on the web server and supporting infrastructure. The browser is just a means of presenting a usable interface to a human being.

What if we separated our tests into those which require the user interface, and those which do not? The tests that must use the user interface can be run manually or using a proprietary tool. We can use free tools to test under the GUI. These tools are much faster, simpler and easier to use than GUI test tools.

Bring a laptop and use a real tool to test on our portable wireless networked environment. Bring a laptop with wireless capability or a 5m standard network cable.

Biography:

Paul is the founder and Principal of Gerrard Consulting, a services company focused on increasing the success rate of IT-based projects for clients. He > has conducted assignments in all aspects of Software Testing and Quality Assurance. Previously, he has worked as a developer, designer, project manager and consultant for small and large developments using all major technologies and is the webmaster of gerrardconsulting.com and several other websites.

He has degrees from the Universities of Oxford and London, is Web Secretary for the BCS SIG in Software Testing (SIGIST), Founding Chair of the ISEB Tester Qualification Board and the host/organiser of the UK Test Management Forum conferences. He is a regular speaker at seminars and conferences in the UK, continental Europe and the USA and was recently awarded the "Best Presentation of the Year" prize by the BCS SIGIST.

Paul has written many papers and articles, most of which are published on the web. With Neil Thompson, wrote "Risk-Based E-Business Testing" – the standard text for risk-based testing. He is a regular keynote speaker and tutorial presenter and has presented over 200 talks at conferences in the UK, continental Europe, USA and Australia since 1993. He is also a coach for Maidenhead Rowing club.

## Lee Clifford, Virgin Mobile

### **MINI-TRACK:** Improving 'Test Quality' through 'Quality Stubs'

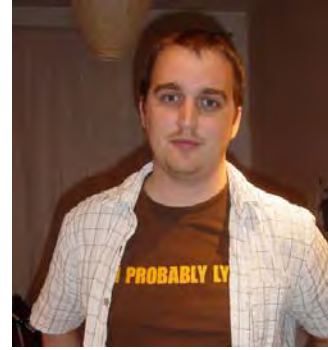
Abstract:

This presentation will explain how the Virgin Mobile Test Team re-wrote the existing static stubs used within our Test and Development Environments. The dynamic capabilities of the new stubs created has enabled more meaningful testing, both at Component level and when testing systems integration. This has resulted in increased systems coverage, and has also reduced the dependencies on connectivity to 3<sup>rd</sup> Party test systems.

The main aim of this presentation is to provide some generic ideas regarding best use of stubs, analysis of current use and future stub development. However, by describing how the VM Test Team developed their new stubs, it will also provide some detail regarding the use of the Green Hat Tester tool used for all Stub Development.

Biography:

Lee Clifford is the Senior Technical Tester from Virgin Mobile in the UK. He has worked in Software testing for over 3 years, with Virgin Mobile UK but has also spent some time with the Virgin Mobile USA Test Team in San Francisco. Lee was the key driver and developer of the 'VM Test Team Stub rework' project.



## Nitesh Shah, CapGemini

### **MINI-TRACK:** Test Automation Model

Abstract:

In his presentation Nitesh will describe the Automation Model (N Model) and how this fits with the traditional V-Model approach. The N model has a number of extra stages relating to test automation:

- Test Automation Strategy
- Test Automation Architecture
- High Level Plan
- Low level design
- Scripting.

Nitesh will briefly explain the purpose of each of these documents and their associated importance in the development lifecycle and how they can be used to improve your test automation regime.

Biography:

Nitesh Shah is the Test Automation Leader on E Delivery for the Aspire project within CapGemini. As a Test Automation leader he provides technical expertise generating automation strategies, architecture documents and low level plans. He has also designed and developed automation infrastructures and scripts. He has worked in testing for over 6 years for a variety of different organisations and industries.





## Featured Speaker:

### **Rick Craig, SQE Inc.**

#### **Author: "Systematic Software Testing" Testing as a Measurement Activity**

##### Abstract:

Testing is a measurement activity. Testers and test managers must collect metrics on the quality of the software under test and of the testing process itself. Collecting, analyzing, and using metrics can be difficult and the problem is further complicated because many developers and testers feel that the metrics will be used "against them". No metric is perfect and all metrics suffer from inconsistencies in collection and interpretation. Rick will address some common metrics such as measures of product quality, defect removal efficiency, defect density, defect arrival rate, testing status, when to stop testing and the benefits and pitfalls of each. He will provide some measurement guidelines, rules of thumb, and tips on how to avoid "metrics dysfunction".



