



 **CODE71**



Agile QA



A Practical Guide to implementing Agile QA process on Scrum Projects

Syed Rayhan

Co-founder, Code71, Inc.

Contact: srayhan@code71.com

Blog: <http://blog.syedrayhan.com>

Company: <http://www.code71.com>

Product: <http://www.scrumpad.com>

My Background

Career

- Co-founder, Code71, Inc.
- 13+ years of total experience
- Co-author of “Enterprise Java with UML”

Expertise

- Iterative incremental development
- Technology planning and architecture
- On-shore/Off-shore software development using Agile/Scrum

Interests

- Cultural aspect of self-organizing team
- Scrum for projects delivered remotely
- Agile engineering practices

Agenda

Section **1** Introduction

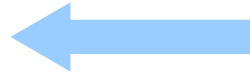
Section **2** Holistic View of QA

Section **3** Individual Practices

Section **4** A Case Study

Section **5** Recap

Section **6** Q&A



What to Expect

Context

- Teams and organizations are adopting Agile/Scrum
- Teams struggle with making the transition from waterfall to Agile/Scrum

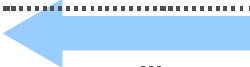
Focus

- Build common base of understanding
- Develop a set of guidelines- process, roles, and team composition
- Address typical questions asked

Key Takeaways

- How to perform QA on an Agile/Scrum project
- Agile/QA best practices

Agenda

Section 1	Introduction
Section 2	Holistic View of QA 
Section 3	Individual Practices
Section 4	A Case Study
Section 5	Recap
Section 6	Q&A

The challenges?

- Is QA part of the development team?
- Can we fit QA in the same iteration as development?
- Who does QA?
- Does QA costs more in Agile as product seems to change from sprint to sprint?
- How can we scale Agile QA?
- Do we need “test plan”?
- Who defines test cases?
- Are story acceptance tests enough?
- When do we know testing is done?
- Do we need to track bugs?

What is QA (Quality Assurance)?

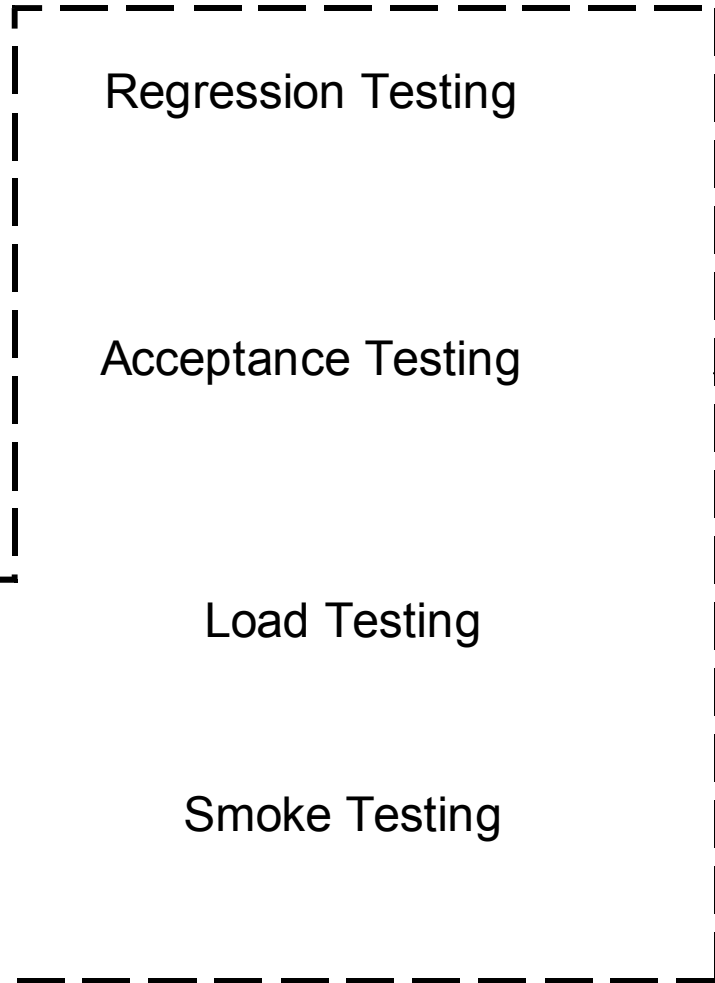
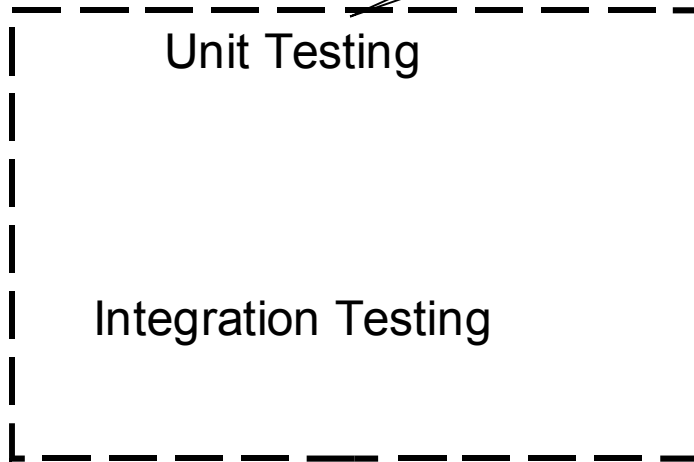
To ensure Software is working right

