QUALIZEAL

How to Build a Successful and Scalable DevOps Test Strategy?

When Development (Dev) Compliments Operations (Ops), Testing Process Instate Quality Assurance.

DevOps testing strategy is a critical aspect of software development that requires careful planning, execution, and monitoring. In today's dynamic business landscape, companies demand faster and more reliable software releases, making DevOps testing an essential component of any successful software development lifecycle.

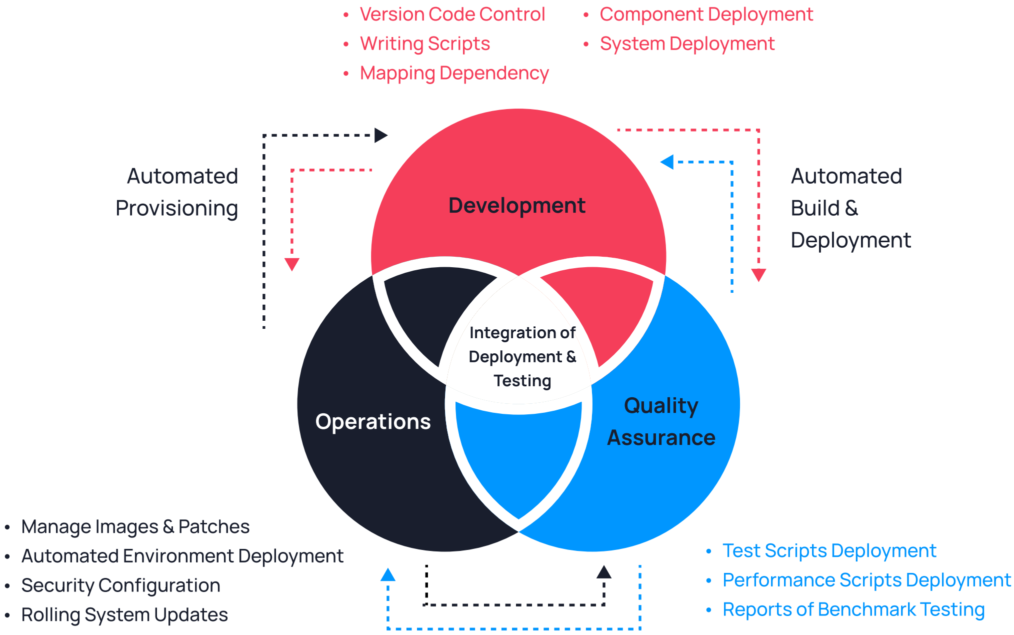
> "DevOps is not a goal, but a never-ending process of continual improvement."
>
> - Jez Humble, co-author of "Continuous Delivery"

In the traditional waterfall model, developers write the code and hand it over to the QA teams, who test the software for bugs. This process is time-consuming, and delays in testing can result in production delays and system crashes.

DevOps, on the other hand, enables teams to work collaboratively and iteratively throughout the software development lifecycle. By breaking down the barriers between teams, DevOps ensures that development, operations, and testing teams work together from the beginning to enable faster delivery of high-quality software, fewer production delays, and reduce friction between teams. It generates real-time feedback, which allows continuous improvements and optimizes the development process.

- A report by DORA found that high-performing DevOps organizations deploy code **46** times more frequently than low performers and have a lead time for changes that is **440** times faster.

- According to a report by MarketsandMarkets, the DevOps market size is expected to grow from USD **3.0 billion** in 2020 to USD **10.3 billion** by 2025, at a CAGR of **27.6%** during the forecast period.



• Version Code Control
• Writing Scripts
• Mapping Dependency
• Component Deployment
• System Deployment
Automated Provisioning
Automated Build & Deployment
Development
Integration of Deployment & Testing
Operations
Quality Assurance
• Manage Images & Patches
• Automated Environment Deployment
• Security Configuration
• Rolling System Updates
• Test Scripts Deployment
• Performance Scripts Deployment
• Reports of Benchmark Testing

Continuous Integration (CI) and **Continuous Delivery (CD)** are two essential components of the DevOps pipeline that ensure smooth and timely delivery of software applications.

CI involves frequently integrating the code changes into the main codebase, followed by automated testing and validation of each commit, thereby providing quick feedback to the development teams. This helps detect any defects or issues early in the development process, leading to improved code quality and faster release cycles.

