Software Estimation

CSSE3012

“You cannot control what you cannot measure”

(Famous Management Adage)

How can you estimate the size of something if you've never measured the size of anything?

(Engineering Perspective)

Estimation

- Get initial feel for project cost
- Determine if project is still feasible
- Cull and re-prioritise features based on cost
- Estimates are very coarse-grained
  - this is a first pass
  - inevitably revised as more info arrives
- Should be a range rather than a single value
- Should be based on real data
- Done as a team

Estimation ≠ Prediction

- Estimates are based on incomplete information
  - particularly early in a project
- Accuracy should increase as project progresses
  - more information is discovered
- Do not set deadlines based on estimates
- Set deadlines based on goals
  - estimates indicate confidence of achieving goal

Estimation Guidelines

- Use a structured and defined process
- Apply throughout the project
- Collect and use historical data
- Adjust to suit new projects
- Apply statistical analysis where possible
- Look for automation opportunities
- Apply multiple techniques and compare results

Estimation Challenges

- Lack of historical data leads to instinctive estimation
  - flawed project plans
  - unhappy customers and developers
- Estimates tend to be overly optimistic
- Estimation depends on experience and historical data
- Unique aspects of software projects compound the difficulties

Personal Note

$E \neq \sum$ *(things I've already worked out how to do)*

- When plucking an instant estimate out of the air, I tend to only remember the things I've thought about and solved
- Leaves the unknowns out of the sum!
- Not all hours are equal

How to Estimate?

- There are many methods available but most organisations use very primitive methods
- Classes of techniques
  - analogy
  - expert judgement
  - parametric (algorithmic) models
  - “price to win”

How not to Estimate (van Vliet)

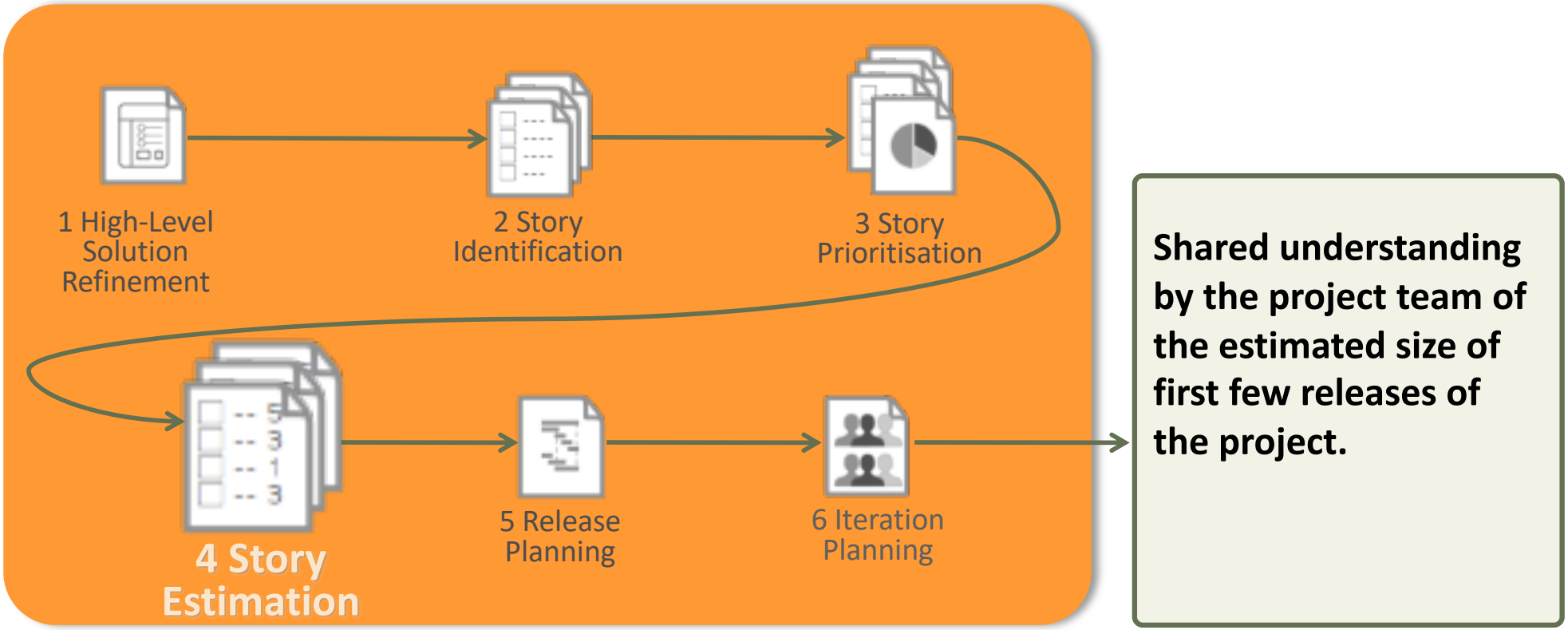
- Given 12 months to do the job, so it will take 12 months.
- Our competitor put in a bid of $1M so we need to deliver a bid of $0.9M.
- We want our product at the trade show next year, so the software must be written and tested in the next nine months.
- Project needs one year but I can’t sell that to my boss. We know 10 months is acceptable so we will settle for 10 months.

