Requirements Management in an Agile and Lean Environment

Systems and Software Technology Conference
Salt Lake City, UT
May 2011

Mr. George Haley
Director of Architecture, Design & Analysis
Northrop Grumman Corp
El Segundo, CA

Dr. Suzette S. Johnson
Agile Engineering
Northrop Grumman Corp
Columbia, MD
**Requirements Management in an Agile and Lean Environment**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

---

**Title and Subtitle**

Requirements Management in an Agile and Lean Environment

**Performing Organization**

Northrop Grumman Corp, 9160 Guilford Road, Columbia, MD, 21046-1803

**Dates Covered**

00-00-2011 to 00-00-2011

**Abstract**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

---

**Distribution/Availability Statement**

Approved for public release; distribution unlimited

---

**Supplementary Notes**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

---

**Security Classification**

- a. Report: Unclassified
- b. Abstract: Unclassified
- c. This Page: Unclassified

---

**Limitation of Abstract**

Same as Report (SAR)

---

**Number of Pages**

25
Discussion Outline

- Defining an Agile Environment
- Requirements, Use Cases, User Stories
- Levels Planning
- User Story Verification and Validation
- Summary
- References
An Agile Environment

- Adaptive, Responsive, Continuous Improvement, Evolving
- Improved transparency of progress
- End-to-end accountability and ownership
- Reduces time-to-deploy operational capability
- Ability to adapt to changing requirements and new technological advancements
Building Practice on Principles

- Eliminate Waste
- Build Quality In
- Create Knowledge
- Defer Commitment
- Deliver Fast
- Respect People
- Optimize the Whole

Adapted from: Implementing Lean Software Development: From Concept to Cash by Mary and Tom Poppendieck

Copyright 2011 Northrop Grumman Corporation
Deliver Value

Creating the Vision
- Customer Needs and Product Vision
- Captured Capabilities

Developing Understanding
- Requirements Use Cases
- System Level Validations
- User Stories and Acceptance Tests
  - Functional
  - Non-Functional

Requirements mapped to stories
Revisit architecture and design each release and iteration

This is the primary focus for our discussion today

Planning and Estimating the Work
- Product Roadmap
- Release Planning
- Iteration Planning
- Daily Plans and Commitment
Use cases

• “A **use case** is the specification of a set of actions performed by a system, which yields an observable result that is, typically, of value for one or more actors or other stakeholders of the system.... Use cases are a means for specifying required usages of a system. Typically, they are used to capture the requirements of a system, that is, what a system is supposed to do”\(^1\).

• Agile methods emerged with a focus on user stories. User stories are similar to use cases but are:
  – Typically more fine-grained & smaller;
  – Not intended to specify requirements;
  – More closely related to schedulable work.

\(^{1}\) OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2
Use Cases and User Stories

Why?
• System behavior is described operationally from users’ perspectives
  – Greatly reduces validation issues
• Drives verification to focus on operationally-relevant cases

How?
• In agile environments use cases are written “just-in-time” (for release planning) versus all up front
• Using use cases in an Agile environment. Ask:
  – How much do we need to write at this time?
  – When do we need to write more?
  – What is the fastest way to write/convey them?
  – Who benefits from more information or more detail?¹

Both approaches can coexist

Use Case/User Story Definition

- Written from the user perspective
- Captures value to the customer/user (not the developer)
- Emphasizes verbal *communication and collaboration*
- *The right size for estimating and planning* (User Stories only)
- Testable
- Demonstrable
Agile Systems Engineering Ontology

- "Action" is a.k.a. "Step"
- "Scenario" is a.k.a. "Flow"
Use Case to Scenario to User Story

1. Use Case

2. One Scenario within the use case

3. A user story is a segment of a scenario

As a [user/system] I want [what] so that [why]....
User Stories Convey Meaning
Example of traditional approach shortcomings

IEEE 830 Software Req. Spec

1. The product shall have a gas engine.
2. The product shall have four wheels.
   The product shall have a rubber tire mounted to each wheel
3. The product shall have a steering wheel.
4. The product shall have a steel body.

Reference: Mike Cohn, mountaingoatsoftware.com
Source: Adapted from The Inmates are Running the Asylum by Alan Cooper. (1999)
User Stories Convey Meaning

As a <lawn service provider> I want to mow lawns quickly and easily.

As a <lawn service provider> I want to sit comfortably while mowing lawns.

Reference: Mike Cohn, mountaingoatsoftware.com
Capabilities to User Stories

The system shall provide the capability for making hotel reservations.

- As a premiere member, I want to search for available discounted rooms.
- As a vacationer, I want to search for available rooms.
- As a vacationer, I want to save my selections.
What About Non-Functional Requirements?

As a vacationer and user of the hotel website, I want the system to be available 99.99% of the time...

As vacationer, I want web pages to download in <4 seconds...

As the hotel website owner, I want 10,000 concurrent users to be able to access the site at the same time with no impact to performance...

Stories for non-functional requirements

Describes system behavior or characteristics

Reference: Mike Cohn, mountaingoatsoftware.com
Advantages and Practices

• Why User Stories are preferred over traditional methods
  – Emphasizes verbal communications
  – Comprehensible by both customer and developer
  – The right size for planning
  – Works well for iterative development
  – Encourages deferring detail until you have the best understanding you are going to have about what you really need.
  – Helps the Team understand to whom they are delivering certain functionality
  – Helps the Team understand when they are “done”

• User Stories can be used with traditional requirements
  – Best to keep the requirements high level
  – A mapping from requirements to users stories needs to be maintained, especially if requirements are part of the contract.

