UNIVERSITY OF ICELAND

HBV401G SOFTWARE DEVELOPMENT

3. Effort Estimation

Matthias Book

Spring 2022

FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Update: Team Consultation Schedule

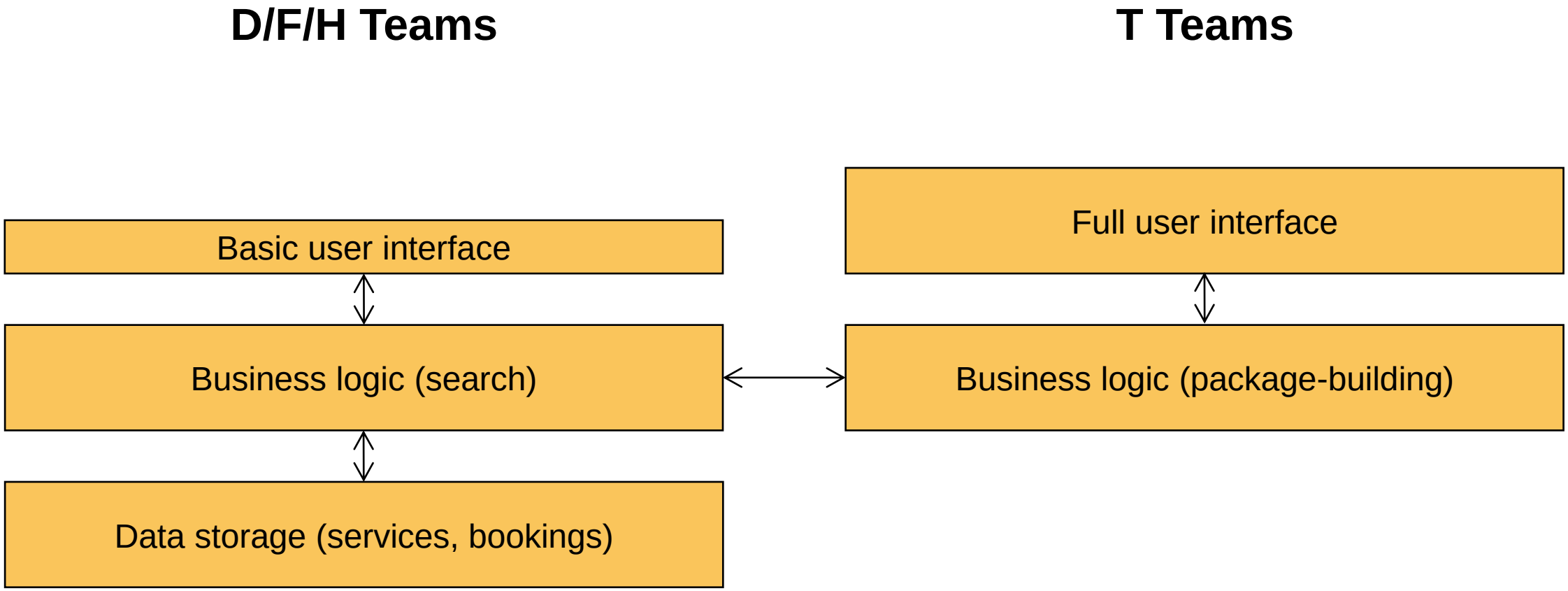
Timeslots	Marcelo	Jaan	Tristan	Valbjörn
Wed 15:00-15:20	1D (4)	3D (4)	5D (4)	7D (4)
Wed 15:25-15:45	1F (4)	3F (4)	5F (4)	7F (0)
Wed 15:50-16:10	1H (4)	3H (4)	5H (4)	7H (4)
Wed 16:15-16:35	1T (4)	3T (4)	5T (0)	7T (4)
Wed 16:40-17:00	2D (4)	4D (4)	6D (4)	8D (4)
Wed 17:05-17:25	2F (0)	4F (4)	6F (4)	8F (4)
Wed 17:30-17:50	2H (4)	4H (4)	6H (4)	8H (4)
Wed 17:55-18:15	2T (0)	4T (4)	6T (4)	8T (4)

- Team IDs: [Cluster][Component] (e.g. Team 4H: Cluster 4, Hotels component)
- Please meet your tutor in your team's voice channel ("Team XX") on Discord at the scheduled time every Wednesday

Project Scope Clarification

- **D/F/H teams**
  - Let users search and book day tours / flights / hotels in Iceland
    - You maintain a database of tours / flights / hotels (about a dozen fictional services are sufficient) that you query based on search parameters of your choice
    - You maintain a database of reservations / availabilities of your services, designed such as every booking reduces the available seats / rooms etc.
    - Your component only provides a rudimentary user interface to test this functionality
- **T teams**
  - Let users search and book travel packages in Iceland through a single search form
    - Input: Combined search parameters (e.g. “5-person 3-day stay in the North with outdoor tours”)
    - Output: Travel packages with suitable combinations of flights, hotels and tours
    - You don’t maintain any data of your own, but call the D/F/T teams’ components to perform the actual searching and booking
    - Your component provides the unified user interface to search for travel services, and the logic to combine search results from the D/F/T components into complete travel packages

