Planning, Estimating, and Monitoring Progress in Agile Systems Development Environments

Systems & Software Technology Conference
Salt Lake City
April 2010

Dr. Suzette S. Johnson
Northrop Grumman Corporation
Agile Engineering and Leadership
Suzette.Johnson@ngc.com
1. REPORT DATE
APR 2010

2. REPORT TYPE

3. DATES COVERED
00-00-2010 to 00-00-2010

4. TITLE AND SUBTITLE
Planning, Estimating, and Monitoring Progress in Agile Systems Development Environments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northrop Grumman Corporation, 2980 Fairview Park Drive, Falls Church, VA, 22042

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:
   a. REPORT unclassified
   b. ABSTRACT unclassified
   c. THIS PAGE unclassified

17. LIMITATION OF ABSTRACT
   Same as Report (SAR)

18. NUMBER OF PAGES
   31

19a. NAME OF RESPONSIBLE PERSON

Standard Form 298 (Rev. 8-98)  Prescribed by ANSI Std Z39-18
Discussion Outline

- Agile Systems Development Introduction
- Scenario
  - Levels of Planning
  - Estimating
  - Monitoring Progress
- Summary
- Recommended Readings and References
Agile Systems Development

Promote rapid delivery of value to customers
What is Agile Systems Development?

- Agility is a set of demonstrated industry best practices for developing software systems

- Agility focuses on:
  - Principles and Values
  - Inspect and Adapt
  - Constant Commitment to Quality
  - Focus on the Value Stream
  - Team Ownership and Empowerment
Agile Principles

- Early and Continuous Delivery of Value
- A Working System is the Primary Measure of Progress
- Welcome Changing Requirements
- Deliver a Working System Frequently
- Business People and Developers Must Work Together Daily
- Motivated and Empowered Individuals
- Face-to-face Conversation
- Promote Sustainable Development
- Continuous Attention to Technical Excellence
- Simplicity
- The Best Architectures, Requirements and Designs Emerge From Self-Organizing Teams
- Regular Team Reflection on How to Become More Effective

http://agilemanifesto.org/
## Agile Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Backlog</td>
<td>Requirements/User Stories to be completed</td>
</tr>
<tr>
<td>Iteration</td>
<td>Fixed time-box in which development occurs</td>
</tr>
<tr>
<td>User Story</td>
<td>Similar to a requirement</td>
</tr>
<tr>
<td></td>
<td>“As a <em>user</em> I want to <em>action</em> so that <em>purpose</em>”</td>
</tr>
<tr>
<td>Velocity</td>
<td>The number of user story “points” delivered in a iteration</td>
</tr>
<tr>
<td>Capacity</td>
<td>The hours the development team is available to work within an iteration</td>
</tr>
<tr>
<td>Product Burn Down Chart</td>
<td>Progress for the release; Focuses on the remaining user story points</td>
</tr>
<tr>
<td>Iteration Burn Down Chart</td>
<td>A development team’s progress; Focuses on the remaining hours</td>
</tr>
<tr>
<td>Product Owner</td>
<td>Owns the product backlog, assigns priority to user stories</td>
</tr>
<tr>
<td>The Team</td>
<td>Cross functional team</td>
</tr>
</tbody>
</table>
High Level Agile Stages

Planning
- Vision
- Customer Needs

Estimating
- Product Roadmap
- Release Planning
- Iteration Planning

Monitoring
- Iteration Execution
- Iteration Demo and Retrospective
- Delivery
An example of an agile management framework

**Inspect and Adapt**

**Visibility and Transparency**

---


Image from: Mike Cohn, MountainGoatSoftware.com

© Copyright 2009 Northrop Grumman Corporation
Scenario

Planning, Estimating, and Monitoring
Project Scenario: RestEZ

- Duration: One Release
- Product: Development of a hotel website for RestEz
- Size of Project: ~10 persons/team; 10 teams = 100 people
- Release cycle is January 4 – March 31
- Six two-week iterations within the Release
- Planning for the Release is started prior to January 4th
  - Determining high level capabilities for the release
  - Story Writing workshop (Capability to User stories)
- Release Planning meeting is January 4th
- Iteration 1 detailed planning is in the afternoon of January 4th
- January 5th is the first day of development
Agile Metrics During the Release

• Planning
  - Velocity
  - Capacity
  - Total story points planned for the release
  - Length of iteration and release
  - Planned work hours

• Monitoring Progress
  - Product Burndown
  - Iteration Burndown
  - User Stories by State
  - Tracking Defects
  - Test Metrics
  - Actual worked hours
Planning and Estimating
Levels of Planning

Vision, Roadmap, Release, Iteration, Daily

Product Backlog (prioritized requirements by business value)

Product Roadmap

Vision

Customer Needs

Release 1

Goals & User Stories

Iteration 1

Iteration 2

Iteration 3

Iteration n...

