

# OZTester



The Quarterly Magazine for the Australian Software Testing Community and Supporters

ISSUE 5 DEC 2014 - FEB 2015

COMPLIMENTARY

## *In this issue:*

**Interview with Matt Wiggins, Infosys Lodestone**

**To QA or To Not QA: That Is the Question**

**Test Summary Reports**

**Combined Acceptance Testing**

**The Pragmatism of Test Estimation**

**Dealing With Professional Manipulation**

**Ever Had That Sinking Feeling?**

**Myths Around Skills Required for Automation Testers**

**Let's Test OZ & StarWest Reviews**

**+ Testing Events and more....**



# OZTester Magazine

Editor: Geoff Horne

[ed@OZTester.co.nz](mailto:ed@OZTester.co.nz)

[geoff@OZTester.co.nz](mailto:geoff@OZTester.co.nz)

ph. 021 634 900

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

[www.OZTester.co.nz](http://www.OZTester.co.nz)

Advertising Enquiries:

[ed@OZTester.co.nz](mailto:ed@OZTester.co.nz)

## **Disclaimer:**

Articles and advertisements contained in OZTester Magazine 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 and publisher, 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 2014, 201 - OZTester Magazine, all rights reserved.

# IN THIS ISSUE...

*Click on title...*

- 5 Interview with Matt Wiggins**  
Test Practice Manager, Infosys Lodestone
- 8 Recognising and Dealing with Professional Manipulation**  
Geoff Horne, Editor, OZTester Magazine
- 11 Myths Around Skills for Automation Testers**  
Min Xu, BankWest
- 14 Ever Had That Sinking Feeling?**  
OZTester Magazine Staff Writer
- 18 Combined Acceptance Testing**  
Mark Taubert, NAB
- 24 Test Summary Reports**  
Heather Liney, Telstra
- 26 The Often Untapped Value of Business End User Testing—Part I**  
Tim Fox, Independent Test Consultant
- 27 The Pragmatism of Test Estimation**  
Rajesh Mathur, Government of Victoria
- 29 To QA or to Not QA: That Is the Question**  
Rowly Emmett, Access Testing
- 33 Review: Let's Test Oz**  
Mike Talks, Datacom
- 39 Review: StarWest 2014**  
OZTester Magazine Staff Writer

## REGULAR FEATURES

- 37 Testing Events**
- 38 NZTester and OZTester Magazine Back Issues**

## The Journal For Australian Test Professionals

Well, its been a loooonnnng time since the last OZTester Magazine and we do apologise for this. With out first conference running in Auckland plus an unexpected hospital sojourn for yours truly plus yada yada excuses excuses....I'm sure you get my drift however here here it is at long last—I hope you think it was worth the wait.

To compensate for the missed issue in there, we've made this one larger than usual so it will hopefully give you something read while at the beach on your summer holidays.

We have articles from first time contributors Heather Liney, Rowly Emmett, Min Xu, Mark Taubert and Tim Fox plus we hear again from Rajesh Mathur after his excellent piece on Authority Bias in OZTester02.

We also reveal just a little of what we're planning for 2015 on our Test Events page.

Lastly, this will be the final OZTester issue for 2014 and so it's fitting that we wish everyone a Merry Christmas and a Happy New Year. By the time this issue is received, we'll all be well into the annual silly season and most of us will probably be thinking of camping, fishing or whatever else we like to do in the holidays.

Keep well and in touch.



**Wishing  
everyone  
a very  
merry  
Christmas  
& a happy  
new year**





# Matt Wiggins

## Test Practice Manager

### Infosys Lodestone

*Our interview this issue is with Matt Wiggins, Test Practice Manager at Infosys Lodestone. Matt has vast experience across the testing industry including stints with TestLogistics, K J Ross & Associates and the Royal Bank of Scotland.*

---

#### **OZTester: Can you please describe Infosys Lodestone?**

Infosys Lodestone provide senior consulting services to our portfolio of clients. Typically we are engaged to undertake Enterprise Testing Strategy/Transformation, Programme Test Manager/Director roles. Additionally we provide practitioner consulting services in areas such as Test Environments and Test Data, Non Functional Testing and Agile Testing strategy and implementation.

Our overriding aim is to evidence the fact that we are able to provide proactive and trusted advice during our engagements, leveraging the thought leadership and wealth of experience each of our consultants offers our client base.

---

#### **OZTester: What testing services does Infosys Lodestone offer?**

As outlined above, and at a high level, Infosys Lodestone will generally consult at an Enterprise or specialist service level, or undertake very senior testing delivery roles. That said, our practice offerings really fall into four main areas:

##### Enterprise Advisory Services

- Advisory services on Enterprise Testing and QA Strategies
- Enterprise Testing and QA roadmap definition
- Business case and value articulation
- Enterprise Testing and Programme reviews and health checks

##### Testing and QA Transformation

- Audits and QA capability assessments
- Capability uplift and roadmap implementation
- Identification and implementation of Enterprise and Programme accelerators, enablers and efficiencies
- Strategising Test Centres of Excellence

##### Testing Delivery

- Test Programme Management
- Specialist consulting for Agile Testing, Mobile Testing, Cloud Testing and SOA Testing
- Release and Implementation Management

##### Specialist Consulting

- Test Environment Management and Test Data Consulting
- Non Functional Test Consulting
- Functional and Mobile automation Consulting
- Testing Tools: selection and consolidation
- SAP Test Consulting

---

#### **OZTester: What do you believe makes Infosys Lodestone different?**

I think it's important to lead rather than be lead. I look for consultants who are recognised practitioners in their field when hiring for Infosys Lodestone. Our brand is built on our reputation of being able to deliver outstanding outcomes to our clients in a transparent and ethical fashion. Sometimes this means sitting down and having tough discussions with our stakeholders, but I'd much rather everyone was on the same page than not, and the feedback we get indicates that our communication approach is appreciated.



Whilst we will undertake delivery type roles, we typically look to provide advisory services, and in many cases, our Test Programme Managers/Directors are providing consultative steer and counsel whilst undertaking the delivery of the programmes for which they are responsible.

Our approach to finding solutions for each engagement we undertake is a simple one - we listen to the clients needs and issues, think about what is required, and draw upon our experience as practitioners. It's important to me that we do this rather than employ an off the shelf solution as I feel that so many players in the marketplace re-badge solutions time and time again. Notwithstanding this, we can obviously employ recognised processes and procedures when required, but I think it's a compliment that more often than not, our clients expect us to provide a bespoke approach and solution.

Another key principle is that we remain agnostic at all times. It's important that the solutions we provide and methodologies we employ best fit the clients needs. We have been engaged on numerous occasions by clients to audit programmes being delivered by Infosys, and I think it's a great compliment that we have constantly received feedback from our clients indicating that our findings and recommendations show no bias or leanings to any organisation, tool-set or methodology whatsoever.

---

### **OZTester: What do you think makes a Test Manager or Analyst come to work for Infosys Lodestone?**

It's been mentioned to me numerous times that Infosys Lodestone employ a very discerning approach to recruiting for its Testing Practice. I'm starting to get more of a sense that people know who we are and what we stand for now that we're well into our second year of existence, which is a positive sign, but also helps me better get a grasp of who really 'gets' what we are about. I think that we offer an outstanding opportunity for consultants to 'own' a real piece of our growing practice. I've been encouraging my team to continually examine the testing marketplace and table suggestions on how we can collectively grow specialised service offerings - mini practices within the overall Testing Practice per se - outlining service provision, opportunity and broader sales strategy, target resources and revenue

goals. It excites me that members of the team are now really showing growth in the way they think about and do business on behalf of Infosys Lodestone's Testing Practice, and are excited to seize such opportunities.

I'm an avid believer of avoiding micro-managing my team at all costs. I'm confident that as senior consultants in the Australian marketplace, they will pipe up if they need help, and have things soundly covered if not. As a group we all get together frequently for practice meetings to share news on progress, wins, losses and views, and we take every opportunity to learn from one another whenever possible.

Each of my team is tasked with speaking at an industry event, writing a paper and collating the IP the produce, as we continue to grow our brand. Additionally Infosys Lodestone supports its staff members wholeheartedly in terms of attaining professional qualification, which is a great way of evidencing its investment in its people.

---

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

It's encouraging that Australian testers have many opportunities to attend SIG's, major conferences and smaller workshop type mini conferences. High profile Testing leaders such as James Bach and Rex Black have spoken in the last few years at conferences here, so we all have a chance to listen to and learn from respected practitioners who are at the forefront of testing as a recognised and skilled capability.

I also think that industry wide, testers are now being encouraged and are indeed finding ways to work smarter, more efficiently and to think outside the box to deliver better outcomes. In the past, following the tried and tested route religiously seemed to be the go, but nowadays, using the tried and tested as a basis for delivering high quality testing, whilst looking for optimum ways to reduce risk, time and cost on a continuous improvement basis, seems to be more en vogue.

---

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

I don't necessarily see the issues that currently impact testing here as being specific to Australia. I think that wherever you work in the world, testers

as a group will face challenges around time, budget and unrealistic expectation from stakeholders. This all then leads us into the worlds of communication, influencing skills, planning/estimation and metrics collation, which over time, will slowly but surely provide us with the chance to redress the balance of expectation more equitably between stakeholders and the testing community.

Oftentimes, I think that soft skills can be overlooked, as it's hugely important to be able to 'sell' testing and its requisite necessities to decision makers, and indeed, other stakeholder groups such as Analysts and Developers, with whom it is so important to work collaboratively and effectively.

---

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

Most certainly! When I first arrived in Australia from London almost 10 years ago, it was clear that some of the practices and technologies I'd been part of in the UK were some way ahead of what was going on here in Australia. I was fortunate in a previous role to have played a significant part in delivering training to Testers across Australia, so I was lucky to have first hand experience in the improvement of fundamental and advanced testing skills on the ground.

Now as I consult to more and more organisations within Australia, it's becoming clear that significant investment is being made by them in the development of capability of their own resources. That said, I think opportunity still remains for consultants and specialist independent contractors to help further increase the standard of testing by imparting our knowledge and experience through our engagements with clients.

---

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

We are currently seeing that consulting in the Agile space is most in demand from our clients. However, this demand is for practitioners, who can help drive Agile Programme delivery from a testing standpoint - which is actually very encouraging to me as a career Tester. There's also demand for Mobility Testing and Cloud Testing consulting, and as has always been the case - Non Functional and Security Testing. I think these are market trends that will likely continue for the short to medium term, both in Australia and internationally.

At last year's IGNite conference in Sydney, Brett Rogers made some very interesting points about testing as a profession, and how he felt that there was a chance that we may soon become a dying breed. I'm a firm believer in the need for Testers to adapt and acquire new skills and experience, as organisations will without doubt look to do things quicker, leaner and more cost effectively. James Bach has been talking for many years now about how to test in a non traditional but thoughtful ad-hoc fashion.

Qualifications and experience in other areas is, in my view at least, a priceless commodity, which I would encourage everyone to try to do in order to maintain ones individual marketability and value proposition. We constantly need to adapt and acquire new skills, and it's fascinating to wonder what the 'next big thing' is.

---

**OZTester: Do you have a testing horror story to share?**

I'm sure most senior consultants do have at least one horror story - but regrettably, I'm not one to 'kiss and tell' .....

