CPSC 491
Lecture 10:
Quiz 4
Estimation

Estimation ...
Estimation

One approach to estimation (approximately):

\[
\text{project estimate} = \sum \text{user story estimate}
\]

Feature (user story) estimation using planning poker

- involves the developers … they do the work!
- estimates based on “consensus”
- a kind of voting model … avoids takeover by one dev

What are we estimating?

Ultimately, how much effort the project will take

- E.g., number of (ideal) developer days needed to finish
- which helps determine cost (number of developers)
- which can in turn help refine project scope
- and can even help better understand features (discrepancy of effort estimate between devs and client)
What are we estimating?

Usually indirect metrics used before time (dev days) …

**Story points** … how difficult is the story/feature?
- E.g., this is a 5 point story vs a 13 point story
- where 13 implies over twice the effort of 5

**T-shirt sizing** … how difficult as XS, S, M, L, XL, 2XL, …

These metrics are agnostic about time
- only relative measures of effort / difficulty

How to Get Started Estimating a Product Backlog

1. Create a small, medium, and large pile (t-shirt sizes)
2. Place each story into one of the piles
   - Perform / agree as a team
   - This is an initial coarse-grained effort assignment
3. Pick “easiest” story and assign it a 1
   - Move through cards assigning 1
   - until you reach a story that is twice as hard
   - assign it a 2, and so on
   - again, do this as a team (e.g., using planning poker)
4. Repeat until points assigned to every story
Why use Indirect Measures of effort?

1. Usually easier to agree on effort than on time
   - Time depends on how fast people are (which can differ)
   - People aren’t great at predicting how long something will take (especially with software development)

2. We want to be careful with announcing time estimates
   - Easy to take/construe time estimates as commitments
   - Time estimates also suggest (unwarranted) accuracy

From Story Points to Time Estimates

We’ll talk more about this later …

1. Use amount of work (points) you can do in a sprint
   - Use historical team data if available (prior sprints)
   - Based on sprint length, you can estimate project length

2. Use rough points-to-hours approximation
   - Use a reference story (e.g., 2 point story)
   - Come up with a consensus hour estimate
   - Use this points-to-hours value to estimate project length
Planning Poker

Everyone has a deck of planning poker cards

Each card has a number (story points, dev days/hours)

- Note: effort implies **design, code, test, and deliver**

What is meant by:

- 0 ...
- ? ...
- coffee cup ...

What’s with the seq?