~2 – 4 weeks fixed

Daily Stand Up

~2 - 6 months

© Copyright 2010 Northrop Grumman
Product Roadmap

Release 1
- Room reservations and payment
- User profiles for future visits

Release 2
- Conference offerings
- Online chat support

Release 3
- Special discounts
- Local information

Release 4
- Google maps
Release Planning: Use Case and User Stories

**Use Case**

- Written at the capability level and can be completed in one Release

**Capability/Scenario**

- Every Epic has validation criteria

**User Stories**

- Every story has acceptance criteria

**Tasks**

- Tasks written by the Team <=8 hours of work
- Based on a definition of “done”
- Owned by the Team

**Sum of all the test cases to fulfill the Release level test criteria**

**Unit Tests**

- Component testing
- Systems Testing

**Demo to stakeholders**
User Stories

What is a User Story?

- Functional stories
  - often based off a scenario of a use case
  - On large projects a user can be another system
- Non-functional stories
- Definition of Done
  - Design, Write tests, code, unit tests, documentation, etc.
- No credit for partial work – either done or not done

Estimation

- User Story Points
  - Bigness of the task
  - Influenced by (a) how hard it is (b) how much there is to do and (c) amount of uncertainty
  - Estimated by the team
  - Relative values
  - Total story points for the Release
## Release 1 Capability 1: Make Room Reservations

### Release Plan (User Stories)

<table>
<thead>
<tr>
<th>Business Value</th>
<th>User Stories</th>
<th>Test</th>
<th>Points</th>
</tr>
</thead>
</table>
| 80             | As a vacationer, I want to **search** room availability… | • Test with search on 1 room  
• Test with search on executive suite… | 12 |
| 75             | As a vacationer, I want to **save** my request… | Test  
Objective | 8 |
| 70             | As a vacationer, I want to **pay** with a credit card… | Test  
Objective | 16 |
|                | Story n…    |      |        |
**Release Plan, Iteration Plan, Daily Plan**

Example: Hotel Website

### Capability 1: Make Room Reservations

#### Release Plan (User Stories)

<table>
<thead>
<tr>
<th>Business Value</th>
<th>Test</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>As a vacationer, I want to search room availability…</td>
<td>Test with search on 1 room</td>
</tr>
<tr>
<td>75</td>
<td>As a vacationer, I want to save my request…</td>
<td>Test with search on executive suite…</td>
</tr>
<tr>
<td>70</td>
<td>As a vacationer, I want to pay with a credit card…</td>
<td>Test</td>
</tr>
</tbody>
</table>

#### Iteration Plan (Tasks)

<table>
<thead>
<tr>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Review</td>
</tr>
<tr>
<td>Install Baseline</td>
</tr>
<tr>
<td>ICD Updates</td>
</tr>
<tr>
<td>Acquire Test Data</td>
</tr>
<tr>
<td>Code</td>
</tr>
<tr>
<td>Develop Tests</td>
</tr>
<tr>
<td>Run Tests</td>
</tr>
</tbody>
</table>

#### The Daily Plan

Yesterday I started on the interface…

Today I plan to…

The one thing standing in my way…
Velocity (Based on history)

- Velocity is the amount of work a development team completes in an iteration (story points completed).
- Velocity is a range; Look for the high, the low, and the mean.

**Velocity for Team A**

- High: 45 story points
- Low: 30 story points
- Mean: 37 story points

**Project Velocity per Iteration**

- High: 155 story points
- Low: 120 story points
- Mean: 137 story points
Determining Team Capacity for an Iteration

- Capacity is the development team members’ available hours to work per iteration
- Revisited each iteration
- Compare planned hours versus actual hours
- Compare team capacity hours to the hours in the iteration

Example for a two-week iteration

<table>
<thead>
<tr>
<th>Team Member</th>
<th>Hours per day</th>
<th>Total hours per iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>Scott</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>Chris</td>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>Andy</td>
<td>7</td>
<td>70</td>
</tr>
<tr>
<td>Cindy</td>
<td>7</td>
<td>70</td>
</tr>
<tr>
<td>Mike</td>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>TEAM TOTAL</td>
<td>40</td>
<td>400</td>
</tr>
</tbody>
</table>
The Release Plan

