Lecture 21 — Planning

Patrick Lam
p.lam@ece.uwaterloo.ca

Department of Electrical and Computer Engineering
University of Waterloo

December 29, 2014

Planning, scheduling, and estimation

Planning and scheduling are notoriously difficult, because estimates are often wrong.

- **Planning.** Given a project, figure out how to do the project. Figure out subtasks.
- **Scheduling.** Assign start and end times to subtasks. More mechanical.
- **Estimation.** Informed guess (based on facts) about resources needed to accomplish a task.
  Used in scheduling.

Part I

Planning

Documents

- **Vision and Scope Document**:
  preliminary document.
- **Project Plan**:
  contains all the details.

Vision and Scope Document

Preliminary requirements gathering:
make sure that all stakeholders are on board.

A **vision and scope document** summarizes the design problem, the stakeholders, the users, the risks, the assumptions, and the desired features of a solution.

Goals of a Vision and Scope Document

A vision and scope document:

- identifies included features;
- discusses features that are out-of-scope;
- summarizes agreed-upon expectations about scope.

Vision and Scope Document Outline

Mousetrap Example

- Problem Statement
- Project Background
- Stakeholders
- Users
- Risks
- Assumptions
- Vision of the Solution
- List of features
- Scope of phased release
- Features not to be developed

Project Plan

The project plan breaks down
what to do and who is to do it.

- Statement of Work: what to do and who will do it.
- Resource List: who can do it, and with what.
- Work Breakdown Structure: what to do (more detail).
- Project Schedule: when to do it.
- Risk Plan: what might go wrong.

We'll look at each part of the project plan.

Statement of Work

A **statement of work** (SOW) describes all work products to be produced and identifies who will do the work, including:

- a list of all project features that will be developed;
- a description of each deliverable that will be developed (at this stage, one paragraph per deliverable); and
- the estimated effort required for each deliverable, if known.

Resource List

- **Resource List**: enumerates all resources required for the project and summarizes the availability of the resources.
- **Resource**: person, hardware component, software licence, room, or anything else necessary for the project but limited in availability.

Components of Resource List

The resource list has the following components:

- resource name;
- brief (one-line) description of the resource;
- resource availability (e.g. start dates, end dates);
- cost of the resource (if applicable).

Work Breakdown Structure (WBS)

A **work breakdown structure** (WBS) describes all project activities:

- comprehensively lists all project activities required to complete the project;
  (includes intermediate deliverables.)
- must evolve with your understanding of the project;
- should be kept in a revision control repository.

Applied Software Project Management describes the "Wideband Delphi Process" for estimating the time required for each activity in a WBS.

Project Schedule

A **project schedule** estimates start & end times for project activities.

One way of getting estimates: run a meeting.

1. Each participant writes down an initial estimate plus assumptions.
2. Moderator collects all estimates and summarizes them.
3. Participants discuss the estimates.
4. Repeat as needed.

Documenting assumptions helps refine estimates and reduce variance in estimates.

Risk Plan

A **risk** is a potential threat to the successful completion of your project.

A **risk plan** describes:

- anticipated risks;
- the likelihood of their occurrence; and
- how to mitigate these risks.

Good risk plans are comprehensive and include potential ways to mitigate significant risks.

Developing a Risk Plan

The risk plan is usually developed in a risk assessment and planning session. Team members:

- brainstorm to list potential risks;
- estimate likelihood of each risk and its impact;
- develop a plan for mitigating likely and severe risks.

Part II

Estimation

Estimates

Plans include *schedules*, which rely on estimates.

Estimates often get overrun
(why? because you make best-case assumptions).

Prerequisites for good estimates:

- an accurate work breakdown structure;
- an effort estimate for each task in the WBS;
- a list of assumptions behind the estimates; and
- consensus that estimates are reasonable.

Assumptions in Estimates

Do assumptions make estimates less accurate?

Not necessarily!

Unstated assumptions, however, are troublesome.

If an assumption is well-documented and properly communicated, it can improve the quality of an estimate, by:

- revealing an inobvious problem or detail; or,
- revealing a simplification that some, but not all, members of the estimation team were relying on.

Finding out that an assumption is untrue is a good time to revisit an estimate.