Coarse Project Architecture



Quiz #2 Solution

- **Indicate if the following are (F)unctional requirements, (Q)uality requirements or (G)eneral constraints:**

a) Attach a file to an e-mail (F)

b) E-mail addresses must not contain spaces (G)

c) E-mail attachments are always sent MIME-encoded (G)

d) Enable sender to encrypt and sign e-mails (F)

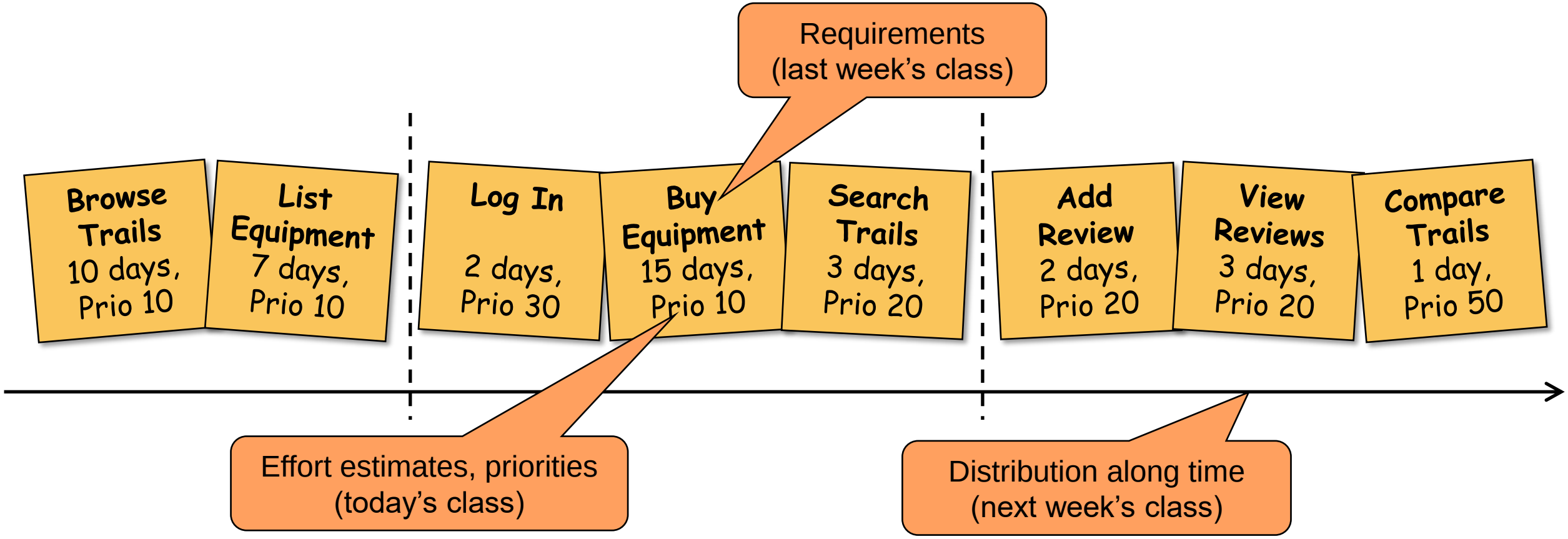
e) E-mails exchanged between users of the same mail server shall arrive within 10 seconds (Q)

f) Prevent unauthorized access to a user's e-mail account (Q)

g) Send an e-mail to several recipients (F)

h) Server must have <5% downtime (Q)

