Hugbúnaðarverkefni 2 / Software Project 2

4. Agile Estimation & Teams

HBV601G – Spring 2019

Matthias Book

HÁSKÓLI ÍSLANDS
VERKFRÆÐI- OG NÁTTÚRUVÍSINDASVIÐ
IÐNAÐARVERKFRÆÐI-, VÉLAVERKFRÆÐI-
OG TÖLVUNARFRÆÐIDEILD

In-Class Quiz Prep

- Please prepare a small scrap of paper with the following information:

ID: _____@hi.is   Date: _________

a) _____   d) _____

b) _____   e) _____

c) _____

- During class, I'll show you questions that you can answer very briefly
  - Just numbers or letters, no elaboration
- Hand in your scrap at the end of class
- All questions in a quiz weigh same
- All quizzes (ca. 10-12 throughout semester) have the same weight
  - Your worst 2 quizzes will be disregarded
- Overall quiz grade counts as optional question worth 7.5% on final exam

Assignment 1: Project Plan and Requirements

Deadline Extension!

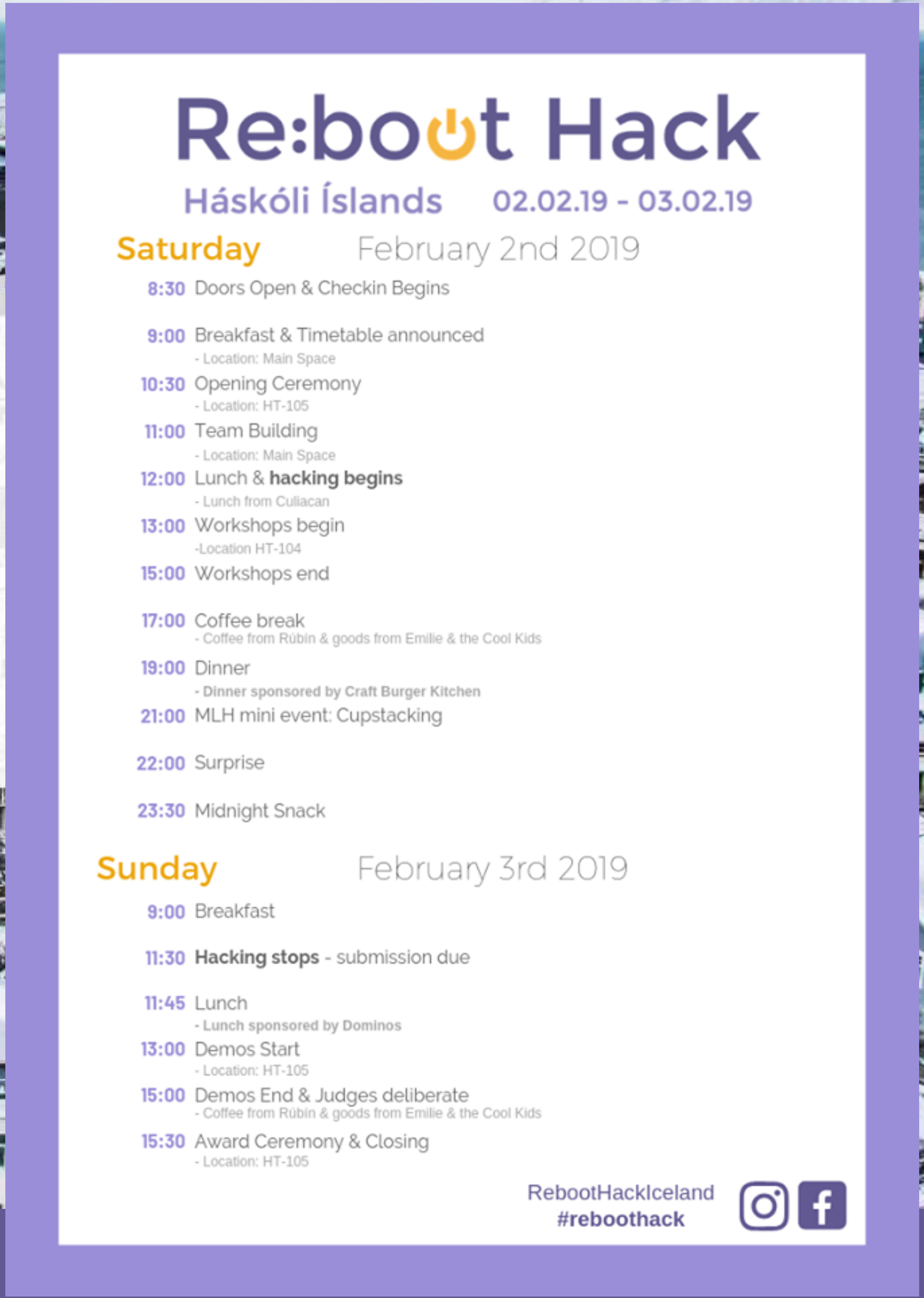
- By **Mon 4 Feb**, submit a **project outline** in Ugla, containing:
  - Project vision (in form of a RUP Vision document, Data Sheet or Press Release)
  - Product backlog (prioritized user stories)
  - User Story estimates (three-point estimates of expected cases, using basic PERT formula)
  - Project schedule (dates for sprints, milestones, assignments 2 and 3)

- On **Thu 7 Feb**, present and **explain** your project outline to your tutor:
  - What considerations influenced your choice of scope, estimates and schedule?

