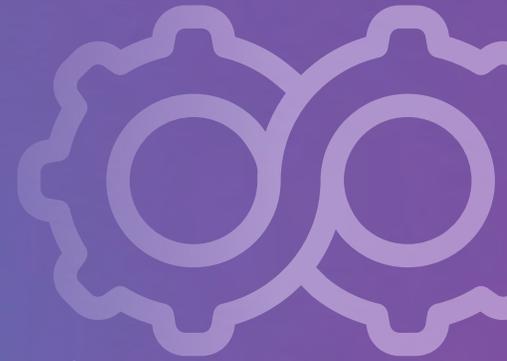


# 8 Best Steps Toward Enabling Shift Left Testing for DevOps

Leverage powerful DevOps capabilities to improve stability and security before production



One defining ability of DevOps is the ability to shift testing left with the right DevOps planning and development tools. As the name suggests, testing is conducted in the early stages of the SDLC, and involves seamless collaboration between developers, testers, and QA teams, resulting in quality applications with less stability or security issues.

Here are some considerations for organizations or teams looking to shift testing left in their DevOps CI/CD practice:

## 1. Clarify test requirements during the planning phase

Involve testers and QA teams early during planning to clarify the types of tests required, tools to be used, and agreed-upon testing requirements.

## 2. Adopt a collaboration-friendly DevOps platform

Enable seamless collaboration and testing between developers and testers with an end-to-end DevOps platform featuring multiple code repositories and test environments. Your chosen platform should also integrate well with multi-code IBM® i applications or environments to facilitate testing on these popular core systems.

## 3. Ensure your chosen testing tool is code-agnostic

Ideally your chosen DevOps platform or testing tool should be code-agnostic, with the ability to integrate with multi-code applications on popular systems like IBM i. This reduces the variety of code-centric test tools needed and streamlines test workloads.

## 4. Determine the required types of tests needed

Whether it's unit testing, integration testing, or regression testing, ensure you can provision the right environment on your chosen DevOps platform. For multi-code testing, ensure you've standardized tools and processes to reduce friction between tests and speed up the process.



## 5. Establish consistent reporting and communication lines

Keep stakeholders and managers in the loop to provide continuous feedback or define scope changes. This preserves the velocity and veracity of tests, while ensuring added features or functions pass both technical and usability tests before entering production.

---

## 6. Retain consistently high testing standards with automation

Reduce the volume of manual testing required while elevating test consistency, frequency, and accuracy with automation throughout the testing process. This reduces the burden of testing on developers while minimizing the risk of manual errors during early testing phases.

Building repositories of tests that are automatically triggered when new code is introduced improves testing productivity and reduces delay, allowing agile development and CI/CD to continue uninterrupted.

---

## 7. Enforce clear separation of duties for testing

Certain data or security regulations mandate a clear separation of duties between those who write the code and those who test it. Make sure your chosen DevOps test platform has strong access controls and permissions built in to enforce compliance while enabling testers to perform their jobs with confidence.

---

## 8. Ensure scalability of test environments and capabilities

As application functionality grows and involves more developers, your test capabilities must also scale to meet development scope and speed. An integrated DevOps platform with high levels of visibility across multiple teams allows for more efficient coordination and communication between development, testing, and QA stages.

**Get the guide** to implementing DevOps CI/CD practices that accelerate development, empower experimentation, and improves time-to-market for IBM® i systems.

Trial the Rocket® DevOps modernization solution that's designed to enable end-to-end CI/CD practices for multi-code environments like IBM® i.