

Embracing Quality Engineering

Point of View

By: Sriram Iyengar



Overview

In today's rapidly evolving technology and cloud era, the success of a company is largely determined by



Delivery Speed



Quality

Every customer-facing application or product must be evaluated for quality from multiple perspectives, including **business utility** and **return on investment**.

This document aims to provide a variety of viewpoints on how quality engineering can be used to rapidly develop high-quality software applications. Consequently, this would enhance the quality of life for consumers and increase customer satisfaction.

Why quality engineering?

Customers are the key to an organization's success and reputation. The current competitive world is tending towards being more **customer centric**. Yet, becoming customer-first is never easy.

It also means having the foresight to develop, say, a procedure to guarantee the application is constructed in accordance with accessibility criteria, which is necessary for expanding the client base and catering to customers with varying degrees of physical ability.

For example, the challenge could lie in ensuring the application's performance in contexts that are always changing, and so the organizations must provide high availability and scalability of applications supporting end users.

Every client expects **less downtime and outages**, and every organization works to achieve it. However, simulating such an environment and fine-tuning the application's robustness presents a significant problem. Businesses need to be able to give their customers a better experience, but it can be hard to spot performance problems and fix the application.



Because security is the most critical feature of any application, the difficulty is to incorporate security into the regular development life cycle.

Improving customer collaboration through the SDLC cycle proves to be a boon for delivering quality applications, but the challenge the organization sees is finding the *right tooling for accomplishing this task with the wide arsenal of tools available*.

Faster time to market with utmost quality also demands the customers to run checks quickly and efficiently and, in any infrastructure, (on-premises, public/private cloud, multi cloud etc.)

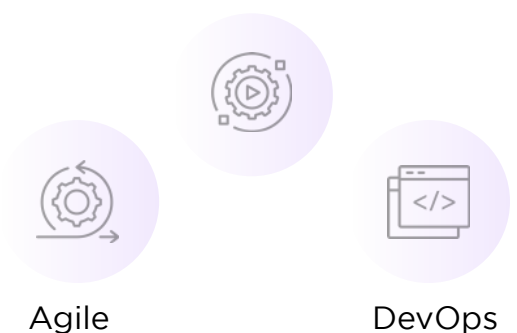
Organizations can tend toward quality engineering practices and address the above challenges to improve their reputation, customer base, and customer satisfaction.

Quality engineering adoption

Traditional testing processes have seen a sharp decline, especially after the development of more:

Automation has numerous advantages and helps in the production of a reliable system, *the improvement of the quality of test efforts, reducing test efforts, and minimizing the schedule*. However, this does not address all business needs in which businesses require extreme flexibility to scale up and down infrastructure to meet business needs while also lowering execution costs.

Test Automation Techniques



This has paved the way for testing types that were not explored earlier, some of which are described below. While this write-up provides some recommended tools that can be leveraged, there can be more. The goal of this piece is to look at places where we should also pay attention to improve delivery quality.

Automation would go hand-in-hand with addressing the challenges that were discussed earlier.

Here are some suggestions for adopting quality engineering practices, which can be done with a wide range of tools. Addressing these would help in improving the organization's quality goals in the current competitive environment and make them deliver quality applications consistently.

Quality engineering adoption



Customer collaboration is a key to the success of any organization that tends to be more customer centric. Behavior-driven tests can be written that follow plain english text. This would improve the collaboration with the customer, and they would willingly contribute to improving the overall quality of the application.

Frameworks like **Serenity BDD + Cucumber** can serve as an aid in accomplishing this.

Fine-tuning of performance glitches



Many organizations do performance testing as a part of feature or product releases; however, even after executing performance testing, there have been instances where performance-related issues have been raised by customers. One of the key reasons for this is that, generally, organizations test the initial page rendering but do not capture the complete journey of the end user. For example, it doesn't capture the time taken to load an image in a huge application or the number of javascript files loaded before an application initializes. Such glitches could be identified early, before an application is deployed to production and starts serving live users.

Tools like **sitespeed.io** in combination with various monitoring tools available help to test various aspects of an application.

Widening the customer base



While most of the organizations promote a culture of inclusion post-COVID, most of the front end is non-compliant with guidelines specified for people with disabilities, such as vision impairment, hearing disabilities, and other physical or cognitive conditions. This also limits products' wider reach, which leads to potential revenue loss.

