

The Impact of Testing on a DevOps Pipeline

Bob Fornal



Entrepreneur

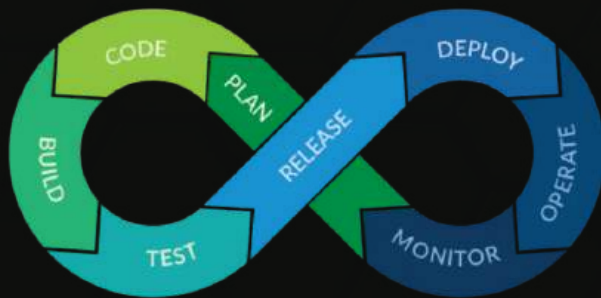
Code-squid provides solid, in-depth frontend training that is supported with real-world code projects. Blessed husband and proud father of two.



Senior Solutions Developer Leading EDJE, Inc.

Passionate about learning, testing, mentoring, speaking, and personal growth.

What is Devops?



The Goal ...

To build better, faster and more responsive software **by** bringing Development and Operations teams together.

DevOps ...

Is a **cultural shift** to remove the barriers for **shorter and more frequent** software deliveries, and to respond well to changing business requirements.

The **cultural shift** depends on continuously optimizing:

- Workflow
- Architecture
- Infrastructure

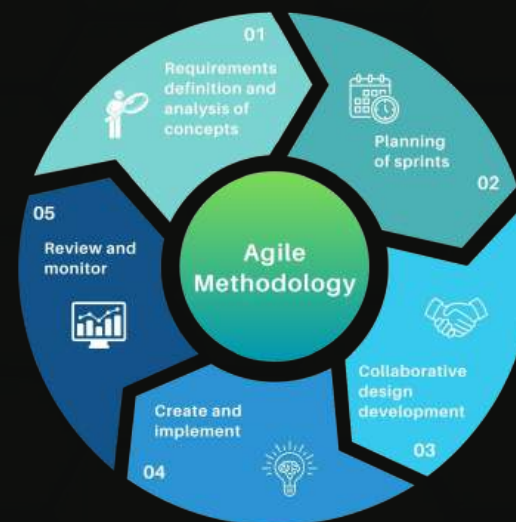
What is the Agile Model?

Software developed with the Agile Model adheres to the following four basic priorities:

1. Individuals and interactions over processes and tools;
2. **Working software** over comprehensive documentation;
3. Customer collaboration over contract negotiation; and
4. **Responding to change** over following a plan.

In Agile development:

- A **test-first approach** is used, rather than a test-at-the-end approach, when developing.
- Code is developed **and tested** in small increments of functionality.



What is Continuous Testing?

Testing is done by the whole team

(not just designated testers or quality assurance professionals).

- Shared responsibility for attaining the highest quality.
- Equal responsibility for the success of the project.

Unit and Component Testing ...

- Written by developers.
- Using **Test-Driven Development** (TDD)
- TDD helps think through the desired behavior.

Collaboration ...

- QA, devs, and non-technical / business.
- Using tools like **Cucumber (GIVEN, WHEN, THEN format)**.

DevOps Continuous Testing: What does it mean?

Activity	Detail
Early Detection of Defects	<ul style="list-style-type: none">• Automated tests are run as soon as code changes are committed, providing immediate feedback to developers.• This early detection helps prevent the accumulation of defects, reducing the time and effort required to address issues later in the development cycle.
Faster Release Cycles	<ul style="list-style-type: none">• Accelerate software delivery, ensure high-quality digital experiences across diverse platforms, and meet evolving customer demands.
Improved Collaboration	<ul style="list-style-type: none">• Collaboration between development and operations teams.• Automated testing provides a common framework for both teams to work within, promoting shared responsibility for quality.
Enhanced Test Coverage	<ul style="list-style-type: none">• Leverages automation to achieve extensive test coverage over a wide range of scenarios.• Ensuring all aspects of the application are thoroughly validated, reducing the risk of undetected issues in production.

DevOps Continuous Testing: Strategy

Basic Agile Testing and Development Methodology:
(and the aim of most DevOps projects)

- Automate manual processes.
- Equal responsibility.
- The whole team tests.
- Testing is done continuously: **Early, Often, and After.**

Roadblocks

- Handoffs (e.g. from Development Team to a QA Team).
- Required signoff; bureaucratic approval.
- Slow, error-prone manual processes and checks.
- Poor-quality, big-bang rollouts that fail in production.

DevOps Continuous Testing: Implementation

Activity	Detail
Continuous Testing (shift-left testing)	<p>A key element of a successful DevOps Continuous Testing strategy for pipelines.</p> <ul style="list-style-type: none">• Performed earlier in the software lifecycle,• With the goal of increasing quality,• Shortening long test cycles,• Reducing the possibility of software defects making their way into production code.
Continuous Development	<p>The first stage in implementing a Continuous Delivery DevOps pipeline.</p> <ul style="list-style-type: none">• Developers write Unit Tests for each piece of code they write (prefer TDD).• New features are checked into a central code base, built, and tested.
Continuous Integration	<p>The use a version control system and integrate work frequently in the same location (main branch).</p> <ul style="list-style-type: none">• Each change is tested in order to detect any integration errors as quickly as possible.• The build happens automatically rather than when a developer triggers the process manually.• Once an application passes all required tests, it's then released into production.

