September 22, 2020

Bad Code Kills: 5 Essential Quality Gates You Need in Your CloudBees Core Pipeline

Eric Mizell
VP Solution Engineering, OverOps

DEVOPS WORLD by CloudBees

Today's Agenda

- Why Aren't Traditional Quality Practices Sufficient?
- 5 Static and Dynamic Quality Gates You Need in Your CloudBees Core Pipeline
  - #1: Static Analysis
  - #2: New Errors
  - #3: Critical Exceptions
  - #4: Resurfaced Issues
  - #5: Unique Error Volume
- Demo

© 2020 All Rights Reserved.

As teams increase speed with CI/CD workflows…

Quality becomes a greater challenge.

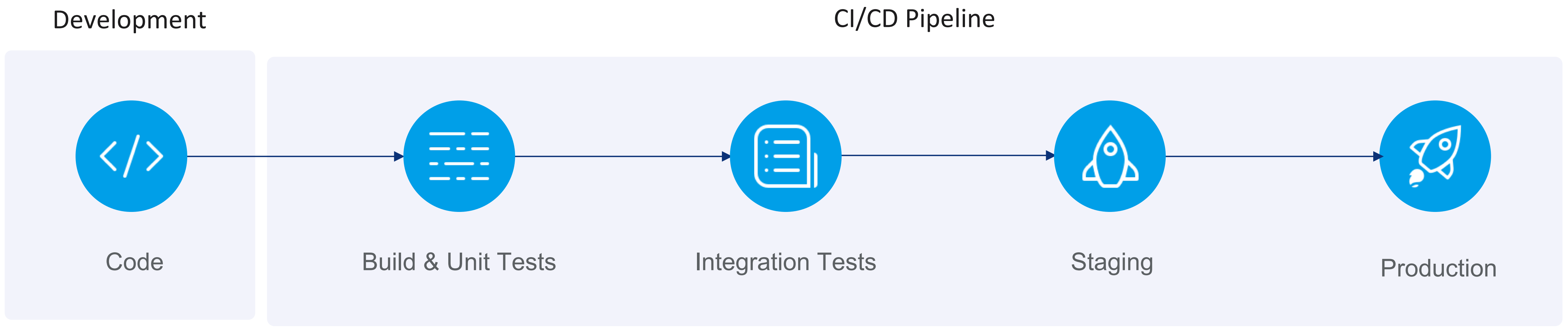
What’s your level of confidence when releasing new code to production?

DEVOPS WORLD by CloudBees

Why aren’t traditional quality practices sufficient?