##### Learn:

- The relationship of testing to development
- Some common and uncommon metrics
- What measures can do for you
- Measurement pitfalls and problems
- Measurement guidelines
- How to avoid measurement dysfunction

##### Biography:

Rick Craig is a frequent speaker at testing conferences and is well-received worldwide as a test and evaluation instructor. He has implemented and managed testing efforts on large-scale, traditional, and embedded systems, and co-authored a study that benchmarked industry-wide processes. Rick is also co-author of *Systematic Software Testing*.

## Featured Speaker:

### ***Neil Mullarkey, Improv. Improvisation***

#### Abstract:

Neil will introduce improvisational theatre skills and to consider how they could be used in the business world where communication, leadership and innovation are much in demand. There will be a range of exercises - individually, in pairs and altogether. The workshop is highly interactive and great fun, but participants will begin to understand more about their own thought processes and begin to embrace new approaches as well as their own and others' creativity.

#### Biography:

Neil Mullarkey is a comedian. He co-founded Britain's top improv troupe, THE COMEDY STORE PLAYERS in 1985 and continues to improvise with them twice a week at London's famous Comedy Store, alongside Paul Merton and Josie Lawrence. His many credits include Austin Powers movies (International Man of Mystery and Goldmember), Whose Line Is it Anyway, I'm Sorry I Haven't A Clue, Just a Minute, and the recent series of QI. He runs workshops using improv techniques and has worked with many organisations including the BBC, KPMG, Saatchi & Saatchi, the NHS, The LloydsTSB, Unilever, Vodafone and BP. For more information about Neil and his workshops, visit [www.Improvyourbiz.com](http://www.Improvyourbiz.com).



## ARTICLE: GONZO QA: FEAR AND LOATHING IN THE MORNING

*By Martin Cunnington*

I arrive at work somewhat frazzled. I had set off an hour early in order to storm up a mountain of work before the mist had lifted but events have conspired to make me arrive at my desk 30 minutes later than usual. I log on to my PC knowing there will be a 15 minute delay between entering my username and password and seeing the desktop, due to network policies, virus scans and Windows patches. Every day this happens. Note to self: raise this to IT Support once more. Two hundred people at an average of £50 per billable hour waiting 15 minutes is, ooh, £2,500 per day lost income, around £625,000 per year. I'd commit to fixing that for half the price, cash in hand, no questions asked. Fortunately my diary is paper-based and the café around the corner is staffed by Croatians with a vigorous approach to coffee-making. As I wait, I plan my day with a vengeance, knowing that whoever booked me yesterday for work due today will get priority over anyone trying to book me today for work due yesterday. I kill the next 14 minutes by taking a walk round the building. There's no one in. Reasons will vary from "my goldfish was poorly in the night" through "my social worker wouldn't bail me" to "I forgot". "I was working till midnight" there's no argument with. Commitment to the work, each other, the client is all.

I'm in. Start Outlook, MSN Messenger (isn't the new Windows Live Messenger limp?), Yahoo! and Gmail. Much Spam has arrived overnight, so I take a quick look at what's getting through; is there anything new or clever of which we should take note? No. There never is, but you never know; inspiration comes from the strangest places. Much of the bulk-mail we sent out last night has come through as well, so I take a quick look at that while I'm on. It's far too late to correct anything that has gone out, but it's never too late to spot mistakes and learn. Here's an unusually long newsletter from one of our clients that, yes! It has the body copy repeated twice. Fortunately it's not one of ours. Ha! It's tempting to dump on the other agency that sent it but discretion is the better part of valour. Unless you're pitching for new business, of course.

My schedule has changed overnight. There's a bunch of unlikely looking assignments which would be better handled by others in my team. Click, click and the reassignments are complete. I like this new scheduling system. It was an internal and thus non-billable project, so we didn't have time to test it. The Technical Director says this means it has no bugs. I like the Technical Director. There's some new business, so obviously that's top priority. Then there's a task due yesterday which wasn't ready to test but now is, so that's obviously top priority. Then there's an email about a project I have never heard of which was due last week, so that's obviously top priority. And here's an escalation to bring a task due tomorrow forward to today. It's marked top priority. By virtue of the power vested in me, I reject the whole lot. This clears my schedule to allow me to concentrate on the future. Now that's top priority. Anything I can do now to reduce the risk of projects going pear-shaped in the future is worth the effort, assuming of course that the projects are real, the risks are high, mitigation factors are available, the costs billable and the clients are up for it. Assuming quite a lot of things, actually. Maybe I should talk to the project managers before I go on.

