



# The Role of Test Automation in DevOps to Accelerate Software Development

Jagadish Anandan



**Microsoft Partner**  
Gold Application Development  
Gold Collaboration and Content



# Introduction

---

DevOps helps IT organizations to deliver applications and services at high velocity. It is this speed, combined with quality that has made DevOps the preferred methodology. As a cultural shift, DevOps provides a cohesive environment where tools, practices, and philosophies work together to accelerate time to market. Many companies which use legacy software find DevOps challenging to embrace and some even believe that the old has to be abandoned for the new. But that is not the meaning or the purpose of DevOps. It is not a static, single time activity but an ongoing process. As a standard for practice, it emphasizes the ways in which development, testing, and operations can collaborate effectively. Gartner defines DevOps as “a change in IT culture, focusing on rapid IT service delivery through the adoption of agile, lean practices in the context of a system-oriented approach”.

As it seeks to improve collaboration, DevOps utilizes technology, especially automation tools.

# Evolution of DevOps

---

DevOps is an offshoot of Agile software development - an outcome born from the need for speed. Where Agile addressed the shortcomings of Waterfall, DevOps maximizes the advancements in Agile accumulated over the last decade and has exposed the method to overcome shortcomings in end-to-end software delivery lifecycles.

Agile Development which includes popular methodologies such as Scrum, Kanban, Lean Development and Extreme Programming incorporate iterations and continuous feedback to refine and deliver a software system. Agile is all about continuous planning, continuous testing, continuous integration with the word 'continuous' being the underlying theme. So where or when did DevOps emerge? To answer this question, we need to understand a typical Agile development environment where, in the absence of a clear definition, the prescribed universal title of 'Developer' was used to describe all team members, leaving "self-organizing" to the team.

In that era, the software teams would be in charge of requirements gathering and writing code and QA would be responsible for testing the code. This fragmented and siloed approach produced more bottlenecks than short-term project wins.

Working separately, both teams would not be aware of what the other is working on. Their overall involvement in the client's business vision would be limited and they would simply focus on deliverables. As a result, there were inefficiencies in delivery with ownership being divided.

Breaking the divide, DevOps combines the development team and QA to streamline the movement of software through its various development cycles. It empowers cross-functional teams from design through the whole cycle ending with production support. For DevOps to work, there is the requirement for teams to work together, i.e. communicate, collaborate, integrate and coming to the crux of the matter, automate. Since QA is involved right from the onset of a project, it is fairly easy to automate processes and this is one of the key advantages of DevOps. Thus, DevOps focusses on standardizing development environments and automating processes to improve predictability, efficiency, and maintainability. It provides software developers control over the production environment and empowers teams to build, validate and deliver together.

While it is definitely an offshoot of Agile, it is a cultural transformation where the entire teams' final goal is customer satisfaction. To achieve this, the teams look at shortening the process of development, reducing iterations and automating processes all of which when combined works hugely in favor of the customer.

DevOps through improved communication and collaboration, careful delivery and automation of processes, results in:

- Faster time to market
- Reduced chances of failure in new releases
- Improved time for fixes and better recovery

## Test Automation in DevOps

DevOps is classically divided into four processes:

- Continuous Integration
- Continuous Delivery
- Continuous Testing
- Continuous Monitoring

Since, testing in DevOps is not the last mile but is in fact present right from the beginning, the team works with the operations team to ensure that continuous integration and delivery are made possible because of continuous testing and monitoring. Testing teams, therefore, need to align their test design, test automation, and test case development with DevOps to ensure that changes are not affecting the end product. It is testing at its most mature level.

