

# OZTester



The Quarterly Magazine for the Australian Software Testing Community and Supporters

---

## ***In this issue:***

ISSUE 1 APR - JUN 2013

FREE

Interview with Robert Abbey of Cognizant  
Why Do Test Managers Go On Stress Leave?  
Why Do IT Projects Fail?  
CITCON Sydney 2013  
TestWest Conference  
Usability Debt  
...and more!



# OZTester Magazine

Editor: **Colin Cherry**  
[colinda@ozemail.com.au](mailto:colinda@ozemail.com.au)

Ph 0412 214 240

Editor-In-Chief: **Geoff Horne**  
[ed@nztester.co.nz](mailto:ed@nztester.co.nz)  
[geoffh@isga.com](mailto:geoffh@isga.com)

Ph. 021 634 900

P O Box 48-018  
Blockhouse Bay  
Auckland 0600  
New Zealand

[www.nztester.co.nz/oztester.html](http://www.nztester.co.nz/oztester.html)

Advertising Enquiries: [ed@nztester.co.nz](mailto:ed@nztester.co.nz)

## **Disclaimer:**

Articles and advertisements contained in OZTester are published in good faith and although provided by people who are experts in their fields, OZTester make no guarantees or representations of any kind concerning the accuracy or suitability of the information contained within or the suitability of products and services advertised for any and all specific applications and uses. All such information is provided "as is" and with specific disclaimer of any warranties of merchantability, fitness for purpose, title and/or non-infringement. The opinions and writings of all authors and contributors to OZTester are merely an expression of the author's own thoughts, knowledge or information that they have gathered for publication. OZTester does not endorse such authors, necessarily agree with opinions and views expressed nor represents that writings are accurate or suitable for any purpose whatsoever. As a reader of this magazine you disclaim and hold OZTester, its employees and agents and Geoff Horne, its owner, editor-in-chief and publisher and Colin Cherry as editor harmless of all content contained in this magazine as to its warranty of merchantability, fitness for purpose, title and/or non-infringement.

No part of this magazine may be reproduced in whole or in part without the express written permission of the publisher .

© Copyright 2013 - OZTester Magazine, all rights reserved.

# The Journal For Australian Test Professionals

Welcome to the first quarterly issue of a brand new venture for the Australian Software Testing community - OZTester. I am deeply honoured to be appointed as the inaugural Editor of this auspicious venture. OZTester is a "sister" magazine to NZTester, launched late last year in New Zealand. NZTester has been such a success that there have been a large number of requests to launch a similar magazine here in Australia and so, here we are!!

For me personally, this is a very exciting time, as I have the opportunity to return to one of my early passions - that of writing and editing a magazine. My previous roles in writing and editing were on a slightly smaller scale and used much older technologies but the creative process was no different. I have a burning passion for software testing and I have a great desire to share this great profession with as many people as possible; therefore having the opportunity to Edit this magazine fulfils both these needs.

So, what will the magazine look like and what does it set out to achieve? Firstly, it is predominantly an online medium; however, we will publish hard copies for specific events, such as a major conference. Secondly, this is a magazine "for the people, by the people" and therefore we will be providing local software testing professionals with the opportunity to enlighten us with their particular take on software testing in Australia. Thirdly, we aim to provide the human side of our profession with a focus on real-world case studies and experiences. Fourthly, we aim to be relevant and creative and therefore we will strive to stay current with the latest developments and initiatives around the Australian software testing landscape.

Finally, I'd like to encourage you to provide feedback and constructive criticism regarding our new venture, in order that we may strive for that "Quality Goal" that we so often want to see within the software we test.

Remember, if you're a happy tester, you probably haven't seen that latest software build yet...

Colin Cherry  
Editor, OZTester

## IN THIS ISSUE...

*Click on title*

- 4** Interview With Robert Abbey,  
Head of Testing, Australia &  
New Zealand, Cognizant
  
- 10** Why Do IT Projects Fail?  
Geoff Horne, OZTester Magazine
  
- 14** Why Do Test Managers Go on  
Stress Leave?  
Colin Cherry, OZTester Magazine
  
- 18** Usability Debt  
David Greenlees, Bendigo &  
Adelaide Bank
  
- 23** Mobile Applications Testing  
Reghunath Balaraman, Infosys
  
- 26** CITCON Sydney 2013  
Oliver Erlewein, Independent  
Consultant
  
- 28** TestWest Conference  
Toby Thompson, Disqover and  
Colin Cherry, OZTester Magazine

### Regular Features

- 17** Test Events
- 21** From the TestToolBox
- 31** Great Bugs We Have Known  
and Loved



This issue's interviewee is:

# Robert Abbey

Head of Quality Engineering  
& Assurance Services,  
Australia and New Zealand,  
Cognizant Technology Solutions

*Our guest interview this issue is with Robert Abbey of Cognizant. He has also worked for First Data Corp. and Datanonix amongst others - Ed.*

## Can you please describe Cognizant's testing capabilities?

Cognizant's Quality Engineering and Assurance services (CQEAS) is one of the world's largest quality assurance practices, with over 23,500 testing professionals distributed across the globe and serving the quality needs of more than 450 clients from across industry domains and geographies. Over 75 percent of the code tested by Cognizant is provided by clients or third-party vendors.

Cognizant, since its inception in Australia in 2007, has served more than 25 clients and currently has more than 150 resources servicing customers in Australia and New Zealand.

Cognizant's Quality Engineering and Assurance practice is aligned to major industries that include Banking and Financial services, Healthcare, Manufacturing / Retail, and Telecommunication, Media and Entertainment.

In addition to that, Cognizant has internal Centres of Excellence focusing on specialised needs of our customers – Automation Centre of Excellence, Performance Architecture, Consulting & Engineering (PACE), Quality Engineering Consulting, Service Oriented Architecture Centre of Excellence, Mobile Testing Centre of Excellence, Data Warehouse and Business Intelligence Testing Centre of Excellence are a few that help provide solutions catering to the specialised needs of our customers.

## What testing services does Cognizant offer?

We have refined and categorised our current service lines to address our clients' current QA challenges whilst preparing them for the future. These categories include:

- **Assurance Services** – Cognizant brings in a combination of domain and testing expertise to enable systems to meet their functional requirements. Services in this category enable clients to assess business readiness of applications by validating functional & non-functional aspects of applications.
- **Specialised Services** – These include a spectrum of services and techniques that complement QA efforts across the development lifecycle. Integrated automation to achieve accelerated testing and application maintenance support services such as Test Data Management are also included in this.
- **Infrastructure Services** – Cognizant's QE&A practice works closely with IT Infrastructure Services and Global Technology Office to build and offer services that can help clients reduce spend around QA infrastructure (Tools and Environments).
- **Consulting Services** – One of our marquee offerings which help clients in creating a long-term QA roadmap or develop detailed process improvements. Our proven methodologies, best practices and structured



evaluation frameworks have helped us deliver through consulting engagements a testing capability fit for business.

- **Engineering Services & Mobility** –These are service lines like Quality Engineering (shift-left services), Performance Engineering, Mobile Enablement, Devices Testing and Product Certifications that will help our clients enhance their business reach and improve end-user experience.

### What do you believe makes Cognizant different?

The aspects that make Cognizant unique, compelling and different are:

- **Customer Focus:** Cognizant has always strived hard to keep the customers' interests in its focus. Our goal has always been to help the customer to get across the line, in terms of their objectives and targets, without trying to oversell our capabilities.
- **People before Profit:** Cognizant has made sure that it has the best and the brightest of the talent with entrepreneurial thought to be part of its teams. Plus, the welfare and the career growth of our people has always been a higher priority than making profit from the business.
- **One-stop Partner for Testing:** Cognizant understands that the problems faced by the customers are varied and the solutions need to be nuanced and comprehensive. Our offerings cover all the tenable bases of QA, and hence, we are in a position to offer an all-encompassing solution & thought leadership to our customers.
- **Innovation:** To complement our efforts in making QA more sophisticated, we have invested in building IPs that can potentially transform the way QA services are demanded and consumed today. We have built IPs in the areas of Test Data Automation, Mobile Testing, Service Virtualisation and QA analytics for Intelligent Reporting, which allow us to provide a futuristic solution to the customers.
- **Local Leadership:** Despite being a global company, Cognizant has always believed in

having local presence to service its customers in the region. Local leadership has allowed us to understand the cultural needs of the region better and provide Thought Leadership and solutions that fulfill the client expectations.

### What do you think makes a Test Manager or Analyst come to work for Cognizant?

Cognizant is one of the fastest growing companies globally, and is also arguably the largest provider of Independent Verification & Validation services in the world.

We have invested significantly in people and believe that we are the pioneers in clear career-tracking for Testing professionals in the industry. Our recruiting, training and certification programs, employee engagement activities and community events help motivate our QA professionals.

Cognizant is an employer of choice for the campus recruits. At Cognizant, we recruit a mix of engineering, science and commerce graduates from the various universities around the globe, business analysts from premier B-schools of the world and also hire testing professionals from the regions where we operate from as laterals.

We have defined three definite career paths for our QA professionals to choose from. These include the project management track (where a tester progresses to become a test manager, handling delivery), the domain track (where a tester specialises in a specific industry), and the test architect track. A resource is provided with the opportunity to choose his / her career path, based on the skill set and the area of interest.

Our employees in Cognizant are provided with the opportunity to work across the globe in cutting-edge technologies. Cognizant has selected and invested in a pool of industry experts focusing on our Testing Centres of Excellence organised by technology and industries to develop innovative tools and reusable artefacts for quicker knowledge acquisition and to address the time-to-market pressure. We make sure that performance and excellence are recognised and well rewarded both monetarily and through career growth.

## Where do you believe the challenges for Australian testing companies lay?

The Australian testing companies have been proponents of technology & quality product / output. This has led to the organisations facing the following challenges in the market today.

