Estimation Workshop

Anthony Oden
BishopTTU@gmail.com
Learning Objectives

- Estimating Size, deriving duration & why it’s important
- What’s Velocity?
- Story Points & Ideal Days – What are they & when to use them
- Estimation effort
- Estimation techniques
- Planning Poker
- Re-Estimation
How long will it take to Tile this floor?

Estimate tiling my floor

17 Tiles

14 Tiles

How will you estimate it?

How long will it take to Tile this floor?
Estimate tiling my floor (con’t)

Now how long?

17 Tiles

14 Tiles
Did you count tiles to determine size?

Duration is determined by how long it can take to set each tile.
Estimate Size, Determine Duration

We *estimate size* but *derive duration*.
Agile Term: Velocity

Velocity is the number of accepted story points a team can accomplish in one Sprint.

Velocity is measured in Story Points or ideal days per Sprint.
Estimate Size, Determine Duration

By maintaining estimates in relative size we do not need to re-estimate the user stories if velocity changes for any reason.

*Estimates are still valid if velocity changes to adjust schedule*
Estimate Size, Determine Duration

However, re-Estimating Stories sometimes is necessary as project knowledge is gained.
Story Points
Estimate User Stories

*Story Points* are an arbitrary unit representing the estimated “size” of a user story relative to similar user stories. Story Points are used because....

- People are good at estimating sizes of things relative to other things.
- Relative estimates stay constant and can be scaled if resources or velocity changes.
Estimate User Stories

Story Points can be expressed in whatever units the team decides.

Fibonacci Sequence 1, 2, 3, 5, 8, 13, 21, 34

T-Shirt Sizes S, M, L, XL, XXL
Estimate User Stories

Story Points can be expressed in whatever units the team decides.

**Gummy Bears - Pink, Green, Blue, Red, Purple, Off-Yellow, Orange, Yellow**

**Ideal Days**
Ideal Days
Estimate Ideal Days

*Ideal Days* are units representing an ideal work day. That is the amount of time in a day that a developer works directly on the project exclusive of other supporting tasks (company meetings, etc.).

Workday times $n = \text{Ideal Day}$

XP teams used to refer to this a “load factor”.

Estimate Ideal Days

Ideal Days can be easier for people to understand initially and might be easier to estimate at first.

Ideal Days force some evaluation into the efficiencies and might drive measures to allow developers to focus with less distractions.
Should I use Story Points or Ideal Days When Estimating?
### Should I use Story Points or Ideal Days?

<table>
<thead>
<tr>
<th><strong>Story Point</strong></th>
<th><strong>Ideal Days</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Help drive cross functional behavior</td>
<td>Easier to explain to others</td>
</tr>
<tr>
<td>Story Point estimates stay viable over time</td>
<td>Force companies to confront time wasting inefficiencies</td>
</tr>
<tr>
<td>Story points are a measure of size alone</td>
<td>Ideal days seem easier to estimate at first when that’s what we’re used to</td>
</tr>
<tr>
<td>Story point estimation can be faster and less complex</td>
<td></td>
</tr>
<tr>
<td>Story Points can be additive while ideal days vary from person to person</td>
<td></td>
</tr>
</tbody>
</table>
Should I use Story Points or Ideal Days?

While this certainly can vary from team to team, the current best practice is to:

✓ Estimate user stories in story points
✓ Estimate tasks in fractional days and ideal days.
How we Estimate
Estimation Techniques

Estimation can be approached in 3 ways...

Disaggregation

Decomposing the user story in to smaller stories estimating the smaller more estimable story. (User Story Decomposition)

Expert Opinion and Experience

Allowing experts to make experienced guesses as to how long it will take. (e.g. Wide band delphi)

Comparative Analysis

Using the size of comparable projects to form a basis of estimates for other user stories. (e.g. Starbucks Coffee Sizes)
Estimation Techniques

Let those who will do the work estimate the work.

In “A Causal Model for Software Cost Estimating Error” the conclusion states:

*The number one way to improve software estimation is to ascribe accountability for the estimate even though no methodology is expressly stated. Tie it right to the estimator’s performance review.*

Agile estimation makes the entire team responsible for the estimate. Again, people over process.
Estimation Techniques

How much estimating is enough?

A little work goes a long way...
Estimation Techniques

One way to estimate story points is to use planning poker.
Planning Poker

1. Customer / Proxy explains story/feature/theme/requirement
2. Team discusses work involved
3. Everybody estimates individually
4. Everybody reveals estimate simultaneously
5. Lowest and highest estimator justifies
6. Team decides on collective estimate

Planning Poker was a term coined by James Grenning (2002).
Estimation Techniques

Why Estimation Poker works..

Multiple expert opinions during estimation

People doing estimates are ones doing the work.

Estimators justify estimates.

Averaging individual estimates leads to better estimates

Avoids anchoring, the tendency to assume an estimate based on existing expectations about estimates
Estimation Techniques

From the Simula Research Laboratory

“Planning Poker is an effective way to facilitate group discussions and collect and combine individual estimates”.

Combination of estimates may increase accuracy
Strive for diversity, independence and decentralization
Use group discussions also to increase motivation, increase ownership, sort out ambiguities and define scope and target quality for each task
Discuss Using Conversation Cafe

Was It Fun?

Was it simple?

Was it accurate?
Estimation Techniques

Kanban

http://availagility.wordpress.com/2008/04/09/a-kanban-system-for-software-development/

http://www.agilemanagement.net/Articles/Weblog/KanbaninAction.html
Re-Estimating
Re-Estimating

When to re-estimate?

Re-estimate the backlog after each of the first few sprints and periodically thereafter.

Knowledge about relative size and details of stories will emerge and improve the collective’s understanding.

The iteration to release plan needs to be re-evaluated at least at the iteration frequency.
Estimation Workshop

Why do we estimate in size but derive duration?

What is a Story Point?

What is an Ideal Day?

What are the recommended units to use?

Why does planning poker work?

Who does the estimates in an agile process?
Questions?
References