

Thoughts on the SIGiST Conference December 2013

The SIGiST conference held at The Barbican in December had a strong focus on test automation. Automation expert [Dot Graham](#), a software testing veteran of 40 years, gave the opening keynote and focused on common mistakes people make when adopting automation. She describes them as “intelligent mistakes” which are “actions based on a faulty premise, with the best of intentions, but with adverse consequences”. Dot highlighted five common unjustified assumptions when adopting test automation:

- ***automation should find lots of bugs***

Tests find bugs not automation. Automation frees up time to write them. Also, a test is most likely to find bugs when it is run for the first time so automating regression suites is NOT likely to find more bugs! So set both your expectations and the expectations of your management!

- ***automation comes out of a box (or download)***

Tools come out of a box. You have to build automation around them and Dot suggests you automate as much as you can, not just execution and comparison but also pre- and post- setup, metrics, etc. Also be aware that a poor “testware architecture” can lead to high maintenance. Dot advises adding layers of abstraction to divorce the tests from the tool by allowing for structured scripts using keywords (e.g. “make call” and “add contact” for mobile phone testing) to allow testers to think in a domain specific way.

- ***let's automate all of our manual tests***

Not all tests can be automated such as “look and feel testing” for a GUI. Also, it may not pay to automate some tests: e.g. taking 10 days to automate a test that takes .5 days to run every quarter? Also, if you automate ONLY manual tests, you aren't using the tools to their full potential.

- ***ROI is essential for automation***

Many people report advantages gained through automation such as faster time-to-market, faster feedback on internal releases, improved coverage and defect containment, etc. But these are benefits and not ROI. Dot cautions that ROI can be hard to evaluate and can cause automation to fall prey to short term goals. She advises on focusing on benefits, measuring improvements and making them visible.

- ***testing tools are testers' tools***

Dot advises that creating a test automation framework is more about software development than testing so it might be best performed by developers. Many a company has “lost a good tester and gained a poor programmer” [Hans Buwalda] when implementing the automation.

When asked if automation can mean fewer testers Dot asked “well, what type of testers do you have currently? Intelligent investigators or mindless morons?” as only the latter can be replaced by automation. This topic is covered well in comparisons of mimeomorphic vs. polymorphic actions. A recording of this talk from a previous conference is also [available](#).

The second speaker was [Dave Oxley](#), a Quality & Security Champion at McAfee. The title of his talk, “Anyone fancy a year off?”, reflected his tongue-in-cheek agreement that if his introduction of scrum saved 1 year of engineering effort then he could take a year off work. So did he get his holiday?

Dave first explained that before the introduction of scrum McAfee decided to put in place metrics to measure the impact of scrum. Dave highlighted the fact that they wanted a metric that meant something to the organisation so they chose “reducing the number of support calls” whose impact could be measured in financial terms and understood at a senior level. They then translated this overall goal into lower level goals. For each goal they defined a metric and a target for the improvement in the metric. Finally, they defined a limit for each goal: a level they **MUST** achieve before the code can be released. Below are some of the details on this:

| Goal | Measure | Target | Limit |
|---|---|--|--|
| Reduce number of defects found in system test | % of total defects removed in peer reviews | 20% of defects removed in peer reviews | 10% of defects removed in peer reviews |
| Increase early code reliability | % decision coverage of automated unit tests | 100% of new code | 80% of new code |
| Increase efficiency through automation | % decision coverage of automated system tests | 40% of total code | 30% of total code |
| Reduce manual test effort | Manual hours spent testing a release | Reduce hours by 10% | Reduce hours by 5% |

At this point Dave made some very important points about introducing metrics:

- Teams must be measured not individuals
- Plan for a monthly review of metrics in front of peers and executives

The result was that the team adopted agile which had a positive impact on both efficiency and quality. The team agreed to include the testing targets in definition of “done” and consequently development and QA worked more closely. For example, developers helped in testing to achieve done rather than racing ahead on features.

The details of the results are in the slides from Dave but the headline: the scrum saved 12 years of engineering effort across 6 teams! Unfortunately Dave has not claimed his vacation yet.

Ranjodh Singh of Dell spoke on “[Continuity testing](#)” which he defined as “testing built up by events and states of a particular feature within a product”. For a particular application, Ranjodh defines a

set of states, the possible behaviours and actions for each state and the next state. The following table gives an example for a flight booking website.

| Current State | Behavior/Action | Next State |
|--------------------|---|--------------------|
| Flight Finder Page | Pick valid "Type" Click "Passengers" dropdown. Select valid number. Click "Departing From" dropdown. Select a city. Click "Continue". New State: Select Flight Page | Select Flight Page |
| Select Flight Page | Check validity of information stated. Pick valid depart flight. Pick valid return flight. Click "Continue" New State: Book Flight Page | Book Flight Page |
| Book Flight Page | Click "Sign off" New State: Home Page | Home Page |

Ranjodh described a tool build in Excel where by adding the above information then the tool will automatically generate a set of test scripts to fully exercise the resulting state machine. This surely supports the layers of abstraction advised by Dot.

Peter Morgan talked on ["Quick wins in Agile"](#) noting that "Agile is not a destination but a journey". Peter highlighted sprint planning as the key to success. He reminded us to include testing in the planning and said that for planning of testing "High/Medium/Low" may be a sufficient level of detail in addition to the same levels for development; it depends (he sometimes uses planning poker cards for the whole team). Peter recommended having BIG visible picture to show progress which can help motivate the team and also engage the casual passer-by. Finally Peter talked about the 4 P's (the Positive Power of Peer Pressure) in the daily stand-up to ensure things get done.

The closing keynote continued with the theme of test automation. [Jonathon Wright](#), Global Head of Automation at Deutsche bank, 14 years global experience in automation and contributor to Dorothy Grahams 'experience in test automation' book explained how they are moving to a Test Automation as a Service in a Hybrid cloud in order to support their global team of over 300 automation testers. The [talk](#) gave us an insight into the complexity of their test automation requirements (i.e. to automate over 1500 applications in the next 18 months) which requires a global solution which can be grown quickly and temporarily according to needs. The slides have a lot of detail which explains how to implement such a global testing service and how to execute tests within the cloud.

All in all a very informative day and I'm already looking forward to the next event on [Tuesday 11th March 2014](#).

[Mike Bartley](#)