



DATASCAN

TECHDRIVEN

STEERING THE FUTURE

V6 - JANUARY 2022

DATASCAN QA STRATEGY - SHIFT LEFT & CONTINUOUS TESTING FOR WI APPLICATION

Problem Statement: "There is too much to test!"

The pressure to deliver more despite ever-growing features and bug fixes, all within shorter and shorter timelines, leaves most organizations struggling to balance the cost of testing with business demands. Testing is a critical area where speed and output quality can't be compromised. At DataScan, our goal is to operate QA at the highest caliber resulting in better testing strategies for our Clients. This paper illustrates how we accomplish this endeavor.

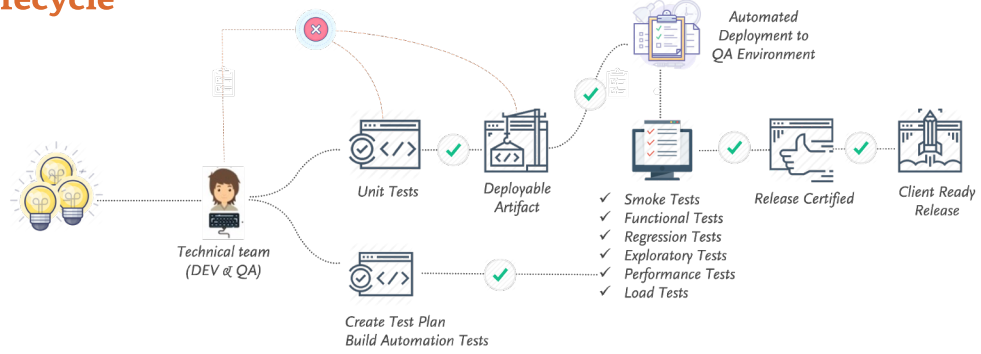
What Is Shift-Left Testing?

Shifting-left simply refers to the idea of performing an action earlier within a process. So, as it relates to software testing, shift-left testing is the approach of testing software and moving it to the left in the delivery pipeline - or testing software earlier in the development lifecycle than is historically typical. The benefits of testing alongside the development lifecycle significantly reduce the chances of critical bugs surviving late into the lifecycle and ultimately finding their way into production. By addressing critical issues during the development phase versus later in the lifecycle where the fix could require major refactoring for resolution, we address the fix immediately, and errors do not hinder the speed of software delivery.

What is Continuous Testing?

Continuous testing can be defined as automated application testing at every stage of the development lifecycle to immediately determine the level of business risk associated with a specific release candidate (RC), i.e., test early and test often. After each RC goes through the initial round of testing, the testing process continues by adding new test scripts and/or modifications to current automated testing and then rerunning the tests. Continuous testing requires as much test automation as possible at each step of the delivery pipeline, thus enabling us to scale at every release leading to a predictable and repeatable process. Our goal of continuous testing is to achieve end-to-end automated testing for the application in development. With this, the level of business risk associated with a potential release candidate can be lowered, raising the confidence in the quality of the release.

Testing Lifecycle



Our Software Testing Life Cycle refers to a testing process with specific steps to be executed in a definite sequence to ensure that the quality goals are met. Each activity is carried out in a planned and systematic way, with clearly defined goals and deliverables.

A big part of continuous testing is to get QA engineers to be consistent in their efforts to automate testing. Test scripts are written to test individual features as they are in the development phase. The requirement for automated testing is part of the acceptance criteria for a feature. It will not be considered complete until automated test scripts have been developed to test the work done by the QA engineer.

As features for the application are developed and automated test scripts are written, QA also ensures that the application can be deployed successfully to our QA test lab environment. We start with a continuous integration environment where we automatically deploy our application several times a day. The test scripts run during each build for each feature, and if they fail, the entire build should fail. This continuous integration between our development build process and our QA environments allows us to shift-left to test the latest code changes in smaller bite-size pieces as we go. A comprehensive suite of automated regression tests provides a safety net, ensuring that code changes do not break existing functionality.

The automated test can be run the same way or test different software features in each iteration, isolating defects to recent changes and providing quick diagnostics and resolution. As the application matures and approaches a potential production release, you will find that much of the testing associated with the application has been automated, thus increasing overall confidence in the release quality and readiness.

Monthly maintenance patches undergo the same rigorous automated testing as a major release through our continuous testing process. This process ensures the quality of the maintenance patch is operating at the same level as our major releases.

DataScan QA Testing Coverage

- **Core batch processing** – Everything that happens behind the scenes. Run daily processing to ensure the processes are complete in their entirety and are not resulting in failures or hang-ups.
- **Core User Interface (UI) functionalities** – Execute automated tests to ensure functionalities are working correctly for the new and legacy UI.
- **UI Layer** – UI testing focuses more on the interface which ties into the API. Ultimately verifying the parts that communicate between your browser and our services successfully connect.
- **All API Layer** – Test the APIs integrations between services that layer the UI.
- **Performance Testing** – Evaluate the overall system's performance and collect metrics such as availability, response time, and stability. Load testing monitors the application's performance at both normal and peak conditions.
- **Security** – We use a third party to conduct a dynamic penetration test for each major release.
- **Deployment Code** – Testing the actual code that deploys the application.

Client's Testing Coverage

DataScan encourages Clients to test the application as it is used in your day-to-day operations. For Client specifics, including downstream usage of any core or Client-specific outputs, you will need to test these processes as we cannot test consumption of files by a Client's organization. However, we can validate the functional requirements provided, including Exports, DDE Extensions, Webi, Imports/Exports, Billing Statements, or any Client customizations. Clients will also want to ensure integration points, especially third-party applications accept the changes.

Maintenance patches include critical bug fixes and security enhancements. We highly encourage you to accept the monthly maintenance patches to ensure the system operates at the highest quality and apply regular security updates to your environment. Clients are recommended to review the release notes for information and focus specifically on the bug fixes, if they apply to you, without the need to do full regression tests. DataScan also releases maintenance patches that have no release notes, as no bugs were addressed, but those patches do include important security updates.

Conclusion

The reality is that whether manual or automated testing, there is not enough time to retest every part of the system with every single use case. If we use a combination of Shift-Left and Continuous testing techniques to know where to focus with deeper testing in the development cycle and shift-right to identify and recover quickly from a problem, our responsiveness and execution towards resolution are more agile. This process does not mean there will be zero risk from a bug. It means that the risk of the bug occurring is less critical than the risk that existing bugs remain unfixed. At that point, the key is to understand the unfixed bug's impact and have the data to determine if the risk is low enough to certify the release versus not releasing at all.

In conclusion, while we exercise Shift-Left Continuous Testing practices, we would like to unify our testing efforts. To understand our Clients better, these are a few areas that would lead DataScan to more efficient testing and quality product delivery:

- Establish a Client-QA relationship to communicate testing efforts, challenges, solutions proactively.
- Clients provided DataScan with testing plans and requirements.
- Identify component of testing efforts that is the most time-consuming or challenging for Clients and what solution we can provide to reduce the effort.