Issue 21

THE TESTER

September 2007

NEXT CONFERENCE

**Tuesday
18 September 2007**

Joined-Up Testing

- **Extreme Testing: A Software Tester's Lessons Learned from Extreme Programmers**
- **Implementing a Test Automation Centre**
- **Lessons Learnt Implementing DDP**
- **Web Testing under the bonnet**
- **Adapting to Agile**
- **Testing: It's in the Game**
- **Joined up testing: A Rightshore™ case study**
- **Test Automation: The Next Generation?**

IN THIS ISSUE:

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

FROM THE EDITOR

We're pleased to welcome Steve Allott and Mike Hendry to the committee following the AGM at the June Conference.

Steve has taken over the role of Programme Secretary, having already undertaken this role a few years ago – so a glutton for punishment! However, I'm sure you will make the job a lot easier for him by volunteering to present a paper at a future conference. If so, do contact him at stephen@electromind.com

Mike is now Secretary and we look forward to an input of new ideas for forthcoming events.

The Library is now being looked after by Sue Atkins (many thanks for volunteering!) and Sue can be contacted at SigLib@iotest.com

Don't forget that the EuroSTAR Conference will take place in Stockholm this year at the beginning of December. Peter Morgan is going and he wants to persuade you to go too – see later in this newsletter. However, this still leaves time for you to attend the December SIGiST one as well!

In the mean-time, please make sure you are booked in early for the SIGiST conference on Tuesday 18 September, especially for any chosen parallel sessions!

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

BCS SIGIST website: www.sigist.org.uk

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

SIGIST UML Testers Forum: www.umltesters.org

Future SIGiST conference dates

13 December 2007

BOOKING INSTRUCTIONS

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

FAX TO:	***OR POST TO:***
Colin Chivers 01793 417444	Colin Chivers Specialist Groups Officer First Floor, Block D North Star House North Star Avenue Swindon SN2 1FA

EUROSTAR

Stockholm is the place to be

The 15th EuroSTAR conference takes place in Stockholm in early December 2007. This is a return to the Swedish capital, with the European software testing conference usually located in one of the countries that support it well. Nevertheless, a ‘home’ venue will increase the number of delegates from that country.

Put the dates in your diary now; the first 1½ days are tutorials, with the main conference running from 14:00 on Tuesday 4th December → 17:00 on Thursday 6th December. After this, there is a Gala Dinner, where the testing community honours its own. This last event is a popular closing to the conference, and tends to sell out some 2 months before the start of the conference.

For me, EuroSTAR is a wonderful opportunity to meet old and new testing friends, and get some good perspectives upon testing challenges that we all face, but outside of the pressure cooker atmosphere of where I am working at the time (deadlines, poor quality code delivered, and scantily clad requirements documents).

Look at the EuroSTAR web-site http://www.qualtechconferences.com/content.asp?id=2 when you are able, and aim to get people from your organisation there. You will see that the UK is well represented amongst the speakers (17 out of 50 speaking slots, and 3 out of 14 tutorials). There are also details of accommodation, and the pricing structure of the conference fees. On this last point, there are reduced rates for early bookings (by 28th September), and a 10% reduction for membership of some testing organisations (of which SIGiST is one). Special rates are available for group bookings.

At the June 2007 SIGiST meeting, I was fortunate enough to win the draw for a free EuroSTAR conference place. Having benefited greatly from attending these conferences in recent years, I want to encourage you to go. Then, before you go, plan what sessions you will attend, and go with an open, blank notebook. Many of my written entries are completed outside of the ‘official’ conference session, and I have built up testing friendships; I do not just stay around the people I know.

I hope to see you in Stockholm in early December – it will do us both good.

Peter Morgan, freelance tester, Nicemove Ltd (morganp@supanet.com)

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 Sue Atkins on 01697 748 748 or email her at siglib@iotest.com. Alternatively, download the book loan form on the SIGIST website www.SIGiST.org.uk. Happy Reading!

