Topics to be covered

- Difficulties of Estimation
- Where are estimates done?
- Problems of over- and under- estimate
- Estimation techniques
What makes a successful project?

Delivering:
- agreed functionality
- on time
- at the agreed cost
- with the required quality

Stages:
1. set targets
2. Attempt to achieve targets

BUT what if the targets are not achievable?
What makes a successful project?

- Targets are set for a project and the project manager tries to meet them.

- A project manager has to produce:
  - An estimate of the effort.
  - An estimate of the activity durations.

- An estimate of effort affects Cost.

- An estimate of activity durations affects The delivery time.
Some problems with estimating

- Nature of software.
  - Complexity and invisibility of software.

- Subjective nature of much of estimating
  - Over-estimating small tasks and
  - Under-estimating large ones.

- Political pressures
  - Different objectives of people in an organization
  - Managers may wish to reduce estimated costs in order to win support for acceptance of a project proposal
Some problems with estimating

- Changing technologies
  - Technology is rapidly changing, making the experience of previous project estimates difficult to use in new ones.

- Projects differ
  - Experience on one project may not be applicable to another
Where are estimates done?

Estimates are carried out at different stages of a software project for a variety of reasons.

- **Feasibility study**
  - Estimates here conforms that the benefits of the potential system will justify the costs

- **Strategic planning**
  - Project portfolio management will involve:
    - Estimating benefits and costs of new applications (projects) to allocate **priorities**.
    - Such estimates may also influence the scale of development **staff recruitment**
Where are estimates done?

- **System specification**
  - Design shows how user requirements will be fulfilled.
  - Estimating The efforts needed to implement different design proposals.
  - Estimates at the design stage will also confirm that the feasibility study is still valid.
Where are estimates done?

- **Evaluation of suppliers proposals**
  - A manager could consider putting development to tender
  - Potential contractors would examine the system specifications and produce estimates (their bid).
  - The manager can still produce his own estimates why?
    - To question a bid that for instance that seems too low which could be an indication of a bad understanding of the system specifications.
    - Or to compare the bids to in-house development
Where are estimates done?

- **Project planning**
  - As the planning and implementation of the project becomes more detailed
    - More estimates of smaller work components will be made
    - These will confirm earlier broad estimates
    - And support more detailed planning (e.g. staff allocation)
An over-estimate is likely to cause project to take longer than it would otherwise.

This can be explained by the application of two laws:

- **Parkinson’s Law**: ‘Work expands to fill the time available’
  - Thus, e.g. for an easy task over estimating the duration required to complete it will cause some staff to work less hard to fill the time.
- **Brook’s Law**: putting more people on a late job makes it later
  - So overestimating the effort required to perform a task (activity) means more staff assigned to it than needed.
Over- and under-estimating

- Underestimating a project: Can cause the project to not be delivered on time or cost
- but still could be delivered faster than a more generous estimate
- On the other side the danger of underestimating a project is the effect on the quality

- **Zeroth law of reliability**: if a system doesn't have to be reliable it can meet any other objective
Basis for successful estimating

A. The need for historical data.

- Most estimating methods need information about past projects
- Care has to be considered when applying past performance to new projects because of possible differences in factors such as:
  - Different programming languages
  - Different experience of staff
  - Different terminology

There are international Data Base containing data about thousands of projects that can be used as reference
B. Measuring work.

- The time and cost to implement software depends on:
  - The developer’s capability and experience
  - The technology that will be used

- The usual practice is to start by expressing work size independently of the effort, using measures such as:
  (a) SLOC OR KLOC: Source lines of code or thousands of lines of code
  (b) Alternative size measure is Function Points (FP)
A taxonomy of estimating methods

- Bottom-up - activity based, analytical
- Parametric or algorithmic models e.g. function points
- Expert opinion - just guessing?
- Analogy - case-based, comparative
- Parkinson and ‘price to win’
Bottom-up versus top-down

- **Bottom-up**
  - use when no past project data
  - identify all tasks that have to be done – so quite time-consuming
  - use when you have no data about similar past projects

- **Top-down**
  - produce overall estimate based on project cost drivers
  - based on past project data
  - divide overall estimate between jobs to be done
Bottom-up estimating

1. Break project into smaller and smaller components

2. Stop when you get to what one person can do in one/two weeks

3. Estimate costs for the lowest level activities

4. At each higher level calculate estimate by adding estimates for lower levels
Top-down Estimation

- It is associated with parametric or algorithmic models.
- A formula for a parametric model:
  - Effort = (System Size) / (Productivity Rate)
- The model of forecasting the SW development effort has two components
  - System size is a method of assessing the amount of work
  - Productivity rate is a method of assessing the rate of work at which the task can be done
Example:

System Size = 3 KLOC.

Productivity Rate = 40 days per KLOC.

\[ \text{Effort} = \frac{\text{System Size}}{\text{Productivity Rate}} \]

Effort = 3 \times 40 = 120 Days.

System Size is a size driver.

Productivity Rate is a productivity driver.
Top-down Estimation

- Other parametric models:
  - **Function points** is concerned more with task sizes.
  - **COCOMO** is concerned more with productivity rate.
Top-down estimates

- Produce overall estimate using effort driver(s)
- Distribute proportions of overall estimate to components

**Overall project**

- **Estimate** 100 days

**Components**

- **Design**
  - 30% i.e. 30 days

- **Code**
  - 30% i.e. 30 days

- **Test**
  - 40% i.e. 40 days
Estimation by Analogy

- It is also called case-based reasoning.
- For a new project the estimator identifies the previous completed projects that have similar characteristics to it.
- The new project is referred to as the target project or target case.
- The completed projects are referred to as the source projects or source case.
- The effort recorded for the matching source case is used as the base estimate for the target project.
- The estimator calculates an estimate for the new project by adjusting the (base estimate) based on the differences that exist between the two projects.
There are software tools that automate this process by selecting the nearest project cases to the new project.

Some software tools perform that by measuring the Euclidean distance between cases (projects).

The Euclidean distance is calculated as follows:

\[
\text{distance} = \sqrt{((target\_parameter_1 - source\_parameter_1)^2 \ldots + (target\_parameter_n - source\_parameter_n)^2)}
\]
Estimation by Analogy Example

- Assume that cases are matched on the basis of two parameters, the number of inputs and the number of outputs.
  - The new project (target case) requires 7 inputs and 15 output
  - You are looking into two past cases (source cases) to find a better analogy with the target project:
    - Project A: has 8 inputs and 17 outputs.
    - Project B: has 5 inputs and 10 outputs.

Which is a more closer match for the new project A or project B?
Distance between new project and project A:
- Square-root of ((7-8)^2 + (15-17)^2) = 2.24

Distance between new project and project B:
- Square-root of ((7-5)^2 + (15-10)^2) = 5.39

Project A is a better match because it has less distance than project B to the new project.