# Unit 4 Software Project Estimation

**UNIT - IV: Software Project Estimation**

**Introduction**

- S/w Testing is a process of evaluating a s/w system by manual or automatic means and verifying it against specified requirements.
- It determines the gap between expected and actual results.
- The estimation of project includes determination of effort, time, resources and money required for developing any project.
- The cost of development of project is the main determinant of its prize and effort.
- Underestimating the project parameters may lead to project failure.
- Overestimating the project parameters may lead to inviable cost and ultimately project may not be taken up.

**Software Project Parameters**

The parameters that determine the project cost includes:

- **Quality**:
  - Quality estimation in project management refers to defining the standards and criteria that the project’s deliverables must meet to satisfy stakeholders' expectations.
  - This process ensures that the final product is not only completed within the specified scope, time, and cost but also meets the necessary quality requirements.
- **Cost of development**:
  - Project is considered successful if the revenue from project is more than the cost incurred.
  - It determines the financial feasibility of the project.

    - It determines the cost to be quoted to the customers.
- **Resource Requirements**:
    - Human Resources (HR) is the most crucial part of s/w development process.
    - S/w development process requires effort from number of people as s/w engineers, system analyst, programmers, data entry operators, system testers, etc.
    - Since high skilled people are paid more, the cost of project is influenced by number of and level of skills of HR.
- **Time**:
    - The benefit of any project is determined after it is completed.
    - An incomplete project may lead to additional cost, time and other resources.
- **Scope/Size**:
    - This refers to defining the work required to deliver the project's goals and objectives.
    - Scope/Size estimation is a crucial part of project planning that involves defining the boundaries, deliverables, and overall size of the project.
    - It answers the fundamental question of what needs to be done and ensures that the project team, stakeholders, and clients have a clear understanding of the work required.
- **Risk**:
    - Risk project estimation involves identifying, analyzing, and quantifying the potential risks that could impact a project’s success.

---

**Approaches to Software Estimation**

- **Top-Down Approach**:
    - Estimation started from System Level.
    - Estimates the cost of system activities as integration, configuration & documentation.
    - Underestimates cost of solving difficult technical & integration problems.

    - Does not provide detailed estimate justification.
    - Less costly than Bottom-up approach.
- **Bottom-Up Approach**:
    - Estimation started from Component Level.
    - System is decomposed into components and efforts required and then the cost is added up.
    - Underestimates the cost of difficult technical problems.
    - Provide detailed justification.
    - More costly than top-down approach.

---

**Project Estimation**

- Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.
- Estimation determines how much money, effort, resources, and time it will take to build a specific system or product.
- Estimation includes:
    - Estimate the size of the development product.
    - Estimate the effort in person-months or person-hours.
    - Estimate the schedule in calendar months.
    - Estimate the project cost in agreed currency.
- Estimation is based on:
    - Past Data/Past Experience.
    - Available Documents/Knowledge.
    - Assumptions.
    - Identified Risks.

---

**Project Estimation Techniques**

- **Estimation by Expert Judgement**:
    - Widely used estimation technique.

    - Expert makes subjective assessment of software size.
    - Cost of individual components are combined to get overall estimate.
    - Estimation can be made by group of experts to overcome limitations of this method.
    - **Limitations**:
        - Prone to human judgement bias.
        - Expert may not have knowledge of all aspects of the project.
- **Estimation by Analogy**:
    - If project to be estimated is similar to any project that is already been completed, estimation is done with reference to completed project.
    - However, no two s/w projects can be the same so there will always remain some differences to some extent.
    - Two projects to be compared are ranked on different project parameters to determine the degree of their similarities.

---

**Classification of Software**

- **Purpose**: Software can be classified as system software (e.g., operating systems, device drivers) or application software (e.g., word processors, games).
- **Platform**: Software can be classified as native software (designed for a specific operating system) or cross-platform software (designed to run on multiple operating systems).
- **Deployment**: Software can be classified as installed software (installed on the user's device) or cloud-based software (hosted on remote servers and accessed via the internet).
- **License**: Software can be classified as proprietary software (owned by a single entity) or open-source software (available for free with the source code accessible to the public).
- **Development Model**: Software can be classified as traditional software (developed using a waterfall model) or agile software (developed using an iterative and adaptive approach).
- **Size**: Software can be classified as small-scale software (designed for a single user or small group) or enterprise software (designed for large

organizations).

- **User Interface**: Software can be classified as Graphical User Interface (GUI) software or Command-Line Interface (CLI) software.

---

**COCOMO Model**

- The COCOMO Model is a procedural cost estimate model for software projects and is often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time, and quality.
- It was proposed by Barry Boehm in 1981 and is based on the study of 63 projects.
- **Key Parameters**:
  - **Effort**: Amount of labor required to complete a task, measured in person-months.
  - **Schedule**: The time required to complete the job, proportional to the effort put in, measured in weeks or months.

---

**COCOMO – Advantages & Disadvantages**

- **Advantages**:
  - Easy to estimate the total cost of the project.
  - Easy to implement with various factors.
  - Provides insights from historical projects.
- **Disadvantages**:
  - Ignores requirements, customer skills, and hardware issues.
  - Limits accuracy of software cost estimation.
  - Largely depends on time factors.

---

**Basic COCOMO**

- COCOMO has three modes of software development projects:
  - **Organic Project**: Small and simple, handled by a small team with good domain knowledge and few rigid requirements. Example: Small data processing or inventory management system.

- **Semidetached Project**: Intermediate project with mixed experience team members. Example: Database design or OS development.
- **Embedded Project**: Complex project with a large team, considering software, hardware, and operational parameters. Example: Banking software or traffic light control software.

---

**Basic COCOMO Estimations**

| Software Projects | a | b | c | d |
|---|---|---|---|---|
| Organic | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi Detached | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded | 3.6 | 1.20 | 2.5 | 0.32 |

Formulas:

- **Effort (E)**: E = a * (KLOC)^b
- **Time**: Time = c * (Effort)^d
- **Person Required**: Effort / Time

Example:

- A software project with 50 KLOC (Organic Project):
  - Effort ≈ 146 person-months.
  - Time ≈ 17 months.
  - Team size ≈ 9 people.

**Intermediate Model:**

- The basic COCOMO model assumes that the effort is only a function of the number of lines of code and some constants evaluated according to the different software systems.
- However, in reality, no system's effort and schedule can be solely calculated based on Lines of Code.

- For that, various other factors such as reliability, experience, and Capability are considered.
- These factors are known as **Cost Drivers**.

**Detailed Model:**

- Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver’s impact on each step of the software engineering process.
- The detailed model uses different effort multipliers for each cost driver attribute.
- In detailed COCOMO, the whole software is divided into different modules and then we apply COCOMO in different modules to estimate effort and then sum the effort.

---