ARE WE AFRAID OF TEST AUTOMATION

After having spent the last 9 months working on a test automation implementation for a team I am not running myself I have started to wonder what are the main success factors for test automation, besides the obvious: achieving the goal you set out in the first place?

One of the main factors that will help test automation be a success in your organisation is to not consider test automation a goal, but use it as a means to achieve a goal. The goal should **not** be something like "have x% of regression automated", your goal should be something in which test automation can be a means to achieve this goal, for example help free up time for the testers to focus on important things rather than having to spend most of their time in regression testing.

Another very important thing to keep in mind in implementing test automation is that you need to keep a sharp eye out for trying to automate everything. Technically it may be possible, but it is hardly ever a good idea to want to automate everything. Quite some things require human intervention or human interpretation and cannot be simply made into a Boolean (which is what test automation is in the end, something is either true or false).

Look & feel testing, or validating, should in most cases not be automated for example. This for the simple reason that, despite it being possible, will more often than not raise either false positives or more likely, false negatives. Since these tests often require some form of image recognition and comparison a change of screen resolution or a small CSS change (font difference for example) will make the test fail, resulting in either maintenance or tests being made redundant.

For me however, the main show of success is that the automated tests are actually used and maintained actively.

Having a test automation framework and relying on it to do its job is not good enough. Just like every other software automated tests also want some TLC or at least some attention and maintenance.

Something that is still often seen in test automation is that the tests run but with non-determinate errors, in other words errors that are not really due to the testcases but are also definitely not a bug in the system under test.

If you show some tender love and care to your automation suite, these errors will be spotted, investigated and fixed. More often however these errors will be spotted, someone will “quickly” attempt to fix it and fail, after which the “constantly failing test” will be deemed useless and switched off.

Besides non-determinate errors there is another thing I have seen happen a lot in the past.

Some automation engineers spend a lot of time and efforts building a solid framework with clean and clear reporting. They add a lot of test cases, connect the setup to a continuous integration environment and make sure that they keep all tests working and running.

They then go ahead building more and more tests, adding all kinds of nice new tests and possibilities. What gets forgotten often however, is to involve the rest of the department, they do not show and tell to the (manual?) testers, they do not share with the developers. So the developers use their own unit tests for themselves and if they do some functional testing they do it manually and sloppily. The (manual) testers go about their usual business of testing the software, as they have always done. Not considering that a lot of the test automation suite they can use and abuse to make their life easier. They will spend time on data seeding, manually. They will spend time on look and feel verification on several browsers, manually.

All this time the developers and (manual) testers could have been using the automation framework and the existing tests in it to make their life a lot easier.

While writing this down it starts sounding silly to me and unlikely that it happens, yet I have seen this happen time and time again. What is it that makes developers but especially testers afraid of hesitant to use automated tests?

I love using test automation to make my life easier when doing manual testing, despite having an very strong disliking for writing code.

Test automation should be seen as a tool, it is there to make life a hell of a lot easier. It cannot replace manual testing, but it can take all the repetitiveness and tediousness of manual testing away, it can help you get an idea of what kind of state a system is in before you even start testing, but it can also help you, a lot, in getting the system in all kinds of states!

Source : http://martijndevrieze.net/2012/02/09/are-we-afraid-oftest-automation/