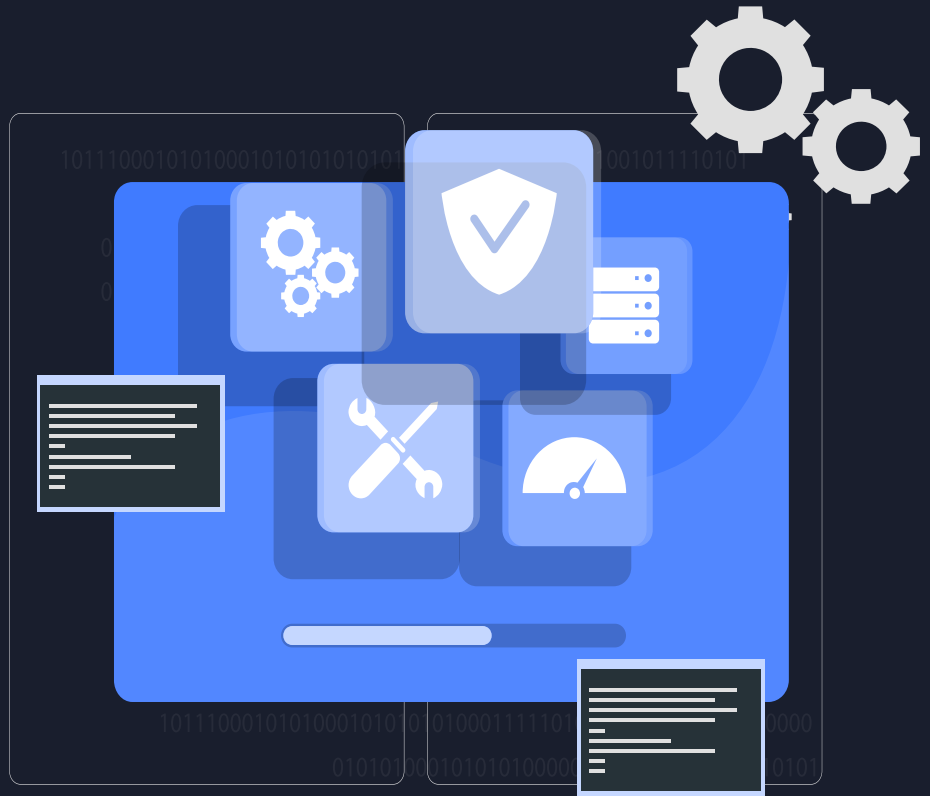


Combining Shift Left and continuous testing for superior results



Preface



Automated shift-left continuous testing in the autonomous digital enterprise helps IT teams ensure consistency, accuracy, and security

Shift-left testing means early testing. It brings a different mindset to the software delivery lifecycle (SDLC), where QE is involved much earlier. When combined with continuous testing, shift left testing enables project teams to quickly identify and prevent defects right from the analysis and design phase.

“29% of users will immediately abandon an app if they don’t find it valuable.” ~ Google APIs

A digital product’s success depends on how effectively you incorporate user expectations while keeping the application free from defects. Increasing competition and higher application development and testing expenses leave very few opportunities to achieve these goals simultaneously, primarily if you still rely on the traditional methodologies. Adding to these concerns, as the demand to release quality software products in a shorter time continues to increase, Agile and DevOps teams must adopt new approaches to avoid falling behind. Continuous and shift-left testing are two must have solutions in this modern era of software delivery that will effectively contribute to resolve these issues and improve software delivery results.

Before we continue the benefits of shift left and continuous testing, let’s understand what these methodologies add to the software development process.

