

OZTester



The Quarterly Magazine for the Australian Software Testing Community and Supporters

ISSUE 4 MAR - MAY 2014
COMPLIMENTARY

In this issue:

Interview with Tony Bailey, Access Testing

What Type of Tester Are You?

Should We Be Testing Spreadsheets?

What's the Deal With Weekend Testing?

A (Very) Brief History of Test Automation

Planit Testing Survey 2013 Review

Great Bugs, From the Test Tool Box

Pinheads & more...

15648 465464

OZTester Magazine

Editor: Geoff Horne (acting)

ed@nztester.co.nz

geoff@nztester.co.nz

ph. +64 21 634 900

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

www.nztester.co.nz/oztester.html

Advertising Enquiries: ed@nztester.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 - OZ Tester Magazine, all rights reserved.

The Journal For Australian Test Professionals

Well here we are, first OZTester issue of 2014, we're hoping it will be a good one...2014 that is! It's also the first issue that I've personally undertaken editor duties for.

In this issue we have new insights from Dean Mackenzie and Nele Nikovic. We also continue our testing company feature interviews with Tony Bailey, CEO of AccessTesting, along with a revisit of the PlanIT Test Survey that was reviewed in NZTester2. This is an update taken from the 2013 survey and some of the changes between the two make interesting reading. You can find this on p16.

We expect this year to be busy on the conference front as well. We will be attending StarWest and ANZTB in May, TCANZ in October while already lining up Ignite, StarWest and the inaugural

Australasian Let's Test conference in Sydney this coming September.

We're also looking forward to seeing USTester Magazine leaving the tarmac this year, probably April time. While this is not yet a guarantee, we hope that we can get our stuff together in time to make it happen before StarEast in early May. Any US-based readers of OZTester are hereby officially encouraged to submit articles for submission!

With three magazines now in the stable, we've decided to stagger releases from here on in so this one is March, USTester will go out mid-April with NZTester mid-May, next OZTester in June and so on. Hopefully this means we can keep on top of the production workload!

Anyway, happy reading and as always, we do appreciate any feedback you may have around our publications.



IN THIS ISSUE...

Click on title...

- 5 Interview With Tony Bailey, CEO, Access Testing**
- 8 Should We Be Testing Excel Spreadsheets?**
OZTester Staff Writer
- 10 Tester Types: What Type of Tester Are You?**
Nele Nikovic, TradeMe
- 13 What's the Deal With Weekend Testing?**
Dean Mackenzie
- 16 PlanIT Testing Index 2013 Review**
OZTester Staff Writer
- 22 A (Very) Brief History of Test Automation**
Geoff Horne, Editor (acting), OZTester Magazine

REGULAR FEATURES

- 4 OZTester Magazine Announcements**
OZTester & NZTester Magazine Back Issues
- 15 Testing Events**
- 18 The Pinheads**
- 19 From the Test Toolbox - The Lazy Tester!**
OZTester Staff Writer
- 25 Great Bugs We Have Known and Loved - Deja Bugs**
Richard Boustead, Statistics NZ

CORRESPONDENTS WANTED!

Its jolly hard to keep up with everything going on testing these days, especially with three magazines now on the go!

We're looking for voluntary correspondents to help us:

- Provide testing news from the region
- Solicit articles of interest
- Help run the local Meetups
- Write an article or two per year

We're currently looking for correspondents in Sydney, Melbourne, Brisbane/Gold Coast, Adelaide, Canberra, Darwin, Perth and any other state regional centre. Email me [here](#).

OZTester Announcements

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/oztester.html



This issue's interview is with:

Tony Bailey

CEO, Access Testing,
Sydney



Our interview this issue is with Tony Bailey of Access Testing. Tony has been with Access for the past 16 years and has guided the company through each of its growth phases to the company that it is today.

OZTester: Can you please describe Access Testing?

At Access we specialise in providing independent assurance and testing services to help our clients deliver technology that really works for their customers. That's the corporate line. When asked at the BBQ at the weekend what we do, I say that we make technology live up to it's promise. We're Australia's and Hong Kong's largest technology testing company. Testing is all we do. We don't do training, we don't do business analysis or project management and we don't try to push you a tool because of some partnering agreement that you know nothing about. We have 6 offices and are part of the Dialog Group of companies - Australia's largest private IT services company - employing over 1,000 people nationwide.

OZTester: What products and services does Access Testing offer?

We provide a complete suite of services that spans the planning of technology systems, the building of technology systems and the analysis of deployed technology systems. We use testing skills, tools and techniques to solve business problems and we put the customer experience at the centre of everything we do.

OZTester: What do you think makes Access Testing different?

What makes Access unique is our fundamental belief that technology is there to serve people and

therefore all technology should put the customer experience front and centre. Unfortunately the Testing and IT industry has lost sight of this basic premise over the years and instead focuses relentlessly on the technology and ignores for the most part the customer.

In the testing world this trend can be seen where we spend the vast majority of our time dealing with functional testing and very little time dealing with non-functional testing activities, that ironically deal with the things that most customers are concerned with. At Access, we get this and that is why all our testing is aligned around delivering a superior customer experience.

OZTester: What do you think makes a Test Manager or Analyst come to work for Access Testing?

With all of the professional testers we employ, I think they like the fact that we are not another "me too" organisation. We don't agree with the direction the industry has taken and we want to stand up and do something about it. They see our vision, our passion to try and change the industry and the testing talent that we've pulled together and they want to join us. I remember interviewing one 20 year veteran of the testing industry, who was quite jaded by the industry and when I outlined our approach he said "you know, I've never thought about my profession that way before. I had thought that we just had accept the way things are but with what you've just shown me, I want to sign up and join the fight". When you can reignite the flame of motivation in intelligent peoples minds powerful things can happen. That's what we're doing at Access and that's why I think people want to work for us.

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

I believe the challenge for Australian testing companies lies in innovating and differentiating themselves over an increasingly commoditised offshore model centred around cost.

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

I think over the 18 years that I've been in this business I think we've done a good job in educating the customer for the need for independent testing. In the early years of our business as with our competitors, there was an awful lot of time spent in client education. That has changed in the last decade where the client now accepts the need for testing but unfortunately they still seem to see the testing activity as a 'throw a number of bodies at the problem' activity rather than a value add activity, which means they are missing out on a lot of value their business could get.

OZTester: Where do you believe Australia's approach could improve?

When I sit with CIO's, they are very frustrated with the level of service they are getting from their Tier 1 providers. They are experiencing what we call the Quality Fade game, where the A team wins the testing business and then over time is removed to win the next piece of work and the C team is left to deliver in order to meet the tight margins the business was won under. Ultimately quality fades and problems occur. I think organisations need to be wise to these sort of games. I think Benjamin Franklin summed it up best with his quote "The bitterness of poor quality remains long after the sweetness of low price is forgotten." If there is one thing that could improve Australia's approach, it would be to remember this quote.