How?

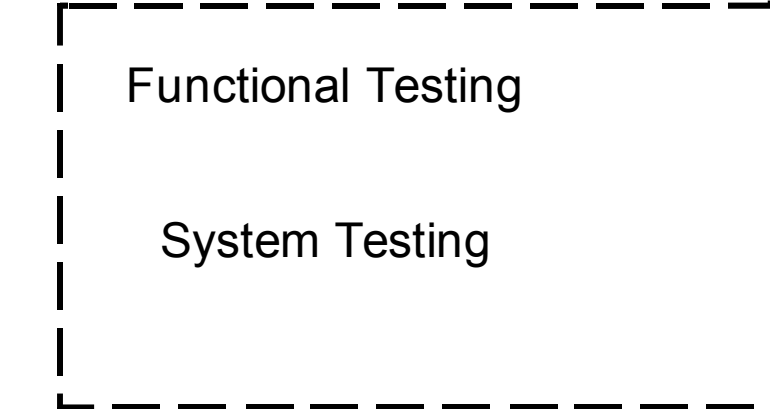
Test, Test, Test

Types of Testing?

White
Box



Black
Box



Who Performs What?

What?	Who?	When?	Automation?
Unit Testing	Developer	Coding	Always
Integration Testing	Developer	Coding	Always
System Testing	Tester	Test	Possible
Regression Testing	Developer/Tester	Build/Test	Possible
Acceptance Testing	Client/Users	Deployment/ Delivery	Possible
Smoke Testing	Tester/Support Engineer	Deployment	Possible
Load Testing	Performance Engineer	Deployment	Always

Right tools for right tests?

Test	Tool
Unit Testing	NUnit, JUnit, Mock, DBUnit
Integration Testing	Unit test tools, HttpUnit, SoapUI, RESTClient
System Testing	Selenium, Fit, WET, Watir, WatiN
Regression Testing	Unit test tools, System test tools
Acceptance Testing	FIT, FitNesse
Smoke Testing	Regression test tools
Load Testing	JMeter, Httpperf

What is missing?

