Test Automation Problems – Survey results

In February/March 2014, I put out a survey on Test Automation Problems (using surveymonkey.com). The main reason was to find out whether or not the problems we thought were common in automation actually were.

The questions we asked are related to the "diagnostic questions" that we now have on the Test Automation Patterns wiki, which I am working on with Seretta Gamba. (See testautomationpatterns.org for a preview of an earlier version of the wiki and how to be invited).

I publicised the survey on Twitter, LinkedIn, Test Automation Patterns wiki members, and by email to my random newsletter contacts list.

A total of 183 responses are included in this report. Anyone who asked (and also left an email address – there were half a dozen who didn't!) has already received the results of the survey.

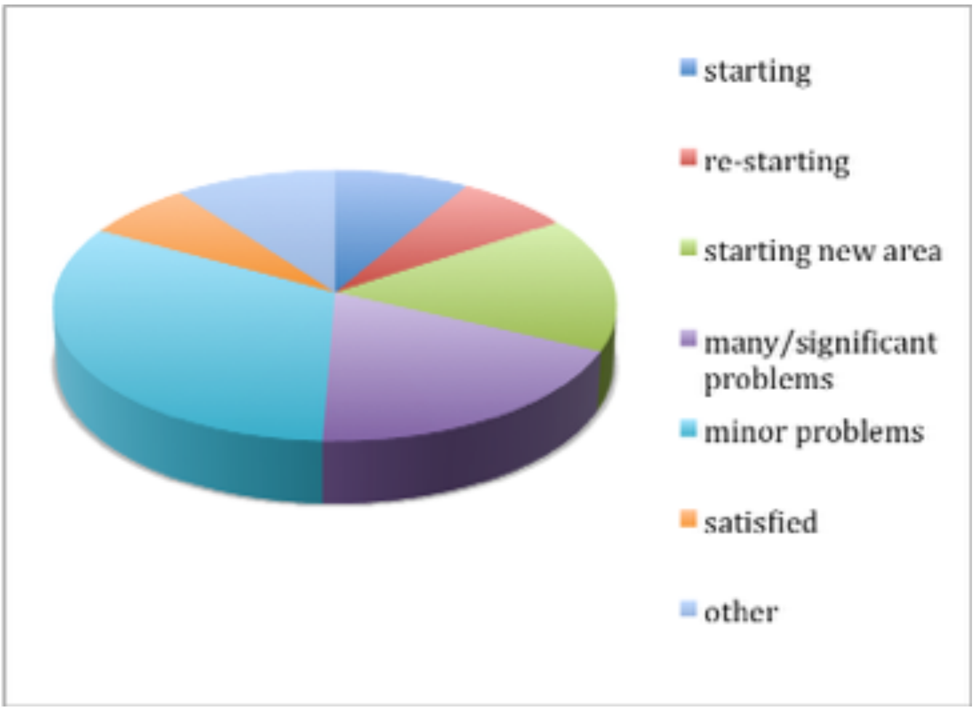
I asked where people worked: 32% of respondents were from North America, 13% from the UK, 10% from Benelux countries. There were also responses from Scandinavia, other European countries, Japan, Malaysia, Australia/NZ, India, Israel, Poland and Romania.

Note that in the questions below, percentages may add to more than 100% because people could choose more than one option!

I have picked out a few comments for each question, but have edited the text in some cases (for brevity & clarity).

The survey shows that there is a wide range of varied problems in test automation; no single one stood out as vastly more common. Many of the problems are related to expectations, management and "people" issues, but there are also significant technical problems. One surprise (for me) was that "unreliable execution" was the top problem in the category of unsatisfactory quality of automation.

Q1: Where are you in your current (system-level) test automation?



There are a lot of people in the "starting automation" stage: 12% first-timers, 10% re-starting after failing earlier, and 23% starting to automate a new area (GUI, web mobile etc.), for a total of 45% (which may include some overlap).

However, 27% of people had significant (7%) or many problems (20%), and the winner in this category are those with automation that works pretty well but has minor problems (45%). The 10% of people who are completely satisfied with their automation does include some automation experts known to us!