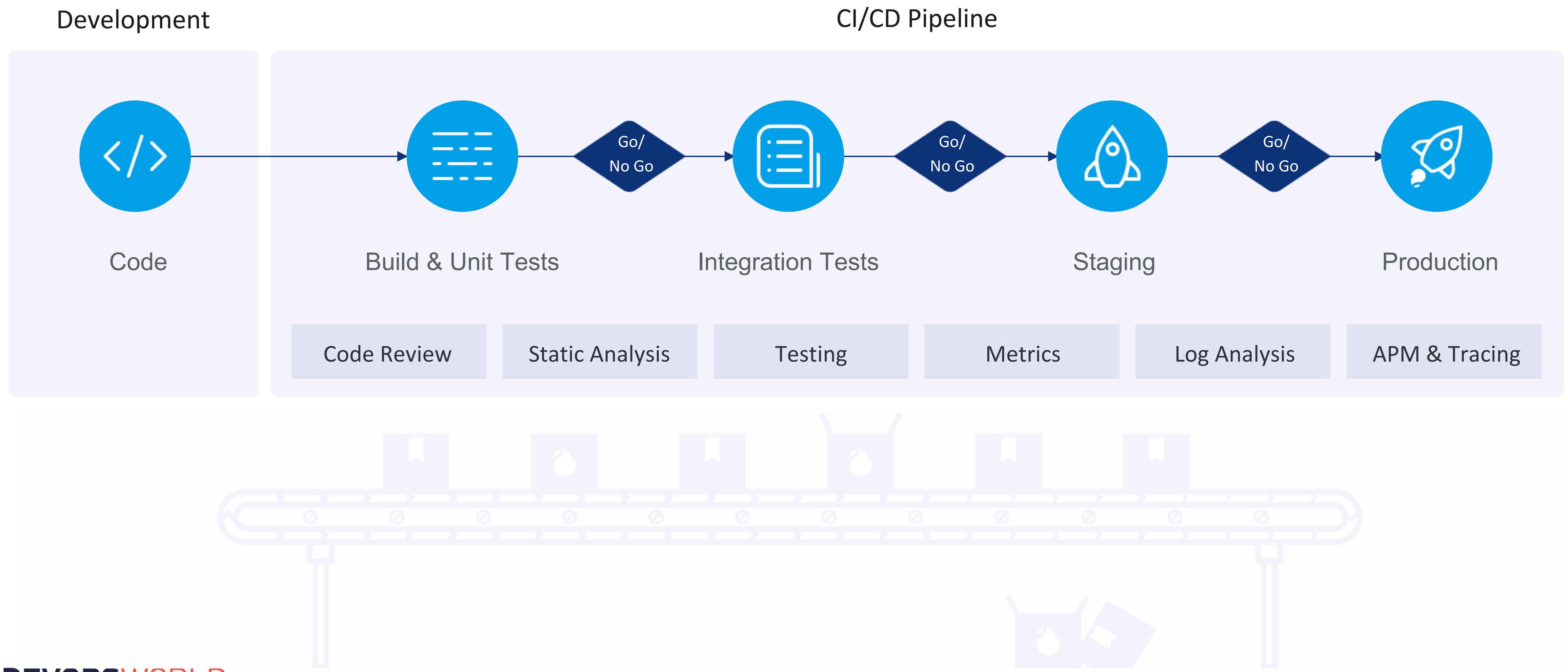
DEVOPS WORLD by CloudBees

The Current Software Delivery Lifecycle