Editor's comments: Thank you for making time to write for us Tony. I'm certainly aware of the Quality Fade scenario and have seen a number of companies fall into this hole. One in particular comes to mind where within 3 months of an assignment commencing, not one of the original team was left onsite, having been replaced by a number of comparatively inexperienced resources. I think the answer lays in the engagement agreement where the client perhaps

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

Yes I think it is. However the market is also growing and with that comes a proportional increase in lower testing standards. This is something we all have to be careful of.

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

There's no doubt that mobile testing is already very important for many clients and that will only continue to grow. Security testing is also a hot topic both here and overseas. I think the next big wave to hit the industry will be Customer Driven Testing - you heard it here first!

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

I obviously can't name specific clients but I can recount the story where we won the tender to provide the independent testing for a large Government Agency where one of the Tier 1 System Integrators was the prime contractor. Needless to say the project didn't go too well and we were witnessing very poor quality in testing, giving rise to large numbers of unresolved defects. The System Integrator started to play the "shift the blame game" by putting in two defect managers to work over the weekend to reassign the defects to everyone else except the System Integrator, so that when the steering committee report was published on a Monday morning, they were looking good. By Tuesday, all the defects were reassigned back to them and on and on it went. Horrendous project that ended up being scrapped. I'd like to say that this was many years ago and that it was a rare occurrence but alas...

needs to ensure that it can state its specific, vendor resourcing expectations.

The old "defect-flick" trick is another one that resounds with me, along with the "bug scrub" - the nasty habit of deprioritising issues that should not be deprioritised, just to make a snapshot-in-time report look good. The numbers can be fudged to suit however poor quality will ultimately never be hidden - Ed.

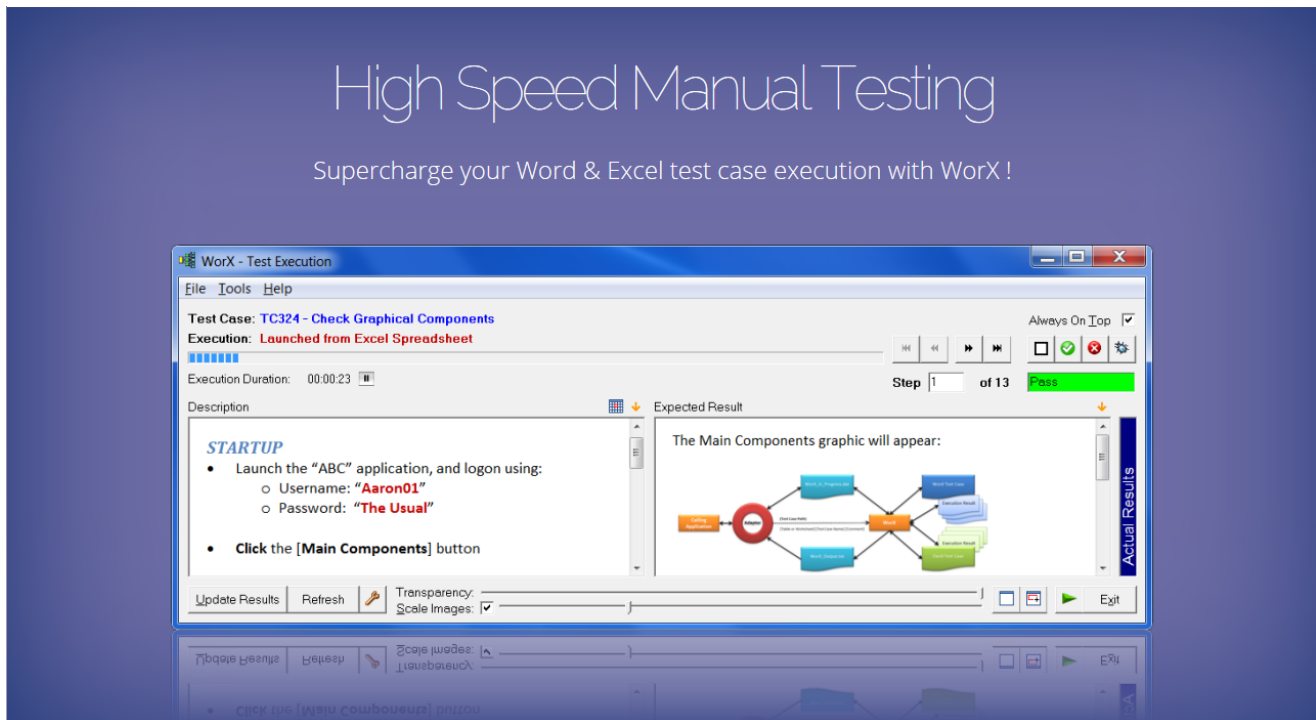


Webinar: Manual Testing 2x Faster in Word & Excel

Register Now: <http://www.autom8.co.nz/webinars/>

Date: **Tuesday, April 1st**

Time: **1:30pm– 2:30pm AEST**



85% of all software tests are still being performed manually. Is there a way to significantly improve manual testing and be able to work 2x faster? This free webinar is designed to change the way you perform manual testing. "Manual Testing 2x Faster in Word & Excel" focuses on new ways to reduce your manual test effort. Join this webinar to learn how to:

- ✓ Build, execute and analyse software tests directly in **Word and Excel**
- ✓ Make **UAT, Exploratory, SAP and CRM manual testing** much easier
- ✓ Test websites and applications with greater **flexibility**
- ✓ Improve the efficiency of manual testing and **test 2x faster**
- ✓ Integrate with popular **test management systems**

Ask questions and take back valuable tips and tricks to use in your **every day** manual testing. Space is limited so sign up today. **Register Now:** <http://www.autom8.co.nz/webinars/>

See you soon.

- Aaron Athfield, Founder and Chief Manual Tester Guy nz.linkedin.com/pub/aaron-athfield/75/811/626
www.autom8.co.nz

autom8

Should We Be Testing Excel Spreadsheets?

by OZTester Staff Writer



Excel has been around for yonks. I remember playing around with an early version sometime in the mid-eighties when Lotus 123 was all the rage and do actually remember a predecessor product running on Apple II called Visicalc, which I understand kicked the whole spreadsheet (or financial modelling as it was known in those days) craze off.

However unlike most crazes, this one is as strong and as sophisticated as ever. Excel is now so powerful that full-blown software applications can be built with it and as long as you don't mind its limited speed, security and sharing capabilities, I'm sure a top-notch Excel practitioner can get it to do just about anything. I've developed Test Estimation and Forecasting tools using Excel which I've used on just about every testing assignment I've been on and I've not seen any test management tool do what these can ('scuse the big head!).

If I use these tools on my commercial assignments then I have to be jolly sure that the information they're providing is accurate and timely. Even a small goof in my VBA code could be providing me and more importantly, my client with incorrect information around the how-long, how-much etc of testing initiatives. But hey, I'm a tester so whilst I can never claim that any code is totally bug-free, especially my own, I do know after many years use that these tools are reasonably good (if I do say so myself).

OK so that's a testing project. Spreadsheets, both simple and complex have long been embedded in commercial, education and government processes and spreadsheet errors, no matter how small or seemingly insignificant, can cause the same negative outcomes when used for critical purposes as defects in enterprise software applications. For example:

- In 2003, a prime US mortgage provider moved to a new accounting system and reconciled much of its data migration using

spreadsheets. An error in one of them led to a \$US1.3b financial hit.

