Course Material No. 3

SOFTWARE ENGINEERING 2

FE LARWA-HABLANIDA
Course Instructor

PROJECT PLANNING and ESTIMATION 1

3

LEARNING OUTCOMES

At the end of the lesson, the learner will be able to:

- Explain the importance of effective project planning and estimation.
- Apply various estimation techniques
- Justify project estimates with appropriate supporting data

RESOURCES NEEDED

For this lesson, you would need the following resources:

- Module/Learning Material
- Pen, Pencil, Paper, and Eraser

DISCUSSION:

INTRODUCTION TO PROJECT PLANNING

Project Planning is the process of defining the project goals, tasks, resources, schedule, and budget needed to deliver a successful project.

Systematic process of defining project objectives, scope, resources, and timeline. Aligning stakeholders and resources for project success

"Failing to plan is planning to fail." – Benjamin Franklin

Why Project Planning is Essential

1. Sets a clear direction
2. Identifies potential risks early
3. Helps in resource management
4. Builds stakeholder confidence

Purpose of Planning

1. Establish project scope and boundaries
2. Schedule timelines and milestones
3. Allocate resources efficiently
4. Minimize risks and prepare for uncertainties
5. Communicate expectations to stakeholders

Importance of Project Planning

- **Benefits:**
  1. Improves decision-making
  2. Enhance team coordination
  3. Facilitates monitoring and control
  4. Helps manage risks and changes
  5. Aligns stakeholders to common objectives
- **Consequences of Poor Planning:**
  1. Missed deadlines
  2. Budget overruns
  3. Unclear responsibilities
  4. Poor quality or incomplete deliverables

Key Components of a Project Plan (15 min)

Component	Description
Scope	▪ Defines project boundaries and deliverables
Schedule	▪ Project timeline with tasks, durations, and milestones
Resources	▪ Human, technical, and financial resources needed
Cost Estimation	▪ Budget forecast based on tasks and resource usage
Risk Management Plan	▪ Identifies and plans for potential issues
Communication Plan	▪ Describes how information will be shared among stakeholders
Quality Plan	▪ Defines quality criteria and control mechanisms

Project Planning Life Cycle

The **Project Life Cycle** is a series of phases that a project goes through from initiation to closure, including all aspects of planning, execution, monitoring, and completion. **Planning** is not a one-time activity—it occurs throughout the entire project life cycle.

- **Phases:**

1. **Project Initiation:**
   - Identify project goals and objectives
   - Determine feasibility (technical, financial, legal)
   - Identify stakeholders and project sponsor
   - Develop **Project Charter**

     Project Charter – outlines project purpose, scope, stakeholders, and initial constraints.

2. **Project Planning:**
   - Define project scope
   - Create **Work Breakdown Structure (WBS)**
   - Estimate time, cost, and resources
   - Develop schedule and budget
   - Perform risk assessment
   - Plan for communications, procurement, and quality
3. **Project Execution:**
   - Carry out the plan
   - Assign tasks to team members
   - Manage resources and stakeholder expectations
   - Conduct team meetings
   - Implement communication plan
   - Ensure quality standards are met
4. **Monitoring & Control:**
   - Track progress and manage changes
   - Track progress against the plan
   - Monitor scope, time, cost, and quality
   - Manage change requests
   - Update schedules and risk registers
   - Communicate status to stakeholders
5. **Project Closure:**
   - Final delivery and post-project evaluation
   - Final product delivery to client
   - Confirm acceptance of deliverables
   - Close contracts
   - Release project resources
   - Conduct lessons learned review
   - Archive documents

Role of the Project Manager in Planning

- **The project manager is responsible for:**
  - Leading the planning process
  - Ensuring team involvement and buy-in

- Balancing competing constraints (scope, time, cost, quality)
- Maintaining alignment with stakeholders

Tools for Project Planning

1. **Gantt Charts** – Visual project timeline
2. **PERT/CPM Charts** – Identify task dependencies and critical path
3. **Microsoft Project / Jira / Trello** – Project tracking and collaboration
4. **Work Breakdown Structure (WBS)** – Breaks down the project into smaller tasks

DEFINING PROJECT SCOPE

Project Scope defines all the work required to complete a project successfully — no more, no less.