© 2020 All Rights Reserved.

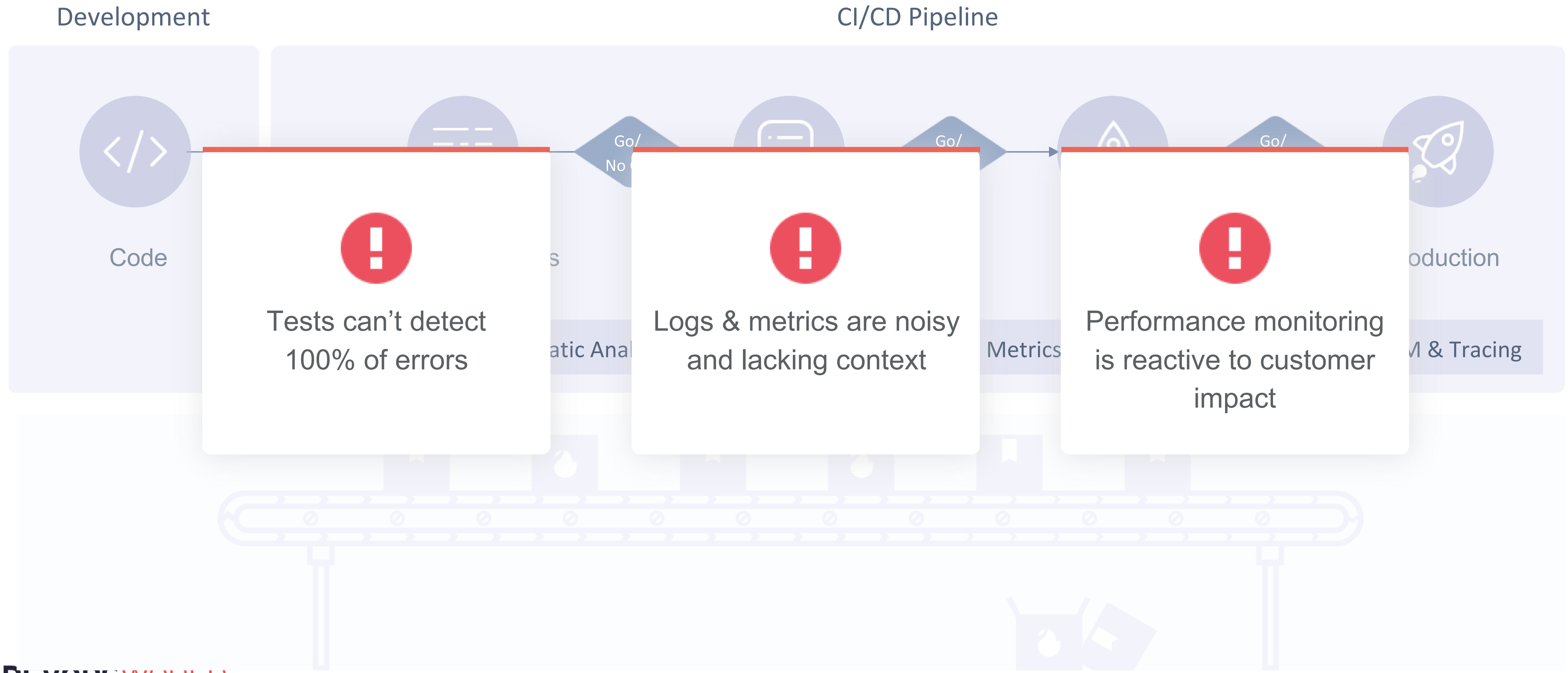
How Do You Ensure Reliability Across the Pipeline Today?