- In 2012, a faulty reference in a US state education planning spreadsheet led to a major discrepancy in student enrolments with a negative impact of \$US25m. Two top state finance officials were forced to resign over the fiasco that ensued.
- Again in 2012, a large financial institution in London blamed a spreadsheet miscalculation for the amount of risk traders were taking on, to the tune of \$US3.1b.

And while we plough our testing efforts into large, complex information systems and networks, these potentially volatile islands of information hardly ever see the testing light of day. The main reasons I think are as follows:

- Business spreadsheets are developed mostly by non-IT individuals, of their own initiative and ability rather than under software development auspices.
- Spreadsheets are often viewed as nothing more than glorified calculators as opposed to bona-fide information processing and data repositories in their own right.
- Non-testing personnel are often predisposed to assuming a new development works unless it actually fails, as opposed to merely producing an erroneous result.
- When we see results that we like eg. a better than expected sales result/risk profile/student enrolment etc, our subconscious bias can kick in and the inclination to double check is reduced, especially if the effort to achieve has been high.
- Spreadsheet creators treasure their creations (I know I do), often believing that no-one

knows as much about its structure and/or relevant business use so it would be pointless for someone else to test or even review it.

Human errors will occur in any software development regardless of the experience and ability of the developer, unfortunately this is nature-of-the-beast and totally unavoidable. And using Excel with any level of complexity **has** to be considered a software development.

Excel provides a huge number of inherent functions to avoid unnecessary coding however while a temptation may be not to test the results of function use, I know that many of my own errors have arisen due to misuses and misunderstandings around these. It's a little like using the configuration tools within enterprise systems (many of which are now code-based); because they're part-and-parcel with the software application, it is often assumed that they work regardless of how they're used.

So should we be testing Excel spreadsheets and applications? In theory, you betcha! And if you think about it, it makes logical sense. These things can be as complex in design and implementation as any software particularly when coding is involved so the kinds of peer review and other quality assurance processes that enterprise systems are subject to should be exercised. Would we ever question the need to test new code within an enterprise application, of course not - even if we **are** sometimes ignored!

In reality however, if you run off and attempt to put a case forward to include every spreadsheet in use within your organisation under the auspices of your test team then you are likely to greeted with fits of laughter. I suggest a better strategy is a soft approach based upon i) risk and ii) your sphere of influence. If you know that a spreadsheet is being used for business-critical purposes in an area that has a track record of issues arising from feral spreadsheet use then you may have grounds to at least risk the question. Otherwise it may be a question of biding your time and waiting for the right opportunity to arise.

If you do get a green light somewhere along the line, what type of testing and quality assurance processes should be deployed with spreadsheets? Pretty much everything we would do on an enterprise system

from a functional test perspective; here's a few suggestions (you can probably think of others):

- Peer reviews of design, structure, references, formulae, VBA code, pivot tables, function use, totals, charts, database connections etc.
- Standard testing approaches of given specified input, did we see the expected output?
- The usual test design techniques: boundary/domain analysis, invalid input, cause/effect.
- Negative test conditions as well as positive.
- Security testing around passwords, sharing, access etc.
- Usability; around intuitive layout and use, ease of understanding etc.
- Investigate charts, pivot tables etc and any other methods of data representation (sometimes the data is correct yet not accurately presented as such eg. displaying the appropriate number of decimal places).
- Total columns and indeed anything that is arrived at from a range selection. Some of the most common spreadsheet errors are caused by missing cells in the first/last row/column in range selections.

In summary, this is probably a classic case of theory versus reality, yes of course spreadsheets should be tested however probably won't be, unless there is good cause to justify. Perhaps a few well-placed questions coming out of a few fortuitous circumstances might help open the door to the opportunity of at least assessing what has occurred in the past and therefore what may occur in the future. Findings may then help to identify potential risks around the continued errant use of spreadsheets in your organisation and the case for raising as a concern then carries far more weight. Good luck!





Testing Types - What Type of Tester Are You?

By Nele Nikovic

So, what type of tester are you? Well, I find asking myself that same question!

Recently, I attended a [WeTest Meetup](#) in Auckland, which I recommend to anyone willing to know more about how we test in New Zealand. It is structured in a way to allow participants to talk and share their experiences. It may not mean that you will completely agree or completely disagree with what is said in the Meetup but you will certainly learn that how you test in your company has an alternative somewhere else. Hopefully, it will make you think, doubt and question current practices.

I have always tried my best to be very open minded and not be 'by the book' as each project is different, each team has a different skill-set when inter-compared, and each company has its own culture with foundations that was either initiated at the start of its journey (startup), or built and maintained over time, or both. It's an approach of mine that I can try and explain by going slightly off topic, philosophically, with a quote from a very famous British social critic, Bertrand Russell, who says: *"The fundamental difference between the liberal and the illiberal outlook is that the former regards all questions as open to discussion and all opinions as open to a greater or lesser measure of doubt, while the latter holds in advance that certain opinions are absolutely unquestionable, and that no argument against them must be allowed be heard."*¹

This is not just factual in our test industry but it is evidently applicable to many things we do in life. Even though the essence of e.g. test processes being the same in two similar companies / teams, you are still not guaranteed that the same solution can be applied to both companies / teams. If need be that you try and experiment or pilot-test a new tool, process, team structure, etc., so be it. At the end, isn't it better to build something empirically to suit your needs and achieve the common goal, than to be stubborn and follow something that is

outdated and produce mediocre results at best? Tough question I believe, but also a very hard one to prove and sell.

I will try to simplify the above paragraph with my personal experience of various physical concepts I came across in my testing career. Though my experience in testing is not vast, the concepts observed are large enough to grab my attention and write about them in hope it is a thought-provoking material. To correlate the concepts mentioned, in the Meetups, a topic of Test Job Titles has been discussed. And not only in a way of reaffirming what a particular job title stand for, but it also created a confusion among many as to what that tester's responsibilities are for certain roles.

So, what is the difference between a Test Analyst, QA Analyst, Test Engineer? Could it be that the terms are completely arbitrary, depending on which recruitment agency you stumble upon? I would really like to hear recruiters approach this topic and perhaps standardise the roles, as it would help many of us. However, it's likely that the employers are the ones driving this naming convention. Google Inc. made an interesting comparison of how they have classified their test roles. See <http://googletesting.blogspot.co.nz/2007/03/difference-between-qa-qc-and-test.html> for more on that.

So, what are the actual, official definitions?

Test Analyst²: Identifies items to be evaluated by the test effort, defines the appropriate tests required and any associated Test Data, gathers and manages this Test Data and evaluates the outcome of each test cycle.

¹ Freedom and the Colleges (Article in The American Mercury, 1940

² http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workers/wk_tstanl.htm

³ **QA Analyst:** Is responsible for maintaining software quality within an organization. Such individuals develop and use stringent testing methods. These professionals are focused on providing the confidence that quality requirements will be fulfilled.⁴

⁵ **Software / Test Engineer :** Designs and develops high quality test plans and test cases. Software Engineer works closely with Development and Program Management teams to provide feedback on design (product and technical) and user scenarios. [Test Engineer] drives improvements in unit testing coverage, develops test suites, expands automated testing harness, automates end-to-end tests, and validates metrics and reporting accuracy.