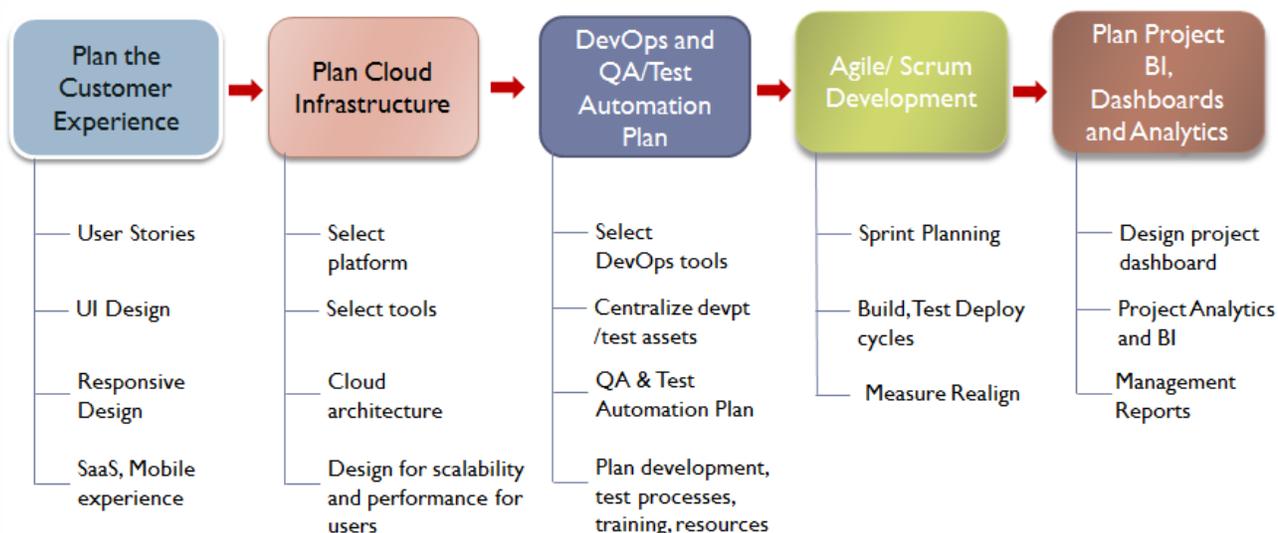
For many organizations, DevOps's secret lies in its test-driven development approach defined by continuous testing. However the fact remains that developers who are primarily responsible for the outcome of their product development efforts have already done what is required to ensure that their coding is relatively bug-free. Combine this with automation tools and one is left wondering what exactly the role of testing is and if there is actually a role for testing in its traditional sense. Herein lies the beauty of DevOps. It is not about doing away with traditional testing roles and passing them on to the operations teams along with testing automation tools.

In this scenario, testing which is often considered a 'last mile' function, has to move into the driver's seat from where the entire product's evolution is carefully monitored. The new QA man is an orchestrator who is not performing transactional services. He is helping people build a sturdy, bug-free product. Therefore, along with running tests and handling the hands-on work, QA engineers will also don the consultation garb and provide valuable assistance to the developers. They join the story at the beginning and remain long after the curtains go down. They learn along the way and create automation tools which ensure that their own work is simplified, resulting in faster turnaround time. As mentioned earlier, it is all about speed and demonstration of finesse when reaching the finishing

“ According to the 2015 State of DevOps Report, “High-performing IT organizations deploy 30x more frequently with 200x shorter lead times; they have 60x fewer failures and recover 168x faster.”

line.

In the traditional testing scenario, several test cases would run, resulting in longer test cycles with manual intervention, all of which would slow down the testing process continuously. In DevOps, automation ensures faster testing as a result of continuous feedback. It checks at very point and acts as an auto-trigger for the delivery chain to continue. Automation testing in a DevOps environment will require moving automation test scripts to an enterprise version control tool to ensure a single source of truth. This centralized enterprise-level testing leads to an integrated test suite to enable centralized execution and reporting. Through automated health checks, it is also possible to verify services are running fine and smoke tests ensure that key system features are operational with no blockages.



## Let us drill down further and review all the automations possible in DevOps.

In a fast phased agile, DevOps model, improvements to a typical technology platform entail new features being introduced, almost on a weekly basis. But an organization cannot afford to have these enhancements affect existing usability in any way. Therefore, to ensure seamless information flow and uninterrupted transactional activity for users, the applications have to be tested every night, issues identified and remediated before business hours, the next day. Most SaaS providers have multiple product lines and they are often delivered via the web, mobile applications, API interfaces to partners and customers via Web-service or RESTful services. Application access via the web, is again through a variety of browsers and devices,



**Automated tests transform fear into boredom.**

– Eran Messeri, Google

and testing should cover all these aspects.

## Test Data Setup & Management

Test data is information that can be used as input for a test. It can be static or transactional. Test data setup is the process of ensuring that the data is of the right quality, suitable quantity, correct format and proper environment, and at the right time. This whole process requires an effective strategy to be in place. If strategy and management of test data are in place it is possible to optimize testing effort and costs. Automation testing can benefit from TDM implementation. TDM by itself helps to save both time and cost by keeping data ready and when this data is automation tested the whole process becomes highly effective and successful.

## Test Harnesses

Test harnesses are 'enablers' that help to execute tests using test libraries and generating reports. To explain further, when the test harness is in place, and data is prepared, it should be possible to click a button or run a command to execute all the tests and generate reports. To ensure that this happens requires a collection of different things, like, for example, the unit tests prepared when the application was being developed. When test scenarios or test cases are substituted with scripts, these can be automated. A test harness provides stubs and drivers. Stubs are typically used to replace missing software and simulate the interface between the software components. For example, if there is a function (A) which is ready to be tested but it is dependent on function (B) which is still under development, a dummy function is created to act in the place of function (B). This dummy function is called a Stub.

