

## **Testing Experience: the [free] magazine for professional testers December 2008 issue: Test Automation Dot Graham's summary and evaluation**

As this is a topic I am quite interested in, I read the whole issue cover to cover - 22 articles - 100 pages. (It was a long flight.)

I give a high-level overview of the issue first, list the articles I think are the best ones to look at if you are short of time, and then give a comment about each individual article in the order they are in the magazine.

### **High-level overview**

The main emphasis is on test execution automation, but there are some articles on other types of automation and other testing topics. Some of the articles are rather technical (code-oriented); others are very readable for non-technical people. There are some interesting case studies, some describing the process they went through to choose what tool to use - this is useful (sales pitches excepted).

A recurring theme in many of the automation articles is that automation is successful when it is planned and invested in, particularly the “framework” – what we refer to in our book<sup>1</sup> as the Testware Architecture or Automation Regime. The main impact of doing this properly is to minimise the maintenance of automated tests.

Unfortunately there is also a recurring theme that the purpose of automation is to find bugs (early and fast) – this is not a good objective for regression automation!

The articles vary in quality with a couple of them being thinly-disguised sales pitches. There are also a lot of typos in this issue throughout – it would have benefited from being reviewed or proof-read before publication! (Testers should practice what they preach, right?)

### **My choice for the best to read in this issue**

- interview with Tauseef Kham (most of);
- Mark Michaelis on GUI automation including 12 useful Rules;
- Dr. Mike Bartley on static and dynamic tools and which tests to automate;
- Marco Torres on Multilanguage (localization) test automation;
- Troy, Mitchell & Black's use of keywords to enable testers to write test without technical automation skills;
- Beer & Menzel's case study (doing it right).

---

<sup>1</sup> Software Test Automation, Mark Fewster & Dorothy Graham, AddisonWesley, 1988.

## Detailed comments on each article:

Markus Kröger: Testing performance in complex environments

A short but good performance testing case study (using Compuware tools)

Thomas Thurner: A brief introduction to testing technical embedded, software-specific systems

A very informative and clearly written description of the challenges faced in testing and test automation in this area (bugs are called “conspicuities”).

Interview: Tauseef Kham, senior director, QA, Borland

Tauseef focuses on automation objectives, Return on Investment (ROI) and using automation in agile environments – the key factor is the short feedback loop. Lots of good advice, though it ends by promoting Borland’s Quality Maturity Curve.

Alberto Vivencio: Profitable, beneficial deployment of test automation in the area of SAP Core-Banking

I admit I don’t know much about SAP, but I found this article a bit strange. I don’t think that it can be “proven” that business processes will continue to work, for example. He describes eCATT, a SAP testing tool, but seems to emphasise capture playback. One interesting metric is that it took 9 times longer to automate a test than to run it manually – this is not good automation long-term, though it may be an effect at the beginning. I couldn’t make sense of the ROI diagram. They did manage to get to data-driven and parameterizing though and he gives some useful examples using a banking simulation.

Koen Wellens: The Record & Playback Fairy Tale

A very good description of why Record & Playback is not test automation, and a way to start moving away from it. Give this to someone who thinks it is!

Mark Michaelis: Boon and Bane of GUI Test Automation

A good article describing his experiences using various commercial tools, eventually finding that QF-Test met his requirements. He ends with a good list of 12 Rules for GUI Testing Automation.

Huw (or Huuw – it is spelled both ways) Price: Practical approaches to improving your testing by maximising code coverage in complex database and SOA environments

He is a fan of Richard Bender’s approach and shows how cause-effect graphing is better than all-pairs, all combinations and Jenny at achieving coverage with fewest test cases. I don’t agree with his conclusion that increasing code coverage is the route to improved testing, however! Coverage is related to thoroughness, so coverage is great if you want to show increased thoroughness. But thoroughness is not the only aspect of good (or improved) testing. Sometimes good testing is

selective rather than thorough. Aside from that niggle, however, it is an interesting article using an example to illustrate his points.

Claus Gittinger: Model-based Test Development and Automation

This is a sales pitch for the expecco tool, which is based on UML models.

Erik Van Veenendaal: Test Process Improvement Manifesto

Short and to-the-point article about how to do test improvement well – a very good article (not specific to test automation).

Adrian O’Leary: Doing Automation the Smart Way!

This is a sales pitch for Cognizant tools, but it also has some good advice, so worth reading. Mark Fewster and I both do a tutorial called “Test Automation the Smart Way” for several years, so it is interesting to see this title (also in the Cognizant ad). What they call a “Driver Script” or centralized engine sounds a lot like what we call the automation “Controller” in our book.