Assumptions

1. Right Spec
2. Right Design
3. Right amount of Tests
4. Right Tests

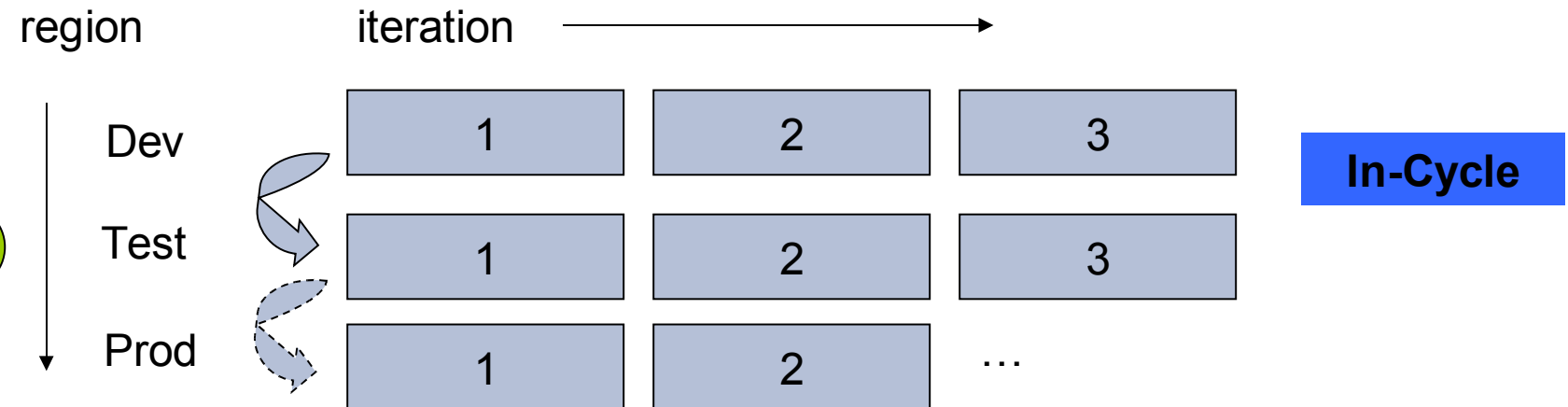
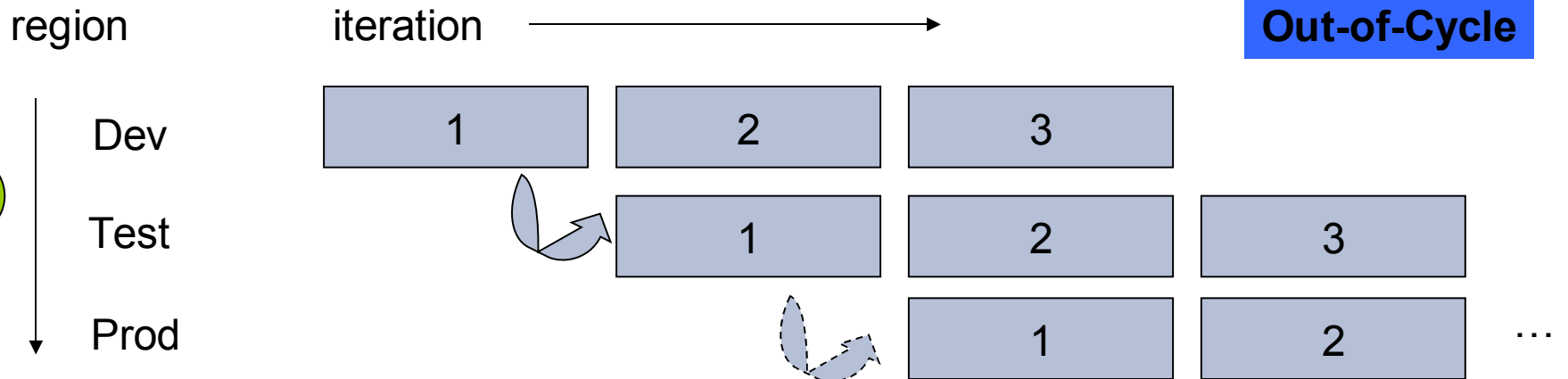
Reality?