- **Faster time to market / reduction of cycle time:** We see technologies growing more complex, making it harder for large companies to keep up with time to market expectations. Companies are now spending longer testing their products, resulting in higher cycle times. On the other hand, smaller companies lack skill sets and this also leads to an increased time to market
- **Lack of skilled resources:** The Australian market uses high-end technologies, and hence, is finding it difficult to get the appropriate resources with the right skill set.
- **Knowledge retention:** Most companies use contractors for their engagements and this leads to lack of knowledge retention once the contractors are rolled off from their engagements.
- **Large companies are finding it difficult to move to agile:** Most of the companies in Australia are running projects in a Waterfall model. To cater to the changing requirements today, there is a widespread increase in agile adoption amongst organisations. However, organisations are facing difficulties, due to lack of knowledge and niche skills in managing agile projects.
- **Minimal implementation of Automation & Risk-based testing:** The companies haven't adopted RBT and Automation in the past. With the adoption of tools and investment in niche skills, the automation is picking up, whilst the risk-based testing helps to meet the demands on time to market.

## Where do you believe Australia's approach to testing is going well?

As I have covered in the previous question, Australia has been one of the early adopters of technology.

- There is clear evidence in the Australian market that focus is on new and emerging technologies to generate productivity efficiencies, especially in the areas of Mobile Application, Device usage, Usage of Cloud etc.
- Australia also leads in technical areas like TOSCA, use of Selenium and Testing as a Service (TaaS)
- Australian testing practitioners have the intuitive knowledge on testing, and hence, know how to achieve high quality products
- The plethora of Test Summits, Testing Conferences and Special Interest Groups in this region, have helped the community to share their collective wisdom and knowhow on testing.

## Where do you believe Australia's approach could improve?

Though Australia has been one of the early adopters of technology, which is competitive with a strong services sector, there are a few areas, wherein, there is room for improvement:

- One of the areas that could be improved is the skill set of the testing resources. Companies should take steps to update the skill set of their resources based on requirement and adaptation of technology in their firm. Training of the resources should be made mandatory on technical, domain and soft skills.
- Organisations should plan for job rotation, which will not only help to bring in new people with newer ideas into the projects / programmes, but also help resources with an opportunity to obtain newer skill set and improve their knowledge horizon
- Steps should be taken for better adaptation of newer approaches like agile
- Capacity of organisations and size of projects demand adoption of On-Demand Software and Services which would help reduce the CAPEX in organisations

and Services which would help reduce the CAPEX in organisations

- There is a lack of centralisation of testing activities in organisations. Therefore, there seems to be a lack of ownership in relation to the testing activities which could lead to underperformance and suboptimal development
- The testing process ought to get smarter with a better balance between Quality, Cost and Time to market

### **Do you believe that overall the standard of testing in Australia is improving?**

At the moment, there is a significant move towards centralisation of testing. Solutions are more focused on driving cost optimisation and reducing time to market without compromising on Quality. Test Automation & Performance Engineering are being given due importance in the QA lifecycle of projects. Organisations have started considering and using on-demand services. Companies are testing their products on mobile platforms and across devices. Plus Australia has got an ever increasing pool of certified testing professionals. These moves in the industry show that standards of testing in Australia are truly improving.

### **Where do you believe the next initiatives in testing lay? What's coming next? In Australia? Internationally?**

Today, applications are becoming more global. This means that there is a growing emphasis on QA departments to assist local users to adapt these solutions for their individual businesses. Common challenges include the integration of disparate platforms, and enhancing the "user experience" of their customer facing applications. QA today is helping businesses discover the pockets of "value gaps" which they can act upon to increase productivity, optimise effort and reduce cost.

In addition, globally, dynamic market conditions and intense competition are pushing organisations to be more aggressive in launching products, enhance user experience while operating at reduced costs. We are witnessing a trend towards IT organisations increasingly consuming specialised (non-traditional) services. These include integrated automation, management of testing tools on cloud,

cloud-based TEMS, Test data management, Performance Engineering, Mobile Testing, Devices Testing and QA analytics. We are witnessing these consumption levels being largely influenced by the trends revolving around SMAC (Social, Mobility, Analytics and Cloud).

Adoption of lean models to avoid waste in the testing process, scientific methods for estimation and design, risk-based approaches to testing, model-based testing are largely enabling QA teams to assure quality at faster rates and at optimised costs. We also witness QA teams adopting virtualisation techniques and cloud to accelerate testing cycles. Companies are adopting new-age approaches such as agile for software development and testing. Evaluating parameters including utilisation and service consumption levels and demand patterns, IT teams can effectively leverage cloud techniques to render services on a shared and need-based model. Cloud Testing and Virtualisation in our view will help orchestrate the way testing services are consumed and delivered today.

### **Do you have a testing horror story to share?**

I wish I could, unfortunately I don't have one I can share, I'm afraid.

*Many thanks Robert, we very much appreciate the time and effort - Ed.*

# COGNIZANT QUALITY ENGINEERING & ASSURANCE SUMMIT SYDNEY 2013

THURSDAY, APRIL 11

SHERATON ON THE PARK, SYDNEY



## TRANSFORMING QA: ENABLING QUALITY ENGINEERING & ASSURANCE

An exclusive, *invitation-only* knowledge-sharing forum for IT leaders, subject matter experts and key industry authorities in the software testing and QA space, the Cognizant Quality Engineering & Assurance Summit creates a platform to mobilise organisations with proven strategies for transforming QA and the impact of this transformation on your business.

For more information on the Summit, please contact Robert Abbey at [RobertAbbey@cognizant.com](mailto:RobertAbbey@cognizant.com).

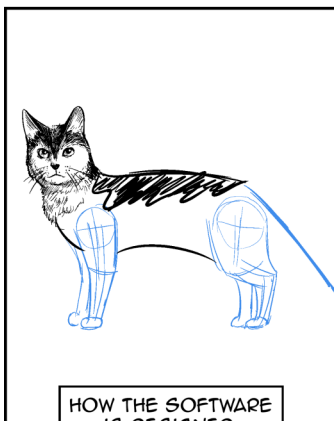
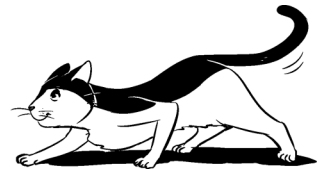
[www.cognizant.com](http://www.cognizant.com)



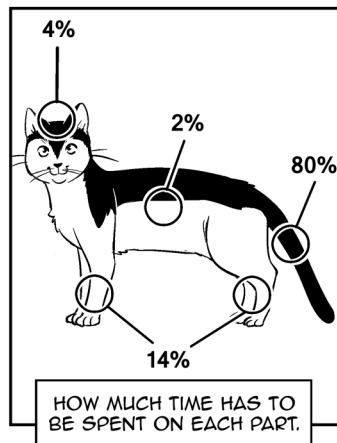


# The Very Latest Software Development Infographic

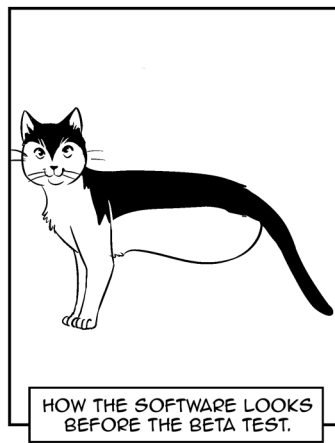
(bye bye swings and trees)



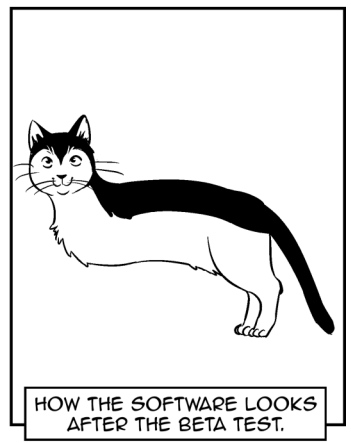
HOW THE SOFTWARE IS DESIGNED.



HOW MUCH TIME HAS TO BE SPENT ON EACH PART.



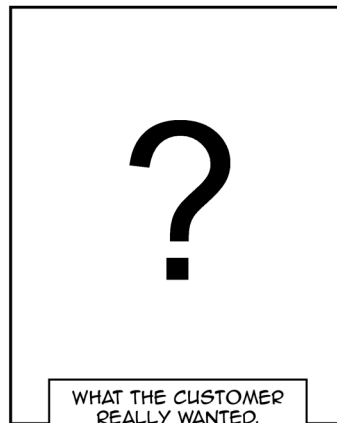
HOW THE SOFTWARE LOOKS BEFORE THE BETA TEST.



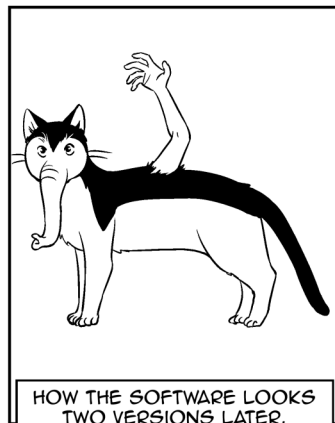
HOW THE SOFTWARE LOOKS AFTER THE BETA TEST.



HOW THE SOFTWARE IS ADVERTISED.



WHAT THE CUSTOMER REALLY WANTED.



HOW THE SOFTWARE LOOKS TWO VERSIONS LATER.

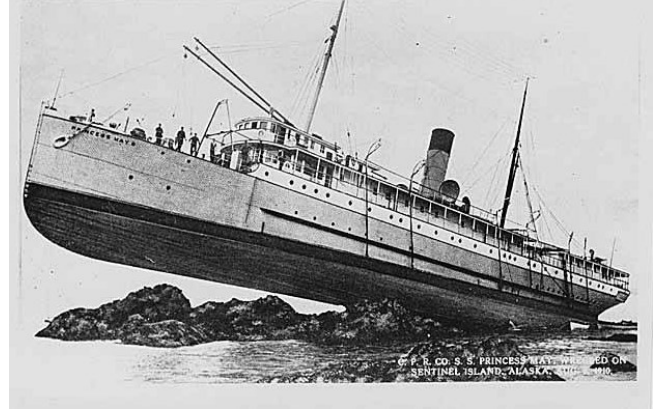


*Reproduced with the kind permission of Sandra & Woo ([www.sandraandwoo.com](http://www.sandraandwoo.com))*

# Why Do IT Projects Fail?

by **Geoff Horne**

The first in a series entitled Testing Practice vs Commercial Reality.