***Editor's comments:** Thanks for making time to write for us Matt. I do find the polarisation of testing thought in recent times to be quite interesting and have even endeavoured to bridge that gap with a few 'reconciliation' attempts. Regrettably those effort came to nought as it appears that some folk are so well and truly entrenched in their own ideas that the wider picture is often lost—never mind, all will come to a natural end point and some stage. - Ed.*

# Recognising and Dealing with Professional Manipulation

by **Geoff Horne**, Editor, **OZTester Magazine**

Have you ever had a sense that maybe you have been played, set up or manipulated? It's not a nice feeling once you finally realise that this is what might be happening to you. If you're wondering why I'm writing on this subject it's because, like most of us, I've had my share of allowing myself to be co-erced, cajoled, persuaded, guided or whatever this behaviour is disguised as, into thinking or making decisions that I really didn't want to make. I've probably also been guilty of putting others in the same position, albeit unwittingly.

Manipulative people endeavour to make you side with them or do something they want you to do that is to their benefit, without directly asking. Mild forms of manipulation can be found in every 'how to win friends and influence people' text ever written. Sales training courses outline similar approaches and teach persuasion tactics and 'closing' strategies to get people to make buying decisions. Not that there is anything illegal, immoral or otherwise about laying out information for others to make decisions upon however true manipulation has a far more sinister edge.

Manipulation uses guilt, shame, threats and other nasty tricks to achieve its aims and most of us have experienced some form of this emotional 'blackmail' at some stage on our journeys. We usually learn to spot it and deploy defensive mechanisms for our own personal safety and protection.

However when it's done in a professional context, there are far more subtle tricks out there that are not so easy to spot and we sometimes fall for these unknowingly. Such mechanisms are often presented to us in the guise of helping to improve our prospects and use professional exclusivity, career limiting, opportunity disappearances and other possible implications as a result of not following a particular train of thought, method or practice. In many instances the people concerned are not even aware that what they are doing is classed as manipulation, genuinely believing themselves to be acting in the best of interests. But then there are others....!

So where does it all start? We occasionally encounter people who have adopted rigid positions, whether around religion, politics, academia or any other particular thought disposition, where everything is seen through the filter of their own convictions. Not surprisingly, they see what they're looking for and the more they use their theories for making sense of things, the more things seem to fit those theories. The more that seems to fit, the more sure of themselves they become to a point where they cannot understand why others do not see things the way they do.<sup>1</sup>

<sup>1</sup> Hugh MacKay, Australian psychologist, sociologist, social researcher, writer and former teacher.





Consequently, there often comes a sense of entitlement to impose their perspectives upon everyone else. Some of these people though are smart and when doing so, they won't be in the least bit obvious. Instead they will use novel anecdotes, new twists on old adages or intellectual mind games to promote their world view. These come down to the old illusionist trick of getting you to concentrate on the primary object of focus and therefore not notice anything outside of that frame. Coupled with a dose of old-fashioned charm and the well-repeated message that it's improving everything for everyone, the trap is set. I remember once being collared by a lady selling cleaning products who demonstrated a cleaner by polishing up a very old, well-circulated NZ 2c piece (before inflation consigned that particular denomination to the past) – a canned presentation designed to validate her pitch. Maybe if I was polishing up NZ 2c pieces for a living, I might have bought it!

We often wonder how some people can be fooled into joining religious cults et al. However at the root of every cult is a charismatic and charming manipulator who is able to bend the wills of those who are i) so thirsty for knowledge and betterment and ii) so dissatisfied with their status quo, that they will allow their boundaries of mind, body and spirit to be contravened, almost as if by hypnosis.

In every profession, we find rigid-thought propagators who believe that they have developed the next level of practice, the new paradigm, a higher plain of ethics or professionalism et al. Out of a sense of misguided entitlement, they will see the need to press down on what they perceive as the lower plains in order to promote their message, all in the name of betterment.

They and those whom they are able to convince and gather around themselves may form a distinct sub-community and often a culture will develop that promotes the agenda, lending further credibility to the propagator's message. The sub-community may give itself a cool-sounding name or title which obviously aids in its promotion and its spread only serves to further confirm the propagator's supposed correctness. In more extreme examples, members of the sub-community may allow it to mould both their professional and personal identities – eg I am a this, I am a that. In professional circles, more than in cultic, if careful investigation is undertaken it often transpires that the propagator is significantly

benefitting financially from the spread of the 'movement'.

The difficulty is that there are nearly always elements of 'bonafidity' present. However they're ultimately enveloped by an infrastructure which has no real bearing on the validity of the practice and end up becoming secondary to the culture and the promotion of the sub-community that adheres to it.

Finally, exclusivity emerges and this is where things get interesting. Unfortunately in extreme cases, it has resulted in lawsuits, lost jobs and lost livelihoods. Parallels can be drawn with masonic lodges, 'secret handshakes' and general professional discrimination against people outside of the 'fellowship'. Examples of these are aplenty in the health profession and education systems. Multi-level marketing is riddled them and we also see presence in social programmes such as parenting, lifestyle enhancement and certain addiction exit initiatives.

So what to do? Learn to recognise such behaviours and structures then deploy strategies to establish reasonable boundaries.

Common traits of manipulative people:

- **Flattery and charm** – this will be poured on thick when the proponent believes it will help to ingratiate themselves with a person. If that person has been subject to unaddressed abuse, denigration, victimisation etc that flattery will be lapped up.
- **Threats and consequences** – these are usually well-veiled: comments along the lines of "well if you're happy to stay in that place then....." or "of course, our people enjoy a much higher level of ....."; words to the effect that you will lose out or be significantly disadvantaged by not following the line.
- **High IQ and Intelligence** – many manipulators are of well-above average intelligence. They often try to reduce their world-views to models then define the criteria upon which to base their discussions and persuasions. They 'win' every argument because they have defined the models from within which they operate.
- **Intellectualising and reasoning** – manipulators can make any argument sound

fair and reasonable especially if the listener has been disaffected in some manner. By getting you focus on a specific train of reasoning and convincing you that it's the only thing that matters, the manipulator can successfully execute the illusion.

- **Guilt and shame** – a manipulator has a knack of being able to turn the tables back on you when challenged and playing the guilt card: “well if you hadn’t been so....”, “maybe that could have been averted if you....”, “no, that’s OK it’s my fault that.....” etc. You may even find yourself apologising to them!
- **Fear and loathing** – manipulators can resort to demeaning, accusations, personal attacks, credibility slating and behind-the-back tactics. Swearing can be common as can shouting and ranting with the intent of creating fear and soliciting ultimate deference to the manipulator.
- **Self-righteousness and self-focus** – a manipulator has great difficulty in admitting when they have made mistakes and in accepting and articulating situations that might expose faults. In short, they are never wrong! If you want to break the flow of a manipulator, simply tell him he is wrong and see what happens!

These behaviours are tantamount to professional bullying and justified in the name of betterment in the same manner – I smack you around the head but it’s OK because my way is right and your’s is wrong. The worst part about it is that followers become accordingly convinced that it’s alright to behave in this manner and start to inflict the same on others.

What to do when you realise you have been hoodwinked in this manner:

- **Extract and distance** yourself from the manipulator, the sub-community culture and its followers. Do not attempt to fight or change perspectives, you will only frustrate yourself no end (this is the voice of experience speaking). The only thing you can change is your reaction to the behaviour and focus.
- **Understand** that you have fallen for an illusion and that the manipulation does not

represent a full reality and possibly only a very small part of it. Know that plenty of others have been taken in before you and there will be plenty more after.

- **Stand firm** on what you know based on your experiences and first-hand learnings. If need be, seek out trusted counsel. Do not allow a manipulator to take away your ability to make your own decisions, form your own opinions and follow your own path. Watch out for the illusionist syndrome and above all, maintain your professional confidences.
- **Knuckle down** into work with a renewed vigour, knowing now that you have an understanding of how and why these situations happen and how you will be better placed to handle in the future.

Work should be enjoyable and you should be surrounded by people with healthy mutual respect for each other regardless of alignment. Those who treat others with disdain because they differ professionally have little respect for anyone apart from themselves. If this was not true then there would be a reasonable co-existence, a viva la difference and not the continued opposition and exclusivity that we see in some circles.

There is always a bigger picture. In professional circles where the practices can be varied there will always be some who’s quest for betterment, no matter how honourable the intent, will fall into the trappings of manipulation and exclusivity. Once caught up it can be difficult to retreat back to a more inclusive perspective however recognising the hallmarks for what they are as opposed to how we may feel about them, may assist in withdrawing from these toxic dispositions and re-establish confidences to operate as effectively and as professionally as possible.

# Myths Around Skills Required For Automation Testers

by Min Xu, BankWest



Over the last six years, I have been fortunate to work on test automation transformation in a fast growing testing team within a large organisation. It is tremendous experience to be part of the team at Bankwest which won **2012 Australian Software Testing Awards – Best project** for our test automation framework. I often get asked about what is the most important factor for building large scale test automation transformation. If I have to point out one of many important factors, people, in this instance, automation testers is on the top of my list. Recruiting and building a team of automation testers with the right skills play a key role on the success of your transformation. But the question is, what skills are the “right” skills? Here. I would like to share with you: what does my “dream team member” look like; in reality, how do we bring testers into test automation and build them into different levels of automation testers; and lastly, how long should we expect the process to take?

## My “dream” automation tester meets these criteria:

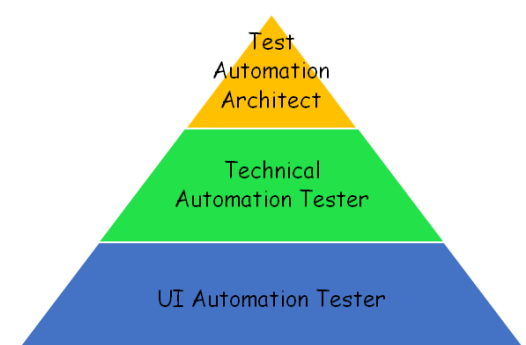
- First of all, automated testing is still testing activity ultimately, just executed using a tool or tools. So all the skills required for testing apply here. Good software testing is a challenging intellectual process. So critical thinking, being able to ask relevant questions, at the right time, in the right way are the most important skills for a tester, and for an automation tester as well.
- Automation testers need to understand the automation concepts and principles: what automated testing is for, when to automate and when not to automate, how to design good automated tests etc.
- Test automation is also a software development process itself. People who are doing test automation need to understand software development principles, especially

good development practice such as how to write clean code, perform code reviews, how to write them to minimise maintenance etc.

- Last but not the least, automation testers need to be able to collaborate closely with stakeholders (especially developers) and engage in relevant conversations with them eg. building testability in development and influencing developers to make test automation work easier, leveraging your product owner and business analyst to analyse and prioritise test automation coverage.

Unfortunately we are not living in Utopia. For those of you who have recruited for automation tester positions in the last few years, you might acknowledge how hard it is to find the right people with any one skill out of the list required above, let alone all of them. Simply put, we do not have so many “super men” or “super women” (as one of my development managers used to call them) available for us to choose and pick, especially if you are looking at an enterprise level test automation transformation. Even if you are lucky to find some (like I did), it would take a significant amount of time (I found three really strong automation testers in six months including hunting overseas a few years ago) so it is not scalable.

