

Spira-ling into Quality Pipelines



Kim Trott

Business Development • Coveros, Inc.



Quick Overview

- Quality through End-to-end Traceability
Adam Sandman, Inflectra
- Quality Gates: Forcing Functions to Bake Quality In
Jeff Payne, Coveros
- SpiraPlan DevSecOps Pipeline Integration Demo
Hugo Sanchez, Coveros
- Lessons Learned from Years of Building (quality through)
Pipelines at Coveros - *Jeff Payne, Coveros*
- Q & A
- Meet and Greet in Creative Spaces



Spira-ling into Quality Pipelines

Quality through
End-to-end
Traceability

 @Coveros | @inflectra



Adam Sandman

Director • Inflectra Corporation



Inflectra Corporation

inflectra.com • sales@inflectra.com



How SpiraPlan Enables Progress Tracking and Visualization through Dashboards



Firstly, What is Quality?



Quality Control vs. Quality Assurance



OK, So What is Traceability?



Why does Traceability Matter?



For many industries it's the law

- ISO 9001 / AS 9100
- Aviation - DO-178C
- Medical - 21 CFR Part 11
- Automotive - ISO 26262



Traceability - Finding the Root Cause

- Let's work the problem people. Let's not make things worse by guessing



Gene Kranz, Apollo 13

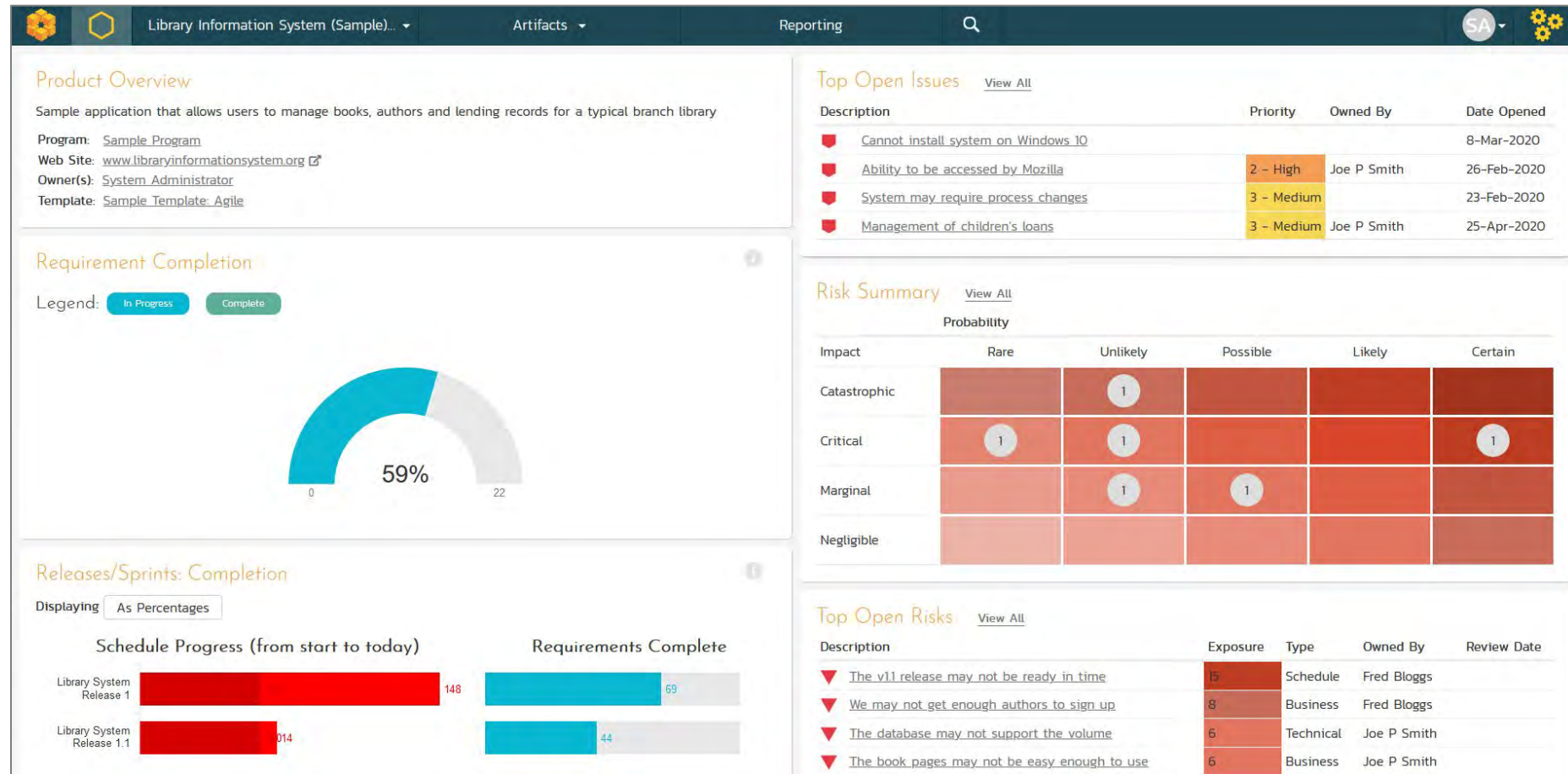
SpiraPlan - Traceability & Visibility



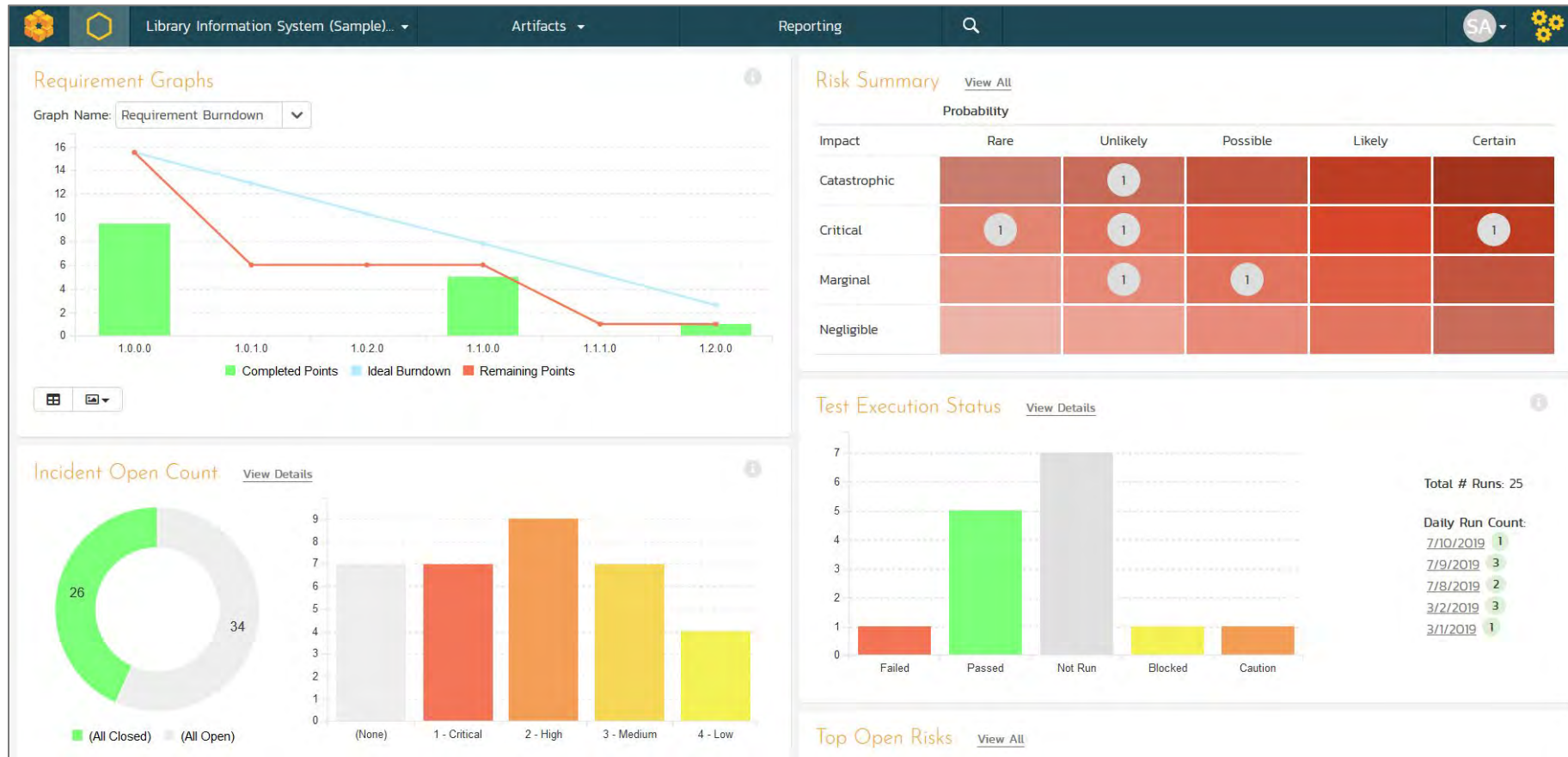
Key Metrics and Artifacts

- Requirements
 - Tests and Coverage
- Risks
 - Known vs. unknowns
- Defects
 - Velocity and root cause