Recap: Planning an Iterative Software Project



Recap: The User Story Refinement Cycle

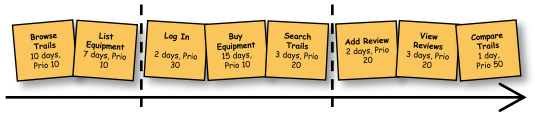
Client ideas

Use requirements elicitation techniques to gather client ideas and requirements

Construct user stories from requirements with client

Fill gaps and add clarity on details with client feedback

Estimate and prioritize user stories



Refine stories into tasks for sprint planning and progress monitoring

How your Brain Works:
Anchoring Effect

Any number that is on your mind will influence your estimate

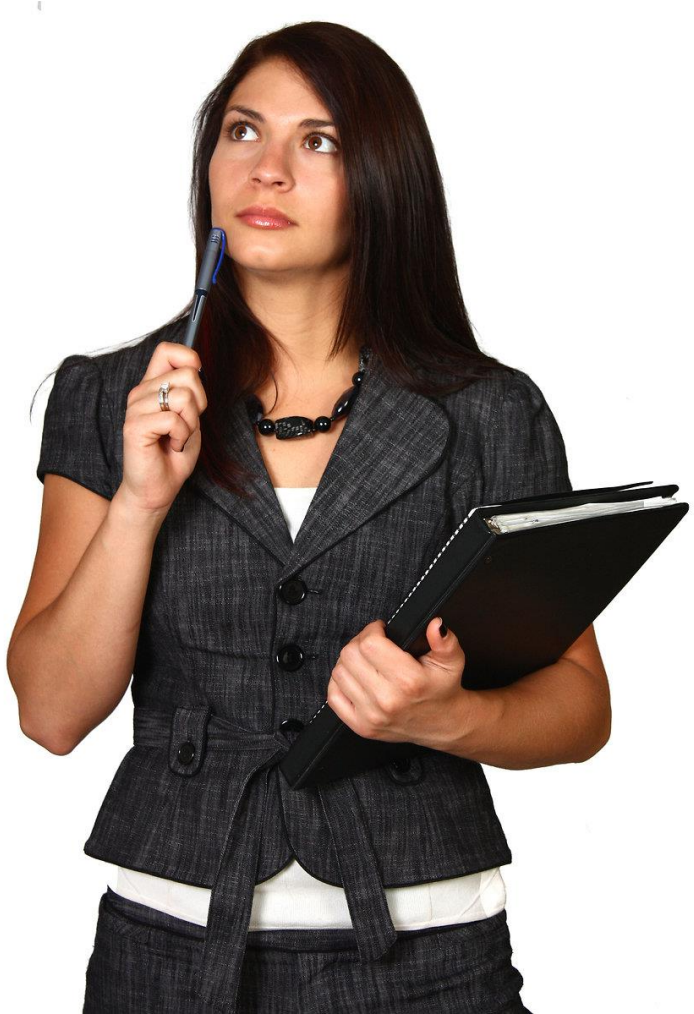
- **…even when you are an expert on the topic**
  - Example: Real estate agents vs. students estimating property values
- **…even when it is obviously wrong**
  - Example: “Did Gandhi get 144 years old? No? How old did he get then?”
- **…even when it is completely unrelated**
  - Example: Spinning a wheel of fortune before estimating

- The anchoring effect has an average impact of about 55%
  - (0%: a person picks the right answer; 100%: a person picks the anchor)
- The effect is impossible to suppress!
  - We can only try to prevent or neutralize it, e.g. through independent estimates

Daniel Kahneman: Thinking, Fast and Slow.
Farrar, Straus and Giroux, 2013

UNIVERSITY OF ICELAND

Effort Estimation



see also:

Head First Software Development, 2nd half of Chapter 2

Typical Software Effort Estimation in Practice

- Usually based on vague recollections of a previous similar project...
  - Guessing, intuition
  - Expert opinions
  - Mismatched analogies
- …combined with unfounded optimism
  - “We’ll be more productive this time.”
  - “The steepest part of the learning curve is behind us.”
  - “We won’t make the same mistake twice.”
  - “We now have a tool / existing components for this.”
  - “We now understand the application domain.”
- …and methodical errors
  - Incorrect conversion between time (days) and effort (person-days)
  - Incorrect derivation of efforts from size and complexity estimates
  - Tweaking estimation parameters until the estimate fits goals / expectations
  - Other stakeholders interpret estimates as precise commitments

Expectation Mismatch

- Clients and developers can have very different ideas of how hard it is to implement a particular feature.

Estimating User Story Efforts

As a passenger, I want to submit a review for a flight I have been on, in order to express my commendation or frustration to someone.

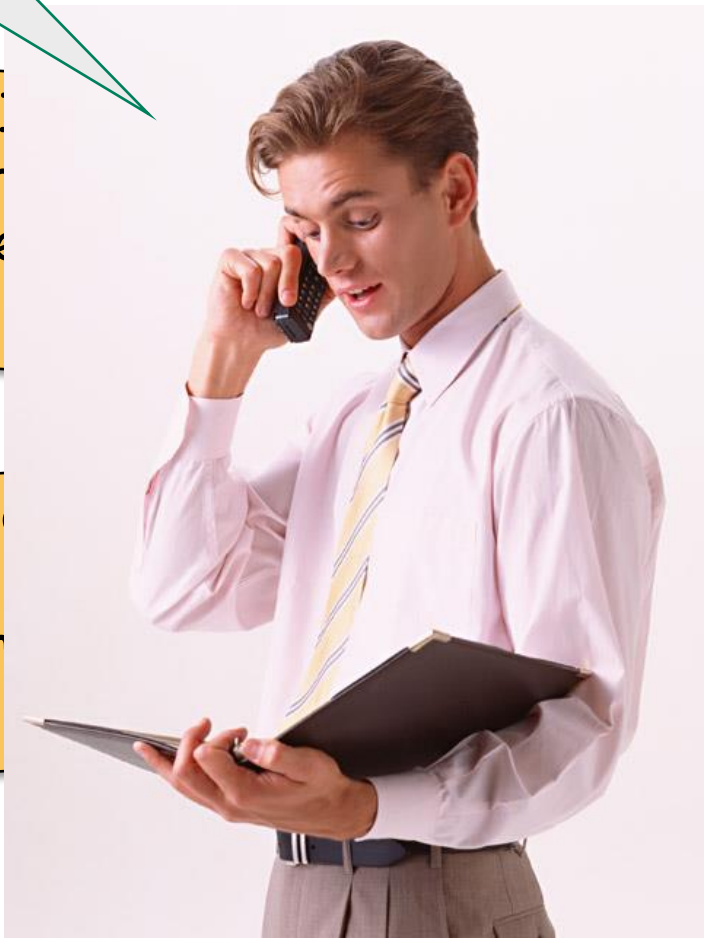
As a passenger, I want to choose whether my review is public or private, so I can provide more candid details if necessary.

As a marketing manager, I want to reply to reviews in public or private, in order to address any mentioned concerns.

As a webmaster, I reviews before th website, to ensure spam.

