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

---

**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
cs314 - 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>Daily Scrum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sprint Check (33%)</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td>Daily Scrum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sprint Check (66%)</td>
<td>Daily Scrum</td>
<td>Review, Retro., Increment</td>
<td>Demo, Product Backlog</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

cs314 - Sprint Check (%)

- Periodic review of team Burndown Charts for the current Sprint milestone
- Burndown Chart should include:
  - Epics with the sprint milestone, but no estimate
  - Tasks with the sprint milestone and estimate
- Backlog should only contain Epics/Tasks with the Sprint milestone
- Icebox should contain Epics/Tasks with no milestone
Sprint Planning

What we can do?

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 • add milestone • add labels</td>
<td>• add assignee</td>
<td>• add release</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.
cs314 - plan sprint in Zenhub

- Add a Sprint Milestone
- Add Epics to Sprint Backlog, associate with milestone
- Add tasks for each Epic with initial Estimate, associate with Epic and Milestone.
- Prioritize the Epics and Tasks in ZenHub pipelines, with highest priority at the top

cs314 - update sprint.md with plan

- Definition of Done
- Policies
- Plan
  - Epics planned for this release
  - Discussion of planning decisions
- Metrics
  - Add values for the start of the Sprint
Daily Scrum

What are we doing now?

Daily Scrum

• A 15-minute time-boxed event for the development team.
• Held at the same time and place every day of the sprint.
• Plans work for the next 24 hours.
• Inspects work since last Daily Scrum.
• Forecast upcoming Sprint work.
• Optimized the probability to meet the Sprint Goal.
Some Key Questions

- What did I do yesterday that helped meet the Sprint Goal?
- What will I do today to help meet the Sprint Goal?
- Do I see any impediments that prevent us from meeting the Sprint Goal.

cs314 - record Daily Scrums in sprint.md

<table>
<thead>
<tr>
<th>Date</th>
<th>Done</th>
<th>In Progress</th>
<th>Impediments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/30</td>
<td>#12,#14,#15</td>
<td>#8,#16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(tasks from Zenhub)</td>
<td>(tasks from Zenhub)</td>
<td></td>
</tr>
</tbody>
</table>
cs314 - plan and track work in Zenhub

- Choose highest priority Task from Sprint Backlog pipeline, assign it, and move it to In Progress pipeline.
- Complete the task (code and tests - no separate testing tasks)
- When completed, move the Task from In Progress pipeline to the Done (?) pipeline.
- REPEAT
- Record tasks Done and In Progress during your Daily Scrum

Sprint Review

What did we do? What is next?
Sprint Review

• Held at end of Sprint to
  – Inspect the Increment
  – Adapt the Product Backlog
• Informal meeting to elicit feedback and foster collaboration
  – 4 hour meeting for 4 week sprint (less in our case)
• Scrum team and stakeholders collaborate on
  – what was done for the Sprint
  – changes to the Product Backlog during the Sprint
  – next things that could be done to optimize value.

Sprint Review

• Explains what Product Backlog items have been done and what has not been done.
• Discusses what went well during the Sprint, what problems were encountered, and how they were resolved.
• Demonstrates and answers questions about the Increment.
• Discusses the Product Backlog as it stands.
• Collaborates on what to do next as input to Sprint Planning
cs314 - Update sprint.md before Demo

- Review
  - Done
  - Not Done
  - What went well
  - Problems encountered and their resolution
- Metrics
  - Add ending values for the Sprint

---

cs314 - Demo in class

- Attendance required
- Demonstrate the increment and answer questions
- Discuss and revise the Product Backlog to start Sprint Planning
Sprint Retrospective

How could we do better?

• Opportunity for the Scrum Team to inspect itself, and create a plan for improvements in next Sprint.
  – development practices
  – product quality
• Occurs after the Sprint Review and prior to Sprint Planning.
• Three hour meeting for 4 week sprints (shorter for us)
• Identify improvements to make in the next Sprint.
Sprint Retrospective

- Inspect how the last sprint went with regards to people, relationships, processes, and tools.
- Identify and order the major items that went well and potential improvements.
- Create a plan for implementing improvements to the way the team does work.

cs314 - Update sprint.md before Demo

<table>
<thead>
<tr>
<th>Topic</th>
<th>Teamwork</th>
<th>Process</th>
<th>Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>What we changed for this Sprint*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What we did well</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What we need to work on</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What we will change next time</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>