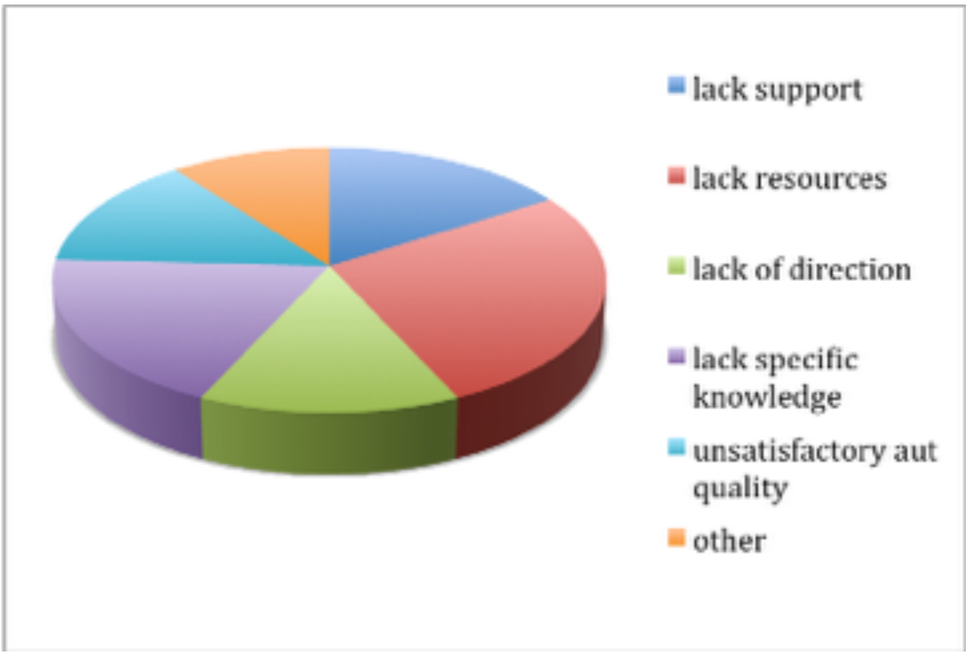
Here are some of the comments (showing some frustration):

"Current automation runs satisfactorily but running on automation software which has now become obsolete and unsupported. Awaiting purchase approval for a different product, to start again from scratch. (but LONG wait! Over a year!!)"

"Tester's creativity and ideas have been put down by upper management."

"We did start it up - but it went 'out of interest'. The reason was largely that the boss didn't want to listen to the advice from several testers and developers about what automatic regression testing can and can't do. Finally we ran out of money and had to limit ourselves to manual regression testing and defect verification."

Q2: What problems do you currently have?



We were a bit surprised to see that the responses cover the range of automation problems fairly equally, although there was a clear "winner" in lack of resources (57%), with lack of specific knowledge 2nd at 41%. The others were all roughly 30% with other at 22%.

Comments included one poor soul who said,

"We have gone too far down a certain path to be able to change."

I would challenge that – change might be difficult, but without changing, your automation is probably doomed (maybe that's what the respondent meant.)

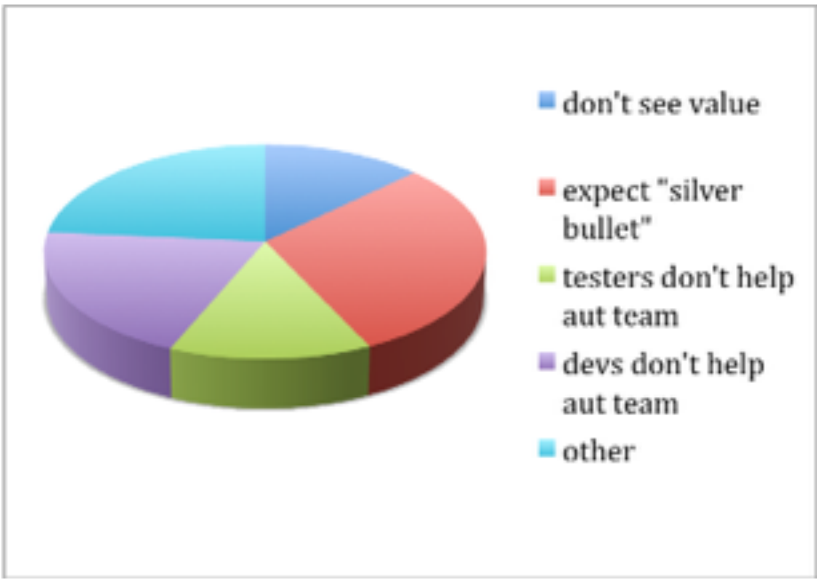
Other comments:

"There are a lot of people who think they know what to do, because they read something on the internet. Therefore it is sometimes hard to make a plan, convince the boss and stay on track. Typically the bosses don't want to hear about the cost of maintaining a test suite, since the tool vendors typically promise something that other people have to deliver ..."