As a website visit reviews I read, in visitors to find th reviews.

So, can you get this done by Easter?

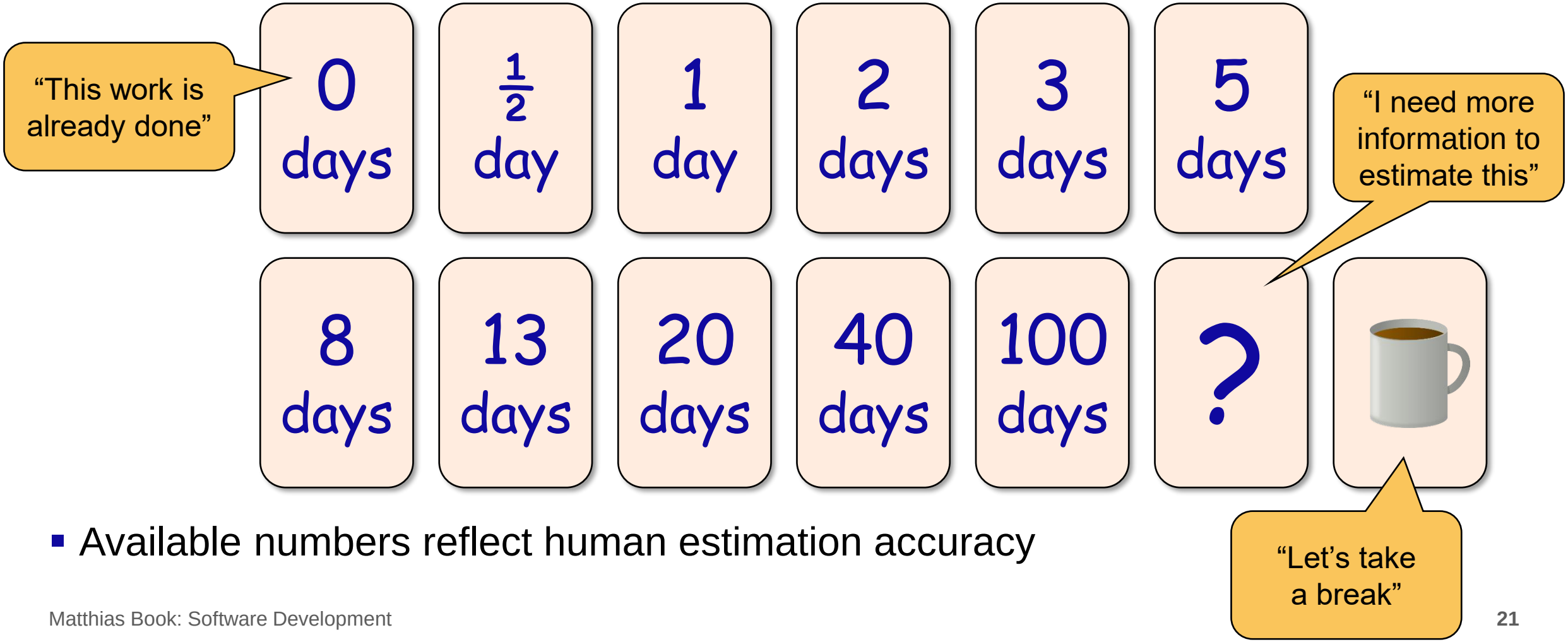


Estimation Approaches

- **Naïve:** Team members try to come up with estimates in joint discussion
  - Problems:
    - Social factors (dominant experts, personal image, …)
    - Subconscious bias (anchors, prior experience, …)
    - Converging opinions (uncorrected assumptions, unfounded confidence, routine, …)
  - ➢ Hard to form independent, objective estimates
- **Analytical:** Formal estimation techniques
  - Problems:
    - Require representative historical data
    - Risk of mismatching project characteristics
    - Suggest false precision
  - ➢ Takes a lot of effort, easy to get wrong
- **Agile:** Pragmatic, joint decision-making based on independent estimates

Planning Poker: The Deck

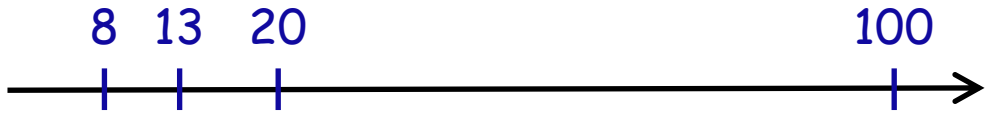
- Every team member receives a hand of 13 cards:



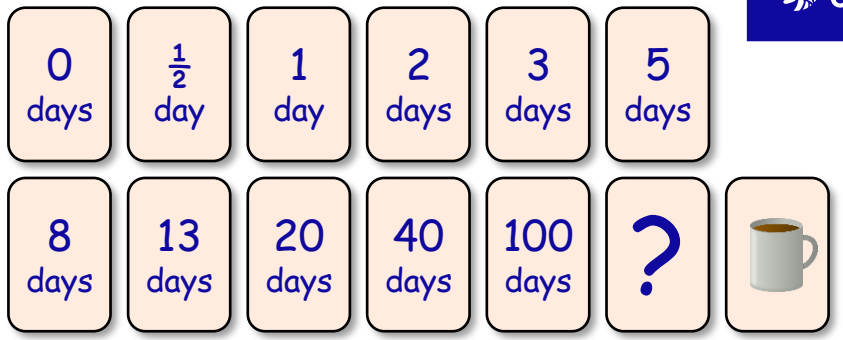
- Available numbers reflect human estimation accuracy

Planning Poker: The Game

1. Place a user story in the middle of the table.
2. Everyone picks an estimate for the story from their hand and places the card face-down on the table.
3. Everyone turns their card over at the same time.
4. Note and discuss the spread of the estimates.



