Automated Testing vs Checking

Over the last decade I have both seen and experienced first-hand the same issue in different organisations - Testers are seen as manual workers who are the cause of most problems (a bottleneck if you will) but are useful when needing to assure stakeholders of product quality. However, this is normally the first department to either be deprived of resources or have cuts if things go pear shaped.

The fault partly lies with the educational system; nobody prepares us to be testers. In most cases, someone falls into or discovers the role by accident. They learn hands on that the role can involve doing multiple jobs - being a combination of anything from customer support and infrastructure maintenance to release management and even development.

This is because the testing curriculum has only surfaced recently, defining the responsibilities of a tester and how the role can evolve within a business. For instance, courses by International Software Testing Qualifications Board (ISTQB) have become popular among testers, and companies are starting to seek potential employees with such qualifications. Of course, there are other testing schools - such as Rapid Software Testing (RST), which was started by James Bach and Michael Bolton, and the Context Driven Testing (CDT) school developed by James Bach, Brian Marick, Bret Pettichord and Cem Kaner to name but a few. I prefer the RST and CDT as they are, in my opinion, developed by people who are not afraid to learn and refine their material, whereas conventional educational establishments tend to stick with the same material for years - even if it’s no longer relevant or even wrong. The beauty of the RST and CDT is that they offer free material online that is also taught (for free) at testing community meet ups.

Even though we are slowly seeing some changes in the educational system and what it means to be a tester (or to be a “qualified” tester), many of our employers are still unaware of the depth of the testing role. It's our responsibility to initially educate ourselves and share this information within the organisations we work for.

Today I’d like to share my thoughts on why we should be referring to automated testing as ‘automated checking/verifying’…

I’ve spoken to many colleagues and potential employers, and read many blogs and books about testing, and one thing that stuck out to me was that the act of following test case instructions is considered by many to be testing. To me following instructions is simply an act of checking or verifying the outcome.

As previously mentioned, we need to educate ourselves, and what better way is there to do this than by attending meet-ups, reading blogs and books and speaking to peers. One of the most important ingredients in self education is critical and quizzical thinking - I would highly recommend applying an ABC principal (Assume nothing, Believe no-one and Challenge everything) when learning something new.

I have used the above to define what the following means to me…

* core basics of software testing: Exposing vulnerabilities in the system (the what) using various techniques and tools (the how) in order to ship quality product (the why).
* automated checking: Performing regression checks (the what) by having a machine follow instructions a tester has written using automation frameworks (the how) to ensure old features are still working and new ones haven’t introduced any bugs (the why).

In summary, as soon as a test case has been created and executed by a tester it becomes something that has been checked - let’s call this a ‘check’. The test case is created by a tester while exploring the system and/or learning about it by going through the requirements or talking to stakeholders etc.

To this day, a machine can’t think about the information it is receiving and act upon it in the way that a human would. Elements such as intuition and external environmental factors come into play that allows us to create test cases that a machine would otherwise miss or not think of at all.

It’s a long journey for us and there are many challenges to be faced as there’s currently a disconnect in the testing community in terms of understanding what automated testing is (or isn’t) and what it should be called. For us to fully understand a constantly evolving subject such as automated testing - which we are still learning about now and working out best practises and what this term actually means - we must always be applying our knowledge and redefining what we know - then we can preach what we practice!

As I write this article I have come to realise that there is a subtle difference in automated testing vs test automation, but we can talk about that another time…