In order not to enlarge the confusion aspect, I have deliberately excluded Software QC (Quality Control), as QC differs to QA, in my opinion, to say the least. QA testing is process orientated, as opposed to QC which is predominantly product orientated. The QC may appear to be correlated to the description of a Test Analyst role, as mentioned above, but its prime focus is to verify that the product does what it is supposed to, unlike QA which ensures the product meets the needs of customers⁷.

Now, the above descriptions are mostly the official definitions we tend to use widely but, in my opinion, not wisely. The testing industry is constantly evolving, and with the IT industry being young and innovative as it is, can we really rely on something that is written 5+ years ago? I believe it is a must that we update our skillsets constantly as the market drives this change more often than we think.

The job title used for people you usually report to, QA / Test Leads and QA / Test Managers, is somewhat easier to explain. Test Managers have dealt with Test Strategies, wider team focus, team budget and reviews. Test Leads also had their share of these responsibilities but are more hands-on with testing. I am certain there is a lot more to add here but I do not feel as competent in trying to describe and distinguish these two in more detail. I would, however, be interested to hear our Editor's (Geoff Horne) opinion.

Once you have a title attached to your name, the next thing you will stumble upon is the testing

terminology which has a great tendency to differ from what has been described in the ISTQB / ISEB course. What struck me first was the name used for Test Scripts, which by definition is "*A document specifying a sequence of actions for the execution of a test. Also known as manual test script*" - (ISTQB).

If I could form a hierarchy, I hope it would not be too incorrect to say that it is the [Master] Test Plan first, which consists of many Test Suites / Levels that have numerous Test Scripts, with Test Cases within (Test Plan->Test Suite / Level->Test Script->Test Case). By definition, a Test Plan is a record of the test planning process⁸, which, if serving as a Master Plan, is done during the strategy meeting(s) and usually by Test Leads / Managers. This is especially evident with large companies, mainly in using Waterfall as a methodology standard.

Trade Me, my current employer, uses Test Plan simply as a Word or Excel document for test coverage of a particular area "infected". Perhaps, a similar path has also been used with HPQC software (Test Plan>Test Cases). There is no further propagation of work but the test execution which follows the moment a fix or a so-called branch is in RTT state (Ready to Test). Recently, we at Trade Me have introduced Test Case Management software which has a Test Suite term used as a highest hierarchical set of Test Cases⁹. There is no mention of Test Script anywhere whereas that is the term I was familiar with as a junior tester (prior to Trade Me). For those with automation experience, Test Script could mean something slightly similar to what you have used in the past but then again, it is still distant (used in e.g. TestComplete)¹⁰.

³ <http://www.pcmag.com/encyclopedia/term/50006/qa-analyst>

⁴ <http://www.rbc-us.com/media/glossary/>

⁵ <http://www.microsoft-careers.com/job/Hyderabad-Software-Test-Engineer-%28SDET%29-II-Job/27536400/>

⁶ <http://jobsearchtech.about.com/od/careersintechology/p/SWTest.htm>

⁷ <http://googletesting.blogspot.co.nz/2007/03/difference-between-qa-qc-and-test.html>

⁸ <http://www.istqb.org/downloads/finish/20/101.html>

⁹ <http://www.gurock.com/testrail/>

¹⁰ <http://smartbear.com/products/qa-tools/automated-testing-tools/test-case-automation-with-scripts/>

Regardless of the Methodology used (e.g. Waterfall, Kanban, Agile, etc.), I found that companies use the terms they are comfortable with. I am in favour of such flexibility, even though it creates confusion initially but it is only a matter of time one gets used to it. Perhaps, it is the Contractors who are best to speak to about this as they are the ones who constantly face synonymous terminologies.

To conclude, testing industry is improving its reputability by continually expanding on the test types, becoming a mainstream activity in SDLC. It continues to spread its tentacles, especially with the resurgence of Agile. The dev-test ratio is becoming smaller and “fairer” (from e.g. 4-1, to an ideal 2.7-1, and even down to 2-1), project segments are a lot more flexible than before (estimation, design input, etc.), our feedback is valued more than ever, we play a vital role in production release, and the job of a tester is therefore gaining the credibility that it deserves. Compared to a decade ago, many independent software testing services are emerging, filling the gaps and offering the sustainability for various size corporations.

Various terminologies and job titles are just a small example of how large our industry is. To a non-IT individual, it is no longer possible to explain in a few words what it is that we do. Regardless, it is this flexibility and general open-mindedness to change processes, teams, titles, etc., that is helping the growth of our domain.

Nebojsa Nele Nikovic is a Test Analyst at Trade Me in Auckland. He can be contacted at nele@trademe.co.nz

Editor's comments:

Thanks Nele, to take you up on your opinion invite, I think you raise some interesting points. While this maybe in the eyes of some folk a Priority 3 testing subject and maybe not one to lose any sleep over, any issue around terminology which has the potential for miscommunication deserves a mention and perhaps discussion.

Regarding titles; I've tended to not put so much emphasis on a title, rather I'm more interested in what the person actually does. A Test Analyst designation can mean anything from a UAT Test Analyst with a few months experience testing on a package implementation to a Systems Test Analyst with countless years testing at infrastructure level. Yes, we could put clarifying terms eg. “senior”, “applications”, “technology” around the words however these can make the situation even more confusing. Other ‘modern’ practices such as putting a pseudo-job description on a business card IMHO only cheapens the role and perhaps leads to recipients wondering why?

I'm not sure whether there is a useful solution to this - only that in most scenarios, we have to make sure that all parties are clear on precisely what the expectations are around any task, position etc. I would hope that this approach would win through in most situations hence the need for glossaries and bibliographies in all documentation and certainly for probing questions in verbal situations. - Ed.

Have you seen this interview in the last NZTester?

OMG, please tell me it's not you!



What's the Deal With Weekend Testing?

by Dean Mackenzie



Around Australia and New Zealand, thousands of software testers ply their craft diligently from Monday to Friday. Come the weekend, testing is one of the last things on their mind (at least until the dreaded Sunday evening). But for a small crowd of dedicated individuals around Australia, New Zealand and North America, the last Saturday of every month is another opportunity to do testing, talk testing and learn more about testing... all from their own homes. These people are gathering online to take part in another Weekend Testing session.

What is Weekend Testing (WT)?

Weekend Testing is an entirely volunteer-run software testing movement that was founded in India by Ajay Balamurugadas and it's since caught on in North America, Europe, Australia and New Zealand.

But what IS Weekend Testing?

Once a month, a group of testers assemble on Skype to “attack” a both a pre-determined web-site or mobile application and a particular theme. The site or app can range from just about anything found on the internet, while the theme can cover a diversity of topics, from different testing techniques to other testing activities (eg. test planning) to the social and psychological aspects of testing.