Estimating by Analogy

- Identify one or more similar projects
  - use these (or parts of them) to produce an estimate for the new project (size or effort)
- Accuracy is often improved by partitioning a project into parts and estimating each part
  - errors cancel out so long as estimating is unbiased
- Use a database of projects
  - from your own organisation
  - from multiple organisations
    - International Software Benchmarking Standards Group (ISBSG)

Story Estimation

GOAL: Assess feasibility of achieving business benefits by producing a quantified and costed story list.



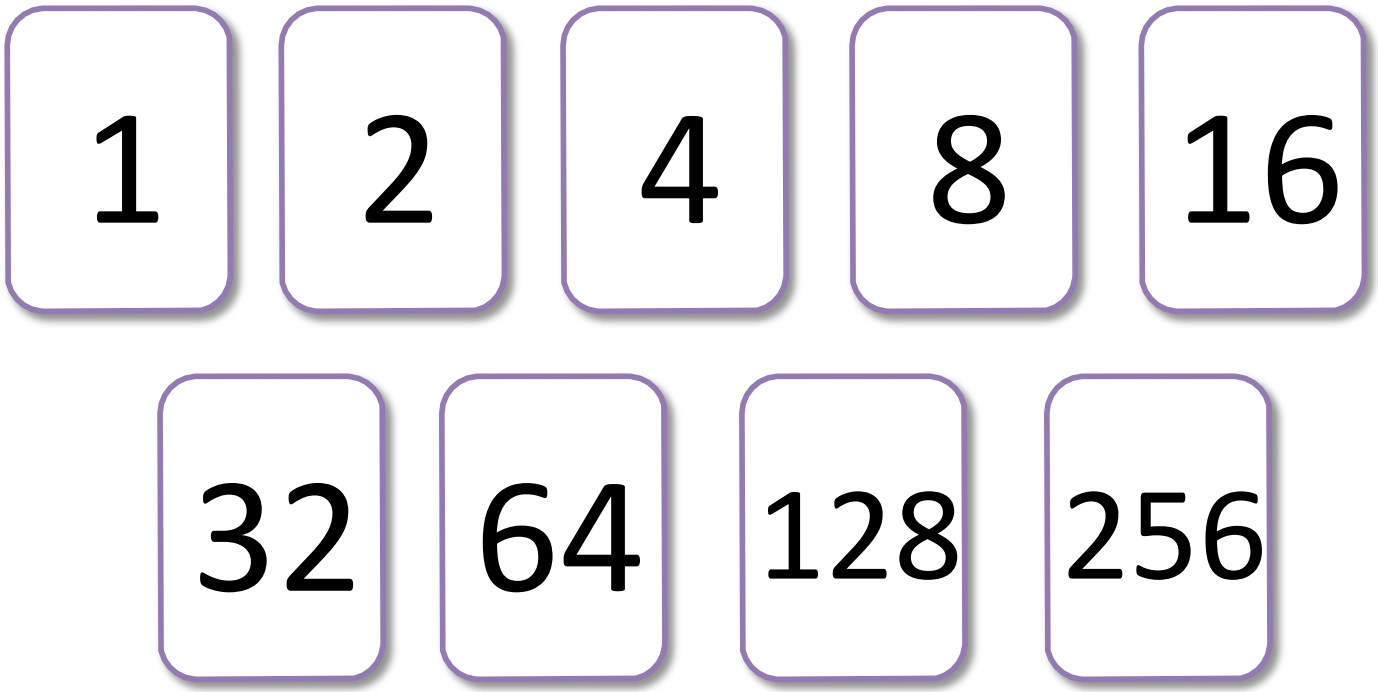
Planning Poker

- Please get into groups of about 4
- **Individually** and **without discussion with anyone** read the user requirement on the next slide
- Decide how long (in hours) you think it would take your group to implement it
  - write your estimate on a piece of paper
- You may only choose from the numbers