• What is in the release plan?
  - Capabilities identified
  - User stories (functional and non-functional requirements)
    • Story points and prioritized
    • Project Teams average about 137 user story points per iteration; for a release with 6 iterations this is about 900 story points. The scope is 720-930 user story points of work.
  - Total number of user stories planned (125)
  - Total number user story points planned (~910 user story points)
  - Known or assumed velocity by development team and project team
  - Planned hours (WBS element)
Monitoring Progress
RestEZ: Product Burndown for the Release

- 910 points planned
- Project team or development team perspective
- Progress made and work remaining
- Based on story points

- Initiates discussion
- Reviewed every iteration
- Can be a “Burnup” chart
- Reports user stories completed against the plan for the release
A Development Team’s Iteration Burndown

User Story: As a vacationer, I want to search room availability...

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Tasks</th>
<th>Owner</th>
<th>Status</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Review</td>
<td>Scott</td>
<td>Completed</td>
<td></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Install baseline</td>
<td>Bill</td>
<td>Completed</td>
<td></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ICD updates</td>
<td>Scott</td>
<td>Completed</td>
<td></td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Acquire test data</td>
<td>Bill</td>
<td>Completed</td>
<td></td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Code</td>
<td>Scott</td>
<td>Completed</td>
<td></td>
<td>24</td>
<td>20</td>
<td>16</td>
<td>14</td>
<td>10</td>
<td>10</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Develop tests</td>
<td>Scott</td>
<td>Completed</td>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Run Tests</td>
<td>Scott</td>
<td>Completed</td>
<td></td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Emphasis on effort in hours
- Based on task hours for each story
- Hours remaining
- Updated daily
- Reviewed daily

The Iteration Burndown is for the Team
Tracking Stories by State

- How work is progressing during the iteration
- State examples: Opened, In Progress, Verified, Accepted, and Completed
- Based on total number of stories for the release
- May help identify bottlenecks
- *Verified* means all tests have passed on the system
- *Completed* means it is ready for release (i.e. potentially shippable)

Testing the User Stories

Compare the two graphs. Graph 1 indicates a relatively constant increase in “passed” user stories.

What about graph 2?

Metrics can help teams identify bottlenecks and other challenges on the project.

What is the challenge in graph 2?
Defect Metrics

- Defect metrics include:
  - Number of open defects at the end of each iteration
  - Number of defects found during systems integration
  - Number of defects found in production
  - The type of defects occurring
    - Useful in determining root cause

- May give indication as to what processes need to change

- Agile teams strive for “zero” defects
  - Part of the Definition of Done is “no critical defects against the story” or the story isn’t “done”
Summary

• Agile systems development environments:
  - Emphasize ongoing iterative development with completed, demonstrable functionality at the end of every iteration
  - Embrace practices that support changing requirements and mission needs
  - Engage in multiple levels of planning

• Valuable measures for estimating and monitoring progress:
  - Velocity (scope)
  - User stories planned versus user stories completed (progress)
  - User stories added to a release; not part of the original plan
  - Task hour burndown (daily progress)
  - Tests passed (what’s working)
  - Hours planned and hours worked
### Creating Adaptive Businesses

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Enterprise</td>
<td>Steven Haeckel</td>
</tr>
</tbody>
</table>

### Agile Development Practices

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile Project Management with Scrum</td>
<td>Ken Schwaber</td>
</tr>
<tr>
<td>Agile Software Development: Adaptive systems principles and best practices</td>
<td>Meso and Jain</td>
</tr>
<tr>
<td>Agile Software Development with Scrum</td>
<td>Ken Schwaber and Mike Beedle</td>
</tr>
<tr>
<td>Agile Testing</td>
<td>Lisa Crispin and Janet Gregory</td>
</tr>
<tr>
<td>Implementing Lean Software Development</td>
<td>Poppendeick</td>
</tr>
</tbody>
</table>

### Metrics

<table>
<thead>
<tr>
<th>Title</th>
<th>Author(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile EVM – Earned Value Management in Scrum Projects</td>
<td>Sulaiman, Barton, Blackburn</td>
</tr>
<tr>
<td>Agile Metrics (Agile 2009 Conference proceedings)</td>
<td>Dan Rawsthorne, Danube</td>
</tr>
</tbody>
</table>
The Need for Change

• Shortened product life cycle and technological advancements
• Shortened development times
• Decreased time-to-market
• Complexity of systems
• Large program failures
• Desire for improved transparency of progress
• Reduce risk