**So how do we as test lead/test manager bring testers onto the journey of automated testing? As testers, how can we progress our career in the dimension of test automation?**



I believe there are three different types of automation testers, divided by their levels of automated testing skills and experience (see above Pyramid of automation testers):

- **UI Automation Testers:** Quite often, we have a large portion of testers in the team who have strong system and business knowledge but little technical background. That is probably because a lot of good testers grow from business users. There are fantastic tools out there that allow the more 'traditional tester' to become productive quickly in this space and start the test automation journey. Typically the tools provide a learning curve for testers to move from the 'no scripting required' type settings to the full blown scripting style of UI automation testing. This experience is a great foundation to progress to the next level.
- **Technical Automation Testers:** Testers competent at the above level or with stronger technical background will quickly see themselves moving up. Depending on the architecture of your system under test, it is more often than not with modern development to build services and APIs (application programming interface) before any user interface is built. Whether it is technology such as JSON, SOAP or REST, we could look at testing functionality on the services or APIs. It means we can start testing earlier, test faster and with lower maintenance. More importantly, it will allow us to have targeted coverage on UI test automation such as usability testing. To do this type of testing, automation testers need to be able to understand system architecture, messaging schemas, integration points and how to carry out validation using some type of tools. To take this to another level, automation testers can even be involved during the development phase such as API design and start testing in parallel with development. Otherwise, for automation testers who have adequate technical knowledge, it is highly beneficial to get involved or even help developers on unit testing coverage. You will be able to identify areas or gaps of weakness in the product and design a more targeted testing and test automation strategy. If your organisation really puts quality at the fore front, you could also look at techniques such as Test Driven Development (TDD). In my

experience, a lot of UI automation testers transition to this level quite naturally where developers and testers can really work together.

- **Test Automation Architect:** Even if we are not living in Utopia, it is still very important to have someone in the team who does have the skills listed in the beginning of this article. At least one of your team members need to have technical skills on implementing all sorts of frameworks and tools; have holistic view on software development rather than tunnel vision on testing itself; have deep understanding of the test automation pyramid and can train, guide and coach the rest of your team in implementing test automation across different levels; have capability to work on more technical trouble shooting such as tool customisation and integration etc. Ideally this is your Test Automation Architect. It might be difficult to find or build but it's definitely worth the investment.

So the automation tester pyramid can hopefully provide some framework for the test lead/manager to motivate testers and drive the automation transformation, step by step. I also hope it provides some career path guidelines for testers in the team, if they are interested in getting onto the exciting automation journey.

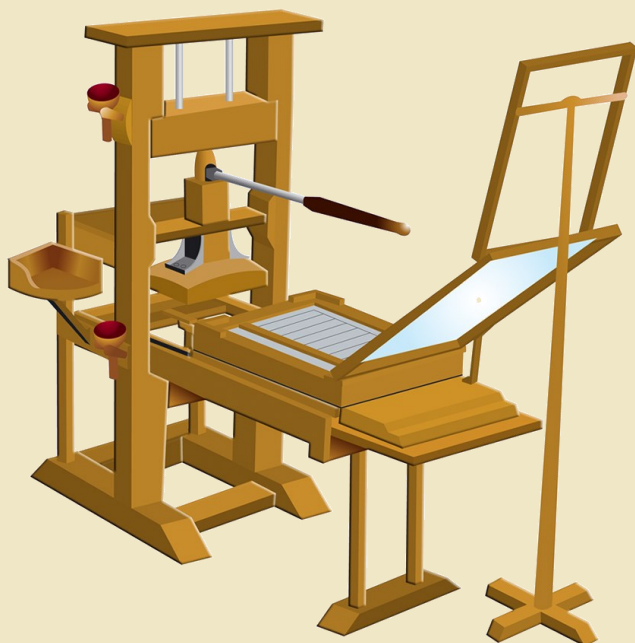
**Due to different backgrounds and speeds of learning, we will unavoidably see different speeds for uptake of automation skills in each individual.**

But as a general rule of thumb, I have found:

- It roughly takes 3-6 months to bring strong testers into UI test automation with adequate tools, training and support.
- It can take up to 1-2 years to build a strong Technical Automation Tester who is really productive in working closely with developers.
- A Test Automation Architect can take 3-5 years to build with right mindset and environment.

While test automation transformation does involve technical implementation and process establishment, it is largely a cultural shift of the team. Therefore it is important to engage our valuable testing colleagues, bring them along on the journey and equip them with the right skills.

Min Xu has been working in the testing industry for over 13 years with 7 years in test management positions. She has vast experience on test automation and performance testing in telecoms and financial services industries, both in Australia and overseas. In the last 4 years, she has been working hard on building the Automation Framework in Bankwest which won the 2012 Australia Software Testing Award - Best project and she has been nominated as one of the two finalists for Best Individual Contribution Award. She is very passionate about testing and test automation in particular and has been a regular speaker at testing conferences and events. Min can be contacted at [min.xu@bankwest.com.au](mailto:min.xu@bankwest.com.au)



## Wanna Get Published?

Our formula for selecting articles for publishing:

**Good + Relevant = We'll Print It** (well, digitally-speaking anyway)

**Good** = *one or more of*: thought-provoking, well-articulated, challenging, experienced-based, technical skill-based, different perspective to mainstream, unique....

**Relevant** = *one or more of*: emerging trends, new technology/methodology, controversial (within reason), beyond the basics (eg. testing is good, defects are bad)....



# Ever Had That Sinking Feeling?

by OZTester Magazine Staff Writer

Ever had that sinking feeling? You know the one, when you walk into the office on a Monday morning and your colleagues are all grim-faced. You figure that their weekend for some reason didn't go so well then you remember that during it, some of them were in working, installing that big patch release that you were testing last week. Your heart jumps into your mouth; what's happened, what did I miss, did I cover everything etc. etc. and very soon, you find out that...oops!

The Business Analyst demonstrates the issue and you know you had covered this-this-and-this, this-this-and-that, this-that-and-this, that-this-and-this, that-that-and-this, this-that-and-that, that-that-and-that BUT did you cover that-this-and-that? And of course the user logged onto the new release, entered that-this-and-that and...kaboom!

The Test Manager reminds you that you had two days to test this feature and two days was (probably) sufficient. And even if it wasn't, the risk was understood in order to get the release out to the user, who of course needed it yesterday. Plus everyone knows that curly, complex issues do arise however as you are again reminded, this error was a simple problem that any tester worth his salt should have found within the first hour or so of testing. You feel inadequate as the Product Manager walks passed, eyes diverted and seemingly emotionless although if his thoughts were any more obvious, he'd be in counselling and so would you! No-one else says anything yet you know you've let the team down or at least that's the way it feels. I mean, two days really was enough wasn't it and the problem was so simple after all and....

Does this ring a bell or twenty? Ever been here? It's not nice. However before you take a stroll down Harakiri Lane, there's a few things you might like to be aware of...

Everyone sees even the most simple of things, at least slightly, differently. It isn't possible to look through someone else's eyes as much as it would be useful sometimes. We can really only imagine how others see and think, and the better we can imagine then the more we can empathise. Take a coffee mug that is red on one side, blue on the other and place it on a table between two people then ask what colour the mug is; one will swear that its red, the other blue. Add just a dash of emotion and ego in there and hey presto, instant argument and empathy strangely apparent by its absence.

Someone else's perspectives and interpretations can tell a different story yet we mere humans seem to love developing and following procedures and processes that so often result in subjectivity being ironed out. Creativity, innovation, experimentation et al seem to get flattened in the course of process institutionalisation. In the example above, there was two days of testing performed however the reality is that even if two weeks of testing was done, there's no guarantee that any or all of the defects would have been found. Remember that a test



result is just a snapshot in time; at that specific time, in that environment, with that data, on that version, with that configuration, in that manner... 'this', is what happened (or was it 'that'?).

Different perspectives and interpretations can also lead to quite different outcomes. Again in the example above, someone else may have found the error that was missed and still someone else may not have come across the errors that *were* found. That's why the more we test, the more defects we'll find and if we test for two days, that's what we get: two days-worth of testing and defects. If anyone asks if testing complete/finished, you might want to ask for a definition of "complete/finished" (tip: you might need to excuse yourself for being a smart alec first). If that means have we covered all the requirements specified, it's relatively easy to measure as long as a traceability matrix of some sort is maintained. Does this mean all the defects have been found, no. Does it mean that we've tested everything that needs to be tested, no. It just means that in this time, we covered this functionality and found these defects.

Different approaches will yield different results. Let's have a look at an example: a while back, I had a smallish enhancement to test for a web-based application (after having been asked to take time out from my usual test automation responsibility). Due to a slight miscommunication, one of my colleagues, Murali, also ended up assigned to the task. However given that we did literally have two days to test, we decided that four eyes were better than two (and no, it wasn't merely a case of me putting on glasses!). Murali went to the specification and spent the first day developing manual test scripts to cover each requirement. He then spent the second day executing those test scripts. I, on the other hand, decided to go exploratory and spent the first day experimenting, investigating and generally playing around with the web pages. On the second day, I developed a bunch of complex-type scenarios (note: not manually scripted, just defined). I recorded these using an automation harness then, so I could define my this-this-and-that's, I developed a .txt data file that was read by the harness and the details spat into the application. It all worked a treat and Murali and I both found some "good" defects however, guess what...

Only about 20% of our defects were duplicates. Murali's scripted approach and my investigative method had both been successful in that we found

errors. Now, if Murali had been asked if had he finished testing, he could justifiably respond that yes, he had covered off the requirements as specified and the defects he found had been fixed and retested. However my defects would have still been there.

I could also respond to the same question by saying yes, I had been through each page and field in the application and applied the usual bunch of tricks eg. field boundaries, maximum lengths, positive/negative conditions, if-I-do-this-what-happens etc. I could also say that I had combined a whole testing magazine-full of complex scenarios executing together and based on running those yes, I had finished my testing. However Murali's defects would have still been there.

The real answer is that we finished two lots of two days-worth of testing using our respective approaches. And as fate would have it, the release went to the client who promptly found a this-that-and-this! Upon investigation it was found that the failed feature had been covered earlier on in the two day cycle by both Murali and myself. So either we had both missed the defect or something further down the line had changed, mostly likely another fix but, who can say? Had we performed more testing, we'd (probably) have found it but again, who can say? However the perception was that we'd finished testing and bang, client finds a defect straight away! Doesn't exactly spell "professional" in the eyes of those peering in from the outside, does it?

So, there are a few lessons here:

- The more we test, the more defects we will find. We'll probably never find them all however if we can track their frequency of discovery (by severity or some other appropriate categorisation) then we can gain a rough indication of when it might be OK to wrap testing eg. when continued testing yields no further high severity defects for a specific period of time; the defect frequency reduces to zip and the lower severities to a trickle. This approach also helps to avoid the 'last minute fix' scenario where final, rushed fixes break all and sundry, effectively winding the quality clock backwards.
- Different testers will find different issues. I know from experience that I'm reasonably adept at finding the curly issue that no-one

else will find yet I can sometimes miss the obvious. For others it's vice versa. Both are needed.