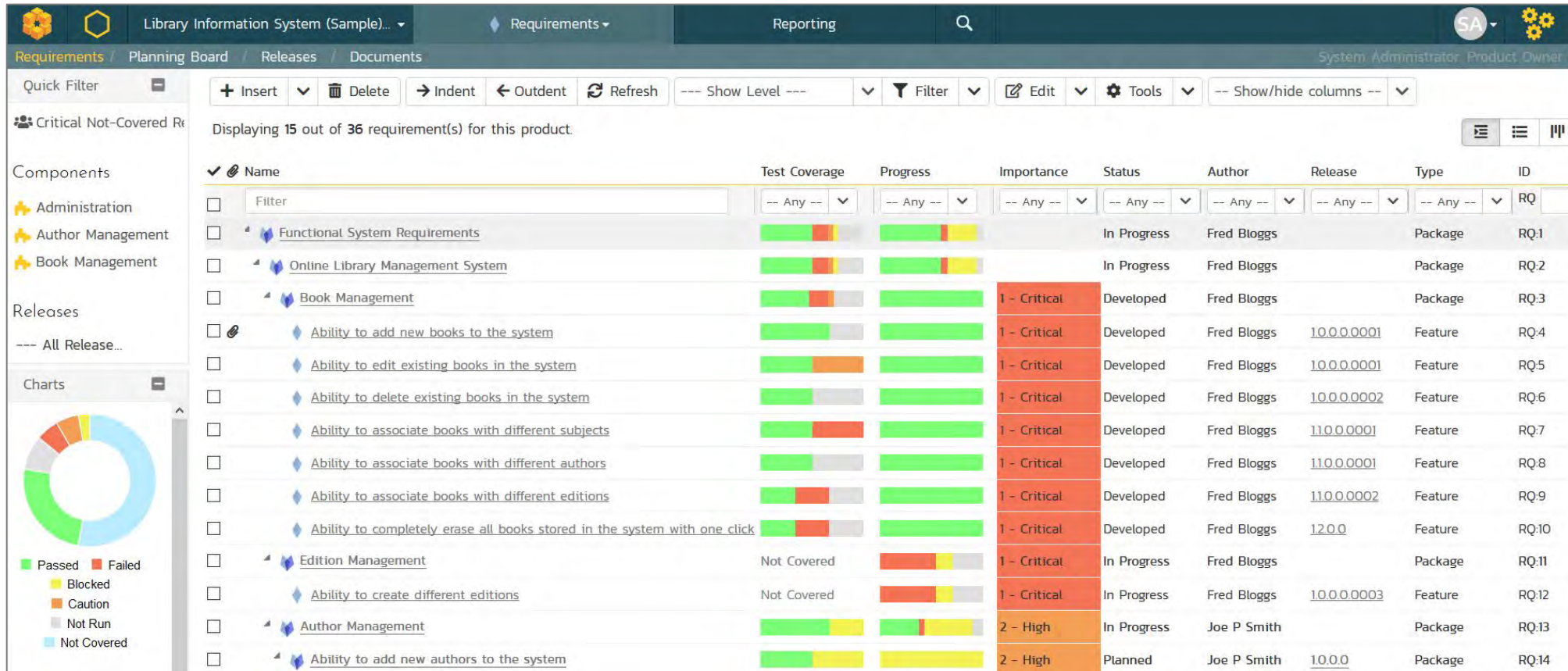
Dashboards



Dashboards



Test Coverage



Traceability to Code

Library Information System (Sample)... Commits Reporting

Source Code / Commits Source Code Commit File Action System Administrator Product Owner

Related Files

- rev0003
- SpiraTestConfiguration.java
- SpiraTestExecute.java

Related Commits

- SpiraTestConfiguration.java
- rev0003 (3-Nov-2020)
- rev0001 (1-Nov-2020)
- rev0000 (31-Oct-2020)

rev0003 SpiraTestConfiguration.java

1024 KB KB Changed By: Fred Bloggs (11/3/2020 7:13:49 AM) Previous Commit: rev0001

Changes * </> rev0001 * </> rev0003 *

Previous (rev0001) => Current (rev0003)

```
1 package com.inflectra.spiratest.addons.junitextension;
2
3 import java.lang.annotation.*;
4
5 /**
6  * This defines the 'SpiraTestConfiguration' annotation used to specify the authentication,
7  * project and release information for the test being executed
8  *
9  * @author Inflectra Corporation
10 - * @version 3.0.1
11 + * @version 3.0.2
12 *
13 + @Retention(value=java.lang.annotation.RetentionPolicy.RUNTIME)
14 @Target(value=java.lang.annotation.ElementType.TYPE)
15 public @interface SpiraTestConfiguration
16 {
17     String url ();
18     String login () default "";
19 - String password () default "";
20 - int projectId () default -1;
21 - int releaseId () default -1;
22 - int testSetId () default -1;
```

Inflectra and Coveros



Introduction to Coveros

- Coveros is a partner we trust when our customers need help with Spira implementation and customization. We work closely together.
- As software people, Coveros understands the role of tooling in improving the production of software and can help you achieve the optimal balance between implementation of best practices and effective tool support.
- Coveros understands when to leverage a process change. They have a deep understanding of our workflows and they know how to customize Spira to support you.
- As you seek to optimize delivery, Coveros will help you understand and streamline tool usage and remove any road blocks in your pipeline implementation and integration with Spira.



Spira-ling Into Quality Pipelines

Quality Gates:
Forcing Functions to
Bake Quality In

 @Coveros | @inflectra



Jeffery Payne

CEO, Founder • Coveros, Inc.



- 30+ Years in Software
- Founder of 2+ software companies
- Passionate about Agile & DevOps

Jeffery Payne

CEO, Founder • Coveros, Inc.



Coveros, Inc.

coveros.com • solutions@coveros.com



DevOps & DevSecOps

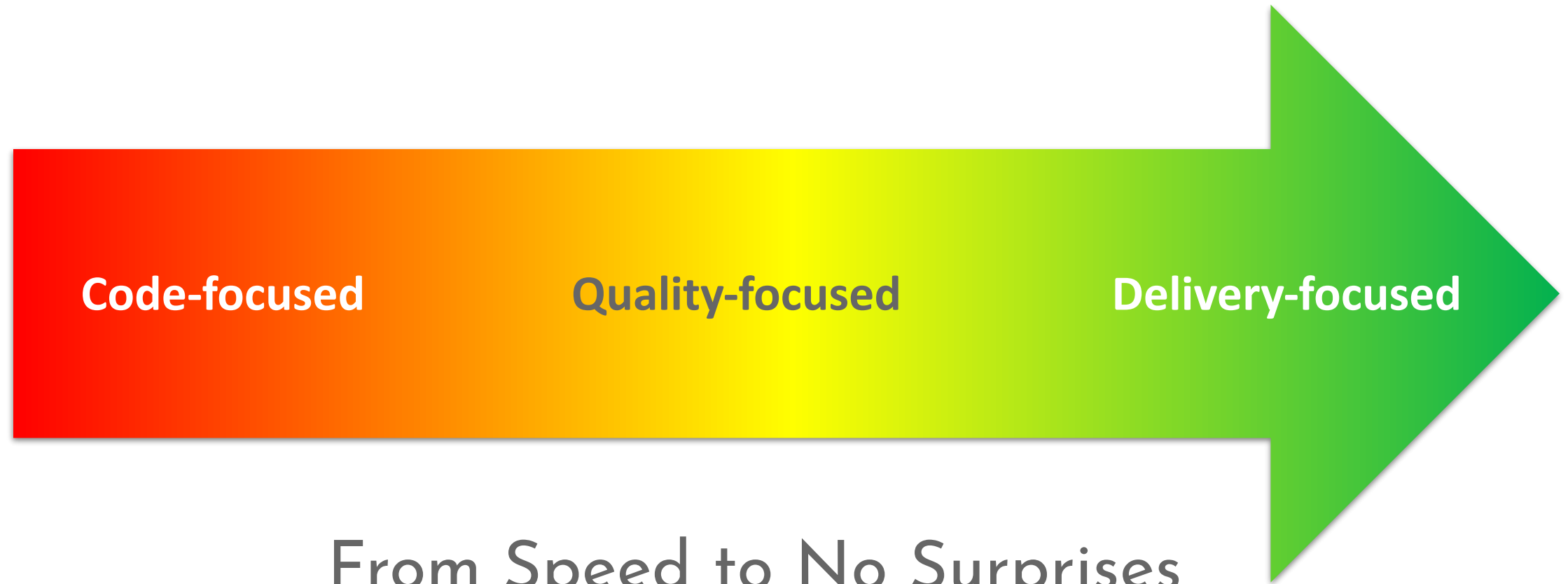
Development and Operations Tension:

- Development wants fast creation and innovation
- Operations wants bug-free, secure, and stable systems

A pipeline is a set of tools, technologies, and automated processes, enabling teams to consistently build, test, and deploy software meeting Quality & Security Requirements.



DevOps / DevSecOps Pipelines



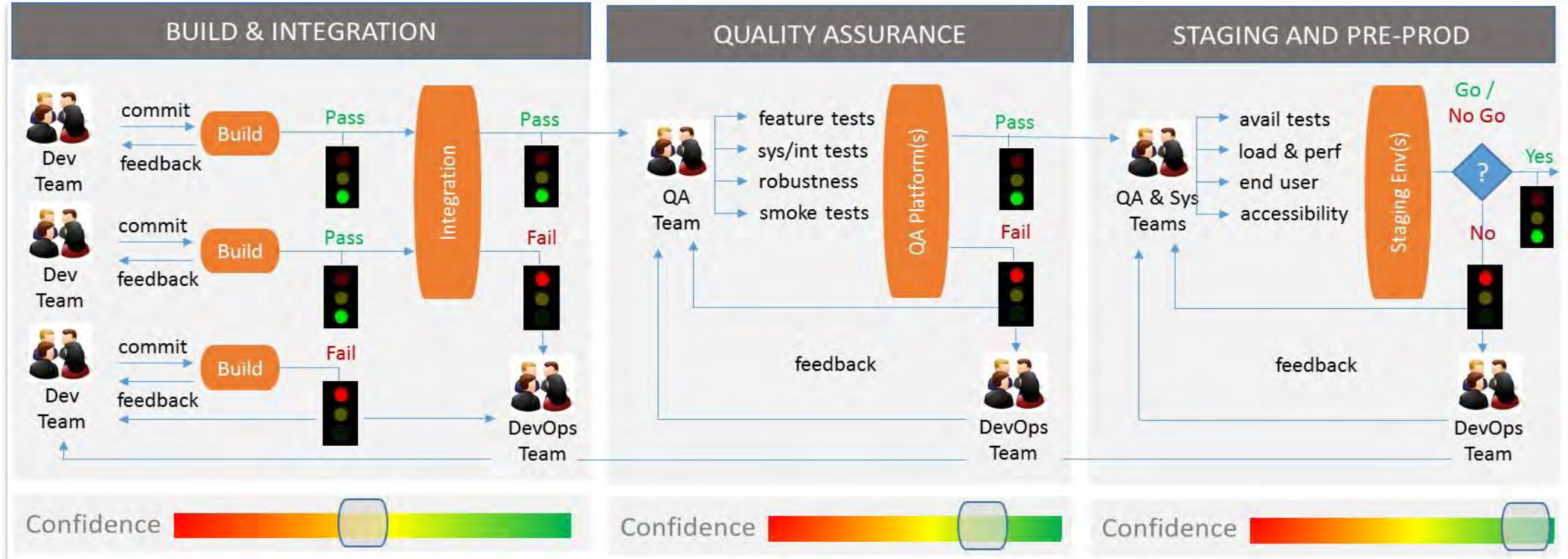
DevOps / DevSecOps Pipelines

Core components and functions:

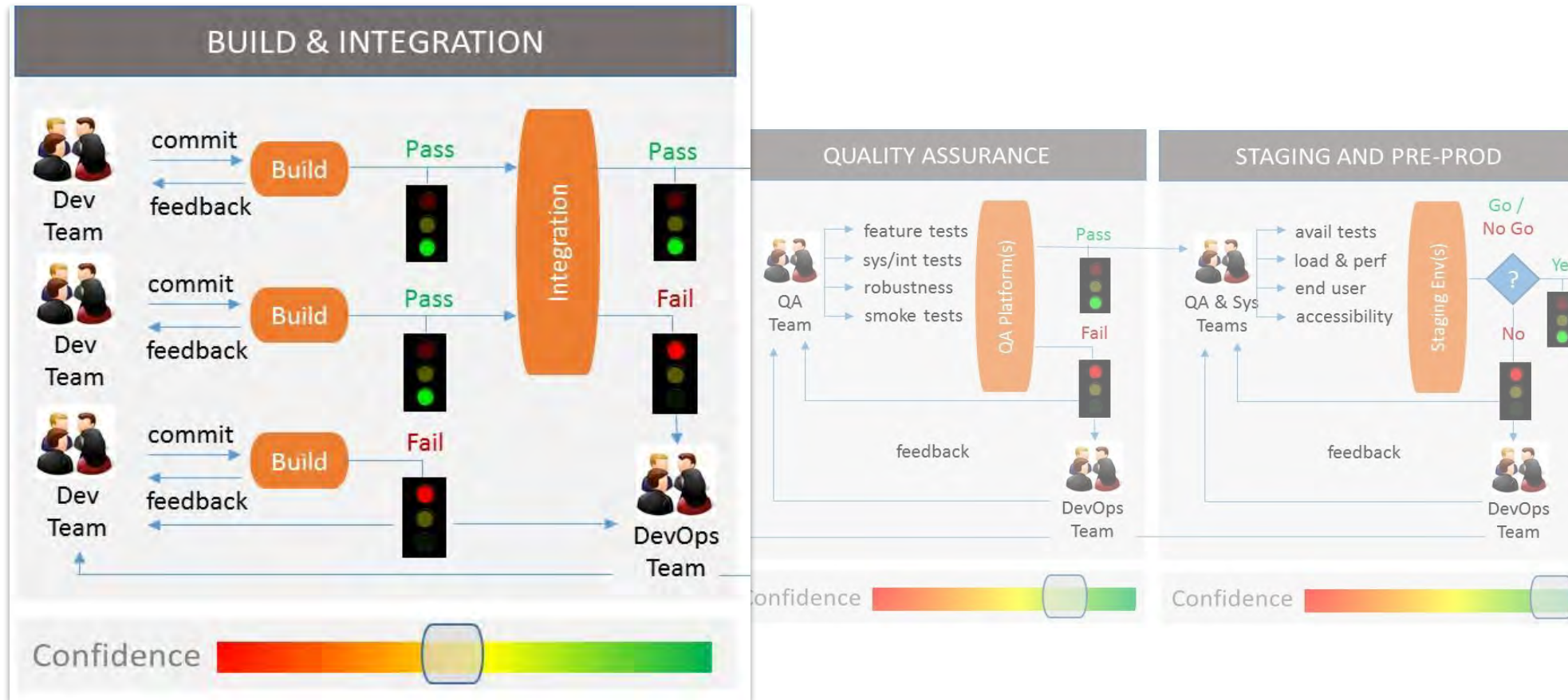
- **Continuous Integration Tools** merge code in a central repository
- **Continuous Deployment Tools** manage configuration and infrastructure
- **Continuous Testing Tools** automate tests
- **Automation Servers** orchestrate pipelines