Assumptions are farther from truth

Measures

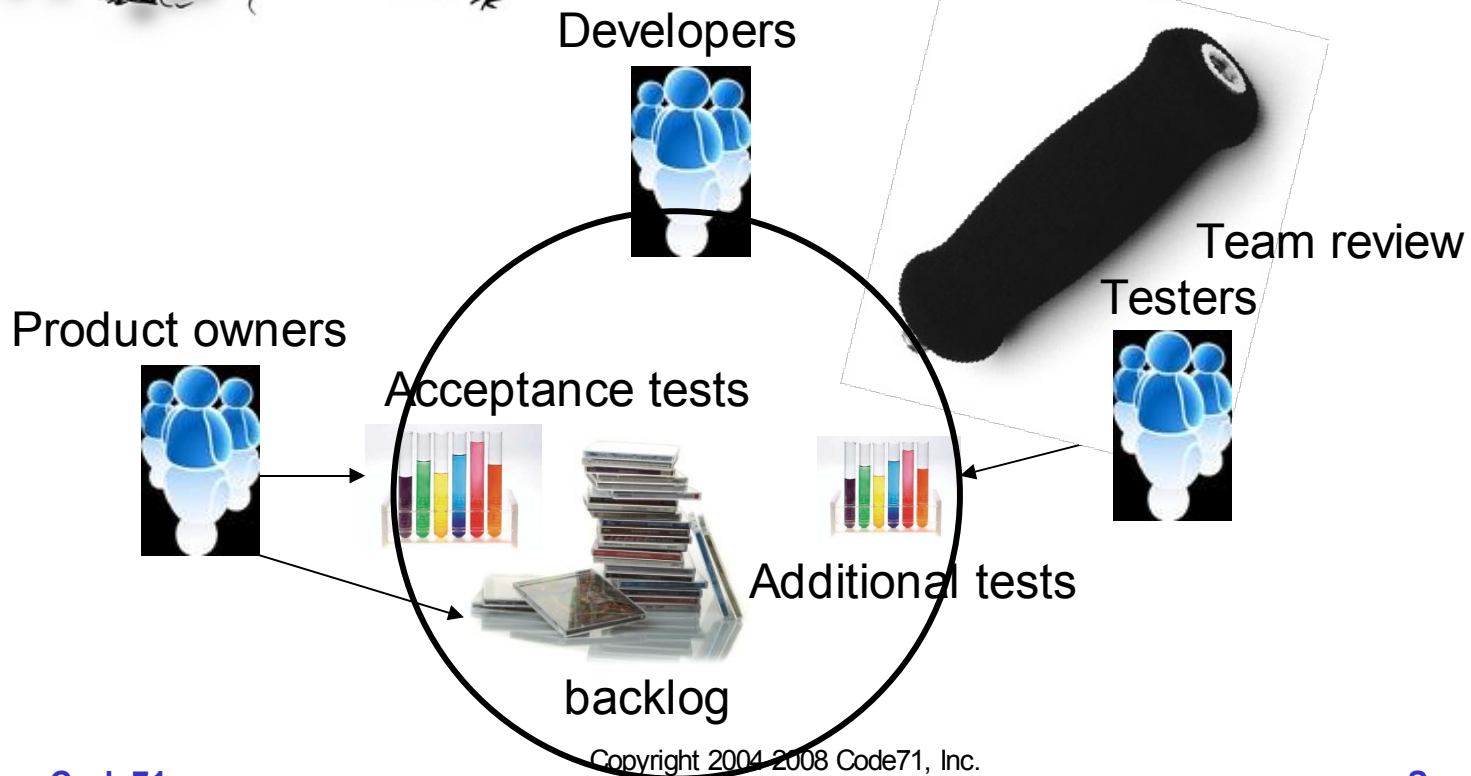
Spec	Spec review
Design	Design review
Code	Code review
Test	Test Coverage

Team Composition?