1, 2, 4, 8, 16, 32, 64, 128, 256

- **Do not reveal your answer to your group yet**

"As a user, I can upload an existing presentation I made in PowerPoint or Keynote, and the system will turn it into something students can page through on the Web in their browser…"



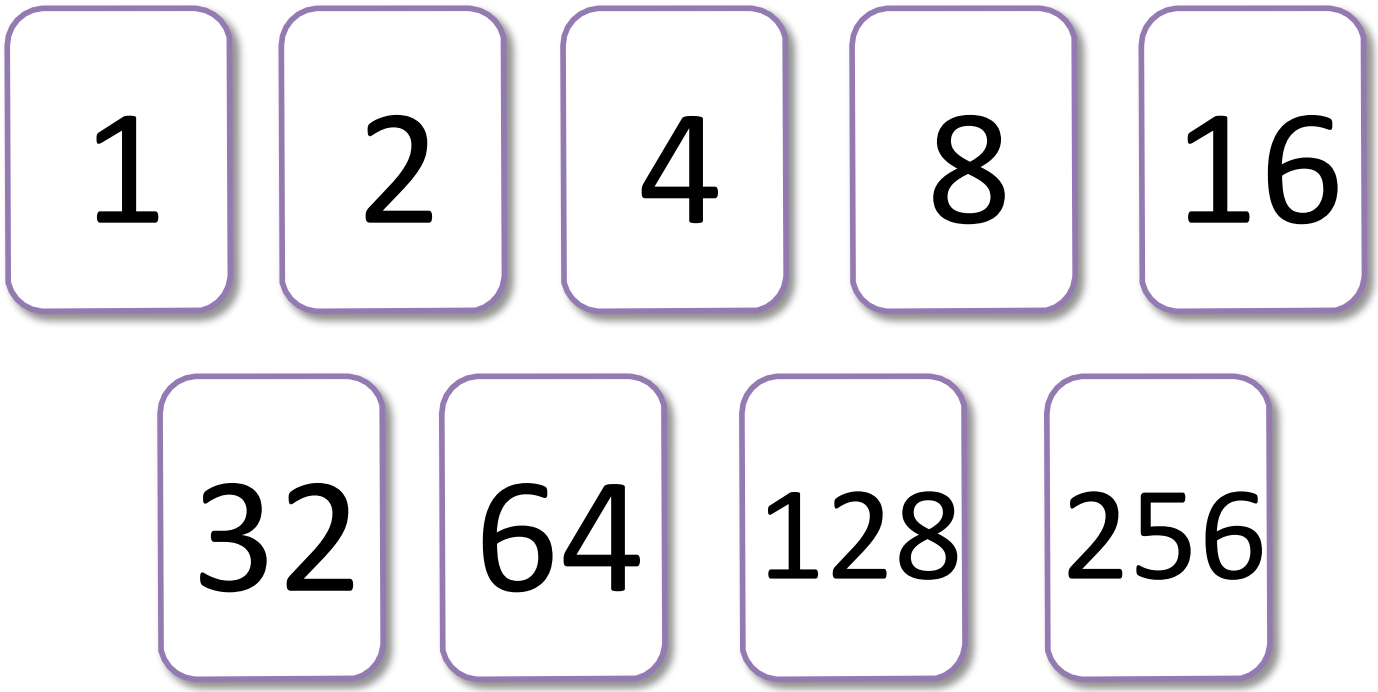
No more than 3 minutes

- Reveal your estimate to your group
- Person with the highest estimate should explain their rationale
- Person with the lowest estimate should explain their rationale
- Quick discussion

Round 2

Again individually write down how long you now would estimate it would take you to implement the user requirement

"As a user, I can upload an existing presentation I made in PowerPoint or Keynote, and the system will turn it into something students can page through on the Web in their browser…"



- Again reveal your estimates to the group
- Are the group's estimates closer together than previously?