After the “mission” is set by the facilitator, the weekend testers explore the site or app in small groups, pairs or by themselves for a set period. Everyone then re-joins the main forum to discuss how they went about it and what they found. The discussion can flow thick and fast as ideas are exchanged, points are met with counter-points and lessons are learned.

I've been testing all week. Why would I want to do more on the weekend?

Despite the name, Weekend Testing isn't just about jumping online to “test more stuff”. In fact, for some, that may not be the purpose of WT at all. So here's a few reasons why Weekend Testing might appeal to you:

1. Learn more...

Unless you're a “test guru” of supreme level (and admittedly there might be a couple around), you will learn more about software testing. In any given session, you might discover a lot more about test planning (WT session 15), assumptions and inferences (WT session 16) or even combinatorial tests for mobile devices (WT session 17).

2. Experiment and try new things...

If you're looking for a forum to practice your testing skills with peers, Weekend Testing is a great choice. However, if you're searching for somewhere to try a new test technique out, Weekend Testing might be an even better choice! And as for feedback or other tester's thoughts on a particular matter, the Weekend Testing forum provides an accessible channel of expertise to call on.

If you're especially keen to explore a particular topic in-depth, you may even be able to approach the facilitators to organise a session specifically around the topic.

3. Exposure to new technologies and concepts...

The world of software development is always changing, with new technologies and concepts materialising almost every month. Weekend Testing looks to introduce some of these to the testing community. For example, mobile application testing has become a popular topic in the WT chapters across the world, while a “Blitz Night Testing” series in India (open to anyone from any country) has run introductory sessions on using developer and web tools from a tester's perspective.

4. Meet new people...

...and take part in an online testing community that spans the globe. Novices to experts from different countries frequent the weekend testing session, bouncing ideas back and forth, talking “shop” and helping each other out to improve and learn.

But I've never tested online or in a group before...?

Testing online can be different to testing alongside other people in the same room. Testing in a pair or a small group is often different to testing alone. But in testing this way, the group dynamics allow a rich set of questions, lessons and possibilities to be raised, so that everyone walks away with something new, be it a new test idea, a new technique, or a completely new perspective on software testing.

Well, I only have minimal (or no) testing experience...?

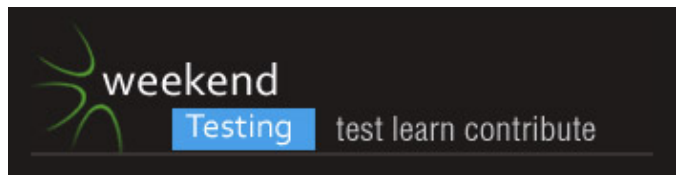
There is no "minimum requirement" in terms of experience or skills to attend Weekend Testing, just a willingness to participate and learn.

OK, I'm convinced. So how do I attend one of these Weekend Testings..?

Just jump on Skype come the day of a session and contact **weekendtestinganz** to be added! Upcoming sessions are advertised a couple of weeks in advance on both the web-site and Twitter (@WTANZ_ for the Australian and New Zealand chapter).

2014 is looking to be a big year in the life of Weekend Testing Australia and New Zealand, with a number of

guest facilitators coming on board to challenge, provoke and improve the testers attending. I hope to see you there.



Dean Mackenzie is a software tester based in Brisbane, Australia. After being "volunteered" to perform UAT, he immediately gravitated towards testing and has been doing it ever since. He loves to learn new things and likes to be involved in the Australian online testing community.

Dean blogs at yesbroken.com and can be found on Twitter as @deanamackenzie.



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



THE BLOCKBUSTER CONFERENCE ON
SOFTWARE TESTING ANALYSIS & REVIEW

BUG OUT!

STAR EAST

MAY 4-9, 2014
ORLANDO, FLORIDA
ROSEN CENTRE HOTEL

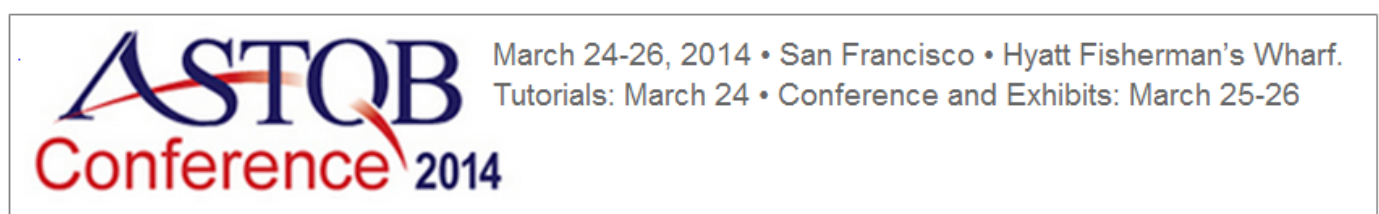
Testing Events



GREAT SOFTWARE REVOLVES AROUND GREAT TESTING

STAR CANADA

APRIL 5-9, 2014
TORONTO, ONTARIO
HILTON TORONTO



ASTOB
Conference 2014

March 24-26, 2014 • San Francisco • Hyatt Fisherman's Wharf.
Tutorials: March 24 • Conference and Exhibits: March 25-26



QUEST 2014 NORTH AMERICA
QUALITY ENGINEERED SOFTWARE & TESTING
CONFERENCE & EXPO

Learn. Assess. Apply. Connect.
Baltimore - Washington D.C.
April 7-11, 2014



STP SOFTWARE
TEST PROFESSIONALS
CONFERENCE SPRING 2014

Register Today

April 14-17 2014 New Orleans, LA



ANZTB
ANZ TESTING BOARD

ANZTB Test 2014
Sofitel Sydney Wentworth
Friday 16 May 2014



Let's Test

For those about to rock...

Let's Test Oz Style, 15 - 17 Sept 2014
Sydney, Australia

If you have an event you'd like to promote on this page, email ed@nztester.co.nz



By OZTester Staff Writer

Planit Testing Index 2013 Review

As a precursor to reading this article, I suggest a visit to www.planit.net.au to download this year's Planit Testing Index Executive Summary - Ed.

I didn't make it to the Planit Testing Index roadshow this time around. Not that I didn't want to; events conspired against me and I never seemed to be in a place long enough to get myself along. However as always, Planit has made the documentation available to all so I have relished the opportunity of working my way through and determining how things (may) have changed from the 2012 survey.

I've noticed that in most specialist industry surveys conducted on an annual basis that the percentages tend to change no more than plus or minus 5% points from year to year. In other words things stay pretty much the same; slightly up one year, slightly down the next et al. So the first statistical difference that caught my attention was that the number of responses from New Zealand was up from 9% in 2012 to 21%. In fact there were more respondents from New Zealand than New South Wales so looks like the Kiwis have taken the encouragement made in NZTester2 to heart and given this year's survey a fair crack of the whip.

Another number that caught my eye, although only a 3% increase on last year was the number of respondents rating Testing as a "Critical Element in Producing Reliable Software", up from 48% in 2012 to 51% ie. just over half. I shouldn't cheer too much though as I would have thought that this one was a no-brainer, warranting a much higher assessment than just over half. Does this mean that 49% of respondents do not see Testing as a "Critical Element in Producing Reliable Software"? The mind boggles.....as mine is so apt to do.

