Quality Assurance in an Agile Environment
Discussion Topic

• The Agile Movement
• Transition of QA practice and methods to Agile from Traditional
• Scrum and QA
• Recap
• Open Discussion…
What is Agile?

Definition of Agile

**Agile software development** is a group of software development methodologies that are based on similar principles. Agile methodologies generally promote a project management process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering best practices that allow for rapid delivery of high-quality software, and a business approach that aligns development with customer needs and company goals.

- Agile Does NOT stand for “Anything Goes Methodology”

- Agile is a lean method of software product development that is highly dependent on team interaction and rapid feedback
Characteristics of an Agile Team

• Execute Project deliverable in small increments with minimal planning, rather than long-term planning

• Development & QA is done short time frames (known as ‘Sprints') which typically last from one to four weeks

• Each iteration is worked on by a team through a full software development cycle, including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders

• Agile methodologies include a routine and formal daily face-to-face communication among team members
That is, while there is value in the items on the right, we value the items on the left more. Misconception that items on the right are not important! It’s the relative value being emphasized.

http://agilemanifesto.org/
The Software Development Evolution – Traditional to Agile

- Growing Realization that Requirements are never fully understood
- Fast paced development environments using new development Technologies and Tools
- Move towards business models like SaaS and software architecture like SOA
- Shortened product delivery cycles
- Increased involvement of business users in the development cycle
- Migration towards Agile product development models like SCRUM, RUP, XP
QA Framework using a Holistic Approach

Infrastructure
- Feature test lab
- Performance/Stress test lab

Tools
- Automation test tools
  - Functional: Selenium, QTP, WinRunner
  - Performance: LoadRunner, OpenSTA, Jmeter
- Defect Tracking: GEM, Bugzilla, TFS, FogBUGZ
- Version control: CVS, GEM
- Project Management: GEM, TFS, MS-Project

Technology
- Web technologies: .Net, Ajax, Java
- Legacy client-server systems
- Database: MS SQL Server, Oracle

Methodology
- Testing templates: SCRUM, Unified model
- Sprint aligned test case preparation
- Defect Classification
- Defect Root Cause Analysis
- Manual and automated testing
QA in Traditional Environment – The Pros and Cons

**PROS**
- Elaborate documentation available for preparing test cases
- Ample time for test planning
- Stable requirements means less re-work on test cases

**CONS**
- Any major requirement change/addition throws the QA schedule out-of-gear
- Any uneven distribution of effort between QA Planning and Execution phases cannot be realized earlier
- QA has to absorb the delays introduced by the BA/Development teams leading to compressed QA execution time and inadequate testing
QA’s Struggle with Agile

- **Scope Creep** – Requirement changes and updates are inherent to Agile methodology and QA’s biggest challenge

- Inadequate time to prepare test plans

- Minimal requirements documentation to prepare the test cases

- Highly compressed test execution cycles

- Minimal time for regression

- Change of role from being a Gatekeeper of Quality to being a Partner in Quality

- Often required to play a semi-developer role
Managing Quality in Agile Model

What moving to Agile does not mean for QA…

- Sprint feature specific testing only with less emphasis on System Integration testing
- Minimal Test documentation
- Ad-hoc Testing with ‘best possible’ coverage
- Being just an extension of development team
- Losing the user perspective of the product
- Freedom from ownership of Quality of the product!
Managing Quality in Agile Model

What moving to Agile **does** mean for QA...

- Participate ownership on quality
- A team of equals. Quality is Team’s responsibility. Role of QA is around defending the quality for the team by developing proper measures
- User or Product representative involved in all phases guiding Quality
- Agile team quality analyst is “adaptive” rather than “predictive”
Unified Scrum - An Agile Methodology
Genesis of "Unified Scrum"

Professionally managed short and flexible projects, presenting a significant challenge to the traditional process paradigm, scaling the given project to a wide and varying range of tasks. The need for a more flexible development process, which can handle large, cross-functional projects, led to the development of Unified Scrum.

Unified Scrum combines the benefits of both Scrum and RUP (Rational Unified Process) with the structure and discipline of PMI (Project Management Institute). RUP adds Unified Process phases, UP templates, and UP roles. PMI brings in PMI knowledge areas, program/portfolio management, enterprise resource management, and reporting and metrics.

