Course Material No. 4

SOFTWARE ENGINEERING 2

FE LARWA-HABLANIDA
Course Instructor

PROJECT PLANNING and ESTIMATION 2

4

LEARNING OUTCOMES

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

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

RESOURCES NEEDED

For this lesson, you would need the following resources:

- 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.

DISCUSSION:

PROJECT SCHEDULING

Project scheduling is the process of listing project activities, deliverables, and milestones with planned start and finish dates.

Importance:

1. Organizes and sequences work
2. Provides a time-based roadmap
3. Enables monitoring of progress
4. Helps in resource allocation
5. Identifies project duration and deadline risks

Components of a Project Schedule

Component	Description
Activities/Tasks	▪ Specific work units to be completed
Milestones	▪ Key dates indicating major progress points
Dependencies	▪ Task relationships (e.g., finish-to-start)
Duration	▪ Time required to complete tasks
Start/End Dates	▪ When a task begins and ends
Resources	▪ People, tools, or materials needed

Task Dependencies

1. **Finish-to-Start (FS)** – Task B starts after Task A ends
2. **Start-to-Start (SS)** – Task B starts when Task A starts
3. **Finish-to-Finish (FF)** – Task B ends when Task A ends
4. **Start-to-Finish (SF)** – Rare; Task B ends when Task A starts
5. **Visual**: Simple dependency diagram

Scheduling Techniques

1. **Gantt Charts**
   - Bar chart showing tasks over time
   - Visualizes start/end dates, durations, and overlaps
   - Easy to update and interpret
   - **Tool:** *MS Project, Excel, Trello with timelines*

2. **Critical Path Method (CPM)**
   - Identifies the longest sequence of dependent tasks
   - Helps determine **project duration**
   - Critical path = **tasks with zero slack**

➢ **Key Concepts in CPM**

Term	Description
Activity	▪ A specific task or unit of work
Event/Milestone	▪ A point marking the start or end of one or more activities
Duration	▪ The estimated time required to complete an activity
Critical Path	▪ The longest path through the project network with zero slack
Slack/Float	▪ The amount of time an activity can be delayed without affecting the project deadline
Earliest Start (ES)	▪ The earliest time an activity can begin
Earliest Finish (EF)	▪ The earliest time an activity can end
Latest Start (LS)	▪ The latest time an activity can start without delaying the project
Latest Finish (LF)	▪ The latest time an activity can finish without delaying the project

➢ **Steps:**

1. **List all tasks**
   - Identify all project activities
2. **Estimate durations**
   - Identify all project activities and their durations
3. **Identify dependencies**
   - Identify dependencies between activities
4. **Draw the Network Diagram**
   - Use nodes (or arrows) to represent tasks
   - Arrows show dependencies (precedence relationships)
5. **Calculate early start (ES), late start (LS), and slack**
   - Calculate **Forward Pass** (ES and EF)

$$EF = ES + \text{Duration}$$

   - Calculate **Backward Pass** (LS and LF)

$$LS = LF - \text{Duration}$$

   - Determine **Slack**
     - If Slack = 0 → The task is on the **Critical Path** (no delay allowed).

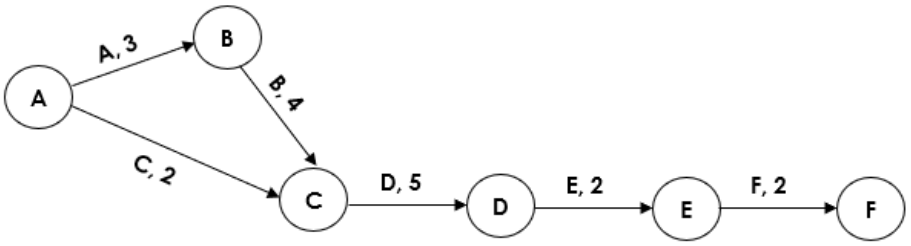
$$\text{Slack} = LS - ES = LF - EF$$

Example:

- *Project Activities*

Activity	Description	Duration (days)	Predecessor
A	Requirements Gathering	3	—
B	Design UI	4	A
C	Set Up Database	2	A
D	Develop Backend	5	B, C
E	Testing	2	D
F	Deployment	2	E

