Agile Software Development

CSCI 479: Computer Science Design Project
Fall 2016
Xiannong Meng

What is Agile Development?

**Agile software development** is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.


Agile Manifesto

- Four core values of Agile
  - Individuals and their interactions
  - Over processes and tools
  - Working software
  - Over comprehensive documentation
  - Customer collaboration
  - Over contract negotiation
  - Responding to change
  - Over following a plan


Manifesto in 2014 by Jama Software (1)

- **IS THE MANIFESTO AGILE OR IS AGILE THE MANIFESTO?**
- Rethink the Manifesto because
  - The world has changed;
  - Software is everywhere;
  - Complexity has increased;
  - Projects still fail.


manifesto in 2014 by jama software (2)

<table>
<thead>
<tr>
<th>2001 Manifesto</th>
<th>2014 Manifesto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individuals and interactions</td>
<td>Over Combined with Processes and tools</td>
</tr>
<tr>
<td>Working software</td>
<td>Over Balanced with Comprehensive documentation</td>
</tr>
<tr>
<td>Customer collaboration</td>
<td>Over Combined with Contract negotiation</td>
</tr>
<tr>
<td>Responding to change</td>
<td>Over Combined with Following a plan</td>
</tr>
</tbody>
</table>


12 Principles of Agile Development 1 of 4

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
12 Principles of Agile Development
2 of 4

- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

3 of 4

- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.

4 of 4

- Simplicity—the art of maximizing the amount of work not done—is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Important Elements of Agile

- Close collaboration between develop team and business experts
- Face-to-face communication
- Frequent delivery of new deployable business value
  - Incremental, iterative development (IID)
- Tight, self-organizing teams
- Craft code in a professional way

Many Forms of Agile

- Agile is a collective term, many different practices fall into this collection.
  - See the map from Agile Alliance

VersionOne 2015 Survey Result

---Top Five Benefits of Agile

- 87% indicate “Ability to manage changing priorities”
- 85% indicate “Increased team productivity”
- 84% indicate “Improved project visibility”
- 81% indicate “Increased team morale and motivation”
- 81% indicate “Better delivery predictability”
### VersionOne 2015 Survey Result

**Agile Methodology Used**

<table>
<thead>
<tr>
<th>Name</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scrum</td>
<td>58%</td>
</tr>
<tr>
<td>Scrum/XP Hybrid</td>
<td>10%</td>
</tr>
<tr>
<td>Custom Hybrid (multiple methods)</td>
<td>8%</td>
</tr>
<tr>
<td>Scrumban</td>
<td>7%</td>
</tr>
<tr>
<td>Kanban</td>
<td>5%</td>
</tr>
<tr>
<td>Iterative Development</td>
<td>3%</td>
</tr>
<tr>
<td>Other</td>
<td>3%</td>
</tr>
<tr>
<td>Lean Development</td>
<td>2%</td>
</tr>
<tr>
<td>I Don’t Know</td>
<td>2%</td>
</tr>
<tr>
<td>Agile Modeling</td>
<td>1%</td>
</tr>
<tr>
<td>Feature-Driven Development</td>
<td>1%</td>
</tr>
<tr>
<td>DISM / Atern</td>
<td>1%</td>
</tr>
<tr>
<td>XP</td>
<td>1%</td>
</tr>
<tr>
<td>Agile Unified Process</td>
<td>&lt; 1%</td>
</tr>
</tbody>
</table>

### Some Further Statistics

- Based on a survey of 3,880 completed responses
- 95% respondents say their company practice some form of Agile, up from 85% in 2012

### Agile Techniques Used (1)

<table>
<thead>
<tr>
<th>Feature</th>
<th>2015</th>
<th>2014</th>
<th>2012</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily Standup</td>
<td>83%</td>
<td>80%</td>
<td>85%</td>
<td>78%</td>
</tr>
<tr>
<td>Prioritized Backlogs</td>
<td>82%</td>
<td>79%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short Iterations</td>
<td>79%</td>
<td>79%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrospective</td>
<td>74%</td>
<td>69%</td>
<td>72%</td>
<td>64%</td>
</tr>
<tr>
<td>Iteration Planning</td>
<td>69%</td>
<td>71%</td>
<td>75%</td>
<td>74%</td>
</tr>
<tr>
<td>Unit Testing</td>
<td>63%</td>
<td>65%</td>
<td>74%</td>
<td>70%</td>
</tr>
<tr>
<td>Releasing Planning</td>
<td>63%</td>
<td>65%</td>
<td>69%</td>
<td>65%</td>
</tr>
<tr>
<td>Burndown/Team based estimate</td>
<td>59%</td>
<td>56%</td>
<td>67%</td>
<td>67%</td>
</tr>
<tr>
<td>Iteration Reviews</td>
<td>54%</td>
<td>53%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous Integration</td>
<td>50%</td>
<td>50%</td>
<td>56%</td>
<td>54%</td>
</tr>
<tr>
<td>Dedicated Product Owner</td>
<td>49%</td>
<td>48%</td>
<td>51%</td>
<td></td>
</tr>
</tbody>
</table>