Last year's eyebrow-raiser of Desired (62%) versus Actual (18%) ratings for Testing starting during the Requirements phase was similar this year; 60% and

15% respectively (I now have both eyebrows raised). However when reading on, we see that the main reason for project failure is still "Poor or Changing Business Requirements" at a whopping 70% (up from 68% last year which in turn was up 9% on the year before) so maybe not such a surprise after all.

So onto the whole Requirements matter; has anything improved on last year? If you've read NZTester2, you will have noted that 28% of respondents reported feeling positive about their company's Requirements Definitions with another 44% feeling OK (=28% not OK at all). A huge 97% conceded that their company could benefit from improving Requirements Definition and 67% believed it was suboptimal at the time. Unfortunately 2013 has seen no improvement and has indeed worsened: just 23% positive, 39% OK and 38% for that other category – a 10% increase on 2012! In addition, 99% of respondents now feel that their company could benefit from improving requirements definition and 71% believed it suboptimal. So there you have it....disappointing to say the least. Testers: we must yell louder!

The Project Methodologies category yielded an interesting statistic; last year respondents reported a breakdown of Waterfall 36%, Agile 29% and V-Model 24% (ignoring the Other or No Methodology responses). We queried at the time whether respondents fully understood the differences between Waterfall and V-Model and wondered whether the categories had been used interchangeably. This year the breakdown is Waterfall 33%, Agile 33% and V-Model 22% showing a small but distinct increase for Agile (4%). It should be remembered that the "Agile" umbrella includes a number of different methods eg. XP, SCRUM, RAD, BDD/TDD/FDD/TLADD etc although I think most folk tend to think of development Sprints as synonymous with the term "Agile".

Project Outcomes showed a significant increase for the “On Time/On Budget” category, up to 52% from 39% last year, good news! The breakdown of this category by project method also makes interesting reading; Agile 52%, Waterfall 49% and V-Model 55% (same caveat on the last two as per above). It is encouraging to see the increase here as it reverses the downward trend from the last three surveys.

The Testing Investment section stills shows expected increases in spending around Structured Test Processes, Testing Tools and Testing Training as the three main areas for 2014 although it's interesting to note that plans to Engage Contract Testing Professionals rose from a 19% increase last year to 29%, with only 25% expecting a decrease – good news for the contract market!

Utilisation of Performance Testing stayed very much the same as for 2012 which is a little surprising given the increased awareness of Performance Testing and Engineering. The specialist Performance Testing companies that I have contact with report that they are rushed off their feet especially in the telecommunications and banking sectors where emphasis on designing and developing IT systems for performance as opposed to purely for functionality appears to be growing.

On the Software Testing Tools front, HP continues to rule across all three categories; Test Management, Test Automation and Performance Testing. Interesting enough though, in each category the next most popular tool is not from another generalist vendor but a specialist tool ie. Atlassian Jira, Selenium and Apache JMeter respectively. Other traditional vendors eg. Rational (IBM), Microsoft, Tricentis and SmartBear notch single figure usages only with a few eg. MicroFocus (Silk), Telerik and Fitnesse not rating a mention. I also wonder whether Jira users might also be including NZ's own EnterpriseTester in their figures.

Finally, I always find the Project Conditions section an amusing one. Last year I homed in on Project Estimation as I've always found that when estimates prove to be too light, that it's testing that cops it at the southern end of the project lifecycle. In other words, the architects, business analysts and developers have spent all the money

so sorry, testing has to be done on and with a postage stamp! In 2012, those who rated Estimation for Budget and Timeline as Poor or Very Poor were 27% and 31% respectively. This year it's up to 33% and 39% so oops, no improvement there! If we add in Realistic Expectations and our old favourite, Requirements Definition (neither of which I mentioned last year) at 34% each for 2012 and 39% and 38% respectively for 2013, this makes these four categories, which of out of the 10 assessed are the most applicable to testing (in my humble opinion), the lowest rated categories of all! Same as last year, gulp! Please excuse me being so negative and cynical, maybe it's the tester in me!

In summary, while some of the categories this year are worthy of further optimism eg. project success rates, New Zealand participation in the survey et al, those that are applicable specifically to testing do seem to be creeping westward (and no, not to Western Australia). Will we ever see a day where we're all satisfied with requirements, estimations, expectations etc, no, probably not and it's possibly quite naïve to think that we will? However, that said, it does mean that we have to continue to i) find better, faster and more innovative ways to test and ii) keep the flag aloft around these areas, and then just maybe we might start to see a swing east again. Until next year....



Accelerate your SDLC & reduce risk with Automated Defect Prevention & Continuous Testing

Test anytime,
anywhere



Accelerate
releases



Reduce
risks



Ensure
quality



Register [here](#) for our live March webinar and an introduction to the [Parasoft](#) solution-set

Speak with *Parasoft's* local distributor and support partner today



NZ: +64 4 473 8535 | AUS: +61 2 8035 3413 | parasoft.info@integrationqa.co.nz | www.IntegrationQA.com

The Pinheads

Early one morning in the PMO....

There's
no way we can
finish testing on
time!

You
just need to take
more ownership

Can I
hire some more
testers?

No!

OK...can we
reduce the number
of modules we test
then?

No!

So....
I'm taking
ownership of the
failure?

You catch on fast.



From the TestToolBox - The Lazy Tester!

by OZTester Staff Writer

Preamble: *experienced test automation practitioners may find this article somewhat juvenile. However it is targeted towards manual testers who may not necessarily have had exposure to automation or perhaps view it only as applicable to regression testing.*

My example is from a traditional testing environment where manual test scripts are deployed however I see no reason that the approach could not be used in lean or exploratory situations.

OK I admit it; I can sometimes be a lazy tester. This term is usually used in the case of testers who don't like writing test scripts and find all sorts of creative ways to make this task a little more bearable. Unfortunately what often happens is that no-one else can follow them and therefore properly execute their scripts without referring back to the author, assuming of course that you are following the traditional objective of making scripts less tester-dependent and more process driven.

I work more in the systems implementation space under project conditions, some very tight, as opposed to pure software development, where there's (usually) a finite number of test conditions to assure a successful implementation. Funnily enough, I don't mind writing test scripts as long

as I follow the good ole risk-based approach of ensuring that my high priority test cases are scripted with a high level of granularity and vice versa. The idea of making someone else's life in the future just a bit easier or at least endeavouring to actually appeals to me. I tend to move from project to project quite often and mostly I'm not around to be referred to and even when I am, I often wonder whether there might be a swear word or two in front of my name!

My self-confessed laziness lays around the execution; the seemingly never-ending task of executing each test step, verifying my actual versus my expected results, recording each one and logging any defects along the way plus there's the retests. Now maybe this approach is going the way of the dinosaurs as more and more innovative methods emerge however there are still market sectors that require this type of discipline eg. defence, hospitals and legislation-driven businesses, where due diligence is paramount across all company operations including IT.