Agreeing on assumptions allows estimators to move beyond the assumptions and come up with good estimates.

Padding Estimates

Padding Estimates

("Scotty Time[1]"). Padding an estimate seems like all win: underpromise and overdeliver, etc.

Pros:

- Look like a hero.
- Less time pressure.

Cons:

- Managers aren't dumb.
- Clients/managers may decide to not proceed based on padded estimate.

[1] `http://tvtropes.org/pmwiki/pmwiki.php/Main/ScottyTime`

Estimation Techniques

Some estimation techniques:

- Wideband Delphi
- PROBE
- COCOMO II
- The Planning Game

Steps in Wideband Delphi Process

Straightforward steps:

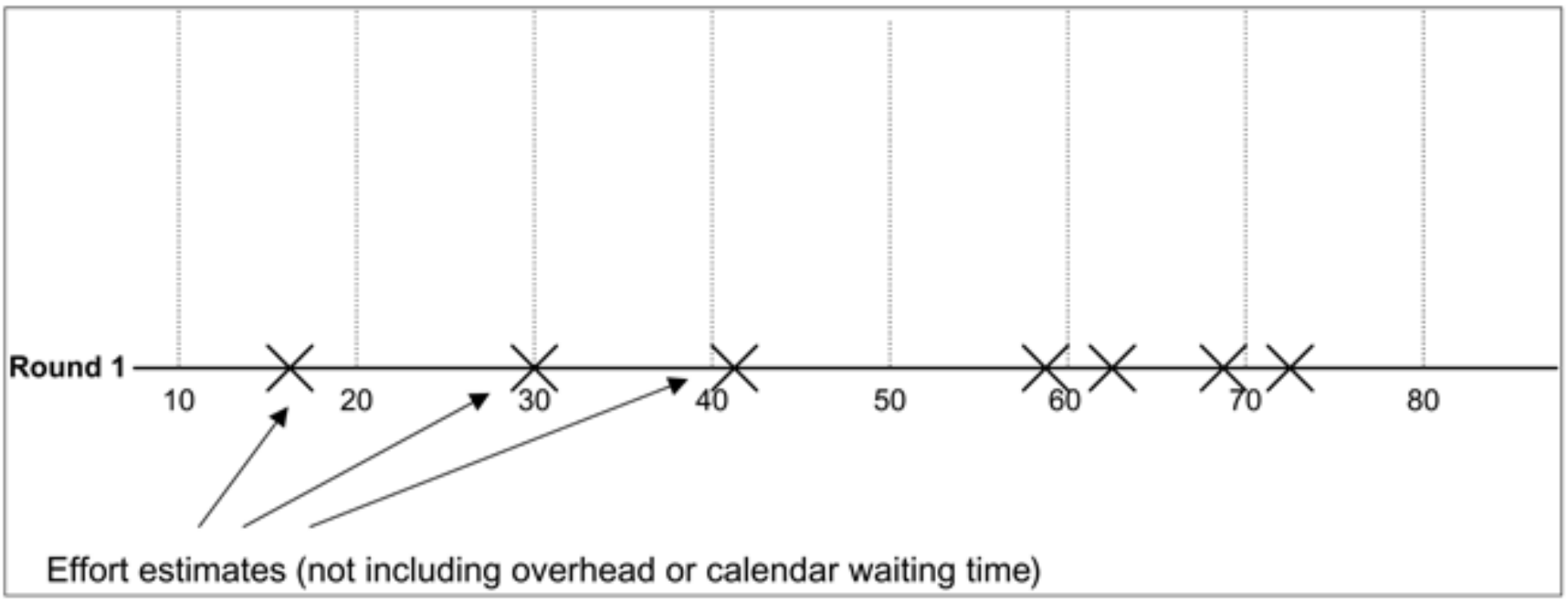
- Choose team and moderator.
- Kickoff meeting: WBS, assumptions, unit of estimation.
- Team members independently prepare estimates and more assumptions.
- Estimation session: the moderator
  1. requests estimates;
  2. attempts to achieve consensus by having members explain their estimates.
- Assemble the final list of tasks and estimates.
- Review results with project manager and team.

