Intelligent Mistakes in Test Automation

Dot Graham

A mistake is where you do something wrong; an intelligent mistake is doing the wrong thing but for reasons that seemed sensible or logical at the time.

There are a number of ideas in test automation that seem sensible at first glance, for very good reasons. However, there are problems lurking below the surface, where what seemed like a good idea turns out to be a mistake. In this presentation Dot Graham will cover five of these “intelligent mistakes”:
- automated tests should find lots of bugs
- make sure you get the right tool
- automate all of our manual tests
- we must prove Return on Investment (ROI)
- testing tools are for testers to use

Knowing the pitfalls of what seems like a good idea is the first step to avoiding the problems.

Dorothy Graham has been in software testing for 40 years, and is co-author of 4 books: Software Inspection, Software Test Automation, Foundations of Software Testing and Experiences of Test Automation. Dot was programme chair for the EuroSTAR conference in 1993 and 2009, and has been attending the SIGIST since it began in 1989. She has been on the boards of conferences and publications in software testing, was a founder member of the ISEB Software Testing Board and was a member of the working party that developed the first ISTQB Foundation Syllabus. She was awarded the European Excellence Award in Software Testing in 1999 and the first ISTQB Excellence Award in 2012. She is currently working on a Test Automation Patterns Wiki with Seretta Gamba.

Anyone fancy a year off?

Dave Oxley

Why wait until you’ve released a product to see how good it’s going to be? With the right mind set, realistic but challenging goals and regular executive reviews we can drive significant improvements across releases. We can also predict prior to release whether we’re on track to release a better quality product than last time. This presentation will look at this process in the context of a team adopting Scrum for the first time. We’ll cover why, what and how to measure and show how big the engineering savings can be.

Dave Oxley is currently a Director of Quality at McAfee. Dave has spent the last 17 years running teams in development, support and testing. For the last 2 years he’s been focused on driving improvements in customer satisfaction across a business unit spanning 14 time zones, which involves a lot of travel for someone who hates flying.
Flowcharting for Software Testers
Graham Thomas & Phill Isles
[Workshop]

By now many of you will have heard about the Raspberry Pi, the $35 British computer that is helping schoolchildren to learn how to write computer programs. To date over 1.75 million have been produced. A real success story.

Some of you may also know that over the last 18 months I (Graham) have been actively trying to reconvert the world to using flowcharts.

Well now Phil and I have brought these two themes together in the form of a highly interactive flowcharting workshop presented using the Raspberry Pi and a programmable Robotic Arm.

This session should be informative, fun, and productive. Informative in that you will find out how really powerful a $35 computer can be. Fun because we will use the Penguins logic puzzle game on the Raspberry Pi as the basis for the flowcharting exercise. And productive because you will learn or relearn how powerful quick and easy it is to generate flowcharts to aid in your daily work.

To play an active part in this workshop you will need something to draw flowcharts with, be that notepad and pencil, computer, tablet or phone.

Graham currently works in two key areas of software testing; program test management and testing change & transformation. His current focus is on helping testers and the organisations he works with in transitioning to more agile ways of working.

Graham has extensive experience in IT across a number of industry sectors including; Finance (Banking, Treasury & Insurance), Utilities and Retail. This has been gained in software house, consultancy and end user environments. He has specialised in software testing since the early 1990’s.

Phil currently works as a Test Manager for HSBC Private Bank. He has worked in the field of software testing since 2000, mainly for banks, but has also had roles as a Tester in media organisations.

Phil’s software testing interests are in process improvement (by whatever means), and also in Tester education. He is a volunteer for the BCS and the UKTB for the ISTQB range of qualifications.

---

Quick Wins in Agile
Peter Morgan

You want to do ‘Agile’ but where do you begin? Or you are on a project that is supposed to be Agile, but is quite heavy-weight (or perhaps ‘clumsy’ is a better term). How do you as a tester bring your contribution to the team, and get real value to the business?

This session will bring some key items that can enable you to turn the corner when testing on an Agile project.

