Getting Agile with Scrum

Mike Cohn
9 November 2009

Mike Cohn - background

Agile coach and trainer
- Founding member and director of Agile Alliance and Scrum Alliance
- Founder of Mountain Goat Software
- Ran my first Scrum project back in 1995
- Typical programmer to manager etc. progression
Agenda

- Overview of Scrum
- Product backlogs
- Sprints and sprint backlog
- Tracking progress
- Scrum meetings

The Agile Manifesto

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

- Process and tools
- Comprehensive documentation
- Contract negotiation
- Following a plan
It shouldn’t be all-or-nothing

Anticipation  Traditional  Adaptation

Anticipation  Scrum  Adaptation

Scrum Overview

© Mountain Goat Software, LLC
We’re losing the relay race

“The... ‘relay race’ approach to product development...may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach—where a team tries to go the distance as a unit, passing the ball back and forth—may better serve today’s competitive requirements.”

Scrum roles and responsibilities

**Product Owner**
- Defines the features of the product, decides on release date and content
- Is responsible for the profitability of the product (ROI)
- Prioritizes features according to market value
- Can change features and priority every sprint
- Accepts or rejects work results

**Scrum Master**
- Ensures that the team is fully functional and productive
- Enables close cooperation across all roles and functions and removes barriers
- Shields the team from external interferences
- Ensures that the process is followed. Participates in daily scrum, sprint review and planning meetings

**Team**
- Cross-functional, seven plus/minus two members
- Selects the sprint backlog
- Has the right to do everything within the boundaries of the project guidelines to reach the iteration goal
- Organizes itself and its work
- Demos work results to the Product Owner

Scrum

- Sprint goal
- Sprint 2-4 weeks
- Sprint backlog
- Potentially shippable product increment
- 24 hours
- Cancel
- Coupons
- Gift wrap
- Product backlog
- Coupons
Scrum

- Is results-oriented
- Is commitment-driven
- Is value-focused
- Empowers and respects teams

Product Backlog
Product backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

A sample product backlog

<table>
<thead>
<tr>
<th>Backlog item</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow a guest to make a reservation</td>
<td>3</td>
</tr>
<tr>
<td>As a guest, I want to cancel a reservation.</td>
<td>5</td>
</tr>
<tr>
<td>As a guest, I want to change the dates of a reservation.</td>
<td>3</td>
</tr>
<tr>
<td>As a hotel employee, I can a run RevPAR (Revenue-Per-Available-Room) report</td>
<td>8</td>
</tr>
<tr>
<td>Improve exception handling</td>
<td>8</td>
</tr>
<tr>
<td>...</td>
<td>30</td>
</tr>
<tr>
<td>...</td>
<td>50</td>
</tr>
</tbody>
</table>
User stories as backlog items

Card
• Stories are traditionally written on note cards.
• May be annotated with notes, estimates, etc.

Conversation
• Details behind the story come out during conversations with product owner

Confirmation
• Acceptance tests confirm the story was coded correctly

Source: XP Magazine 8/30/01, Ron Jeffries.

Samples from a travel website

As a user, I want to reserve a hotel room.
As a vacationer, I want to see photos of the hotels.

As a frequent flyer, I want to rebook a past trip, so that I save time booking trips I take often.

Use this template:
As a <type of user>, I want <some goal> so that <some reason>.
Write some user stories about things some “users” of the software development process would want

As a developer, I do not want to be forced to work inordinate or prolonged amounts of overtime.

As a customer, I want a high degree of predictability of scope and date.

The product backlog iceberg

Sprint
Release
Future Releases

Priority
Some useful terms

**Theme**
A collection of related user stories.

**User Story**
A description of desired functionality told from the perspective of the user or customer.

**Epic**
A large user story.

---

An example

As a VP Marketing, I want to review the performance of historical promotional campaigns so that I can identify and repeat profitable ones.

Clearly an epic

As a VP Marketing, I want to select the timeframe to use when reviewing the performance of past promotional campaigns, so that I can identify and repeat profitable ones.

Epics??

As a VP Marketing, I can select which type of campaigns (direct mail, TV, email, radio, etc.) to include when reviewing the performance of historical promotional campaigns.
An example

As a VP Marketing, I want to see information on **direct mailings** when reviewing historical campaigns.

As a VP Marketing, I want to see information on **television advertising** when reviewing historical campaigns.

As a VP Marketing, I want to see information on **email advertising** when reviewing historical campaigns.

---

Sprints and Sprint Backlog
A sprint backlog

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write the foo class</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Add error logging</td>
<td></td>
<td>8</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Potentially shippable product increment

- At the end of each sprint, the team must produce a potentially shippable product increment
  - High quality, tested, complete, done, what it does it does well
- Potentially shippable ≠ shippable
- Product increment may not be cohesive
  - Print preview but no print
Always deliver

- You must have a potentially shippable product increment at the end of each sprint
- Do not miss the end of the sprint
  - The deadline is sacred
  - Functionality may vary

Sprints

1. Is there such a thing as an “analysis sprint” where requirements are pulled together?
2. Is there such a thing as a “testing sprint”?
3. What is a “stabilization sprint” and what should be done with it?
Architecture built over time

Reciprocal commitments

The team commits to delivering some amount of functionality

The business commits to leave priorities alone during the sprint
No changes during a sprint

• What the team commits to—and what the product owner agrees to—during sprint planning should be what is delivered

However, keep in mind that...