Scrum
- Agile mindset
- Frequent feedback
- Use of Product Backlog
- Delegation of authority
- Smaller working groups

Unified Scrum

RUP
- Unified Process phases
- UP templates
- UP roles

PMI
- PMI knowledge areas
- Program/Portfolio management
- Enterprise resource management
- Reporting and metrics
Sprint Cycle

- Business case and Approval
- Product Roadmap
- Initial product backlog
- Initial release plan
- Architectural Direction

- Update product backlog
- Sprint Planning meeting
- Sprint retrospective
- Backlog review
- Daily scrum
- Release

Artifacts
- Feature backlog
- Product backlog
- Impediment list
- sprint backlog
- Product burndown
- Sprint backlog burndown

Scrum roles
- Product Manager
- Scrum master
- Scrum roles
- Users
- Team members
- Stakeholders

Inception
Release Cycles
Interchangeability of Roles in Unified Scrum

- Horizontal tracks define different roles in Unified Scrum team BUT
  - Roles can be interchangeable for a given resource
  - One resource does NOT have to tie themselves to One Role
  - Crossover of roles is encouraged for example, a BA in one sprint may be a QA in the next or sometimes, in the same sprint
  - Developers can perform BA roles often times during peak analysis times at the inception of the project
  - QAs often play the role in UAT and deployment and rollout.
- Key is to adapt to what best suits your organizational culture and skills
Typical QA in Unified Scrum Model

Week 1

- Test planning and Test Case development
- Test Data requirements definition
- Test Scripts documentation in collaboration with BAs and Developers
- Test cases reviewed by BAs
- Regression tests conducted on prior features
Typical QA in Unified Scrum Model

Week 2

- Additional test cases written
- Reviews from BAs incorporated
- Initial Builds are reviewed and defects reported
- Regression tests conducted on prior features
- Sprint test execution begins in full force at the end of Week 2
Typical QA in Unified Scrum Model

Week 3

- Sprint test execution conducted in full force
- Defects reported and daily defect review meeting conducted
- Fixes Retested
- End of Week
  - Quality Signoff Release
  - Release notes identifies open defects and approved features
  - Sprint Retrospective
QA in a typical SCRUM – the Issues

Less emphasis on test documentation can lead to half-baked cases

Test coverage is dictated by time available

Requirement changes - if not communicated effectively – results in QA executing invalid test cases or rendering test cases already executed useless

Often System Integration test cases get left out with focus only on the features delivered for the sprint

Low emphasis on documented status reporting and management reporting makes it difficult to track overall state of the Quality of the product
Sample QA Process Flow in Unified Scrum Method

Inception
- Review of Documents and Requirements
- Test Planning & Strategy

- Use of QA templates for documentation
- High Level Test Plan creation
- Meetings for Requirements clarification

Design & Development
- Test Case Creation

- Track Change Requests & update test cases
- Provide weekly execution coverage report
- Provide Bug trend report

QA
- Test Case Execution & Documentation (current sprint)
- Test Case Creation (next sprint)

- Test Case Execution & Documentation (current sprint)
- Test Case Creation (next sprint)

- Test Case Execution & Documentation (current sprint)
- Test Case Creation (next sprint)

Implementation & Deployment
- Full Regression Cycle
- Performance / Load Testing & Documentation
- Defect Testing & Documentation
- UAT (User Acceptance Testing)
- Defect RCA (Root Cause Analysis)
- QA Execution Summary Report

Product QA Certified
Evolution in Agile QA – Lean S/W Development

Eliminating Waste in the System!

• Don’t produce code that is not testable. Think QA before writing the code!
• QA is NOT a separate team with separate function at the end of the assembly line
• QA activities and people should be involved in the development of the product during the development
• Minimize “Handoffs” - avoid the traditional User-BA-Arch-Dev-QA model
• Write test first and use all the frameworks that will facilitate your test suite
• Automate testing, building, installations, anything that is routine, but do it smartly (read flexible)
• Re-factor code to be more streamlined, simpler the code- more effective it will be in production and to maintain
• Most importantly build quality in the process
RECAP - QA Mindset in an Agile Methodology

• Quality is responsibility of the ENTIRE Scrum Team

• Quality Gate does NOT only mean System Testing but the entire application

• Continuous Integration will Lead to Continuous QA

• Progressive inclusion of System Integration test cases from sprint to sprint

• QA resource plays a role of BA/QA removing any gaps in requirements communication

• Prioritize Test Cases by Risk and Frequency of Feature Usage

• Utilize the Sprint Retrospective to Identify “Where” in the scrum process did failure of quality occur (Root Cause Analysis)
Open Discussion