- Completing testing doesn't mean that everything has been tested or that all defects have been found and fixed. It may mean that we've done what we set out to do or otherwise, but that's all.
- Varying the approaches to testing can pay dividends; the experiences above indicate that multi-pronged methods can be useful. Both the experimental and the predestined approaches work depending on your context however doing both; now there's a novel approach!
- Pair testing can pay dividends; another set of eyes may see what yours do not. Peer testing can pay dividends; a fresh set of eyes may see what tired eyes do not. Commercial realities can sometimes preclude however this is not simply a case of two doing the work of one. Think crowd-sourcing here!
- It's also a cruel fact of life that things beyond our control can change, yes it's true. I have had many situations where a release has been passed through for implementation, someone tweaks a parameter or updates a utility or something, the release goes in and.....bang! What can you do? Testers should not be held responsible for those elements beyond their control. It's an unfortunate misconception that because testing is so often considered the last link in the chain that it's the catch-all.
- The outcome of testing is always....information (I really don't know how many times I've said that over the years); information about the state of the product or system-under-test. However it's a fact of life that we testers will always be asked whether we believe a product is ready to go or not, so we have to be succinct eg. "we specifically tested for these situations" and "we did not specifically test for those situations" (tip: have a good reason as to why); avoid the "we don't know", "we think so" etc. And your stress levels will go down as a result....promise! Resisting the temptation to merely say or even imply what others want to

hear, will ultimately be to everyone's advantage even if it doesn't appear so at first.

In summary, this testing profession of ours comes with all sorts of lessons to be learned. Every journey we undertake within it will always reveal more and adding these to our arsenal of sneaky testing tricks only serves to improve the overall positive impact we can have on software development, and IT in general. If we stop learning then we might as well retire and as our experiences from these lessons grow, the more valuable our "tester's nose" becomes. Test on, people!



# TEST YOUR WORLD WITH 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*



**SAVE TIME**



**GET MORE**



**BE EMPOWERED**

Enterprise Tester is chosen by:



[catchsoftware.com](http://catchsoftware.com)



# Combined Acceptance Testing— CAT Synopsis

by Mark Taubert, National Australia Bank



*Combining several test phases into a single test phase to enable accelerated delivery and cost reduction for an IT Program or Project.*

The use of CAT as a strategy for testing a program/project has evolved out of the need to deliver projects within fixed timeframes (e.g. Compliance Programs/Projects) and to suit quarterly production release cycles. CAT becomes fundamental as an approach when time, quality and cost are vital to the success of program/project delivery. Quarterly release cycles limit the time a program/project has to deliver in so it's essential that efficient methods are utilised to ensure success. By combining standard functional test phases into a "**Combined Acceptance Test**" phase risks can be mitigated to allow for on time delivery.

Standard functional test phases such as (System Test, System Integration Test & User Acceptance Test) can be merged to reduce testing timeframes and therefore costs while maintaining quality assurance.

The approach required for CAT to be used as a successful testing methodology is to rethink how the standard functional testing phases can be reorganised to run in single test phase. The key to success is understanding the key technical and business functions/transactions and convert them into test scenarios which cover all standard functional testing phases. The test conditions are categorised into (4) test cases functions.

1. **E2E Test Cases:** These test cases are constructed to cover the main transaction and paths to ensure interactions between applications are working as designed. The test cases may have steps that refer to "Master Test Cases" and should also cover the traditional System Test and System Integrations Test scope.
2. **Business Use Cases:** These test cases are constructed to cover all the business functions

simulating day in the life test scenarios. The test cases may have steps that refer to "Master Test Cases and E2E Test Cases" and should also cover the traditional System Integration and User Acceptance Test scope.

3. **Master Test Cases:** These test cases are constructed to cover transactions and interactions that are common (e.g. User login procedure). These test cases are referred to in the test steps of the E2E Test Cases or the Business Test Cases. The benefit being reduced test preparation time.
4. **Regression Test Cases:** A comprehensive suite of regression test cases (Manual and Automated) that can be run frequently to stabilise the code base. This test suite should contain the current production code base plus any changes in scope for the Program/Project.
5. **Quality Assurance:** CAT quality is achieved through early testing of input documents, stage gates, and metrics governance. Quality assurance metrics are captured throughout all test phases to produce a final quality score card

## Key Success Factors

- Early engagement of the operations and business teams to provide input, support and approval of the CAT approach. The program/projects goal is to bring all stakeholders along for the journey where "everyone knows the plan."
- Experienced test team who are able to guide and support business users and change management teams through the CAT test phase. The test team's role is to clear the path for the business and change teams to come in and validate new and existing functionality.



## Combined Acceptance Testing

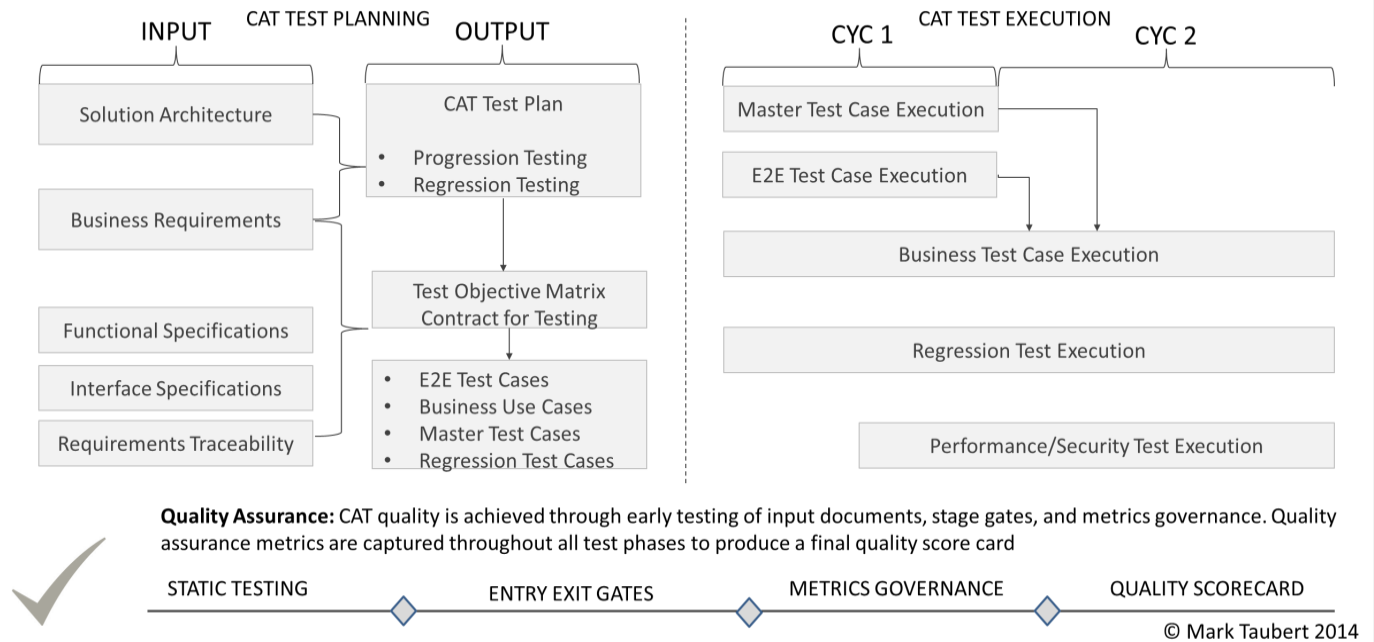
Combining several test phases into a single test phase to enable accelerated delivery and cost reduction for an IT Program or Project.



Early engagement of the operations and business teams to provide input, support and approval of the CAT approach.



The test team's role is to clear the path for the business and change teams to come in and validate new and existing functionality.



- Approved or base lined input documentation (Business Requirements, Functional Specifications, Interface Specifications & Requirement Traceability Matrix) that can be used to formulate test scope and coverage in agreement with all stakeholders. All changes to requirements after the approved/ base lined point in time should be managed as a "Change Request" for full traceability.
- Onsite, Offsite and Automation capability to provide extended testing hours over and above standard business hours of operation. The ideal scenario would be to build a 24 hours test capability by utilising extended testing hours but anything more than an 8 hour business day would be an advantage.

throughout the CAT planning and test execution timeframe.

- The CAT approach allows for rapid stabilisation of the business requirements and code base through the construction of the test cases and extended test operating hours.
- Allows the Change team to be part test execution and allow the documentation of business process prior to implementation Go Live.

### Challenges:

- Requires an experienced test team to support business and change team members. May prove difficult to build and experienced team in some vendor resourced test engagements. Experienced test resources maybe more costly than some vendor resourced test engagements.
- Defect density maybe higher using a CAT approach due to combining test phases into a single phase. Root cause analysis may prove more difficult due to the lack of discreet test phases which may impact test execution progress in the initial stages of CAT execution.

## Benefits and Challenges

### Benefits:

- Allows Programs/Projects to meet aggressive timeframes while still covering all standard test phases delivered through standard methodologies.
- Gives all stakeholders and opportunity to be involved in the early stages of the Programs/ Project to agree test scope and to participate

- Business Requirements need to document the current business process and future business process before testing is complete. There may be a challenge in doing this when CAT testing is in flight in such tight time frames.

## **Conclusion**

Combined Acceptance Testing as a strategy and approach is advantageous where there are hard deadline dates or cost constraints. The average cost of testing for a program/project is around 25% of the budget. Using CAT approach to deliver testing should reduce overall testing costs through the rapid delivery of testing by running combined test phases. CAT has similar advantages to the Agile methodology where testing is run in parallel test phases or sprints to include functional and regression testing to reduce time frames and costs. The rapid delivery approach allows a business to implement changes more efficiently and bring greater flexibility and functionality to a business.

**Mark Taubert is a Programme Test Manager at NAB in Sydney. He can be contacted on [mark.taubert@nab.com.au](mailto:mark.taubert@nab.com.au)**

# TestAnalytics = TestIntelligence

Hidden in the depths of your test repositories and not even accessible by your management tools is testing gold!

What gold you may well ask? Well, for example, your project might only be wanting to know that all the test cases have passed and all the defects are fixed, in other words, whether the completion criteria have been met. However what if on the Friday before your Monday go-live, the team found and fixed 10 Severity 1 defects? Completion criteria would still be met however what would your confidence be like and how would you communicate it? Better still, how could you pre-empt the situation?

The gold is information about your product-under-test beyond the garden-variety and vanilla-flavoured. And it's almost certainly sitting in your test repository or database yet your management tools won't know its there or even what it is, because they only count things.

Want to get it out and use it? We can help.

We've developed a set of tools and services to get this gold out from its hidden recesses, onto paper and into the hands and before the eyes of those needing it.

Click below to find out more!

## OZTesterMagazine TestAnalytics

Find the gold and convert to cash!



Software Education Group

## Press Release

5 November 2014

For immediate release

### **Discover becomes a part of the Software Education Training Group**

The Software Education Group has acquired Melbourne-based training company Discover to provide a combined Software Testing curriculum for 2015.

Software Education is internationally recognised for providing comprehensive courses in software development training. The Australasian-based company today announced that Discover would become part of the Software Education Group.

Discover is a top Australian specialist Software Testing training company with an industry-leading pass rate in International Software Testing Qualifications Board (ISTQB) certified training.