When these test harnesses are automated, there is the possibility of increased productivity, better quality of software components and applications. It is also possible to ensure that subsequent test runs are the exact duplicate of previous ones. More importantly, testing can occur at night or at times when the office is not staffed.

## Automated Functional Tests

Functional testing in continuous integration ensures that every single, minute change is regression tested. However, functional tests end up creating a huge number of test cases, which are inconvenient, to say the least. It is, therefore, preferable to create a high-value test suite which focusses on business critical areas. In a particular case of web application testing with limited

functionality, it is possible to perform functionality testing with Selenium. Selenium tells a browser to click on an element, populate and submit a form, navigate or do any form of user interaction. With a few commands one can fetch web content using a URL and its content can be verified. If any of the commands fail the validation test, the web page is considered to not match the required specification and the test is considered to be a failure. All the interactions that the application has with the web application, as well as the validations it performs, are recorded in a log file. Thus, each time a test run is performed, a log is generated, which serves as documentation evidence that can be shared with the customer.

Using SoapUI which is an open-source web service testing application for service oriented architectures, test automation can be simplified. In a typical scenario, the entire application is divided into three parts: data, logic, and presentation. If this model is strictly adhered to, test automation becomes easy and any one of the tiers can be replaced with an automation tool.

## **Automated Performance Tests**

With continuous integration, comes performance testing where issues can be detected and mitigated before they derail the project in the real world. Automated performance testing takes this to the next level where this cutting edge practice ensures that companies across the spectrum are building applications using a continuous integration server. It is a process where automation tools such as Selenium, QTP are used to perform actions by executing automated tests to compare actual outcomes with predicted outcomes. Performance testing can be achieved with automation.

## **Manual Functional Tests**

Finally, there may be a necessity to do final manual verification as a check point or approval. It is preferable to keep this to a minimum so as not to impact the timelines.

To ensure comprehensive testing, it is necessary to include functional manual testing in the overall strategy. It is a black-box testing method that involves converting functional requirements of the product into manual functional test cases to provide a swift and objective way to assess behaviour. Manual functional testing should adhere to a proper plan and industry best practices to ensure good results.

# Incorporating Test Automation in DevOps

## I

Slicing through silos, in a DevOps environment, the entire group of stakeholders involved in a project meet to discuss the project. They plan and collaborate right from the starting point. Every day when new code is created, automated testing ensures that the code is ready to be deployed. Once the code passes through automated testing it is deployed to a small group of users and the new code is monitored for a short period of time to ensure that it is stable. Then the new code is proliferated to remaining users. The important point to be noted is, many of the steps are done with no human intervention.

Bringing together the two teams, i.e. developers and testers requires a paradigm shift in the approach to a project. This shift in thinking is what many mistake to be the meaning behind DevOps. However, just bringing the two together by itself, while it will resolve many challenges associated with typical development, may not optimize development cycles. What will help to optimize development cycles are the best practices in automation processes that are incorporated as a result of the knowledge earned from cohesive teamwork.

Thus, while it is easy to say that incorporating test automation in DevOps will result in increased velocity of software delivery, the picture behind it is large, starting from different teams working together to understand project requirements to delivering short term releases with lesser bugs, to translating learnings into automation to continue to inversely

### Automation Testing Tools:

- Selenium
- Cucumber
- Jasmine
- Jmeter
- Junit

reduce manual testing time.

DevOps focuses heavily on establishing a collaborative culture and improving efficiency through automation with DevOps tools. While some organizations and people tend to value one more than the other, the reality is it takes a combination of both culture and tools to be successful.

