SCRUMP (Scrum + RUP) and CMMI:
The Story of a Harmonious Process and Product Deployment

March 22, 2011
Session ID 2565

Jennifer Turgeon
Sandia National Laboratories
Introduction

- Software Engineering Institute (SEI) Certified SCAMPI℠ Lead Appraiser
- SEI Authorized SCAMPI℠ B&C Team Lead
- Certified Scrum Product Owner
- Certified Scrum Master
- Member of the SCAMPI℠ B & C Development Team

℠SCAMPI is a service mark of Carnegie Mellon University.
©CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
Outline

• Case Study Description
• SCRUMP (Scrum + RUP) Implementation Plan
• Implementation Results
• Lessons Learned
• Implementation Strategy Recommendations

RUP = Rational Unified Process
Case Study Background Information

• **Goal:** Integrate three separate tools currently in production while providing usability enhancements
• **Currently in second development year**
  – **Year 1:** Prototype development using a modified RUP framework and an experimental modeling tool
  – **Year 2:** Decision to not use the experimental modeling tool and transition from RUP to Scrum
• **Corporate Information Technology (IT) application**
• **Resources:** ~6 FTEs, 20 belly buttons
• **End Users:** Potentially anyone in the company
Agile RUP Approach (Iterations, Plan Driven)

- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

Time

Phase Gate
Agile RUP Approach (Iterations, Plan Driven)

- FIX
- REQUIREMENTS
- ESTIMATE
- COST
- SCHEDULE

Predictive Vision Plan Driven

App 1
App 2
App 3
Agile Scrum Approach (Sprints, Value Driven)
Scrum Implementation Strategy
## Scrum and CMMI Relationships

<table>
<thead>
<tr>
<th>Maturity Level 2 Process Areas</th>
<th>Maturity Level 3 Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Requirements Management</td>
<td>• Requirements Development</td>
</tr>
<tr>
<td>• Project Planning</td>
<td>• Integrated Project Management</td>
</tr>
<tr>
<td>• Project Monitoring &amp; Control</td>
<td>• Risk Management</td>
</tr>
<tr>
<td>• Measurement and Analysis</td>
<td>• Technical Solution</td>
</tr>
<tr>
<td>• Supplier Agreement Management</td>
<td>• Product Integration</td>
</tr>
<tr>
<td>• Process and Product Quality Assurance</td>
<td>• Verification</td>
</tr>
<tr>
<td>• Configuration Management</td>
<td>• Validation</td>
</tr>
<tr>
<td>• Decision Analysis &amp; Resolution</td>
<td>• Decision Analysis &amp; Resolution</td>
</tr>
<tr>
<td>• Organizational Process Focus</td>
<td>• Organizational Process Definition</td>
</tr>
<tr>
<td>• Organizational Training</td>
<td>• Organizational Training</td>
</tr>
</tbody>
</table>

- **Strong relationship with a little work**
- **Some relationship with more work**
- **Little relationship**
## CMMI® Implementation Plans

<table>
<thead>
<tr>
<th>Strong CMMI® to Scrum Relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Requirements Management</strong></td>
</tr>
<tr>
<td>Document stories in the tool. Conduct backlog</td>
</tr>
<tr>
<td>grooming (SMEs, analysts), sprint planning,</td>
</tr>
<tr>
<td>peer reviews. Changes captured within the tool.</td>
</tr>
<tr>
<td><strong>Requirements Development</strong></td>
</tr>
<tr>
<td><strong>Project Planning</strong></td>
</tr>
<tr>
<td>Project vision, road map, and sprint planning,</td>
</tr>
<tr>
<td>demos, retrospectives, processes documented in</td>
</tr>
<tr>
<td>tool. RUP artifacts/templates (SW Dev Plan,</td>
</tr>
<tr>
<td>Dev Case, etc.) used.</td>
</tr>
<tr>
<td><strong>Project Monitoring &amp; Control</strong></td>
</tr>
<tr>
<td><strong>Integrated Project Management</strong></td>
</tr>
<tr>
<td><strong>Measurement &amp; Analysis</strong></td>
</tr>
<tr>
<td>Burndown charts, capacity spreadsheets, velocity,</td>
</tr>
<tr>
<td>etc. tracked and captured in the tool.</td>
</tr>
<tr>
<td><strong>Risk Management</strong></td>
</tr>
<tr>
<td>Retrospectives and sprint planning, RUP template</td>
</tr>
<tr>
<td>template for tracking risk/mitigation plans.</td>
</tr>
</tbody>
</table>
### Some CMMI® to Scrum Relationships

