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

<table>
<thead>
<tr>
<th>Section</th>
<th>Introduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Holistic View of QA</td>
</tr>
<tr>
<td>2</td>
<td>Individual Practices</td>
</tr>
<tr>
<td>3</td>
<td>A Case Study</td>
</tr>
<tr>
<td>4</td>
<td>Recap</td>
</tr>
<tr>
<td>5</td>
<td>Q&amp;A</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Holistic View of QA</td>
</tr>
<tr>
<td>3</td>
<td>Individual Practices</td>
</tr>
<tr>
<td>4</td>
<td>A Case Study</td>
</tr>
<tr>
<td>5</td>
<td>Recap</td>
</tr>
<tr>
<td>6</td>
<td>Q&amp;A</td>
</tr>
</tbody>
</table>
The challenges?

- Is QA part of the development team?
- Can we fit QA in the same iteration as development?
- Who does QA?
- Does QA cost 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
  - Unit Testing
  - Integration Testing
  - Functional Testing
  - System Testing

- Black Box
  - Regression Testing
  - Acceptance Testing
  - Load Testing
  - Smoke Testing
## Who Performs What?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Testing</td>
<td>Developer</td>
<td>Coding</td>
<td>Always</td>
</tr>
<tr>
<td>Integration Testing</td>
<td>Developer</td>
<td>Coding</td>
<td>Always</td>
</tr>
<tr>
<td>System Testing</td>
<td>Tester</td>
<td>Test</td>
<td>Possible</td>
</tr>
<tr>
<td>Regression Testing</td>
<td>Developer/Tester</td>
<td>Build/Test</td>
<td>Possible</td>
</tr>
<tr>
<td>Acceptance Testing</td>
<td>Client/Users</td>
<td>Deployment/Delivery</td>
<td>Possible</td>
</tr>
<tr>
<td>Smoke Testing</td>
<td>Tester/Support Engineer</td>
<td>Deployment</td>
<td>Possible</td>
</tr>
<tr>
<td>Load Testing</td>
<td>Performance Engineer</td>
<td>Deployment</td>
<td>Always</td>
</tr>
</tbody>
</table>
## Right tools for right tests?

<table>
<thead>
<tr>
<th>Test</th>
<th>Tool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit Testing</td>
<td>NUnit, JUnit, Mock, DBUnit</td>
</tr>
<tr>
<td>Integration Testing</td>
<td>Unit test tools, HttpUnit, SoapUI, RESTClient</td>
</tr>
<tr>
<td>System Testing</td>
<td>Selenium, Fit, WET, Watir, WatiN</td>
</tr>
<tr>
<td>Regression Testing</td>
<td>Unit test tools, System test tools</td>
</tr>
<tr>
<td>Acceptance Testing</td>
<td>FIT, FitNesse</td>
</tr>
<tr>
<td>Smoke Testing</td>
<td>Regression test tools</td>
</tr>
<tr>
<td>Load Testing</td>
<td>JMeter, Httpperf</td>
</tr>
</tbody>
</table>
### 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
<table>
<thead>
<tr>
<th>Spec</th>
<th>Spec review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design</td>
<td>Design review</td>
</tr>
<tr>
<td>Code</td>
<td>Code review</td>
</tr>
<tr>
<td>Test</td>
<td>Test Coverage</td>
</tr>
</tbody>
</table>
Team Composition?

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

Acceptance tests

Product owners

Additional tests

Developers

Team review

Testers
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Holistic View of QA</td>
</tr>
<tr>
<td>3</td>
<td>Individual Practices</td>
</tr>
<tr>
<td>4</td>
<td>A Case Study</td>
</tr>
<tr>
<td>5</td>
<td>Recap</td>
</tr>
<tr>
<td>6</td>
<td>Q&amp;A</td>
</tr>
</tbody>
</table>
# 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

Monitor Source Code → Build → Regression Test → Test Coverage → Report

Continuous Integration is Continuous QA
Test Case Prioritization

Frequency of use

Risk of having bugs

- Normal (3)
- Cold (4)
- Hot (1)
- Warm (2)

Copyright 2004-2008 Code71, Inc.
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?

Presentation

Business

Data

SOA

Application#1

Service#1

Service#2

Service#3

Application#2
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
Day 1, 2
- Refine scope
- Acceptance test cases

Day 3, 4
- Identify UI elements
- Test data
- QA schedule

Day 5-8
- Write test scripts
- Test data

Day 9
- Final Test
- Fix

Day 10
- Demo & Acceptance Test
## Agenda

<table>
<thead>
<tr>
<th>Section</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Holistic View of QA</td>
</tr>
<tr>
<td>3</td>
<td>Individual Practices</td>
</tr>
<tr>
<td>4</td>
<td>A Case Study</td>
</tr>
<tr>
<td>5</td>
<td>Recap</td>
</tr>
<tr>
<td>6</td>
<td>Q&amp;A</td>
</tr>
</tbody>
</table>
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