## II

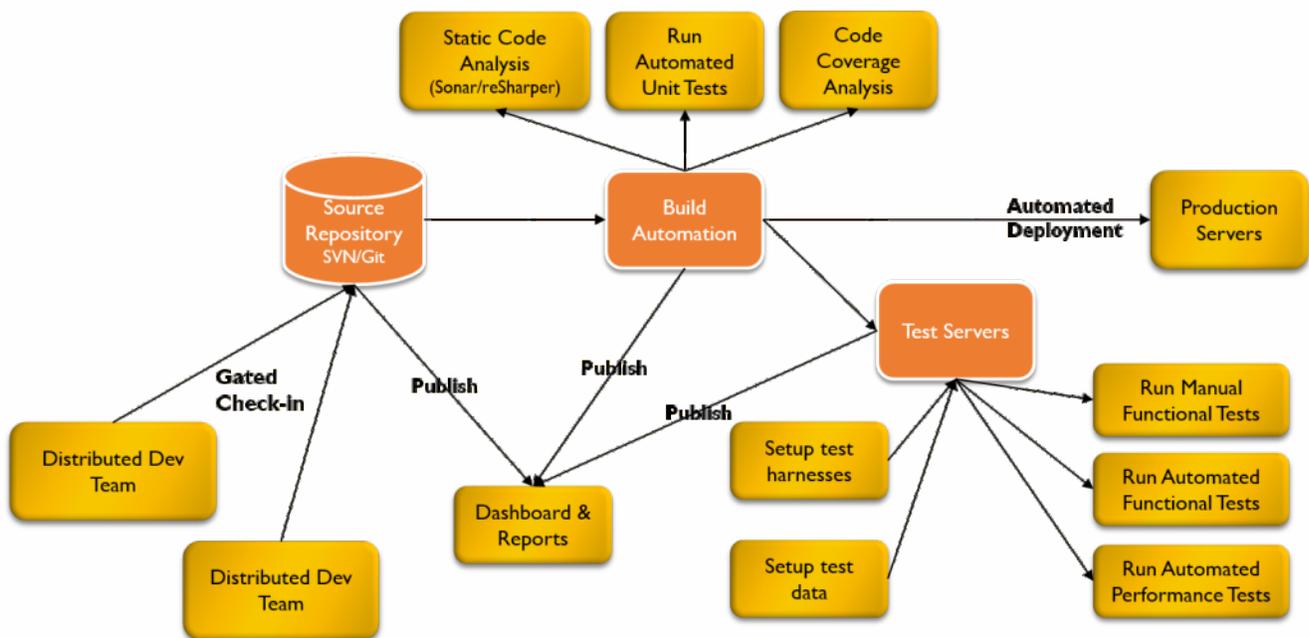
Test automation in DevOps is shifting left. It is about transferring the onus for the project into the hands of the developers to ensure quality. It is the process of designing, planning and implementing tests very early within a sprint. As author Stephen Vance highlights in his book, 'Quality Code: Software Testing Principles and Practices', "Of all the practices you can leverage to assist your craftsmanship you will benefit most from testing."

Test scripts work very well in situations where there is a clear understanding of the code and modules that need to be tested. Also, functional testers who understand the business needs may not have the know-how to construct effective tests and develop test scripts. This makes the process of developing scripts and then using them for testing long and arduous.



While developers have access to IDE's and object-oriented technology, test automation teams are largely dependent on monolithic scripts. Objects from each application screen become the building block for test automation. In a DevOps environment, the entire team reviews, plans the project based on business goals. The test plan is initiated at this stage of the user interface or the business goal related part of the project is first tested. QA ensures high-quality tests and configures tests into various workflows. Thus, when the product actually starts evolving, tests are scheduled to run automatically every day/night. These findings are included in the test suite. Operations can use these automated tests for regression testing too.

When code changes are made, intelligent test automation identifies these changes and these are automatically added to the test cases. QA enhances test data and tests are rerun to validate software changes. The ease with which reconfiguring can happen is a key advantage of test automation in a DevOps environment.



## Trigent's Commitment to DevOps and Test Automation

---

Trigent's DevOps solution accelerates the software delivery process by providing a comprehensive Continuous Delivery environment. It leverages your existing ALM tool investments, uses industry best practices and provides real-time dashboards. It offers a self-service mechanism for Dev - Test provisioning either in a cloud or hybrid environment.

With over two decades of experience, Trigent's Test Automation, reinforced by strong partners such as Borland (Microfocus), is aligned with today's business environment with the ability to provide cost benefits, performance, and agility.

As niche test automation experts we have significant experience in open source and commercial testing tools. Our large library of modular, reusable and resilient frameworks simplifies scenario-based automation. We provide on-demand testing and next-gen scheduling.

---

## About Author

### Jagadish Anandhan

Jagadish Anandhan is a Project Manager in Trigent Software Inc. He has over 10 years' experience in functional, automation and performance testing. When he is free he explores / evaluates new software /tools and contributes to the open source community.

---

## About Trigent

Trigent is a privately held, professional IT services company and a Microsoft Gold Partner with its U.S. headquarters in the greater Boston area and its Indian headquarters in Bangalore. We provide consulting services in various technologies including Microsoft Solutions. Our operating model is to conduct sales, customer relationships and front-end consulting (e.g., business case, requirements, architecture) onsite with our clients and perform the detail design, development, integration, testing and quality assurance offshore at our world class development and support center in Bangalore. We are a SEI CMM Level 4 company and is ISO 9001:2000 TickIT certified organization.

For sales contact [sales@trigent.com](mailto:sales@trigent.com) or call 508-490-6000.



**Microsoft Partner**  
Gold Application Development  
Gold Collaboration and Content