Wide-Band Delphi

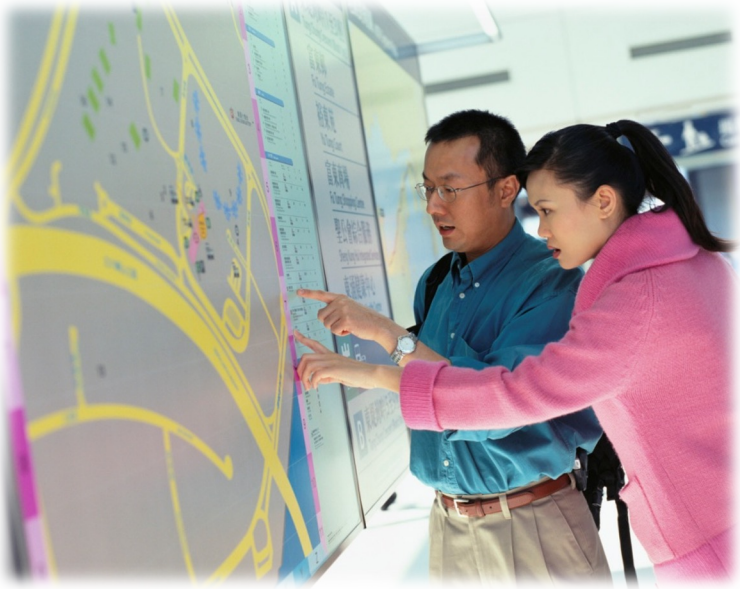
- Get multiple experts/stakeholders
- Share project information
- Each participant provides an estimate independently and anonymously
- All estimates are shared and discussed
- Each participant estimates again
- Continue until consensus, or exclude extremes and calculate average

Planning Poker

- Each player has a set of cards
  - labelled 1, 2, 4, 8, 16, 32
- One team member reads the story
- Estimate the size in story points
  - put a card with this value face down on the table
- When all cards are down they are revealed

Story Points

- Not units of time!
- Consistency
  - all 2's require the same amount of effort
- Relativity
  - a 4 is twice as big as a 2
- Fungibility
  - all 4's are interchangeable



Story Points ≠ Time

- 2 Point Story
  - Dev 1: would take 1 day
  - Dev 2: would take 3 hours
- Both agree it is a 2 point story
  - consistent
- Delivery schedule based on team's velocity
  - *not* time per point

