Understanding the Shift Left DevOps Approach

DevOps is one of the most preferred production methodologies. DevOps is preferred for its swift delivery process, drawing cues from the agile process, which doesn't hamper the quality. If anything, it fosters the quality and robustness of the deliverables by bringing together the development and operations teams' members under one umbrella.

While the developers slog and modify, the members of the operation test it and leave no stone unturned when it comes to deployment. This rapid development and deployment of deliverables have gone on to birth an offshoot termed **'Shift Left'**.

Shift Left DevOps is an approach that can border on **DevSecOps** as it emphasizes security and testing. Typically, in a DevOps approach, developers code first, and after the iterative update is passed on, the testers spot the bugs. In Shift Left, you initiate testing at a very early stage of the software development lifecycle (SDLC).

With testing done at a very early stage, the quality requirements along with resiliency are both met. In addition, the approach allows the DevOps members to ingrain security by working backward, and by doing so, the development speed is tremendously increased.

As the approach has gained traction, new tools have cropped up, **DevOps shift testing tools** have cropped up, and companies have taken cognizance of this new methodology. In this blog, we shall explore what exactly is DevOps Shift Left and how it fares against DevOps Shift Right.

Useful link: DevOps Tools for Your Business

DevOps Shift Left

DevOps is an infinite loop where the products are continuously integrated and developed. In that realm, Shift Left testing incorporates software testing procedures, such as security, as early in the software development lifecycle as feasible (SDLC).

In other words, methods and tooling enable development and operations teams to participate in the objective of providing safe, high-quality products. Organizations may release software more often by avoiding typical defects and security issue bottlenecks using shift-left testing and services.

In the DevOps cycle, testing is often the fourth phase in the continuous integration/continuous delivery (CI/CD) pipeline. On the other hand, **Shift left testing** physically shifts security and bug detection to the left by integrating many testing components into the Build and Code processes.

Understanding Shift Left Testing

Before the rise of agile development methodologies and cloud computing, developers would ask IT for infrastructure and wait weeks or months to get a server. As a result, IT has moved to the left during the previous two decades. Currently, the programming environment is automated and self-serve.

Developers don’t need to include operations or IT employees when adding resources to public clouds like AWS, GCP, or Azure. One of the integral elements of DevOps is Continuous Integration and Continuous Deployment (CI/CD).

These CI/CD procedures automatically create testing, staging, and production environments on-premises or in the cloud, then demolish them when they are no longer required.

Environments are frequently deployed declaratively using **Infrastructure-as-Code (IaC)**, which uses existing cloud technologies. In addition, organizations may dynamically provide containerized workloads utilizing automated, adaptive procedures thanks to Kubernetes, which is widely used.

Although this change has greatly increased the efficiency and speed of growth, it also poses severe security risks. There isn’t much time to examine cloud computing setups or post-development security checks of new software versions in this hectic atmosphere.

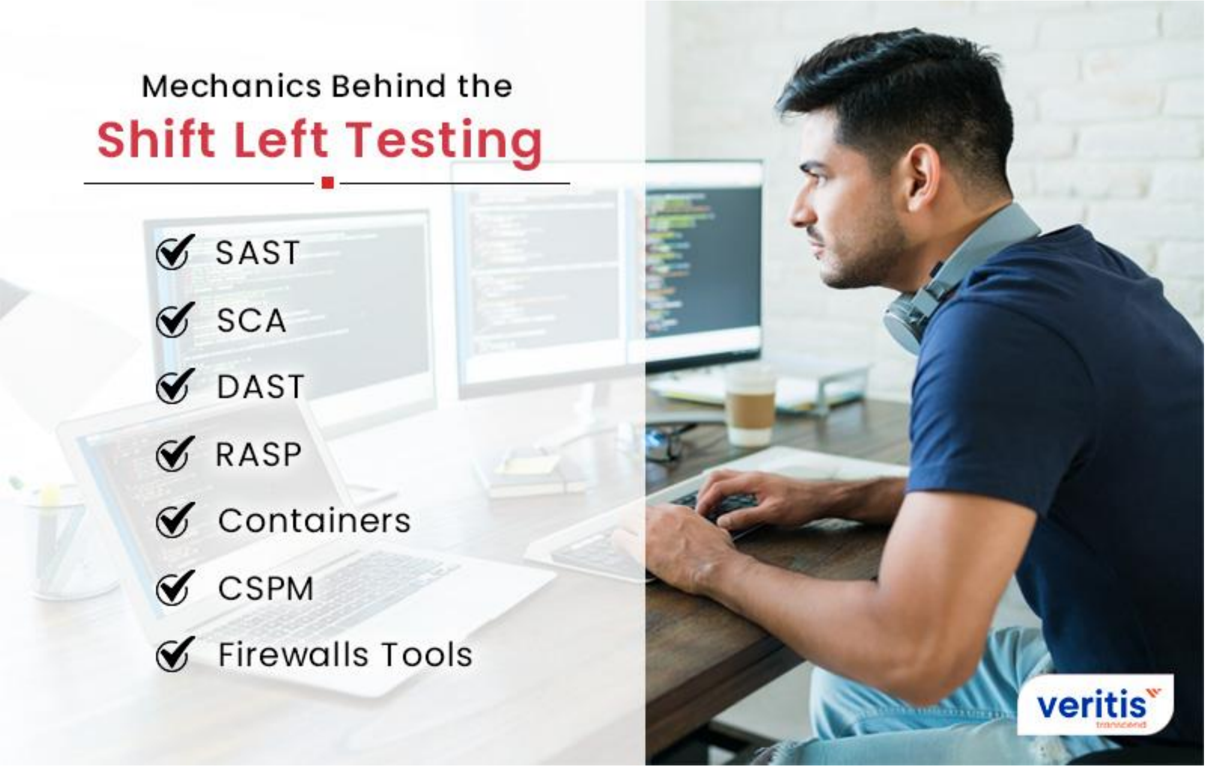
However, when issues are found, there is not much time for correction before the start of the following production sprint.

