Topics

- Quiz 5
- Estimation

Homework

- HW5 due
- HW6 out
Estimation Basics

One approach to estimation (approximately):

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

Feature (user story) estimation using planning poker

- developers determine the estimates ... they do the work!
- estimates based on “consensus”
- a kind of voting model ... e.g., to avoid takeover by one developer

Ultimately, trying to estimate amount of effort project will take

- the amount of work is often expressed in units of time
- 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

Often easier to use indirect metrics 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, ...

Both of these metrics are agnostic about time

- only relative measures of effort / difficulty
Indirect Effort Estimation

One approach to start estimating a project backlog

1. Create a small, medium, and large pile (e.g., 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. Need 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 ... (more later)

1. Use amount of work (points) you can do in one sprint
   - Use historical team data if available (prior sprints)
   - Based on sprint length, you can estimate project length
   - e.g., 400 pt project, 20 pts/sprint, 2-wk sprints, implies 10 mo.

2. Use rough points-to-hours approximation
   - Use a reference story (e.g., 2 point story) ... or multiple
   - Come up with a consensus time estimate
   - Use this points-to-time 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

Basic Idea: (a). Pick story, (b). consider estimates individually, (c). play estimate card simultaneously, (d). discuss assumptions, (e). develop consensus estimate (i.e., where everyone agrees)

What is meant by:

- 0 ...
- ? ...
- coffee cup ...
- What about the sequence of numbers?
Exercise:

1. Try out planning poker to estimate story points
   - Find a 1-point “reference” story
   - Estimate 2 other stories based on the “reference” story

2. Try planning poker to estimate hour estimates
   - Determine consensus hour estimates for 2 stories
More on Estimation

Estimate

• prediction of how long a project will take

• at a certain cost, scope (and quality) level

Target

• like an external (business) objective

• “We need version 2.1 ready for a trade show in May”

• “We need the release before the holiday sales cycle”

Commitment

• a promise to deliver product by a specific date

• also can include specific scope, cost, and quality

Q: What is the target for your senior design project?
When things go wrong ...

Targets vs Estimates

- may be the same, more aggressive, more conservative
- both useful to know! ... e.g., change scope, hire more devs

Problems arise when there is a large gap between commitment and estimate

- specifically commitment $<<$ estimate

Is any gap okay?

- depends on the project, the team, uncertainty of estimates ...
- e.g., ± 5-20% may be okay (to “speed up”, “adjust scope”, etc.)