Shift Left Testing

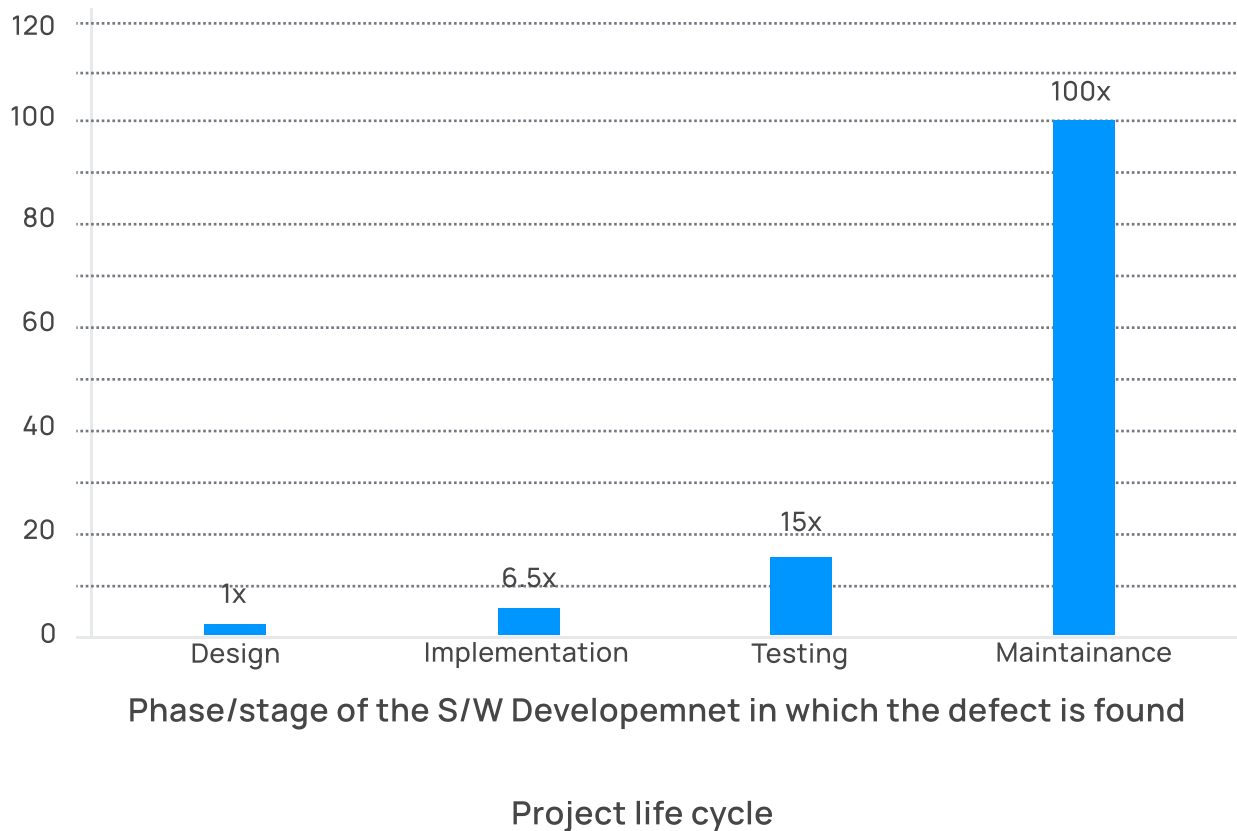
Shift-left testing is the approach where the testers start testing earlier in the development process. It makes testing a priority. Testing occurs at every stage, reducing the amount of time spent resolving bugs. This process is cost- and time-effective, increasing the application quality.

Continuous Testing

Continuous testing is the process of testing throughout the development lifecycle. It enables constant feedback for developers making it easier to fix bugs early in the development stage.

As you can guess, shift left testing and continuous testing, when combined, improve the testing results, and automate many redundant processes without increasing the budget of the development project.

The hidden costs of software testing



The overall software testing costs consume approximately 15 to 25% of the total project cost. Also, in 2019, a panel of technology professionals reported that around 23% of their organization's annual IT budget was allocated towards quality assurance and testing on average.