• We start with vague requirements
• Our understanding of those requirements is refined during the sprint

Abnormal terminations

• If change cannot be kept out of a sprint...
  • The sprint may be abnormally terminated
  • An extreme circumstance, not done very often
  • Raises visibility of priority changes

STOP
Deciding to abnormally terminate

- Team can abnormally terminate if…
  - They feel they cannot meet the sprint goal
- Management can abnormally terminate if…
  - Business priorities change

After abnormally terminating...
- All work from the current sprint is undone
- Code reverts to where it was at the end of the prior sprint
- Next step is to plan a new sprint

Release sprints

- Always target a potentially-shippable product increment
- But, some polishing can occur in a release sprint
  - Mean Time Between Failure (MTBF) testing
  - Some stress, performance or usability testing
  - Compliance
  - Documentation touchups (final screen shots)
## A product backlog

<table>
<thead>
<tr>
<th>Story</th>
<th>Points</th>
<th>Done</th>
</tr>
</thead>
<tbody>
<tr>
<td>As the site editor, I can add an article to the site.</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>As a site visitor, I want to read a new article on the front page about once a week.</td>
<td>5</td>
<td>✓</td>
</tr>
<tr>
<td>As the site editor, I can include a teaser with each article.</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>As a site member who has read a teaser on the front page, I want to read the entire article.</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>As a site visitor, I can do a full-text search of article body, title, and author name.</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>As a site visitor, I can subscribe to an RSS feed of articles.</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>As a site visitor, I can post comments about articles so that others can read them.</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>
Velocity

- A useful long-term measure of the amount of work completed per sprint

![Velocity chart]

Velocity is measured in the units you use to estimate product backlog items.

When will this project be released?

Four Lessons

Burndown charts:
- Show net progress
- Raise questions: they don’t answer them
- Facilitate early discussions
- Make it impossible to lie
Knowing more precisely where we are at the end of sprint is very powerful.

What would be the impact in your organization of knowing how the project is going each sprint?

A sprint burndown chart
<table>
<thead>
<tr>
<th>Tasks</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the user interface</td>
<td>8</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code the middle tier</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Test the middle tier</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>Write online help</td>
<td></td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Task boards**

**Story** | **To Do** | **In Process** | **To Verify** | **Done**
---|---|---|---|---
As a user, I... 8 points | Code the... 8 | Test the... 4 | Code the... LC 8 | Test the... MC 8
| Code the... 8 | Code the... 8 | Test the... SC 8 | Test the... SC 4
| Code the... 4 | Test the... 4 | | |

As a user, I... 5 points

| Code the... 8 | Test the... 8 | Code the... Dc 8 | Test the... SC 8
| Code the... 8 | Test the... 8 | | |
A sample task board
A cork task board

Velcro in a team room
Scrum meetings

- Sprint Planning Meeting
- Daily Scrums
- Sprint Review Meeting
- Sprint Retrospective
- Sprint Planning Meeting
Sprint planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
  - Tasks are identified and each is estimated (1-16 hours)
  - Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

As a vacation planner, I want to see photos of the hotels.

- Code the middle tier (8 hours)
- Code the user interface (4)
- Write test fixtures (4)
- Code the foo class (6)
- Update performance tests (4)
The daily scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings

Everyone answers 3 questions

1. What did you do yesterday?
2. What will you do today?
3. What, if anything, is in your way?

- These are not status for the ScrumMaster
- They are commitments in front of peers
The sprint review

• Team presents what it accomplished during the sprint
• Typically takes the form of a demo of new features or underlying architecture
• Informal
  • No slides
  • 2-hour prep time guideline
• Whole team participates
• Invite the world

Sprint retrospective

• Periodically take a look at what is and is not working
• Typically 15–30 minutes
• Done after every sprint
• Whole team participates
  • ScrumMaster
  • Product owner
  • Team
  • Possibly customers and others
Start / Stop / Continue

• Whole team gathers and discusses what they’d like to:

  Start doing

  Stop doing

  Continue doing

This is just one of many ways to do a sprint retrospective.

A start, stop, continue list

**Start**
- Showing the software to customers early
- Specifying acceptance tests early and with customers
- Doing code inspections
- Getting FitNesse into the nightly builds
- Trying to finish one story before moving to the next

**Stop**
- Being disrespectful of QA

**Continue**
- Making progress with the canonical database
- Emphasizing test automation
Scrum of scrums

Scalability in Scrum comes from having teams of teams, not larger teams.

Scrum of scrums of scrums
Is Scrum right for you?

Yes, probably. But...

- Remember that Scrum requires empowered teams
- Can you allow the team the freedom to self organize?
- Do your teams have the discipline to be agile or do they equate agile to lazy?
- Will you have the courage and stamina to resolve the issues that Scrum surfaces?
  - The problems were always there, but now they’re more visible
- Remember, Scrum is an approach, not a list of rules

Upcoming public classes

<table>
<thead>
<tr>
<th>Date</th>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feb 1–2</td>
<td>Certified ScrumMaster</td>
<td>Dallas</td>
</tr>
<tr>
<td>Feb 3–4</td>
<td>Certified Scrum Product Owner</td>
<td>Dallas</td>
</tr>
<tr>
<td>March 1</td>
<td>User Stories for Agile Requirements</td>
<td>Boulder</td>
</tr>
<tr>
<td>March 2–3</td>
<td>Certified ScrumMaster</td>
<td>Boulder</td>
</tr>
<tr>
<td>March 4</td>
<td>Agile Estimating &amp; Planning</td>
<td>Boulder</td>
</tr>
<tr>
<td>April 12</td>
<td>User Stories for Agile Requirements</td>
<td>San Diego</td>
</tr>
<tr>
<td>April 13–14</td>
<td>Certified ScrumMaster</td>
<td>San Diego</td>
</tr>
<tr>
<td>April 15</td>
<td>Succeeding with Scrum (new!)</td>
<td>San Diego</td>
</tr>
</tbody>
</table>

See mountaingoatsoftware.com for details