Estimating

“It is difficult to make predictions, especially about the future” - Attributed to Yogi Berra
(… but also to Niels Bohr and others)
Goals of the Unit

• Understand the fundamental (and simple) relation among Duration, Effort, and Manpower

• Understanding the perils of estimations in software development

• Learn the techniques commonly adopted to estimate effort in projects
Effort, Duration, and Resources
Estimation

- **Effort (Work):** how much work will the activity need to be completed
- **Resources:** type and quantity of resources available for the activity
- **Duration:** how long will the activity last for
Effort

- The **amount of work** an activity requires to be completed. *A very good starting point.*
- Measured in *(work-)*days, *(work-)*weeks, *(work-)*months
- Often the term *man-* is also used (e.g. 3 man-months = 1 person working for 3 months; 3 people working for one month)
- Mind you, though: the work required in a project includes direct and indirect activities (i.e., getting the stuff done, but also email, communication, reports, meetings, ...)

*spm* - ©2014 adolfo villafiorita - introduction to software project management
Resources

• The resources needed to carry the work out. Typically a constraints (limited)

• Expressed as **manpower**, that is, number of people and percentage of availability

• For instance: 1 person full time; 2 people at 50%

• Certain tasks might require **material resources** (e.g. bricks & pipes) or **equipment** (e.g. a machine for DNA sequencing)

• Material resources are consumed by the execution of an activity; equipment can be reused

• In software development usually resources = manpower
Duration

- How long the activity will last for
- Measured in **hours, days, months, ...**
- Often:
  - 1 week = 5 days = 40 hours
  - 1 month = 20 days ... why?
- In some countries:
  - 1 week = 36 hours (7.12 hours/day)
- Calendar time differs from duration: calendar time includes non-working days, holidays, ...
A (simplistic) view

$$D = \frac{E}{M}$$

- Fix any two among $D$, $E$, and $M$ (= manpower), and you get the third
- Typically effort and man power are the variables you will be working with (and derive duration from it)
- Notice also that manpower is

$$M = \sum_{i=1}^{N} p_i$$

$N$ = number of resources

$p_i$ = percentage of availability
Some Examples

• 1 week = 40 hours

• Effort: 40 man-hours; Resources: 1 @ 100% →
  \[ D = \frac{40 \text{ man-hours}}{1 \text{ man}} = 40 \text{ hours} = 1 \text{ week} \]

• Effort: 80 hours; Resources: 2 @ 100% →
  \[ D = \frac{80 \text{ man-hours}}{2 \text{ man}} = 40 \text{ hours} = 1 \text{ week} \]

• Effort: 80 hours; Resources: 1 @ 50% →
  \[ D = \frac{80}{50\%} = 160 \text{ hours} = 4 \text{ weeks} \]
  (a person at 50% will be able to work 20 hours/week; it takes 4 weeks to get to the 80 hours needed for the activity)
Important Remark

• The equation is a simplification... good enough for various cases (do not take it to extremes)

• The hypothesis of “take any two variables” in $D = \frac{E}{M}$ is not always reflected in practice (e.g. the variables are not completely independent)

• Estimating is hard: deciding how much effort a task requires is difficult ... in the next few lessons we will look at techniques and tools for estimation
Uncertainty in Planning
Uncertainty in planning

- Planning has a certain degree of uncertainty
- (In software and not only) we are over-optimistic
- “best guess” might also be a problem
Uncertainty in planning

• Three practices (not necessarily good) to account for uncertainty
  – **Implicit padding**: each activity includes some contingency time
  – **Explicit padding**: the contingency time is explicitly modeled as an activity
  – **React and re-plan**: when a delays occurs, you re-plan and re-define a new realistic schedule

• Some suggestions:
  – Always evaluate the cost of delays
  – Choose a strategy and make it clear (with yourself and with your stakeholders, if possible)
Implicit Padding

• Each activity includes some extra duration/effort to take into account delays

• Estimations become inaccurate and difficult to control

• Always being pessimistic (and always delivering earlier, according to a wrong pessimistic plan) is not necessarily good... the plan is still inaccurate

• Interaction with other projects might still be a problem (you deliver earlier and the next project needs to re-allocate resources in order to start an activity earlier)
Explicit Padding

• The plan includes some extra activities or slack to take into account delays in finishing activities

• Contingency is not explicit in the plan. Data is accurate; no problems in budgeting/monitoring/...

