Planning and Tracking Agile Projects

Mike Cohn
August 15, 2007

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

© Mountain Goat Software, LLC
Imagine...

• That you’re fed up with software development as a career
• And you decide to go into the landscaping business
• Your first job is moving this pile of rock from the front of my house to the back

How might you estimate this?

• One way:
  • Look at the pile of rock and estimate how many wheelbarrow loads it represents
  • After an hour, see how many wheelbarrow loads you’ve moved then extrapolate the total duration

• I think that’s 80 wheelbarrow loads
• After an hour I’ve moved 20 loads
• So, I’ll be done in a total of 4 hours
My landscaping

Wheelbarrow Loads vs Time

- An iteration is a short, constrained period of time
- Typically 1-4 weeks

Velocity is the amount of work planned or completed in an iteration.

A release typically comprises more than one iteration.
The planning onion

- Agile teams plan at the innermost three levels.
- Others (on the team in the company) plan at the outer levels.

Relating the different planning levels

<table>
<thead>
<tr>
<th>Product Backlog</th>
<th>Iteration Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>Code the UI 8</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>Write test fixture 6</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>Code middle tier 12</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>Write tests 5</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>Automate tests 4</td>
</tr>
</tbody>
</table>

“Yesterday I started on the UI; I should finish before the end of today.”
**Product, release, sprint planning**

We'll focus here today

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task A</td>
<td>8</td>
</tr>
<tr>
<td>Task B</td>
<td>16</td>
</tr>
<tr>
<td>Task C</td>
<td>5</td>
</tr>
<tr>
<td>Task D</td>
<td>8</td>
</tr>
</tbody>
</table>

**Agenda**

- Estimating
- Release planning
- Burndown charts
Story points

• Probably the most commonly used estimating unit among agile teams today
  • Name is derived from agile teams commonly expressing requirements as “user stories”
• Based on a combination of the size and complexity of the work
• Unitless but numerically relevant estimates
  • A 10-point user story is expected to take twice as long as a 5-point user story

Consider these two piles of work

What story point values might we put on these?
Zoo points

Assign “zoo points” to the following breeds:

Lion
Kangaroo
Rhinoceros
Bear
Giraffe
Gorilla
Hippopotamus
Tiger

Three key advantages

• Estimating in story points:
  1. Forces the use of relative estimating
     • Studies have shown we’re better at this†
  2. Focuses us on estimating the size, not the duration
     • We derive duration empirically by seeing how much we complete per iteration
  3. Puts estimates in units that we can add together
     • Time based estimates are not additive

Comparing apples to apples

Product Backlog

<table>
<thead>
<tr>
<th>Task</th>
<th>Estimated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>30</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>50</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>50</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>20</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>20</td>
</tr>
</tbody>
</table>

Sprint Backlog

<table>
<thead>
<tr>
<th>Task</th>
<th>Estimated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the UI</td>
<td>8</td>
</tr>
<tr>
<td>Write test fixture</td>
<td>6</td>
</tr>
<tr>
<td>Code middle tier</td>
<td>12</td>
</tr>
<tr>
<td>Write tests</td>
<td>5</td>
</tr>
<tr>
<td>Automate tests</td>
<td>4</td>
</tr>
</tbody>
</table>

“Yesterday I started on the UI; I should finish before the end of today.”

Planning poker

- An iterative approach to estimating
- Steps
  - Each estimator is given a deck of cards, each card has a valid estimate written on it
  - Customer/Product owner reads a story and it’s discussed briefly
  - Each estimator selects a card that’s his or her estimate
  - Cards are turned over so all can see them
  - Discuss differences (especially outliers)
  - Re-estimate until estimates converge
## Planning poker - an example

![Planning poker cards](image)

<table>
<thead>
<tr>
<th>Estimator</th>
<th>Round 1</th>
<th>Round 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susan</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Vadim</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Ann</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Chris</td>
<td>5</td>
<td>8</td>
</tr>
</tbody>
</table>