Reference: Mike Cohn, MountainGoatSoftware.com
User Stories

- A story is either “done” or “not done”.
- As stories are completed, the status of the high-level capability is updated (verified, partial...).
- A set of tests is linked to each story and the high level requirement.
- Each story has a set of test objectives and automated tests.
- Stories are for communication and to better understand the work.

Focused on the Conversation
Release Plan, Iteration Plan, Daily Plan

Example: Hotel Website

**Capability 1: Make Room Reservations**

### Release Plan (Stories)

<table>
<thead>
<tr>
<th>Relative Value</th>
<th>User Stories</th>
<th>Points</th>
<th>Tests:</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>As a vacationer, I want to search room availability...</td>
<td>12</td>
<td>Test with search on 1 room</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Test with search on executive suite...</td>
</tr>
<tr>
<td>75</td>
<td>As a vacationer, I want to save my request...</td>
<td>8</td>
<td>Test Objective</td>
</tr>
<tr>
<td>70</td>
<td>As a vacationer, I want to pay with a credit card...</td>
<td>21</td>
<td>Test Objective</td>
</tr>
</tbody>
</table>

### Iteration Plan (Tasks)

- Design Review: 4 hours
- Install Baseline: 4 hours
- ICD Updates: 8 hours
- Acquire Test Data: 8 hours
- Code: 24 hours
- Develop Tests: 8 hours
- Run Tests: 8 hours

Yesterday I started on the interface....

Today I plan to...

The one thing standing in my way...

### The Daily Plan

**Product Backlog**

- The high-level requirements with stories mapped
- Prioritized by the product owner in terms of business value and risk
- Reprioritized at the start of each iteration
User Stories and Testing

Typically during release planning

Requirements

Team Develops Use Cases

One Use Case = Multiple User Stories

Functional and Non-functional

Team write user stories

Team writes test objectives/cases for each story

Design, Code, Integrate, Test

Individual user story tests verified

Test Case Testing

Regression Testing

Performance Testing

Test Pass?

Y

Verified and Validated

N

Enter defect/error report
The Iteration Demonstration and Acceptance

- Transparency and information sharing
- Team presents what it accomplished during the iteration
- Typically takes the form of a demo of new features or underlying architecture
- Time-boxed
- Whole team participates
- Feedback from stakeholders and users
- User Stories validated and accepted

User Story Validation
Requirements Mapping

- Requirement to story mapping

- Requirement to Story to Test to Verification
**Monitoring Progress: Product Burndown**

A project team’s burndown (team of teams)

- Based on story points planned
- Updated and reviewed each iteration
- As stories are accepted and tests passed, requirement progress is updated

A team’s product burndown

---

<table>
<thead>
<tr>
<th>Story Points</th>
<th>Iteration 0</th>
<th>Iteration 1</th>
<th>Iteration 2</th>
<th>Iteration 3</th>
<th>Iteration 4</th>
<th>Iteration 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story points baseline</td>
<td>35</td>
<td>30</td>
<td>25</td>
<td>20</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Burndown (Pts Remain)</td>
<td>0</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td>20</td>
<td>25</td>
</tr>
</tbody>
</table>

Copyright 2011 Northrop Grumman Corporation
Summary: Bringing It Together

Creating the Vision

Customer Needs and Product Vision

Captured Capabilities

Developing Understanding

Requirements mapped to stories
Revisit architecture and design each release and iteration

Requirements Use Cases
System Level Validations
User Stories and Acceptance Tests

As needed

- Architecture
- Sequence Diagrams
- Activity Flow Diagrams

Product Backlog

Planning and Estimating the Work

6-9 months
Product Roadmap

1-6 months
Release Planning

1-4 weeks
Iteration Planning

Daily
Daily Plans and Commitment

• Functional
• Non-Functional

- 6-9 months
- 1-6 months
- 1-4 weeks
- Daily
Final Notes

- Requirements and User Stories
  - High level requirements can be mapped to user stories
  - User stories convey understanding (user, need, why)
  - User stories create the Product Backlog

- Requirements Analysis and Design
  - Upfront requirement analysis is done during release planning for the capabilities being delivered in that release
  - Designs are developed/built upon each iteration
  - Design reviews for user stories are part of the story’s tasks and are done as needed

- Requirements Priorities and Changes
  - Requirements and user stories may be reprioritized
  - Contract modifications may be needed, but would not be done every iteration

- Requirements and Tests
  - High level requirements (end-to-end capabilities) have tests and each user story has tests.

- People
  - Systems engineers part of the team
  - Those responsible for performing the end-to-end capabilities testing should be part of team planning and regular collaboration
## References and Recommended Readings

### Agile Requirements and Collaboration

| Requirements by Collaboration | Ellen Gottesdiener, EBG Consulting |
| Collaboration Explained        | Jean Tabaka, Rally Software         |
| User Stories Applied           | Mike Cohn                           |

### Agile Development Practices

| Agile Software Requirements    | Dean Leffingwell                   |
| Agile Software Development with Scrum | Ken Schwaber and Mike Beedle | Lisa Crispin and Janet Gregory |
| Agile Estimating and Planning  | Mike Cohn                           |