CD is the next step in the process, where the code built in the previous stage is pushed to production. The code goes through various stages like QA, staging, etc., and is automatically tested and verified before being released. This process of automated testing and validation helps identify and resolve any issues before the code is released into production, leading to improved application quality and faster time to market.

However, for successful execution of the DevOps pipeline, the code integrated through CI and pushed through CD must be thoroughly tested for any bugs. This is where QualiZeal outperforms others in building a DevOps testing strategy.

Why building DevOps strategy is important?

DevOps is neither a tool nor a physical mechanism. It's a philosophy or mindset you adopt to remove bottlenecks in the software development lifecycle. And building a strategy breaks down the silos between development, operations, and other teams. DevOps streamlines collaboration and enables quick responses to cater to changing market demands. Framing the right strategy is a prerequisite to availing these benefits.

- **Faster Time-to-Market:** DevOps enables organizations to deliver software faster by automating many of the tasks associated with software development and deployment, resulting in reduced cycle times and faster time-to-market.

- **Improved Collaboration:** DevOps promotes a culture of collaboration between development and operations teams, resulting in improved communication, alignment, and teamwork.

- **Better Quality:** DevOps emphasizes automated testing, continuous integration, and continuous delivery, which helps to identify and resolve defects quickly, resulting in higher quality software.

- **Increased Efficiency:** DevOps automates many of the repetitive and time-consuming tasks associated with software development, such as testing, deployment, and monitoring, resulting in increased efficiency and productivity.

- **Greater Flexibility:** DevOps provides teams with the ability to quickly and easily respond to changing business requirements and customer needs, making it easier to adapt to new challenges and opportunities.

- **Reduced Risk:** DevOps helps to reduce the risk of downtime, data loss, and other issues by implementing robust security measures and continuously monitoring and testing software.

- **Improved Customer Satisfaction:** DevOps helps organizations to deliver high-quality software that meets the needs of their customers, resulting in improved customer satisfaction and loyalty.

Components of a successful DevOps strategy:

Collaboration:

Collaboration between development, testing, and operations teams is essential to ensure that everyone is working towards the same goals. Collaboration helps identify potential issues early in the development process and ensures that testing is integrated into the DevOps pipeline.

Continuous Integration and Continuous Delivery (CI/CD):

CI/CD is a key component of a successful DevOps strategy. It involves automating the software build, testing, and deployment processes to ensure that new features are integrated quickly and frequently. This approach helps reduce the risk of errors and ensures that the software is always in a releasable state.

Test Automation:

Test automation is critical for enabling continuous testing in the DevOps pipeline. It helps ensure that tests are run quickly and frequently, reduces the risk of human error, and allows developers to get fast feedback on the results of their tests.

Infrastructure as Code:

Infrastructure as Code (IaC) is the practice of managing and provisioning infrastructure using code. This approach helps ensure consistency and repeatability in the DevOps pipeline, reduces the risk of configuration errors, and makes it easier to scale infrastructure as needed.

Shift-Left Testing:

Shift-left testing involves moving testing earlier in the development process. This approach helps identify issues earlier in the development cycle and reduces the risk of defects making it into production. It also helps ensure that quality is built into the software from the start.

Monitoring and Feedback:

Monitoring and feedback are critical components of a successful DevOps strategy. They enable teams to get real-time feedback on the performance and reliability of their software and identify potential issues before they impact customers. This feedback loop helps teams continually improve the quality of their software and the effectiveness of their testing efforts.

How to build DevOps testing strategy?

Identify the key objectives of your DevOps test strategy:

Start by defining the goals of your testing efforts. This may include improving quality, reducing time to market, increasing customer satisfaction, or reducing costs. This will help guide your testing efforts and ensure that you are focusing on the right areas.

Select the appropriate testing tools and frameworks:

Choose the right testing tools and frameworks that align with your objectives. This may include unit testing, integration testing, performance testing, security testing, or any other testing types that are relevant to your application.

Create a culture of continuous testing and feedback:

Develop a culture of testing and feedback that encourages everyone in the organization to participate in testing efforts. This may include developers, testers, product owners, and other stakeholders.

Implement a test automation strategy:

Automate as many tests as possible to reduce manual testing efforts and increase the speed of testing. This can include automated unit tests, integration tests, functional tests, performance tests, and other types of tests.

Integrate testing into your development and deployment pipelines:

Ensure that testing is integrated into your development and deployment pipeline, so that tests are executed automatically as part of the development process. This helps catch issues early on, before they become bigger problems.