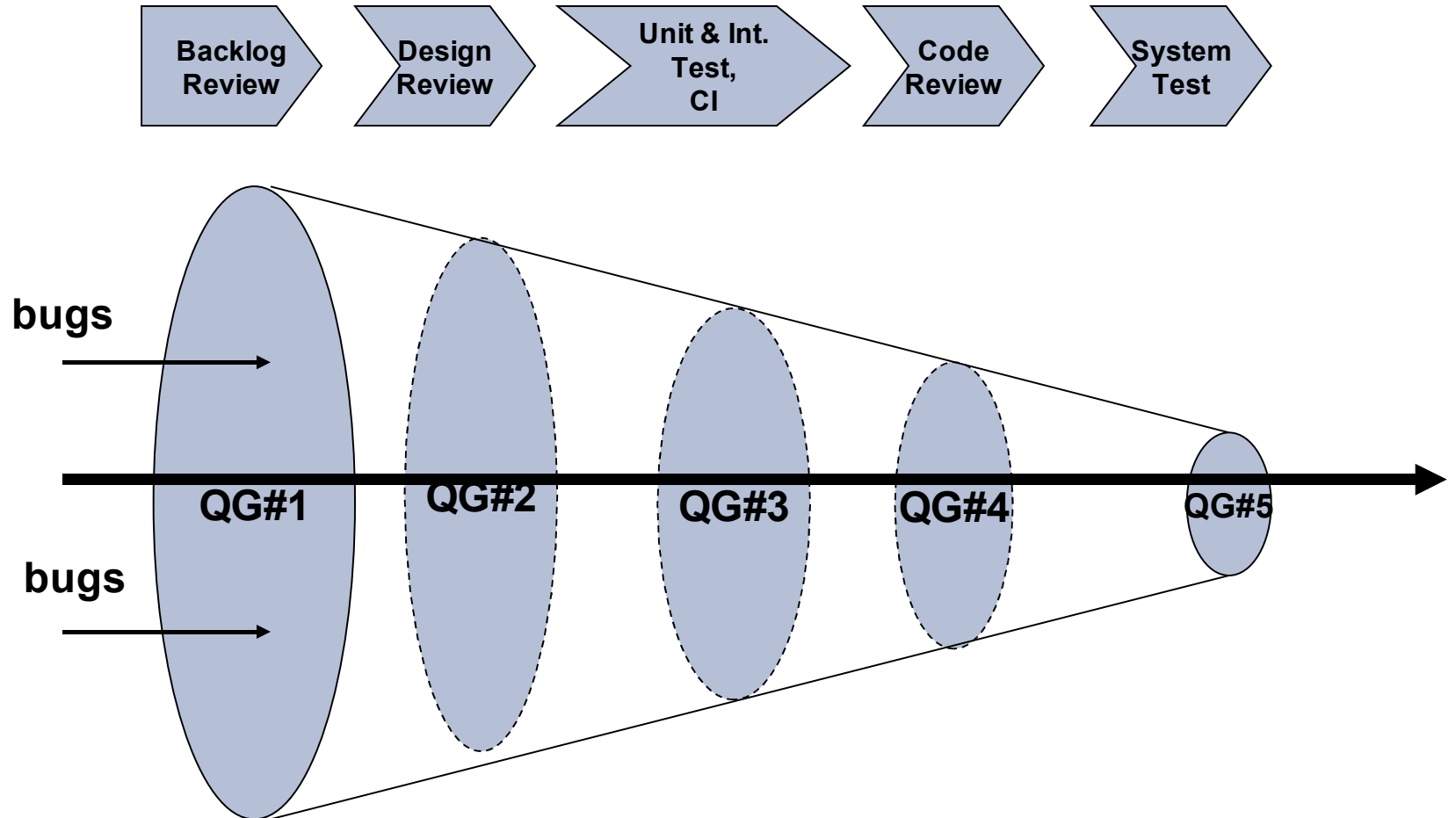


We will primarily focus on single-team model for our discussion

Team Dynamics



Quality Funnel



Agenda

Section **1** Introduction

Section **2** Holistic View of QA

Section **3** Individual Practices 

Section **4** A Case Study

Section **5** Recap

Section **6** Q&A

Test Coverage

Definition

“A measure of the proportion of a program exercised by a test suite, usually expressed as a percentage.”