*With the current coverage of the New Zealand Ministry of Education's Novopay project looming large in our news media, I thought I would kick the series off with this piece on why IT projects are prone to failure, especially the larger ones.*

Of course there can be many reasons and most are subjective however I believe there are but a few predominant root causes that spawn multiple symptoms and consequences later in the project.

Sometimes projects fail through circumstances beyond anyone's control. My intent here is to focus on the reasons for failure of those that need not have. As larger projects and programmes of work are often very complex, the consequences can be so much more impacting and difficult to recover from.

First off, let's define what is meant by failure. For the purposes of this discussion I'll include any project that is terminated, delivered significantly late, over budget and/or so severely compromised so as not to deliver on pre-project suppositions.

As a test manager who has been around the block more than a few times and engaged on a number of occasions to turn supposedly troubled testing efforts around, I've observed first-hand what the outcomes can be when the "rubber hits the road". In most cases, I can quite clearly see what the issues are and their causes. And guess what....they're not usually anything to do with testing!

Based on this experience and avoiding generics such as poor communication, lack of collaboration, bad management, absence of governance etc., the most common root causes of project failures I believe can be summed up in the following:

## **It Is Not Properly Understood What Is Needed In a System.**

Regardless of development methodology, unless it is understood what is required of an IT system it is very difficult to determine how to best design, build, test and implement it.

I have always found that the more information that can be provided around what a system needs to do, the more precise the design, build and delivery processes can be. However the exercise to gather, analyse and document requirements can be very time consuming. If the timetable is challenged, it is very tempting to dilute the effort required to adequately address this part of the project lifecycle in order to save time, effort and money.

I've been on projects where requirements have been well understood and articulated (albeit precious few unfortunately) to other end of the scale where they are identified progressively as the system is built. That is not to say that anything but the former is impossible to do however the risk of not meeting expectations increase exponentially the further along the dilution scale we travel.

I have also found that the less known, the greater the likelihood of changes arising from within what is known. Whilst a certain amount of change is to be reasonably expected and accommodated, those resulting from a dilution of effort are really not desirable.

## **System Capability is Over-Estimated.**

While I'd like to think that the days of the gun-toting, silver-tongued IT sales representative have in the main gone, there are often occasions where certain assumptions are made by all concerned, around exactly what a system is capable of. Some of the more common situations I've come across are:

- The system is over-sold as neither vendor nor client personnel properly understand the business issues that the system needs to address.
- There is a temptation to design and build the most leading edge system using the very latest technology whilst deploying the absolute best of breed tools and methods - with little regard for the tried, true and proven.

- Even though a system may be based upon packaged software, it is expected that it will be able to withstand wholesale modification and still behave as per a standard package.
- It is assumed that the system will work or work the same way for one client as it does for others of the same ilk.
- The system is promoted as being so flexible that it can be configured to do just about anything without realising that “anything” still requires definition.
- Configuring systems is sometimes still considered easy. Many are now configured via coding mechanisms as opposed parameter “switches” and this should really be treated as a development exercise rather than configuration.
- Scoping and assessment can be too focused on the highest priorities without proper verification of a system’s ability to meet lower priority yet still essential requirements (eg. just give us what we’ve already got on a more scalable platform – real live example!)

The end result of these and others examples of over-estimation is usually one of inappropriate expectations being set with the client, with the vendor(s) saddled with an exercise that may be high impossible to perform professionally. Not to mention that both will probably end up seriously out-of-pocket.

### **Over-Zealousness to Keep to Delivery Commitments**

Once expectations have been set around what the system will do and when it will be deployed, it is highly undesirable and perhaps even career or contract-limiting to have to go back to request additional resources, whether it be people, budget, tools, time or whatever. It becomes even more so when dealing with larger projects and programmes of work where the most senior executives are involved.

In project-land much is at stake: reputations, egos, credibility, career advancement, contract renewals and in many cases compensation packages,

especially at management levels.

Maximising the opportunities for success is perceived as best achieved by protecting the pre-set expectations with as much vigour as possible. The further away from these the project becomes then the more the opportunities are seen to fall under threat.

When things don’t go according to plan, the need arises to push for resolution quickly and get the project back on track as fast as possible, and fair enough. However what often happens, is that only the symptoms of the issues are uncovered and remedied, due to the need for speed. The root cause(s) remain hidden and untouched to strike again via another symptom, another day, then maybe another and another and.....

I’ve often wondered whether expectation-protection mania would change if management performance measurements were expanded to include not only budgets and deadlines but also on how well the system is meeting business requirements after 3-6 months. Hmm....

### **Inherent Assumptions That Technology Will Actually Work (Or At Least Can Be Made To)**

A good example that springs to mind is an ERP implementation I was on a few years back. The core ERP application was surrounded by a raft of new and legacy sub-systems that required interfacing to and from the core, as well as to the outside world via the web.

Cutting a long story short, one of the new sub-systems utilised a technology that it turned out could not be made to accommodate the interfacing requirements without considerable compromise. So the system went live without one of the key functions that it was cost-justified upon. It was rectified nine months or so later.

### **Once a System Has Been Built and the Defects Fixed, it is Assumed to be Ready For Implementation**

The prescribed test completion criteria have been met eg. all high and medium priority tests have passed and no severity 1 and 2 defects are outstanding. Therefore the system is ready to go? Not necessarily.

A couple of false confidences are possible here; it is assumed that 1) testing has covered off all that is required of a system and 2) all the issues have been found and fixed.

Too often decisions around system readiness are based on status and progress snapshots without consideration of the trends that emerged during development and testing. For example, if lots of high priority defects are found (and fixed) in the last week of testing, it's a pretty good bet that if the system goes live the following week, more will be found.

### **Failure to Acknowledge the Facts**

Unfortunately no matter how glaringly obvious it may be that something is amiss, the facts and supporting evidence can be still very hard to swallow, especially when so much time, energy and money has been spent in getting a project to the point it's at.

"Flogging the dead horse (or even a half alive one)" is unfortunately still all too common. Despite the most genuine of intent, reality can sometimes be that it makes more sense from both a project and business perspective to bite the bullet and address the root causes of the issues once and for all. While nearly always painful, as is commonly quoted; no pain = no gain.

The alternative is usually a "slug it out" exercise that is ready for implementation only when it is!

### **The Client Bites Off More Than it Can Chew**

Business pressures can lead to not-well-thought-out decisions and subsequently it's those at the coalface who often end up wearing the consequences. In the case of larger projects, I've known companies to reject multiple or multi-staged projects in favour of one large exercise as the latter is perceived as requiring less time, money and business disruption.

However the ability of the business to accommodate a large systems implementation *and* maintain operational business levels is often not properly assessed and the end result can be far more costly, time-consuming and disruptive than it need be.

I have to say that I am a fan of the gently-gently approach: keep-it-simple to start, prove as many assumptions as possible first, get some runs on the board then ramp up.

### **Summary**

As a result of these and other challenges, more often than not the reason that tough decisions are never made to address sooner, is because no-one is game enough to stand up and call the spade a spade, whether it be at coalface-to-management, management-to-executive or executive-to-owner/sponsor.

As we know, there have been many high-visibility project "disasters". I wonder how many of them could have been avoided had 1) a few courageous souls spoken up early enough to have made a difference and 2) decision-making managers and executives had listened to them.

If you've not already read the Planit Index article I suggest you do and if you have, read it again as it presents interesting findings on the causes of Australasian project failures.

**Geoff Horne is the editor of NZTester Magazine and a Senior Test Manager. He can be contacted at [ed@nztester.co.nz](mailto:ed@nztester.co.nz)**





# TAKE THE NEXT STEP.



SOFTWARE TESTING /  
ISTQB CERTIFICATION  
TRAINING / SPECIALISTS

[DISQOVER.COM.AU](http://DISQOVER.COM.AU)

P: +61 3 9836 2901  
E: [INFO@DISQOVER.COM.AU](mailto:INFO@DISQOVER.COM.AU)





## Why Do Test Managers Go On Stress Leave?

by Colin Cherry

Being passionate about my job has got me into more hot water than a cooked lobster and I still have the marks to prove it. When I first began my quest to understand software testing (and ultimately Quality Management) I took the approach that I had always taken with other goals in my life - I embraced everything and everyone and fully immersed myself, until I felt that I understood the very essence. The result, initially, was pretty messy. But I did come out the other side with greater perspective and harmony - although some of you who have worked closely with me may think I'm being over-optimistic here.

In my experience, software testing can be a challenging (almost thankless) task with many *battles* fought in the search for a perceived *quality nirvana*. I have worked on over twenty programs and projects during my career, some very large and some quite small and they all have several characteristics in common - they have deadlines, they have budgets, they have goals and they have personalities. This last one doesn't appear in any textbook, but it does occur in real life. Just like people, programs of work and projects have personalities. Some are peaceful; some are fiery; some are happy and some are painful and difficult. This is the world we enter when we take on

the role of Test Manager, have no doubts about that.

I have worked on a program of work where we engaged over a hundred Test Managers and their Test teams (located on multiple continents). Gaining agreement on how and what to test was obviously a significant challenge. Not just because of the scale of the work but also because of the multiple organisations and agendas that we had to work with. I was responsible for developing the overall Test Strategy for this program of work and therefore I had a unique insight into what was to be done and why. Sadly, this program had at least one fatality (due to a heart attack, not due to my Test Strategy) and several stress-related incidents where staff left after breakdowns, never to return. This is why I think in terms of a *quality battle* - because, that's the dance I've been part of for most of the past 30 years.

I'm not saying that every project I've worked on has been a painful experience, because it hasn't - I wouldn't have worked as a Test Manager for almost 30 years if I found every day a chore. I love being a Test Manager. I love being involved in the day-to-day cut and thrust of delivering complex and challenging projects. I love learning about the intricacies of an organisation and how software will make a difference. I love

working with the people who will ultimately use the finished product. But - and it's a **BIG BUT**, there are multiple stakeholders running a business and many of them have an opinion (however misguided) on how much time and money should be spent on testing the software that supports and/or drives their business. If you (as the Test Manager) do not grasp the significance of this then you are going to struggle in your pursuit of happiness in your work. If you do not develop coping mechanisms for these challenges you will always struggle. This is regardless of whether you have sufficient technical experience, leadership skill or management technique.