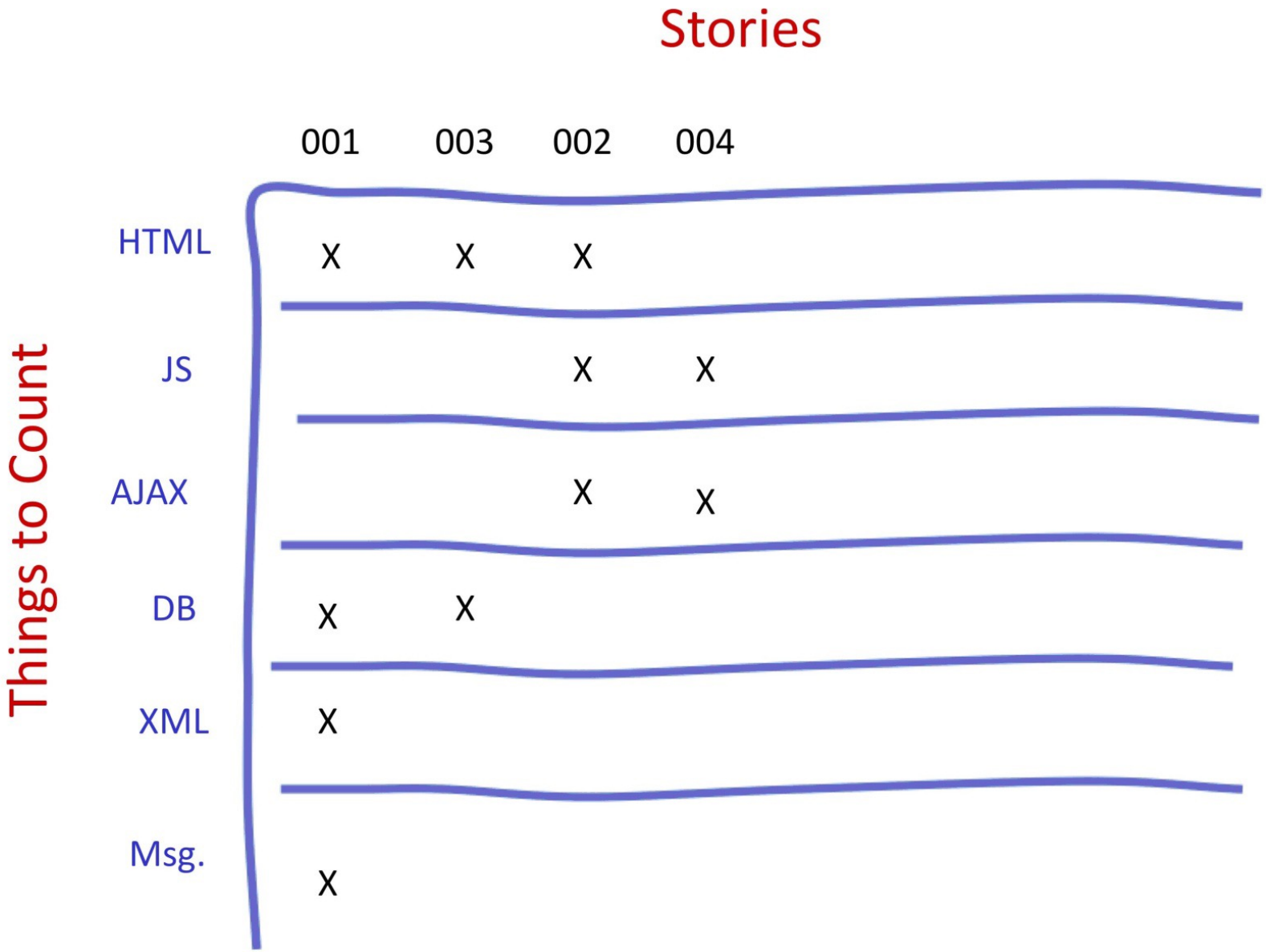
Velocity

- Team counts number of story points completed at end of each iteration
- Probably will complete about the same number in the next iteration
  - assuming no changes in the team
  - assuming similar technology
- Use velocity to convert story points to calendar effort
  - requires consistent iteration length
- Requires stable team membership
  - no “spin up” cost

Technology Grid

- It is easy to miss aspects of a requirement
- Draw a grid with a row for each technology
  - or complex system interaction
- Use as a cross-reference when discussing user story
  - helps ensure complete scope of story is understood
- First pass
  - select 10 highest priority stories
    - discuss scope and high-level acceptance criteria
      - with customer representative
  - mark each needed technology on the grid

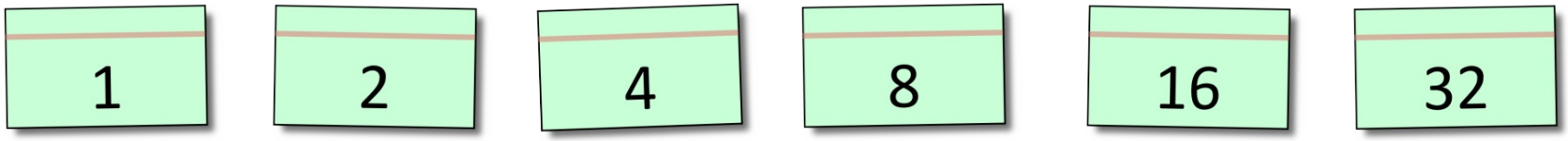
Technology Grid



Agreeing on Estimate

- Consensus
  - holders of outlying estimates explain their reasons
  - others ask clarifying questions
  - revote based on new information
- Most votes wins
- Maximum wins
- Place card on tally board

Tally Board



- Group stories that are the same size in columns
- Limited numbers make it easier to get consensus
  - emphasises that larger estimates are “fuzzy”
  - conveys lack of precision in estimates

Triangulate Estimates

- Revisit tally board after each estimation round
- Compare estimates
  - Are stories in a column roughly equal in size?
  - Are stories in adjacent columns roughly double in size?

