

# How to Achieve Faster, More Efficient Software Testing in 2020

You may save time during the software development lifecycle if you reduce testing or skip it altogether. But this approach will likely add time, cost, and aggravation down the road. Thankfully, there are many ways to speed up your testing process so you can achieve both speed and quality.

Whether it's establishing a more consistent testing plan, [hiring a quality assurance outsourcing firm](#), or simply shifting the way your team conceptualizes this step of the process, you're likely to reduce time and increase efficiency by a large degree.

## Create a Culture of Quality

Consider the difference between a team that views testing as a function separate from development that just takes up time and resources and a team that understands testing as an integral part of a process that ultimately leads to the highest-quality products. The latter group is more likely to take a collaborative approach, follow efficient processes, and engage in shared company success.

The key to developing this type of team is communication. Make sure all employees, not just those in development and testing, understand the beginning-to-end process of how your company's products are made, including market testing, initial design, funding, and sales. Tell them how each job fits into this process and how each person contributes to the company's success.

## Improve Processes

Just as you would have a workflow schedule for development, have one for testing as well. This approach helps ensure that you go through all the steps you need to follow for each project and that you don't have to waste time coming up with a new plan for each one. You may want to include the following elements in your schedule:

- The types of tests you will perform
- Performance criteria and KPIs
- Whether your testing will be manual or automated
- Whether you'll use emulators or physical devices
- Your methodology
- The number of testers needed
- Project-specific factors

Adjust your plan when you encounter more efficient steps. Regularly review the entire plan to ensure it still meets the needs of your operation.

## **Hire Additional Testers**

If you find that the testing process requires more staff than you have available, consider hiring additional testers. Not sure you'll need them for ongoing work? Think about hiring a software QA outsourcing team instead. With an outsourced team, you get the ability to ramp your operations up or down depending on the number of projects you have going at any given time.

With outsourcing, you also get quality services from highly trained and skilled professionals who are up to date on the latest testing methods. Hiring an outsourced team can take a bit of effort up front. You'll need to ascertain your needs, search for the best agency to meet them, and conduct interviews to find the right one. But it's time well spent because you'll end up with a valuable resource that will save you time and money in the long run.

## **Implement Continuous Testing**

With ever-shorter delivery cycles, you should complete testing as quickly as possible. Yet, with the need to remain competitive, you shouldn't risk compromising quality. You can reconcile these opposing needs with automated continuous testing. With this approach, you run tests every time an app's source code is updated rather than waiting for later milestones.

With this method, developers, operations, and quality assurance professionals all collaborate to efficiently monitor user experience, functionality, and marketability. The process is integrated into the full development cycle and you get feedback early enough to make changes before problems become unmanageable.

If your team doesn't use continuous testing now, transitioning to this model may require a bit of a culture shift at first. So, provide enough training and change management up front to ease employees into it.

## **Improve Automated Testing**

As noted above, automated testing is important within a continuous testing paradigm. It's also useful generally because it's more efficient than manual methods for regression, performance, and compatibility testing. Automated testing is also faster and more accurate than manual testing and can be performed without human involvement.

While automated testing achieves these benefits and more, it also requires up front investment in the form of writing scripts and setting up environments, as well as hiring and/or training automated testers and purchasing a license for an automated testing tool.

Keep in mind, though, that testing doesn't have to be done exclusively one way or the other. You can use automation to perform more tedious tasks and human testers to perform those that deal

with user experience. This approach takes advantage of the capabilities of both types.

### **In Summary**

If testing is a bottleneck holding up efficient software development and delivery, there are ways to speed it up. The methods listed here are just a few to help you prioritize both speed and quality. The key is being open to new ways of thinking, using available resources, and planning your work.