Today we are living in an ever increasingly connected world and therefore the integration of solutions is becoming more and more complex. We are also living in an era where time to market has never been more of a focus. More complexity, less time, tighter budgets are individually major challenges, but when they all coincide the pressure to deliver increases exponentially; no wonder it is hard to convince software testers to move into Test Management.

In my early days as a Test Manager I took up the Quality/Risk mantle on far too many occasions for my own good. "We must do more Testing" was a line I used far too often as the grey

hairs began to frequent my balding head and the dark lines surround my once youthful and eager eyes. That was until the day Phil King (an exceptional human being and *ManU* supporter) and I shared a taxi on the way to Sydney airport. This was the day he challenged me on my current thinking around software testing and it's place in the *business change universe*.

It was 1996 and Phil was the CEO of *Planpower* (my employer at that time) and he wanted to understand how much our company should invest in what I was peddling at that time. So we talked about all the stuff CEO's care about when running their businesses. We talked about how my skills as a Test Manager could help *Planpower* as well as our clients. We talked in terms of the quality of our services and the risks we undertook when engaging with new clients. Ultimately, we worked on a *Quality Blueprint* that morphed into a Quality Plan that was updated yearly to reflect our growing business. When I left *Planpower*, four years later, it had grown (from a fledgling 5 personnel in 1995) to over 300 amazing consultants and was being primed for sale by the four owners. Today, *Planpower* is still an integral and very successful part of UXC Consulting. That initial discussion with Phil cemented many concepts and beliefs about what a Test Manager should and shouldn't take on in the pursuit of happiness. I will be forever grateful to Phil for challenging my beliefs and ultimately triggering my transition from a software-testing *amateur* into what I consider to be a true professional.

As I said before, software does not exist within a vacuum - it is fully connected to everything that exists within the known universe and therefore it must obey the same forces of nature as the rest of us. Time is finite (though not as one dimensionally finite as some would believe). Highly skilled people are in short supply. Bug-free software is a goal we are yet to achieve; bug-minimisation is all we can hope for with our current assets. So, why do we care when a project stakeholder asks us (for the "nth" time) to explain why the number of bugs found in the latest software release has increased? Why do we take it personally when we didn't write the code or specify the requirements? Why do we test less than we originally planned, so that project managers can meet their own self-imposed deadlines? Why do we allow others (less skilled than ourselves) to estimate how much time it will take to test a solution? Individually, each one of these situations has the ability to result in a stressed Test Manager, let alone if they all happen on the same project - and they can.

My belief is that it comes down to the very make up of the majority of us that have been drawn to our profession. I believe this because I have spent a considerable amount of time (over the past 30 years) analysing my own reactions and responses to these everyday situations and how they make me feel.

I know, for example, that I am a very passionate individual. I am also fussy, conscientious, flippant, playful, competitive, peace-loving etc. etc. I have undertaken many self-assessments, be it Myers-

Briggs, Belkin and several other commercially available products. For example, my Myers-Briggs categorisation is "**ENFP**" which translated into English means that I am (somewhat borderline) extrovert (the **E**), strongly intuitive (the **N**), overtly feeling (the **F**) and perceptive (the **P**). The last three attributes are all very clear when my assessment is analysed but the Introvert/Extrovert indicator has oscillated over the years and is linked directly to my role within the company or project at any given time. When I am asked to lead, I lead and tend towards the extrovert characteristics, but when I'm in a more supporting role (or part of a management team) I'm very happy to let others take up the mantle of "front person". This is because I never seek to lead and prefer to strategise and plan - tasks that are difficult to focus on when you are expected to take the lead on a day-to-day basis. This self-awareness has helped me considerably in the prevention of (internal) stress and the easing of (external) pressure. My interest in patterns has also assisted me in prevention over cure; i.e. I focus on root cause and not symptoms, allowing me to anticipate situations before they become unmanageable. I guess this is akin to managing something as a risk and preventing it from becoming an issue - it's far easier to manage a risk than an issue.

Something else that has helped enormously is my study of Japanese culture and more specifically Samurai methods and techniques. Simply put, I practice and train extensively so that I can function in the "now" and not

be sidetracked with the past or worry about the future. I have read extensively and practiced as often as possible specific techniques to validate their appropriateness and effectiveness. This has led to the development of my own Testing Toolkit that gets taken from project to project and from organisation to organisation in order that I may re-use and adapt as many of my tools as possible.

As I said earlier, I am a very passionate person and this can cause far more pain than pleasure, if misused or misplaced. Many Test Managers and Testers that I have had the pleasure to work with are passionate about their work but many have yet to learn how to harness this for their own and other's benefit. Passion is a very powerful emotion that can transcend culture and beliefs. Normally placid colleagues can suddenly become whirling dervishes if you criticise their ideas or question their beliefs. It is very important to learn when to challenge and when to accept a situation. If you challenge everything you will be seen as difficult and a roadblock, if you accept everything you will be seen as easy to influence and soft. Therefore, choosing when (and how) to challenge is crucial in order to maintain your equilibrium and (sometimes) your sanity.

All too often the Test Manager takes on the role of *Quality Champion* without ever understanding the organisations strategic views on what is considered *good enough* or high/medium/low risk (in terms of running the business). How can we (yes, I still consider myself a

Test Manager) then take ownership of the quality of the software when we are often not engaged until the solution is already half cooked? How can we assume responsibility for the quality of an outcome when we had no involvement in the original business meetings held to discuss a problem and whether it needed fixing?

When I said earlier that many Testers have no desire to become Test Managers it is this sort of behaviour that the Testers see and want no part of - it's a no-win situation.... If you are a Test Manager then your job is to manage (and possibly strategise) the Testing - not to take on all-comers when it is time to triage the latest bug-hunting safari. You are there to represent the efforts of the Test Team, not to pass judgment on the impact of a specific bug remaining unfettered when the "Go Live" date arrives. This is the domain of the business representatives.

I have witnessed countless Test Managers go on stress leave due to the contortions they put themselves through in the name of the *Quality God*. They have lost sight of the independent role that Testing should play and became embroiled in the crossfire between the Business Owners/Representatives and the Project Manager. Provide the ammunition, but **don't ever fire the gun!!**

Stress is something we take on, not something that is given to us. Stress less and have fun, that's why you're doing it - because you enjoy it.

Colin Cherry is the editor of OZTester Magazine and an experienced Senior Test Manager. He can be contacted at [colinda@ozemail.com.au](mailto:colinda@ozemail.com.au)



## Testing Events



## The Next Generation in Software Testing Training Software Testing Essentials



Why is this a next generation Software Testing course?

- Breaking the mold from other fundamental Software Testing courses, this course is competency driven and prepares you for more than just an exam.
- We cover the ISTQB Syllabus and more, so that you are prepared for any scenario. Rich content and hands-on lab based activities.
- Go back to your workplace as a confident and capable Software Tester.

### BOOK NOW

Go to [www.softed.com/courses/sts](http://www.softed.com/courses/sts)  
email [info-au@softed.com](mailto:info-au@softed.com)  
course fees \$2,860 inc. GST

\* Every 5th person attending the same course (date and location) from the same organisation attends for free.



**software  
EDUCATION**  
[www.softed.com](http://www.softed.com)

# Usability Debt

by David Greenlees



We've all heard of the term Technical Debt. It's an easy concept to grasp and an equally easy debt to accrue. Unfortunately it's not so easy to pay off. Technical Debt is a metaphor, created by Ward Cunningham, referring to what is left behind (the debt) after engineering trade-offs are made by IT and business stakeholders in order to meet various project milestones (time, budget etc).

One of the more common forms of Technical Debt is something I like to refer to as Usability Debt. In a relatively short period of time your project can have a multitude of usability issues that need to be resolved in future iterations. Why? Because these issues are so much easier to be given a lower priority by stakeholders. The *training* issue is one of my favourites. "We don't need to fix that, it's a training issue. Just give the users some help text to guide them." If I had a dollar for every time I heard that or something similar... now, where did I put my keys to that Lamborghini?

Usability issues don't generally block functionality in software, and therefore are easy to move past if the context calls for it hard enough. With any project there will be various pressures that will force decisions to leave these types of issues for future resolution. In some cases these decisions are fine and will

produce minimal Usability Debt, in other cases the debt grows at exponential rates. It's important to take stock and make sure you're tracking such issues, even if the decision is to push it into the future. As we all know, these decisions are often out of our direct control.

## Case Study:

At a previous company I spent many years testing and managing a team of testers who tested a web based application. It was customer facing, so usability *should* have been even more important than usual. What was somewhat unique about the application was that there were no direct competitors nor any danger of them. This made the fight for usability issues even harder.

In the early days the team identified several of the 'to be expected' issues surrounding usability of a new web application. Poor navigation, confusing, large amount of clicks to perform what should have been simple tasks, harsh colour schemes, etc. At the time the argument that they were simply usability issues and could be addressed later was easier to swallow, because we truly believed that they would be addressed later. They were not.

Multiple iterations of the project were executed and each one just added to the Usability Debt of the

application. Each time stakeholders would use the same argument.

It wasn't until the application moved into pilot phases that the issues really started to hurt the project. The feedback from users was as we expected, terrible. Yes the application served its functional purpose, but no-one wanted to use it. What would take nearly thirty minutes in the new application took at most five minutes in its predecessor. There were more features and functions in the new application so more time was expected, but not five to six times more.

The types of changes that were required to resolve some of the usability issues were now very complex. By this time the relationship between the code design & development and the UI design & development was well and truly meshed together in the application. Even simple changes would have resulted in significant flow on impacts and possible regression of the application. The cost associated with such change was deemed prohibitive by the project owners and stakeholders.

It was decided to move ahead with the implementation and provide extra training and support to users instead. I'm not sure that an assessment of the cost of the extra training and support versus the cost of making

the changes was ever done. I think that would have had an interesting result.

The users hated it with a passion. Not long after several attempts to implement in different areas of the business the application was withdrawn. Yes there were other factors involved with this decision and Usability Debt cannot be solely blamed, but it was a major factor.

There are parts of the technology that are still in use today so not all was lost. However there is still significantly less shelf space due to area the application takes up while collecting dust. A very expensive mistake.

