OnPath

Automation Testing: Are you ready to start?

Contents

Introduction 3

Understanding your postition 4

Have you got what you need for automation? 4

Why do you want to automate? 5

What do you want to automate? 6

Considerations to make before implementation 7

Scaling up 7

Is the testing tool compatible with your current tech? 8

Is your testing environment stable? 8

Reporting and metrics 9

Future maintenance 9

Remember, automation isn't everything 10

Deadlines are fast approaching. Your QA teams are racing through their manual testing phases to get the product out the door. You're getting the job done but you're working long days (and sometimes nights) as launch approaches. You're itching to branch out into automation testing. You're sure it'll speed up your delivery time and further reduce errors.

Wait. Remember what Jim Hazen said, "It's automation, not automagic."

Before you dive in, it's better to build a clear idea of what automation will do for you and assess whether you're ready for it. From our experience at OnPath Testing, there are several key factors that will tell you if automation testing is right for you.

Let's help you assess your current capacity, understand whether your QA teams are ready for automation and guide you towards implementation.

Understanding your position

Have you got what you need for automation?

Take a look at your current QA capacity and evaluate your resources from an automation perspective.

Ask yourself:

- **How budget conscious am I?** If you're running on the idea that automation testing is going to save you money, determine the upfront costs. Have you considered the price of the tool? While some automation tools like selenium offer free versions, others might have one-off license payments or subscriptions to consider. What about staff costs, such as hiring or training? Finally, getting your automation testing underway will take time, that will cost you too.
- **Do my processes match my people?** Analyze the hard skills you've got on hand. Do your people have the know-how required for implementing automated testing? Think about how you want to scale and execute your ambitions for automation as well. You're not borrowing someone just to get things rolling, you'll need dedicated team members for the long-haul.
- **Is our current staff capable of executing the automation?** You might intend to integrate automation testing alongside your current manual testing. You might be looking to augment your manual testing with entirely separate automated test cases. However you choose to implement it, consider whether current staff will be running the automation. Do you need to hire specialists?
- **Have we looked into training?** If your team is lacking hard skills, will you offer training to get your automation testing off the ground? If so, where is that training coming from? Have you considered the cost?

Why do you want to automate?

It's important to articulate why you want to automate. How will it benefit your projects? Will it bring you closer to your goals?

Reasons to automate may include:

- **Decreasing execution time.** Automation testing allows you to execute consistent and regular test cases, saving time and effort. The result will be a reduced time-to-market and happier stakeholders.
- **Greater test coverage.** Time is no longer your enemy. Automation testing means you can test more features in the same, or less, time and with reduced impact to the team.
- **Struggling with errors.** Manual testing is susceptible to human errors and that's lengthening your testing phases. Automation testing will accelerate this process and improve your overall accuracy.
- **Ability to reuse the test suite.** Once you've built your automated tests, you will reuse them for other use cases. This will help you with future projects.

What do you want to automate?

Identify what is important to the business. What takes a lot of time to execute, but is done by rote? What gives you the most coverage for the recommended user flow? What is an area of focus for sales?

You also might choose to automate these pre-identified suites:

Integration testing. Your developers are eager to find out how their code works with other components. Automation will help you speed up the testing process, so they can get on to fixing bugs quicker. This reduces feedback cycles and keeps teams progressing towards completion.

Functional testing. An automation tool won't skip any steps here and makes sure data is always inputted correctly. It'll also enable you to run sequential tests with different data and compare the results in a much shorter timescale than manual testing.

Performance testing. Automated performance tests will keep your stakeholders happy as it avoids launch failures and performance regressions.

Considerations to make before implementation

You've gone through the questions. You've got a strong understanding of your current position. You know you're ready for automation testing. You know why you want to automate, and you know what you want to automate.

What's next?

There are a few things to consider as you move towards implementation:

Scaling up

You might face a top-down approach, with executives dictating where you should start. But the direction is up to you.

There's no point making changes where you won't see value. Thinking back to the Pareto Principle, that means hitting those 20 percent of causes hard. You can identify those causes with an impact analysis - look for areas where you can have the most impact with the least amount of effort and start there.

Is the testing tool compatible with your current tech?

In an ideal world, your chosen automation testing framework supports all applications, platforms and operating systems. However, sometimes there are compatibility issues.

It might seem obvious to match the tool with your current stack, but there's a bit more to consider here. Is your testing tool adaptable to possible changes? If you're choosing a specialized tool with limited compatibility because it's just right for the tech you have on-hand, you may want to consider how this will affect your future plans.

Is your testing environment stable?

To get the most out of automated testing, you need a stable environment. Unstable test automation can produce false alarms in reporting meaning you lose some of those potential time savings searching for errors where there are none.

External factors that may affect stability include:

- Suboptimal test environments
- Poor hardware performance
- Network issues
- Incorrect configurations
- Slow servers

These factors project issues onto your software, creating inaccurate results and anomalies.

Reporting and metrics

What equals success? What's important for you to measure?

Typical metrics to capture may include:

- **Total test duration.** How long are your tests taking? Is the difference in time between manual and automation negligible or tangible? Does it offer any additional opportunities?
- **Percentage of tests passed or failed.** A simple, but valuable metric to track your testing progress across each iteration or release. It's also an easy metric to convert to graphs and charts—detailing the tests that passed, those that failed, and tests yet to run.
- **Number of defects.** Knowing the number of defects in your tests can help you with predictive modeling by indicating what other issues are left to dig up and how different testing environments affect the outcome.

Make sure you have a baseline comparison for each metric. You'll know if your automation tests are having the desired effect or making anything more difficult.

Future maintenance

Understand the maintenance costs in your implementation.

Build your automation framework with the goal of minimizing maintenance. Keep the code in your automation scripts streamlined. Avoid repeating code and write only what's necessary to complete the task. Keep it simple where applicable.

Remember, automation isn't everything

Ray Bradbury once said:

> **"Life is about trying things to see if they work."**

and your automation testing is no different. If things aren't working out how you expected, don't panic. There are many other avenues to get you to your QA testing goals.

Here at OnPath, we've found time and again stopping to consider these points before leaping into automation is invaluable. They help you understand why you're implementing automation testing and ensure you're doing it in a way that supports your goals.

If it helps to sit down and talk it over with us, get in contact today.

Contact us