The purchase of Discover further increases Software Education's share in the software testing market through the expansion of training capabilities.

Toby Thompson, Founder and Managing Director of Discover, will become the Software Testing Practice Lead for the Software Education Group and had this to say about the acquisition:

"Our focus will remain on delivering the highest quality software testing courses. In the current market Software Testers and software project team members alike require multidisciplinary skills to work effectively in cross-functional teams. The combination of offerings will allow our customers' the opportunity to leverage a suite of courses with improved depth of coverage in 2015"

Discover will continue to operate under the same name as a specialised brand of the Software Education Group. A combined curriculum of Software Testing courses will be available in early January 2015.

Managing Director of the Software Education Group, Martyn Jones is excited about the possibilities that a partnership with Discover will provide for the future saying that:

"We are thrilled about having Toby and Discover on-board with the Software Education Group, increasing our software testing practice to include more publicly scheduled courses across Australasia as well as new course offerings, providing more options for our customers".

Ends.

## **About Software Education**

Established in 1990 Software Education is an independent software development company offering training and consultancy services across all sectors of the software development life cycle. SoftEd provides world-class training for software development teams in Australia, NZ, USA, India, Saudi Arabia, Canada and Singapore. SoftEd's mission is to provide customers access to leading-edge content and connect clients with an unrivalled network of international software development experts.

Website: [www.softed.com](http://www.softed.com)

## **About Disqover**

Melbourne-based company Disqover are a highly-regarded, specialist software testing training company offering a range of ISTQB certified courses. Disqover is fully accredited by the Australia and New Zealand Testing Board (ANZTB) and all trainers have been approved by the ANZTB to deliver International Software Testing Qualifications Board (ISTQB) certified courses. The company's wide range of training solutions address the issue of software quality in large public and private sector software development groups. Disqover was founded by Toby Thompson and has a strong reputation in the Australian marketplace for providing high quality courseware and excellent customer service. Disqover offer their courses publicly throughout Australia as well as in-house.

Website: [www.disqover.com.au](http://www.disqover.com.au)

**For more information or further comment please don't hesitate to contact**

Martyn Jones

Managing Director

The Software Education Group

+64 21 590 254

[martynj@softed.com](mailto:martynj@softed.com)





## Test Summary Reports

by Heather Liney, Telstra

I want to talk today about an area of particular interest to me, and that is the quality of Test Summary Reports. By whatever name you call them, they are the Reports produced at the end of a Testing Phase or effort to collect together what was planned and then what happened, to recommend and explain those recommendations, to reflect and clarify, to create a worthwhile artefact.

What is an 'Artefact' exactly in this context? It is a document of value, for now and in the future. It should be thought of in the same terms that we reserve for archaeological digs or forensic accounting - one day someone might use these to discover what happened here. In the very close future it may be reviewed to not repeat the mistakes and/or to learn the outcomes of good strategy. If there is no value or quality in the document to start with then it is doubtful it will occur in future- how can it? If you dig up pottery and it is all pottery, then it is not such an interesting story as an item of gold. As the writer, you are the craftsperson of this work- are you making plentiful pottery or striving for gold?

Forget the future, what about the now? What are the elements of a TSR that make it a worthwhile artefact and not dross. Let me address this in sequence of the document layout. Yours may vary a bit but all the main elements should be there.

The document purpose and scope should contain basic information that lets the reader know they are reading about the exact project, the precise phase and the size and outline of that phase. This must be accurate as there may be a multitude of phases to keep track of, and scope can become vague- be sure to start as you mean to continue, with robust statements and minimal flim flam. It may help to align this part with the Test Strategy and Test Plan, didn't they also explain what was to be tested and what wasn't? So any variance can be noted. Explain also to your audience in general terms what will be

outlined in the document to come: the test results, the outstanding incidents, the conclusion and assessment of risk. Try to engage your reader, a well written TSR is a story that unfolds and so make sure your reader knows they are going to be looked after with well thought out sentences and minimal jargon.

Usually there is standard report aspects at the start of the document about who is reviewing, the assumptions made by the Project, Roles and Responsibilities. Filling these out fully will let the reader know this is a complete document and not a work in progress.

The Executive Summary at this point is designed for the reader, not the writer- do not jump straight in and write the whole thing, never to be revised. Always come back at the end of the report to make sure the Summary truly reflects all that is to follow. It needs to be short, clear and its conclusions obvious, but not without personality and never as a "white wash". In reviewing Vendors' and internal TSR's over the years I must say the biggest red flag to me is the perfect report of 100% perfect results written in a bland standard boilerplate. I advise you also as a reviewer to BEWARE!

If you got 100% of your tests passing in one run- you did not test far enough, or the right thing- you tested only what would work, and didn't think of other paths or boundaries. Maybe you lied. I am looking at you with deep suspicion now, and will be inclined to look very carefully at the whole test effort. Maybe you don't care about your report- well maybe that extends to testing as well, and I am now assessing the risk you present me with. It's amazing to me that vendors would disregard this as a selling tool for their company, proof of the product, a showcase moment. Instead I get a bland report that doesn't warrant writing- for it means nothing in a very careful way.

The other extreme is the op ed piece- the Jeremy Clarkson of reviews in which the product is fiercely promoted or slammed as the greatest/worst idea since sliced bread and not a word can be said in it's detriment/defence. I was on a project where we read the draft and were less than impressed by the suggestion that the code be "re-written from scratch" to improve the poor performance. The PM asked me to re-write it as the previous author "hasn't written one before (!), and the Sponsor has just seen it, and all hell has broken loose". Try changing "this thing won't even perform in normal conditions" to "the product did not perform as expected during time trials" with a straight face.

Find good examples of reports or writing styles and adapt them for your use, encourage the team to write honest and complete reports, and coach and mentor so this is embedded. It is a life-long skill even with Twitter and SMS throwing style out the window- good written expression shows through even if you type, which is why you love particular authors and not others. If you have to write things, it is worth working on to find your style and best method to do so. I was lucky to have some early coaching from a Manager who was sure we would all be sued, so every line of the report had to stand up and be proved by the data. It tightened up the motherhood statements and really read well. And you need to find out how best you write- it is a focussed and concentrated exercise where you can get into the zone. For me it is two-three blocked out hours with Prodigy in the headphones.

On to the rest of the report, and this covers the whole story of what was tested, how, the results, the outstanding defects, the analysis and other relevant aspects such as the original strategy. These should be traceable and referenced to actual artefacts and test tools, and they should be factual with relevant changes from plan outlined. We want a well informed audience so don't expect people to guess or miss out areas. Hyperlinking graphs to other parts of the report are helpful, or referencing. Our Conclusions are our Risk assessment and can include the Exit Criteria and tables or charts that prove our point. By all means highlight assumptions being made, by all means flag risks- and refer to the proof of them. If testing was interrupted, changed focus, key personnel walked, technology didn't work- this should all be the proof of your reasoning to the risk for the business, for the project for interested

stakeholders. It's a tricky balance to avoid the political, to maintain the testing creed of independence yet to stress the problems correctly to the audience. That's why it is a craft to write these, not an exercise in boredom.

Apart from sign-off, references, version control and appendix, this should be about it for your Test Summary Report. Now go back and review that Executive Summary to make sure it all hangs together in a clear and concise way. What did you conclude when you read the report- and is that what you want your audience to think? Maybe a colleague will read it for you to verify the expectation, and of course you are putting it out for review- there could be some changes to come to your masterpiece, so accept that it may not be perfect, but nothing is. However the satisfaction is in reflecting the team's work and commitment, the projects' journey and a reasonably agreed outcome that explains the situation today. If you can achieve that it is a key skill to add to your kit.

I started by talking about the Test Summary Report as an artefact and I want to conclude on that theme. It seems anachronistic in this fast paced world where we move on before we finish and dynamic means chaos. Who bothers to write these days and is it just too absurd to bother to do it well? While there are still TSR's and Plans and Strategies that are expected to be written and provided, then why not do it in a meaningful way that can later be recalled and traced to the work that you and your teams did in an organisation at a point in time. It is not a building or a Policy that was rolled out to millions but it was important for the people involved and the effort should be recorded through the work that was done. It is one of very few documents that will be considered to reflect the project. It is something you can take pride in writing, and it deserves to be a thing of quality, just as the project hopefully strived towards, because why else spend your time on it if not to try and make things better than they were.

**Heather Liney has a long and happy history in projects and testing and is currently back in Telstra missing writing Test Summary Reports. She can be contacted at [heather.liney@telstra.com.au](mailto:heather.liney@telstra.com.au)**



# The Often Untapped Value of Business End-User Testing - Part 1

by Tim Fox, Independent Test Consultant

## UAT Observations

Approximately 70% of my experience in the ICT testing industry has been in Federal Government sector – in which (despite best efforts from the Agile movement), stricter waterfall-type methodologies remain dominant for new/major ICT releases.

Whilst my specific testing assignment has varied, I have consistently observed the user acceptance testing (UAT) phase being an unfortunately dull and providing low returns on investment within the software development lifecycle (SDLC).

Over the years, I've often struggled to hide my disappointment with how the end-user stakeholders who look forward to their project contribution involvement in the project are engaged with one or more of the following approaches;

- Superficial, unstructured... use this system and tell us what you think
- Micro-managed, blinkered...pass or fail these test cases (which we've already run)
- Requirement driven... accept or reject this system (within the scope of this signed agreement)

## So, what are you going to do about it?

As fate would have it, I was recently engaged to test manage 'business experience and assurance testing' (BEAT) for a major Federal Government ICT delivery programme. The experience has been quite extraordinary for all involved, providing benefits well beyond the traditional UAT goal of delivery acceptance.

## Delivery success = "We can't screw this up!"

The level of importance that the project team place upon end-user satisfaction varies significantly within ICT project environments. As a test manager, I consider it important to understand (and appreciate)

is how delivery success is defined, although the definition of failure (and the avoidance of it) is often more motivating.

My new project's executive group and industry stakeholders certainly knew what *failure* looked like. Sufficiently similar/recent ICT projects had caused temporary shut downs within industry whilst system and business process issues were resolved. "We can't screw this up, like XYZ did" became the answer to questions regarding 'delivery success' and 'quality determination'.

## How is the project tracking, as is?

I was engaged towards the end of the project's last delivery iterations, just as system testing was recommending acceptance of the ICT delivery package.

Following investigation into the project's artefacts, engagements, delivery and quality methodologies, my gut feel was that the project's movement into Production was being determined by;

- An over-reliance on documented requirement/design specification correctness, and
- Testing services being almost entirely verification based (to the exclusion of validation).

In summary, I was working on a project that had a high likelihood risk of producing a 'fit to spec, but not fit for business' ICT delivery. Project sponsors were not receiving any visibility of the system's impact to business nor having their transition/adoption dependencies identified.

Sound familiar?

Tim Fox is an independent applications testing professional, based in Canberra, Australia. He can be contacted at [tim.fox@rbd solutions.com.au](mailto:tim.fox@rbd solutions.com.au)



# The Pragmatism of Test Estimation

by Rajesh Mathur, Government of Victoria

There are hundreds of approaches and methods available for test estimation. Some of them are very theoretical in nature and do not fit in a day-to-day project delivery environment; others are too casual and do not satisfy the need either. Software projects are 'living creatures' in nature and every change affects cost & schedule. Due to this dynamic nature of software projects, it is vital that estimation is done in a way that it becomes practical and pragmatic for the project.

