Agile Development Overview

Presented by Jennifer Bleen, PMP
Project Services Practice of Cardinal Solutions Group, Inc.
Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others to do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more
Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face to face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity - the art of maximizing the amount of work not done - is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Agile Methods

- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Extreme Programming (XP)
- Feature Driven Development (FDD)
- Lean Software Development (LD)
- Scrum
Extreme Programming
An Engineering Framework

- A collection of software engineering techniques
- Useful as a repository of techniques
- Useful as a case study on what works

Copyright 2001. J. Donovan Wells
12 Practices of XP

- Fine scale feedback
  - Pair Programming
  - Release Planning
  - Unit Tests
- Continuous process
  - Continuous Integration
  - Refactoring
  - Small Releases
  - Acceptance Tests
- Shared understanding
  - Coding Standards
  - Collective Ownership
  - Simple Design
  - Metaphor
- Programmer welfare
  - Sustained Pace
Why XP Works

• The 12 Practices are mutually supporting – strengths offset weaknesses.
• Extreme Programming embraces, rather than fights, change
  The cost of change remains level, and does not rise exponentially.
• Extreme Programming fosters partnership and collaboration between development and the business stakeholders.
• Extreme Programming focuses on delivering business value quickly
  Automated tests, refactoring and pair programming ensure the delivery of robust, “production ready” software.
• Extreme Programming focuses on simplicity and on building for today rather than tomorrow
SCRUM: A Management Method

- Scrum is geared toward management
- Emphasizes team emergent behavior
- Tracks product in 2 levels (product and sprint)
- Emphasizes people and requires good people

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
Key components of Scrum

• Process Roles
  – The Product Owner
  – The Scrum Master
  – The Team

• Work Product
  – The Product Backlog
  – The Sprint Backlog
  – The Sprint Burndown
  – Product Increment

• Activities
  – The Sprint Planning
  – The Sprint
  – The Daily Scrum
  – The Sprint Review
  – The Sprint Retrospective
Step 1- Create the Backlog

Roles Involved:
• Product Owner
• Scrum Master
• Team

Artifacts Involved:
• Product Backlog

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
### Sample Product Backlog

#### Use excel

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
<th>Est</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very High</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Finish database versioning</td>
<td>16</td>
<td>KH</td>
</tr>
<tr>
<td>2</td>
<td>Get rid of unneeded shared Java in database</td>
<td>8</td>
<td>KH</td>
</tr>
<tr>
<td>3</td>
<td>Add licensing</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Concurrent user licensing</td>
<td>16</td>
<td>TG</td>
</tr>
<tr>
<td>5</td>
<td>Demo / Eval licensing</td>
<td>16</td>
<td>TG</td>
</tr>
<tr>
<td>Analysis Manager</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Analysis Manager</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>High</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Enforce unique names</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>In main application</td>
<td>24</td>
<td>KH</td>
</tr>
<tr>
<td>9</td>
<td>In import</td>
<td>24</td>
<td>AM</td>
</tr>
<tr>
<td>10</td>
<td>Admin Program</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>Delete users</td>
<td>4</td>
<td>JM</td>
</tr>
<tr>
<td>12</td>
<td>Analysis Manager</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>When items are removed from an analysis, they should show up again in the pick list in lower 1/2 of the analysis tab</td>
<td>8</td>
<td>TG</td>
</tr>
<tr>
<td>14</td>
<td>Query</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>15</td>
<td>Support for wildcards when searching</td>
<td>16</td>
<td>T&amp;A</td>
</tr>
<tr>
<td>16</td>
<td>Sorting of number attributes to handle negative numbers</td>
<td>16</td>
<td>T&amp;A</td>
</tr>
<tr>
<td>17</td>
<td>Horizontal scrolling</td>
<td>12</td>
<td>T&amp;A</td>
</tr>
<tr>
<td>18</td>
<td>Population Genetics</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>19</td>
<td>Frequency Manager</td>
<td>400</td>
<td>T&amp;M</td>
</tr>
<tr>
<td>20</td>
<td>Query Tool</td>
<td>400</td>
<td>T&amp;M</td>
</tr>
<tr>
<td>Medium</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Additional Editors (which ones)</td>
<td>240</td>
<td>T&amp;M</td>
</tr>
<tr>
<td>22</td>
<td>Study Variable Manager</td>
<td>240</td>
<td>T&amp;M</td>
</tr>
<tr>
<td>23</td>
<td>Haplotypes</td>
<td>320</td>
<td>T&amp;M</td>
</tr>
<tr>
<td>24</td>
<td>Add icons for v1.1 or 2.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>25</td>
<td>Pedigree Manager</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>26</td>
<td>Validate Desired kindred</td>
<td>4</td>
<td>KH</td>
</tr>
</tbody>
</table>

- Number each item for easier reference
- Name of each task
- Product owner prioritizes work
- Team estimates task

© 2007 Cardinal Solutions Group
Step 2: Expand the backlog into tasks in a sprint planning meeting

Roles Involved:
- Product Owner
- Scrum Master
- Team

Artifacts Involved:
- Product Backlog
- Sprint Backlog

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
The Sprint Planning Meeting

- Product Owner describes highest priority features to the Team.
- Team decides what the can commit to delivering in the Sprint.
- Both create the Sprint Backlog.
### Sample Sprint Backlog Example