PROGRAMME COMMENTARY: JOINED-UP TESTING

Stephen Allott, Programme Secretary

Joined up testing is the theme for the September conference and I thought it worthwhile to add a few words to try and explain my thinking. In my humble opinion, far too many people seem to claim to have a magic wand to fix testing or quality problems with software. Also, there are many testers out there that I meet who are looking for the “silver bullet” or “one size fits all” answer to their problems that will make their lives easier. Well, those of us working on testing projects in the real world know that testing is a complex and challenging task and that the solutions come from a variety of sources and in many different shapes and sizes.

So today’s test manager has a lot to consider. Communicating between all stakeholders in the project using the right level of information, creating end to end tests based on realistic scenarios, choosing a flexible and appropriate development and testing model, involving the business in testing, introducing automated tools at the right time, balancing the resources onshore and offshore. A joined up approach to the problem seems to me to be a step in the right direction and I trust you’ll find the September conference exciting and rewarding with many ideas to take away and implement in your own organisations.

NEXT MEETING – PROGRAMME

BCS SIGIST – Joined-Up Testing

Tuesday 18 September 2007
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** Extreme Testing: A Software Tester's Lessons Learned from Extreme Programmers Elisabeth Hendrickson, Quality Tree Consulting		
10:30	**Networking session and commercial break**		
10:45	Coffee & opportunity to visit the exhibition		
11:15	Implementing a Test Automation Centre Victoria Pearson, *BT Plc*	Intermediate Workshop (bring a wireless enabled laptop) Web Testing under the bonnet *Paul Gerrard* *Gerrard Consulting*	**Featured Speaker** Advanced Workshop Adapting to Agile *Elisabeth Hendrickson* *Quality Tree Software Inc.*
12:00	Lessons Learnt Implementing DDP Richard Durham, Citrix		
12:45	Lunch & opportunity to visit the exhibition		
13:45	Testing: It's in the Game *Chris Ambler, Electronic Arts*	Intermediate Workshop Web Testing under the bonnet ***Part 2***	**Featured Speaker** Advanced Workshop Adapting to Agile ***Part 2***
14:30	Joined up testing: A Rightshore™ case study Peter Hanson, Capgemini UK		
15:15	Tea & opportunity to visit the exhibition		
15:45	**Live and Unscripted** *Stephen Allott talks to Paul Gerrard about current projects*		
16:00	**Featured Speaker** **Test Automation: The Next Generation** Elisabeth Hendrickson, Quality Tree Software Inc.		
17:00	Closing Remarks		

Workshops MUST be booked in advanced, as places are limited

The SiGiST committee reserves the right to amend the programme if circumstances deem it necessary. Workshops will have limited places, to avoid disappointment try to book in advance.

ABSTRACTS AND BIOGRAPHIES

Featured Speaker:

Elizabeth Hendrikson, Quality Tree Software Inc.

Elisabeth Hendrickson founded her company as Quality Tree Software Inc. in 1997 to provide training and consulting in software quality and testing. She incorporated the company as Quality Tree Software, Inc. in 1998.

Elisabeth began working in the software industry in 1984. She has held positions as a Tester, Programmer, Test Automation Manager, Quality Engineering Director, and Technical Writer working for companies ranging from a 20-person startup to a large multi-national software vendor.

Elisabeth is an experienced facilitator and trainer. A student of Jerry Weinberg's, Elisabeth is a graduate of the Weinberg & Weinberg PSL, ChangeShop, and SEM programs. She also studied Experiential Training Design with Jerry and his wife Dani.

Elisabeth is frequently invited to speak at conferences around the world. She has given keynote addresses at conferences in the US, Sweden, Portugal, and Australia.

In 2003, Elisabeth became involved with the Agile community. In 2005 she became a Certified Scrum Master and in 2006 she joined the board of directors for the Agile Alliance.

These days Elisabeth splits her time between teaching, speaking, writing, and working on Extreme Programming teams with test-infected programmers who value her obsession with testing.