### **Recommendations:**

In the time since the disaster above I have learned a lot. Some through different jobs and some through self education. Having my time again, or being faced with a similar situation I would be far stronger in my approach to managing the Usability Debt. Here a few things I would recommend:

**Usability Tours** - Software testing tours are very powerful. The metaphor of tours in relation to software testing has been around for a long time. It's hard to pin point exactly who first developed it so credit goes to whomever that was. Taking a tour through the software can provide the user a significant amount of knowledge in a relatively short amount of time, and when those tours are focussed on particular aspects (namely usability) they can be even more powerful.

A simple example can be a Click Tour. Tour the application or particular workflows within it and record the amount of clicks that are required from start to finish. Then compare this to similar workflows in other applications if possible. Having such raw data is a powerful tool when negotiating for a resolution.

Having such tours planned early and highlighted to stakeholders can influence the importance of usability. It will prompt the conversation early, in most cases before you even start testing, which will allow more time negotiate that importance. Get the stakeholders on board early in relation to usability and the battle will be half won.

**Bug Advocacy** - This is all about selling the bugs that you're reporting. These reports can be a powerful tool of persuasion if written well and *sold* to the correct market. Read the following and think about which one would gain a higher priority for resolution:

- The Buy It Now button is in an obscure location and is not differentiated from the other text on the page.
- The Buy It Now button is in an obscure location and is not differentiated from the other text on the page. This has been the cause of similar application's end of life. See the article here on [www.techdisasters.com](http://www.techdisasters.com).

Yes, this is a very simple example. However it highlights the importance and potential power of Bug Advocacy.

**Research** - Often this is a big part of Bug Advocacy; however it can also sit separately. Undertaking research into similar applications and previous problems, then presenting the information found can greatly assist negotiation where required.

A good place for this information is in your up front communications, be it a Test Strategy, a Test Plan or any form of communication that precedes your test execution. If you include various forms of usability tours in your plan for testing then having the researched data and information to back them up will save a lot of time.

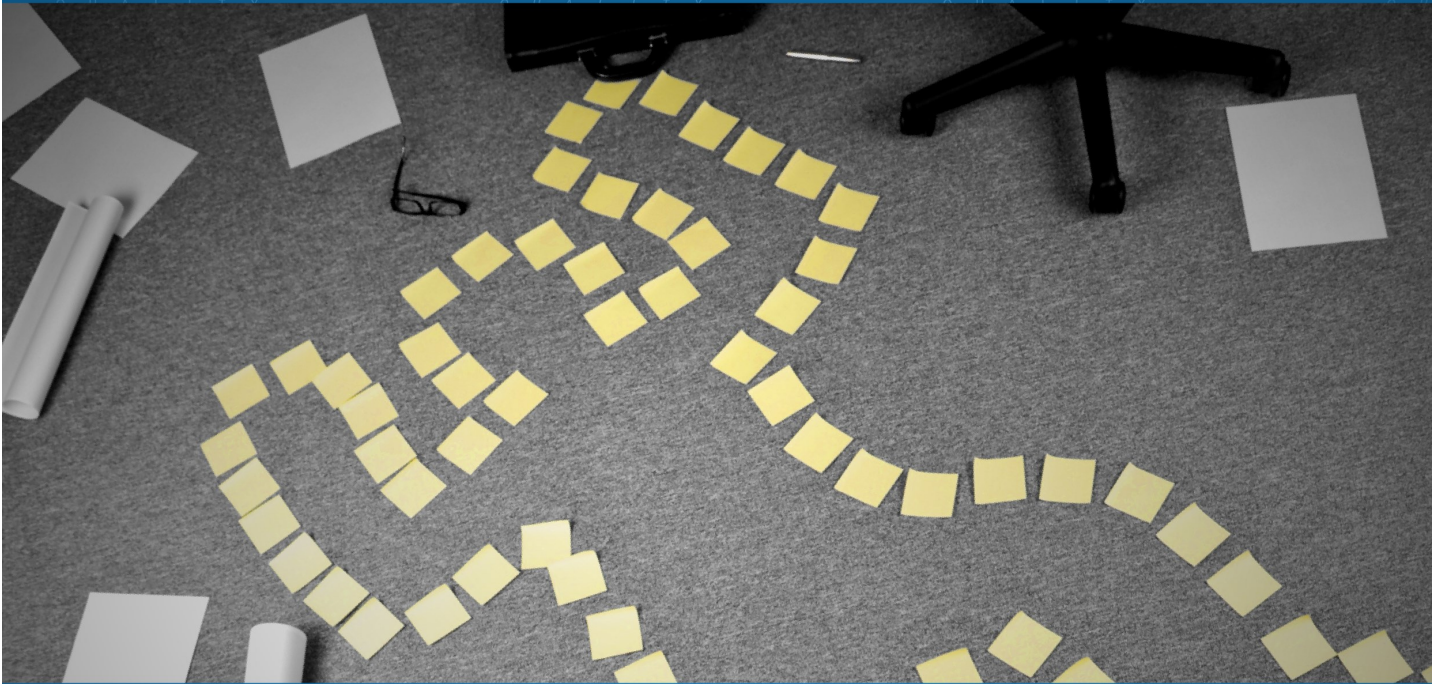
In summary, the key element to all of this is awareness. You need to maintain an awareness of the Usability Debt they may be building up during each iteration. Work on techniques that will help to influence the decision makers. Highlight the importance of early design changes so you don't get to the stage where the cost and time of such changes becomes prohibitive.

Unlike other forms of debt, you cannot negatively gear Usability Debt. Pay it off as soon as you can.

David Greenlees is a Test Lead at RevolutionIT. He can be contacted on [xtremedmg@gmail.com](mailto:xtremedmg@gmail.com)



# IMMEDIATE OPENING: FULL-TIME QUALITY ASSURANCE MANAGER



## Don't work yourself to death. Manage your testing effort with TestTrack TCM.

TestTrack TCM puts you in control of test planning and tracking, providing better visibility over the testing effort and giving you more time to manage your team.

- Know instantly which test cases have been executed, what your coverage is, and how much testing remains.
- Use test variants to target multiple platforms with the same test case for more efficient test case management.
- Manage suites of platform-specific compliance tests, functional tests, and performance tests in one central location.
- Link defects to failed test runs and improve test coverage by creating test cases based on existing defects with seamless TestTrack Pro integration.
- Assign tests to your QA team, track results, and report on performance and workload.
- Schedule automated tests to run 24x7.



Visit [www.seapine.com/oztester](http://www.seapine.com/oztester) to learn more.

**Free  
30-Day Trial**

© 2013 Seapine Software, Inc. All rights reserved.



# From the TestToolBox

by OZTester Staff Writer

## The Test Managers Toolkit Series: Tools 1 and 2

I've been a Test Manager for over 20 years and in that time have developed my own version of a tradesman's Toolkit - the equivalent of what my Dad carried around when he was a carpenter. The only difference is that my toolkit contains spreadsheets, URL's, e-books, checklists, estimation techniques, testing reference materials, management tools etc., not hammers, drills, saws and screwdrivers.

My toolkit has changed in many ways since I first began putting it together. Initially it was a (hard copy) manual, then I converted it to digital and carried it around on a USB drive (with a back up on my home computer). Then it expanded to several USB drives (backed up on a laptop), before going through several more iterations until it's latest incarnation on my iPad (which, in itself, is part of the toolkit) - backed up to the Cloud. In fact, my iPad is currently the most important tool in my toolkit.

In this issue I am going to share some of the tools I use for selecting, building and managing a Test team. Over my career I have interviewed hundreds of testing professionals, performed annual assessments of them, designed career paths and development programs, developed and delivered training programs while also coaching and mentoring people.

I always start from the premise that I can teach folks technical skills far easier than I can influence their behaviours and beliefs and therefore I tend to select people based upon their attitude and behaviours. In order to make this a repeatable process I have developed a list (it's an A to Z list in it's current form) of behaviours, beliefs and attitudes where the people being assessed need to meet 80% of the criteria (based upon the Pareto Rule). This list comes in various flavours depending upon the type of organisation, the industry, the type of projects they are likely to work on etc.



### Tool 1: The Behaviours & Beliefs Interview List

Ambition  
Bravery (in making decisions)  
Creativity  
Daring (to be different)  
Equality  
Family & Friends  
Grace  
Healthy Lifestyle  
Instinct  
Justice (for all)  
Knowledge (the quest for learning)  
Love (for one another)  
Maturity  
New Ideas (open to)  
Optimism  
Passion  
Quality  
Respect  
Sustainability  
Tolerance  
Understanding (of others)  
Vitality  
Wisdom  
Xcellence (yes, I know that's a slight cheat)  
Youthful (in attitude, not necessarily years)  
Zealous

We also identify at least three of these attributes as ESSENTIAL and not negotiable in order for someone to join the team. Our success rate with this approach is only one failure in over 200 hires over the past 10 years....

The second tool I'm going to share with you today also comes in several variations and has multiple uses. It's something I call my "**Do I Care?**" Tool. I need this tool because I need to take in enormous amounts of information constantly and sometimes very quickly. The only way that I can do this is to allocate everything with a rating from 1 to 4, where 1 is the least important and 4 is the most important. The reason I use four categories is that any more than four categories is too complex and it also prevents me putting everything in the middle (i.e. if I used the usual convention of *five* as a small sample allocation).

#### **Tool 2: The Do I Care Organiser (Version 1 - Documents, Email etc.)**

**Category 1:** Stuff I will probably never read

**Category 2:** Stuff I will ask one of my team to review and categorise as either a **1** or a **3**

**Category 3:** Stuff I will read within 72 hours

**Category 4:** Stuff I need to read within 24 hours

#### **Tool 2: The Do I Care Organiser (Version 2 - People)**

**Category 1:** People I don't need to influence or meet

**Category 2:** People I will delegate to one of my team to meet and assess as a **1** or a **3**

**Category 3:** People I will spend time with whenever I can

**Category 4:** People I will spend time with every day (such as my team)

#### **Tool 2: The Do I Care Organiser (Version 3 - Issues, Risks, Dependencies & Constraints)**

**Category 1:** Stuff I will never care about

**Category 2:** Stuff I will ask one of my team to review and categorise as either a **1** or a **3**

**Category 3:** Stuff I will review every 24-48 hours

**Category 4:** Stuff I will review every 2-4 hours

#### **Tool 2: The Do I Care Organiser (Version 4 - Meetings)**

**Category 1:** Meetings I delegate to one of my team

**Category 2:** Meetings I will attend when I can, but don't require preparation (therefore I can delegate whenever possible)

**Category 3:** Meetings I will attend that require preparation and are therefore harder to delegate

**Category 4:** Meetings I am organising and therefore will almost certainly attend

**Note:** With the Meeting version of this organiser there are always exceptions due to project priorities and emergency meetings taking precedence.