• Might be difficult to have it accepted... the customer might think of padding as useless
React and Re-plan

• When a delay occurs, it is dealt with and specific actions are decided. The actions are incorporated into the plan, which is revised.

• **Flexible:** takes into account different strategies for dealing with contingencies (e.g. removing dependencies, adding resources).

• This is not a **planning practice**. It is a monitoring and executing practice.

• **The plan** does not show possible alternative courses of actions to the occurrence of a risk/contingency.
Estimation Techniques
Approaches to Estimation

- **Expert Judgement** is “quick and dirty” and based on experience. It can be applied either top-down or bottom-up.

- **PERT (Program Evaluation and Review Technique)** takes into account the probabilistic nature of estimations.

- **Algorithmic Techniques** provide estimations by measuring specific qualities of a system and applying algorithms (Function Points, COCOMO, WebObjects).
Expert Judgement

• Efficient and fast. Based on personal (rather than organizational) assets

• Underlying assumption: the project uses a product WBS

• Top-down
  – Start at the top of the WBS and break estimations as you move down

• Bottom-up
  – Start at the bottom of the WBS and sum as you move up
PERT
Program Evaluation and Review Technique
PERT

- Program Evaluation and Review Technique
- Developed in the sixties
- It is a **methodology** to define and control projects
- Variations exists (e.g. PERT/COST developed by NASA/DOD)
A Motivating Example
PERT Formula

• Estimation in PERT is based on the idea that estimates are uncertain
  – Therefore uses duration ranges
  – And the probability of falling to a given range

• Uses an “expected value” (or weighted average) to determine durations
PERT

• For each task, three estimates:
  – Optimistic
    * (would likely occur 1 time in 20)
  – Most likely
    * (modal value of the distribution)
  – Pessimistic
    * (would be exceeded only one time in 20)
Variance and Standard Deviation

- Variance ($\sigma^2$) and standard deviation ($\sigma$) measure how spread a population is from the average
- Standard deviation ($\sigma$) is the square root of variance
- **Example: normal distribution**: a bell shaped probability distribution function

Beta Distributions

- Average is given by the formula:

\[ t_e = \frac{(a + 4m + b)}{6} \]

- Variance (\(\sigma^2\)) and standard deviation (\(\sigma\)) are given by:

\[ \sigma^2 = \left(\frac{b - a}{6}\right)^2 \]

\[ \sigma = \frac{b - a}{6} \]
PERT Formula

- Task duration is an average of three estimations:

\[ t_e = \frac{(a + 4m + b)}{6} \]

\( t_e \) = expected time
\( a \) = optimistic time estimate (1 in 20)
\( m \) = most likely time estimate
\( b \) = pessimistic time estimate (1 in 20)
Algorithmic Techniques
Introduction

• Goal: find a way to systematically determine the effort (duration) required for an (arbitrary) task/project

• Ideally:
  – Identify a set of measurable characteristics of a project that determine the project’s effort/duration
  – Define a function that, given the characteristics mentioned above, computes the effort/duration

\[ f(x_1, \ldots, x_n) = e \]

Problem: how do you find \( f, x_1, \ldots, x_n \)?
Solution

• Look at existing projects/datasets; each project is represented by a vector:

\[ < a_1, ..., a_n, \text{effort} > \]

• Find correlations between (some of the) variables in the datasets:

\[ f(a_1, ..., a_k) \propto \text{effort} \]

• Find appropriate measurement means for the variables at the beginning of a project (so that we can apply the function to a new project)
Discussion

• Advantages:
  – Replicable
  – Objective

• Limitations of the models:
  – Size of the dataset used for defining the model and accuracy of the model

• Limitations of their application:
  – Resources needed to collect the data (time and expertise)
  – Applicability of the model to the system at hand
  – Accuracy of the data collected to estimate for a new system
Main Techniques

• Function Points (FP)
  – Function-based, it estimates effort based on its functional characteristics
  – Duration/Team size computed through productivity metrics
  – It requires a critical analysis of the requirements

• Constructive Cost Modeling (COCOMO)
  – Size-based, it estimates effort, duration, and team size based on the (presumed) size of a system in source lines of code
  – Different families of models

• Sometime used in conjunction (FP to get the system size; COCOMO to do the rest)