CS314 Software Engineering  
Project Management 

---

Dave Matthews

---

Software process movements

• Predictive – 1970
  – Waterfall
• Iterative – 1980s, 1990s
  – Spiral, RAD, RUP
• Adaptive (Agile) – late 1990s
  – XP, Scrum, Crystal, FDD, Lean, DSDM, Kanban, …
• Enterprise Adaptive (Lean & Agile) – late 2000s
  – SAFe, Nexus, …
Plan Driven versus Value Driven

<table>
<thead>
<tr>
<th></th>
<th>Fixed</th>
<th>Flexible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>Plan Driven</td>
<td>Resources</td>
</tr>
<tr>
<td>Resources</td>
<td>Value Driven</td>
<td>Time</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others 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.

http://agilemanifesto.org/
Agile Principles

1. Customer satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Working software is delivered frequently (weeks rather than months).
4. Close, daily cooperation between business people and developers.
5. Projects are build around motivated individuals, who should be trusted.
6. Face-to-face conversation is the best form of communication (co-location).
7. Working software is the principal measure of progress.
8. Sustainable development, able to maintain a constant pace.
9. Continuous attention to technical excellence and good design.
10. Simplicity – the art of maximizing the amount of work not done – is essential.
11. Best architectures, requirements, and designs emerge from self-organizing teams.
12. Team regularly reflects on how to become more effective, and adjusts accordingly.

https://en.wikipedia.org/wiki/Agile_software_development

Scrum

Scrum (n): A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.
Definition of Done

A shared understanding of what it means for work to be complete.

Team

- Product Owner
- Development Team
- Scrum Master
Artifacts

• Product Backlog
• Sprint Backlog
• Product Increment

Events

• Sprint Planning
• Daily Scrum
• Sprint Review
• Sprint Retrospective
## Sprint Timeline

<table>
<thead>
<tr>
<th>Week</th>
<th>Sun</th>
<th>Mon</th>
<th>Tue</th>
<th>Wed</th>
<th>Thu</th>
<th>Fri</th>
<th>Sat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Demo, Product Backlog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Sprint Planning</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td>Review, Retro., Increment</td>
<td>Demo, Product Backlog</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Sprint Planning

From Product Backlog to Sprint Backlog
Sprint Goal

- An objective set during the Sprint Planning meeting.
- Selected Backlog Items (Epics) deliver one coherent function.
- Provides guidance to development team on why it is building the increment.
- Gives flexibility regarding the functionality implemented for the sprint.
- Causes Development Team to work together rather than on separate initiatives.

Sprint Planning

- Product Owner can help to clarify Product Backlog items (epics) and make trade-offs.
- Development Team may renegotiate selected Product Backlog items (epics) if it has too much or too little work.
- By the end of Sprint Planning, Development Team should be able to explain how it intends to work as a self-organizing team to accomplish the Sprint Goal and create the anticipated Increment.
Sprint Planning

• Time-boxed event
  – maximum 8 hours for 4 week sprint (160 work hours)
  – maximum 1.5 hours for our 3 week sprint (24 work hours)
• Plan answers two questions:
  – What can be delivered in the increment this sprint? Epics
  – How will the work to deliver the increment be achieved? Tasks

What can be delivered?

• Product Owner discusses the objective that the sprint should achieve and backlog items (Epics) that would achieve it.
• Scrum Team collaborates on understanding the work.
• Development Team selects Product Backlog items (Epics) to forecast the functionality to be developed during the sprint.
• Scrum Team crafts a Sprint Goal
How will the work get done?

• Development Team decides how it will build the functionality into a “Done” product increment.
• Enough tasks are planned for Development to forecast what it believes it can do in the upcoming sprint.
• Tasks planned for the first days/week of the sprint is decomposed by the end of the meeting, often to units of one day/hours or less.
• Development Team self organizes to undertake the work.

Task Estimation

• Estimates are guesses
  – The larger the project, the less accurate the estimate
  – The farther from completion, the less accurate the estimate
• When paired with historical data they become more useful
  – burndown charts - are you on track to achieve your goal
  – velocity charts - how fast are you progressing
• Story points versus time estimates
  – story points are unit-less, sized in relation to other tasks
  – independent of person doing the work
Planning Poker

- Discuss a backlog item (epic / task)
- Estimate the size using Fibonacci cards
  - relative to each other / hours to do
- If large range, discuss further to understand why
- Done when estimates are similar (not identical)
- Breakdown tasks with estimates > 3

---

Story Board

<table>
<thead>
<tr>
<th>New Issues</th>
<th>Icebox</th>
<th>Backlog</th>
<th>In Progress</th>
<th>Done/Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>• add new epics / tasks</td>
<td>• epics / tasks that will not be completed in this sprint</td>
<td>• add estimate</td>
<td>• add assignee</td>
<td>• add release</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• add milestone</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• add labels</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Create milestones with start and end dates
- Create labels for tasks
Burndown Charts

- Monitor your team's progress during the sprint.
- Based on your initial sprint planning and refinement during the sprint.

Daily Scrum
Sprint Review
Sprint Retrospective