Abstract: Extreme Testing: A Software Tester's Lessons Learned from Extreme Programmers

Extreme Programming (XP) teams are test infected. They practice Test Driven Development (TDD), writing an executable unit test before writing the code to be tested. Many also practice Acceptance Test Driven Development (ATDD), writing executable acceptance tests before implementing a feature. They use Continuous Integration (CI) to give them rapid feedback about the effects of changes. They practice pair programming, a technique that results in all code being peer reviewed before it’s checked in. In short, XP teams test continuously from the very first moment of any given project. You could even call them Test Obsessed. That explains why Elisabeth Hendrickson, author of www.testobsessed.com, likes XP teams so much. As a professional tester, Elisabeth has spent the last several years on a quest to discover how testers can contribute effectively on Extreme Programming projects. In this talk, Elisabeth shares her experiences as a tester and programmer (yes, programmer) on XP teams, and the sometimes surprising lessons working on XP teams has taught her about effective software testing.

Abstract: Test Automation, the Next Generation

Development tools have become orders of magnitudes more powerful in the last several years with intellisense; keyword coloring; automated refactoring across entire code bases; tight integration with xUnit-style unit testing frameworks; and tight integration with source control repositories. While we’ve seen huge leaps in development tools, tools to support functional testing haven’t kept pace.

The biggest steps forward in test automation include the idea of Domain Specific Languages, and frameworks like FIT and Fitnesse that break down barriers between developers and testers or subject matter experts. And yet there is still a long way to go, and we’re overdue for a major step forward in functional testing tools.

Several people have begun prototyping test automation solutions that could hold the keys to a giant leap forward. In this talk, Elisabeth Hendrickson explains what's missing in the current generation of test automation solutions while providing an inside look at what's next.

Abstract: Adapting to Agile

When a software development team adopts an Agile process such as Scrum or XP, QA team members often find that their traditional practices no longer fit the new context. Extensive up front test planning and design, heavyweight test documentation, and formal entrance and exit criteria all serve a traditional context well, but tend to get in the way in an Agile environment.

In this workshop, participants experience a transition to Agile in a paper-based simulation (no programming required). In a series of iterations, the team attempts to deliver a product that the customer is willing to buy. Each successful delivery generates revenue for the company. But as with real projects, producing a working product on a tight schedule can be challenging.

After each iteration, participants reflect on key events, then adjust their team practices to increase their productivity for the next iteration. As a result, participants learn to apply the principles of visibility, feedback, communication, and collaboration to increase their rate of delivery. By the end of the workshop, participants will have a visceral understanding of Agile, and in particular the shifting role of Test/QA in Agile development.

Richard Durham

Richard has been testing software professionally for over a decade in a variety of different industries. Richard has presented at both EuroStar and StarEast on Agile testing practices but also maintains a strong interest in testing metrics and the use of small scale test automation with model based testing. Richard is currently employed at Citrix.

Abstract: Lessons Learnt Implementing DDP

Defect Detection Percentage has been described as one of the most important testing metrics. In theory it is a fairly simple metric – what percentage of the total number of defects in a release were found internally (which be extension tells you what percentage was found by the end users/customers. In this presentation find out what happened when DDP is put into practice in a large software company.

Victoria Pearson, Head of Testing at BT

Biography not yet available.

Abstract: Implementing a Test Automation Centre

BT Group is remaking itself into a global leader in the market for networked IT services in part by undertaking the largest and most ambitious network transformation project in the telecom industry. BT's £10 billion 21st Century Network (21CN) programme will create an integrated voice and data network that will drive a new wave of converged products and services while drastically reducing network operations costs. To get there, BT is adopting new techniques and strategies for system and product development, emphasizing the need to reduce project cycle times while simultaneously increasing the percentage of things "done right the first time." A key component of this effort is end-to-end testing and, in particular, a massive automated testing
initiative.

While in the early stages of test planning for the 21CN programme, it became apparent that the only way to meet the proposed timetables was through the aggressive use of test automation. BT determined that its limited pool of experienced testers would be better used in defining test requirements and designing test cases rather than working on automation. BT then collaborated with an Indian partner to establish a cost-effective, offshore team focused entirely on automation. In effect, BT decided to industrialise the production of test scripts in this newly created Test Automation Centre (TAC).