- **Grading criteria:**
  - Project vision is clear and plausible (15% of this assignment's grade)
  - Product backlog describes requirements clearly and with realistic scope (40%)
  - Project estimates calculated using appropriate formulae (35%)
  - Project schedule is clear, realistic, and specifies dates for assignments 2 and 3 (10%)

Re:boot Hack

Reykjavík 02.02.19 - 03.02.19



Re:boot Hack

Háskóli Íslands 02.02.19 - 03.02.19

Saturday February 2nd 2019

8:30 Doors Open & Checkin Begins

9:00 Breakfast & Timetable announced
- Location: Main Space

10:30 Opening Ceremony
- Location: HT-105

11:00 Team Building
- Location: Main Space

12:00 Lunch & **hacking begins**
- Lunch from Culiacan

13:00 Workshops begin
-Location HT-104

15:00 Workshops end

17:00 Coffee break
- Coffee from Rúbín & goods from Emilie & the Cool Kids

19:00 Dinner
- Dinner sponsored by Craft Burger Kitchen

21:00 MLH mini event: Cupstacking

22:00 Surprise

23:30 Midnight Snack

Sunday February 3rd 2019

9:00 Breakfast

11:30 **Hacking stops** - submission due

11:45 Lunch
- Lunch sponsored by Dominos

13:00 Demos Start
- Location: HT-105

15:00 Demos End & Judges deliberate
- Coffee from Rúbín & goods from Emilie & the Cool Kids

15:30 Award Ceremony & Closing
- Location: HT-105

RebootHackIceland
#reboothack

reboot.hi.is

Software Estimation

(continued)

see also:

- McConnell: Software Estimation, Ch. 1, 3-5, 7-13
- Leffingwell: Agile Software Requirements, Ch. 8

In-Class Quiz #2 Solution

1.a) Counting defects late in project

- For the last 250 defects we fixed, we needed 2 hours per defect on average.
- 400 defects are currently open in our project.

➢ It should take us approximately 400 * 2 = 800 hours to fix the open defects.

1.b) Counting web pages

- So far, each dynamic web page in the project cost us a total of 40 hours to design, code and test, on average.
- We have 10 web pages left to build.

➢ It should take us approximately 10 * 40 = 400 hours to build those pages.

Recap: Estimation Approaches Discussed in This Course

Motivation:

- The aim of effort estimates is **not to predict** the project result…
- …but to judge whether the project target is **sufficiently realistic**..
- …to be achievable through corresponding project management.

A. Projection from counts
   1. Concrete items
   2. Function points

B. Individual expert judgment

cont'd now

C. Group expert judgment
   1. Wideband Delphi
   2. Planning Poker

D. Qualitative estimation

Recap: Calculating Expected Cases

Weighted average, e.g.
ExpectedCase = (BestCase +
4 * MostLikelyCase + WorstCase) / 6

Feature	Best Case	Most Likely Case	Worst Case	Expected Case
1	1.25	1.5	2.0	1.54
2	1.5	1.75	2.5	1.83
3	2.0	2.25	3.0	2.33
4	0.75	1	2.0	1.13
5	0.5	0.75	1.25	0.79
6	0.25	0.5	0.5	0.46
7	1.5	2	2.5	2.00
8	1.0	1.25	1.5	1.25
9	0.5	0.75	1.0	0.75
10	1.25	1.5	2.0	1.54
Total				**13.62**

[Days to Complete]

Dangers of Adding Up Best / Worst Case Estimates

- Suppose we had the following estimates of weeks to complete 10 features:

Feature	Best Case	Most Likely Case	Worst Case	Expected Case
…	…	…	…	…
Total	~~20.0~~	~~28.3~~	~~38.6~~	**28.62**

- Adding up the individual estimates to obtain e.g. a “total best case” is pointless!
  - Suppose the probability of achieving any one of the best cases is 25% (i.e. ¼).
  - The probability of achieving any two of the best cases then is ¼ * ¼ = 6,25%.
  - The probability of achieving the best case for all features is ¼^{10} = 0.000095%.
  - **This makes achieving the “total best case” completely unrealistic!**

- **Caution:** Since single-point estimates are typically subconscious best-case estimates, adding those up to obtain a total estimate is particularly misleading!