- **It answers:**
  - What will be delivered?
  - What will not be included?
  - How will it be accomplished?
- **Difference Between:**
  - **Product Scope**: Features and functions of the final product (e.g., features of a mobile app)
  - **Project Scope**: Work required to deliver the product (e.g., design, coding, testing)

Importance of Defining Scope

- **Why it Matters:**
  - Sets clear expectations for stakeholders
  - Helps in estimating time, cost, and effort
  - Reduces risk of misunderstanding and scope creep
  - Serves as a reference throughout the project
- **What Happens Without a Clear Scope:**
  - Unclear objectives
  - Frequent changes
  - Conflicts among team members and stakeholders
  - Project delays and budget overruns

Key Elements of a Scope Statement

A well-defined **Project Scope Statement** typically includes:

Element	Description
Objectives	▪ Specific goals of the project
Deliverables	▪ Tangible outputs/products
Inclusions	▪ What is covered
Exclusions	▪ What is NOT covered
Constraints	▪ Time, cost, tech limitations
Assumptions	▪ Things assumed to be true

Work Breakdown Structure (WBS) (15 min)

WBS is a hierarchical decomposition of the total scope into manageable parts. It breaks the project into smaller, well-defined **work packages**. Each descending level represents an increasingly detailed definition of project work.

- **WBS Guidelines:**
  - Should be deliverable-oriented
  - Each level provides more detail and clarity

- Helps in estimating cost and duration

➢ **Purpose of WBS**

- Define the scope of the project clearly
- Enable accurate estimation (time, cost, resources)
- Facilitate task assignment and accountability
- Serve as a foundation for scheduling, budgeting, and monitoring

➢ **Principles of Effective WBS**

a) **100% Rule**

- The WBS must capture 100% of the project scope, including internal, external, and management work.

b) **Mutually Exclusive Elements**

- No overlap in scope between elements at the same level.

c) **Level of Detail**

- Work packages should be small enough to:
- Estimate cost/time accurately
- Assign to a team or individual
- Track progress clearly

➢ **Structure of WBS**

a) **Levels of WBS**

Level	Description
Level 1	▪ Entire Project (e.g., "Library Management System")
Level 2	▪ Major Deliverables or Phases (e.g., Requirements, Design, Implementation)
Level 3	▪ Sub-deliverables or Components (e.g., UI Design, Backend Coding)
Level 4	▪ Work Packages (e.g., Create Login Page, Build Database Schema)

b) **Format Types**

- Indented Outline Format
- Hierarchical Tree (Graphical)
- Tabular Format with Codes

Scope Creep and How to Control It

➢ **Scope Creep?**

- Unauthorized or uncontrolled changes or additions to project scope.

➢ **Causes:**

- Vague scope definition
- No change control process
- Poor communication

➢ **How to Prevent It:**

- Get formal approval for all changes
- Document scope clearly
- Use a Change Control Log
- Educate clients and team on the process

ESTIMATION FUNDAMENTALS

Estimation is the process of approximating the effort, time, and cost required to complete a specific project or activity. It's about making informed predictions, not guesses.

Estimation is Needed For:

1. Scheduling and deadlines
2. Budgeting
3. Resource allocation
4. Risk identification
5. Managing stakeholder expectations

Importance of Accurate Estimation

- **Why It Matters:**
  a) It ensures realistic planning
  b) It prevents overruns in cost and time
  c) It builds trust with stakeholders
  d) It enables better decision-making
- **Effects of Inaccurate Estimation:**
  a) Missed deadlines
  b) Budget overruns
  c) Low team morale
  d) Loss of client trust
  e) Project failure

Types of Estimations

Type of Estimate	Description
Effort Estimate	▪ Time/effort needed (usually in hours or person-days)
Cost Estimate	▪ Monetary cost of resources and activities
Time Estimate	▪ Calendar duration to complete a task or project
Resource Estimate	▪ Human and technical resources needed

Levels of Estimation Accuracy

Estimation Level	Time in Project	Accuracy
Rough Order of Magnitude (ROM)	▪ Early stage	▪ -25% to +75%
Budget Estimate	▪ Planning phase	▪ -15% to +25%
Definitive Estimate	▪ Execution phase	▪ -5% to +10%

Early estimates are more uncertain; accuracy improves as the project progresses.

