Software Testing in the Agile World

Krishna Gummuluri
Your Feedback

http://www.esri.com/events/devsummit/session-rater/

Offering ID - 309
Agenda

• Traditional SDLC
• What is Agile?
• Agile SDLC
• Agile requirements management
• How is Agile different?
• Agile testing
  - SDLC phases and testing activities
  - Testing strategy
• Implications for testers
• Test design
• Test automation
Traditional SDLC

- Requirements
- Architecture
- Design
- Construction
- Unit Test
- Function Test
- Integration Test
- Acceptance Tests

Decomposition and Definition
Integration and Validation
What is Agile?

“An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework, with just enough ceremony that produces high quality software in a cost effective and timely manner which meets the changing needs of its stakeholders.”
Agile SDLC

- Initial Funding
- Initiate the project
- Potential Projects

Defect reports

- Initial architecture vision

Work Items

- Highest Priority

Tasks

- Daily tasks

- Independent testing

Iteration Review and Retrospective

- Release
- Production

Feedback

Enhancements And Defects

- Remove From production

- Initial Requirements

Planning for current iteration

<table>
<thead>
<tr>
<th>Sprint -1</th>
<th>Sprint 0</th>
<th>Construction Sprints</th>
<th>Release</th>
<th>Prod.</th>
<th>Ret.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 2 3 4 n-1 n</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright 2010 Scott W. Ambler
Agile requirements management

- **High priority**: Modeled in greater detail
  - Each new work item is prioritized and added to the stack
  - Work item may be reprioritized at any time
  - Work item may be removed at any time

- **Low priority**: Modeled in lesser detail

**Work Items**
How agile is different

- Greater Collaboration
- Shorter work cycle and constant feedback
- Agilists embrace Change
- Greater flexibility
- Greater discipline
- Greater stakeholder accountability
- Greater range of skills
Benefits of Agile approach

• Greater ability to deliver required functionality

• Greater quality

• Better architecture and design

• Greater ROI
SDLC phases and testing activities

Start work on release N+1

**Sprint 0 (warm up)**
- Initiate the project

**Construction Sprints**
- Deliver working software that meets the changing needs of stakeholders
  - Development team testing
  - Independent team testing
  - Defect management

**Release Sprint (End Game)**
- Deploy Release N to production
  - End-of-life-cycle testing

**Production Support Release N (Maintenance)**
- Explore Reported issues
- Report defects

• Setup test environment
• Identify test team
Implications for test practitioners

- Learn coding
- Be prepared to work closely with developers
- Focus on value added activities
- Be flexible
  - Agile testing must be iterative
  - Cannot rely on having complete specification
Why Agile strategies work
Test design considerations for mobile

• Functionality
• Installation
  - Download
  - Install
  - Launch
• Mobile UI
• Interrupt
  - Voice calls
  - SMS
  - GPS
  - Low memory or battery
Test design considerations for mobile (contd.)

• Security
  - Authentication and authorization
  - Encryption - information being sent over the network

• Performance, Durability and Scalability
  - Three main components – app, network and server
  - CPU and memory usage
  - Power consumption

• Device compatibility

• Network
  - Consider where your end users are
  - Type of network service

• Submission criteria
Test design techniques

- Equivalence classes
- Boundary value analysis
- Positive and Negative testing
- Pairwise testing
  - http://aetgweb.argreenhouse.com/
- State transition
Fault model

Region Fault

Isolated Fault

Test Domain or Problem Space

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>Y1</td>
<td>Z1</td>
</tr>
<tr>
<td>X2</td>
<td>Y2</td>
<td>Z2</td>
</tr>
<tr>
<td>X3</td>
<td>Y3</td>
<td>Z3</td>
</tr>
</tbody>
</table>
Functional coverage and defect detection potential

Pair wise or OA based testing

One factor at a time testing
Test Design Demo
Test automation strategy

• Adopt a planned approach to developing test automation
• Increase the quality of test automation code
• Promote code reuse across teams
• Decrease maintenance cost
• Increase test automation coverage beyond functional testing (e.g. performance and stability testing or localization testing)
Selecting test areas for automation

- Risk Based
- Incidental execution of the test area
- How long it takes to run the test manually
- What is the cost of automating the tests
- How easy is the test cases to automate
- How many times is the test expected to run in a project
Test Automation Demo
Your Feedback

http://www.esri.com/events/devsummit/session-rater/

Offering ID - 309
esri
Understanding our world.
Thank You!
Sprint 0 testing activities

- Test environment setup
  - Test tools for developer testing
  - Test tools for independent parallel testing
  - Shared defect tracking system
  - Hardware, Virtualization and lab management
  - Continuous Integration tool

- Identifying test team
  - Development team with embedded testers
  - Independent test team
Construction phase testing

- **Development team testing**
  - Continuous Integration
    - build, test [and fix], deploy
  - Test Drive Development
    - Add test, run tests, change, run tests again
    - Test immediately after

- **Parallel Independent team testing**
  - Supplements development team testing
  - Complex environment and test scenarios
  - Support multiple development teams
  - End of life cycle testing
Parallel independent testing

- Internal release and change list
- Changed Stories
- Independent testing

Sprint n
- Internal release and change list
- Changed Stories

Sprint n+1
- Internal release and change list
- Changed Stories

Acceptance testing
Functional testing
Exploratory testing
Scenario testing
- Documentation
- Software
System testing
Usability testing
Construction Sprint testing activities

- Acceptance Tests (Large)
- Functional Tests (Medium)
- Dev. Tests (Small)
- Certification
- Architecture UX/code Tests
- Design
- Code
- Product Engineer
- Product Owner
- Developer
- PE
- UX
- API Service
- Automation library
- Epics PBls
- Source Code Validation
- Coding standards
- Product
- Owner
- Developer PE
- UX/PE
- UX/Code
- UX
Release Phase testing

- Independent test team
- Very short
- Reasons
  - Professional
  - Legal obligation
  - Stakeholders requirement
- Required to scale agile
“Embedded Tester” approach

- Testers embedded in agile teams
- Flexible to contribute in any way then can
- Wider range of skills with one or more specialties
- Shorter feedback cycles
- “Sufficient” in straightforward situations
- Focus on “confirmatory” testing

Issues
- Group think
- Lack of skills
- Lack of knowledge of skills needed
Test design – Acceptance Criteria categories

- Functional
- Integration/Touch points
- Accessibility
- L10N/I18N
- Performance
- Scalability
- Security
- Platform Awareness
- Help Documentation