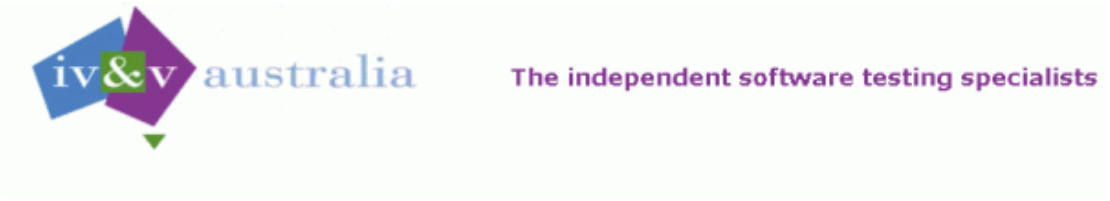

Welcome to IV&V Australia Testing Newsletter

This e-newsletter provides a practitioner's view of how to manage and perform SOFTWARE TESTING in today's world.

IN THIS ISSUE – January 2010

Happy New Year from IV&V Australia

The summer break is over for some of us, and as such we here at IV&V Australia would like to wish you all a Happy New Year. We hope you all had a chance to unwind over the past few weeks and that you have an exciting year ahead! We also hope to continue providing little insights into testing through our monthly newsletters and look forward to working with you on addressing any of your testing needs throughout 2010.

Test Tools – Making the right choices.

Choice of test tool is a very case by case subject. Boil them down, and (assuming all the tools in the pot technically do the job required) the three main factors that need to be addressed are; ability to integrate within the development and test framework, time and cost of use, time and cost of on-going use.

Evaluation trials will show if the tool will do the job technically but only once it is used in the intended environment can it be seen if it is a good choice. Most test tools that become shelf-ware do so because they require changes in the work-processes which cannot be implemented as they lack a crucial piece of functionality when deployed with the other test systems (for example a test-automation system may be unable to communicate test results to the specific defect management system being used) or the costs of training staff in use of the tool are too high or simply due to staff inertia.

Below we'll expand on some of the factors that should be considered when investigating and implementing test tools.

Ability to integrate within the development & test framework

Any tool used must integrate within your processes with the least interruption or change to your current processes. The more changes people have to make to the way they currently do things, the more resistance will be encountered to those

changes. Any changes should be visibly beneficial to the individuals affected and to the Project as a whole. It would be inappropriate to consider using *Quality Centre* as the defect management tool, if you're already using *SilkTest* as the automation tool for example. However, if you're using *Quick Test Pro (QTP)* (or even *Winrunner*) as the test automation tool and having difficulty managing defects then *Quality Centre* would be the ideal choice (even if other defect management systems are cheaper and/or have more functionality) as it integrates very well with *Winrunner/QTP*. So prior to adding any new test tools look at your existing tools and decide which will be easiest to integrate. The smoother the transition in adding test tools, the easier it will be to improve your existing processes and add more tools in the future.

Time & Cost of Use

Any tool used in the development/test of software has the underlying point that it is supposed to reduce costs & timescales. If, for example, you have a simple HTTP-only website that has no user input, then use of a test automation tool will most likely cost much more than manual testing.

So what will be the human resource costs of using a particular tool? A test tool may look good on paper with the marketing video and the salesperson making it look straightforward to use, but what skills will be needed to make it fly as gracefully as the salesperson made it look? With a great many test tools (including frontrunners like *QTP*) anyone using it must be highly skilled in all aspects of it. A salesperson may proudly show the record/playback function, but in the real-world the record/playback is hardly ever used as most tests need to be written in a readable way and often are querying back-end databases, checking data-files or performing other pre-test or post-test validations that have little to do with the UI. If you have a test-team of 5 people, and every one of them needs to be skilled in VBScript (with possible T-SQL, Webservices, etc) to use the newly purchased *QTP* test tool, then there is going to be a very high startup cost and very long delays as the test-team comes up to speed in the use of the tool.

Cost of on-going Use

Once a tool is in place and functioning as envisaged, what will be the on-going costs? If the tool has a low user-base then it means that new team members will probably need to be trained in use of the tool and won't be fully productive for some time. If personnel are found with the required skills to use the test tool, then they will cost more than if the tool had a greater number of users. When a core unit in the test target is modified, how quickly can the tests affected be identified and re-engineered? Some tools make it very difficult, with the Test technician having to read through many lines of code to identify which tests are affected, and some make it quite simple with self-documenting tests etc.

Another factor affecting on-going costs is the maturity of the tool. Very rarely does a test tool pay for itself (and the associated costs) on its first project with a Company. So it must be used on future projects, taking advantage of the skills base its use is creating. But will it still be around and supported in future years? If the tool will cease to be supported – maybe the company developing it goes bust for example – then it will have a short lifetime due to changes in the target host system (It may be fine for testing under Windows XP, but may fail under Windows 2007 without any update), losing any investment built up in its use. Additionally many tools may fail to adequately test if development moves to a new platform, so always keep in mind any future plans for technology or architecture changes to your existing applications when choosing test tools.

Test Tool Listings

Doing research on what test tools are out there and what functionality is provided can often be an exhausting and time consuming process. As such, below are a couple of sites that may provide some basis for test tool research. As a warning, some of the information on these sites may be a little out of date, but the information could still prove a useful starting point if you have been tasked with investigating test tools.

http://www.softwaretestingwiki.com/doku.php?id=testing_tools:testing_tools
http://www.testingfaqs.org/

Why we don't test in production environments

You may have heard the story regarding a recent Airport security test by Slovak police? If not, Slovak airport security services recently conducted a test to examine their security procedures and accidentally allowed an explosive parcel contained in an unwitting travellers baggage to board a flight to Ireland. Upon arrival in Ireland, the unlucky traveller was detained by local authorities. Fortunately he was later released upon admission from Slovak authorities of the botched (and some might say, poorly planned) test.

And this is why we NEVER test in our production environments!

Full story: http://news.bbc.co.uk/2/hi/europe/8441891.stm

FEEDBACK

Have you found this issue useful? We want to hear your comments and suggestions. Email us at info@ivvaust.com.au.
For more information about IV&V Australia, visit our web site at http://www.ivvaust.com.au.

If you do not want to receive further correspondence, please respond to subscribe@ivvaust.com.au with "unsubscribe" in the subject line.