Please don’t expect a silver bullet to success; no two projects are the same and there is no single item that is universally successful.
Peter will call upon his experience in both traditional and Agile projects to bring real things you can take home:

- The nature of the stand-up – expectations vs accountability
- The whole team has to deliver. Testers do other things ..... and others do testing
- The 4 P’s: the Positive Power of Peer Pressure
- Agile should be fun, but it can also be quite scary
- Get developers to demo things to you
- Not every sprint has to deliver new functionality
- Not every delivery is implemented into PROD on its own
- Planning is the key

**Peter Morgan** is a freelance testing professional with more than 30 years’ experience in the ICT industry. His time has sometimes moved from testing to ‘development’, but he would add “always using the mind-set of a tester”. An enthusiastic speaker and author, Peter tries to base his output on hands-on experience, attempting to relate fine sounding ideas back to how it will affect Joe or Jane Tester in their everyday working lives.

---

**Test Automation in the Cloud**

**Jonathon Wright**

What is the future of test automation? The possibilities associated with cloud computing provide instant scalability, flexibility and availability for testing on demand with no upfront investment. This provides the industry with a perfect opportunity to utilise powerful test automation solutions.

The global testing cloud marketplace will allow for the joint collaboration of leading test specialists following industry best practice. This enables firms of all sizes to access the latest test approaches and methodologies whilst providing a unified platform for domain experts to represent business processes and user story acceptance criteria in a natural language.

**Jonathon Wright** has over 13 years of commercial automation experience with a number of international organisations. Currently working on providing Test Automation as a Service to a number of global clients.

Jonathon also contributed to the recent "Experiences of Test Automation: Case Studies of Software Test Automation (2012)" and a number of upcoming books on test automation and testing in the cloud. He is an active blogger on “Test Automation as a Service” (TaaaS.net) as well as presenting at various international testing conferences (StarWEST(California)/Fusion(Sydney)/ ANZTB(Melbourne)).
Test Automation Patterns

Dot Graham

[Workshop]

Many people encounter problems in automating test execution. Typical problems are: trying to get started with automation, unrealistic management expectations, such as too high a Return on Investment (ROI), or high maintenance cost for the automated tests. The bad news is that lots of people have problems like this with their automation. The good news is that there are solutions to these problems, which have been used by lots of other people and really work.

A “pattern” is a general reusable solution to a commonly occurring problem. Patterns have been popular in software development for many years, but they are not commonly recognized for test automation. Seretta Gamba initiated a collection of common problems (issues) and their solutions (patterns) which she and Dorothy are developing as a wiki.

In this workshop, Dot gives you a brief guided tour of some issues and patterns. You will then work with an offline version of the wiki (supplied on a USB stick to copy to your laptop) to investigate your own issues and identify patterns to help resolve them.

Dorothy Graham has been in software testing for 40 years, and is co-author of 4 books: Software Inspection, Software Test Automation, Foundations of Software Testing and Experiences of Test Automation. Dot was programme chair for the EuroSTAR conference in 1993 and 2009, and has been attending the SIGIST since it began in 1989. She has been on the boards of conferences and publications in software testing, was a founder member of the ISEB Software Testing Board and was a member of the working party that developed the first ISTQB Foundation Syllabus. She was awarded the European Excellence Award in Software Testing in 1999 and the first ISTQB Excellence Award in 2012. She is currently working on a Test Automation Patterns Wiki with Seretta Gamba.

Continuity Testing

Ranjodh Singh

Most of the defects captured by the test team, particularly which are above the line (ATL) defects are found using the expert testing method. However, the depth of expert testing varies between test engineers. This may lead to test engineers, being allocated a certain amount of time to perform expert testing but lack the capacity or idea to execute it properly. Another problem is that engineers who constantly perform expert testing may experience “burn out” whereby, it will be more difficult for these test engineers to generate new expert test scenarios.

Continuity testing will help expand expert testing capabilities by producing more chances to find issues and will assist test engineers to generate more robust test scenarios. The workings of Continuity Testing are that every interaction will produce a certain state. Hence, it will take the newly produced state of an interaction as the current active state and then interact with another event to produce a new state of the product being tested and so forth.

The result would be able to increase the test coverage of a product. It will also help to improve Phase Containment Effectiveness (PCE) and reduce Cost of Poor Quality (CoPQ) as more defects are uncovered earlier during testing. It would also allow test teams to improve its test efficiency rate.

Ranjodh Singh has over 10 years in software testing field and has held positions as a Test Architect and Test Program Manager. He currently works as a Test Engineering Advisor for Dell where he leads all QA and Testing activities from conception through delivery and maintenance. In addition to a Bachelor’s degree
in Computer Science, Ranjodh holds certifications in ISEB (Software Testing), PMP, ITIL (Expert) and CobiT.