BT tests each new system and process both from the operations and management perspective and the customer perspective. This process is achieved using a team of about 130 employees producing automated scripts, running them and reporting on the results.
The TAC also employs approximately 150 manual testers which includes test managers, test designers, test environment support people and actual testers.

In order to make the whole process work effectively and provide the benefits that justified the cost, BT had to consider many other factors. One of the early challenges was to identify tests that should not be automated due to technical difficulties that would make automation too costly. BT considered factors such as the frequency of test runs, lifespan of tests, cost of automation vs. the cost of manual testing and cost of script maintenance.

In addition to the time saving benefits of automating the testing process, BT also found additional benefits such as identifying errors and omissions in the tests and identifying bugs in the applications that had not been spotted by manual testers.
Additionally, BT has employed simple automated scripts to provide ongoing hourly error testing to confirm that all the major components in its very complex test environments are up and running.
Lastly, BT has used automated scripts to cleanse data on a regular basis.

Paul Gerrard

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.

Abstract: Web Testing Under the Bonnet

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. You will need a laptop with wireless capability or a 5m standard network cable or be prepared to share with a colleague.

Peter Hanson, Capgemini UK

Peter is a Senior Test Manager at Capgemini, a global leader in consulting, technology, outsourcing, and local professional services. Headquartered in Paris and operating in more than 30 countries, Capgemini has approximately 75,000 employees.

Peter's role at Capgemini includes consulting, strategy and project delivery work for clients, and the development of capability within the organisation's community of testing professionals. He has broad-ranging experience in the software industry: as a business manager he worked with the financial, telco, aerospace and defence industries, founded in an initial career as a systems designer and developer.

Abstract: Joined-up testing – a Rightshore^{TM} case study

This presentation describes testing in a joint onshore/offshore project which is currently underway for a client. With a "perfect storm" of fixed price, a fixed – and very visible – delivery date and rigorous quality requirements, a joined-up testing approach is the only possible solution.

The system is being developed for an independent non-governmental regulatory body, and will enable its customers to report against a clear, consolidated reporting schedule. It will provide the flexibility to amend the reporting required as new legislation takes effect or the scope of regulation changes; it must also handle expansion in the numbers and types of organisations within the regulatory remit.

An iterative methodology is being used to shape the project requirements, design, development and testing. The application is also being developed using a Rightshore strategy to maximise the project's cost-effectiveness, with some 80 per cent of its 60-strong project team being based at an Application Development Centre in Mumbai, India.

The operating approach for Rightshore uses global delivery to place processes, services and functions in the best location, blended and coordinated to meet specific business goals:

- on site, at the client's offices - for example for business requirements analysis and UAT
- on-shore - delivering specialist methodology and application architect services in Woking
- off-shore - using large-scale development and testing services in Mumbai to provide capacity, capabilities and competencies at reduced costs - without compromising product and service quality.

So the challenge for project testing is to join up processes across

- geographies and time zones
- project disciplines, and
- client and third-party stakeholders

…. all set against the background of an iterative project in a fast-changing environment.

The presentation discusses the testing approach taken, the lessons learned so far – and the plans for the future of testing on the project.

ARTICLES

The Exploratory Tester’s Logbook

James Lyndsay, Workroom Productions

A particular phrase has rung through my life as an experimenter. I can remember the day I first heard it, and it's followed me round ever since. Let me set the scene. It's Double Physics. I'm sitting in a classroom - not our usual one, with the high lab desks and the ticker-tape timers, but a smaller one. One where a precious video recorder can be connected to a jittery television without interference from crocodile clips and galvanometers.

On screen, a succession of experiments. Bits of metallic stuff are being dropped into dishes of water. Before each experiment, we're shown chemical symbols, the periodic table. The voice from the screen says 'Write it down'. We do. We're shown the weight of the stuff, its colour, the ambient temperature, the volume of the dish, the air pressure. Each time, the voice from the screen tells us to 'Write it down'. We do, navy-blue hardback lab notebooks balanced on knees and the tops of seatbacks. We write it all down. The stuff drops into the water, nothing happens. We write it down. The camera zooms in; nothing. We write it down. Another experiment, more stuff. Still nothing happens. Still we write it down.