# Mobile Application Testing - Return of Technical Tester?

by Reghunath Balaraman

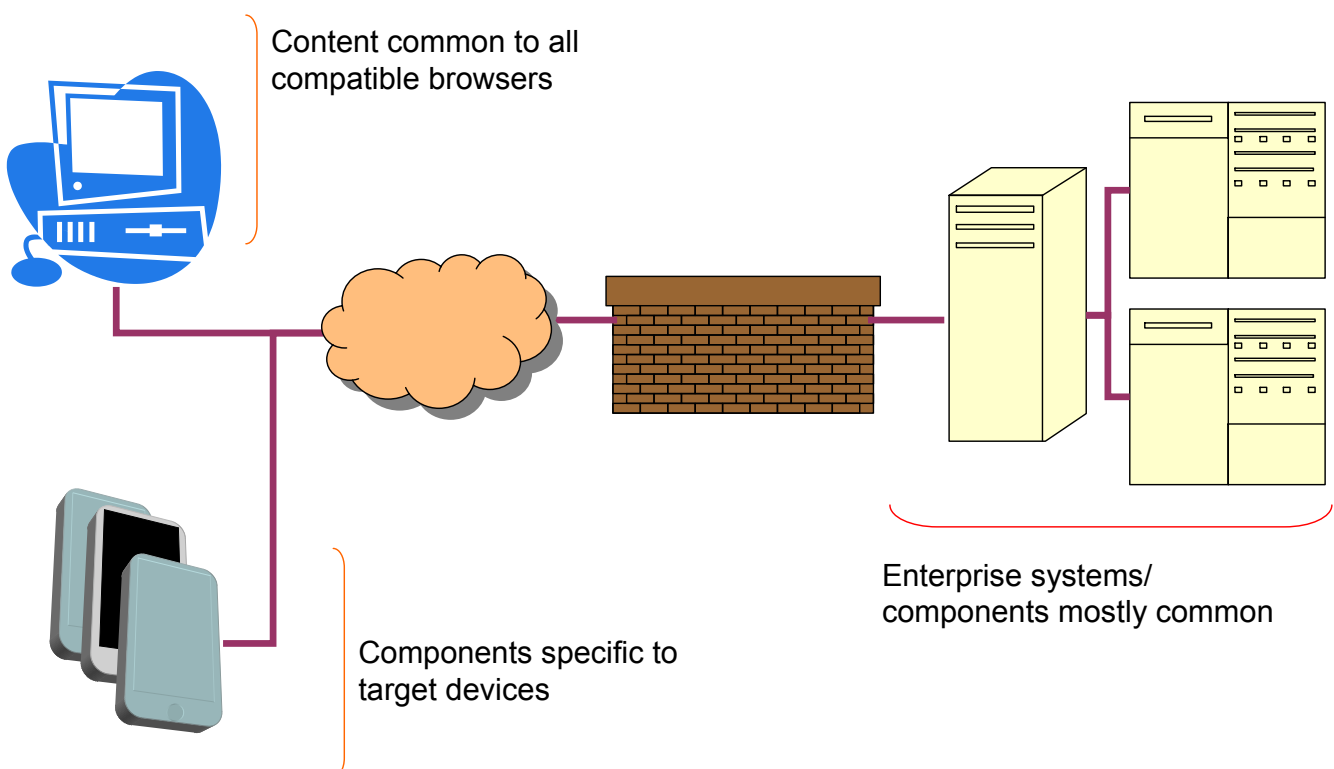


The trend in the number of mobile devices, including smart phones and tablets, is a very significant development that will have a far-reaching impact on applications development and testing. Applications are not only becoming mobile-enabled, they are also becoming an integral component in completing business transactions. Take, as an example, placing a trade in a trading system. Examining the technical skills and tests required to validate a mobile application (app) reveals that not only are there several commonalities to testing apps across domains but there is also a limited dependency on domain-specific knowledge. Are we seeing a shift in focus from the domain knowledge that made testing vertical-oriented (segmented based on industry or domain) to testing becoming a horizontal offering (spanning across industries) as it used to be? In order to answer this question, we

need to deep dive into the technical aspects of testing in the world of mobile application quality assurance.

In theory, we have three broad categories of software for mobiles – the native apps, mobile web and hybrid. The native apps reside in a hand-held device whereas the mobile web accesses web content using a mobile browser. Hybrid applications utilise a combination of both hybrid and web features. While there are several commonalities between the three, they are unique in several ways.

Testing a native or hybrid application needs a different strategy and approach than testing browser-based applications on mobile devices. Functionalities ranging right from the code base to the capabilities of the native hardware will have an influence on the testing outcome. Testers need to



Mobile devices share with the web based systems, several enterprise systems and components

understand and develop suitable strategies to focus on functionalities as well as device and OS-dependant features that need to be tested on each target device.

When testing the mobile web app, the focus is primarily on the mobile browser's capability to interpret the code and deliver the content correctly and consistently. Different browsers use different engines internally. For example, Chrome uses WebKit while Microsoft uses Trident. The way they render content is not universal, especially while handling evolving standards like HTML5. Native apps reside in the device and interact more closely with the host OS and the hardware and software resources of the device. Here, testing becomes very critical as different operating systems use a different code base. Hybrid apps extend the challenges by combining both native and web characteristics into one.

Device manufacturers are stretching their creativity in an attempt to increase their market sharing by designing devices with all possible variations in form factor, processing power, user interface standards, and network capabilities. OS choices are as diverse as the hardware options. With the availability of different Android versions, device manufacturers and telecom carriers modify operating systems to optimise performance in their networks. Android currently has around 75% of the market share followed by Apple with a share of 15%. However, the market is very dynamic and the user base is not uniform across geographies. Arriving at an optimum combination of devices to test is one of the key factors for a successful test strategy for mobile devices.

The nature of the application is the main consideration while developing a test combination matrix for mobile applications. Building a target user map is the first step towards identifying the device combination. Market research can provide pointers to the end user preferences of devices and platforms. Depending on the type of the application, the dependency of the design elements on the device and OS also needs to be determined. This helps further narrow down the selection of target devices. Testing of carrier-dependent capabilities is an additional test that mobile testers need to perform. Overall, the testers need to repeat a large number of tests on a variety of devices and several operating systems and their variations.

Application development tools and mobile manufacturers provide emulators to test mobile applications. These emulators help run the code in an emulated environment to check how it will perform when it is deployed on a real device. This is a very good tool for unit testing and to some extent, the initial sanity testing. However it can never replace a real device as the emulator is a piece of software with its own limitations and cannot completely mimic the application behaviour on a real hardware. This necessitates a large investment in device inventory.

Device obsolescence and new device introduction are other factors that challenge the test combination on an ongoing basis. The current trend of major mobile manufacturers rolling out at least two new or upgraded devices every year – along with several OS updates – calls for dynamism in the test combination matrix coupled with frequent updates. In most cases, where the target devices are large, maintaining the device inventory is a large investment as well as ongoing expense. Many providers have come up with innovative solutions that help users access devices remotely, and for fixed periods, thus eliminating the need to invest and manage devices in-house.

Given the repetitive nature of the tests, automation is an obvious choice. However, it comes with its own challenges. There are several technologies that drive mobile automation from complex image recognition techniques to a combination of image and text identification. The devices and their form factors have a high level of influence on the way the user interface components are positioned and displayed. Thus, the ability to identify these components agnostic to their position and rendering is an important requirement for a successful automation tool. The learning curve needed to use the tool, its integration with test management tools, and ability to extend the existing automation in non-mobile components of the application are also of utmost importance in selecting a tool. My personal preference is to extend the existing functional automation tool rather than introducing a new tool that is aimed solely at automation of mobile components. A common tool will provide more opportunities for end-to-end automation by not limiting itself to testing the mobile functionality, and will also help extend the existing functional



automation scripts to cover mobile specific tests in cases where such scripts exist.

Security is another important consideration when going mobile, especially using native and hybrid applications. These applications tend to store a variety of information at the user end, enhancing the performance and user experience. As these devices are exposed to several other applications that may have malicious intent, verifying the security of the mobile application and its interface with the enterprise systems is of prime importance. Similar to the discussion on functionality, security also needs to be evaluated in combination with the device and OS, with an extended focus on jail-broken devices which are more vulnerable to security threats.

Mobile applications have a significant part to play in enhancing the customer experience. With a good number of companies using multiple channels, including social media, providing an intuitive and consistent user experience is of paramount importance. This includes consistency in user interface design and across internet channels, ease of operations, and performance.

A common pattern that we see in the above is an independence from the functionality that is

dependent on a domain or industry. While the functionality is very important, the purpose of making applications mobile-enabled; making them easier and more accessible, and providing a better user experience makes it less domain-dependant. In fact, most of the discussion above was completely domain-neutral.

It requires deep understanding of the architecture of mobile applications and design patterns, their relative merits and drawbacks, and experience in evaluating usability and performance to analyse the user experience potential of an application. This also allows extensive reusability across industries and the ability to use common tools and techniques across scenarios. Hence there is an increased focus on the technical abilities of the tester and perhaps the re-birth of the 'technical tester'. This surely will not make the domain-focused or context-driven testers obsolete. But it will emerge as a desired skill – if not establishing itself as a parallel role to the domain-focused functional tester.

