WHAT IS CONTINUOUS TESTING?

DevOps practices and Agile methodologies have taken the IT industry by storm. This surge in adoption of these newer practices comes as no surprise to those who have witnessed the impact they have on software enterprises. Today's customers have raised expectations thanks to the increased output of modern tech colossi such as Google and Amazon. The success these enterprises found came on the back of DevOps structuring and Agile methodologies. Most IT enterprises large and small now utilize some form of DevOps systems.

The key to DevOps is in part the tools, but mostly in the embrace of a culture of transparent collaboration. DevOps utilizes cross-discipline teams comprised of members from various IT backgrounds including development, operations, and QA. The goal of most DevOps teams is to achieve continuous integration and continuous delivery (CI/CD). To achieve this goal, DevOps enterprises leverage powerful tools that bolster the communication and throughput of their teams. Automation is a key aspect to a properly executed DevOps system.

The rate of deployments in a successful DevOps enterprise is incredibly rapid and automation is a key component of keeping pace. One of the issues that arise with shortening the software development life cycle (SDLC) is the immediate need to speed up the testing process to keep up with deployment schedules. Maintaining a testing pace with the increased rate of development can mean cutting corners in testing or intentionally slowing down the deployment process, giving testing time to catch up to speed.

Cutting corners isn't an option for enterprises that value the quality of their services and the customer's experience but slowing down isn't an option either. It defeats the purpose of DevOps. The only way to achieve CI/CD while putting out healthy and stable services is through the use of continuous testing.

Continuous Testing Explained

Continuous testing takes advantage of automated tests as a core piece of the software delivery pipeline to get feedback throughout the SDLC. Continuous testing is an automated end-to-end testing solution that integrates into your existing development processes. Modern application development and delivery pipelines have resulted in deployment cycles with less time between each delivery as well as more complex software releases. Continuous testing is implemented into the development process so that QA happens every step of the way to ensure risk is constantly measured and mitigated.

The primary goal of continuous testing is assessing business risk coverage by providing instant insight into the overall health of each release candidate. Embedding testing into the software development process from beginning to end ensures that issues are found sooner and are more readily manageable. Continuous testing is seamlessly integrated into the software delivery pipeline and DevOps toolchain. The pursuit of continuous testing is to eradicate bottlenecks completely by performing the rights tests at the right stages of each development cycle.

This is achieved by having continuous testing which delivers actionable feedback for each step of the development process. Realistically assessing the end user's experience provides invaluable feedback that helps DevOps teams ensure the user experience is protected without slowing down the SDLC. Achieving all of this requires the implementation of automated tests that work together with DevOps development tools to integrate directly into each stage of the process.

Continuous testing is a cultural shift from testing at the end to testing early, often, and at all stages of development with the utilization of automation wherever possible. The approach to testing should be systematic and, as with all processes in DevOps, should be in a constant state of improvement. The beauty of speeding up the SDLC is the fact that it gives DevOps teams more experience dealing with each stage of the process which enables them to have a deeper insight into ways they could be improved. The goal of DevOps teams and continuous testing is to improve constantly while looking for ways to optimize each step of the process.

Goals of Continuous Testing

We briefly mentioned the primary goal of continuous testing is assessing business risk coverage, but what does this mean in practice? The information you glean from your automated tests must be actionable data that also provides enough meaningful data to inform deployment decisions. Risk assessment requires low-level details as well as high-level information that can be used as data for supporting deployment decisions. If your tests aren't impacting the business decisions you make, then your tests aren't telling you enough.

Continuous testing should focus on the user's experience and whether or not changes have impacted not just performance but also functionality. Protecting the end user experience is paramount when it comes to DevOps deployment schedules because of how rapidly changes should be released. The rapid rate of deployments can result in quickly breaking your services if you don't implement continuous testing throughout the process from start to finish. Tests should be broad enough to detect the impact of changes made on the user's experience and the functionality of the application.

Risk assessment performed by continuous testing practices should cover risk mitigation tasks, technical debt, quality assessment, and test coverage optimization. This ensures builds are ready for the next step of the process before they move on. Continuous testing should also test for policy compliance. The information provided by continuous testing needs to be actionable and relevant to the current stage of the process the software is in the SDLC. This allows fixes to be performed right away without allowing issues to corrupt steps further down the process and require additional backtracking to address.

There are various tools available for DevOps practitioners that can be invaluable additions to your technology stack. This is true for continuous testing systems as well. Tools like GitHub and Selenium are open source options that help with testing the functionality of code every step of the way. DevOps teams can utilize the same automated tests regardless of where the changes are within the development life cycle. Continuous testing provides unparalleled access to actionable data that helps prevent the customer experience from degrading while also providing detailed information for business decisions.

DevOps: Solutions for You

If DevOps sounds like a good fit for your organization's needs but you want to make sure you get it right the first time, BMC is the IT solution partner you need. Read more about how automation and DevOps systems can help increase the rate at which you deploy products with BMC's free eBook: *Automate Cloud and DevOps Initiatives*.

BMC expert consultants are available to work with you to bring their knowledge and expertise to your organization. BMC provides custom-tailored [Deployment Services](#) for your organization to tackle the unique challenges you face. When partnering with BMC, you get:

- **Faster service delivery**: Agile releases that keep up with rapid demand
- **Visibility across data**: Ensure compliance and data accuracy
- **Cost-effective service**: Increased productivity and performance
- **Experienced DevOps professionals**: Equip you with the tools you need for success
- **Conversion or upgrade**: Seamless modernization or total replacement
- ***All tailored for the specific needs of your organization.***

Download or view the [Solution Implementation Overview](#) online to learn more about how [BMC Consulting Services](#) can help you. Check out how BMC's [Truesight](#) can help your enterprise maintain uptime and remediate vulnerabilities and then contact the experts at BMC to find out more about how to leverage DevOps practices for enhanced building, testing, and deployment success.