And so we get used to the nothing. When the metal looks odd after a moment or two's submergence, we write it down. When a sheen of tiny bubbles gradually creeps over it, we write that down too. When the first, single bubble escapes its hold and rushes to the surface of the water, we write it down.

The next hunk of junk promptly surrounds itself with a silver sheath of gas, and as it bobbles on the bottom of the bowl, we realise we really should have used a stopwatch. It’s the 80s – so the geekier and richer among use are using their digital watches. Another experiment fizzes like an aspirin, the next positively leaps about. We're shown more metal - it's yellow grey and skinned with oil - the voice tells us it is Cadmium. We write it down. There's barely a moment after it hits the surface of the water, and the bowl explodes. Water pours from the shards, soaking the black cloth that covers the studio table. Steam rises. We write it down.

Oddly enough, I've no real idea if that last bit of stuff was cadmium. The choice of subject seems odd, now I think of it, for a Physics lesson. Perhaps it was Double Chemistry - but I've always hated Chemistry. I'm twenty-five years older, and writing it down hasn't helped me retain many of those facts at this distance. The lesson itself, however, has followed me round, whispering 'write it down' in labs and libraries, concerts and car journeys, wrapping my fingers around a pen or pushing them over a keyboard even as my eyelids drop and I'm called in to bed. I've used notepads, jotters, exercise books, Moleskins, dictaphones tape and digital, Palm Pilots, laptops. Everywhere I've gone, everything I've done, I've written it down.

- = + = -

My life, then, is filled with scraps of rubbish paper, tapes, files in obscure formats. It'll come as no surprise that I've been writing notes throughout my time as a tester. For me, recording what I do is fundamental to doing it. I believe that I do a better job, just because I'm making notes. My mind is clearer, my concentration better, my decisions more justified – and sometimes, more surprising. It's a pain to find that I'm halfway through something, and I've lost my notes. It's worse to find I've not been making any – because although I do it more often than not, writing stuff down is hardly my default behaviour.

Last year, I made a loaf of sourdough bread every week for six months; not a note to be seen. I started to make notes - had to find paper, made less bread, got wet flour in the laptop - but the bread got better. Way better. Why did I kid myself by not bothering?

- = + = -

Sometimes, I teach people to test - and sometimes, I teach the systems beneath the peculiar magic that is exploratory testing. I teach – and advocate – *session-based* exploratory testing, and making a reasonable log of what is done in a session seems to be a particular problem for my student explorers.

I've coached good testers, who show me three lines of post-test scribble to describe ninety minutes of exploration. I've worked with interested and well-informed teams, with only a buglog to show for their efforts. I've had a class full of people look at their pretty session templates, and write not one thing - not a bug, not a plan, not a target for testing, not their name or a date in the labelled boxes at the top of the paper. It strikes me, at these points, that perhaps I'm getting something wrong in my teaching.

- = + = -

I guess the most immediate thing I want to explain when I see an empty test log is *why* one might want to keep good notes. Better still, *how* keeping good notes can help. Lets do that, just to tick some of them off:

- Writing notes is one way of clarifying ones thoughts, one way of identifying imprecision and hidden assumptions.
- Keeping to the discipline of writing notes helps you recognise when you're being inappropriately distracted by an interesting problem – and lets you make that problem to let you come back
- With notes to consult, you are more able to make decisions based on the information, not on habit or expectation
- You don't have to remember everything – if you write it down you can move on with a clear mind
- The notes are a mnemonic; you'll remember more when you come back
- You can show your notes to someone else, any time you like
- You can demonstrate that you've actually been working
- Notes don't decay over time

Perhaps there are a bunch of 'why not's. If you loaded a noteless tester with truth serum and asked away (not something I've yet tried - but that serum's tricky stuff to administer in a classroom), what might they say?