Reghunath Balaraman an Enterprise QA Transformation Principal Consultant with Infosys in Sydney. He can be contacted on [b.reghunath@gmail.com](mailto:b.reghunath@gmail.com)



**Enterprise tester**

**“Enterprise Tester is by far the best tool out there for agile test case management and integration into open tooling, such as JIRA”**

Tom Gilmour, Head of Quality Infrastructure, PayPal

Logos of partner organizations: intel, LG, Department of Veterans Affairs, UnitingCare, SAIC, SKY, Open Universities AUSTRALIA, eBay, CSC, PayPal, orange, ORION HEALTH, accenture, BAYER, Bayer HealthCare, STANFORD UNIVERSITY

[www.enterprisetester.com/oztester](http://www.enterprisetester.com/oztester)



# CITCON Sydney 2013

by **Oliver Erlewein**

At the beginning of February, I headed over to Sydney to attend CITCON ANZ 2013. CITCON (spoken

KITKON) stands for Continuous Integration and Testing Conference. The last time I took part in one was in 2010, when it was held in Wellington. Already back then I was sold on this conference style. Just imagine a conference that is self-organising! And here is how it works...

The conference is held over two days, but the actual conference is only one day. The evening before the conference day is used to get to know the 100+ attendees and to define the conference contents. The MCs and creators of CITCON are Paul Julius and Jeffrey Frederick. They have a unique way of guiding you and your fellow attendees through the procedure. After all the introductions are done, you are asked to come up with potential topics on post-its and one-by-one present them to all before putting them up on the roster. Apparently down-under we have a reputation for being the noisiest while this happens (probably because at this point alcohol gets added to the eclectic mix).



*The roster is just a table showing the track times*

In Sydney the conference was held at the Atlassian ([www.atlassian.com](http://www.atlassian.com)) premises, and we had 5

separate rooms and 5 tracks throughout the day. Once the topic gathering was over, it was voting time. Everyone could come up and “tick” the topics they’d like to hear more about. Max one tick per card but you could tick as many cards as you liked. As you can imagine this is even more chaotic. At some stage, the marking subsides, and now the day can be planned.....BY EVERYONE!



By this time PJ & Jeff have handed over the control of the conference to us attendees - and only us! We were all responsible for making it a success (or not). It sounds a bit scary but it actually works amazingly well. People go up to the board and sort topics together, shift them according to interest into bigger and smaller rooms, try and make topical day tracks from them or bring in some personal aspect. This planning and re-planning actually takes place throughout the conference day too, so a constant eye on the schedule is advisable.

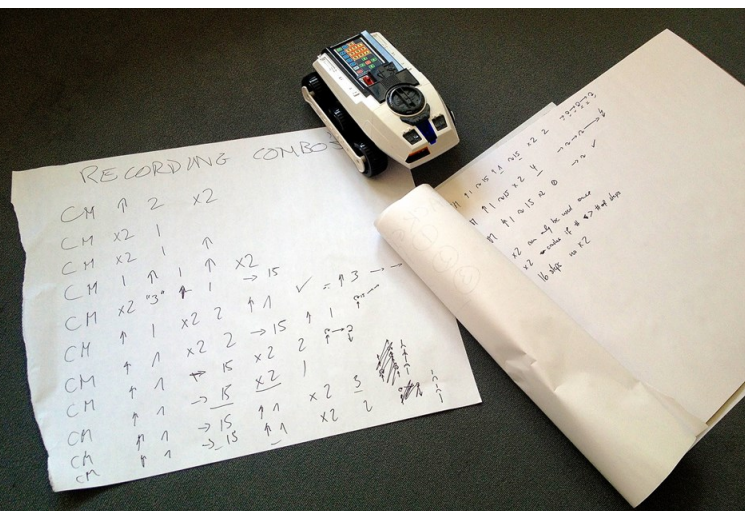
On the day itself, people scatter quickly to the tracks they want to participate in. One golden rule (and I think the only real CITCON rule) is, that you use your two feet. If you want to change tracks or get a beer/juice, do so, whenever you feel like it. There is no reason to stay where you are, and presenters/participants shouldn’t feel offended by you getting up and leaving.

If you find yourself in a track, usually the person(s), who suggested the topic will start the discussion off. It starts with something like “I need to know more about the topic and have no clue so please can we discuss it?” or someone giving a 15 min intro to the topic in all detail. Unlike the (X)WST workshops, there is no controlled discussion style and the discussions evolve as they are needed. That means that from track to track the experiences are wildly different. So far I have never been disappointed by a track.

With a whole bunch of testers attending this conference had more of a testing lean. I visited mostly the more testing related tracks.

Anne-Marie Charrett was 1<sup>st</sup> off the rank with a fun hour of testing/exploring her BigTrack robots.





*Some tests for the BigTrack and rules found*

This was heaps of fun, and I learned a lot about how different testers tackle a given problem. Makes me wonder, if assigning one tester to a task is actually ever a good idea. This was followed by “Testing vs Checking”, which was my topic. It was interesting to discuss this with a room full of testers and developers. Generally the feel was, that the differences were understood and the discussion was around semantics and who was slated to be doing what.

“Jeff is angry at silo thinking and DevOps isn't helping” was probably the best presented presentation I attended. His rant was around how we are missing the opportunity to really cooperate and help each other, thinking of others, and being more accepting of new and important input from left field in order to deliver better projects. IT is still dominated by silos, even in Agile projects. The ensuing discussion was “vivid” to say the least, and would have probably gone on till the wee hours if we hadn't been kicked out by the next track.

Another interesting track was “Testing Lightning Talks”. I only got to it at the tail end but it was a melee of quick-fire 60sec presentations by everyone

in the room. The topics mutated through a whole string of presentations. It was absolutely fun and very interesting to see everyone take the essence and throwing it out there. I'm sure this is not the last we have seen of this type of session.

At the end of the day PJ & Jeff took the lead again and gave the whole event a solid ending with another check-out by everyone. It was all definitely worth the trip! I found it amazing that nearly 10% of attendees came from New Zealand.

CITCONs are mostly financed through the Open Information Foundation (OIF) and company sponsorships so you are not required to pay for the attendance though you can donate if you so wish. The next CITCON will be in February 2014 in Auckland. See you there!

For more information see <http://www.citconf.com>. Get on to the mailing list, especially if you want to sponsor, contribute a venue or be a volunteer.



*PJ and Jeff kick CITCON off*

**Oliver Erlewein is a Wellington-based Test Consultant. He can be reached at [oliver.erlewein@gmail.com](mailto:oliver.erlewein@gmail.com)**

*Some headed straight for the pool room*



## TestWest Conference, Perth, February 20, 2013

*Colin interviews Toby Thompson, MC of the inaugural TestWest Conference recently held in Perth.*

As a very interested party in the establishment of a Software Testing Community (or communities) in Australia, I was very interested to hear that WA has just experienced its first Software Testing Conference. It took place at the Perth Arena on February 20. As I was overseas at the time of the event, I couldn't attend so I thought I'd get a view of the day from Toby Thompson of Disqover (a leading Melbourne-based ISTQB Training provider), who was the *Master of Ceremonies* at the event.

**OZTester:** *How did you (as the Principal of a Melbourne-based Training provider) come to be MC at this inaugural Testing Conference in Perth?*

**TT:** Disqover got its first big break in Perth back in 2006 with a defence organisation called Thales – we have been providing training to them and many other companies in the Perth area for the past 7 years. Perth is a relatively small testing community and so news began to spread about our training delivery capability and success rates for ISTQB Certifications. We are specialists in software testing certification training and we focus on providing a great experience for all our trainees, before, during and after our courses. As a result many of our clients come to us through word of mouth.

People obviously move around in the testing market and as a result one of my former students referred Disqover to the team at Bankwest around 5 years ago. Since then we have been really fortunate to be Bankwest's dedicated ISTQB training provider from the Foundation to Advanced Certification level. I met with Caroline Carder, who is Head of Solution Testing - Enterprise Services at an Agile conference in Melbourne about a year ago and she mentioned that she was planning the first testing conference to be held in Western Australia and she asked me to be the MC! I was *gobsmacked*, it isn't something I've ever done before, but I jumped at the chance.

During her introduction at the TestWest 2013 conference she said the easiest decision they had to make for this conference was to ask Toby to be the

## TestWest 2013

February 20<sup>th</sup> at Granite Room - Perth Arena

Agility, Mobility, Automation, virtualiZation, Environments

Master of Ceremonies; I was so flattered and really happy to be asked.

**OZTester:** *Why do you think it has taken so long for WA to host a Software Testing Conference?*

**TT:** I'm really not sure. Perhaps the size of the market compared to the Eastern states is relatively small. Maybe there's a belief that there is greater commercial potential on the East Coast and because that is where the *Testing Hubs* are located.

**OZTester:** *The Conference was booked out. Do you think this was due to pent up demand or because it was a FREE event?*

**TT:** A bit of both, I think. What became very apparent was how pleased everyone was and that it was about time the Perth testing community had an event of this kind. I have been to many conferences where people didn't show up even though they had paid! Having a free conference is always going to be highly appealing, but still the absentee rate was tiny – I'd say 95% of the people who said they would come did come, which is impressive.

**OZTester:** *Given your exposure to similar East Coast events, how would you rate this one?*

**TT:** Well I may be biased (because I was the MC), but I thought it was first class; in fact, Ed Cortis (Head of Solutions Delivery at BankWest) called it a *World Class* event. I think the fact that the majority of the people who presented were *home grown* talent also made it special.

**OZTester:** *Maybe you could give me three reasons why this event was so successful?*

**TT:** just off the top of my head:

- 1) It was a breath of fresh air for the testing community in WA
- 2) The balance, mixture and quality of presentations and the panellist session meant that there was something for everyone



3) *Bankwest's* brilliant team Caroline Carder, Kevin Chapman, Min Xu, and Bruce Dominguez did a fantastic job of organising the event. They did it all on top of their already demanding work schedules. There was also a great team of *Bankwest* helpers (the "Smurfs") making sure everyone was okay during the day.

**OZTester:** *The Agenda seems well balanced in its content, was there a standout session from your perspective?*

**TT:** Following an opening address by the CIO of Bankwest, Three *Agile Coaches* started the proceedings and really set the tempo of the conference. These guys presented a very *energising* social experiment which attempted to prove Darwin's theory of survival of the fittest and the importance of adapting to a new way of working – adapt or die was the underlying theme and it underpinned the agile transformation taking place at Bankwest and about how they were moving away from the old silo working environments to a more collaborative and coexistent style of working. The ultimate goal was to achieve customer satisfaction through agility and collaboration.

