11.3 Cost estimation
To estimate how much software-engineering time will be required to do some work:

- Elapsed time: The difference in time from the start date to the end date of a task or project.
- Development effort:
  - The amount of labour used in person-months or person-days.
  - To convert an estimate of development effort to an amount of money:
    You multiply it by the weighted average cost (hourly cost) of employing a software engineer for a month (or a day).

Principles of effective cost estimation

Principle 1: Divide and conquer.
- To make a better estimate, you should divide the project up into individual subsystems.
- Then divide each subsystem further into the activities that will be required to develop it.
- Next, you make a series of detailed estimates for each individual activity.
- And sum the results to arrive at the grand total estimate for the project.

Principle 2: Include all activities when making estimates.
- The time required for all development activities must be taken into account.
  * Including:
    - Prototyping
    - Design
    - Testing
    - Debugging
    - Writing user documentation
    - Deployment

Principle 3: Base your estimates on past experience combined with knowledge of the current project.
- If you are developing a project that has many similarities with a past project:
  - You can expect it to take a similar amount of work.
  - Base your estimates on the personal judgment of your experts.
- Use algorithmic model(s) developed in the software industry as a whole by analyzing a wide range of projects.

Algorithmic models

Allow you to systematically estimate development effort.
- Based on an estimate of some other factor that you can measure, or that is easier to estimate:
  - The number of use cases
  - The number of distinct requirements
  - The number of classes in the domain model
  - The number of widgets in the prototype user interface
  - An estimate of the number of lines of code

- COCOMO:
  \[ E = a + bN \]

- Functions Points:
  \[ S = W_1 F_1 + W_2 F_2 + W_3 F_3 + \ldots \]
Different levels of requirement stability

Different technical complexity of the requirements

Different development processes and maturity levels

Different software developers

Different types of customers and users

Accounting for everything that could go wrong

—

Allowing for typical things going wrong

—

Optimistic (O)

Features are more likely to have been completed

Likely (L)

Principle 6: Combine multiple independent estimates.

• Use several different techniques and compare the results.
• If there are discrepancies, analyze your calculations to discover what factors causing the differences.
• Use the Delphi technique.
  —Several individuals initially make cost estimates in private.
  —They then share their estimates to discover the discrepancies.
  —Each individual repeatedly adjusts his or her estimates until a consensus is reached.

Principle 7: Revise and refine estimates as work progresses

• As you add detail.
• As the requirements change.
• As the risk management process uncovers problems.