My work on software projects during the last two decades has taught me that software estimation is not an easy task. In the actual fact, it is very difficult or even close to impossible. Close to impossible because no one can see the future, we can only guess. We can only imagine about how long it will take to test. Attaching a number on imagination is not good.

Test estimation is more difficult because testing is a cognitive process. Basing it on requirements is hard. Project sponsors ask for estimates for Initiation or Conceptualization phase. "Give us a number +/- 100%". Hang on, how do you know that it will not be +/-1000% or +/- 10% of the given estimate? Requirements are often vague, or ambiguous or both. So if I estimate on wrong requirements, the estimate will be wrong, isn't it?

But business owners still want an estimate and in certain circumstances it is possible to give a decent estimate. Often in the situation when you are doing exactly what you did before plus buffer for external factors that you cannot control.

I am following the #NoEstimates movement but I am not a follower of the movement. Think it this way. You want to build a house. You approach a builder/ contractor and describe your requirements to them. Your requirements are still a wish list and somewhat vague. It is possible that some of them are not doable, out of your budget or the builder

does not have capability to do them. But that is okay because you have a budget and you want something done within that budget. You ask the builder how long will they take to build the house and they reply that they belong to #NoEstimates movement and will not tell you how much it will cost. You will agree with it only if you have won the Australian lotto or you are the heir of a massive fortune with a buffer more than your original budget.

Business owners want 'some' estimates. For software it is not entirely impossible to not estimate. Business needs it or they do not need you. Simple! So what to do to estimate for testing. One has to be aware of multiple aspects and dynamism of project delivery methodologies, people working on projects, organizational culture, nature of software solution, other supporting application like upstream, downstream, front end or backend applications, time constraints, dependencies on other external factors etc.

One thing that testers must keep in mind while estimating is that estimates are 'just estimates'. I have seen project managers asking for 'accurate estimates', which is nothing more than a fallacy. If it is accurate, then it is not an estimate. My advice is to provide three different numbers to projects; optimistic, pessimistic and realistic.

Realistic are those number that you derive after studying and analyzing requirements, defining a clear cut scope of your tests, referring to architectural designs, viewing and skimming through any other supportive documents that you feel would be worth reading for estimation, and by considering all the external factors I mentioned earlier for including in your estimate.

Pessimistic estimates are those that are derived after a discussion with project manager, architects and business analysts. I would recommend a collaborative approach for these discussions. This



will also save you from politics that often follows once projects start getting pear shaped. It is also always worthwhile educating your non-testing team members of testing approach if they are not aware of it. Most non-testers I have met believe that testing is easy and all we do is to ensure that software is working fine. My response to these people has been consistent that my job is not to make sure software is working fine; rather it is to discover that in what conditions or situations it does not work. What approaches or techniques I use for this discovery depends on the context. Another thing that I consistently mention is that I am not the one who ensures quality of software. I only show where are the issues. Quality Assurance is a broader term that is project team's responsibility not just testers'.

Optimistic estimates provide an option to the project management to show that there is a possibility that testing might finish earlier than expected. See, I always make it very clear that complete testing is a myth. If this was not a myth, all those giant software firms out there would not have kept their testing teams after the first release of their software. They would not also need to release patches of bug fixes. There is a very good article by Doug Hoffman which talks about completeness of testing. If you have done #BBST course you will know what I am talking about. If not, you must visit [www.associationofsoftwaretesting.org](http://www.associationofsoftwaretesting.org) to find out.

The Optimistic estimates are based on a simple fact which is the Exit Criteria of your testing approach that states that if certain conditions meet, no further testing will be performed. Again, this totally depends on the context of the project.

In another context, your manager or client may give you a schedule or time to complete your testing. In this case you don't really have to estimate.

As a tester you must understand that while test estimation is required to define a solution delivery timeline, there is no control over what drivers are pushing your business teams or the client. If the business is risk-averse, the chances are that you will get what you have asked for. If the risk-appetite is high (or business teams has no clue what they are doing), product will be shipped without much testing. In this case they are probably looking for a scapegoat or a box-ticker.

Whatever I mention here in terms of terminology, is purely context-driven. I do not follow so-called

standards of testing and I do not want you to comment about a certain term referring to ISTQB or any other glossary of terms. I have my own views about testing and I believe I have not harmed any of my projects by using or not-using a specific set of standards.

If you are interested in learning or discussing more about test estimation, you may contact me. Or, if learning is the objective, you must look at Jerry Weinberg's book 'Perfect Software' and blog posts by Michael Bolton on Developsense blog.

**Rajesh Mathur is Head of Testing at Government of Victoria. He has held senior test management positions with Cathay Pacific, Nokia and Steria Group amongst others.**



# To QA or To Not QA: That Is the Question!

by Rowly Emmett, Access Testing



As testers, we get called the quality assurance or the QA team, yet by limiting our role to merely dynamic testing of a product we're just quality assessment; we provide information as to how good . . . or how bad, an application is.

We all know that requirements are often bad, yet we accept it and deal with the problem only when it gets to testing. Different process methodologies could be employed, ensuring all those within the project clearly understand the requirements and the requirements are clear, however too often the project has already started without us, and changing the process for a project halfway through would be mutinous.

How about we assert ourselves and truly embraced quality assurance?

Taking on those who have written the requirements puts you in the firing line, but I believe this can be the cheapest way of ensuring project delivery in terms of both budget, timelines and most importantly addressing end-user satisfaction.

If it's wrong, let's fix it at the source of the problem, rather than letting us wait to report the problem when it may manifest itself during testing.

## The Situation

I suggest that there is a fundamental problem not with process methodologies but with requirements development. We've been addressing the symptoms through Agile processes and Shift-Left but not necessarily the problem.

Requirements have lost their importance to project managers; due to being blinded by Business Analysts with an overinflated confidence in the quality of their own requirements, or those within the project who misunderstand the connection between requirements and the development and testing activities, or of course some project managers are ignorant to the costs of rework caused by poor requirements.

I feel that the situation has deteriorated over the last 4-5 years. Projects are run more by the Business Analyst team than the Project Manager. The project will drift from feasibility, through requirements definition, into development, and it is only during testing that we start discovering how close or far we are from the *requirements*, let alone from the *end users expectations and needs*. This latter point seems to me to be the most critical; I have met some exceptional Business Analysts over my time, but there are a large number of Business Analysts who don't recognise they are there to articulate the end-user's needs and wants.

By testing it is too late, and who gets the blame for project delays, and the bad news? . . . it seems to be that testers are the harbingers of doom, the proverbial pain in the project derriere, and the only ones that present a different view on the project compared to the up-beat and positive Business Analysts and Developers.

Some of the justifications for poor quality requirements I have personally heard over the last few years include:

*"The requirements are deliberately vague so we can catch out the vendor"*

*"The requirements weren't designed to be tested against"*

*"It's a cover-all requirement as we're not sure how it works yet"*

*"Yes I know the requirements aren't clear but we haven't got time to get clarification from the end-users, and it's not appropriate for you to speak to them"*

...or my personal favourite:

Client Business Analyst: *"We have specified the system to do x"*

Vendor Test Manager: *"We've already discussed this; the system won't be able to do that due to upstream"*

*systems not providing that data/the system can't do that/you haven't configured it to do that"*

Client Business Analyst: *"We have to write it down; the users want it"*

Vendor Test Manager: *"Then the tests we have to contractually perform against these requirements will fail"*

Client Business Analyst: *"That's ok, we can log that and work out what to do then"*

If the requirements are so important that you want me to test them, get it right, else let's not make them part of the project.

### **The Problem**

How do we as testers influence quality?

Currently we have a game of Chinese whispers

**End Users (needs) > Business Analyst (requirements) > Developer (code) > Tester**

We interpret requirements and compare it against code. We aren't even talking directly with the end user; we're already 2 and 3 steps removed. We're reading the analysis of a domain inexperienced Business Analyst who has tried to elicit requirements from an end user who doesn't necessarily understand the entire business nor the technology that may be able to assist them.

The whole concept of 'Shift-left' was designed to help us get engaged upstream, at least one step earlier, learning the requirements with the developers, ensuring that both developers and testers are aligned with their understanding of the requirements.

The benefits for Agile processes are obvious; allowing the entire team to understand the intent of the requirements at the same time.

[INSERT here your TYPICAL AGILE / SCRUM TEAM STRUCTURE]

However, if you're in a contractual-based requirements-assessed project, the options are limited. The terms are already decided and the process has been defined.

In my experience, collaboration can help but not resolve the situation. The silos are there 'protecting' the teams within. Therefore the only way to exert

influence is to use the silos as the software gods intended - governance.

### **The Resolution**

The first step was to educate both the client and the vendor in the risks that the situation presented with such poor quality requirements:

- The vendor would be held to account for the inaccurate requirements - identified consequences could be reputational and financial.

The client would suffer for the inaccurate requirements - identified consequences could be operational, safety, security and of course reputational, and most importantly end-users would suffer.

- There would be a large delta between the requirements, other specification documents, and the build. As the requirements were the main contractual measurement point, then as a vendor you would be held *contractually* to something you hadn't agreed to.

Once the organisations and stakeholders (I'm not a fan of this word; it always conjures up images of sirloins-on-a-stick) understand the risks and consequences of the poor quality requirements, I start providing solutions.

Establish your position as custodian of quality for the project - no one else is going to do it, and by creating the perception that you are responsible for quality, others find it natural to accept when you push back on poor requirements.

Enforce the approval of requirements prior to entry into testing.

- The test team must review all requirements and determine if they were testable, or else rejected and taken out-of-scope, until clarification or updates had been received by the Business Analyst team.
- Only those requirements reviewed and approved by the test team will be tested against.

Change management was implemented to ensure requirements would be released to the test team at various stages.

- This took work but at least we're able to control the influx of requirements

- It was created as a perceived ‘concession’ to the Business Analysts to provide them an opportunity to update the requirements, treating the release and rework of BRSs in the same way that code is managed.

Treatment of requirement issues as defects.

- We have a defect reporting tool yet we only use it to incident reports on code. I’ve used defect reporting to track environment issues, functional specification issues, and more recently, requirement issues. *Everything* erroneous in a project can and should be reported and assigned with an incident reporting tool.

Test only those requirements that have gone through the quality control process.