B_1) Computing Probability Ranges for ≤10 Individual Estimates

1. Calculate standard deviation of the estimate totals:
   - StandardDeviation = (∑ WorstCaseEstimates – ∑ BestCaseEstimates) / 6
2. Decide how confident your estimate needs to be
3. Pick according factor from table on the right, and calculate
   - PercentConfidentEstimate = ExpectedCase + Factor * StandardDeviation

- Example:

Feature	Best Case	Most Likely Case	Worst Case	Expected Case
…	…	…	…	…
Total	20.0		38.6	**28.62**

[weeks]

- Standard deviation = (38.6 – 20.0) / 6 = 3.1
- ➢ 25% likely estimate: 28.62 – 0.67 * 3.1 ≈ 27 weeks
- ➢ 75% likely estimate: 28.62 + 0.67 * 3.1 ≈ 31 weeks

These are *estimates* – don't kid yourself with unwarranted precision!

Confid.	Factor
2%	–2
10%	–1.28
16%	–1
20%	–0.84
25%	–0.67
30%	–0.52
40%	–0.25
50%	0
60%	0.25
70%	0.52
75%	0.67
80%	0.84
84%	1
90%	1.28
98%	2

Computing Probability Ranges

- Problem: Statistics of previous approach assume that 99.7% of actual results fall into estimated range (between best and worse case).
  - i.e. only 3 out of 1000 estimates could fall outside estimated range!
- Obviously completely unrealistic
  - Most people are actually about 30% sure when they think they are 90% sure
  - With practice, about 70% can be achieved

- For ≤10 estimated items, the impact of this error is negligible, so we can still use the previous simple approach.
- For >10 estimated items, the impact becomes more pronounced, so we need to use a somewhat more complex approach to calculate the standard deviation:

B_2) Computing Probability Ranges for >10 Individual Estimates

1. Check how many of your actual outcomes fall within your estimation range
2. Determine according divisor in table on the right
3. Calculate the variance of *each individual* estimate:
   - Variance = ((WorstCaseEstimate – BestCaseEstimate) / Divisor)^2
4. Calculate total of the individual variances
5. Take the total's square root to obtain the standard deviation
6. Continue with step 2 of previous approach

% in range	Divisor
10%	0.25
20%	0.51
30%	0.77
40%	1
50%	1.4
60%	1.7
68%	2
70%	2.1
80%	2.6
90%	3.3
99.7%	6

Example: Percentage Confident Estimate

- Based on data we saw earlier, this team's results fall into the estimated range 80% of the time
  - ➢ Use divisor 2.6 to calculate variances of individual estimates
- Standard deviation:
  - sqrt(6.48) = 2.55

➢ 25% likely estimate:

- 28.62 – 0.67 * 2.55 ≈ 27 weeks

➢ 75% likely estimate:

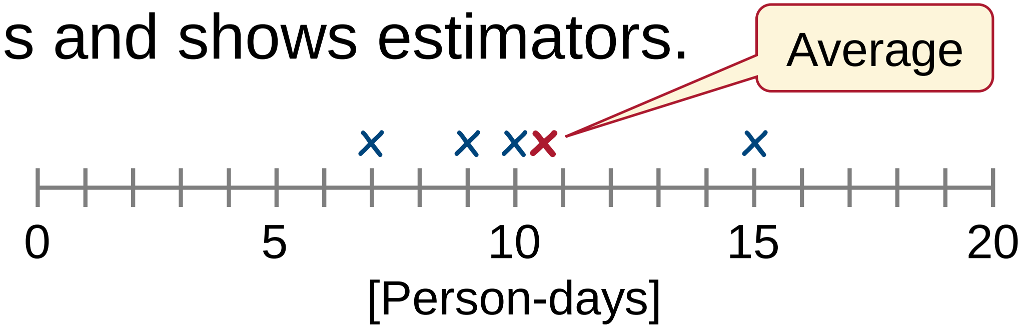
- 28.62 + 0.67 * 2.55 ≈ 30 weeks

Feature	Best Case	Most Likely Case	Worst Case	Expected Case	Variance
1	1.6	2.0	3.0	2.10	0.290
2	1.8	2.5	4.0	2.63	0.716
3	2.0	3.0	4.2	3.03	0.716
4	0.8	1.2	1.6	1.20	0.095
5	3.8	4.5	5.2	4.50	0.290
6	2.2	5.0	3.4	4.97	0.716
7	2.2	2.4	3.4	2.53	0.213
8	0.8	1.2	2.2	1.30	0.290
9	1.6	2.5	3.0	2.43	0.290
10	1.6	4.0	6.0	3.93	2.864
Total				**28.62**	**6.48**

[weeks to complete]

C_1) Group Expert Judgment: Wideband Delphi

1. Delphi coordinator presents each estimator with specification, estimation form
2. Estimators prepare initial estimates for features individually.
3. Coordinator calls group meeting in which estimators discuss estimation issues.
4. Estimators give their individual estimates to coordinator anonymously.
5. Coordinator prepares summary of estimates and shows estimators.