I have half an hour before a project status meeting, so I will just check these banners. What can go wrong with Flash banners? The concept is story-boarded, the copy written, checked and signed-off. The height and width are specified, along with the number of rotations, the maximum file size and the version of Flash required. The requirement for GIF back-ups is stated and so is the type of click-through code. Into the banner test harness we go (another internal and thus bug-free application). Hmm. The brief I am reading did not reach the person who did this work, at least not all of it, or possibly this release. Not only that, but not all of the banner is clickable and the background is transparent and not solid. I've not seen that before. Note to self: add to list of cruel and unusual bugs. An hour later I have missed the project status meeting discussing banner production with the person responsible. It turns out she means well but is new and hasn't been inducted. It is difficult to get medieval on someone who should have been trained to know better but hasn't. I'll reserve that for the Creative Director. Talking of whom, here he is now, dazed smile on face, award in hand, last night's formal dress jacket draped over shoulder. We like awards. Especially big shiny heavy ones we can use to prop open doors when the air-conditioning is jammed on hot all summer (e.g. this summer). We spend 27 minutes congratulating each other on another outstanding agency success and I almost forget what I came here to do. We spend three minutes discussing Flash banner production. It turns out that all we had to do was to localise some banners developed in the USA to suit the UK market. The issues I described were present in the supplied originals and are therefore

not ours to fix. We succumb to 'Not Invented Here Syndrome' for a few moments and then decide to fix the banners regardless. Note to self: We must define and agree minimum specifications for assets we will accept from third-parties. This will make it easier to identify faulty asset deliveries and have them sent back for rework at their cost, not ours.

I walk back to my desk and check my watch. Is it lunch-time yet? I need to speak to HR about developing an induction programme for new hires. We traditionally drop people in at the deep end. Sink or swim! Only the fittest survive! It's the agency way! But the turnover of new staff this year is unsustainable. It's time to stop filling these peoples' pockets with lead. It's time to start offering them brightly-coloured buoyancy devices like rubber rings, arm bands and swim noodles. On brand, of course.

To be continued? Let Pam Frederiksen know if you want to read part II of this article. Let Pam know if you do \*not\* want to read part II of this article. Vote early, vote often!

### **About the author**



Martin Cunnington is Head of Quality Assurance at MRM Worldwide, a leading digital marketing agency servicing some of the world's bluest of blue chip companies. Martin joined MRM Worldwide (then Zentropy Partners) in 2000 from HP (then Compaq) after 10 years Marketing IT service in Munich (then München), Germany. His influences include John von Neumann, Donald Knuth, Philip Kotler, Geoff Quentin, James Whittaker, Johanna Rothman and Hunter S Thompson.

From the Editor (and the net):

Gonzo journalism is a style of reporting that mixes fiction and factual journalism. It uses an unconventional, exaggerated and highly subjective style, often including the reporter as part of the story. It is used to describe the style of American journalist Hunter S. Thompson, among others.

# ARTICLE: HORSES FOR COURSES

*By Kevin Buchta*

---

## **Is it always appropriate to employ a test specialist on a project?**

On many systems projects, testing is carried out by staff with different job roles, having skill sets and disciplines that do not include system or software testing. For a professional test specialist this may appear to be flawed approach to the way that the project team is built and managed, however from a business manager's viewpoint this may be perfectly reasonable. A project team structure and size is governed by many factors, because projects often differ in scale, complexity, timescale and business needs. The organisation's lifecycle and culture govern the approach to the management of projects. Test managers and test specialists will have observed examples of this, and should be confident in promoting reasoned arguments about the appropriate level of testing that should be applied.

These arguments need to be put forward within the context of the organisation and project environment, and this background is considered next.

## **Organisational Lifecycle**

### ***Start-up Company***

There are many examples of large companies that evolved from a business idea that was implemented by two or three innovators using few resources. They spotted a business opportunity, developed and produced a product, tested it and sold this to a customer, without the support of a project team. The successful innovators then identified further opportunities and the customer base grew. One of the more well known ones was Hewlett-Packard, when the two founders produced their first product in a garage. I have previously worked for several companies with similar origins, and have noticed patterns in the way that development and testing has evolved with the growth of the organisation. Some of these patterns are described next.