Estimation sessions are key to the Wideband Delphi Process.

Running an Estimation Session

1. Team members share effort estimates for each WBS task.
2. Moderator records estimates and plots them on a graph or records them in a table.

Estimates can range widely, because team members may make different assumptions.



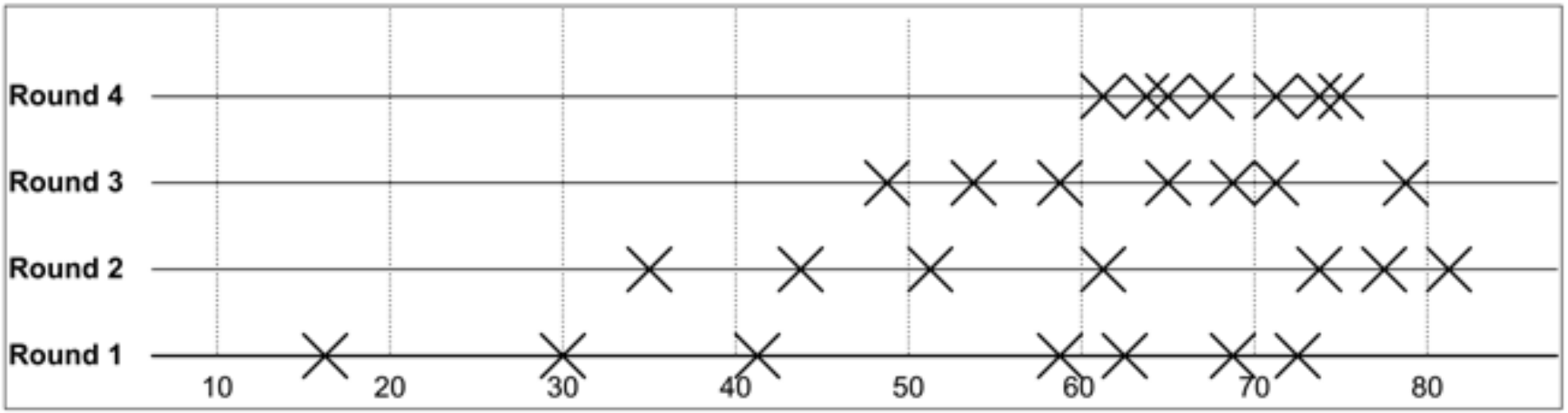
Achieving Consensus

Point of working in a team: do better than individuals would.

Wideband Delphi: multiple rounds of discussion.

In each round, team members discuss assumptions, clarify project details, and revise estimates.

Here's an example of convergence in estimates:



Other Estimation Techniques: PROBE

Idea: Doing something again should take
about as long as it did last time.

Obsessively track how long it took to do things in the past, then use linear regression to estimate how long it'll take in the future.

There are some problems with this approach, though.

Other Estimation Techniques: COCOMO II

Idea: Feed in guesses about project's scope, apply a formula, get estimate of size and effort.

The formula is full of fudge factors (from empirical data).

Examples of inputs:

- Memory constraints
- Analyst capability
- Product complexity

COCOMO stands for *COnstructive COst MOdel*.

Other Estimation Techniques: The Planning Game

Not this kind of game:

The Planning Game: Description

Instead: an estimation technique developed by Kent Beck (inventor of extreme programming) while working at Chrysler in the 1990's.

The Planning Game is a full planning process that:

1. identifies the scope of the project;
2. identifies tasks required to complete the project; and
3. estimates the effort required for these tasks.

The Planning Game: Phases

Two phases:

- Release planning—plans the scope of the project.
- Iteration planning—plans the activities and tasks of the developers.

The Planning Game requires a team consisting of customers and developers.

The Planning Game: Phases

Each week,
team meets to plan the immediate future of the project:

- writes user stories
  describing project requirements on index cards;
- assigns effort estimates for the stories
  (e.g., 1, 2, or 3 weeks); and
- prioritizes the requirements.

The Planning Game: Iteration Planning

Occasionally, the team:

- divides requirements into sets of tasks to be completed;
- assigns these tasks to developers; and,
- estimates effort for these tasks.

Developers complete tasks and
match them with the user stories.