"Lack of discipline. The BAs and developers know they should do it, and (mostly) have the skills, there is the direction from the top (at least from the top of the development org chart) but they lack the discipline and/or culture to do it consistently."

We then asked more specific questions about each of these major problem areas, in Questions 3 to 7.

Q3: Lack of support:



The “winner” in this category was managers expecting a “silver bullet” (54%), but developers not helping the automation team came out at 39%. The Other category included specialists not helping the automation team and no-one helping new people.

Comments:

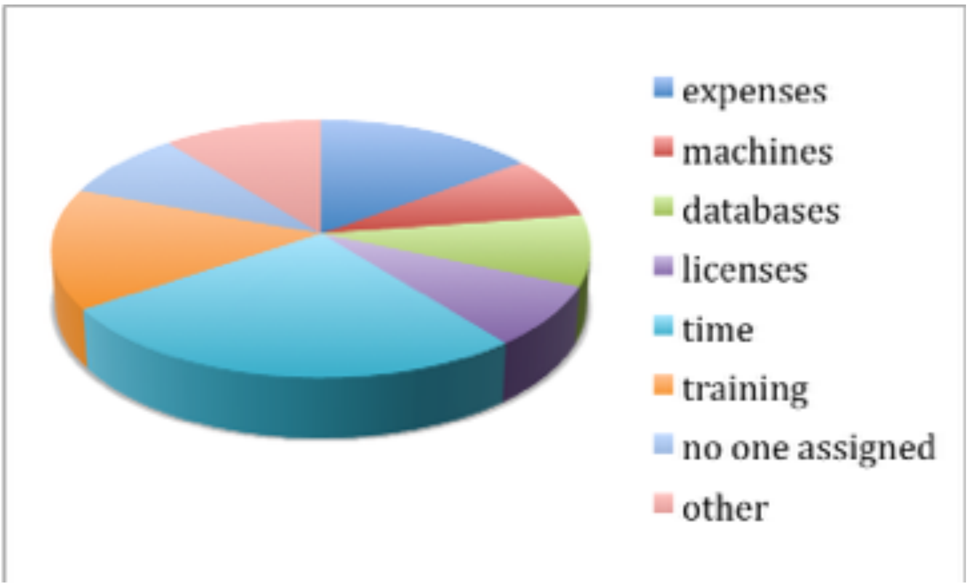
“There is management support at the engineering level, however higher up the chain we are getting resistance (from product owners, stakeholders) who fail to understand the importance of test regression, and therefore feel that money spent upfront in building a regression suite is waste.”

“As for management, ROI seems to be [in the] driver's seat all throughout the program. Team looses focus to create quality test suites with good test technique.”

“I think everyone sees the benefit of automation but if you use vendors, particularly offshore then how are they incentivized to do process improvement (including automation)? It's easy to demonstrate the advantages of automation but it requires other people to be onboard to ensure that manual test cases are good quality, and the test environments (particularly complex ones) are in a state that you can develop automated script successfully against them.”

“Very often it is an 'either-or' when you try to sell automation to the project management; you may have Automatic Unit Testing together with Continuous Integration or you may have Automatic Regression Testing on the UI or you may pursue exploratory testing. In general; the test project will rarely be allotted sufficient resources to implement a successful automatic testing. Therefore successful automators are hard to find.”

Q4: Lack of resources



The major factor here is no time or insufficient time being planned for automation work – 63%. The next two, training for automation not planned, and expenses not budgeted were at 37% and 35% respectively. The rest were approximately 20%.