### ***Initial company expansion.***

The company expansion results in hiring additional staff, and development and sales staff are often among the first staff hired. With further sales, multiple project streams are created. Projects may vary in size and complexity and the systems functionality sometimes diverge to meet individual customer needs. As the company grows further, separate teams are formed with responsibility for supporting different customers. Developers in each project team work directly with customers to identify requirements, produce the product and test/demonstrate that it meets these requirements.

There were incomplete records describing the delivered functionality of the initial product. In some cases the customers understand the functionality better than the development staff. New projects are based on the initial product, and further development starts to affect the functionality that customers wished to have retained. The newly formed development team may not fully understand the core functionality, and they occasionally rely on customers to perform their testing for them. Customers identify defects in the core functionality that the developers then need to rectify within the existing project timeframe. The late discovery of problems by the customer, impact the schedule and cost. Tensions within the team increase as developers have less time to retest, because of the additional unplanned effort in reworking and re-releasing the system.

The first test specialist is recruited so that:

- Developers can concentrate on their primary task – developing code and delivering products.
- Business risk will be reduced by testers identifying problems early enough for developers to fix, and before customers get visibility of them.
- Systematic regression testing of the core functionality is carried out.
- Managers can receive independent status reports regarding test progress.

### ***Established company***

The company may now have established separate business units addressing different types of markets. Testing may be applied with varying levels of rigour according to the type of system, and in

some cases its life expectancy. Quality and development standards may be established for certain business areas.

Development silos have been discouraged within business units, and testing across projects provides benefits through common tools usage, domain knowledge of the test specialists and re-usable test artefacts. Interoperability testing is applied to test system interfaces across multiple systems. Automated regression testing is introduced, improving test efficiency and reducing test cycle times.

### **Regeneration**

Changing market conditions and poor cash flow will affect the organisation, and attempts will be made to cultivate an environment that encourages staff to innovate and return to the core business. In this climate, roles and responsibilities are reviewed, often with reduction in roles considered non essential. There will be an attempt to return to the approach taken during the start-up and early expansion of the company for some groups, and developers once again may be expected to perform system testing. Test specialists with good domain knowledge and an understanding of the customers systems and business processes will be well placed to deliver the business benefits that testing can bring to the organisation.

### **Providing Business Benefits**

What are the benefits of testing? Professional testers understand this, however most managers will understand this as one means of risk reduction, and to demonstrate that the system meets the customer requirements.

Each manager however is likely to have different expectations from testing e.g.:

- The senior management team will use test reports as one source of management information to measure progress.
- A project manager will have to deliver business benefit to the customer and reduce risk while balancing cost, schedule and quality.
- The development manager will see testing as a way that the development team have checked their output before delivery to a customer.
- Business managers will view it as key to the business users' acceptance of the product.
- A test manager will view all test phases within the project, but extending to other projects.

The test manager and test specialist need to recognise the viewpoints of these key stakeholders on a project, particularly when the following question is raised:

What are the benefits of having test specialists perform testing when developers or business users could carry out the tests themselves?

The most obvious answer is this is the test specialist's job role, and that the division of labour, where specialists concentrate on specific areas is a well established technique to improve productivity. A trained and experienced test specialist will be better placed than most other specialists to deliver business benefits through the following areas:

- Maintaining business resilience.
- Early validation and verification.
- Risk management.
- Maintaining quality.
- Applying test techniques utilising industry standards.
- Preparation and execution of manual and automated test suites.
- Test tool familiarity, and newly emerging toolsets.
- Timely status reporting to management of the quality of the deliverables.

