

A Framework for Test Automation

Automated testing increases the speed of development allowing teams to spend more time on tests that require more detailed manual testing. Higher test coverage offered by automating your testing increases team responsiveness, better user experience, higher customer satisfaction, and ultimately enhances product/release quality. However, to make sure that your automation strategy produces these results, you should be aware of these simple yet effective steps before moving ahead with your test automation strategy.

With over 20 years of experience, Mike Maliska, Senior QA Manager and Automation Architect at PrismHR, has gathered a lot of helpful knowledge when it comes to test automation. According to Mike, here are the **5 essential tips you need to read before automating your testing:**

STEP 01 | Clearly define the steps

Mike's tip: "If you're in the middle of development and you don't know how the feature is going to work, it's not good for automation."

When starting a process with automation in mind, think through each step it takes to get the end result to define tasks that need to be done during development.

If you have clearly defined steps, then you have a good candidate for automation and team members are able to make better assessments for completion with measurable results.



STEP 02 | Make sure your steps are repeatable

Mike's tip: "Can you perform those steps, the same way over and over and over? If the answer to that is 'YES', that's a great candidate for automation! If the answer to that is 'NO', then it's probably not a good candidate for automation."

Start by determining the repeatable tasks, there's no sense in automating a test that can only be run once. This frees testers from repetitive manual tests giving them more time to deal with complex features that need manual eyes.

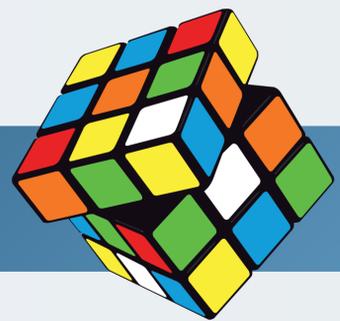


STEP 03 | Work out all of your business logic beforehand

Mike's tip: if you catch yourself saying, "Sometimes we run it this way and sometimes we run it that way and sometimes, if it's a Tuesday after a full moon we add tests." then your business logic is not worked out and your automation is doomed to fail.

Working out all of your business logic beforehand allows testing teams to prevent issues in quality earlier on. It is important to be able to justify how your business goals will be achieved through automation and how you will measure them.

Focus on process insights and optimization, ask yourself, "How do you run it? When do you run it?" Business logic is the means that helps QA achieve its testing strategy end goal: to make sure that the product does what it is supposed to do.



STEP 04 | Make sure you know if your steps are being done manually and by whom

Mike's advice: "The whole point of the automation is to remove manual steps to free up that manual person to do more user acceptance tests. There are plenty of things during a software test lifecycle that require manual eyes, so the less a user is doing steps over and over and over again, the more they can actually spend on the quality of the product."

Automated tests execute and record the proper steps every time to free up manual testers to spend more time on tests that need manual eyes.

One great example: With usability and UX, manual testing is as necessary as ever. Testers manually testing UX use their intuition and instincts to pick up on broken or inadequate things that automation cannot.

Effective management should always invest in the quality of their products and making sure you know if your steps are being done manually and by whom is one way to do so.



STEP 05 | Figure out who will have ownership of the automation

Mike's advice: "I've seen automation get spun up and it works great for three months and you move on and then all of a sudden it falls down."

So ask yourself, if that test fails, who owns that? Should QA go manually verify that that test failed? Who owns the maintenance? Who owns the upkeep? Who owns the results?

Make sure you're agreeing on who will own specific areas such as scripting, maintenance, and yes, even failure. Identifying this upfront will save you time and hassle in the future.

If you really want to make sure you're in a position to successfully automate your testing, you have to be prepared to answer the question of who will own automation going forward?