Interview Andreas Golze and Stephan Goericke

This is a sales pitch for the new QAMP (Quality Assurance Management Professional) qualification scheme also advertised in the magazine. It consists of four elements: IREB, ISTQB Foundation, a specialised module, and an assessment of practical experience, and must be renewed annually.

Konrad Schlude: Risk Investment Test Automation?

Two short case studies, one a success, the other, rather predictably, a failure. He quotes an interesting statistic that 63% of test automation projects fail (from Brian LeSuer).

Dr Mike Bartley: Improved time to market through automated software testing

A very good article with lessons learned from his experience. He makes some interesting points about static testing (static analysis tools) but also discusses dynamic testing, with useful advice to deciding which tests to automate.

Marco Torres: Automating the testing of the GUI for Multilanguage applications

A very good article with lots of tips for using automation to pick up things that are difficult to spot manually in localization testing. Also includes references.

Stephen Troy, Jamie Mitchell and Rex Black: A Simplified Automation Solution Using WATIJ

A very good article, covering both selection of the tool, and the way they made it work for lots of testers using only a few test automators. The examples given are excellent examples of the keyword-driven approach (see our book ;-))

Rami Jaamour: Ensuring SOA ROI

Difficult to say what this article says, other than fairly basic “motherhood” things, such as that web services need non-functional testing as well as functional testing. It claims to illustrate best practices that can ensure secure, reliable and compliant SOA, but the article is just a list of things that can go wrong with no solutions. Doesn't explain what WSDL is (I presume Web Service Definition Language?). There is no mention of Return on Investment (ROI) in the article. I suspect this is the FUD part of a sales pitch.

Alon Linezki: Root Cause Analysis – Dealing with problems not just symptoms

Not automation-related but a very good article describing root cause analysis using cause-effect graphing. Some things are a bit confusing – I wonder why the table has two of its Types repeated – I suspect it is an error. The diagrams are not easy to read (I printed it from the downloaded version) and no reason is given for the very last one. But problems aside, a useful guide to finding out what is really going on, with a view to actually fixing problems once and for all.

Armin Beer & Michael Menzel: Test automation patterns: Closing the gap between requirements and test

A somewhat misleading title, but a very good article describing a case study of doing it right! Nice to see a concrete ROI calculation – they were ahead after 6 cycles. Lots of good advice (and useful references).

James Christie: The Seductive and Dangerous V-Model

A well-argued critique of the V-model, making many points that I thought were well understood by the testing community, but apparently are not, nor are they understood in government or project management communities – this is a bit scary. They regard the V-model as a testing model (it isn't, it is a development model that includes more detail of testing than the waterfall). But without the “Test early” emphasis, the V-model facilitates the illusion of control, but at the expense of quality. The article is UK-biased and references things over a number of years.

Marcin Michalak & Pekka Laukkanen: Robot Framework for Test Automation

Robot Framework is an open source Python-based framework that provides support for keyword-driven automation. It has some nice features – I like the tagging. However, their claim that it can be used for acceptance testing by those with no programming experience is certainly not supported by their examples – many of them are very code-looking. However, it is great to have this type of tool available as open source.

Albert Farrē: The cost of automation: A method to resolve the convenience of test automation

A rather mis-leading title; this article describes a very detailed way of calculating whether or not an individual test should be automated. He makes a very good point on the first page about not trying to automate everything, with a nice diagram. However, he then gets rather carried away with very detailed assessment of individual test steps and mathematical formulae. This seems to imply that the decision to automate a test is done only once, but this should be an iterative process. It also restricts automation to existing tests, which is not good. There are also other factors to take into account in the decision, not just ease of automating (e.g. criticality of test, how often it is run, how error-prone the manual test is, etc.) For advice on which tests to automate, read Mike Bartley's article instead.

Torsten Zimmermann: Software Testing with T2

This is a sales pitch for T2, a test framework for integrating test artefacts and tools. However, I have no idea whether T2 is a commercial product or open source, though the commercial tool CaseMaker is mentioned in the article. Or it might be a case study with a proprietary in-house tool - it would be nice to know. The approach has been influenced by TMap and fits with that terminology and life cycle model.

Yaron Tsubery: Yaron's Forecourt: Test Automation – What's going on behind the scene?

This presents some generic points about automation in a different way, but I'm not sure what the point is. There are also many typos (14 on less than a page is not good).

I hope that this summary has been useful to you, and I am always interested to hear from people who are doing test automation.

See my web site [www.DorothyGraham.co.uk](http://www.DorothyGraham.co.uk) or my blog at <http://dorothygraham.blogspot.com> or email me on [info@DorothyGraham.co.uk](mailto:info@DorothyGraham.co.uk).

Dot Graham, 5 January 2009