### How to review and gate the requirement

- Use Gilb inspection categories:
  - I found that the categories quickly allows the reviewers to communicate the issue, and also demonstrates a certain level of rigour is being applied. Perception is everything, and I’ve often found that addressing poor quality work with high quality professional analysis, strengthens your position and makes it harder for other parties to argue against you. Again this shouldn’t be taken as a first step but a gradual deepening of maturity which helps gives credence to your concerns to both your management and the management of the Business Analyst (especially important if these Business Analysts are in the employ of the client)
  - By characterising the issues in this way, you are able to bring an element of independence into the assessment; the review is no longer *your* opinion, but rather your implementation of a standard technique and approach. This tends to increase it’s validity in the eyes of others, and removes the argument that it’s the Business Analysts opinion versus yours.
- Quantify the issues:
  - Present the information as ratios of the total number of requirements
- Document what requirements you consider are missing from the equation. Providing a metric which only represents specified requirements would be misrepresentative of the situation. Adding a vague number of non-documented requirements doesn’t produce clarity but does truthfully show that the situation is incomplete.
- Summarise how many requirements are impacted. This will again alert project management as to how serious the requirements quality situation is.
- Quantify the issue types
  - The Gilb inspections will assist with this
  - Ensure information regarding past reviews is included. If you have provided the advice multiple times, whilst it is frustrating, it also demonstrates the intransigence of the protagonists.
  - Analyse to see if the issue types fall into particular patterns depending on the requirement types:
    - Are all the security requirements duplicated elsewhere? Suggests just inherited and not reviewed or developed with intent.
    - Do interface requirements conflict with others? Suggests that teams have not collaborated or that source systems may not have their interfaces clearly defined.
    - Are there gaps in business process definition? Suggests that the Business Analysts do not understand the domain, the end-users have differing processes, or that the Business Analysts have not accurately captured the business.
- Clearly report what has been considered in or out-of-scope

We provided detailed reports and summaries of the status of requirements.

- Identify specific contradictory requirements - not an elegant tool, but when used in a meeting to defend your approach, obvious contradictions can be pulled out and demonstrated as to why your approach is sensible. It will often surprise people to the blatant mistakes you are having to

deal with, and helps push the responsibility upstream.

- Approve the requirements
  - If possible, ensure that the test team is able to work with others and be part of the decision to approve the requirements at their sign-off. This doesn't always work and you're often fighting the perception that the Business Analysts are experts and you're just the testing, doing that testing stuff. It will take enlightened project management and line management support to integrate testing approval as part of the requirements sign-off process.
- Gating the requirements
  - If you can't make test approval part of the requirements process, you can prevent the requirements from entering the testing process.

### The Challenges

Some might perceive that you aren't the experts in the field of requirements so why are you doing the assessments?

- I would position myself as coordinator and look to SMEs to assist in the review.
- Testing is seen as causing a delay
  - I would show how addressing this now will save substantial rework and possible contract wrangling later.
- We haven't got time to do this
  - I would work through with others an example requirement issue already discovered, and estimate how long a requirements issue, if implemented, would take to resolve. Quite often it is easy to demonstrate the impacts to multiple dependent systems and modules along with the effort to rework requirements later; you still have to correct the requirements however if left you also have multiple configuration and code issues to address, plus the effort of testing. Barry Boehm's cost model and NASA studies are always useful to back you up.

### The Result

- Instead of the test team fighting the development team for what is or is not an issue, we're looking at the source of the problem.
- You've established testing as the gatekeeper of quality. Your workload and responsibility will increase but I find the project checks with you on what the correct path to take is.
- You've educated the organisation on how requirements are the basis of the project, not just a means to an end.
- You've also educated the organisation as to what testing does:
  - The purpose of testing is not to improve the quality of the product
  - The role of the test team is not to accept everything that comes its way
  - The purpose of testing is to act as advocates for the end user, ensuring that they get what is needed and what ensures they can do their jobs
  - Without access to end users, both testers and developers are dependent on the information provided by business analysts. If this information is considered poor quality, then the business analysts will have to answer to the testers and developers.

With a silo based organisation it is up to each disciplinary team to take responsibility for their work; I forced the issue with this approach. A rigorous software methodology needs to be exactly that; taking inspiration from Hamlet, then when deciding if 'tis nobler to suffer the slip-ups and errors of outrageous defects, or to take arms against a sea of troubled requirements, I took up arms.

When faced with poor quality requirements that you are forced to test against, for the benefit of you and your team's reputation, the end users you represent, and for your own sanity, take up arms.

**Rowley Emmett is a Corporate Trainer at Access Testing in Sydney. He can be contacted on [rowley.emmett@accesstesting.com](mailto:rowley.emmett@accesstesting.com)**





## Review: Let's Test Oz

by Mike Talks, Datacom

### The Personal Journey

There were a lot of nerves on my behalf leading up to the Let's Test Oz conference this week. I'd never been to Australia and somehow quoting Crocodile Dundee doesn't seem the best way to win friends and influence people.

What I'm going to do is pick up on a few of the best moments and revelations from the conference, to condense my experience ...

### The Train Ride

I had a two hour train ride in the morning with tester Kim Engel, fresh from our conversation about flat earth's. It was a great journey and a great conversation to get me in the mood - I almost talked myself hoarse. It reminded me that one of my favourite pastimes with my son is travelling, because whether hiking or driving, it gives you opportunity to talk and explore. With my son it's about exploring history and ideas around it. With Kim it was exploring aspects of testing and mental health that we've both had personal journeys with.

With such conversations it's actually often a disappointment when you reach your destination, because you've enjoyed the journey too much. Yup - I was actually slightly sad to arrive (*but not for too long*).



### Coaching Testers Workshop



I've spoken previously about James Bach and Anne-Marie's workshop on coaching testers. It had some great ideas circulating and James had found some examples from movies of people coaching others. I had to say I feel slightly ashamed that I have not yet watched the Magnificent Seven despite liking Westerns - I need to remedy this at some point!

Great focus was given to understanding yourself as a coach, who you are and what your behaviour is. How do you interact with people? What's important to you in others? Then to look at the person who is looking for coaching and asking what they need in a coach. *Sometimes that's not you*. James and Anne-Marie talked frankly about their coaching and that occasionally they will recommend an individual goes to the other for coaching.

As always with a good workshop this included reams of hands-on. We logged in anonymously to Skype and we got to work with other people in the room, taking turns to be the coach and to be the student.

I'll be absolutely blunt, I thought the person I was coaching knew me and was playing a game. I kept getting incredibly frustrated but trying to be calm with them. When it came to debrief it turned out that the person I paired with had never used Skype

before and was a slow typist. I learned a valuable lesson there that especially online, you need to get some form of check-in about the student, how they feel and what their backstory is, rather than "leap into" the coaching.

It's a lesson I really should know but it's amazing how the session helped to reinforce that and instead I made my own assumptions that it was someone trolling me.

### So you wanna be a boxer?



SoftEd ran a couple of boxing training sessions which were absolutely superb. I occasionally do something similar in Wellington (*I've talked about Josette, one of our instructors here*).

Boxing training is a very intimate kind of



training. You pair up with someone and take turns doing exercises, with one person using gloves to hit and the other holding the pads being hit (*you hit the pads, not the person!*). You have to both be mentally in a similar zone and develop a kind of rhythm with each other.

That makes it oddly quite a social activity. The best kind of pairing is when you're both supporting the other with "*try moving your stance*", "*we're half way, don't give up*", "*c'mon, keep going, nice*" - it was some

of Bach and Charrett's coaching tips in miniature. Half the success of any conference like this is being able to mingle with people who you don't now. Meet new people, make new allies. The boxing sessions proved to be a great "*meet and greet*" event, with conversations with other boxers strung out through the rest of the conference.

Which brings me to this tweet ...



The boxing was one method (*I'll talk about the other in a moment*), but there were the experienced keynote speakers touring the conference and there were other speakers such as myself. But mingling and listening I realised something important, "everyone has an experience report and a story inside them, just keep your ears open".

As I said in that tweet, in some ways listening to the raw stories from others was a great opportunity to really spread my net over the conference.

### Put yourself around people with passion

The other method of putting myself around a bit came at lunchtime. It all felt like being a bit at High School with "*who shall I sit with?*". Occasionally I just really needed to eat and dash, and I'd just sit alone. But I tried to use it as an opportunity to sit with new people and introduce myself.

I however earned myself the "*special little snowflake*" achievement on Monday lunchtime though - finding a table of people I didn't know and asking if it was okay to sit with them. *They were from a completely different conference!*

But this actually led to a bit of a revelation. They were a conference of ultrasound operators and were curious about who we all were. So I got to tell them a little about software testing as well as ask them a bit about their conference.

My interest was heightened a bit. Recently my brother and his wife had a daughter. One thing that surprised me was the ultrasound. When my son was "*under development*", we had a couple of ultrasound





I am attracted to spending time with people with a passion and an ability to animatedly talk about that passion. It doesn't have to be anything I'm interested in - in fact often it helps if it's not. Just this year I've spun pieces off from conversations with Josette my boxing instructor and Lotz my musician friend.

Testing has a good few parallels and if you listen out, there is knowledge out there from people going through similar experiences which you can shanghai and add to yours!

**An interesting chat with Erik Peterson leads to some self-reflection ...**

pictures of him and they were a bit like one of those 3D puzzles. If you stared at really hard, you couple perhaps make out a skull. .

But for Thea, her pictures were strikingly clear, the technology had really come along leaps and bounds. So surely being an ultrasound operator was a lot easier now? Wrong.

Turns out they do more and more checking with ultrasound as it's such a non-invasive procedure. They were spending a conference looking at example images of different ailments such as a damaged appendix. In their normal routine they might not see some of these examples so it was all about improving their ability to look and and recognise issues. They were using the conference to broaden their experience from other operators so when they went back to work on the Thursday they were just that bit sharper.

The improvements in technology made some things a lot easier. But at the end of the day it required a human eye and human judgement. And damn it - wasn't the same true about testing's relationship to technology over the same timeframe? Some things had got easier but at the end of the day, it's about the human eye and human decision making.

This led me to an important understanding - I learn a lot about testing but not always from testers. I'm quite a talker but I'm a good listener too. In fact, when we go touring around New Zealand, my wife despairs of me as I really enjoy going into quiet shops and talking at length to the owners, where we come from and finding out some of their history.



I had an interesting lunchtime chat with Erik Peterson where we talked about heuristic models for testing. I came to the realisation that I'm heavily dependant on an "experiential model" (although I do use others) - basically "when I used to program, I once saw this happen" or "I've seen a bug like this in a similar system".

That's of course okay, as long as you realise it's fallible. And its greatest fallibility is you aren't looking for a bug you've never experienced before - you have a blind spot to anything you've never seen or heard. It also made me look at some of my writing - overall my writing heavily leans towards a "series of experience reports", occasionally postulating a model from this experience.

It's an interesting look in the mirror at my way of thinking - also tying into my previous post about trying and failing. It's like I'm drawn to having a pool of experiences to base judgements on.

# This is the Cargo Cult trap

- Project context changes
- Use same procedures
- Expect same result



## My talk

Yup - I'm not being egotistical here but not only did I enjoy giving my talk on *"deprogramming the cargo cult of testing"* but to my shock, I walked out with an expanded take on it. Some of the questions asked allowed me to think and explore the subject in ways I'd not expected.

The topic was really talking about the system of testing we've put into place over the last 12 months, and I talked about it back in my piece on exploratory testing. We put together a new way of testing when we moved to being agile but we engaged with our customer to talk to them about what they felt they got out of our old methodology? What did they feel they get from a test plan, a test script or a test report?

The point was this was to form a matrix of values from our customer - this meant whatever approach we took for testing it needed to address these values in some manner. If it didn't then we weren't done with our approach, it wasn't hitting the needs and we needed to rethink. But not only that, we had to make sure we were making *"how our testing worked"* visible to the customer.

An example of this would be how the customer saw test scripts both as *"proof of testing"* and *"training material"*. We ended up using qTrace to record our sessions as *"proof of testing"* and for *"training material"* sharing an internal testing handbook we already had and making sure we kept it up to date sprint on sprint.