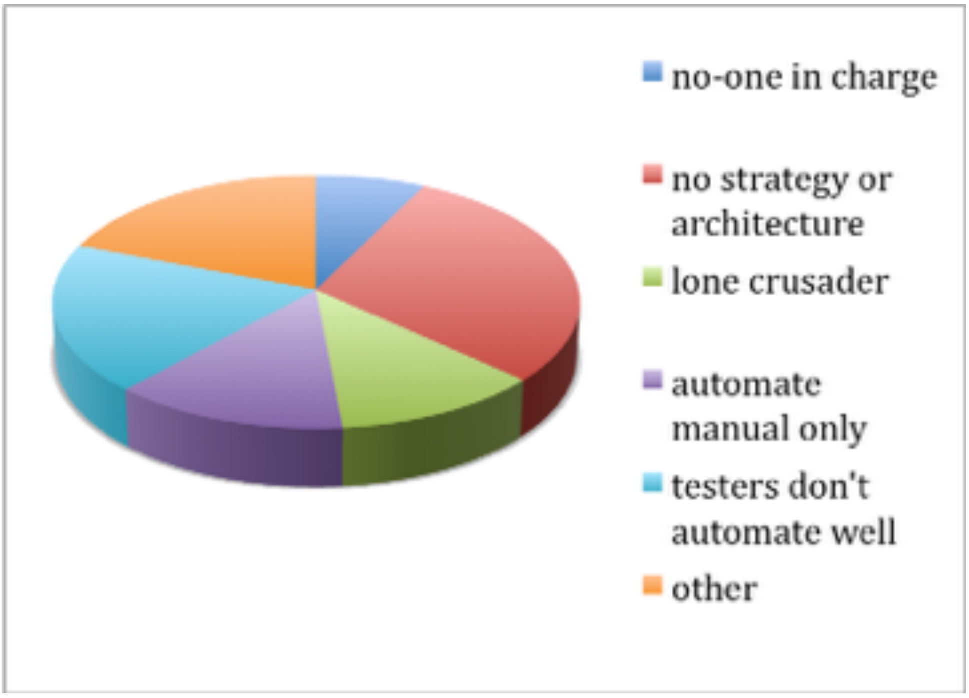
Some interesting comments in this section:

"If the entire IT group worked on automation for 6 months, we'd be in parity with development."

"Automation doesn't work well in a project-driven organisation – project managers don't see beyond their project."

"While you're busy delivery [good automation], you forget to watch your back and run into a political ambush – being considered an 'expensive error' by the next 'rising star'."

Q5: Lack of direction



For this question, respondents needed to choose the one option that best applied, rather than being able to tick as many as applied (so the percentage values are lower).

The most common problem here was automation being done without a strategy or architecture (27%), followed by testers doing automation don't know how to do it well (19%). Only 8% voted for no-one being in charge of automation, everyone does what they want.

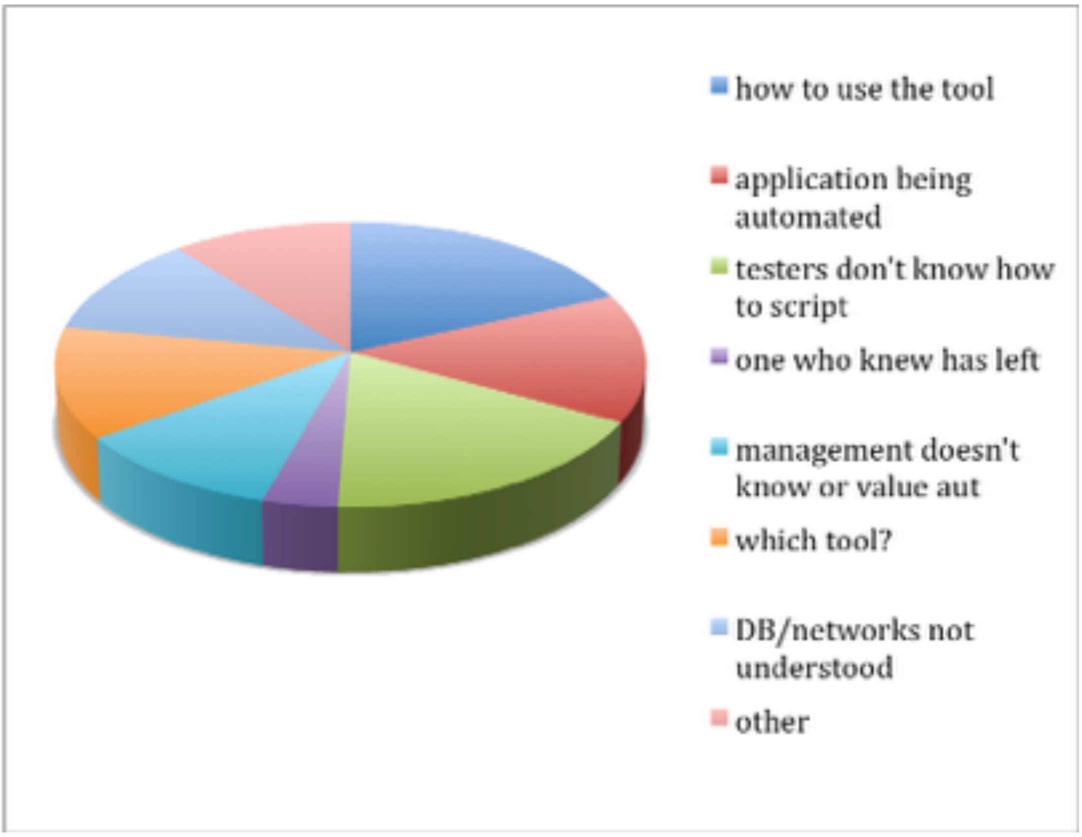
Comments:

"Testers/management think they have to automate the manual test cases as is, and hire an automator to do that job - any issues resulting from this setup are blamed on 'bad automation'"

"Perhaps it is not 'lack' of direction but more 'wrong' direction as it relates to an earlier statement: "Managers expect a silver bullet" and therefore insist on automating basically everything (which is obviously not possible, nor efficient let alone useful..) "

"Each application group is doing their own thing, multiple tools and frameworks with no overall architecture for cross-platform, cross-application automation. "

Q6: Lack of specific knowledge



The two highest responses here are The people that are to use the tool don’t know how to use it (38%) and Testers are supposed to write automation scripts but don’t know how (34%). For 30%, the automators don’t know the application [whose tests] they are automating. The least popular response was 9% where the only one who knew about automation has left the company. The remaining responses were all between 21% and 25%.

Comments:

“People requesting automated tests have unrealistic expectations of what can be automated or how long it will take. “

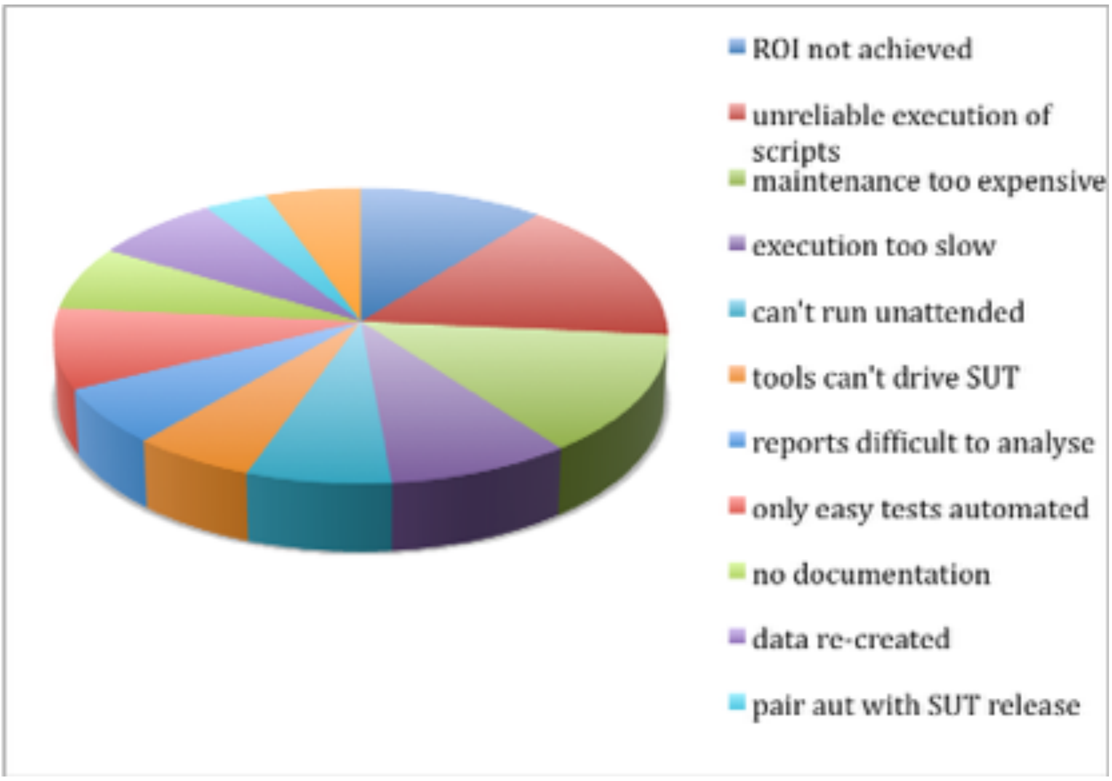
“The tool is selected by the umpire: 'Look - we HAVE this tool. Make do with that - somebody told me that it can get the job done, so I don't wanna hear that YOU can't do it...'”