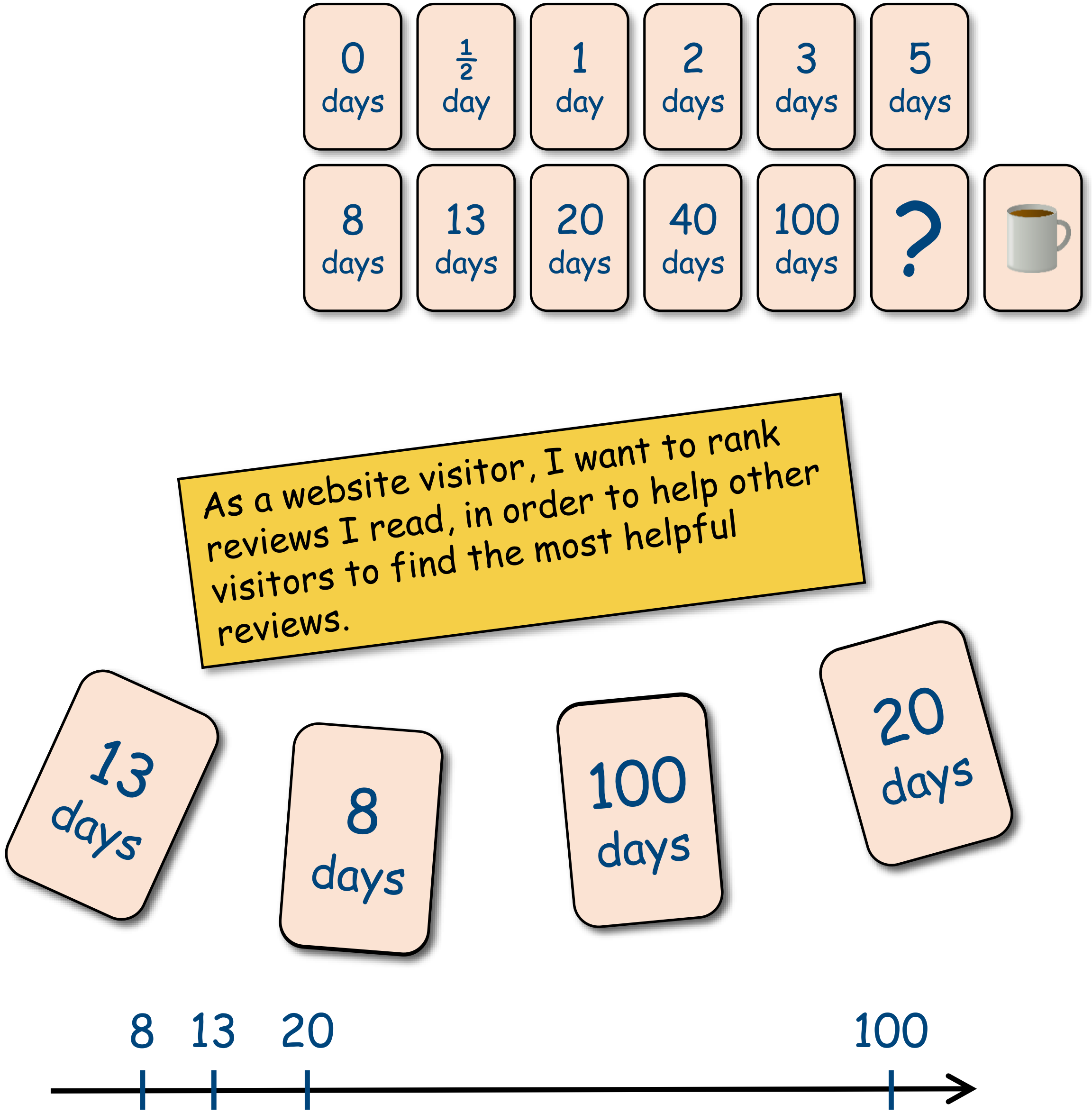


6. Coordinator lets estimators discuss variations in estimates.
7. Estimators vote anonymously on whether to accept average or return to #3.

➢**Final estimate is a *range* created in discussion, average is expected case.**

C_2) Group Expert Judgment: Planning Poker (Recap)

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.
5. Decide on an estimate that the whole team can agree on.
   - Not necessarily the average!



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

(see HBV401G, Lecture 4 for details)

What to Estimate in Planning Poker?