Continuously improve your DevOps test strategy:

Regularly review and improve your DevOps test strategy to ensure that it remains effective and aligned with your goals. This may include adding new testing types, updating testing tools and frameworks, or changing your testing processes.

QualiZeal's approach for building a successful DevOps strategy:

Start with a clear understanding of goals and objectives:

Before embarking on a DevOps testing strategy, it is important to have a clear understanding of the goals and objectives of the organization. This includes defining the key business outcomes that the DevOps strategy is intended to achieve, as well as identifying the key metrics that will be used to measure success.

Create a cross-functional team:

Implementing a successful DevOps test strategy requires collaboration between different teams and departments, including development, operations, and testing. Creating a cross-functional team that includes members from each of these areas can help to ensure that the DevOps strategy is aligned with the needs and goals of the entire organization.

Establish a testing framework:

A testing framework is essential for implementing an effective DevOps test strategy. This framework should include a clear definition of the types of tests that will be performed, the tools and technologies that will be used, and the processes and workflows that will be followed.

Automate every testing activity:

Automating tests can reduce the risk associated with continuous integration. It is important to automate only those tests that are feasible and practical to automate. This can help in identifying issues quickly, reduce the testing time, and minimize the risk of human error.

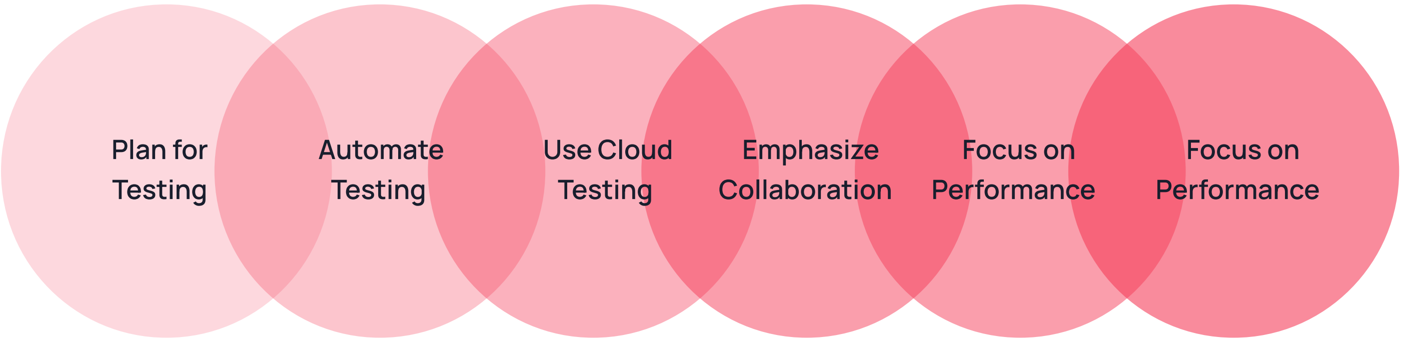
Use pair tests for manual testing:

Pair testing is an efficient way to train team members to keep up with the DevOps pipeline velocity. Knowledge transfer from senior testers is critical to pair testing. A pair from the team tests work simultaneously, with one of them testing and the other analyzing the results. Agile cross-functional teams allow for pair testing from any two participants taking turns, not necessarily only the testing team.

Implement shared responsibility in testing:

Shared responsibility is fundamental to DevOps pipelines to ensure high-quality delivery.

DevOps testing is done by the entire team or teams instead of only designated testers or members of the QA team. It is a way to ensure smoother collaboration with customers or stakeholders of the product.



Implement continuous testing:

Continuous testing is a key component of a successful DevOps test strategy. This involves integrating testing into the development process at every stage, from planning and design to development and deployment. Continuous testing helps to ensure that software is tested thoroughly and consistently throughout the development process.

Use automation tools and frameworks:

Automation is an essential part of DevOps testing, and there are a variety of automation tools and frameworks available that can help to streamline testing processes and reduce the time and effort required to perform tests. Using automation tools can help to increase the speed and efficiency of testing, while also reducing the risk of human error.

Foster a culture of collaboration and continuous improvement:

DevOps is more than just a set of tools and processes - it is a culture of collaboration and continuous improvement. Creating a culture that encourages open communication, feedback, and experimentation can help to ensure that DevOps testing is implemented effectively and continuously improved over time.