A downside of conventional testing is that it becomes expensive the more we delay it. IBM's Systems Sciences Institute confirms that the cost of fixing bugs or issues in the production or maintenance phase of the development life cycle is 100x more than the design stage. For an organization, it is expensive and compromises the work-life balance and work culture in the long run.

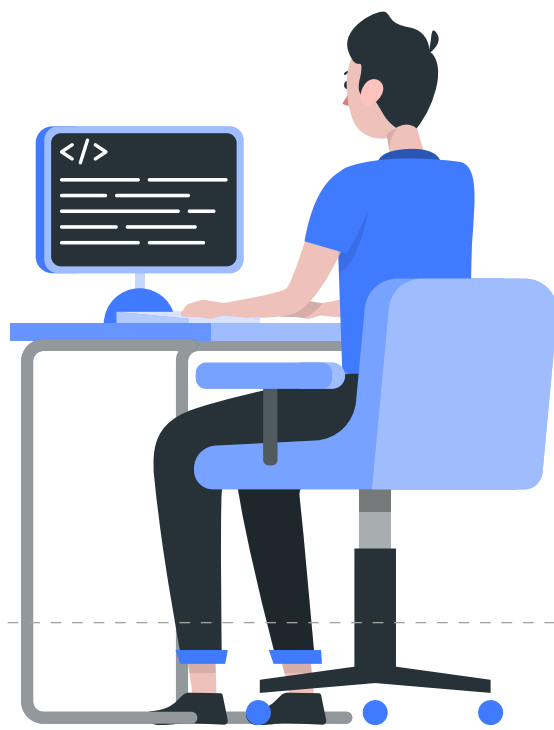
Shift left testing saves these additional expenses by starting the process of testing early in the SDLC. Combined with continuous testing, the process doesn't only begin saving costs but also saves development and testing teams from spending additional hours on unnecessary, redundant tasks.

Why Shift Left Testing Matters in Continuous Testing

The ultimate objective of shift left testing is to improve developer productivity by enabling them to prevent defects and reducing rework efforts. It matters because it emphasizes the test early principle and improves the results while also making the most part of testing automated.

Most often automation is misunderstood as automating testing of features or feature validation through heavy focus on UI based automation. However, UI automation is at the top of the "Automation Pyramid" and automating this layer provides limited value. Real benefits accrue once the focus shifts to the bottom layer of the pyramid. Refocusing on significantly automating the lowest layer of the pyramid that consists of Unit testing followed by automating middle layer consisting of Services / API layer is the key to a successfully implemented Shift Left in a continuous testing approach. Automation in this context includes automated test that validate performance of different components, validate security vulnerabilities and interface testing of those components by integrating all these tests into the CI-CD pipeline.

Shift-left testing should be incorporated into continuous testing to generate frequent, holistic, and practical tests on a larger scale. "Shift-left continuous testing" gives testers the power of automation while testing an application. It ensures that the testing capabilities are utilized to complete potential starting in the early stage of testing. The team involved in the software development and testing process gets real-time reports on the test results.

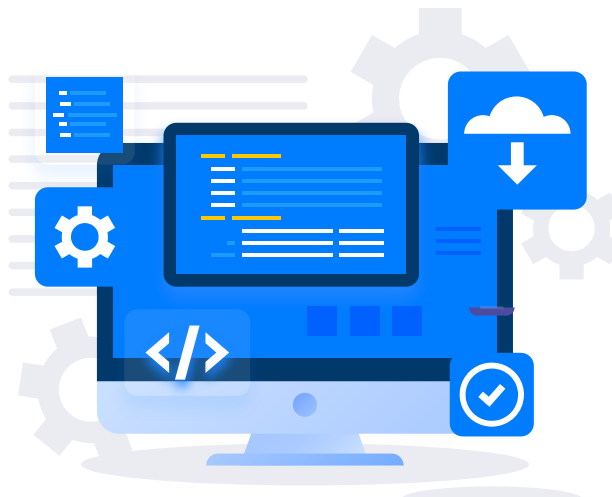


Why Traditional Testing Doesn't Work Anymore

Conventional testing approaches were relevant when applications were small, and their reach was limited. In the early days of the internet, software users were also fewer compared to the number today.