Measure

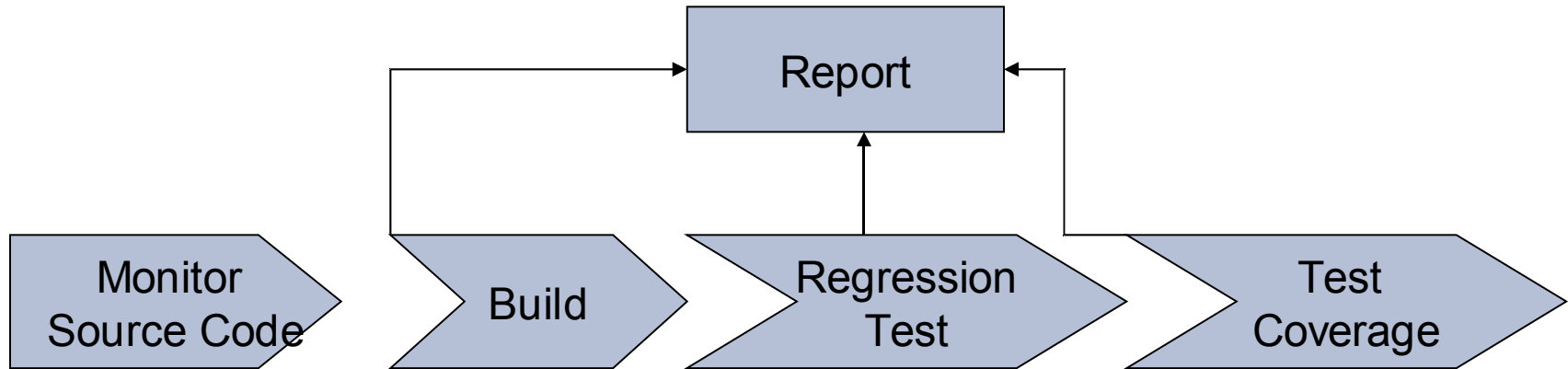
Usually expressed as a percentage

Types of coverage

- Function coverage
- Path coverage
- Statement coverage

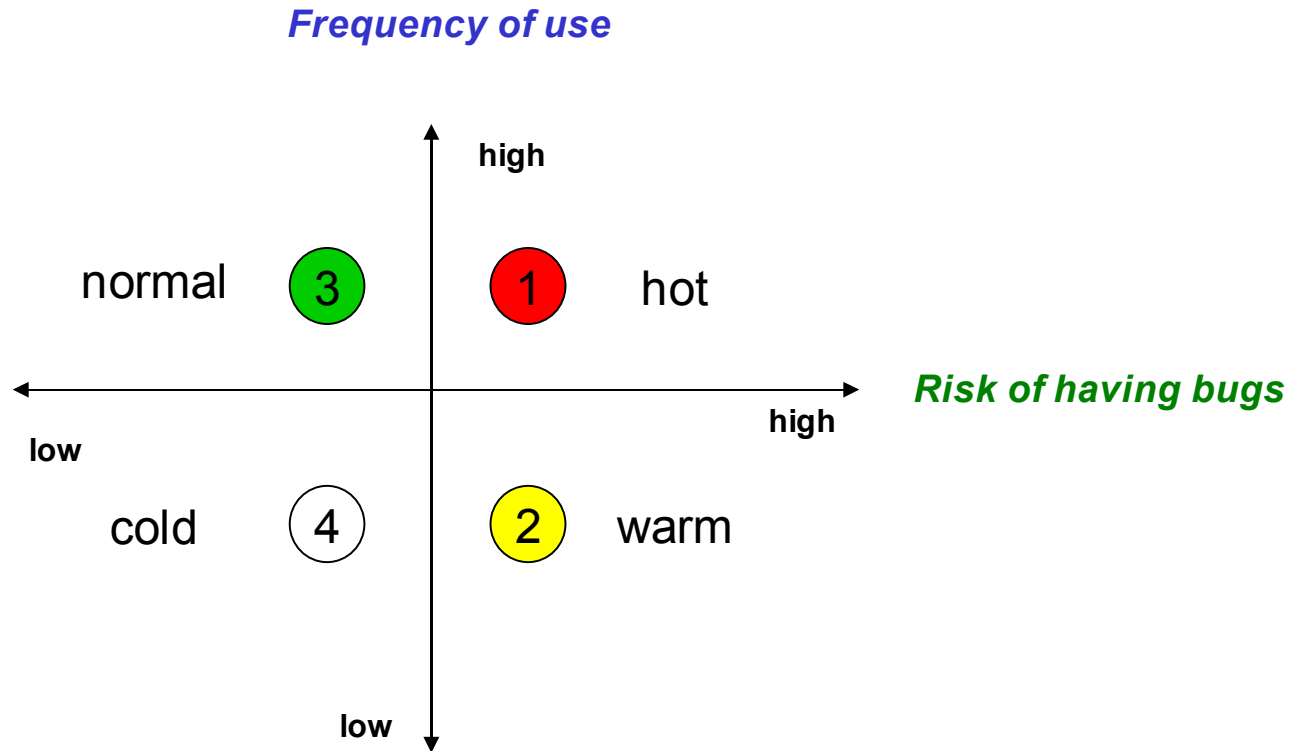
Tests coverage metrics can tell you what code is not tested