It's really difficult to say which session was the best but I really enjoyed Louise Capper's presentation. She is the Software Verification & Validation Manager at Thales. *Automation - Mission Impossible* was the name of her presentation. It was an interesting piece with great audio-visuals - frigates and missiles and helicopters and jet fighter planes dashing here, there and everywhere, accompanied by some very vibrant music; so that was that was one of the most exciting to watch.

I was also really impressed with Bankwest's presentations – a few of the guys were former students of mine, they weren't professional presenters and didn't have a great deal of exposure to presenting but they did present articulately and eloquently about topics that were reasonably technical such as Continuous Delivery, Service Layer testing and Service Virtualisation. I've been to too many presentations at conferences where the more technical presenters have made it really difficult for the audience to understand what they're talking about. The guys did a sterling job at getting their stories across.

Overall, I thought the quality was high across the board – the theme of the conference was Agility, Mobility, Automation, Virtualization and Environments – interesting topics based on real world scenarios.

**OZTester:** *Are you aware if there was any live Twitter or other social media chat on the day?*

**TT:** I wasn't aware of any social media communications on the day, we mentioned the twitter hashtag for the conference but there wasn't much of an uptake on that kind of social media. Collaboration, sharing and networking was encouraged during all the *breaks* and during the drinks session at the end of the conference.

**OZTester:** *Is there anything else you'd like to add?*

**TT:** Caroline Carder shared some of the attendee feedback with me the other day. Here are a few quotes:

"I came to the event to test the water for the rest of my team. I'll be encouraging them to all come along next time"

"*Gary the Squirrel* was great fun"

"Thank you for a great first Conference; already looking forward to the next one"

"Could we have a bigger venue next time, so that more people can attend?"

"Great organisation - and lots of fun :)"

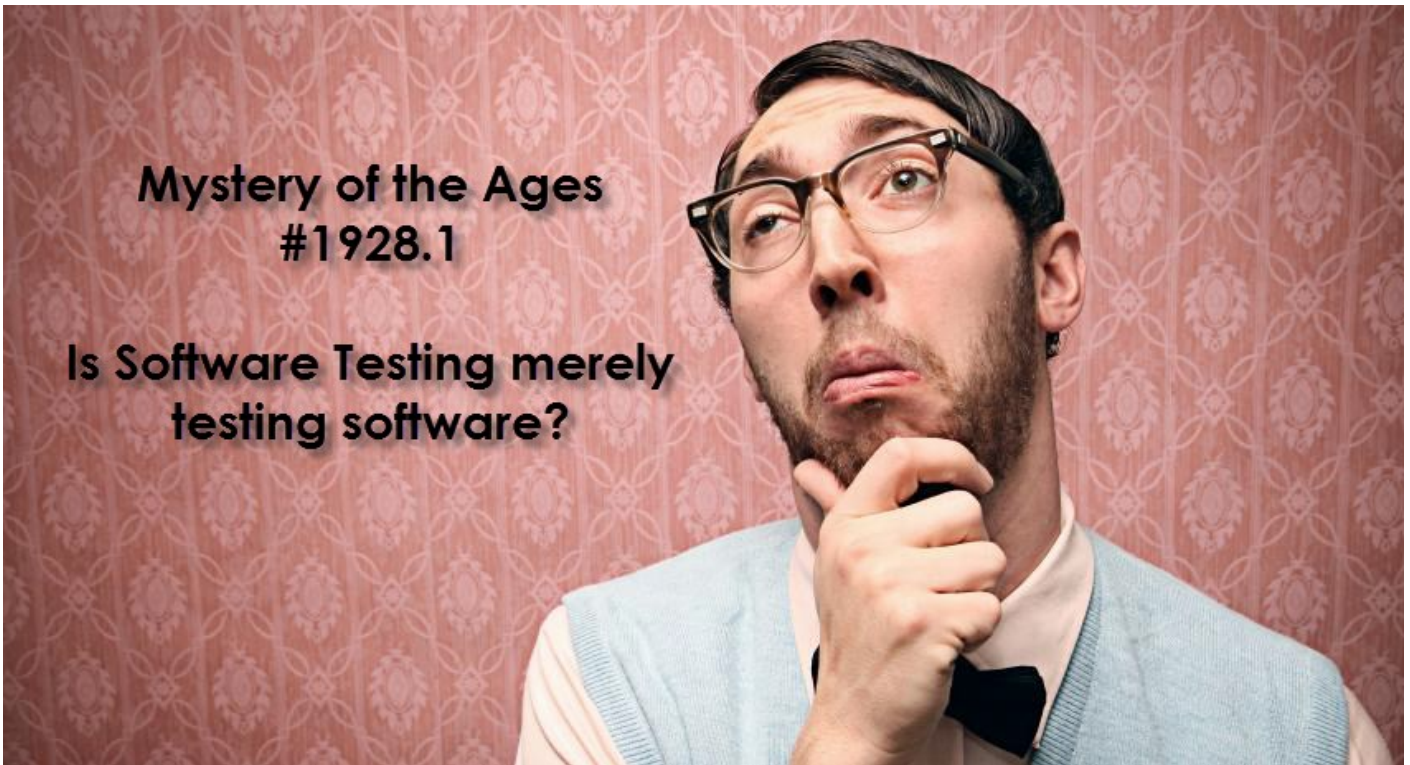
So, it seems that I missed out on a great event over in WA. Let's hope this is the start of something bigger and maybe ANZTB, STANZ or IGNITE can be convinced to take their Conferences to WA very soon - there is obviously pent up demand.

**Presentations from the day are available on the TestWest website at**

<http://www.bankwest.com.au/testwest>

**Mystery of the Ages  
#1928.1**

**Is Software Testing merely  
testing software?**



Does this get you thinking? If so, drop me a line at [colinda@ozemail.com.au](mailto:colinda@ozemail.com.au) with your thoughts and we'll publish the best ones in the next OZTester issue -



# Great Bugs We Have Known and Loved

by OZTester Staff Writer

*"Great Bugs We Have Known and Loved" will be a regular feature in OZTester and to kick off our first report we will return to a Bug that one of our leading Banks would rather we forget.*

The majority of businesses today have some sort of online presence, or at the very least rely on computer software in order to transact on a day to day basis. It therefore follows that based upon what we know, as software testers, that this software will occasionally fail. Whether it fails spectacularly or in a way that no-one (except the ever vigilant technical support staff) notices is sometimes down to luck, but mostly down to good software development processes and procedures. Unfortunately, these processes and procedures are not always fool-proof and the odd "super-bug" gets through.

Having tested software inside nearly all of the first and second tier banks in Australia, I am in a pretty unique position to provide an insight into why these organisations are not infallible; however, it would be unprofessional and probably a case of gross misconduct/negligence if I were to divulge their weaknesses in this column. Therefore, I will provide an example by way of a previously

reported software bug that was classified as **nab** (not a bug).

It is definitely within the interests of major organisations to "blame" a faceless member of staff (we all know them affectionately as *Mr & Mrs Human Error*) or an unexpected technical glitch (spin for that well known army laggard *Major Software-Stuff up*).

Unfortunately, the general public rarely understands the differences between a flawed SDLC, a hack or a self-inflicted performance wound.

Every major bank in Australia has hurt some of its customers with a software-related bug in the past 10 years and some are serial offenders. My *favourite* bug from these (once-upon-a-time fee-friendly financial fiends) occurred a few years ago as a result of a minor software update being applied without sufficient Regression Testing (my words not theirs). It wasn't a complex change, but it **was** in a *sub-routine* that was accessed hundreds of thousands of times every night as part of the organisations "Batch processing" function. Now, anyone who has an insight into how most of our Banks process transactions will know that they are multi-streamed in order to meet the tight overnight window and

therefore if a mishap occurs it is almost impossible to stop processing in all streams immediately and simultaneously.

It is also impossible to interrupt this processing without first analysing and confirming that there has been a problem that requires scrutiny.

The main reason I am so fond of this bug is that it was (what I classify as) a *Super Bug* - it not only infected the host bank, but it spread itself far and wide by infiltrating all the other banks and even some non-banking trading partners. I just love it when Integration Testing gets under-cooked... The result of these infections is countless hours of data manipulation and file fixes in order to undo something that you innocently accepted as valid data. Incidentally, how much compensation is paid to trading partners is very rarely uncovered and the exposure to the public is always under-estimated.

Software bugs never die, they just crawl under a rock and await their next outing. Au revoir, Monsieur Le Bug - until next time...

## And now it's your turn...

If you would like to be involved with and/or contribute to future OZTester issues, you're formally invited to submit your proposals to me at [colinda@ozemail.com.au](mailto:colinda@ozemail.com.au).

Articles should be a minimum of ½ A4 page at Cambria 11pt font and a maximum of 2 A4 pages for the real enthusiasts. If you wish to use names of people and/or organisations outside of your own, you will need to ensure that you have permission to do so.

Articles may be product reviews, success stories, testing how-to's, conference papers or merely some thought-provoking ideas that you might

wish to put out there. You don't have to be a great writer as we have our own staff writer who is always available to assist.

Please remember to provide your email address which will be published with your article along with any photos you might like to include (a headshot photo of yourself should be provided with each article selected for publishing).

As OZTester is a free magazine, there will be no financial compensation for any submission and the editor reserves the sole right to select what is published and what is not.

Please also be aware that your article will be proof-read and amendments possibly made for readability. And while we all believe in free speech I'm sure, it goes without saying that any defamatory or inflammatory comments directed towards an organisation or individual are not acceptable and will either be deleted from the article or the whole submission rejected for publication.

### Feedback

OZTester is open to suggestions of any type, indeed feedback is encouraged. If you feel so inclined to tell us how much you enjoyed (or otherwise) this issue, we will publish both praise and criticism, as long as the latter is constructive. Email Geoff on [ed@nztester.co.nz](mailto:ed@nztester.co.nz) and please advise in your email if you specifically do not want your comments published in the next issue otherwise we will assume that you're OK with this.

### Sign Up to Receive OZTester

Finally, if you would like to receive your own copy of OZTester completely free, even though we're real low tech right now. there's still a couple of easy ways: 1) go to [www.nztester.co.nz/oztester.html](http://www.nztester.co.nz/oztester.html), or 2) simply click [here](#) - Ed.



- 8 Cognizant
- 13 Discover
- 17 Software Education
- 20 Seapine Software
- 25 Catch Software