Break / Discussion



- Why is there a gap between 40 and 100 days?
- 100 days seems really long – half a year in work time! Why have that card at all?
- What to do about people who always pick crazy numbers?

- Should we consider who will implement a user story when estimating?
- Should our estimates consider just implementation time?

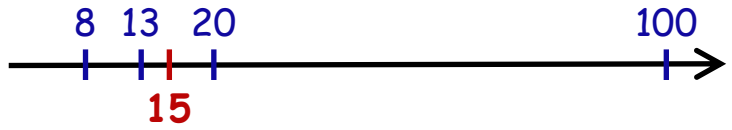
- What if our estimates differ wildly?
- What if my team members all put down exactly the same estimate? Can we be confident about the estimate then?

Eliminating Assumptions

- The **larger** the **difference** between the estimates, the **less confident** you are in the estimate, and the **more assumptions** you need to root out.
  - Large differences can be indications of misunderstandings, missing experience, etc.
    - No problem, as long as the team is aware of this and can rectify it early.
  - But small differences can also mean everyone made the same (wrong?) assumptions.
- Assume any assumption is incorrect (or imprecise) until confirmed by the client.
  - The only way of eliminating assumptions is to clarify them with the client.
  - The client may not be able to clarify some assumptions either.
- An assumption that cannot be resolved turns into a risk.
  - OK for assumptions (= risks) you are aware of – keep trying to resolve them along the way
  - Dangerous for assumptions (= risks) you are unaware of – they may hit you unexpectedly

Strive for Convergence of Estimates

1. Talk to the client
   - Understand the requirements
2. Play planning poker
   - Come up with your estimates
3. Clarify your assumptions with the client
   - This will help to adjust and build confidence in your estimates
4. Reach a consensus on the estimates
   - Agree on a final figure for the user story's estimate
     - This is usually not a simple average, but an informed decision
   - Write the agreed estimate down on the story card



As a website visitor, I want to rank reviews I read, in order to help other visitors to find the most helpful reviews. (15 days)

Breaking Down Large Estimates

- **A big user story estimate indicates a risk.**
- 40 days means about 2 months of work time
  - but an entire iteration is typically just 1 month long!
  - You could assign two people to the task
    - but that doesn't take the complexity and risk out of it
    - and you'll have additional overhead for coordination.
- Estimates over 15 days are usually much less likely to be accurate than lower estimates.

- **Break large estimates into smaller user stories.**
  - User stories with an "and" could possibly be broken in two.
- **Clarify underlying assumptions with the client.**
  - You might eliminate assumptions that raised your estimate.

Should Efforts Rather Be Over- or Underestimated?

Arguments against overestimation

- “Work will fill the available time.” (Parkinson’s Law)
- “Too much time invites procrastination (and not getting done).” (Goldratt Syndrome)
- “Some pressure doesn’t hurt.”
- “The schedule contains some buffer anyway.”
- “The software would be too expensive otherwise.”