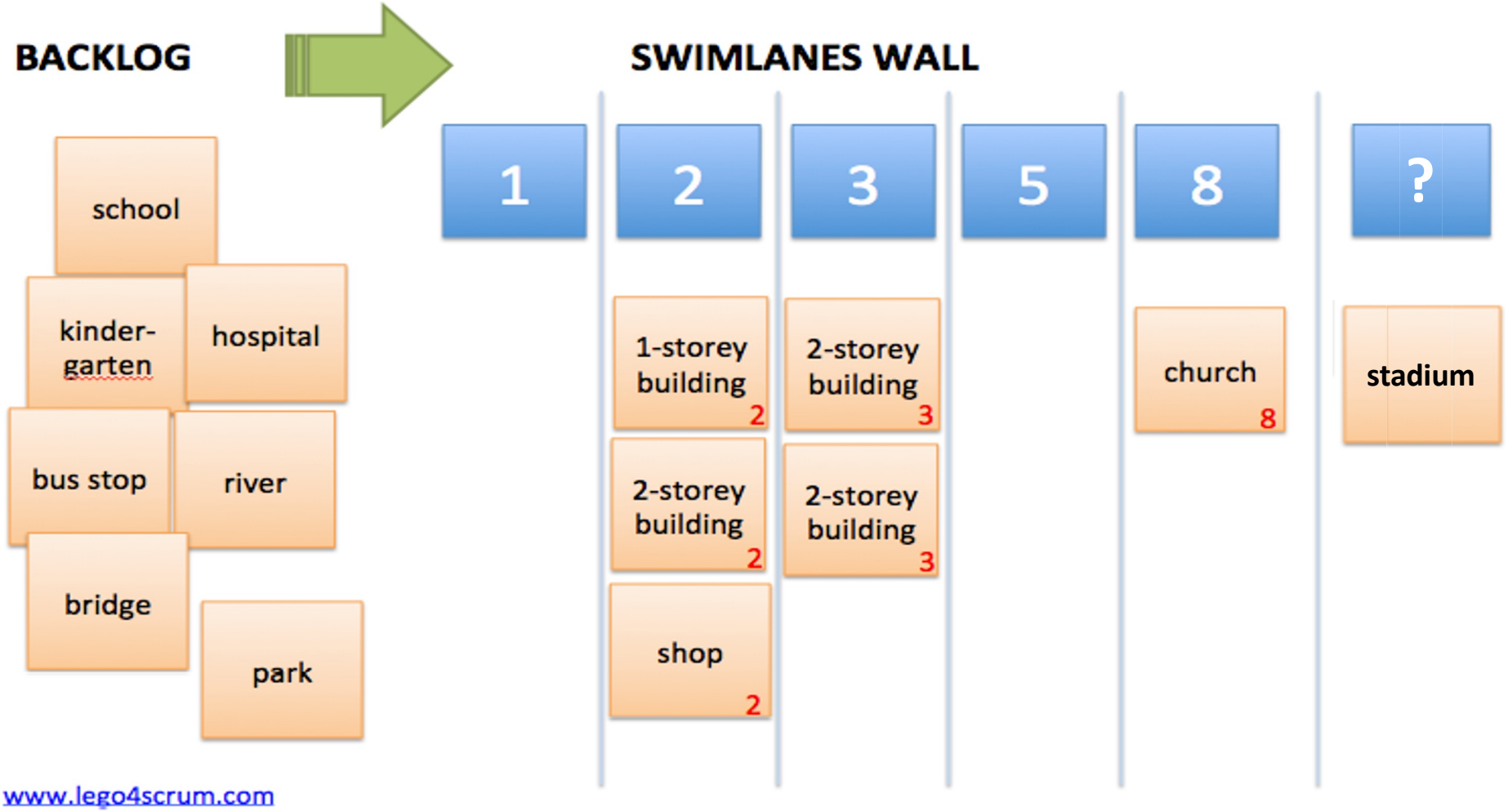
Poker Variations

- Optional control cards
  - “!” – statements are heard immediately
  - “?” – questions are queued
  - “∞” – can’t be estimated at this time

Affinity Estimating

- Once team is comfortable estimating together
- Speeds up estimating consensus cases
- Move stories into columns (*swim lanes*) on tally board
  - no discussion
  - can move stories to new swim lanes
- Quickly settle == consensus estimate
- Stories move back & forth
  - need further consideration
    - e.g. planning poker