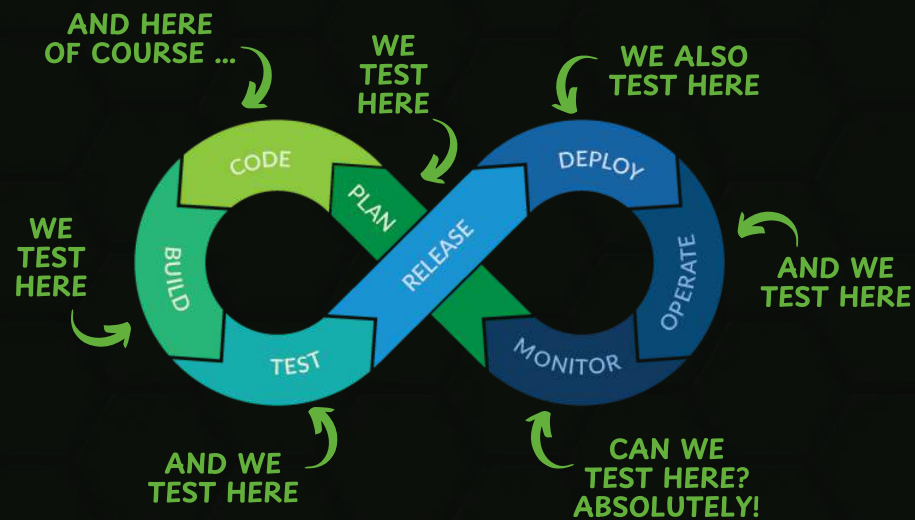
DevOps Continuous Testing: Pros

Activity	Detail
Quick Delivery	<ul style="list-style-type: none">• New functionality to users within minutes
Fast Feedback	<ul style="list-style-type: none">• Instant feedback that, in turn, allows rapid response to customer demand.
Handling Risk	<ul style="list-style-type: none">• Risk-based Testing: Identifying the highest-risk areas of a software application and focusing testing efforts on those areas. This approach helps to ensure that the most critical areas of the application are thoroughly tested.• Reduces testing time and effort in areas that are less important.

DevOps Continuous Testing: Cons

Activity	Detail
Bugs in Production	<ul style="list-style-type: none">• Risk of releasing buggy software into production faster.
Loss of Customers	<ul style="list-style-type: none">• Alienate users and cause your company to lose customers.
Tech Limitations	<ul style="list-style-type: none">• TA 2022 Mabl report found that 53% of respondents indicated that technology limitations were holding back their testing strategies.
Budget Constraints	<ul style="list-style-type: none">• 1-in-3 aspiring DevOps teams reported budget as their most pressing obstacle.

DevOps Testing: Where?



DevOps Continuous Testing: Methodologies (1/3)

Activity	Detail
Shift-Left Testing	<ul style="list-style-type: none">• Prioritizing early software and system testing in the Software Development Life Cycle (SDLC) to minimize debugging challenges later in the process.
Shift-Right Testing	<ul style="list-style-type: none">• Emphasizing testing towards the end of the SDLC to enhance user experience, overall performance, failure tolerance, and functionality.
Smoke Tests	<ul style="list-style-type: none">• Initial, cursory screenings (manual or automated) for glaring flaws in software, offering a swift and cost-effective solution for eliminating gross errors.
Unit & Component Testing	<ul style="list-style-type: none">• Ideal for small-scale stress, load, volume, or memory leak checks in early developmental stages to identify potential degradations.

DevOps Continuous Testing: Methodologies (2/3)

Activities	Detail
Integration & Message Testing	<ul style="list-style-type: none">• Checking errors when software modules collaborate, with continuous testing in DevOps virtualizing missing dependencies to assess end-to-end processes and scenarios.
Performance Testing	<ul style="list-style-type: none">• Assessing application software performance, with integrated system testing required to evaluate overall solution performance, accounting for hardware and middleware in the final production environment.
Functional Testing	<ul style="list-style-type: none">• Verifying if the user experience aligns with expectations and if functional workflows are executed across the software system as needed.

DevOps Continuous Testing: Methodologies (3/3)

Activity	Detail
Regression Testing	<ul style="list-style-type: none">• Ensuring no changes in performance, functionality, or dependencies after correcting errors in dependent software, maintaining system performance.
User-Acceptance Testing	<ul style="list-style-type: none">• Also known as application or end-user testing, assessing the application in real-world situations by a subset of intended users.

DevOps Continuous Testing: Best Practices



Automate as Many Tests as You Can

- Continuous Testing means testing early and often.
- Eliminate much of the risk that comes with CI/CD.
- Get quick feedback on application quality.
- Reduces the opportunity for new errors.

Use Pair Testing & Pair Programming

- On non-automated tests.
- Knowledge Transfer is an efficient way to train new members and maintain velocity.
- One tests; the other analyzes or reviews the results.
- Useful in breaking down communication barriers.