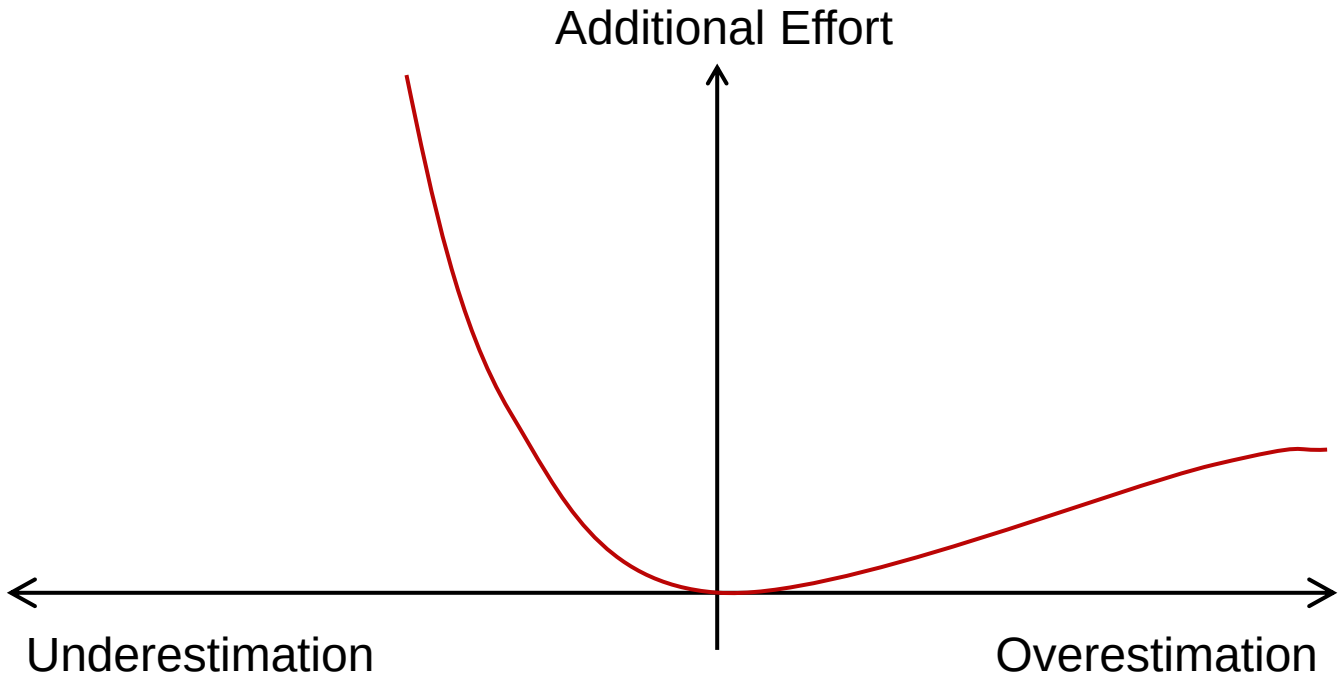
Arguments against underestimation

- Lower probability of meeting deadline
- Risky foundation for project planning
- Ignoring project complexity and necessary analysis work
- Things will get worse close to the end
  - More time for crisis meetings
  - Frequent reprioritization, re-estimation
    - ➢ Further planning uncertainty
  - Customer appeasement
  - Effort for intermediate demo releases
  - Problems caused by workarounds (“technical debt”)

Steve McConnell: Software Estimation. Microsoft Press, 2006

Should Efforts Rather Be Over- or Underestimated?

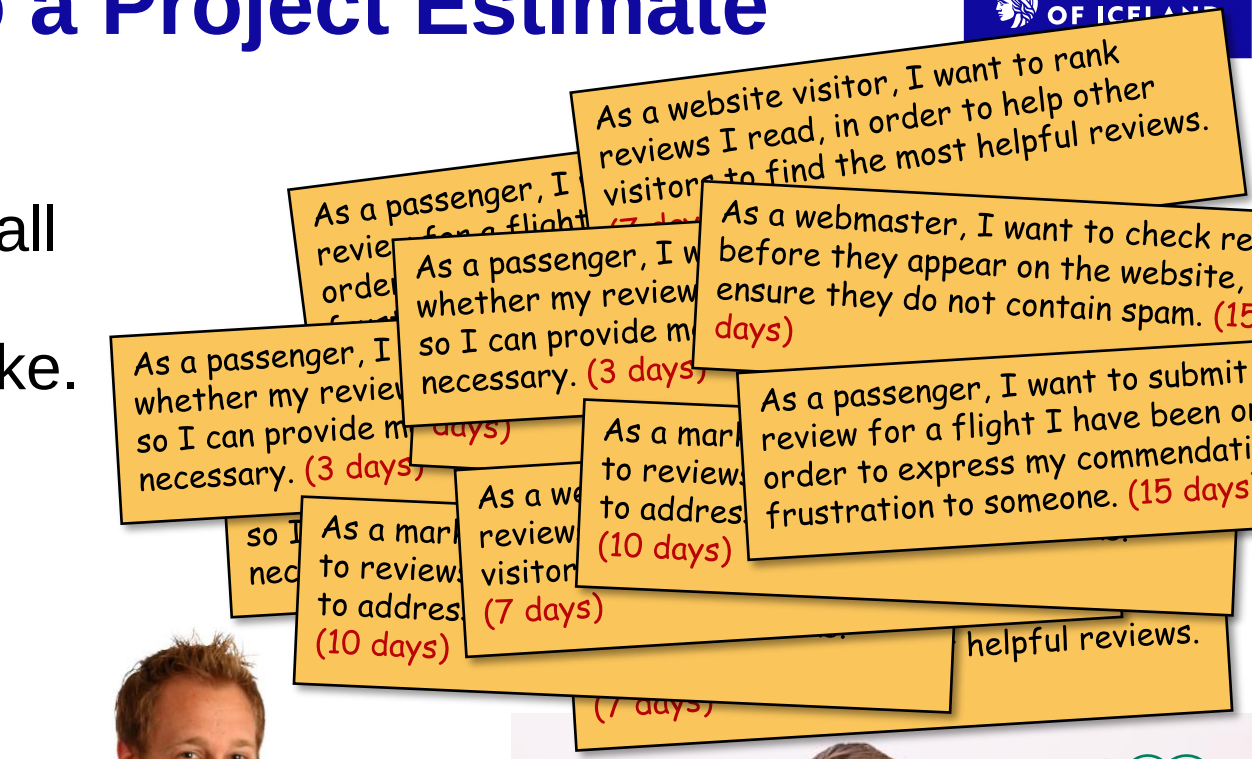
- Additional effort caused by underestimation is usually higher than that caused by overestimation.
- Additional effort caused by overestimation is limited by required work.
- Additional effort caused by overestimation can be handled through project management.