Story Points

- Synthetic measure
- Only indicates relative complexity (and therefore relative effort) of user stories, but does not directly translate into absolute effort
  - Translation into absolute effort is implicit as team executes sprint and achieves a certain velocity
- Estimate focuses on complexity and is kept free of real-time considerations
  - more objective, but hard to communicate

Person-Days

- Describes an actual property
  - Easier to handle by teams who are new to agile approach
  - easier to communicate to management
- Estimates may be politically biased
  - Developers may not dare to estimate high so as not to appear “slow”

- **Suggestion:** Use story points, but have informal convention that 1 story point corresponds to 1 person-day

D) Qualitative Estimation: T-Shirt Sizing

- Chicken and egg problem in agile projects:
  - Team members shouldn’t spend effort on understanding / refining a feature’s requirements until they are reasonably sure it is actually going to be built
  - Management doesn’t want to decide on whether a feature shall be built until they have some idea of the required effort

- Solution:
  - Initially use a highly intuitive, relative, coarse and fast estimation scheme
  - Classify features by Business Value and Development Cost
    - using “t-shirt sizes” (small, medium, large, extra large) as categories

- Example:

Feature	Business Value	Development Cost
A	Large	Small
B	Small	Large
C	Large	Large
…	…	…

HÁSKÓLI ÍSLANDS

Calculating Net Business Value Based on T-Shirt Sizes

- Discussion on which features to retain and which to cut is easier when feature list is sorted according to cost/benefit evaluation
- Use following matrix (or similar one of your own) to assign **net business value:**

Net Business Value		Development Cost			
		Extra Large	Large	Medium	Small
Business Value	Extra Large	0	4	6	7
	Large	–4	0	2	3
	Medium	–6	–2	0	1
	Small	–7	–3	–1	0

- Sort features by resulting net business value
- Identify “definite yes” and “definite no” features, discuss others in more detail

Example: Calculating Net Business Value

Definite YES area

Feature	Business Value	Development Cost	Net Business Value
A	L	S	3
F	L	M	2
C	L	L	0
D	M	M	0
G	S	S	0
I	S	S	0
H	S	M	–1
E	M	L	–2
…	…	…	…
B	S	L	–3

Up for discussion

Definite NO area

When to Use Which Approach?

- **Use rough qualitative estimation, e.g. T-Shirt Sizing (D)**
  - WHEN: Initial stages of a project
  - WHY: To make initial decisions on the general scope of project
  - BASIS: General vision of project, high-level business and technology constraints
  - ACCURACY: Quite reliable in definitive YES and NO areas, very fuzzy in the middle region
  - REFINEMENT: Refine included features in product and sprint backlog discussions (B, C)

- **Use projection from counts, e.g. average effort per [item] (A)**
  - WHEN: Early stages of a project
  - WHY: To get an initial rough idea of the overall size, effort and schedule of the project
  - BASIS: Reasonably complete overview of requirements, comparable historic data
  - ACCURACY: Highly dependent on accuracy of data, will be deprecated by later fluctuations
  - REFINEMENT: Use more precise approaches for product / sprint backlog estimates (B, C)

When to Use Which Approach?

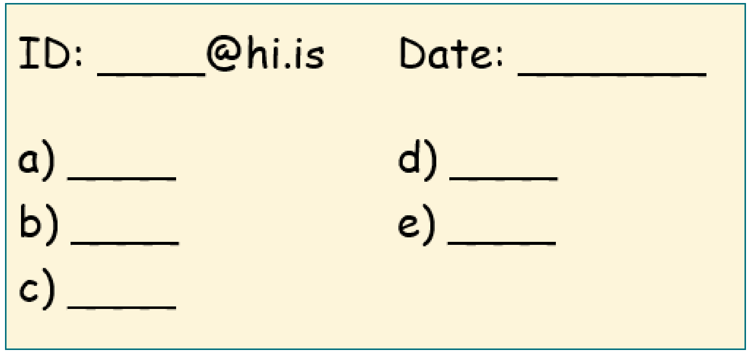
- **Use individual expert judgment, e.g. Three-Point Estimates (B)**
  - WHEN: Early stages of a project
  - WHY: To obtain initial estimates for product backlog items, based on different scenarios
  - BASIS: Discussion of requirements, experts' experience
  - ACCURACY: Higher than "gut feelings" that otherwise go into group expert estimate (C_1)
  - REFINEMENT: Individual sprint planning (C_2) will make more informed short-term estimates

- **Use elaborate group expert judgment, e.g. Wideband Delphi (C_1)**
  - WHEN: Early stages of a project
  - WHY: To obtain initial estimates for product backlog items, based on prior experience
  - BASIS: Discussion of requirements, experts' experience
  - ACCURACY: Reasonably high if experts have sufficient domain knowledge
  - REFINEMENT: Individual sprint planning (C_2) will make more informed short-term estimates

When to Use Which Approach?