A recent hands-on test exercise I was engaged for was to test the Terminations functionality of a large Payroll/HR system that included parameters and configuration options for Africa. Given that I have a background that includes test automation, I wondered whether my trusty TestComplete licence could help me out. With the many different types of Employee, Leave, Allowances and Payment options, I could see a very long road ahead if I restricted myself to pure manual testing, whether it be scripted or otherwise. Like every good tester,

I had been taught and told that automation was best used for Regression and Performance Testing due to the high maintenance inherent with accommodating continually changing functionality. So using automation for pure Functional Testing is something I've tended to steer away from. However given that my requirement in this instance involved the setting up of so many different combinations and variables to test, doing it all manually would have been a large, ongoing task - this was one of those applications where the test set up can take far longer than the actual execution.

So what did I do? I recorded some very basic automated scripts to set up configurations and parameters, input employee attendance, perform pay runs and terminate employees. Once working, I then developed a .txt file containing a record for each different test combination with the input data required and the configuration options needed to be established or tweaked, with the data combinations used to determine the functionality within the recorded automated script to follow. I then amended the scripts to read in the .txt file, replaced the hard-coded data and configuration values with variables containing data from the .txt file, executed through the application loading data and configuration options as it went. After I managed to get everything working, I then expanded both the test data and automated script coverage to cater for more and more test scenarios with increasing complexity. By the end of the exercise I could easily amend the .txt file data and configuration options to terminate employees every which way but loose (sounds a bit morbid doesn't it?), amend payout options; running pays whenever I needed to make it all happen - a mix'n'match to create test automation for every possible scenario I so desired.

The results I checked manually as the application made it quick and easy to see whether an expected result had been achieved or not. However I could have included this in the scripts if I had wanted to

provide a complete automated solution. Hopefully in the future someone can pick this work up and use it as the basis for a full-blown set of automated Regression Tests. I do have to proffer a small amount of caution here however; the application I was testing was mature software and the testing of Terminations was required due to modifications to expand this functionality. I cannot necessarily recommend this approach for the building of a new application or any application for that matter without understanding its specific context and requirements.

One of things I have found over the years is that success with test automation often comes down to the way the framework is designed ie. the .txt files, their data combinations, methods of reading and writing etc. Carrying too much of this logic and detail within an automated script can make the traditional overhead of script and test case maintenance an unnecessary burden.

This approach as per the preamble I know is fairly basic for test automation experts and is known loosely as data-driven testing. It's common in suites of automated Smoke and Regression Tests however in this instance, I used automation to reduce the amount of manual testing time for the Functional Testing of expanded as well as existing functionality. I estimate that I saved around 60% of my execution time versus manual. It also helped that a minimal level of defects was found as the development team was pretty darn good at Unit Testing (bless 'em) and no, the defects didn't all come out in Production!

There have been many situations over the years where I could have done the same or encouraged my teams to do so. However in days past, the features of automated tools have been more limited and the costs higher so it has not always been a cost-effective option. It may be more viable now that more and more open source tools are available, covering an ever-increasing range of architectures and technologies.

IMMEDIATE OPENING: FULL-TIME QUALITY ASSURANCE MANAGER



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

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

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



Visit www.seapine.com/oztester to learn more.

**Free
30-Day Trial**

© 2013 Seapine Software, Inc. All rights reserved.

A (Very) Brief History of Test Automation

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



Test automation is nothing new, it's been around in one form or another for 20+ years. It actually existed in a primitive form in the old "green screen" days however for the purposes of this discussion, I'll keep to the "Windows era".

Early automation proponents pitched it as the end of manual testing and started selling brightly-packaged boxes of diskettes (lots of them) and manuals by the truck load. Names such as SQA, Mercury and Segue dominated the automation landscape and the IT industry heaved a sigh of relief, after the advent of client/server architecture had introduced so many more layers of software complexity that needed testing.

However the glory days didn't last. Very soon automation users realised that there were many "gotchas" to using these new "fangled" tools – not the least being that every time their application-under-test changed, the automated tests had to change also. While the toolset manufacturers introduced additional features over time to help minimise this effort, users gradually got to the point where the time taken for automation development and upkeep far outweighed any benefits derived.

Many early adopters attempted to apply automation to applications-under-development, not exactly the most stable of platforms and found themselves spending more time developing automated test scripts than they did actually testing software.

Others discovered over time that their development environments yielded varying degrees of incompatibility with the available tools and even if usable, the scripts had to be so severely hobbled and cobbled that deployment became unviable. And by the time the tool manufacturers caught up, the next round of environmental and development enhancements had been delivered.

Even those users who had been highly inventive with their automated libraries found that newer

versions of their databases, operating systems or development tools etc. would come along and all of sudden, automated test suites no longer worked without serious update. Through attempting to outdo each other, the tool manufacturers built in so many bells and whistles, that their out-of-the-box technologies and features evolved from providing useful options to creating a rigidity which ended up actually constraining rather than enhancing the automation value proposition.

So test automation took a hit and many of those brightly-coloured boxes either remained unopened or went back on the shelves for many a day and in some cases many a year. Test automation suffered a "Toy Story" fate and became an old, discarded plaything - one moment, a favourite, the next a cast-off.

However...

While the rest of the IT world carried on its merry way and new technologies continued to be produced, the potential of automated testing and the enthusiasm for it in some quarters of IT never waned. After many years of experimentation and implementing new and improved products plus the application of fresh techniques to automation deployment, the tools have started to make a comeback. With the explosion of internet and cloud technology, software has become even more complex and coupled with similar explosions around software development tools and methodologies, test automation has been cast right back into the limelight. In many cases, it has become not merely a nicety but a necessity if we are to continue building ever-increasingly feature-rich yet stable and robust applications software, especially in agile-type environments.

As well as the raft of commercial test tools available, low-end web-based products have appeared. These are basic and simple yet offer a level of flexibility that the more expensive tools cannot – as they're