<table>
<thead>
<tr>
<th>Who</th>
<th>Description</th>
<th>Days Left in Sprint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td></td>
<td><strong>Total Estimated Hours:</strong></td>
<td>554</td>
</tr>
<tr>
<td><strong>User's Guide</strong></td>
<td></td>
<td>-</td>
</tr>
<tr>
<td>SM</td>
<td>Start on Study Variable chapter first draft</td>
<td>16</td>
</tr>
<tr>
<td>SM</td>
<td>Import chapter first draft</td>
<td>40</td>
</tr>
<tr>
<td>SM</td>
<td>Export chapter first draft</td>
<td>24</td>
</tr>
<tr>
<td><strong>Misc. Small Bugs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JM</td>
<td>Fix connection leak</td>
<td>-</td>
</tr>
<tr>
<td>JM</td>
<td>Delete queues</td>
<td>8</td>
</tr>
<tr>
<td>TG</td>
<td>Fix tear-off messaging bug</td>
<td>8</td>
</tr>
<tr>
<td>JM</td>
<td>View pedigree for kindred column in a result set</td>
<td>2</td>
</tr>
<tr>
<td>AM</td>
<td>Derived kindred validation</td>
<td>8</td>
</tr>
<tr>
<td><strong>Environment</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TG</td>
<td>Install CVS</td>
<td>16</td>
</tr>
<tr>
<td>TBD</td>
<td>Move code into CVS</td>
<td>40</td>
</tr>
<tr>
<td>TBD</td>
<td>Move to JDK 1.4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KH</td>
<td>Killing Oracle sessions</td>
<td>8</td>
</tr>
<tr>
<td>KH</td>
<td>Finish 2.206 database patch</td>
<td>8</td>
</tr>
<tr>
<td>KH</td>
<td>Make a 2.207 database patch</td>
<td>8</td>
</tr>
<tr>
<td>KH</td>
<td>Figure out why 481 indexes are created</td>
<td>4</td>
</tr>
</tbody>
</table>

**Heading from product backlog**

**What is the task**

**Who is responsible**

**All tasks MUST be assigned eventually**

Tasks are detailed from product backlog.

How must work per week for the task.
Step 3: Execute a sprint

Roles Involved:
- Scrum Master
- Team

Artifacts Involved:
- Product Backlog
- Sprint Burndown Chart

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
The Sprint

- Strictly 30 consecutive calendar days:
  - Date takes priority over features
- Sprint Backlog and Sprint Burndown Charts show progress
- The Product Owner cannot change priorities
- Sprint results in a demonstrable result
Step 3A: Hold Daily Meetings

Roles Involved:
- Scrum Master
- Team

Artifacts Involved:
- Product Backlog
- Sprint Burn down Chart

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
Scrum Meetings

- Meeting Format
  - Daily
  - 15-minutes
  - Stand-up (keep it short)
  - Not for problem solving – to id problems

- Three questions:
  1. What did you do yesterday
  2. What will you do today?
  3. What obstacles are in your way?

- Management and team are invited
  - Help avoid other unnecessary meetings

- Only team can talk
The Sprint Burndown Chart

Updated as a result of the Scrum meeting

Similar to Earned Value
Step 4 - Demonstrate Functionality

Roles Involved:
• Product Owner
• Scrum Master
• Team

Artifacts Involved:
• Product Backlog
• Product Increment

Source: Adapted from Agile Software Development with Scrum by Ken Schwaber and Mike Beedle.
Common Findings and Best Practices

• It is a good idea to combine parts of Methodologies
  – No out of the box SDLC will work immediately
  – Always tailor
  – Pick pieces that work best
  – Evaluate the process explicitly after each iteration

• Methodology will not work without acceptance by team
  – Methodology will not work unless aligned with people, project, and corporate behavior
Timebox the Work

• Create 1 – 6 week timeboxes
• Negotiate amount of work in timebox with everyone involved
• Don’t interrupt the timebox
• Drop functionality rather than slip
  – Avoid burning people out though
• Estimate for next iteration is counterintuitive
  – If you slipped, make time shorter and do less
  – Eventually you should land on about 4 weeks
Adopt Good Requirements Practices

• Focus on functionality rather than traditional requirements
  – Use use cases to describe functionality
• Tie tests to requirements
• Prioritize and cost requirements
  – Manipulating requirements should have a cost
• Requirements development is a skill
  – Train and practice for good requirements
  – Make examples of requirements
Grow Software

- Start with core and add richness to features
- Don’t polish features to completion, grow them
- Organize around the architecture

This is not bolt on software
Create Automated Integration

- Create a build area for nightly check-in
- Create a smoke test for nightly build
- Automate reporting on build and test

Must create a culture for this technique
Prefer Visual Models

- Faster to create, change, absorb visual models
- Use UML (it’s a standard)
- Apply visual modeling to requirements and business modeling, not just design
Resources

- Agility and Discipline Made Easy (Kroll and MacIissac)
- Agile Software Development with SCRUM (Schwaber)
- Balancing Agility and Discipline (Boehm and Turner)
- Agile Software Development: Evaluating Methods for your Organization (Koch)
- Agile Alliance visit http://www.AgileAlliance.org
- Columbus XP User Group visit http://www.cardinalsolutions.com/XP/
Contact Information

Jennifer Bleen, PMP

Principle Consultant, Cardinal Solutions
jbleen@cardinalsolutions.com

President, XP User Group
xpusergroup@yahoo.com