- **Use pragmatic group expert judgment, e.g. Planning Poker (C_2)**
  - WHEN: Sprint planning meetings
  - WHY: To obtain detailed estimates for all sprint backlog items and commit to sprint contents
  - BASIS: Elaboration of and discussion of requirements, breakdown of tasks
  - ACCURACY: High w enough domain knowledge, performance experience, breakdown detail
  - REFINEMENT: Tracking velocity helps team to calibrate estimates in subsequent sprints

In-Class Quiz #3: Estimation Methods



ID: ____@hi.is Date: ________

a) ____ d) ____

b) ____ e) ____

c) ____

- **Indicate which approach you would use in the following situations:**

a) Establishing a rough project scope

b) Establishing first release date (new project, new domain)

c) Establishing next release date (project you've been on for a year)

d) Getting an idea of whether you can still complete a user story before end of sprint

e) Sprint planning

- **Possible choices:**

- (A) Projection from counts
- (B) Three-Point Estimates
- (C_1) Wideband Delphi
- (C_2) Planning Poker
- (D) T-Shirt Sizing

Advantages of Precise Estimates

- Confirmation that project is on track
- Higher software quality due to reduced stress
- Better alignment with plans of other business units
- Higher financial safety
- Reference values for later projects
- More competent image of developer team

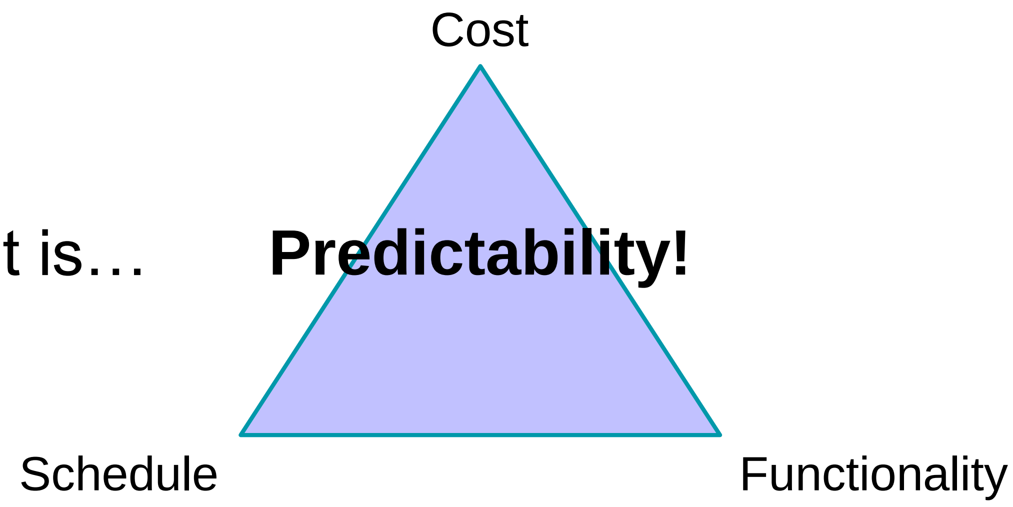
- Early risk indicators when target and estimate diverge
  - i.e. early chance to remedy the situation…
    - Changed project scope
    - More staff
    - Use of experts
    - Incremental deliverables
    - Decision against project
  - …instead of ignoring/negotiating the estimate and accepting the risk of
    - Schedule or budget overruns
    - Forced functionality reductions

Project Qualities Expected by Customers

- Every customer prefers a project that is
  - on schedule
  - within budget
  - fulfilling expectations

- If you need to compromise on any of these,
  - let the customer know early what to expect
  - and work on an acceptable solution together

➢ What customers value most is… **Predictability!**



Practicing Estimation and Controlling Techniques

- Try to re-estimate your project based on a more refined list of user stories, using one of the methods presented in class
  - Check how the results of different estimation methods compare
  - Gain experience with the estimation process
- Throughout the project, record your working hours so you can compare them to your estimates later
  - Gain experience with how long you need for certain tasks
  - Get a feeling for how your actual performance differs from your estimated performance
- Throughout the project, maintain a burndown chart recording your progress through the sprints
  - Gain experience with velocity tracking
  - Get a feeling for how much you can commit to, and what overhead you need to consider