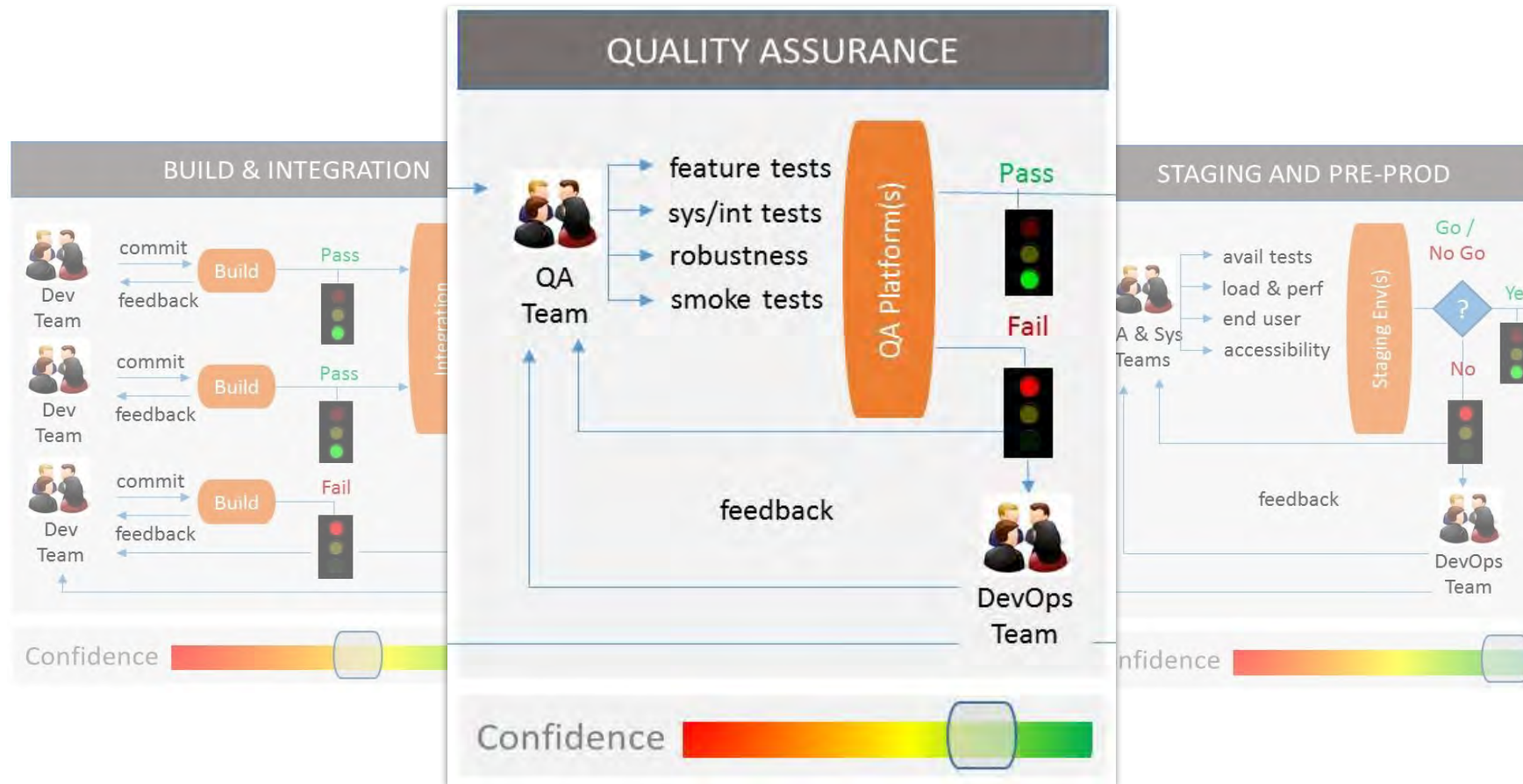
Incorporating Quality & Security Gates



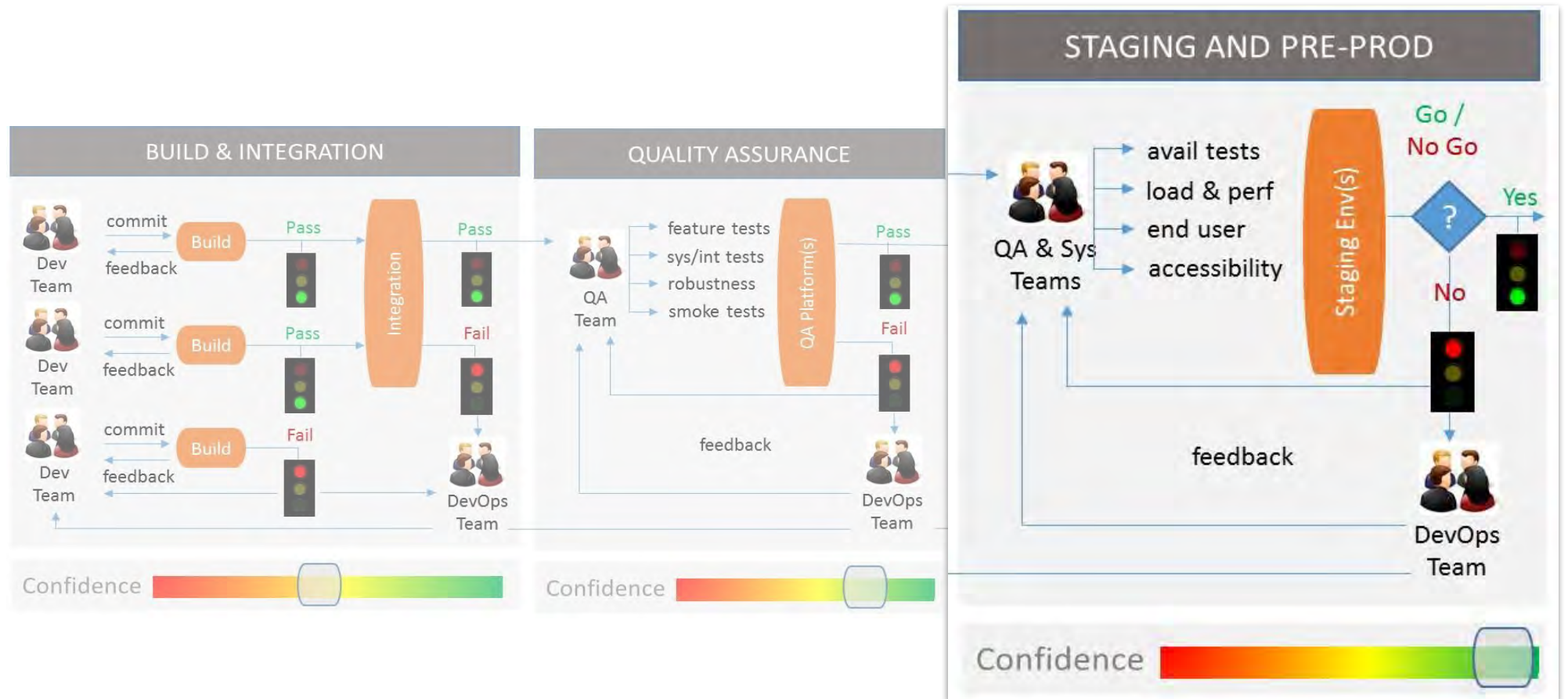
Incorporating Quality & Security Gates



Incorporating Quality & Security Gates



Incorporating Quality & Security Gates



Spira-ling Into Quality Pipelines

Integrating a
DevSecOps Pipeline
with the Spira Family
of Tools

🐦 @Coveros | @inflectra



Hugo Sanchez

Managing Consultant • Coveros, Inc.



Coveros, Inc.

coveros.com • solutions@coveros.com



Quick Overview

- **What is a DevSecOps Pipeline?**
- Finding defects before production quickly and preventing continuous bugs
- **Integrating Spira with a Pipeline**
- Jump-start your efforts to get a working pipeline on your project

Our Sample Application

Librarian login Name: <input type="text" value="name"/> Password: <input type="text" value="password123"/> <input type="button" value="login"/>	Register librarian Name: <input type="text" value="name"/> Password: <input type="text" value="password123"/> <input type="button" value="register"/>	Borrow a book Book: <input type="text" value="locked - no data"/> Borrower: <input type="text" value="borrower"/> <input type="button" value="lend"/>	Register a book Book: <input type="text" value="book"/> <input type="button" value="register"/>	Register a borrower Borrower: <input type="text" value="borrower"/> <input type="button" value="register"/>	Database versioning <input type="button" value="Clean"/> <input type="button" value="Migrate"/> <input type="button" value="Clean / Migrate"/> Database console Console help
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Built from the ground up using a variety of Agile development best practices like BDD and TDD.