Affinity Estimating

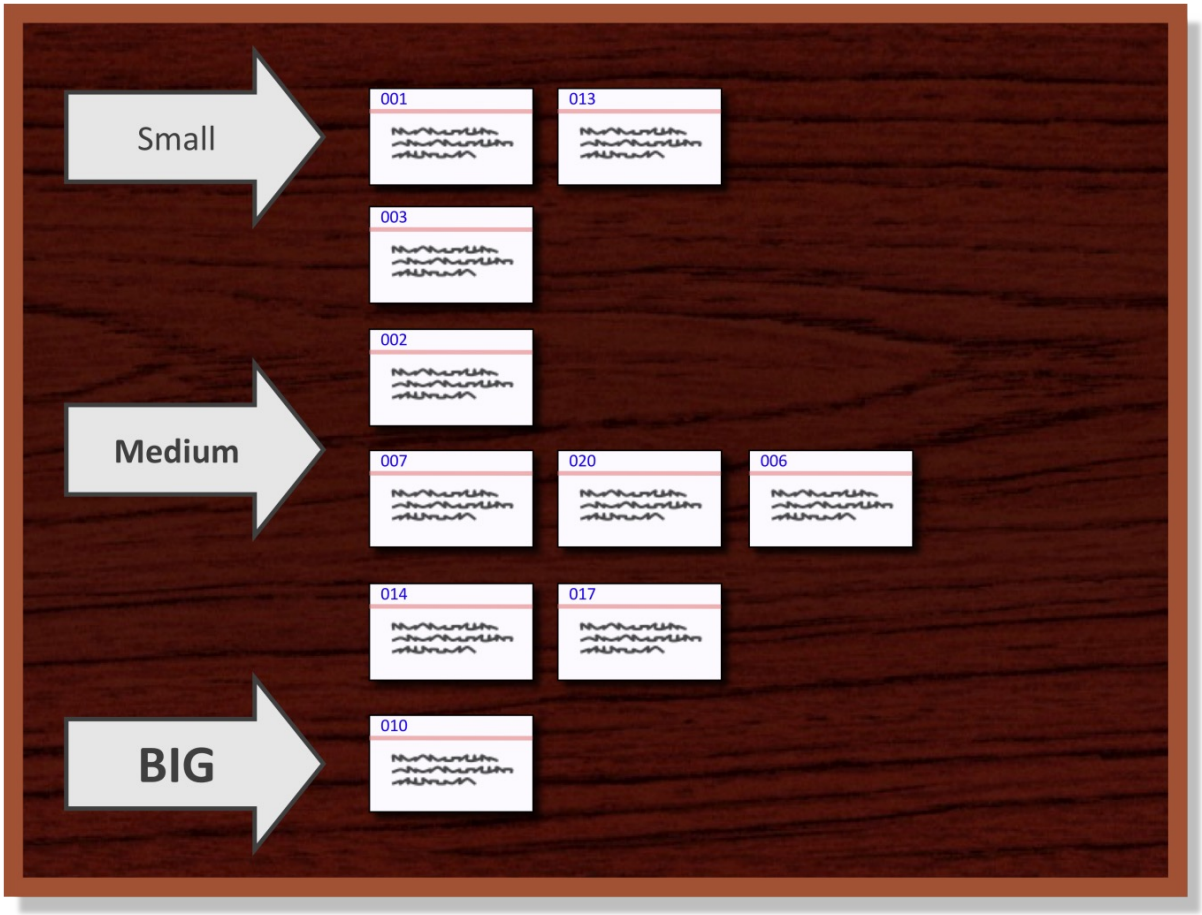


Poker Variations – Fibonacci

- Previous slide used Fibonacci numbers
  - 1, 2, 3, 5, 8, 13, 21, 34, 55, 89
- I prefer powers of two
  - makes comparison easier
    - triangulation
- Both approaches are commonly used

T-Shirt Size Estimation

- Quick, first pass estimation
- Sort stories into intuitive relative size
- Discuss any disagreement about sizes



Reading

- Sommerville, Chapter 23
- Larman, Chapter 40
- https://www.mountaingoatsoftware.com/blog/why-the-fibonacci-sequence-works-well-for-estimating
  - the point is to not get hung up on being precise
- https://cleancoders.com/blog/2021-02-16-clean-coders-planning-poker
  - providing optimistic, realistic and pessimistic estimates

Next Steps

- Lecture
  - Release and sprint planning
  - Use case modelling
- Tutorial
  - Prioritisation and Estimation