- I don't have time
- Note-taking disrupts my testing karma
- I don't want to get caught doing a bad job
- Doing the work is more interesting than keeping notes
- I don't think it helps, so I'm not going to try, even in this classroom, even after you've pleaded with me to give it a go, even after you've told me my every effort is worthless without the backup that notes provide

Perhaps it's the means of recording that causes problems. I use paper and a pencil, which may be a hangover from my schooldays, but has the advantage of continuing to work when the computer stops. When I need something more sophisticated and searchable, I use outlining software and attach my machine to a camera or to the test machine so I can infiltrate my outline with pictures and files. I don't think that the way you make a log is terribly important, so long as it doesn't get in the way of making a good record, and lets you take in a chunk of testing at a glance. For the record, I think that notepad is a dreadful tool for making a worthwhile log.

Perhaps it would help to show the key that I've arrived at over the years that seems to do a good job of helping me pick out information from my test logs; Here are markers that I put at the start of important lines.

-	A new thought or action.
*	A more important thought or action - sometimes used for 'return to this'
!	One you'll want to remember at the end of the test. Sometimes reserved for bugs.
[	An aside - a thought or observation that needs to go down, but that isn't in the flow]
¿	Something I'm not sure of - may need more tests. A question for *me*.
?	A question for someone, or something

Plenty of arrows and circles - not forgetting diagrams, underlining, tables etc.

However, I have another idea. I think that some testers might not have had that voice in their head for most of their adult life, telling them to Write It Down. Some testers just haven't been Writing It Down. Indeed, it is possible, I believe rather patronisingly, that some testers really aren't altogether sure *What* to Write Down.

Me? I carried on with Physics until I graduated. Then I segued neatly into testing, which I've done for years, too. I never really knew what to write down, but I wrote it down anyway. A quarter of century of notes. If you're a compulsive writer-down of unconsidered trifles, I suggest you need read no further. On the other hand, if you'd like a shortcut to my personal take on the Secret Stuff that should be Written Down, I crave your attention for a few paragraphs more.

- = + = -

If you make a plan for your session, write it down. If you're just tootling along all planless, you need a strategy, an approach. A sticky note will do. There are no excuses – accept no substitutes.

You'll want to remember the actions you take, the data you use, your expectations, your observations - including the time. Don't necessarily limit yourself to exactly what you're testing - you're working in some kind of context. You'll get better at this over time; there's an instinct that comes with practice that lets you separate the wheat from the chaff. There's always going to be a bit of chaff.

Keep track of things that repeat. Even if nothing happens. Dullness is a virtue in most working systems. And without track of dullness, how will you notice . .

Surprises. Is that a goat among the sheep? If you didn't expect it, it's worth writing down. If someone else wouldn't expect it, it's a bug. Perhaps you've seen an exploitation. Have you a hypothesis? Are you making a model? And when you've supported your hypothesis, found a potential bug, had a surprise, or the dullness is just too much to bear, you need to . . . .

Make a Decision - many people get so used to testing by instinct, or by the book, that they don't notice they're making decisions. Worse, they've no idea what the decisions might have been. Scripted testing can be decisionless, but decisions are *key* to exploration. When you decide to take a different approach, to try different data, or just to consciously do exactly the same thing again, but watching more closely this time, you're taking a decision. Make a quick note.

- = + = -

A lot to keep track of? Sure – but that's *why* you write it down. You *can't* keep track of all this stuff without a bit of paper by your side, Superman. Just as integrated test design has strange and positive effects on the quality of your code, integrated note-taking works wonders on your testing - and on your thinking.

A bare minimum? I always have a spare moment for a bare minimum. For me; strategy, data, surprises, decisions. For you, something else. Keep notes, and you'll be there in no time. It's not hard, it's not dull, but it needs a little persistence, a little focus, a little discipline. So; if this article has triggered one new idea, a decision, an observation, a single spark of intent or insight, I urge you to stop reading, right now, take pen and paper and . . .

Write it down.

- = + = -

James Lyndsay is a test strategist.

