Continuous Testing VS Automation Testing

Md Rezaul Hasan

May 31, 2020

Contents

1 Introduction

In many IT organizations there is a "Wall of Confusion"[2] between development and operations. On the one hand , development wants change, but on the other, operations wants stability. Between them is a wall with a lot of confusion.

There has been a shift to break down this "Wall of Confusion." More teams are starting to work with a DevOps mindset places more emphasis on delivering quality in a service, with end-to-end responsibility.

To assure the quality of the service, testing was and will be an essential part of the process. For this reason we need to justify the difference between continuous testing and automation testing. Continuous testing is basically the outcome or the boundary of automation testing where automation testing is executed by some tools and continuous testing is performed as a dependency of that tools.

Many testers today feel that there is too much overhead to maintain the automated tests and that it does not work that well these days. The tests just can not be automated and run the same way anymore.[8]. For changing this thought maximum well known company switched their developing criteria to continuous testing. But they also chose automation testing for their small project and some api related projects.

The main aim of this essay is to provide a clear understanding about the differences between continuous testing and automation testing focused on DevOps. As we know that continuous testing and automation testing both are used in parallel in real environment. The objectives of the study are to identify the benefits of implementing continuous testing and automation testing in organizations where different tools is used for the challenges faced by organizations during DevOps adoption, to identify the solutions/ mitigation strategies, to overcome the challenges,the DevOps practices, and the problems faced by DevOps teams during continuous integration, deployment and testing.[8]

2 Testing in DevOps

“The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements to demonstrate that they are fit for purpose and to detect defects.” [3]

This definition of software testing is known throughout the world. In DevOps these activities still need to be undertaken to verify quality of a product. Before DevOps, testing tended to be a phase in the SDLC which ended before the product was delivered to Operations. Testing, however, should not be a phase. It should be present from the start and should be a responsibility shared by the entire (delivery)team [7]. Done right, testing helps to “Build quality in” as how W. Edwards Deming described this [5]. The shared responsibility together with embedding it in the entire process should lead to a good quality product.

2.1 Automation

DevOps will not work without automation. Error-prone manual tasks can and should be replaced by automation. DevOps teams require fast feedback and automation is the way to get this to the team. It can speed up the existing processes and make sure the team receives feedback about the process as soon as possible. When automation is working, team members can focus on tasks which do require human intervention. Automation can play a role in the breakdown of the "Wall of Confusion." It could be possible that Development and Operations used their own set of tools for deploying and other processes. Within DevOps it is best if teams start using the same tools for the entire SDLC. This can be a way of bringing team members together and make the SDLC clear and coherent. Different skills present in the team can shape the automation to where it fits the needs of the entire team.

2.1.1 Automation Testing

In testing, more and more tests are being automated. Test engineers work more with testing tools and automation supporting their tests. It creates fast feedback loops which drives development design and release [7]. In DevOps you want to automate as much as possible. The testing quadrant diagram, as created by Brian Marick, was adopted by Lisa Crispin and Janet Gregory to create the Agile Testing Quadrants[4]. The quadrants show different kinds of tests where automation can play a role. Technology facing tests supporting the team, like unit tests, are automated.Business facing tests supporting the team, like functional tests, can be automated or done manual. These functional tests in DevOps should be automated, based on the DevOps principle "Automate everything you can" [1]. These functional tests are the tests that should be part of customer-centric testing as mentioned before.Technology facing tests critique to the project, however, require mostly tools and are therefore already automated.

The automation of the first three Agile Testing Quadrants should leave time and space for the last quadrant with business facing tests that critique to the product. These tests should be done manual and cannot be automated. With multiple skill sets in a DevOps team, it would benefit the team to perform these tests with the team during the time they saved with implemented automation.

The test pyramid can help the implementation of test automation. It makes a distinction between tests that can be executed fast on low levels (unit tests) and tests that are slower to execute (UI tests) [6]. The lower-level tests are most suitable for automation, which is why they are usually fast to execute. The test pyramid combines tests originally performed by developers (unit tests) and those performedby test engineers (service, UI tests). This is a testing strategy that will work with DevOps because it is a cross-functional strategy. Engineers in a DevOps team should share their knowledge and expertise to fully implement this test strategy. This strategy also helps teams making testing a shared responsibility within the team.

2.2 Continuous Testing

Automated testing can be part of a deployment pipeline and can be part of Continuous Integration, Delivery, or even Deployment. Deployment pipelines are the "automated implementation of your application's build, deploy, test and release process" [7]. The deployment pipelines are a way to empower teams to take control over their deliverable. A pipeline can help a team deploy their service and verify the quality in an automated way. In DevOps teams, it should empower all team members to deploy any version on any environment with the right controls in place. The pipeline can limit the complexity of deploying and testing a service. The knowledge difference between team members can be smaller when every team member could deploy and test a service with a push of a button.