<https://github.com/7ep/demo/>



Our Sample Application

Librarian login Name: <input type="text" value="name"/> Password: <input type="text" value="password123"/> <input type="button" value="login"/>	Register librarian Name: <input type="text" value="name"/> Password: <input type="text" value="password123"/> <input type="button" value="register"/>	Borrow a book Book: <input type="text" value="locked - no data"/> Borrower: <input type="text" value="borrower"/> <input type="button" value="lend"/>	Register a book Book: <input type="text" value="book"/> <input type="button" value="register"/>	Register a borrower Borrower: <input type="text" value="borrower"/> <input type="button" value="register"/>	Database versioning <input type="button" value="Clean"/> <input type="button" value="Migrate"/> <input type="button" value="Clean / Migrate"/> Database console Console help
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Testing includes: Unit tests, Database tests, Acceptance Tests, Security static analysis, API tests, UI BDD tests, UI tests, Security dependency analysis, Performance tests and Mutation tests.



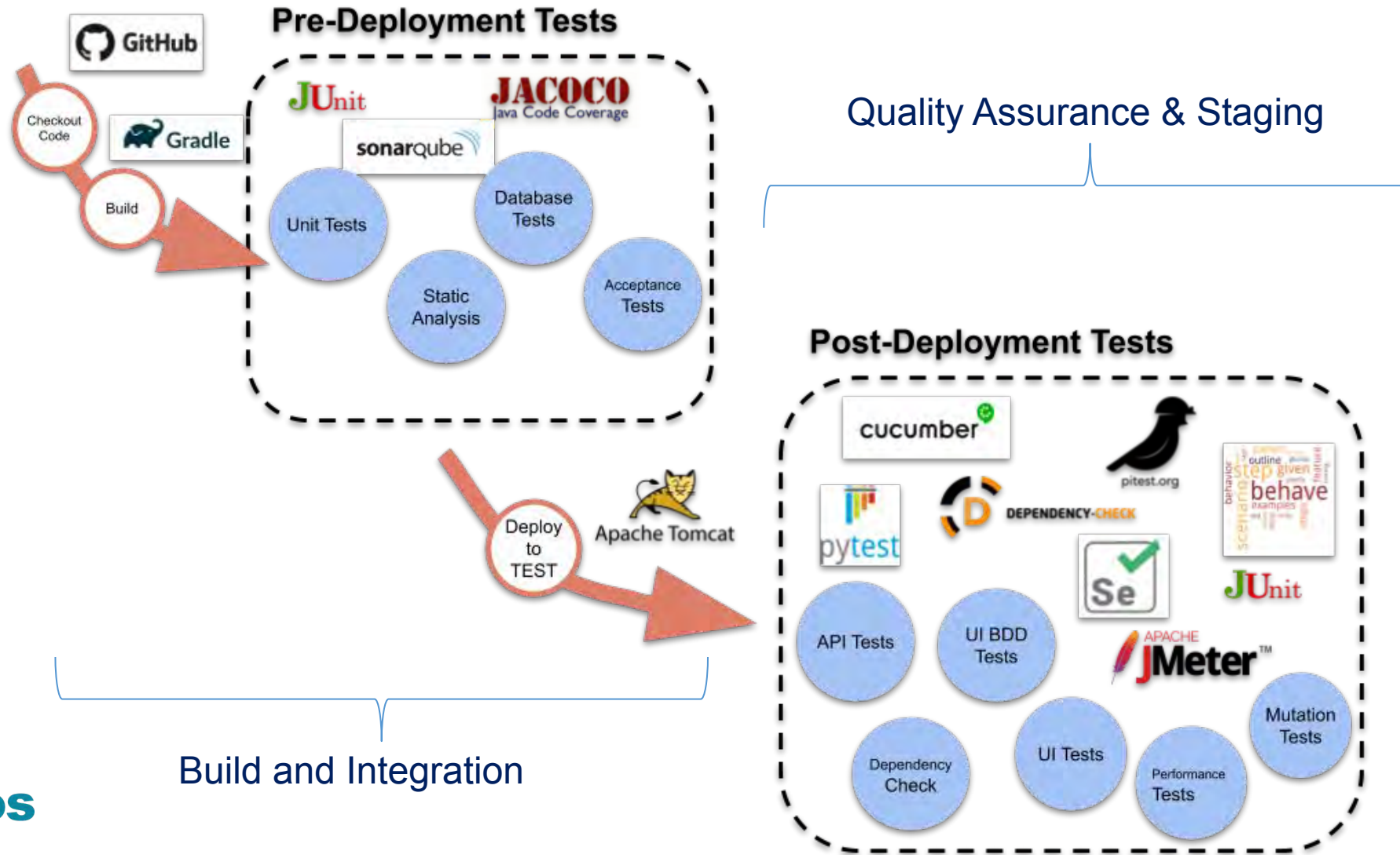
<https://github.com/7ep/demo/>



Application Artifacts in Spira

- Release
- Requirements
- Test Cases
- Test Runs
- Product QA Dashboard

Our Demo Pipeline

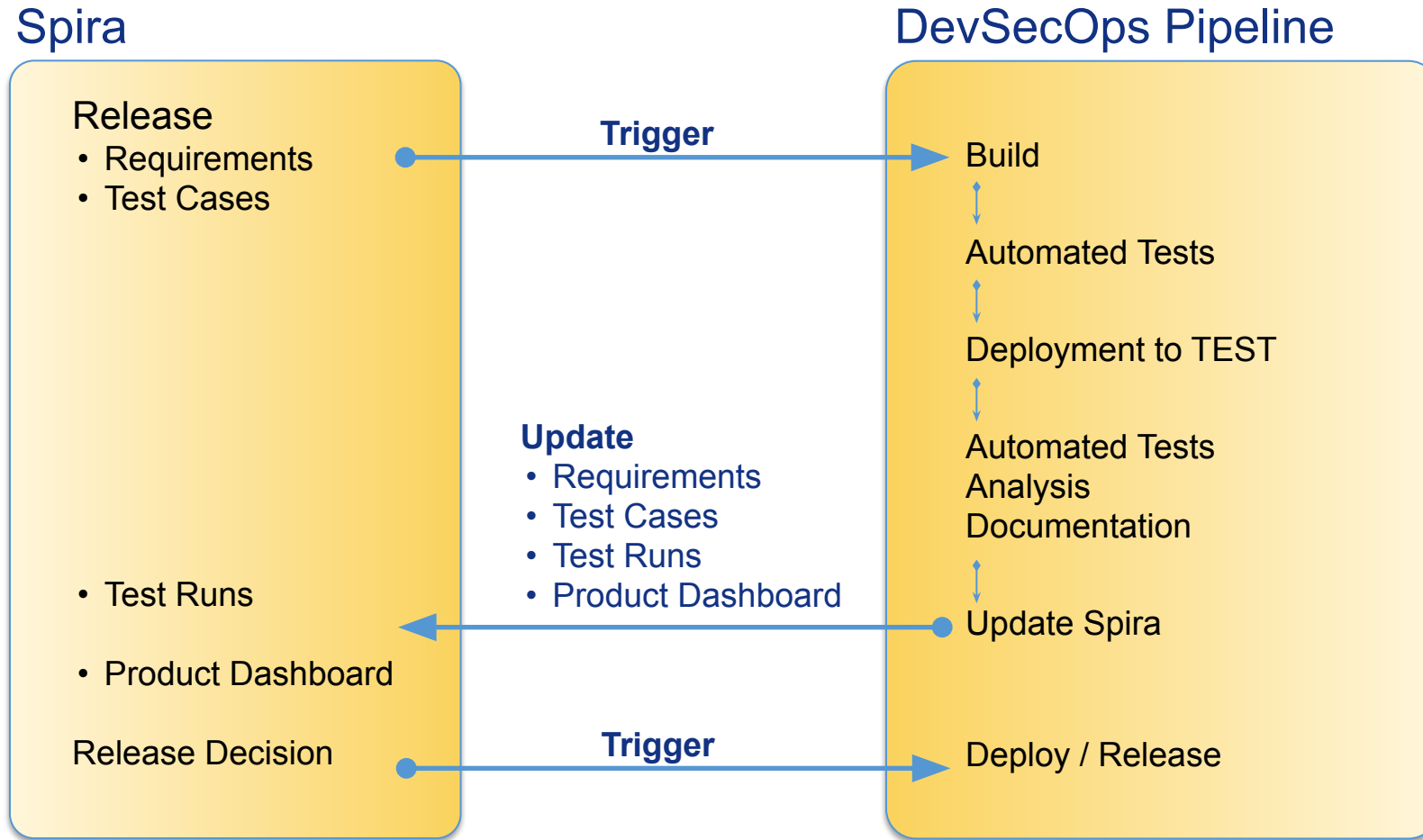


Our Demo Pipeline

- Code Repository - GitHub
- Build - Gradle
- Unit Tests - JUnit / JACOCO
- Database Tests - Executed with JUnit
- Automated Acceptance Tests - Cucumber
- Security: Static Analysis - SonarQube
- Deploy to TEST - Deployed with Tomcat
- API Tests - PyTest
