

THE ROLE OF AUTOMATION IN DEVOPS



Internet, connectivity and software technologies have raised customer expectations for service performance, reliability and quality. And since every organization is a software organization, businesses are forced to respond faster than ever before when it comes to improving product features, fixing issues, delivering customer service or even to market competition and trends.

Organizations have responded by embracing modern [Software Development Lifecycle \(SDLC\)](#) strategies such as [DevOps](#) that streamline the organizational, process and tooling workflows to push quality software features to end-users faster and effectively. Fast response to customer requirements often comes with eliminating the process bottlenecks and replacing manual tasks with automation. The resulting agility and speed helps organizations align every step of the SDLC workflow with business goals in context of changing end-user and market requirements.

Enabling Standardization in DevOps

As progressive business organizations develop software products from the perspective of customer or end-user experience, developers need to work their way backward in identifying customer needs and delivering appropriate products. Varying customer needs, unpredictable market trends, stringent compliance requirements and internal business goals open too many variables for developers to follow throughout the SDLC pipeline. This means that every step of the SDLC presents several variables to consider, each with its own technical and business dependencies. This variety bottlenecks the software development performance as disparate SDLC teams have no consistent workflows to follow and end up in silos.

To address this challenge, DevOps teams must adopt standardized workflows, processes, technologies, protocols and metrics. This reduces the variety, and the associated risks, and enhances the potential for automating otherwise manual processes. In many ways, standardization can be seen as the necessary enabler to automation and therefore has a key role in the automation strategies of a DevOps organization.

At the same time, DevOps organizations must also maintain a delicate balance between standardization and adaptability, especially when it comes to the tooling. DevOps is continuously evolving, and every organization can have a variety of DevOps workflows, strategies and implementations. The tooling standardized today could be detrimental to innovation promised by new or alternate technologies available in the industry.

However, the concept of automation in DevOps, encompassing the approach of standardization, also applies to the governance models associated with the standardization of tooling, among other aspects. Certain mechanisms should be in place to facilitate the adoption of new technologies that prevent employees from adopting shadow IT practices, which in turn create silos between teams. For instance, creating a library of tools that can be requested, vetted and made available as needed can help organizations prepare for the changes associated with adopting new tools and yet maintain standardized DevOps workflows.

Considering the wider view of the DevOps landscape, organizations must consider the role of automation not just as describing the technology stack as executable, but also as the ability to eliminate performance bottlenecks, communication gaps and silos between Devs, Ops and QA, as well as mechanisms that facilitate agility through standardized SDLC process.

The role of automation extends to the following key tasks of the DevOps SDLC pipeline:

- **Code Development:** Automation in applications such as source control allow developers to keep on top of the DevOps SDLC pipeline. For instance, defining certain changes to the build and triggering appropriate changes or process activities save time and simplify the development of large, complex software projects.
- **Visibility:** Ops teams can keep on top of the code changes, existing issues and the resulting impact on project goals by automating traceability and the issue tracking processes. Devs and QA need to collaborate internally and across teams to ensure that the SDLC process runs smoothly. For instance, if a security team member identifies code issues weeks after it was submitted, Devs already have a challenge at their hand to fix the root cause without delaying or wasting the progress made since. A tight feedback loop between Devs, QA and Ops is required to eliminate silos and bottlenecks between the teams.
- **Continuous Testing:** Automation is critical to support, execute and manage continuous testing in DevOps environments. Since testing in DevOps frameworks is performed on a continuous basis during the SDLC pipeline, Devs and QA must also manage the requirements of a continuous testing environment. These range from triggering automated communications between systems and team members, integrating multiple tests, tracking and predicting issues, and facilitating continuous integration of approved automated builds.
- **Enabling CI/CD:** Automation in the Continuous Integration and Continuous Delivery pipeline ensures that appropriate software builds, data, tests and code changes are delivered to appropriate target environments. DevOps teams can therefore perform frequent code changes, stage the builds for testing and ultimately push frequent software changes to the market.

- **Monitoring and Incident Management:** The log and metrics big data cosmos can be overwhelming for organizations with infrastructure of all sizes and scale. Instead of swimming through the flood of alerts, a high-level incident reporting is required to make sense of infrastructure performance and potential issues. Automation therefore becomes necessary to intelligently prioritize events, identify root cause and deliver proactive actionable intelligence. To enable true DevOps, IT Ops might often need to replace antiquated IT service assurance strategies with automation capabilities and software-defined IT Ops, among other automation strategies.

Striking the Balance

It's important to understand that DevOps inherently is not all about the tooling. The concept of automation in DevOps is intended to facilitate agility and effective collaboration between all members of the DevOps organization. While it makes sense to automate the repeatable tasks that otherwise require manual efforts, it also comes with the consequence of repeating and expediting the results of those tasks, sometimes as automation waste. So if a process is flawed, the result through automation is consequently, flawed, and emerges faster than the manual approach. Secondly, automating processes in legacy environments may lead to additional challenges that tend to diminish the returns on automation investments and capabilities.

Automation is best employed only after eliminating the bottlenecks and bad designs that lead to automation waste. The core purpose of DevOps is to release software builds at better quality with higher frequency. Automation in DevOps should be embraced to realize this goal, without introducing a net negative effect into the quality or frequency of software released to end-users.