<b>Stage in the Organisational Lifecycle</b>	<b>Developer</b>	<b>Test specialist</b>	<b>Business User</b>	<b>Summary</b>
Start-up – First single product with a short life expectancy	Performs unit test, and then the functional tests together with Business users.	Not engaged.	Performs initial functional test and then the final acceptance test.	Business user has early visibility of problems and this may impact the credibility of development team. Lack of IS test records is not likely to be a problem with the short life expectancy of the product. Future regression test pack is not required for a non strategic product that will not be developed further.
Startup – Strategic product that will form the basis for the company expansion.	Performs unit test, and then the functional tests together with Business users.	Engaged to establish test artefacts – tools and test re-use pack. Test team takes responsibility for test tools, scripts, records and defect management.	Performs UAT tests on specific Business transactions.	Business user has early visibility of problems and this may impact the credibility of development team. Records of delivered working functionality exist.
Startup –Application interfaces to other established subsystems.	Performs unit and functional test of the application.	Reviews the functional specification and identifies system interface issues. Uses system level knowledge to check that the application works with other systems. Regression test pack produced for future use.	Performs UAT tests on specific Business transactions. Access to tools and test artefacts that the test specialists make available.	Test specialists review the requirements and identify potential interface problems early. Test cases are produced and reviewed by developers and business users.
Initial company expansion.	Performs unit tests and few functional tests on the application. Code management across projects is introduced, but faults introduced in the core functionality may impact multiple projects.	Use test artefacts to perform regression tests of core functionality across multiple projects. Manage defects across projects. Review functional specifications for testability; identify early problems before coding starts, reducing the number of defects delivered into system test. Produce test cases that are reviewed and agreed with developers and customers. Perform system tests.	Performs UAT tests on specific Business transactions. Access to tools and test artefacts that the test specialists make available	Developers focused more on coding and unit tests. Less time wasted on rework because less defects are delivered after test specialists review requirements and have test cases signed off. Test specialists responsible for running regression tests, and reporting on test successes across projects. Fewer defects delivered to the Business User team for UAT testing.
Established company	As above	As above. Automated testing implemented to reduce test effort	As above	As above, and test cycle times reduced as regression test effort is automated.
Regeneration	Performs unit test, and then the functional tests together with Business users.	Test specialists preserve the test artefacts, and technical systems knowledge base for the support of existing customers. Regression testing is a stable point of reference to the management team.	Performs initial functional test and then the final acceptance test.	The test artefacts take on increased value as a definitive record of the systems capability. Test specialists with good systems and business process knowledge are retained to support the existing customer base.

Developers and test specialists may have identical technical training and qualifications but choose to follow different career paths. Some test specialists may

Developers and test specialists may have identical technical training and qualifications but choose to follow different career paths. Some test specialists may also have become test specialists after an early career within a business team focusing on business processes. A test team with this mix of technical and business skills can provide a holistic view of the delivered system using their collective systems and business process knowledge.

## **Comparison of testing provided by developers, testing specialists, business users**

As described earlier in this article, arguments regarding testing need to be considered within the context of the organisation and the project environment. An organisational lifecycle was described, together with some of the patterns noticed across several companies.

In the table on the previous page, the stages in the lifecycle are shown, with the test activities compared between for developers, test specialist and business users. Some of the benefits of including a test specialist at each stage are identified.

Although this is not a comprehensive list, the exercise illustrates that each of the three types of specialist have a role to play during testing. The emphasis will change dependent on the project type and organisational maturity.

When promoting an argument about which role delivers the best benefit, then the business benefit should be considered first, and then the appropriate balance applied to testing between the developer, test specialist and business user.

## **Summary**

It is almost always appropriate to consult a test specialist in the early stages of a project to develop a test strategy in consultation with the management team.

Some of the benefits in employing a test specialist, rather than relying only on developers and business users to perform testing are:-

- By consulting a test specialist for each project, even for a newly formed company, then the risk of project failure could be reduced by reducing waste and rework.
- Test specialists tend to focus on traceability to requirements, and through an early review of technical specifications they can identify unclear and ambiguous requirements before they are converted into code. They will help to minimise the number of defects delivered, and can reduce the loading on a developer even before they have prepared a test case or executed any test.
- Production of a re-usable set of regression tests and test records for a new product will help to minimise disputes between the project team and the business users regarding the core functionality on subsequent versions of the product.
- Test tool expertise, can be applied and made available across project teams, reducing costs by economy of scale, and producing standardised management reports.
- Test script and defects management can be applied across projects.
- Automated testing can be introduced to improve test coverage of regression tests with shorter test cycles. A number of industry standard tools exist that can support this approach.
- By looking outside the individual project, focusing on business benefits, understanding business processes, and taking a holistic approach to systems, test specialists will be well placed to add business value while developing their skill sets further.

Test managers and test specialists should be aware when presenting arguments about the benefits of employing a test specialist on a project, that there isn't always a clear benefit perceived by the management team. The viewpoints of different managers, the organisational maturity, project type, complexity, project team size and structure all need to be considered when identifying the benefits.

**Kevin Buchta**, [kevin.buchta@dsl.pipex.com](mailto:kevin.buchta@dsl.pipex.com)