See www.workroom-productions.com

Gonzo QA IV: Mistakes, I've made a few

By Martin Cunnington

My biggest mistake was to delete the UK master invoice file of the major chemical company I worked for at the time, shortly before going home for the evening. I had been promoted to Database Administrator (DBA) a few weeks before and I was carrying out routine house-keeping activities – or so I thought. It turned out that my recently-departed predecessor had not been naming database objects logically, had not been carrying out routine house-keeping activities and further, the database management system (DBMS) was quite capable of deleting files in use without warning or protest. The result of all this was I went home unaware. Further, the overnight batch file which wrote the days invoices to the master file and then deleted itself wrote the days invoices to null and then deleted itself. When I came in the next day, Accounts Receivable staff had just been told that as well as inputting the day's invoices, they would have to re-input the previous day's too, essentially doing two days work in one day and nobody was to go home until it was done. They gave me the cold shoulder, the Finance Director gave me an earful which included the full cost of my error rounded to the nearest five thousand pounds and the IT Director sent two of his people down to give me a kicking on his behalf. The Senior Systems Programmer beat me up himself; he always was a hands-on kind of guy. The previous night's dump had been restored in my absence but it turned out that transaction logging had never been enabled, so roll-forward until a few minutes before I had accidentally deleted the file was not possible. At the time I did not know that you could run a DBMS without transaction logging enabled. My response of "how about that?" was not appreciated by sysadmin staff at all.

This incident was my first true insight into the importance of recognising, assessing and managing risk, whether or not it is part of your job description.

I failed to recognise the risks I was accepting during the hand-over from my predecessor. I had assumed that he had been approaching his work logically and by the book. He had not. He had an idiosyncratic approach which worked for him but, rather spectacularly, did not work for me. I had assumed that the DBMS contained a series of checks and balances which would aid me in my work. It did not. It did what it was told immediately and without question. I had assumed that transaction logging was turned on. It was not. An investigation the previous year had concluded that transaction logging required too many valuable CPU cycles and too much valuable disk space, i.e. it was too costly to implement when compared to the potential return.

Having made a whole series of false assumptions, I had failed to identify and assess the risks inherent in my new job. What was I required to do that might be risky? How was I going to do it in a way more likely to succeed than fail? What would be the impact of failure? What would I do if things failed? What could others do if things failed in my absence? What alerts would signal failure and who would receive them?

Having failed to recognise and assess the risks, there can be no surprise that I was not managing risk in a meaningful fashion. All of which changed after this incident, of course. Everyone involved now recognised that a risk existed. Senior management recognised that they might have been complacent in vetoing the cost of transaction logging. The revenue lost by my error was approximately twice the cost of installing and running transaction logging for a year. I was encouraged to buy and build a suite of tools to minimise the risk of human error when carrying out my tasks. I was also given permission to build a test installation; now I didn't have to do everything straight on to live, I could safely practice somewhere else first.

Although this story is twenty years old, it is still relevant today. I work in a fast-paced, high-pressure environment and it is not unusual for a project to go through a complete change of personnel on both the agency and client side as it races from a bright idea to a shiny finished product. A new team member, especially one subbing for another, will typically assume that all is well with the project so far, and want to build their contribution on top of the sound foundations built by their predecessors. I encourage staff to recognise that this is not necessarily so and to assess the situation in as much detail as possible, given the circumstances. This then allows them to consider mitigating, eliminating or otherwise insuring against the major risks they have identified. Should the only reasonable course of action turn out to be to tolerate the risk, then at least this is done in an informed way rather than by default (which is what I did). I consider this to be part of on-going quality planning. As a software quality assurance professional, I have seen many deep issues identified during quality control that can be traced back to new project team members unknowingly building their structures on sand. What a waste.

For further articles on how I apply lessons learned from my mistakes in the past to my current position, please visit www.participationmarketing.co.uk

About the author

Martin Cunnington is Head of Quality Assurance at MRM Worldwide, a leading digital marketing agency servicing the world's bluest of blue chip companies. A Chartered IT Professional, his influences include Alan Turing, Mark Rothko, Isambard Kingdom Brunel and Hunter S Thompson. Martin is currently LinkedIn and battling a facebook addiction.

P.S. Anyone who thinks that hand-over is only a risky business in the world of IT should speak to members of a 400 metre relay team.