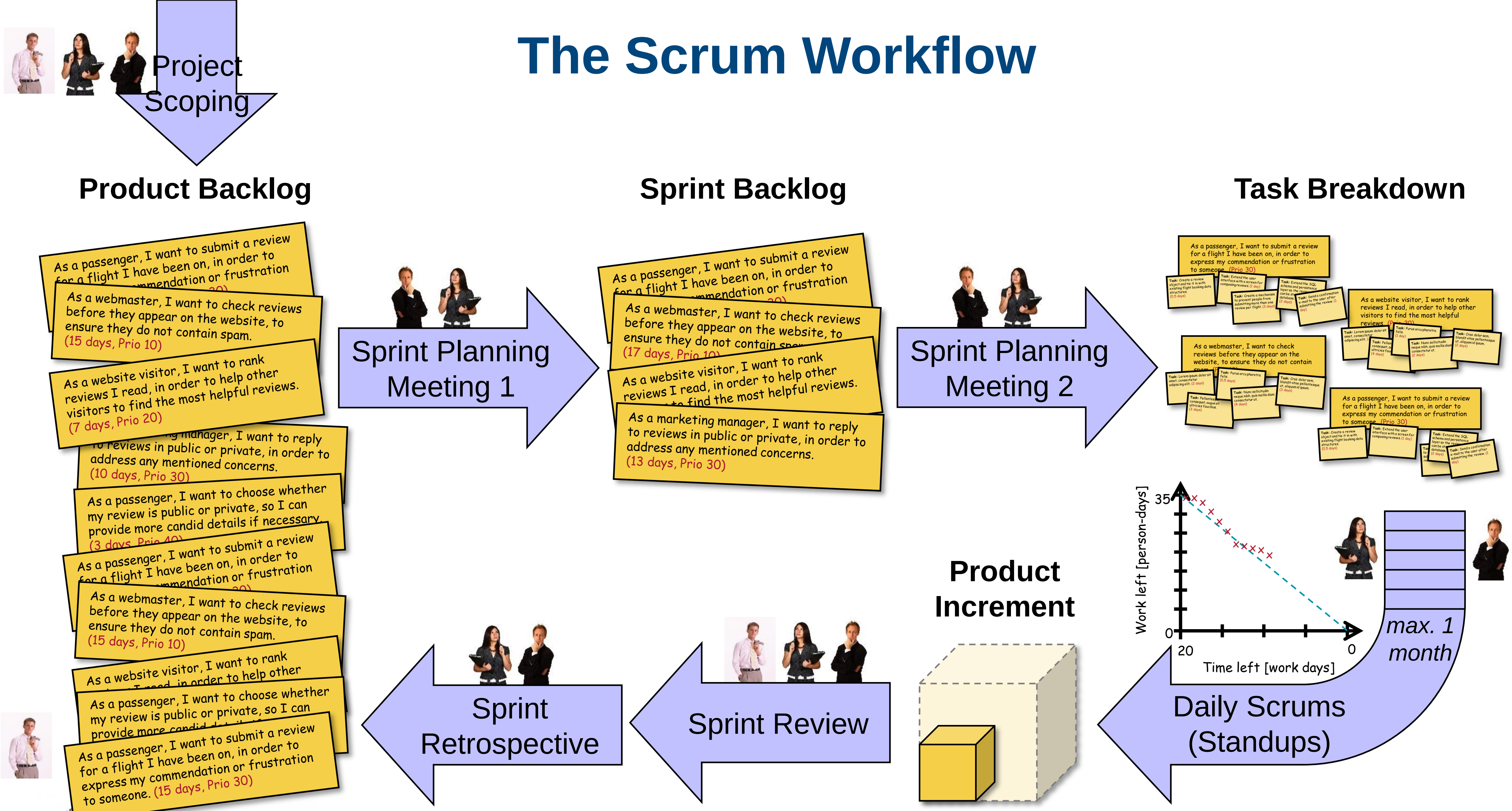
➢ **In your final presentation (Ass. 4), present and discuss estimated vs. actual efforts and burndown chart as part of your project retrospective**
  - **Your quantitative performance will not be graded, but your critical reflection will be**

Agile Teams

see also:

Leffingwell: Agile Software Requirements, Ch. 3, 11

The Scrum Workflow



Project Scoping

Product Backlog

Sprint Planning Meeting 1

Sprint Backlog

Sprint Planning Meeting 2

Task Breakdown

Daily Scrums (Standups)