Continuous Integration as the Glue



Continuous Integration is Continuous QA

Test Case Prioritization



Inspect and Adapt through QA Lens

Five whys of root cause analysis

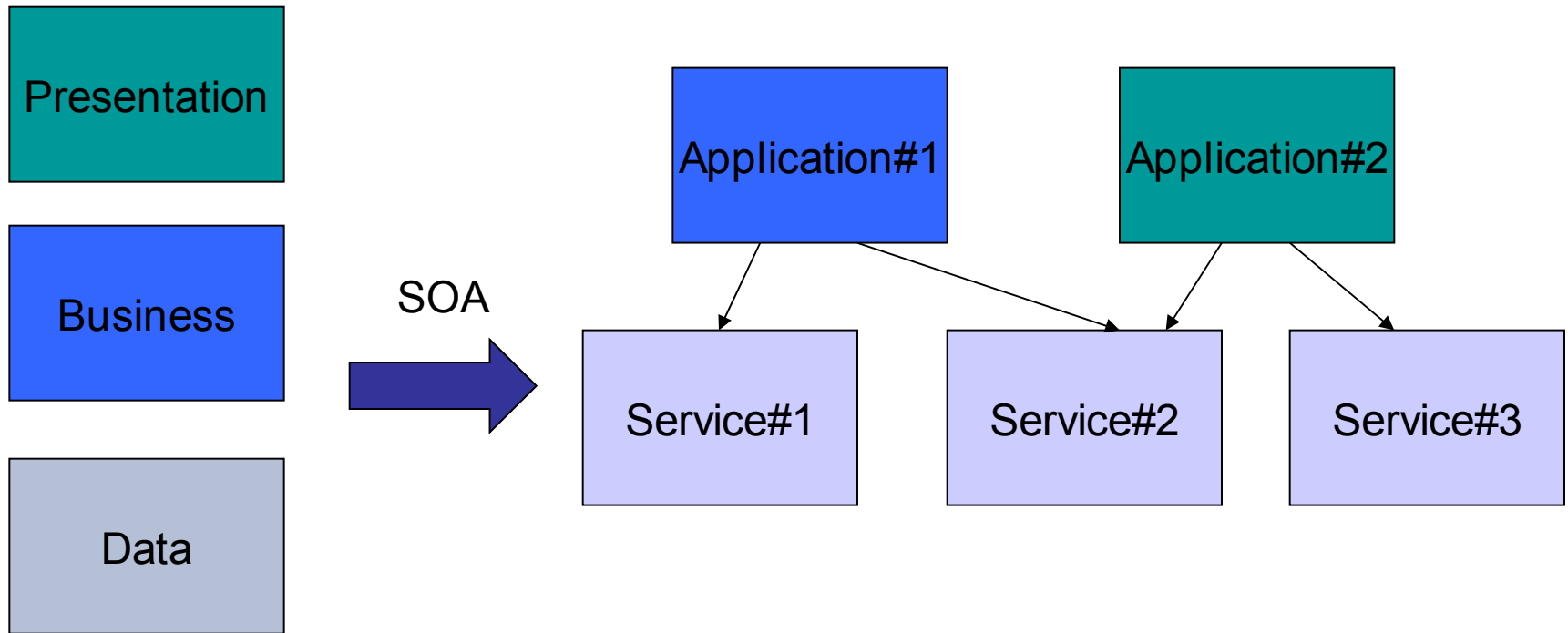
Prioritize bugs over stories

Log bugs found by testers

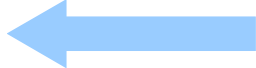
Quality Metrics

- **Defect Rate** Bug count per iteration
- **Defect Density** Bug count per module
 Bug count per function point

How to Scale?



Agenda

Section 1	Introduction
Section 2	Holistic View of QA
Section 3	Individual Practices
Section 4	A Case Study 
Section 5	Recap
Section 6	Q&A

A Case Study

Project

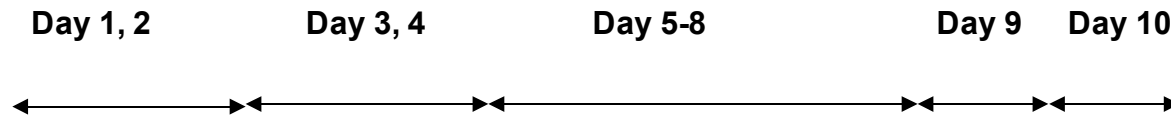
A large enterprise system that includes technologies like ASP.Net, BizTalk, Workflow, Scanning, SQL Server, Data Warehouse, and Mainframe

Team

2 product owners, 1 scrum master, 1 architect, 5 developers, 1 QA tester

Sprint

2 weeks



QA Process

Refine scope, acceptance test cases

Identify UI elements, test data
QA schedule

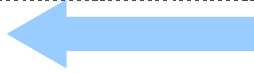
Write test scripts
test data

Final Test & Fix

Demo & Acceptance Test

Agenda

Section 1	Introduction
Section 2	Holistic View of QA
Section 3	Individual Practices
Section 4	A Case Study
Section 5	Recap
Section 6	Q&A



Recap

- “In-cycle” QA is critical to the success of a project
- System testing is not the only “quality gate,” it includes all types of testing and reviews
- Test automation is critical to “in-cycle” QA
- Target at least 90% test coverage
- “Continuous Integration” is “Continuous QA”
- Prioritize test cases based on risk and frequency of usage

Recap contd.

- All known bugs should be fixed first
- Right size story with well-thought out acceptance tests improves quality
- Include all “Quality Gates” as part of definition of “Done”
- Analyze each bug to understand where (Quality Gate) it should have been caught and improve (Inspect and adapt)
- QA is not a designated person’s responsibility, it is a team’s responsibility (self-organizing team)

Q&A

“QA is making sure **right** software works **right**”

“QA is not an act, but a habit”

Contact: srayhan@code71.com

Blog: <http://blog.syedrayhan.com>

Company: <http://www.code71.com>

Product: <http://www.scrumpad.com>