“The testers are generally well-informed but lack management and governance. It is also a large test team moving towards DevOps and Continuous testing but are constantly beset by co-ordination problems, a lack of direction and real management and DB set up problems. (Each test environment is over 100GB because they are dealing with large media files) “

“New application being built and automated at the same time. Issues with build constantly changing, application knowledge, disconnect between vendor builders, scripter and automators. We are still working out how this process will work best. “

“How to measure effeciency of automated tests? “ (a very good question!)

Q7: Unsatisfactory quality of automation



The most common problem here was a bit of a surprise (to me) – it is Automated scripts don't execute reliably (50%). The runner-up in this category is Maintenance of automation is too expensive (44%). The 3rd is Expected Return on Investment (ROI) has not been achieved (34%). Two factors were voted in by 28% of respondents: Execution of automated scripts is too slow and Important tests have not been automated, only easy ones. The rest were all approximately 20%.

Comments:

"The biggest problem by far is inadequate requirements management."

"Most test environments require too much simulation behavior (mocks) which means that coverage is rather low or people do not trust the results."

"The above all depends on vendor and quality of the resource. Typically where I have used a UK test company the quality is generally good, but where the work is predominantly done offshore then they struggle with key basic concepts. Offshore SHOULD only be used for factory level development of basic scripts, not for any automated scripts which require some clever development or for any innovation. What I have experienced is the common sense element is just not there, but where it has worked is where the framework is developed onsite and then basic stuff is rolled out offshore where the manager goes offshore as well to manage the work."

Automation books

I listed six books on test automation, and asked if people had heard of them, read them, and/or recommended them. I was pleased to see that my own two books seemed to be the best known, most read, and most recommended! But then, I did send this survey to a lot of people who I already knew! Each book was recommended by at least 3 people (and mine by over 20 - ☺)

Thanks to one respondent for this:

"Experiences of Test Automation: Case Studies is probably the only other book to do a decent job balancing the technical side and the management side."

There were also a number of suggestions for other books on automation, as well as mentions of blogs, etc.

- The "A" Word, Alan Page (download from leanpub – new to me, interesting short blog posts about automation, including why not to do [much] GUI automation – well worth a read)
- Automated Testing Handbook, Linda Hayes (re-published in 2004 but looks to be the same content as the 1995 version I have. A good summary of automation principles – good for high level managers?)
- Lessons Learned in Software Testing, Kaner, Bach & Pettichord (very good book on testing with useful automation wisdom too)
- How Google Tests Software, Whittaker, Arbon & Carollo
- Testautomatsierung, Seidl, Baumgartner & Bucsics

The following books are more technical; I am not familiar with them.

- .NET Test Automation Recipes, James McCaffrey (mainly good reviews)
- Next Generation Java Testing: TestNG and Advanced Concepts, Cédric Beust & Hani Suleiman. (written by the author of TestNG, mixed reviews.)
- QTP Unplugged, Tarun Lalwani
- Java Power Tools
- Django 1.1 Testing and Debugging
- The Cucumber Book: Behaviour-Driven Development for Testers and Developers, Wynne & Hellesoy
- ATDD by Example, Gärtner
- Continuous Delivery, Humble & Farley

Thanks to all those who took the time to fill in the survey!

Dorothy Graham, 27th March 2014