### Agile Features Used (2)

<table>
<thead>
<tr>
<th>Feature</th>
<th>2015</th>
<th>2014</th>
<th>2012</th>
<th>2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Team (Integrated Dev &amp; Testing)</td>
<td>45%</td>
<td>46%</td>
<td>49%</td>
<td></td>
</tr>
<tr>
<td>Coding Standards</td>
<td>44%</td>
<td>43%</td>
<td>57%</td>
<td>51%</td>
</tr>
<tr>
<td>Kanban</td>
<td>39%</td>
<td>31%</td>
<td>32%</td>
<td>24%</td>
</tr>
<tr>
<td>Open Work Area</td>
<td>38%</td>
<td>38%</td>
<td>43%</td>
<td>38%</td>
</tr>
<tr>
<td>Refactoring</td>
<td>37%</td>
<td>36%</td>
<td>48%</td>
<td></td>
</tr>
<tr>
<td>Test Driven Development (TDD)</td>
<td>33%</td>
<td>34%</td>
<td>40%</td>
<td>38%</td>
</tr>
<tr>
<td>Story mapping</td>
<td>30%</td>
<td>29%</td>
<td>38%</td>
<td>35%</td>
</tr>
<tr>
<td>Collective Code Ownership</td>
<td>25%</td>
<td>27%</td>
<td>32%</td>
<td>28%</td>
</tr>
<tr>
<td>Pair Programming</td>
<td>24%</td>
<td>21%</td>
<td>30%</td>
<td></td>
</tr>
<tr>
<td>Behavior Driven Development (BDD)</td>
<td>10%</td>
<td>9%</td>
<td>10%</td>
<td>9%</td>
</tr>
</tbody>
</table>

### What Do These Terms Mean?

- Check out the guidelines from [https://www.agilealliance.org/agile101/guide-to-agile/agile-glossary/](https://www.agilealliance.org/agile101/guide-to-agile/agile-glossary/)

### A Brief Overview of Scrum

- *Scrum is an iterative and incremental agile software development framework for managing software projects and product or application development.*
**Scrum Process**

![Scrum Process Diagram](http://sandervanpeer.wordpress.com/2012/01/05/why-should-we-choose-scrum-for-our-projects/)

**Sprint**

- A *sprint* is the basic unit of development in Scrum.
  - The sprint is a “timeboxed” effort, i.e., a specific duration.
  - The duration usually is a week to a month.
  - Each sprint is proceeded by a planning meeting (e.g., daily standing), followed by a review meeting.

**Sprint Meetings**

- **Planning meeting**: where the tasks for the sprint are identified and an estimated commitment for the sprint goal is made.
- **Review meeting**: where the progress is reviewed and goals for the next sprint are identified.
- **Retrospective meeting**: where team members reflect their experience and draw lessons from it.

**Daily Scrum Format**

- Each day during the sprint, a project team communication meeting occurs. This is called a *daily scrum*, or *daily standup*. This meeting has specific guidelines:
  - All members of the development team come prepared with the updates for the meeting.
  - The meeting starts precisely on time even if some development team members are missing.
  - The meeting should happen at the same location and same time every day.
  - The meeting length is set (timeboxed) to 15 minutes.
  - All are welcome, but normally only the core roles speak.

**Daily Scrum Content**

- During the meeting, each team member answers three questions:
  - What have you done since yesterday?
  - What are you planning to do today?
  - Any impediments/stumbling blocks? Any impediment/stumbling block identified in this meeting is documented by the Scrum Master and worked towards resolution outside of this meeting. No detailed discussions shall happen in this meeting.

**End of Cycle**

- At the end of a sprint, two meetings are held:
  - *Sprint Review Meeting*, four things to do:
    - Review the work that was completed and the planned work that was not completed;
    - Present the completed work to the stakeholders (a.k.a., "the demo");
    - Incomplete work cannot be demonstrated;
    - Four-hour time limit.
End of Cycle

• At the Sprint Retrospective:
  – All team members reflect on the past sprint;
  – Make continuous process improvements;
  – Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint?
  – Three-hour time limit;
  – This meeting is facilitated by the Scrum Master.

Scrum Artifacts

• Product Backlog
  – an ordered list of “requirements” that is maintained for a product.

• Sprint Backlog
  – the list of work the Development Team must address during the next sprint.

• Increments
  – the sum of all the Product Backlog Items completed during a sprint and all previous sprints

• Scrum task board
  – publicly displayed chart showing remaining work in the sprint backlog.

Sample Scrum Task Board

What Are We Going To Do?

• Have team meetings with the customers every week including the instructor;
• Discuss and record what’s accomplished and what’s to do for next week at each meeting;
• Have milestones (sprints) twice during the semester and one at the end of the semester.

References

  – http://stateofagile.versionone.com/
• Agile Manifesto
  – http://agilemanifesto.org/
• Agile manifesto in 2014
• Agile Principles
  – http://agilemanifesto.org/principles.html
• Scrum