Continuous testing can act as continuous process where every time the pipeline is started tests are being executed. Tests can act as go or no-go points in the pipeline to go to the next step in the process. It also gives the team up-to-date feedback on the quality of their service in different stages of development. Testing can also be used to test the automation. It can help understand if the automation executes the correct steps in a correct way. This includes the automation used for deploying services on different environments. With deployment testing, a team can take control of the infrastructure and check whether it is in the state where it should be. Testing will give teams control on their automation when they are relying much more on it.

3 How Continuous Testing Different from Test Automation

In Automated Testing, the code is integrated with the mainline, then automation test scripts are developed, and the binaries are tested against these scripts automatically using automation test suites.

In continuous Testing, the test scripts are written before the coding begins. So, when the code is integrated, the automation tests are automatically run one after another - hence the term Continuous Testing. Development methodologies such as TDD and BDD gel well with the DevOps framework as they one of the enablers for Continuous Testing.

The difference between automation testing and continuous testing is can be best explained in Figure:

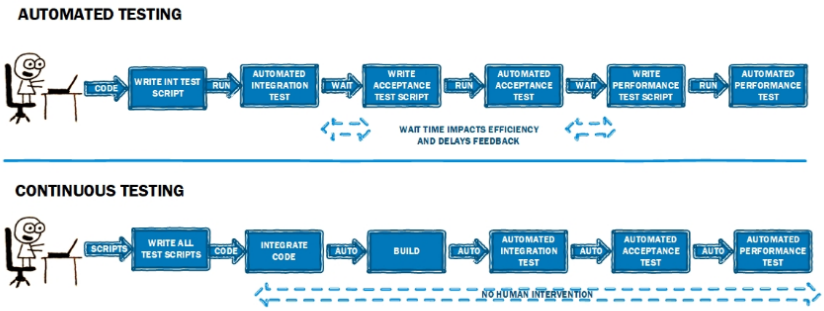

Figure 3: Automated Testing vs Continuous Testing

There is another differences between Continuous Testing and Automation testing which I have given below as a tabular form which make it clear view.

Parameter	Test Automation	Continuous Testing
Definition	It is a process where tool is used for automating tasks.	It is a testing methodology which focuses on achieving continuous quality and improvement.
Purpose	A set of similar or repetitive tasks, a machine can execute, faster, with a fewer mistake.	The continuous testing process helps to find the risk, address them and improve the quality of the product.
Prerequisite	Automation in testing possible without integrating continuous testing.	Continuous testing can not be implemented without test automation.
Time	Software release can take a month to years.	Software release may be released weekly to hourly.
Feedback	Regular feedback after testing each release.	Feedback at each stage needs to be instant.

From the figure we can say that the advantage of Continuous Testing over Automation Testing is that once code is checked into the source code repository, the process to build and validate begins, and the feedback is obtained rapidly. There is no gap in the process where the integration, testing and feedback mechanism await human intervention.

4 Conclusion

In software engineering, all developers complete the coding activity and hand it over to testing folks. While the testers are busy at what they do best, developers often have to await feedback. DevOps came in with a single objective of cutting down the software delivery timeline to deliver a better quality software. For

development time and quality assurance we can use continuous testing as we know the above description. It cuts down the feedback cycle significantly and this benefits in avoiding rewriting pieces of code.

As a result we can say that if someone uses continuous testing it means they basically uses automation testing as a part of it.

There are a number of organizations, such as Amazon and Netflix, who are able to deploy multiple times a day. They keep their deployments small, apply Continuous Testing, and deliver unparalleled results. In this age, we must move away from automation testing into continuous Testing. This is the future and survival depends on how quickly organizations are able to transform.

References

[1] 6 principles of devops, 2017.

[2] Anand Srivatsav Amaradri and Swetha Bindu Nutalapati. Continuous integration, deployment and testing in devops environment, 2016.

[3] International Software Testing Qualifications Board. Certified tester foundation level syl-labus version 2018 version, 2018.

[4] Gregory J Crispin, L. Agile testing, 2008.

[5] W.E Deming. Out of the crisis. mit press, 2000.

[6] M.: Bliki Fowler. Test pyramid, n.d.

[7] Farley D Humble, J. Continuous delivery. addison-wesley, upper saddle river, nj, 2011.

[8] A. Shafer. Agile infrastructure. speech presented at velocity conference, 2009.