In today's world, traditional testing approach doesn't work due to three prime reasons:

- Application architectures are more distributed and highly complex. Design and development readily embrace the cloud, APIs, microservices, etc. Most development activities take place by combining different protocols and technologies within a single framework.
- Methodologies like Agile, DevOps, and Continuous Delivery have made it possible to release applications at a faster pace and update it multiple times in a short timeframe. It has decreased the timelines for test design, maintenance, and execution.
- Application is the primary identifier of the business. Organizations cannot take risks with their applications as their failure would ruin their reputation and market appeal.



The change in the status quo of applications has also changed the challenges for software testers. Testers are working to cater to the increasing demands of complex applications. Due to this, the old traditional testing approaches are not reliable anymore. Better effective testing is only possible if the standards and practices are shifted to the left.

Improving The Results of Shift Left Continuous Testing

In fact, a good project may well have more test code than production code. The time it takes to produce this test code is worth the effort. It ends up being much cheaper in the long run ~ David Thomas and Andrew Hunt in The Pragmatic Programmer. To understand the benefits of shift left continuous testing, learning why both work better together is essential.

The objective of this activity is primarily to automate application testing, maximize our test coverage, and perform testing as early in the development pipeline as possible.

Product development teams can make the most out of shift left continuous testing through the following practices:

Incorporate and utilize automated testing

The best way to ensure that testing teams gain the maximum benefits from shift left continuous testing is to ensure that testing includes automated testing. It keeps testing and notifying the testing teams of errors until automated test scripts can test the work done by the developer. This process speeds up the scope of testing.

Integrate automated test scripts with CI builds

Developers should ensure that all the new features and automated tools are integrated with the CI tool. This way, the test script runs continuously during each build. If the script fails, the build also fails. By integrating automated test scripts with CI builds, application development teams can ensure that no existing features are broken while developing new codes.

Standardize Coding

Development teams should also have a common consensus on coding standards. Staying on the same page speeds up the code review process while generating a higher quality of code. Coding standards eventually reduce the number of bugs while avoiding nasty or insecure bugs.

Improve the relationship between tester and developer individually

Teams should be compact and more interconnected. Management should encourage collaboration in every stage of development. Reports should be simple and accessible.

Organizational impact of shift left continuous testing

More focus on early continuous testing implies a shift from manual testing to test automation. Shift left continuous testing helps testing teams in multiple ways, some of them being:

Role	Key Change	What organizations should do
Development Team	<ul style="list-style-type: none">• Embrace quality engineering in testing and development practices.• Improve the coding process with automation.	<ul style="list-style-type: none">• Embrace quality engineering in testing and development practices.• Improve the coding process with automation.
Testing Team	<ul style="list-style-type: none">• Keep an open channel of communication between testers and developers.• Encourage testers to improve their soft skills to make communication more effective.	<ul style="list-style-type: none">• Keep an open channel of communication between testers and developers.• Encourage testers to improve their soft skills to make communication more effective.
Marketing Team	<ul style="list-style-type: none">• Keep the marketing team aware of the testing and development practices through open communication channels.	<ul style="list-style-type: none">• Keep the marketing team aware of the testing and development practices through open communication channels.

Transform Your Business by Harnessing the Power of Shift Left Continuous Testing

As digital applications gain prominence for competitive advantage, more and more enterprises focus on developing better products that identify with their brand and connect with their customers. The modern development process considers both time and quality as critical factors. Now that shift left testing has gained relevance and organizations readily adopt it, continuous testing has emerged as a critical catalyst for enabling quality at speed.

QualiZeal combines shift left testing and continuous testing and helps your organization adopt the philosophies of quality engineering. With QualiZeal's expertise, dev teams would find an advantage to catch bugs early and often validate the code.