Steve McConnell: Software Estimation. Microsoft Press, 2006

From User Story Estimates to a Project Estimate

- Once you have agreed on estimates for all user stories, add them up to get a total estimate for how long your project will take.
- Developing everything the client wants typically takes much longer than the client is willing or able to wait.
- What can we do?
  - ~~Adjust our estimates~~
  - ~~Work overtime~~
  - Set priorities
  - Deliver features in multiple releases

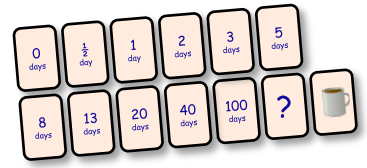


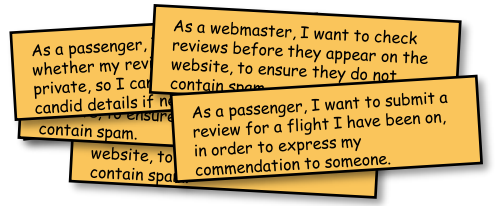
Preview: Assignment 1b: Effort Estimates

	A	B	C	D	E	F	G
1	**Team:**	[your team ID]			*Unit: person-hours*		
2							
3	**Sprint**	**User Story ID**	**User Story**	**Priority**	**Estimated Team Effort**	**Actual Team Effort**	**Done?**
4	...	...	...	...	...	...	...
5							

- **Sometime between Wed 2 and 9 Feb:**
  - Estimate effort for your user stories using planning poker
  - Prioritize user stories using card sorting
  - Select the stories you'll work on in your first two sprints

- **Please prepare for your planning meeting:**
  - For each team member: A deck of planning poker cards
    - Print and cut from template from Canvas
  - For each team: The list of stories you submitted for Assignment 1a
    - (possibly with revisions suggested in the Wed 3 Feb consultation)
  - For each team: A shared spreadsheet to record your estimates
    - Use the template provided in Canvas; fill columns B to E

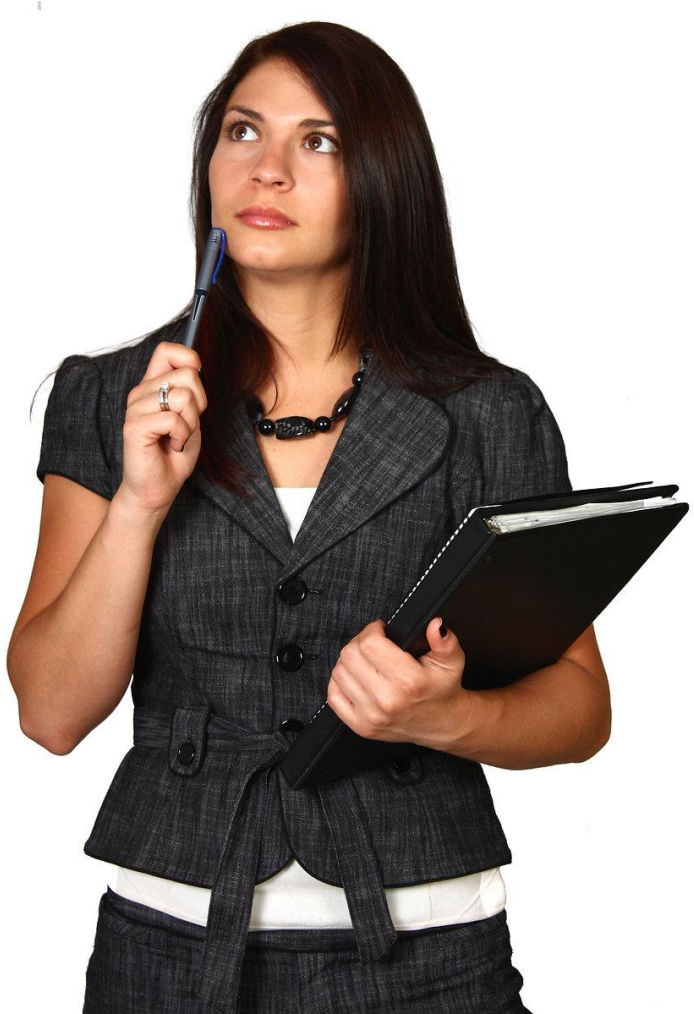




- Submit your product backlog spreadsheets **by Sun 13 Feb** in Canvas

UNIVERSITY OF ICELAND

Project Planning



see also:

Head First Software Development, Chapter 3

Person-Days vs. Calendar Days

- The user story estimates are **person-days**
  - i.e. the number of days one person would need to get the job done
  - This is a measure of *effort* (not time!) that is independent of team size, working days, etc.

- Your client cares about **calendar days**
  - i.e. the days you will need in actual time until you can deliver completed functionality
  - With one person working, the number of calendar days seems equal to the person-days…
  - …and in theory, two people could get the job done in ½ the calendar days, three people could get it done in ⅓, four in ¼, etc. *(we'll discuss the caveats in this soon)*

- However, people are only working on **work days**
  - i.e. no work occurs on weekends and holidays (or at least you shouldn't plan for it)
  - So the calendar time required to complete a certain number of person-days will be longer.

