

TEST AUTOMATION – FINDING YOUR STARTING POINT

The other day I had a job coaching of a colleague who, for his assignment, needs to start setting up test automation within an agile environment. He is very interested in test automation and the possibilities it gives you, however his knowledge on the subject, as well as on coding, is very limited.

In preparation for this first coaching session with him I was pondering where to start with him and I ended up starting with the, to me, obvious points :

- just start with a first step
- KISS
- DRY

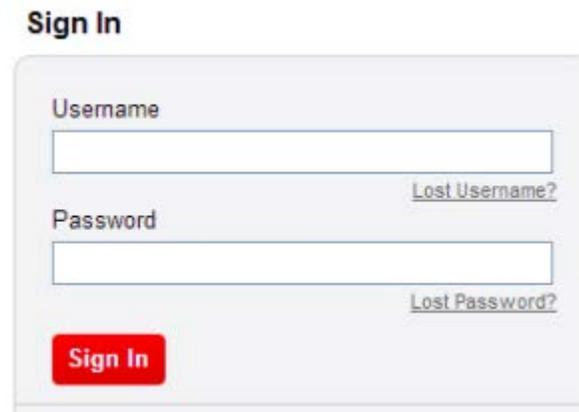
Just start with the first step

My colleague was struggling on how to start and more importantly, where to start with test automation. This is an issue I have faced myself often in the past as well. You have this huge task of automating a system laying ahead of you, there is so much to do and so little time.

Where do you start?

Over the years I have come up with a simple strategy for that: just start with the basics and move on from there.

In other words, start with the basic functionality you will need for any test you are about to execute. In most systems that is setting up a simple sequence for validating the home- or landingpage, going to a login screen, validating that loads properly and then logging in.



Login

Once you have this start it often becomes easier and more logical to move ahead and start automating other functions and functionalities. In this particular case once logged in there is a landing page which consists of a dashboard, so we scripted the verification of all required items on the dashboard. Once this was done my colleague again was questioning, what do now?

Again, we kept it simple and straightforward, I asked him to name another functionality he often uses and which help add value to the test immediately and make life easier in testing in general. He came up with logging off.

Over the course of this hands-on coaching session we ended up writing several scripts, which when put together in a scenario ended up being a very simple and fast sanity check on the

application under test, which can immediately be reused in Continuous Integration and on every new code drop they will be doing on any environment.

Keep it simple and don't repeat yourself.

Once we got through the initial start we went on to two other very important things you need to keep in mind when automating (or writing any code for that matter): KISS and DRY.

KISS



Keep It Simple

KISS is well known to mean “Keep It Simple, Stupid” the main thing I generally take from this is the “Keep It Simple” part. For test automation it is important to keep things simple, you need to be able to maintain the suite, others need to be able to understand what you’re doing and why you did things in a certain way. Therefore, I rigorously stick to keeping things as simple as possible considering circumstances. I generally try to follow the KISS rule in all aspects of test automation: naming, coding, hierarchy, refactoring, running etc. Some of the ways this is visible is: I try to keep scripts short and concise, I try to ensure names of scripts clearly state what the script (supposedly) does, I try to keep variables I use clearly names (e.g. I try to stay away from statements like `i = 1`, instead I give `i` a meaningful name).

DRY

DRY, or Don't Repeat Yourself, is a well known practice in software development and should be used within test automation as well. I apply DRY both in the scripts and in my daily work. When starting with automation on any project some of the first things I generally put on my list of things to automate sooner rather than later are all functions and functionalities I have to use more than rarely. In other words, starting an application and logging in I generally automate as quickly as possible, even on a manual testing project!



Don't Repeat Yourself

One of the reasons I am a strong advocate of test automation is that I am lazy. Why would I want to start an application and login to it manually and thus have to execute a whole bunch of things myself, several times (sometimes tens or hundreds of times a day) when I can automate that same flow in a few minutes and not have to deal with it anymore? Quite often starting and logging in of an application, when automated, is faster than when you do it manually as well, so it not only saves me from repetitive work, it also speeds up my work!

In other words, the DRY principle can also work outside of the test automation code, in your daily work. If a task is repetitive, chances are you can automate it!