Challenges in Estimation

- **Common Pitfalls:**
  a) Incomplete or unclear requirements
  b) Overconfidence or pressure to reduce estimates
  c) Ignoring past project data
  d) Not involving technical experts
  e) Underestimating complexity or risks
- **Example of Poor Estimation:**
  - A team estimated a data migration project would take 2 weeks.
  - It ended up taking 6 weeks due to data quality issues and system incompatibility.

Basic Estimation Process

1. Understand requirements – Clarify scope and deliverables
2. Break down the work – Use a Work Breakdown Structure (WBS)
3. Use historical data or expert input
4. Choose an estimation method (e.g., expert judgment, PERT, analogy)

5. Add contingency/buffer for uncertainty
6. Document assumptions clearly
7. Review with stakeholders/team

ESTIMATION TECHNIQUES

A **structured method** used to forecast the effort, time, or cost required to complete a project or task. **Estimation** is not guessing—it's a calculated prediction based on experience, models, or data.

Why Use Techniques?

1. Improve accuracy
2. Reduce bias and assumptions
3. Enable consistent and traceable estimation
4. Assist in risk and cost management

Categories of Estimation Techniques

Category	Examples
Expert-Based	▪ Expert Judgment, Delphi Technique
Analogous-Based	▪ Historical Projects, Similar Systems
Algorithmic/Model-Based	▪ COCOMO, Function Point, Use Case Points
Decomposition-Based	▪ Bottom-Up, WBS-based Estimation
Probabilistic-Based	▪ PERT, Three-Point Estimation

Key Estimation Techniques

1. **Expert Judgment**
   - Based on the knowledge and experience of subject matter experts
   - Often used when historical data is unavailable

   ➢ **Pros:**
   a. Quick and easy

   ➢ **Cons:**
   Subjective, biased

2. **Delphi Technique**
   - An anonymous group of experts submits estimates independently
   - Facilitated discussion and re-estimation rounds to reach consensus

   ➢ **Pros:**
   a. Reduces bias, encourages consensus

   ➢ **Cons:**
   a. Time-consuming

3. **Analogous Estimation**
   - Uses data from past similar projects to estimate the current one

   ➢ **Pros:**
   a. Quick, uses real data

   ➢ **Cons:**
   a. Less accurate if projects differ

Example:

- *If a previous e-commerce website took 10 weeks, a similar site may also take 10–12 weeks.*

4. **Bottom-Up Estimation**
   - Break down the project into **smaller tasks (WBS)**
   - Estimate each component and roll up the total

➢ **Pros:**

a. High accuracy

➢ **Cons:**

a. Time-consuming, requires a detailed breakdown

5. **Three-Point Estimation (PERT)**
   - It is a technique used in project planning and estimation to improve the accuracy of time or cost estimates by considering uncertainty and risk.

➢ **Different Estimate:**

a. **Optimistic Estimate (O)**
   - The **best-case scenario** – the shortest possible time/cost, assuming everything goes better than expected.

b. **Most Likely Estimate (M)**
   - The **realistic scenario** – the expected time/cost based on normal conditions.

c. **Pessimistic Estimate (P)**
   - The **worst-case scenario** – the longest possible time/cost assuming significant problems or delays.

➢ **PERT Formula:**

a. **Expected Estimate (TE)**
   - This weighted average puts the most weight on the **most likely** estimate.

$$\mathbf{TE} = \frac{\boldsymbol{O} + 4\boldsymbol{M} + \boldsymbol{P}}{\mathbf{6}}$$

b. **Standard Deviation (SD)**
   - To measure the **uncertainty** or **risk** around the estimate:

$$SD = \frac{P - O}{6}$$

➢ **Pros:**

a. Incorporates uncertainty

➢ **Cons:**

a. Still based on subjective values

Example:

- *If **O**=4 days, **M**=6 days, **P**=10 days*

1. **Expected Estimate**

$$\begin{aligned} TE &= (O + 4M + P) / 6 \\ &= (4 + (4 \times 6) + 10) / 6 \\ &= 38 / 6 \\ &= \underline{\mathbf{6.33}} \end{aligned}$$

2. **Standard Deviation**

$$SD = (P - O) / 6$$
$$= (10 - 2) / 6$$
$$= 8 / 6$$
$$= \underline{\mathbf{1.33}}$$

- *So, the expected time is **6.33 days**, with a standard deviation of **1.33 days**.*

6. **Function Point Analysis (FPA)**
   - Measures the size of software based on the number and complexity of inputs, outputs, files, etc.
   - Independent of programming language
   - **Pros:**
     a. Objective size estimation
   - **Cons:**
     a. Requires experience and training
7. **Use Case Point (UCP) Estimation**
   - Based on use cases and their complexity
   - Accounts for technical and environmental factors
   - **Pros:**
     a. Aligned with UML and object-oriented analysis
   - **Cons:**
     a. Can be complex to apply manually
8. **COCOMO Model (Constructive Cost Model)**
   - Algorithmic model based on LOC (Lines of Code) or Function Points
   - Based on size measured in KLOC (thousand lines of code)
   - **1 KLOC=1,000 lines of source code**
   - KLOC depends on the counting rule; comments and blanks may or may not be included.
   - **Algorithmic model** for estimating
     a. Effort (person-months)
     b. Time (months)
     c. Cost of software projects
   - **Modes**

Project Type	a	b	c	d	Description
Organic	2.4	1.05	2.5	0.38	▪ Small, simple software in a familiar environment
Semi-detached	3.0	1.12	2.5	0.35	▪ Intermediate complexity with mixed teams
Embedded	3.6	1.20	2.5	0.32	▪ Complex software with tight hardware/real-time constraints

   - **Basic COCOMO Model**
     - **Formula for Effort Estimation:**

$$\textbf{Effort (PM)} = a \times (\textbf{KLOC})^{b}$$

     - **Formula for Time Estimation:**

$$\textbf{Time (TDEV)} = c \times (\textbf{KLOC})^{d}$$

     Where:
     - **KLOC** = *Thousands of lines of code*

- **a, b, c, d** = *Constants depending on project type*

Example:

- *You are estimating a **semi-detached** project with **50 KLOC**.*

1. **Effort:**

$$\text{Effort (PM)} = a \times (\text{KLOC})^b$$
$$= 3.0 \times (50)^{1.12}$$
$$\approx 3.0 \times 99.24$$
$$\approx \underline{\textbf{297.72 person-months}}$$

2. **Time (Duration):**

$$\text{Time (TDEV)} = c \times (\text{KLOC})^d$$
$$= 2.5 \times (297.72)^{0.35}$$
$$\approx 2.5 \times 9.39$$
$$\approx \underline{\textbf{23.48 months}}$$

Comparing Techniques

Technique	Accuracy	Speed	Complexity	Best For
Expert Judgment	Medium	Fast	Low	▪ Small, undefined projects
Bottom-Up	High	Slow	Medium	▪ Well-defined projects
Three-Point (PERT)	Medium	Medium	Medium	▪ Projects with uncertainty
Function Point	High	Medium	High	▪ Software systems
COCOMO	High	Medium	High	▪ Large software projects

Strengths and Limitations of COCOM

➢ **Advantages:**

1. Based on empirical data
2. Simple to apply in basic form
3. Useful for early-stage estimation

➢ **Limitations:**

1. Assumes availability of accurate size estimates (KLOC)
2. Not suited for modern agile methodologies
3. The original model is outdated for non-structured programming environments
4. Assumes waterfall lifecycle

REFERENCES

- Pressman, Roger S, Software Engineering: A Practitioner's Approach, 9th Edition, Published by Mc Graw-Hill (2019)
- Sommerville, Ian, Software Engineering 10th Edition, Published by Pearson Education, Inc (2016)
- Bass, Len, Clements, Paul, & Kazman, Rick., Software Architecture in Practice (3rd Edition). Addison-Wesley, 2012.