It will be a good idea to make applications content accessibility compliant (**Web Content Accessibility Guidelines**) and it can be validated as a part of a continuous delivery channel.

This is still unexplored, but will gain importance as technology evolves. It is being tested as part of the CI/CD pipeline so that every time an application is built, this part of it can give early and automated feedback regarding accessibility.

The tool that is recommended for accomplishing this would be **Axe-WebDriver**. This will make this type of test an integral part of our build and test pipeline.

Reduce frequent outage



In the current scenario, every customer wants a premier customer experience, even when some unforeseen events strike. This needs a check to ensure how the application performs in such environments. To certify this, we are in need of a simulator that can configure such environments on demand and allow tests to be performed.

Testing this would involve simulating such an environment and running tests to fine-tune it.

Fine tuning these aspects will ensure better customer experience. The tool that is recommended is **Chaos Monkey/Kube Monkey**. This tool would serve as an aid in setting up chaos environments thereby helping in analysis of the application behavior.

Premier security



Scan for vulnerabilities and penetrate the application to validate the security measures put in place. This has become a prime requirement for any application that is being built. For a quicker time to market, *security scanning should be embedded during application development.*

There is a huge arsenal of tools available for accomplishing this, as this type of test has been around for a while. The challenge would be to run these tests in a CI/CD pipeline in an automated way. This will ensure that security tests are blended into the everyday application lifecycle, leading to applications of premier quality without compromising security.

This also helps in identifying the vulnerabilities that exist within the application early in the application development lifecycle.

The tool recommended is **OWASP ZAP** that can help to run vulnerability tests in an automated way for UI and API applications.

Validating infrastructure



This is one type of testing that has become more important as the cloud has changed and configuration management tools have been added. In the current scenario, every customer prefers managing their infrastructure using code.

There also exists a need to validate the infrastructure being created against the customer application requirements.

Tests are written to ensure that the cloud infrastructure being created conforms to the requirements of the application.

The tool recommended is **TerraTest**.

Live application performance monitoring under different loads



Scalability is a factor that helps in giving an edge to the customer over its competition, as applications that are scalable give a better customer experience.

Below, considerations for scalability testing are given.



Throttle the number of users randomly



Throttle the number of transactions randomly and continue this for an hour.



Following a closed workload model by specifying the number of concurrent users expected over a period of time.

Performance metrics can be captured by connecting the tests to time series databases for the purpose of gathering all performance parameters, which are then projected on a graphical application. Alerts are set in the Graphical Application for quicker feedback.

Custom metrics are also configured and tested to capture the scaling behavior of an application based on various loads.

Tools that would be helpful here would be **InfluxDB (Time Series Database), Grafana (Graphical Representation), Performance and Scalability Tests can be carried out using (Jmeter).**