- UI BDD Tests - Quick readiness check, Behave (similar to Cucumber, Gherkin syntax), Selenium
- UI Tests - JUnit, html UnitTests and Selenium
- Security: Dependency Analysis - OWASP Dependency Check
- Performance Tests - JMeter
- Mutation Tests - PiTest
- Send Results to Spira - Python / Jenkins Pipeline
- Build Documentation - Javadoc



Integrating Spira



DevSecOps Pipeline Demo



Updated Artifacts in Spira

- Release
- Requirements
- Test Cases
- Test Runs
- Product QA Dashboard



Key Takeaways

- The move to DepSecOps is happening. Testing is needed otherwise what ends up in production might not reflect the level of quality you desire
- Quality is important on a pipeline because you don't want continuous bugs
- Pipelines enable rapid feedback (Defects) from your customers and the ability to detect defects prior to production quickly and fix them
- You already have a foundation with the Spira toolset and we can integrate them for you



Spira-ling Into Quality Pipelines

Lessons Learned from
Years of Building
(quality through)
Pipelines at Coveros

 @Coveros | @inflectra



Lessons learned

- Start with simple quality & security gates
- Focus first on automating you code-focused gates
- Expect to have to coach development teams on how to pass gates
- Watch for those who circumvent the process



Questions?



Coveros helps companies accelerate the delivery of secure, reliable software with agile methods



Agile/DevOps Transformations

Agile Software Development

Agile Testing & Automation

DevOps Engineering

DevSecOps

Agile, DevOps, DevSecOps,

Testing Training

Open Source Products



Thank You!



Enjoy Creative Spaces

Chat with us about
hosting your next
event in a
Creative Space

