An Introduction to Agile Estimating & Planning

Mike Cohn
February 10, 2010

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
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
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>
<tr>
<td>As a frequent flyer, I want to...</td>
<td>“Yesterday I started on the UI; I should finish before the end of today.”</td>
</tr>
</tbody>
</table>

© Mountain Goat Software, LLC
Product, release, iteration 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
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 everything that influences the effort to develop a feature
- Unitless but numerically relevant estimates
  - A 10-point user story is expected to take twice as long as a 5-point user story

Dog points

Assign “dog points” to the following dogs

- Labrador retriever
- Dachshund
- Great Dane
- Terrier
- German Shepherd
- Poodle
- St. Bernard
- Bulldog
Consider these two piles of work

What story point values might we put on these?

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

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

<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 your uncle who owns a clock store and now wants an eCommerce site.</td>
<td></td>
</tr>
<tr>
<td>Recruit, interview, and hire a new member for your team.</td>
<td></td>
</tr>
<tr>
<td>Create a 60-minute presentation to introduce agile software development to your non-agile team.</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 work¹
• Estimators are required to justify estimates²,³
• Focuses most estimates within an approximate one order of magnitude⁴,⁵


Why planning poker works

• Combining of individual estimates⁶ through group discussion⁷ 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 impact of irrelevant information

**Group A**
- Given project spec.
- 20 hours

**Group B**
- Given same spec but with estimation-irrelevant details added:
  - end users’ desktop applications
  - user passwords,
  - etc.
- 39 hours

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

Specification length

**Group A**
- Given a one-project spec.
- 117 hours

**Group B**
- Given a spec with exactly the same text but was 7 pages long
- Increased length achieved through
  - double line space
  - wide margins
  - larger font size
  - more space between paragraphs
- 173 hours

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

Group A
- Given requirements R1–R4
- 4 hours

Group B
- Given requirements R1–R5
- 4 hours

Group C
- Given requirements R1–R5
- but told to estimate R1–R4 only
- 8 hours!

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

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 Grimstad, Simula Research Laboratory, Simula Research Labs Estimation Seminar, Oslo, Norway 2006.
Free, or I wouldn’t mention it

Agenda

- Estimating
- Release planning
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

Iteration 1

<table>
<thead>
<tr>
<th>Story</th>
<th>Iteration 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story A</td>
<td>5</td>
</tr>
<tr>
<td>Story B</td>
<td>8</td>
</tr>
<tr>
<td>Story C</td>
<td>3</td>
</tr>
<tr>
<td>Story D</td>
<td>5</td>
</tr>
<tr>
<td>Story E</td>
<td>1</td>
</tr>
</tbody>
</table>

Iteration 2

<table>
<thead>
<tr>
<th>Story</th>
<th>Iteration 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Story F</td>
<td>5</td>
</tr>
<tr>
<td>Story G</td>
<td>1</td>
</tr>
<tr>
<td>Story H</td>
<td>13</td>
</tr>
<tr>
<td>Story I</td>
<td>5</td>
</tr>
<tr>
<td>Story J</td>
<td>8</td>
</tr>
</tbody>
</table>
Updating the release plan

Mean (Best 3) = 37
Mean (Last 8) = 33
Mean (Worst 3) = 28

Extrapolate from velocity

Assume: There are five iterations left.

At our slowest velocity we'll finish here (5×28)
At our long-term average we'll finish here (5×33)
At our best velocity we'll finish here (5×37)
Fixed-date planning

1. Determine how many iterations you have
2. Estimate velocity as a range
3. Multiply low velocity $\times$ number of iterations
   - Count off that many points
   - These are “Will Have” items
4. Multiply high velocity $\times$ number of iterations
   - Count off that many more points
   - These are “Might Have items”

Fixed-date planning: an example

<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 iterations</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
Fixed-date contracting

Will have

\[ 6 \times 15 \]

Might have

\[ 6 \times 20 \]

Won’t have

It's a risk issue

Where do you want to be?

If you write a contract for just the will haves:

- You won’t likely win the contract
- But you’ll probably make money if you do

If you write a contract that includes the might haves:

- You will likely win the contract
- But probably not make money on it

Upcoming public classes

<table>
<thead>
<tr>
<th>Date</th>
<th>What</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>March 1–2</td>
<td>Certified ScrumMaster Succeeding with Agile</td>
<td>Boulder</td>
</tr>
<tr>
<td>March 3–4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>April 12</td>
<td>Effective User Stories Certified ScrumMaster</td>
<td>La Jolla</td>
</tr>
<tr>
<td>April 13–14</td>
<td>Agile Estimating and Planning</td>
<td></td>
</tr>
<tr>
<td>April 15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>May 17–18</td>
<td>Certified ScrumMaster Succeeding with Agile</td>
<td>Sunnyvale</td>
</tr>
<tr>
<td>May 19–20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>July 19–20</td>
<td>Certified ScrumMaster</td>
<td>Orlando</td>
</tr>
<tr>
<td>July 21–22</td>
<td>Certified Scrum Product Owner</td>
<td></td>
</tr>
</tbody>
</table>

Information and registration at
www.mountaingoatsoftware.com