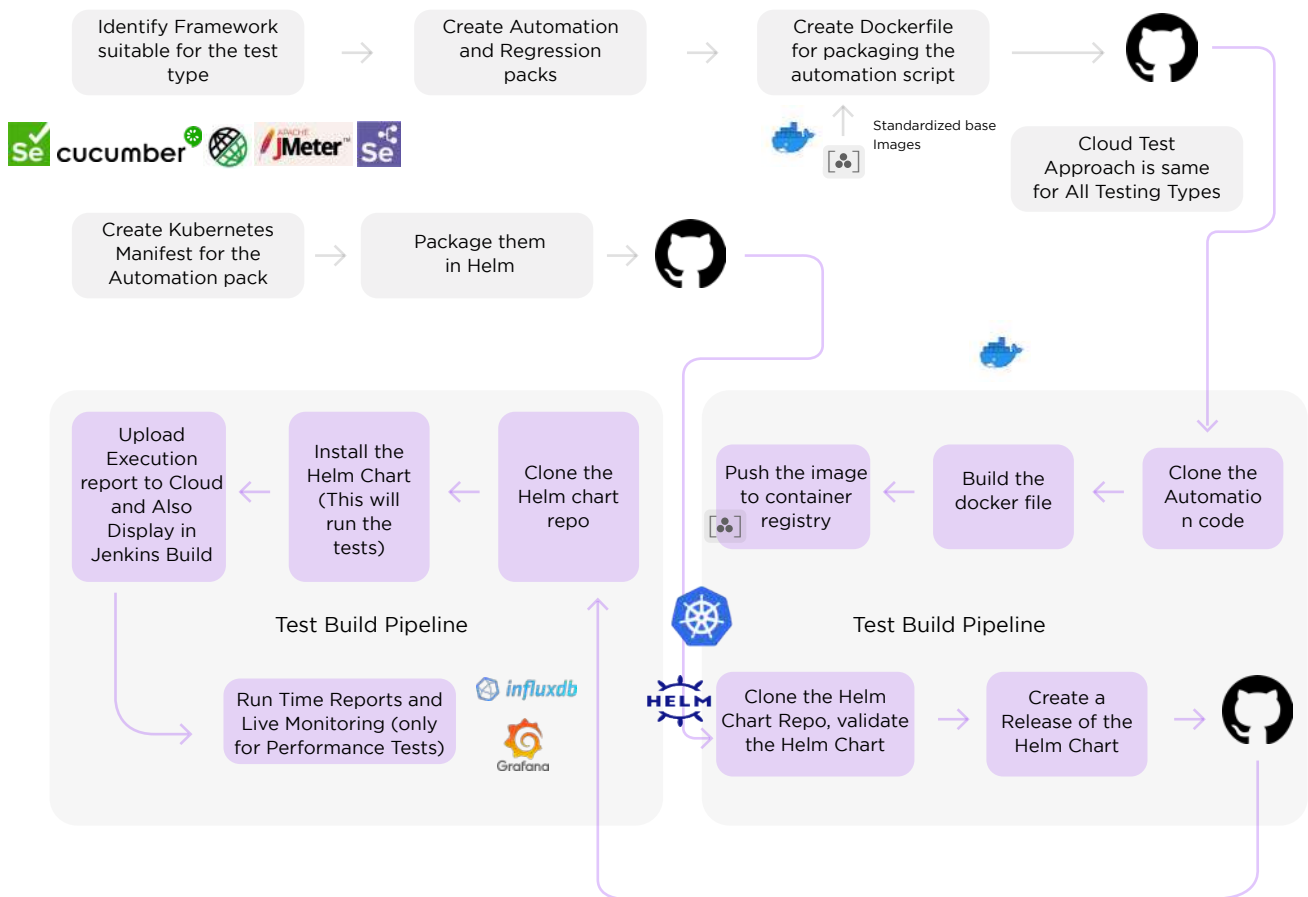


Quality engineering in cloud

Addressing the challenge of faster time-to-market, the organizations should be well equipped to deliver a quality application on any infrastructure that meets the customer's demands. The diagram below shows how quality checks can be done in a cloud-based environment.

This approach would remain constant for any type of test. This feature makes it a *plug-and-play framework* that can be adopted and implemented quickly for any project.

QE Cloud Strategy



Cloud Quality Engineering also comes with additional benefits that help in further improving the time-to-market SLA; listed below are the benefits that are seen when cloud-based frameworks for tests are implemented.



Frequent Releases Bundled with Sanity Tests: Dark and Canary Releases



On-Demand Environments:

Ability to tear down and re-create infrastructure on demand using code.



Optimized Cost:

Utilization of infrastructure based on needs and usage (Pay as per use model).

Conclusion

Today, quality engineering has become one of the “big bangs” in the industry. Most organizations are now leaning towards adopting modern tooling for ensuring holistic quality. This paves the way for new ways of testing that have been highlighted here. To remain competitive in the business, it is critical to adopt a new mindset and apply modern testing principles.

About the Authors



Sriram Iyengar
Technical Product Owner

Sriram has around 16 years of experience in IT, with major experience in Test Automation, Cloud Engineering, DevOps. Sriram is an Innovative and Tech Savvy person and has great interest in learning new technologies and programming languages. Currently working as a Technical Product Owner

On a personal front, Sriram is a movie freak and loves watching movies in any language as long as subtitles are readable. He is a big fan of John Grisham books. A sports enthusiast person, who likes playing Cricket and Badminton.



ABOUT US

STL Digital - Engineering for Experience

STL Digital is a global IT Services company that enables enterprises and industries to experience the future of digital transformation. With an end-to-end portfolio of services across product engineering, software, cloud, data and analytics, enterprise application services, and cyber-security, STL Digital works with global businesses to deliver innovative experiences and operational excellence with agility. STL Digital is a wholly owned subsidiary company of Sterlite Technologies Limited (STL), one of the industry's leading integrators of digital networks, providing All-in 5G solutions (<https://www.stl.tech/>)

To know more about STL Digital, contact us: contact.us@stl.tech or click on <https://stldigital.tech/>