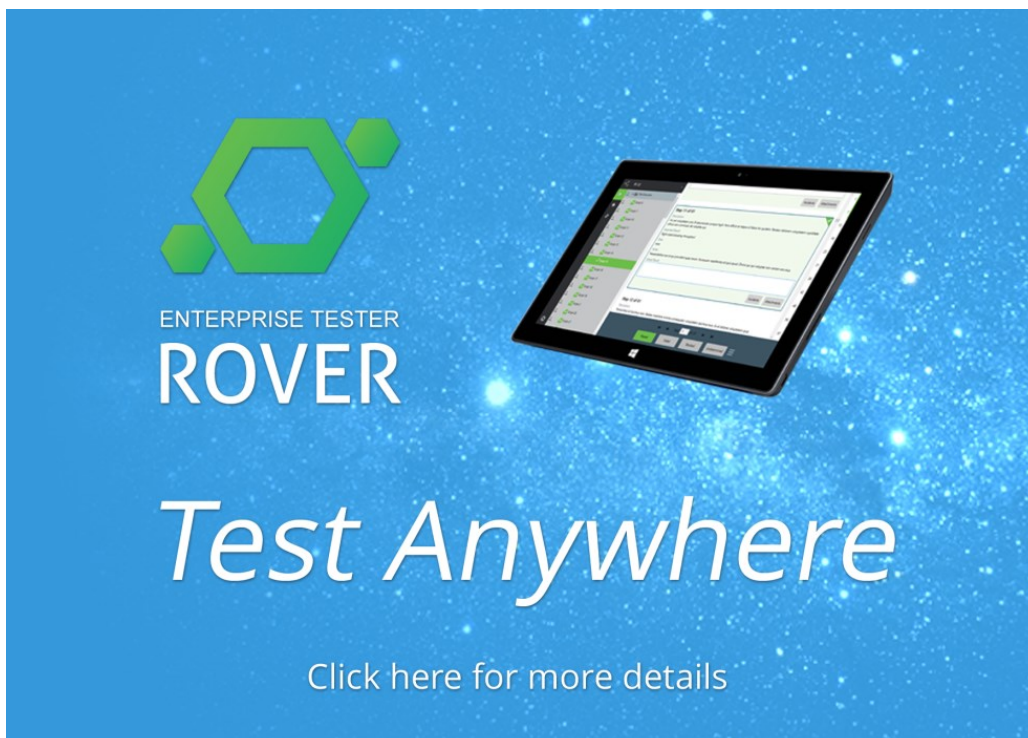
not constrained by their out-of-the-box features (because they hardly have any)! Believe it or not, this is quite OK as due to their simplicity, many are available at low cost some even as shareware and a flurry of new names have come to the fore; Selenium, Fitnesse, WinTask etc. This type of tool, because of the lack of the feature set of their commercial-grade big brothers, can provide much greater scope for developing automated scenarios - by focusing more on specific application and business test requirements as opposed to a generic black-and-white/pass-or-fail type of approach. While the “real” tools can sometimes perform such feats, the lower end options do so without the pricing overheads of purchasing a feature set that may not really be required.

Now, in general the lower-end tools will require more development effort than some (but not all) of the big guys and this may offset the tool cost savings somewhat. But as always if you plan to evaluate test automation, do so with your own specific requirements in mind and select the best value proposition for your situation. I always suggest a focus on cost-of-ownership over 3-5 years rather than dwell on mere pricing or potential savings.

While I’ve yet to see a comparison of low versus high-end tools applied to the same requirement with cost, effort and time metrics over the 3-5 for each, I am now convinced that test automation is no longer the money pit it often used to be. If deployed in a manner that promotes maintainability and efficiency, the productivity gains once touted 20 years or so ago may now become a reality.

Lastly, I know many readers will be asking: do tell about “a manner that promotes maintainability and efficiency”. Well, sorry folks but like the cliff-hanger at the end of a season of Dallas (anyone remember Dallas?), that’s a subject for the next time round.

PS Wondering what happened to SQA, Mercury and Segue? SQA was acquired by Rational Software which in turn was acquired by IBM. Newer versions of the toolset are still branded as Rational. Mercury was acquired by Hewlett-Packard and the updated toolset rebranded as HP. Segue branded their toolset under the Silk banner and although the company was acquired by Borland which in turn was acquired by MicroFocus, the toolset and its descendants are still branded as Silk.



The advertisement features a blue background with a starry, nebula-like pattern. On the left, there is a green hexagonal logo composed of several smaller hexagons. Below the logo, the text "ENTERPRISE TESTER" is written in a small, white, sans-serif font, followed by "ROVER" in a large, white, serif font. To the right of the logo, a tablet computer is shown at an angle, displaying a software interface with various fields and buttons. At the bottom of the advertisement, the phrase "Test Anywhere" is written in a large, white, cursive font. Below this, the text "Click here for more details" is written in a smaller, white, sans-serif font.

“If you adopt a rigid world view; religious, anti-religious, political, economic, academic, aesthetic or otherwise - you tend to see everything through the filter of your convictions, and, not surprisingly, you see what you're looking for. The more you use a particular theory for making sense of things, the more things seem to fit that theory. That's why fundamentalists feel so sure of themselves and why they can't understand how other people could fail to see things the way they do. It's why they eschew the mystical: they don't want to marvel as Einstein did or to rest with mysteries; they want to wrestle them into submission. It's also why they feel entitled to impose them upon others - believers with different beliefs and non-believers alike”

- Hugh MacKay, Australian psychologist, sociologist, social researcher, writer and former teacher.





Great Bugs We Have Known and Loved

(Deja Bugs!)

By Richard Boustead

In a team environment, assisting other testers on their tasks is not an unusual requirement – if anything, it is to be expected. So when I entered the meeting room for that first meeting, I was startled to see my departing colleague was grinning in a most disconcerting way.

I went through the introductions around the table, took the hand outs and asked a few questions. Everything seemed pretty set though – apart from the minor issue of being nearly three months behind schedule.

The issue there was that the software was for a scanner, and the hardware (the scanners themselves) had not yet shown up. After three months. I rolled with it, even though I was beginning to suspect what my now absent colleague had been grinning about.

The scanners arrived, which is where the second wrinkle showed up. Software was loaded, sample machine was set up, and my list of scripts were lined up and waiting. Just like any other project at any other time.

Our first run-through of Scanner V1.0 brought up some small, low severity bugs. Nothing especially major. Bugs were identified, logged, evaluated and escalated to the vendor.

Scanner v1.1 arrived and all was well. Identified faults were fixed, performance had improved and there was a host of additional new bugs to process. I mentioned to the project manager that despite being three months late in starting, I really couldn't see anything that would delay testing too much. I should have known better than to taunt Murphy. ScannerV1.2 arrived. The new bugs were fixed, and there were some new ones to deal with. At revision 3, the number should be shrinking, but it was a

complex bit of software. We began logging the new set, and then sat back with a terrible feeling of *deja vu*. These bugs had already been logged – three weeks ago. And they were marked as closed....

At this point, the other tester I was working on the project with noted that those were identical to the bugs in V1.0. Investigation soon proved him right. The cycle continued through V1.3, and 1.4. Each time, we were picking up previously fixed bugs and having to re-open them. We hit the end of scheduled testing, still with 140 bugs outstanding, and a good two thirds of them were the reoccurring phantoms.

While idly speculating among ourselves as to the cause and bouncing crazy ideas around, my colleague suddenly blurted out "They're giving us fixes based on the original, not the latest version!" The BA overheard, and called the Project Manager, presenting our overheard speculation as The Truth. Project Manager called Senior Manager; Senior manager called Vendor, and the Vendor? The Vendor admitted to giving us fix versions based on the original release of V1.0, not the latest build. Huh.

It wasn't too long later that I was moved off the project and moved back to my regularly scheduled BAU releases. As my replacement came into the meeting room, I was careful to give him a great big grin.

The lesson I learnt from this escapade? Bugs aren't always in the program. They can be in the process as well. Also, *déjà vu* tends to mean someone *didn't* change something in the Matrix

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@nztester.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@nztester.co.nz and please advise in your email if you specifically do not want your comments published in the next issue otherwise we will assume that you're OK with this.

Sign Up to Receive OZTester

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



Click on title

WorX / Autom8	7
IntegrationQA	18
Seapine Software	21
Catch Software	23