

Bridging the Gap Between Acceptance Criteria and Definition of Done

PNSQC 2013

Amith Pulla, Intel Corp.

Twitter: @pamith

Sowmya Purushotham

Clinicient Inc.

Introduction

Amith Pulla

- *QA Manger at Intel*
- *Involved in software testing strategies and processes*
- *Works on sales and marketing applications*
- *Worked with Scrum and Scaled Agile Framework*
- *Got PMP and CSTE in 2006, CSM in 2012*

Sowmya Purushotham

- *Software Quality Engineer at Clinicient Inc.*
- *Extensive background in Agile testing practices and tools*

Quality in Agile

- Quality is everyone's business (Owned by the team)
- Testing is way of life, everyone tests
- Development team is a cross-functional team that has *Developers, Test Analysts/Testers, Integrators, Deployers, Automators, UX Designers, Business Analysts etc.*
- Quality is defined & delivered by collaborative team effort
- Acceptance Criteria and DoD(s) have a huge influence on quality
- Team needs to work together to define Acceptance Criteria and DoD(s)

Scrum

Scrum Roles:

ScrumMaster, Product Owner and Development Team

Scrum Artifacts:

Product Backlog, Sprint Backlog, User Story, Tasks, Story Tests, **Acceptance Criteria, Definition of Done (DoD)**, Burndown Chart, Product Increment etc.

Scrum Events/Meetings:

Backlog Refinement, Release Planning, Sprint Planning, Daily Stand Up (DSU), Sprint Review, Retrospective(s)

Acceptance Criteria

- Requirements that have to be met for a story to be assessed as complete or accepted
- Story focused (unique for each story)
- Defines boundaries and constraints of the user story
- Discussed and defined as part of the product backlog grooming sessions
- Needs to be clearly written in a language that customers, product owners, and the development team can easily understand

Example

- As a User (Account Manager), using the letters feature, the User should be able to fax letters to a contact directly from the system instead of printing them out and manually sending fax to the contact
- **Questions and discussion for the product owner by the development team may include:**
 - *Will the User be able to fax multiple letters from the application to same contact?*
 - *Where does the User choose content to fax?*
 - *Can the User edit a letter for faxing?*
 - *Does the User need to manually enter fax number?*
- **The Acceptance Criteria could include:**
 - *User is able to see a 'Fax' button under Actions > Create a Letter menu*
 - *User should be able to choose a Client, Account and Contact*
 - *User can select a letter template and edit as needed*
 - *User needs to click 'Fax' button to fax a letter*
 - *User should see the fax chart notes pop up on clicking 'fax' button to choose a Fax cover page*

Definition of Done (DoD)

- Team's agreement stating that all the tasks or activities are completed for a Story, Sprint or a Release
- It is a set of common conditions across all Stories, Sprints and Releases that states that no more work is left to be done
- The DoD can also serve as a contract between a development team and its stakeholders
- The Scrum team(s) needs to collaboratively define DoDs at Story, Sprint and Release levels

- Some suggestions on writing good DoDs: (Rally Publications, 2013)
 - *Use conditions like, "all code checked in" or "unit test coverage > 80%"*
 - *Use "Code review completed" instead of "code review"*

DoD Example: For a Story

- Code Completed and Reviewed
 - Code is refactored (to support new functionality)
- Code Checked-In and Built without Error
- Unit Tests Written and Passing
- Release Configuration Documentation Completed (if Applicable)
- Acceptance Tests written and Passing
- Pass all Non-Functional Requirements if Applicable (Cross browser compatibility tier 1, 2)
- Product Owner Sign Off /Acceptance
- User Acceptance
- Manual regression scripts updated
- Test Automation Scripts Created and integrated
- Localization (truncation, wrapping, line height issues, string array issues, etc.)
- Analytics (Non-Functional Requirements) integrated and tested
- Story level device support (big browser, tablet, mobile device) tested

For Iteration:

- Unit Test Code Coverage >80%
- Passed Regression Testing
- Passed Performance Tests (Where Applicable)
- End user training team hand-off
- UAT (User Acceptance Testing)
- Production Support Knowledge Transfer done

For a Release:

- Regression tests completed in an integrated environment
- Performance or Load Tests completed
- Open defects reviewed by stakeholders and production support team
- Workarounds documented
- UAT and end user training completed

Technical Debt and Quality Risk Considerations

- Good Definition of Done(s) ensures stable Velocity and better Quality
- Consistent development and testing practices across all Stories
- Decide if a activity or condition belongs in the DoD for a Story, a Sprint or a Release
- As we move the conditions from a Story to Release level, it temporarily creates technical debt (adds risk)
- Try to keep as many conditions or activities as possible at the Story level and move them up to Sprint or Release level only if it's inefficient to do it at Story level
- Test automation scripting example: Iteration Vs. Story Definition of Done

Definition of Done for Enterprise-Class Agile Development

- Scrum of Scrums setup or Enterprise-Class Agile development method like SAFe (Scaled Agile Framework) or DAD (Disciplined Agile Delivery)
- Teams need to adhere to common enterprise architecture, release cadence, UX design and platform constraints
- Release DoD must be shared by all the Scrum teams in the Scrum of Scrums setup

SAFe Definition of Done Example

From <http://scaledagileframework.com/>

Story	Feature	Releasable Feature Set
Acceptance criteria met	A stories for the feature done	All features for the releasable set are done
Story acceptance tests written and passed (automated where practical)	Code deployed to QA and integration tested	End-to-end Integration and system testing done
Nonfunctional requirements met	Functional regression testing complete	Full regression testing done
Unit tests coded, passed and included in the Build Verification Tests (BVT)	No must-fix or Showstopper defects	Exploratory testing done
Cumulative unit tests passed	Nonfunctional requirements met	No must-fix defects
Code checked in and merged into mainline	Feature included in build definition and deployment process	End-to-end system, performance and load testing done
All integration conflicts resolved and BVT passed	Feature documentation complete	User, release, installation, and other documentation complete
Coding standards followed	Feature accepted by Product Owner or Product Manager	Localization and/or internationalization updated
Code peer reviewed		Feature set accepted by Product Management
No Showstopper or must-fix defects open		
Story accepted by the Product Owner		

Bridging the Gap

- **The Role of QA Leads and Test Engineers**
 - Help the team bridge the gap between Acceptance Criteria and Definition of Done
 - Apply their extensive background in product quality and testing
 - Help define the optimal Definition of Done for Story, Sprint and Release, balancing quality and risk

Focus Areas

- **Horizontal Capabilities**
 - BI (Business Intelligence) and Analytics
 - UI (User Interface) Design
 - Training and Help Files
- **Agile Engineering Practices**
 - Unit Testing and Code Reviews
 - Test Automation
 - Continuous Integration (CI)
 - Test-Driven Development (TDD)
 - Acceptance Test Driven Development (ATDD)
 - Automated Deployments

Focus Areas Cont....

- **Integration Testing**
 - Enterprise Ecosystems
 - User Authentication Solutions
 - Security
 - Disaster Recovery
- **Performance and Load Testing**
- **Mobile Device Compatibility (if applicable)**

Conclusion

- Acceptance Criteria and Definition of Done are two important artifacts of Agile development that will help teams deliver quality to the users or customers
- Invest time and collaborate to define and document
- Accessible and visible to the overall team including stakeholders, development team members and management
- Apply knowledge of product quality and experience in developing test strategies
- Help define and implement good DoD conditions at Story, Sprint and Release levels, bridging the gap between Acceptance criteria and Definition of Done

References

- Lisa Crispin and Janet Gregory. 2009. Agile Testing: A Practical Guide for Testers and Agile Teams
- James A. Whittaker, Jason Arbon and Jeff Carollo. 2012. How Google Tests Software
- Rally Publications. 2013. Agile Definition of Done Guide
 - https://www.rallydev.com/sites/default/files/defining_done_guide.pdf
- Dean Leffingwell. 2013. Scaled Agile Framework
 - <http://scaledagileframework.com/>
- Ken Schwaber and Jeff Sutherland. October 2011. The Scrum Guide
- CollabNet. Scrum Backlog Grooming.
 - <http://scrummethodology.com/scrum-backlog-grooming/>