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Solutions</td>
<td>Alternative solutions and selection process conducted using RUP DAR process/templates and prototyping. Design evolving with plans to capture at first release.</td>
</tr>
<tr>
<td>Product Integration</td>
<td>Integration issues addressed in each sprint, and strategy is evolving. Team definition of “done” includes integration issues (procedures, establishing environment, evaluation, etc.). Process to be documented at first release.</td>
</tr>
<tr>
<td>Verification</td>
<td>All implemented stories have acceptance criteria that is tested by team tester. All work products go through the Product Owner. Peer reviews conducted on code and stories (backlog grooming). Determining how to document better.</td>
</tr>
<tr>
<td>Validation</td>
<td>All products and product components are validated through the Product Owner. Product VAL done in conjunction with VER activities. PO acceptance documented in the tool at story level.</td>
</tr>
</tbody>
</table>
## CMMI® Implementation Plans

### Little CMMI® to Scrum Relationships

<table>
<thead>
<tr>
<th>Area</th>
<th>Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supplier Agreement Management</td>
<td>N/A for this project.</td>
</tr>
<tr>
<td>Process &amp; Product Quality Assurance</td>
<td>Use RUP PPQA checklists. Issue is determining when to conduct PPQAs with no phase gates in Scrum. Also, consolidating current checklists is still under review.</td>
</tr>
<tr>
<td>Configuration Management</td>
<td>Use RUP CM plan template. Team tool is the CM repository/environment. CM audits with no phase gates is difficult.</td>
</tr>
<tr>
<td>Decision Analysis &amp; Resolution</td>
<td>Use the RUP DAR process and templates.</td>
</tr>
</tbody>
</table>
# CMMI® Implementation Plans

## Little CMMI® to Scrum Relationships

<table>
<thead>
<tr>
<th>Organizational Process Focus</th>
<th>RUP processes are used here and Scrum activities are actively being shared with the RUP team to see how Scrum can be better defined for the organization.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Process Definition</td>
<td>RUP processes/templates are the standard and are then tailored to Scrum environment. Primary issues have been identifying useful measures to contribute to the organization and tailoring.</td>
</tr>
<tr>
<td>Organizational Training</td>
<td>Use the RUP OT process and templates.</td>
</tr>
</tbody>
</table>
Lessons Learned – The Rewards

• Old RUP roles fit well into Scrum roles
  – Nobody lost their job
• Scrum significantly improved communication and trust between the software development team and the customer(s)
• Scrum empowered team members
  – Team members took on some project management roles
  – No waiting for approvals before moving forward
• Scrum is an excellent method for handling rapidly changing requirements
Lessons Learned – The Challenges

• Useful organizational measure contributions takes work
  – Comparison across projects is difficult
  – Inflated estimates slow productivity
  – Limited success masking

• Determining when to document is a dilemma
  – Design and product integration attributes

• Quality issues can be difficult to address
  – When to test
  – Lacking software development practices (pair programming, coding standards, etc.)
Implementation Strategies

1. Upper management sponsorship will greatly decrease the time necessary to adopt a new software development method.
   - Process adoption is strongly encouraged
   - Conflicting project demands that impact milestones can be reprioritized
Implementation Strategies

2. A full time, experienced Certified Scrum Master (CSM) is ideal.

- ½ time works if there are others on the team who will assist with CSM responsibilities
- A Scrum coach could greatly improve the adoption rate
- An experienced Scrum Master will be able to manage team dynamics through stressful times
Implementation Strategies

3. Don’t wait for the perfect plan, dive in and learn from your experiences as you go.

- But…make sure you have established code development and testing principles
- Use and modify existing organizational processes
- As new or modified processes emerge, test them through a few sprints before documenting
- Be willing to take risks; be willing to fail
Implementation Strategies

4. Determine your planning cycles

- **product vision**: yearly by the product owner
- **product roadmap**: bi-yearly by the product owner
- **release plan**: quarterly by the product owner and teams
- **iteration plan**: bi-weekly by the teams
- **daily plan**: daily by the individuals
5. If Scrum practices need to meet higher rigor requirements (CMMI, ISO, etc.), tools are highly recommended.