### Estimate these

<table>
<thead>
<tr>
<th>Product backlog item</th>
<th>Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read a high-level, 10-page overview of agile software development in <em>People</em> magazine.</td>
<td></td>
</tr>
<tr>
<td>Read a densely written 5-page research paper about agile software development in an academic journal.</td>
<td></td>
</tr>
<tr>
<td>Write the product backlog for a simple eCommerce site that sells only clocks.</td>
<td></td>
</tr>
<tr>
<td>Recruit, interview, and hire a new programmer to join your 20-person startup.</td>
<td></td>
</tr>
<tr>
<td>Create a 60-minute presentation about agile estimating and planning for your coworkers.</td>
<td></td>
</tr>
<tr>
<td>Wash and wax your boss’ Porsche.</td>
<td></td>
</tr>
<tr>
<td>Read a 150-page book on agile software development.</td>
<td></td>
</tr>
<tr>
<td>Write an 8-page summary of that book for your boss.</td>
<td></td>
</tr>
</tbody>
</table>
Why planning poker works

- Those who will do the work, estimate the work
- Estimators are required to justify estimates
- Focuses most estimates within an approximate one order of magnitude


Why planning poker works

- Combining of individual estimates through group discussion leads to better estimates
- Emphasizes relative rather than absolute estimating
- Estimates are constrained to a set of values so we don’t waste time in meaningless arguments
- Everyone’s opinion is heard
- It’s quick and fun

Reduces likelihood of anchoring

Control group
- Given a product spec
  - 456 hours

High anchor group
- Given the same product spec
- Told the customer thinks 500 hours is a reasonable estimate but that
  - The customer knows very little about the implications of his spec on the estimate
  - You shouldn’t let his number influence you
  - 555 hours

Low anchor group
- Same as high but customer thinks 50 hours
  - 99 hours

Source: How to avoid impact from irrelevant and misleading information on your cost estimates, Magne Jørgensen and Stein Grimsæd, Simula Research Laboratory. Simula Research Labs Estimation Seminar, Oslo, Norway 2006.
Release planning

Purpose
To answer questions such as:
• How much will be done by 30 June?
• When can we ship with this set of features?
• How many people or teams should be on this project?

Inputs
• Velocity
• The length of the project
• Prioritized product backlog
An example with velocity=14

What if I’m given a fixed date?

<table>
<thead>
<tr>
<th>Desired release date</th>
<th>30 June</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today’s Date</td>
<td>1 January</td>
</tr>
<tr>
<td>Number of sprints</td>
<td>6 (monthly)</td>
</tr>
<tr>
<td>Low velocity</td>
<td>15</td>
</tr>
<tr>
<td>High velocity</td>
<td>20</td>
</tr>
</tbody>
</table>

Will have

6×15

Might have

6×20

Won’t have
What if I’m given a fixed scope?

<table>
<thead>
<tr>
<th>Total story points desired</th>
<th>120</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low velocity</td>
<td>15</td>
</tr>
<tr>
<td>High velocity</td>
<td>20</td>
</tr>
</tbody>
</table>

120 ÷ 20 =

120 ÷ 15 =

Updating the release plan

- Use multiple views of observed velocity

Mean (Worst 3) = 28
Mean (Last 8) = 33
Last Observation = 36
Extrapolate from velocity

At our slowest velocity we’ll finish here

At our long-term average we’ll finish here

At current velocity we’ll finish here

Agenda

- Estimating
- Release planning
- Burndown charts
How's my landscaping coming?

This is called a burndown chart.

Remember the different levels?

Product Backlog

<table>
<thead>
<tr>
<th>Iteration 1</th>
<th>Product Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>3</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>5</td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td></td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td></td>
</tr>
<tr>
<td>As a frequent flyer, I want to...</td>
<td></td>
</tr>
</tbody>
</table>

Iteration Backlog

<table>
<thead>
<tr>
<th>Iteration 1</th>
<th>Iteration Backlog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the UI</td>
<td>8</td>
</tr>
<tr>
<td>Write test fixture</td>
<td>6</td>
</tr>
<tr>
<td>Code middle tier</td>
<td>12</td>
</tr>
<tr>
<td>Write tests</td>
<td>5</td>
</tr>
<tr>
<td>Automate tests</td>
<td>4</td>
</tr>
</tbody>
</table>

We can track burndown at both levels

Today I started on the UI; I should finish before the end of today.
An iteration burndown chart

A release burndown chart

Four Lessons
- Burndown charts:
  - Show net progress
  - Raise questions; they don’t answer them
  - Facilitate early discussions
  - Make it impossible to lie
Mike Cohn contact info

mike@mountaingoatsoftware.com
www.mountaingoatsoftware.com
(720) 890-6110 (office)
(303) 810-2190 (mobile)