Challenges associate with scaling DevOps test strategies:

Managing testing environments:

As the number of teams and applications increases, so does the number of testing environments. Managing multiple testing environments can be complex and time-consuming.

Ensuring consistency:

As the number of teams increases, maintaining consistency in testing becomes more challenging. Ensuring that all teams are following the same testing procedures, using the same tools, and reporting results in a consistent way becomes a challenge.

Ensuring test coverage:

Scaling DevOps test strategies requires testing across multiple applications, platforms, and devices. Ensuring that all tests are executed in a timely manner, and that all critical paths are covered can be a challenge.

Test data management:

With multiple applications and teams, managing test data can become challenging. Teams need to ensure that they have access to the right data sets for their tests, while also protecting sensitive data.

Managing test scripts:

With multiple applications and teams, managing test scripts becomes more challenging. Teams need to ensure that test scripts are up to date, and that changes in one application do not affect other applications.

Collaboration and communication:

As the number of teams and applications increase, collaboration and communication become more challenging. Teams need to work together to ensure that testing is coordinated and that everyone is on the same page.

Key Challenges in Implementing DevOps Test Strategies

Lack of testing expertise

Implementing a DevOps test strategy requires a deep understanding of testing practices, automation tools, and testing frameworks. Many organizations lack the in-house testing expertise needed to effectively implement DevOps testing.

Integration challenges:

DevOps requires close collaboration and integration between development and operations teams. However, integrating testing into the DevOps process can be complex and require significant effort to ensure that all tools, processes, and systems work seamlessly together.

Difficulty in automating testing:

One of the key tenets of DevOps is automation, and this applies to testing as well. However, automating testing can be challenging and time-consuming, particularly for organizations with complex and legacy systems.

Resistance to change:

Adopting DevOps requires significant cultural and organizational change, as it involves breaking down traditional silos between development and operations teams and embracing a culture of collaboration and continuous improvement. Resistance to change from employees, management, or other stakeholders can be a significant challenge in implementing DevOps testing.

How QualiZeal Overcome DevOps Challenges?

- **Invest in training and upskilling:** To overcome the lack of testing expertise, QualiZeal invests in training and upskilling its employees to ensure that they have the necessary knowledge and skills to implement DevOps testing effectively. This includes providing training in testing practices, automation tools, and testing frameworks.

- **Use open-source tools and frameworks:** To overcome the challenges of integration and automation, QualiZeal leverages open source tools and frameworks to streamline its DevOps testing process. This allows the company to take advantage of a wide range of tools and technologies that are readily available, and reduce the time and effort required to integrate different systems.

- **Establish clear testing processes:** To ensure that testing is properly integrated into the DevOps process, QualiZeal establishes clear testing processes that define the roles, responsibilities, and workflows for testing activities. This helps to ensure that testing is properly integrated into the overall development process and that testing activities are coordinated and streamlined.

- **Foster a culture of collaboration and communication:** To overcome resistance to change and improve collaboration between development and operations teams, QualiZeal fosters a culture of collaboration and communication. This involves promoting open communication, encouraging feedback and continuous improvement, and breaking down silos between different teams and departments.

- **Start small and scale up gradually:** To ensure that DevOps testing is properly implemented and scaled effectively, QualiZeal adopts a phased approach, starting small and scaling up gradually. This allows the company to test its strategies and processes in a controlled environment and gradually build up its capabilities over time.

DevOps Test Strategies

DevOps is the most reliable approach to speed up the software deployment and its delivery maintaining high-quality and agility. However, building a testing strategy that supports continuous development and integration demands the experience QualiZeal carries in its arsenal.

Implementing a successful DevOps strategy is not just about achieving a specific goal but an ongoing continual improvement process. Therefore, we focus on critical aspects such as tool and framework selection and performance tracking metrics for pass/fail pattern identification. We also apply best documentation practices to CI/CD process.

QualiZeal nurtures a culture of collaboration, agility, and innovation within its development teams. We capitalize the latest DevOps testing tools such as Selenium, Eggplant, Jenkins, and Appium to scale test automation for web, software, and mobile to sustain the long-term test coverage.

Our cloud-based solutions automate testing activities regardless of the complexity of test scenarios while integrating CI/CD tool. By leveraging QualiZeal's in-house resources, you get faster feedback and multiple platforms with minimum time for test executions.