max. 1 month

Product Increment

Sprint Review

Sprint Retrospective

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. (15 days, Prio 30)
As a webmaster, I want to check reviews before they appear on the website, to ensure they do not contain spam. (15 days, Prio 10)
As a website visitor, I want to rank reviews I read, in order to help other visitors to find the most helpful reviews. (7 days, Prio 20)
As a marketing manager, I want to reply to reviews in public or private, in order to address any mentioned concerns. (10 days, Prio 30)
As a passenger, I want to choose whether my review is public or private, so I can provide more candid details if necessary. (3 days, Prio 40)
As a webmaster, I want to check reviews before they appear on the website, to ensure they do not contain spam. (17 days, Prio 10)
As a marketing manager, I want to reply to reviews in public or private, in order to address any mentioned concerns. (13 days, Prio 30)
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. (Prio 30)
As a website visitor, I want to rank reviews I read, in order to help other visitors to find the most helpful reviews. (Prio 20)
As a webmaster, I want to check reviews before they appear on the website, to ensure they do not contain spam.
Task: Create a review object and tie it in with existing flight booking data structures. (0.5 days)
Task: Extend the user interface with a screen for composing reviews. (1 day)
Task: Extend the SQL schema and persistence layer so the review can be stored in the database. (2 days)
Task: Create a mechanism to prevent people from submitting more than one review per flight. (3 days)
Task: Send a confirmation e-mail to the user after submitting the review. (1 day)
Work left [person-days]
35
0
20
0
Time left [work days]

(see HBV401G, Lecture 4 for details)

Roles in an Agile Team

- **Product Owner**
  - Works with product managers, business analysts, customers and other stakeholders to determine requirements
  - Maintains the backlog and sets priorities based on relative user value
  - Sets objectives for iteration
  - Elaborates stories, participates in progress reviews, accepts new stories

- **Scrum Master**
  - Facilitates team’s progress towards goal
  - Leads team’s efforts in continuous improvement
  - Enforces rules of agile process
  - Eliminates impediments
  - Moderates daily standup meeting

Roles in an Agile Team

- **Developers**
  - Collaborate with product owner and testers to make sure the right code is being developed
  - Write code
  - Write and execute unit tests for code
  - Write methods as necessary to support automated acceptance tests and other testing automation
- **Testers**
  - Collaborate with product owner and developers to make sure a story is understood, and that acceptance tests track the desired functionality of the story
  - Write acceptance test cases while code is being written
  - Test code against acceptance tests
  - Develop ongoing test automation to integrate acceptance and component tests into continuous testing environment

Roles in an Agile Team

- **Other roles** (possibly part-time only):
  - Enterprise or software architects
  - Quality assurance managers
  - User experience designers
  - Database designers
  - Deployment specialists
  - Domain experts
- **Typical variations:**
  - Scrum Master may be external coach or part-time role
    - Experienced teams may not even have it
  - Developer and tester roles may be assumed by same people

- **Ideal agile team characteristics**
  - 7±2 members
  - Co-located
  - Dedicated
  - Focused
  - Cross-functional
  - Self-organizing

Product Owner Responsibilities

- Managing the backlog
- Performing just-in-time story elaboration
- Planning iterations
- Reviewing and accepting completed stories
- Co-planning the release
- Collaborating with product management

➢ **Agile methods do not have a dedicated requirements phase, but they do have a dedicated requirements engineer.**

  - Arguably the most critical job on the team.
  - You want someone who understands the business domain and technology equally well, and who can communicate with and broker between stakeholders of both sides.

Attributes of a Good Product Owner

- **Communication skills**
  - Bridge between development team and all other project stakeholders, esp. the customer
  - Translating user and business objectives into suitable detail level for implementation
- **Good business sense**
  - Understand which value user stories deliver to end users, and establish according priorities
  - Make decisions and trade-offs balancing customers’ and business’ needs
- **Technical foundation**
  - Understand capabilities and challenges of different technologies
  - Intelligently prioritize features, refactors, defects, technical debt vs. user value stories
- **Decisiveness**
  - Ability to make business and technical decisions every day based on incomplete knowledge
  - Be available to make those decisions, and be empowered by management to make them
- **Trustworthiness**
  - Establish trust with business side that requirements will be communicated faithfully to team
  - Establish trust with developers that scope and schedule decisions will be defended to mgmt.

Product Owner Anti-Patterns: What Won’t Work

- **Several product owners**
  - Tedious decision-making between product owners (possibly pursuing different goals)
  - Developers getting inconsistent answers from different product owners
- **Part-time product owner**
  - Delays in user story elaboration lead to delays (or worse, guessing) in implementation
  - Delays in user story acceptance lead to delays in deployment
- **Product manager as product owner**
  - A traditional “product manager” is a more business- and market-oriented role
  - Rarely has the technical skill or motivation to advise developer team all the time

➢ **Each team should have exactly one product owner**

- If your organization already has business-oriented *product managers* working with clients, pair them up with technology-oriented *product owners* working with the development team

Product Owner vs. Product Manager

Product Owner

- New role introduced by agile processes
- Product/technology-facing
- Co-located with / reports to development / technology department
- Focuses on product and implementation technology
- Owns the implementation
- Drives the iterations

Product Manager

- Classic role already existing in many large organizations
- Market/customer-facing
- Co-located with / reports to marketing / business department
- Focuses on market segments, portfolio, return on investment
- Owns the vision and roadmap
- Drives the release