And every methodology in DevOps is actively driven by technology. So let’s see the technological aspects in the **Shift Left Testing**.

Useful link: Pros and Cons of DevOps Methodology and its Principles

---

Mechanics Behind the Shift Left Testing



DevOps companies understood they needed to move security to the left to prevent adding more security risks than their operations and security teams could handle.

The DevSecOps concept bridges the gap and enables quick, automated security evaluation as part of the CI/CD pipeline using a range of tools and technologies:

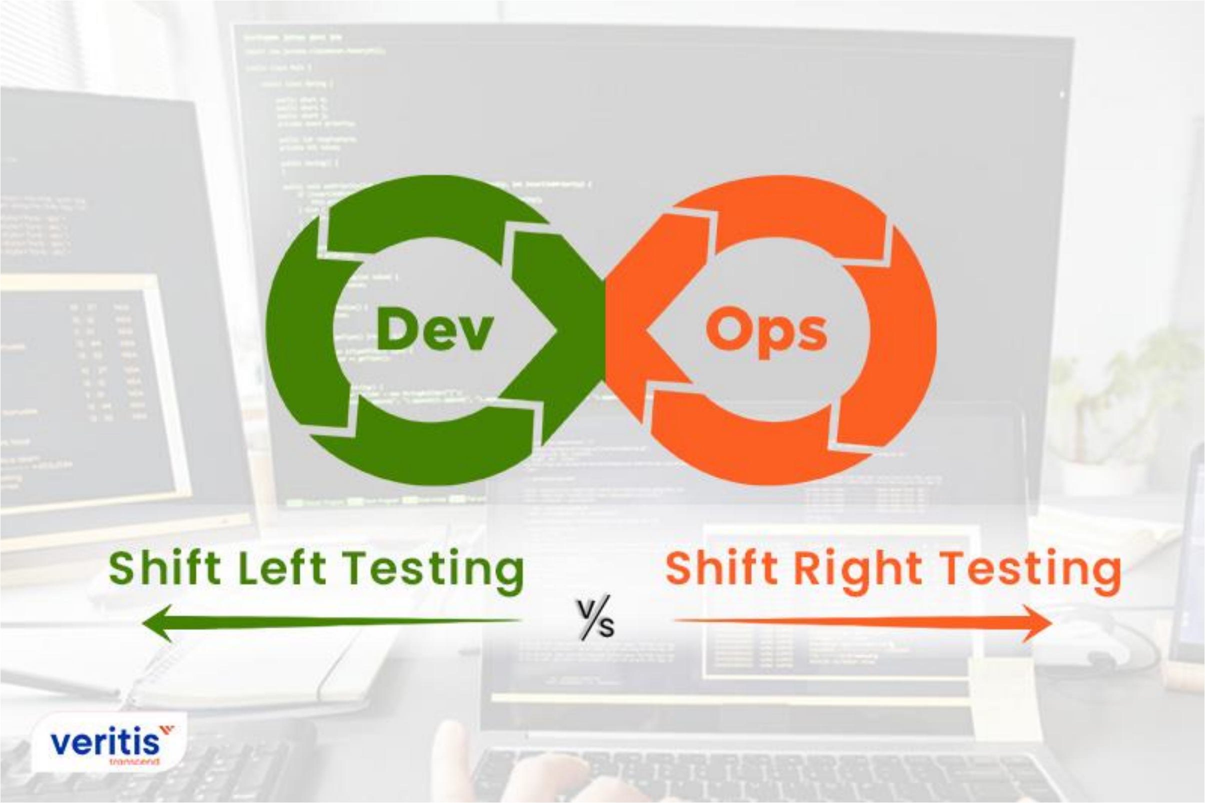
- **SAST:** Source code is examined using **Static Application Security Testing (SAST)** to look for obvious vulnerabilities and unsafe development techniques. This screening is frequently incorporated into programmers' development environments in DevSecOps to provide real-time feedback on security risks.
- **SCA:** Software Composition Analysis (SCA) examines programs to locate the open source and third-party libraries, other well-known software components, and any risks. By identifying potential risks that cannot be found by analyzing source code, SCA enhances SAST.