Person-Days vs. Calendar Days

As a marketing manager, I want to reply to reviews in public or private, in order to address any mentioned concerns.
(10 days)

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

- A task of 10 person-days effort
- requires 10 work days with 1 person
- so it is finished after 12 calendar days

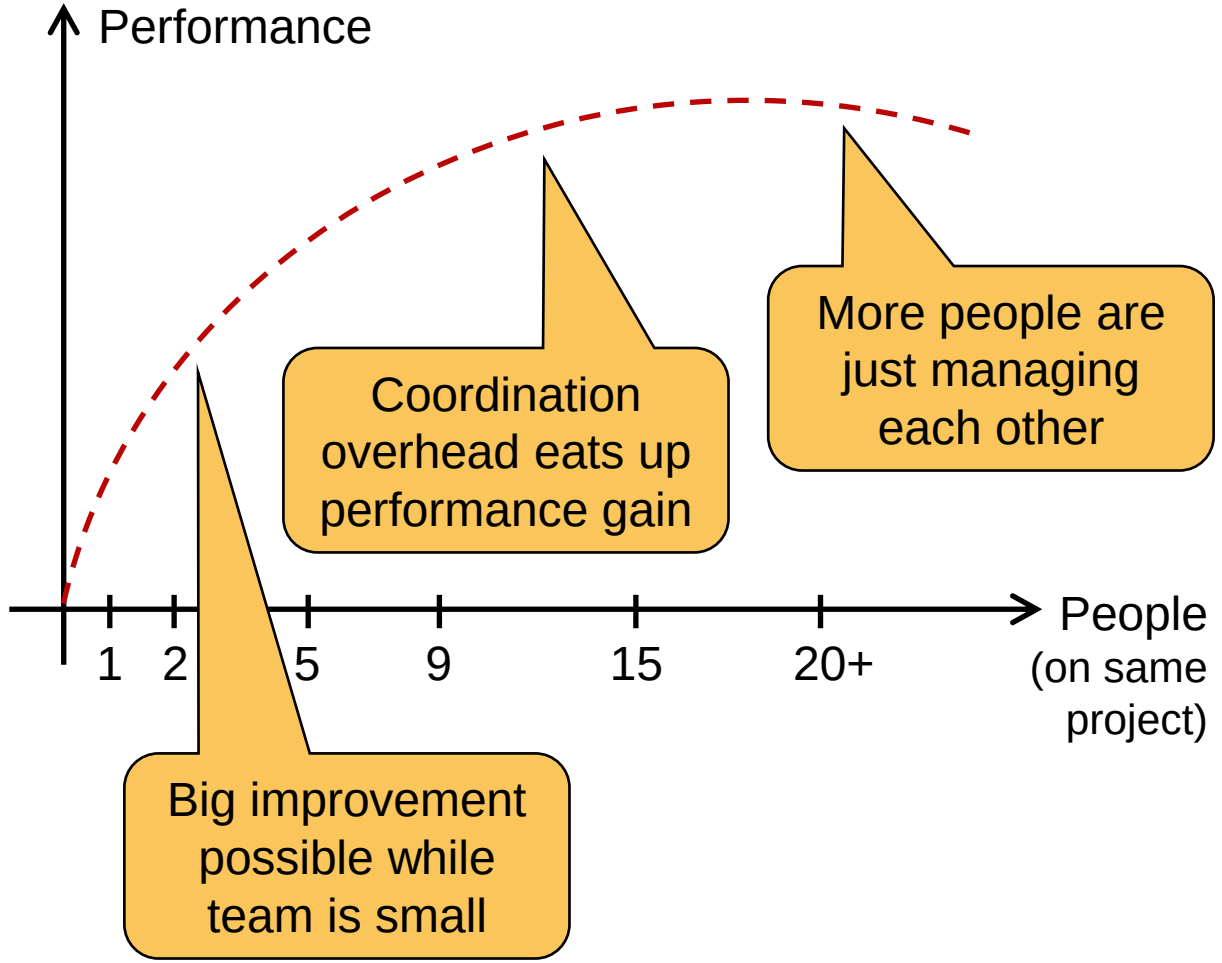
Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

- A task of 10 person-days effort
- requires 5 work days* with 2 people
- so it is finished after 5 calendar days*

* In theory! Realistically, probably 1-2 days more.

More People Produce Diminishing Returns

- Relationship between person-days and work days is *not linear* in practice
- New team members need to be trained on application domain, technologies, existing software, etc.
- Existing team members need to communicate in order to coordinate their work, agree on interfaces, resolve dependencies, etc.
- With more than a dozen people, consider creating sub-projects



Life Is What Happens While You're Making Other Plans

- Not every work day is 100% productive
- Foreseeable and unforeseeable events keep team members from working on the project
- Calendar time required for completion extends accordingly
- Schedule estimates are often too optimistic by 20-30%

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

- A task of 10 person-days effort
- requires 10 work days with 1 person
- who can't be productive on 3 days
- so it is finished after 17 calendar days

Quiz #3: Project Planning

- **Indicate how many working days the following tasks should take to complete, assuming 100% productivity:**

a) A task estimated to take 10 person-days, assigned to 2 people
b) A task estimated to take 10 person-days, assigned to 1 person
c) A task estimated to take 1 person-day, assigned to 2 people
d) A task estimated to take 3 person-days, assigned to 2 people

Thank you!

book@hi.is