Someone noted the piece tied in a bit with Keith Klain's keynote where he talked about avoiding being overly whiney or self-centred about testing's

problems but understand the person you report to *"has problems and needs"* that you don't know of. To try and go to them not with more problems but trying to help and aid them.

The bottom line to this approach was that we made sure we had an evangelical fervour to delivering real value to our customers in a testing approach that we felt accurately addressed their needs. In this I really was pleased we seemed to carry on the spirit of Alessandra Moreira's talk about engaging and influencing people. In fact the conclusion from our talk was it would be a mistake to wait for a major shift from waterfall to agile before engaging with a client to ask if the testing you're performing is really *"ticking the boxes"* from both the client and the test team point of view.

## Final thoughts ...

These are the things that really stuck with me - a very interesting conference with a lot to take home. I sadly missed the Fiona Charles keynote at the end of the Wednesday which I was looking forward to.

There was a lot to take in but also fun to be had along the way - one of the funniest activities was being Joanne Perold and Carsten Feilberg's workshop where we replicated problems in communication by using a Lego building exercise to replicate the software building process. This was an exercise I would love to try again with different rules to see if it causes some of the outcomes I expect. Likewise the boxing and the coaching activities were nicely hands on.

The team behind Let's Test Oz really did an excellent job in making this happen - the venue and food was amazing, everything ran well, and everyone seemed to come to the conference ready to really share and engage. I made sure before writing this that I sent an email to the key players, asking them to circulate to all who needed to read it.

Great work guys!

Mike Talks is a Test Manager with Datacom in Wellington and a regular contributor to NZTester Magazine. He can be contacted at [mtalks@datacom.co.nz](mailto:mtalks@datacom.co.nz)

# Testing Events

Coming from OZTester Magazine in 2015:

NZTester Magazine Testing Leadership Summit

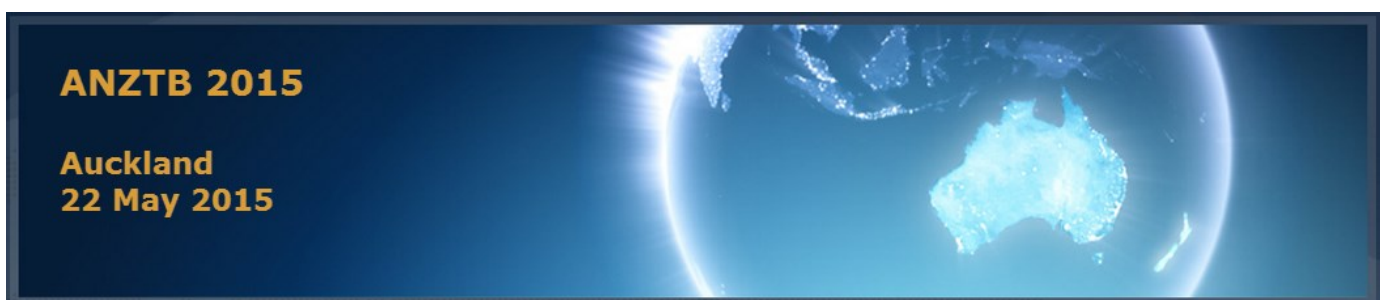
- Wellington, March 2015

NZTester Magazine Conference

- Wellington, August 2015

The Great Big OZTester Magazine Road Trip - August/  
September 2015

+ regular meetups in Sydney, Melbourne, Canberra, Brisbane, others  
as requested



If you have an event you'd  
like to promote on this page,  
email [ed@nztester.co.nz](mailto:ed@nztester.co.nz)



Subscribe to the magazines free at [www.nztester.co.nz](http://www.nztester.co.nz)



Calling all

## PROGRAMME TEST MANAGERS

There's now a LinkedIn group for all test professionals operating at Programme Test Management level (or at least aspire to). Click on the title above.

Subscribe to the magazines free at [www.nztester.co.nz](http://www.nztester.co.nz)



It seems like I always begin these reviews along the lines of 'it doesn't seem like  $n$  months since the last...' so this time I won't. Needless to say that another StarWest Conference has come and gone and at a rough guess, I'd say this was the biggest one for many years, 1,200+ attendees I understand.



*My test management class in full swing*

I was down for two tutorials this time; my traditional Testing the Data Warehouse and the Programme-Level Test Management session first introduced at our own conference back in August. Having done four tutorials back-to-back then, doing two at StarWest didn't seem like such a big deal. My sessions were well attended with Data Warehouse Testing selling out and the discussions lively and healthy. I do find it refreshing in some ways to find that no matter where on the planet we ply our testing trade, the challenges are always similar.

Wednesday 15 October saw the start of the StarWest conference proceedings. Unfortunately the day of tutorials on the Tuesday coupled with a bout of unexpected jet lag and I missed the first few sessions –which annoyed me no end as I really wanted to catch Julie Gardiner's keynote on testing web services, libraries and frameworks. The previous evening I'd caught up with Julie and Dawn Haynes

for a sneak preview of Julie's presentation so I was particularly keen to catch.

Wednesday afternoon I attended Mary Thorn's session on the Test Manager's Role in Agile, which I found interesting although not too much new for me. I then jumped over to the World Quality Report 2014 -5: Emerging Testing Trends presentation. At last year's StarWest I had attended the same session with a great amount of interest as this report provides a huge amount of depth and meat around the current state of testing and general software quality assurance. As expected, the results revealed not only a greater investment in the testing of mobile applications, which has been growing steadily for the past couple of years now, but also a continued focus on getting smarter, faster and more functionally rich software applications to market in much quicker time right across the board.

This trend has led to some marked changes upon our industry which I can only see continuing. I can liken it perhaps to the recording industry where I have a modicum of experience. In days past, to get a record or CD onto the streets, an artist usually had to be signed by a record company and have had many hundreds of thousands of dollars invested in both recording process and in artist development, a future investment. Then with the recording company marketing machine kicking in behind, an artist could sell kazillions of CDs and make megabucks for both the record company and themselves, usually in that order. Getting a top-flight CD out usually required input of at least \$500,000 however the returns for top artists were in the thousands of percentage points so no-one really worried too much. However in the last decade or so, technology has provided ways for artists to record themselves and produce professional sounding, studio quality recordings for as little as a couple of thousand dollars - for a PC and some clever software. On the surface, this sounded good as it meant that no longer did artists have to be signed to properly record and thousands more CDs et al became available from artists who otherwise would not have had the opportunity. In addition,



with the advent of iTunes and other media, record company marketing is no longer an essential requirement either, so a whole new industry paradigm has been created and that can only be good, right?

Possibly, however think about this: with thousands more artists 'in the market', the range of product is so much greater and the pressure so much more present to be able to compete and get ahead of the competition for the same sized customer dollar. With the record companies no longer investing in the manner they did in the past because the returns are no longer there, there is little or no professional artist development so the quality of artist and thence the resulting product is driven down also. If we're not careful, we'll end up with a general 'dumbing down' of the market and quality, which in my opinion is not good for anyone in the longer term, even if it means that there's more available, for less money and in greater quantities than ever before.

Could the same be about to happen with software, especially in the mobile space? To check on the results of my beloved West Ham United Football Club, I can choose from at least a couple of dozen apps on the Apple AppStore. I've been through at least five of them in the past two years with the main reason for changing being that I get sick and tired of trying to get them all to work properly and consistently over time. The quality has been 'dumbed down' and no matter, for free or at most another couple of bucks I can grab the next one of rank. Problem is in a commercial sense, every time I change I have to learn how to use, understand the features, learn to get the best out of etc etc. Then no sooner have I done this and got it working for me and a new version comes along full of holes and the cycle starts again. Not exactly the best use of my time. Now while this scenario is not necessarily present yet in the case of non-mobile, commercial software applications, the continued upwards pressure from this sector can only impinge upon this space eventually. Will we then see the software investors, like the record companies, pull away completely, seeking better returns from their dollars elsewhere? We've already been here once with the dot.com crash of ten or so years ago as part of the internet advent revolution. Is it looming again?

Anyway, maybe enough pontificating. Later Wednesday afternoon saw the ever-popular Lightning Strikes the Keynotes strike where

presenters volunteer for a five-minute lightning talk on anything around testing. I did my perennial 'Which Came First; the Bug or the Test?' which seemed to go down well as always although I really must come up with another lightning talk someday.

Thursday was another busy day; I sat in on Julie's 'Rainmaking for Test Managers' session although the title engendered in me a sense of dread. As a test manager I'm often accused of kicking up too much of a storm and here's Julie propagating the message? I needn't have been worried, as always Julie put forward a number of thought-provoking concepts, in particular one around taking a make-it-happen approach when everything else seems to merely roll along.

After a lively discussion around test metrics with Pablo Garcia and Michael Bolton, I then trotted off to Pablo's presentation around the same subject. I do wonder whether we get too hung up on metrics; as to whether they're valid or not. I've always seen metrics more as indicators, not as harbingers of absolute truth. And we have to be cognizant of the fact that they change on a day-to-day basis so is a daily snapshot anything more than just that? Is it not a trend over time that counts for more and provides more information around the state of our product, system or project than mere numbers? Hmm, anyway I'd better not get on my hobby horse here however suffice to say that Pablo was promoting a similar message in his presentation.

Fast forward to the 4:15pm session and Pablo Hope's keynote around Security Testing. Appearing on the podium resembling Gandalf out of the Lord of Rings trilogy, Pablo proceeded to peel back the layers of *Pablo Hope's Gandalf rendition!*





mystery around this subject point-by-point while at the same time shedding a layer of his wizard outfit accordingly. Certainly an entertaining approach to a subject that is still sometimes considered one of the testing 'dark arts'! I learned a few things too eg. that security testing is a 'must' for any product or system that is exposed via the internet (as if I didn't know that already but did I really know it?).

Thursday evening was a late one; caught up with Pablo Garcia, Rob Sabourin and Scott Barber along with a few others for enjoyable drinkies, dinner and healthy conversations. I think we all agree that the testing landscape is changing and that the whole 'shift left' momentum is gathering around bringing the tester's nose closer to the core of software development. I can't help but wonder though whether that now more than ever that the differences between testing a software product versus testing a systems implementation are taking on quite unique perspectives and that good ole approaches from way back need tempering accordingly. While the whole exploratory approach is riding the popularity wave right now, I still feel that we cannot lose sight of the tried, true and proven within the arenas where that approach is still valid. To say that one approach is the best one, the right one, the only one is in my humble opinion short-sighted and perhaps revealing that the propagator really does not yet have the breadth and depth of testing and management experience to be able to fully appreciate the wider perspectives and annals of testing. Anyway hereendeththerant!

Again as always, StarWest was a blast and well-executed by Lee and the SQE team. May they ever continue to be so!

## 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 [ed@OZTester.co.nz](mailto:ed@OZTester.co.nz)

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 me on [ed@OZTester.co.nz](mailto:ed@OZTester.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 still real low tech right now, there's two easy ways: 1) go to [www.OZTester.co.nz](http://www.OZTester.co.nz), or 3) simply click [here](#) - Ed.



*Click on title*

<b>Catch Software / EnterpriseTester</b>	<b>17</b>
<b>OZTester Magazine TestAnalytics</b>	<b>21</b>