- **DAST:** Programs are scanned during **Dynamic Application Security Testing (DAST)** before deployment into production settings. This makes it possible to examine apps from the outside in for exploitable circumstances that could not be found in a static state.
- **RASP:** Runtime Application Self-Protection runs concurrently with software in the production environment to monitor and analyze behavior and alert or prevent illegal and abnormal activities. Although it could put more of a strain on production environments' infrastructure, this provides a real-time view of possible security issues.
- **Containers:** Before deployment into production settings, container image scanning technologies may continually and automatically scan container images throughout the **CI/CD pipeline** and in container registries. This makes it possible to find flaws or dangerous components and gives developers and DevOps teams clear advice on fixing or mitigating their effects.
- **CSPM:** Solutions for **Cloud Security Posture Management (CSPM)** find system failures in the cloud that perpetuate possible risks and attack vectors unchecked. Based on internal regulations or external security protocols of a business, CSPM systems can suggest or autonomously implement safeguards.
- **Firewalls Tools:** WAF or Web Application Firewalls allow for the monitoring of application servers to identify possible attacks and efforts to exploit security holes. Even without fixing the underlying software flaws, WAFs may be set up to prevent certain possible attack paths.

While the approach has its pros, there is always a variety to choose from, and for Shift Left testing, the main contender is Shift Right testing.

Useful link: Why Should You Adopt DevOps and What are the Benefits it Offers?

Shift Left Testing Vs. Shift Right Testing



Until now, we delved into what is Shift Left. Let's dig into what Shift Left's contender is about. Shift-right is the process of doing testing, quality control, and analysis during actual production. Applications running in production can handle real user demand while maintaining the same high standards thanks to shift-right techniques.

DevOps teams evaluate a developed application using shift right to guarantee performance, resilience, and software dependability. The objective is to identify and address problems it would be hard to foresee in software platforms.

Teams can test code using shift-right in a setting that resembles actual production settings that are not mimicked during development. As a result, teams can identify real-time issues with this technique before consumers do.

In addition, teams can utilize application programming interface calls to automate a portion of the procedure. Organizations may also use shift-right testing to check the code configured or monitored in the field.

Shift-left testing can reduce software bugs and shorten its launch time. Teams that practice shift-left frequently test before any code is developed and through production. Shift-left testing verifies that technology complies with the requirements established by the company rather than testing for usability.

Shift-right methodologies, on the other hand, may more effectively guarantee operational dependability by testing software in real-world settings and during operation. Consequently, teams benefit from broader testing coverage, which solves customer experience challenges better.

Useful link: DevOps vs DevSecOps: Approaches Which Amplify Automation and Security

---

Capping it off

While it is a discussion for another time as to the different types of Shift Left tests and Shift Right tests, each testing method has its advantages and challenges. **Shift Left and Shift Right** are methods that companies embrace, but like **DevOps**, both approaches are more than testing methodologies as they bring in a cultural change, which is the aspect that companies should brace for.

Deciding to adopt an approach is not a big deal, but making it succeed is what counts, and this is where Stevie Award winner Veritis comes in. Recognized for its DevOps excellence, **Veritis** is the preferred choice of Fortune 500 companies and emerging organizations. We have developed cost-effective solutions which don’t hamper quality or reliability. So reach out to us and embrace the best.

Services