- Tools help to document team activities and decisions as you go (wikis, Scrum lifecycle management tools, etc.)
- Tools assist with configuration management and continuous integration builds
- Tools provide a central collaboration area for all team members
Implementation Strategies

6. Backlog grooming is critical.

- Vision & Roadmap
- Product Backlog: Prioritized Features desired by Customer
- Select User Stories (Product Owner, Analysts)
- Conversations and Confirmation (Product Owner, SMEs, Analysts, Developers, Testers)
- Each Iteration
  - Implementation (Developers)
  - Update Acceptance Tests (Testers, SMEs, Analysts, Developers)
- Sprint Demo
- 2 sprints ahead
- Backlog Grooming (Product Owner, SMEs, Analysts)
7. Resist the urge to go with longer sprints at first.

- Eliminates the need to cancel a sprint because “something” isn’t going as planned
- Provides more opportunities for trying new ideas
- Allows the development team to be more responsive to changing customer needs
### Implementation Strategies

8. Establish a realistic team capacity.

<table>
<thead>
<tr>
<th>Person</th>
<th>Commitment</th>
<th>Possible Hours Available</th>
<th>Hours Away</th>
<th>Base Hours</th>
<th>Buffer</th>
<th>Planned Hours Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scooby Doo</td>
<td>90%</td>
<td>64.8</td>
<td>0</td>
<td>64.8</td>
<td>35%</td>
<td>22.68</td>
</tr>
<tr>
<td>Scrappy Doo</td>
<td>90%</td>
<td>64.8</td>
<td>0</td>
<td>64.8</td>
<td>50%</td>
<td>32.4</td>
</tr>
<tr>
<td>Scooby Dee</td>
<td>90%</td>
<td>64.8</td>
<td>8</td>
<td>57.6</td>
<td>50%</td>
<td>28.8</td>
</tr>
<tr>
<td>Howdy Doo</td>
<td>70%</td>
<td>50.4</td>
<td>0</td>
<td>50.4</td>
<td>50%</td>
<td>25.2</td>
</tr>
<tr>
<td>Scooby Dum</td>
<td>30%</td>
<td>21.6</td>
<td>24</td>
<td>14.4</td>
<td>50%</td>
<td>7.2</td>
</tr>
<tr>
<td>Yabba Doo</td>
<td>30%</td>
<td>21.6</td>
<td>0</td>
<td>21.6</td>
<td>50%</td>
<td>10.8</td>
</tr>
<tr>
<td>Dooby Doo</td>
<td>30%</td>
<td>21.6</td>
<td>0</td>
<td>21.6</td>
<td>50%</td>
<td>10.8</td>
</tr>
<tr>
<td>Momsy Doo</td>
<td>30%</td>
<td>21.6</td>
<td>0</td>
<td>21.6</td>
<td>50%</td>
<td>0</td>
</tr>
</tbody>
</table>

**Total**

- 4.6
- 331.2
- 32
- 316.8
- **Capacity** 138

- Hours in Sprint: 72
- Capacity Factor: 50%
Implementation Strategies

9. Identify a strong product owner.

- Final decisions need to be made by one, not a group
- Product owners should be trained
- Full-time product owners are ideal
- Ensure that the product owner is an integral part of the team
10. Value your retrospectives.

- Discuss process improvements
- Create action items out identified opportunities for improvements with a goal to close the action items by the next retrospective
Contact Information

Jennifer Turgeon
Sandia National Laboratories
jturgeo@sandia.gov