DEVOPSWORLD by CloudBees

© 2020 All Rights Reserved.

Yet Code Still Breaks!



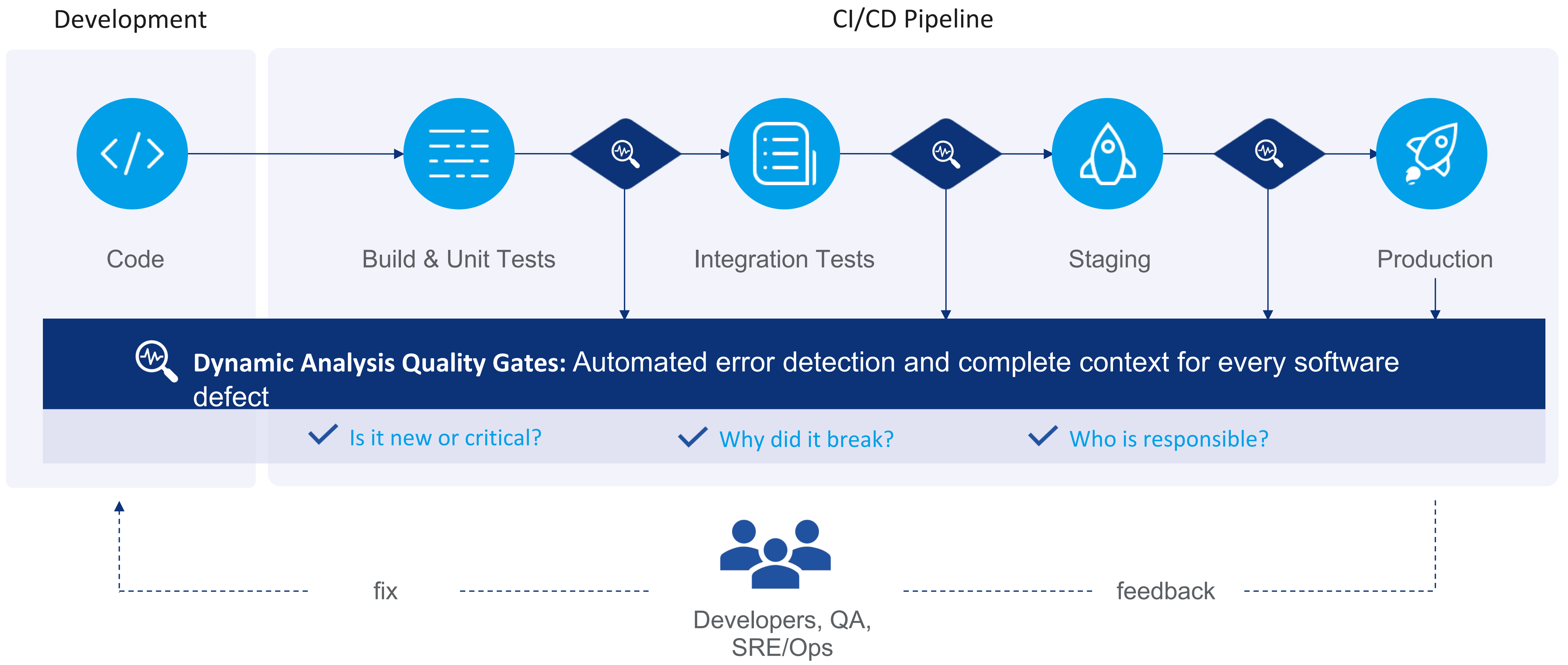
Tests can't detect 100% of errors

Logs & metrics are noisy and lacking context

Performance monitoring is reactive to customer impact

© 2020 All Rights Reserved.

Introducing Dynamic Analysis Quality Gates



Dynamic Analysis Quality Gates: Automated error detection and complete context for every software defect

- ✓ Is it new or critical?
- ✓ Why did it break?
- ✓ Who is responsible?

DEVOPSWORLD by CloudBees

© 2020 All Rights Reserved.

5 Static & Dynamic Quality Gates You Need in Your CloudBees Core Pipeline

DEVOPS WORLD by CloudBees

Quality Gate 1: Static Analysis

Can the code run? Does it follow best practices in readability/security?

✓ **Pass:**
No style issues or code smells detected. No known security vulnerabilities

✗ **Fail:**
Release introduces a new code smell or security vulnerability

Example Scenario:

In a recent change, you forgot to close a file descriptor which may cause memory leakage or you didn't wrap an if statement with parenthesis.

With static analysis, you can catch resource leaks and code style issues.

DEVOPSWORLD
by CloudBees

© 2020 All Rights Reserved.

Dynamic Quality Gate 2: New Errors

Did the release introduce any errors that didn't previously exist?

✓ **Pass:**
No new errors are detected

✗ **Fail:**
Release introduces a new error never previously seen

Example Scenario:

Your latest release in a mission-critical Java app introduces a 'ResourceNotFoundException' in a database that has never before experienced this issue.

Using the New Errors Quality Gate, CloudBees automatically marks the release as unstable and blocks it from moving forward to production.

© 2020 All Rights Reserved.

Dynamic Quality Gate 3: Critical Exceptions

Did the release introduce any severe/showstopping errors?

✓ **Pass:**
No critical exception types detected

✗ **Fail:**
The release introduces one or more predefined critical errors. In a Java application, some examples could be:

- NullPointerException
- IndexOutOfBoundsException
- InvalidCastException

Example Scenario:

Your latest release for a Java application introduces a ‘NullPointerException.’

Using the Critical Exceptions Quality Gate, CloudBees/Jenkins automatically marks the release as unstable and blocks it from moving forward to production.

© 2020 All Rights Reserved.

Dynamic Quality Gate 4: Resurfaced Errors

Did an error that was previously resolved appear again in the current release?

✓ **Pass:**
No previously resolved errors are detected

✗ **Fail:**
Release resurfaces a previously resolved issue

Example Scenario:

You encounter a ParseException in a new release that you had previously addressed and resolved.

Using the Resurfaced Errors Quality Gate, CloudBees/Jenkins automatically marks the release as unstable and blocks it from moving forward to production.

© 2020 All Rights Reserved.

Dynamic Quality Gate 5: Unique Error Volume

Did the release introduce an unusually high number of many discrete errors?

✓ **Pass:**
Total number of unique errors falls below the standard baseline

✗ **Fail:**
Release introduces an unusually high number of unique errors

Example Scenario:

Your latest release introduces 12 unique events. Your baseline value is set to 10 events.

Using the Unique Error Volume Quality Gate, CloudBees/Jenkins automatically marks the release as unstable and blocks it from moving forward to production.

© 2020 All Rights Reserved.

Demo

DEVOPS WORLD by CloudBees

Questions?

DEVOPS WORLD by CloudBees

Thank You!

Visit OverOps at our virtual booth to win a Visa gift card and check us out at www.overops.com

DEVOPS WORLD by CloudBees