Agile Testing 101
For Product Managers

Bay APLN

Lisa Crispin
With Material from Janet Gregory
Introductions: Experience, Goals
Introduction - Me

- Programmer from way back
- Tester since early ‘90s
- Agile from 2000
  - Always learning!
Takeaways

- What “agile” principles, values might help your team?
- Whole Team responsible for quality
- Avoiding the testing “squeeze”
- Your goals?
Some Vocabulary

- **Iteration**: (1 – 4 weeks) (“sprint” in Scrum)
- **Whole Team**: programmers, testers, BAs, DBAs, sys admins…
- **Done**: Coded and tested
- **Velocity**: Work completed in one iteration
- **Continuous Integration**: each time code checked in, regression tests run, instant feedback
- **TDD**: Test-Driven Development/Design
- **Specification by Example**: examples/tests guide development (aka ATDD)
Are We “Agile”?

- Agile = delivering value frequently, at a sustainable pace (Elisabeth Hendrickson)
- Incremental and Iterative
- Sustainable pace requires good practices
  - Continuous Integration
  - Short feedback loop
  - TDD, ATDD/Specification By Example
  - Good design, refactoring
Think About Your Teams

- Do you work at a sustainable pace?
- Does testing get squeezed to the end?
- Can you gauge the true progress of your projects at any given time?
Traditional vs. Agile SDLC

Phased or gated – eg. Waterfall

Requirements → Specifications → Code → Testing → Release

Agile: iterative and incremental
- Each story is expanded, coded and tested
- Possible release after each iteration

Copyright 2011: Lisa Crispin
Testing is Not a Phase

- Development = testing + coding
  - Write test, write code, test
  - At unit and acceptance test levels
- Automated tests
  - Designed for efficiency, maintainability
  - Free time for exploratory testing
- Team plans testing activities
  - All four “quadrants”
Agile Testing Quadrants

- Automated & Manual
  - Functional Tests
    - Examples
    - Story Tests
    - Prototypes
    - Simulations
  - Q2

- Business Facing
  - Exploratory Testing
    - Scenarios
    - Usability Testing
    - UAT (User Acceptance Testing)
    - Alpha / Beta
  - Q3

- Supporting the Team
  - Unit Tests
    - Component Tests
  - Q1

- Critique Product
  - Performance & Load Testing
    - Security Testing
    - “ility” Testing
  - Q4

- Technology Facing
  - Tools
Testing and Coding

Create a user story

Write Customer (Q2) Tests

Expand tests – Story Tests

Start thinking how to code

Automate Q2 Tests

Pair “Show Me”

TDD

Exploratory testing

Customer User Acceptance

Product owner
Product owner/ Tester
Tester
Tester/Programmer
Programmer
The Whole Team Approach

- What’s our commitment to quality?
- It has to mean something
- Everyone responsible to make sure testing activities are planned, completed
Team Organization - Traditional vs. Agile

Agile: roles overlap
So Do We Need Testers?

- Agile testers have unique viewpoint, mindset, attitude
- Eye on “big picture”, ripple effects
- Specific skills such as exploratory testing
- Ask the right questions
- Part of development team
- Learn the business domain
- Involved from beginning to end
What About Your Teams?

- Write down three things you now know about testing on agile teams
- Does your organization take a “whole team approach” to quality? If not, does it seem desirable? What could you do to encourage it?
Agile Isn’t a Mini-Waterfall

- Include testing activities in planning
- Focus on quality, not speed or velocity
- Scrum teams must under-commit
- Focus on completing one user story at a time
- Testing problems solved by whole team
Levels of Planning – Product, Release, Iteration

- Product Roadmap
- Release 1
- Release 2
- Release 3

- Project 1
  - It 0
  - It 1
  - It 2
  - Iteration 3 - 6
  - End Game
  - Story 1

- Project 2

- Release Plan

- Task A: 8 hours
- Task B: 5 hours
- Task C: 12 hours
- Task D: 8 hours

- Test Plan

- Story Tests

- Test Plan
Feedback Guides Development

- Team uses feedback to improve
- Testers are expert feedback providers
- Feedback lets team make course corrections
Managing Technical Debt

- Cut corners when needed
- Help business face reality
- But pay back your debt quickly and often
Shorten the Feedback Loop

- Continuous Integration
- Close collaboration
On Your Own Team/Project

“Agile” or not:

- What could you do to shorten the feedback cycle, and keep testing from being squeezed to the end?
- Write it on an index card
- Share your ideas with the person next to you

Involve testers throughout project
Agile Testing Resources

- lisacrispin.com
- janetgregory.ca
- gokjo.net
- exampler.com
- testobsessed.com
- agile-testing@yahoogroups.com
- testingreflections.com
- Local upcoming Agile Testing course at Agilistry Studios: http://agilistry.com/events/agile-testing-class/ Apr. 26-28
Proceeds to Charity

Beautiful Testing: Leading Professionals Reveal How They Improve Software

Edited by Tim Riley, Adam Goucher

Includes chapter by yours truly
Agile Samurai

Jonathan Rasmussen

Great intro to agile and agile testing
Bridging the Communication Gap

Specification By Example and Acceptance Testing

Gojko Adzic
Specification by Example

How successful teams deliver the right software

Gojko Adzic

Case studies from > 50 teams
Questions?