- *Network Diagram*



- *Paths and Durations*
  1. **A → B → D → E → F**

     Duration: 3 + 4 + 5 + 2 + 2 = **16 days**

  2. **A → C → D → E → F**

     Duration: 3 + 2 + 5 + 2 + 2 = **14 days**

- *Critical Path:* **A → B → D → E → F**
  - **Minimum Project Duration: 15 days**
  - Any delay in these tasks delays the whole project.
- *Forward Pass*

  EF = ES + Duration

  Activity A
  - ES = 0 (Start of project)
  - EF = 0 + 3 = **3**

  Activity B
  - ES = EF of A = 3
  - EF = 3 + 4 = **7**

  Activity C
  - ES = EF of A = 3
  - EF = 3 + 2 = **5**

Activity D

- ES = max(EF of B, EF of C) = max(7, 5) = 8
- EF = 7 + 5 = **12**

Activity E

- ES = EF of D = 12
- EF = 12 + 2 = **14**

Activity F

- ES = EF of E = 14
- EF = 14 + 2 = **16**

- *Backward Pass*

Activity F

- LF = 16 (Project ends here)
- LS = 16 - 2 = **14**

Activity E

- LF = LS of F = 14
- LS = 14 - 2 = **12**

Activity D

- LF = LS of D = 12
- LS = 12 – 5 = **7**

Activity C

- D is a successor. LS of D = 7
- LF = LS of D = 7
- LS = 7 - 2 = **5**

Activity B

- D is a successor. LS of D = 7
- LF = LS of D = 7
- LS = 7 - 4 = **3**

Activity A

- B and C are successors.
- LS of B = 3, LS of C = 5
- LF = min (3, 5) = 3
- LS = 3 - 0 = **3**

- *Slack*

Activity A

- Slack = LS−ES = 0 – 0 = 0

Activity B

- Slack = LS−ES = 3 – 3 = 0

Activity C

- Slack = LS−ES = 5 – 3 = 2

Activity D

- Slack = LS−ES = 7 – 7 = 0

Activity E

- Slack = LS−ES = 12 – 12 = 0

Activity F

- Slack = LS−ES = 14 – 14 = 0

Activity	Duration (days)	ES	EF	LS	LF	SLACK
A	3	0	3	0	3	0
B	4	3	4	3	7	0
C	2	3	5	5	7	2
D	5	7	12	7	12	0
E	2	12	14	12	14	0
F	2	14	16	14	16	0

- *Interpretation*
  - **Slack = 0**: Activities A, B, D, E, and F are on the **Critical Path**
  - **Activity C** has **2 days of slack** → it can be delayed by 2 days **without delaying** the project

3. **Program Evaluation Review Technique (PERT)**
   - PERT is a project management tool used to plan, schedule, and control complex projects by analyzing the time needed to complete each task and identifying the minimum time required to complete the entire project.
   - It is like CPM but uses **three-point estimation**
   - Useful for estimating time in uncertain environments

↳ Project Scheduling Tools

Tool	Features
MS Project	▪ Advanced scheduling, resource tracking
Trello	▪ Visual task boards (w/ Power-Up)
GanttProject	▪ Open-source Gantt chart tool
Excel	▪ Manual scheduling
Primavera	▪ Enterprise-grade scheduling

PROJECT RISK IDENTIFICATION

A **risk** is an uncertain event or condition that, if it occurs, will have a positive or negative impact on a project's objectives.

- **Importance in Planning:**
  1. Risk management is proactive, not reactive
  2. Help reduce surprises and improve success rates
  3. Supports realistic schedules, budgets, and resource plans

Types of Project Risks

Category	Examples
Technical	▪ Technology failure, unrealistic requirements
External	▪ Legal issues, regulatory changes
Organizational	▪ Resource availability, internal politics
Project Management	▪ Scope creep, poor estimates
Operational	▪ Process inefficiencies, infrastructure issues

Risk Management Process

The process is iterative and ongoing throughout the project.

1. **Risk Identification**

Risk Identification is the process of systematically *identifying potential risks*—both *threats and opportunities*—that could impact the success of a project. It is an early and ongoing activity in project risk management, forming the foundation for later analysis and response planning.

Example Risks Identified:

Risk Description	Type
Key developer may leave mid-project	▪ Human resource risk
Delay in supplier delivery	▪ Schedule risk
New technology may not integrate well	▪ Technical risk
Change in regulatory laws	▪ External risk

- **Methods of Risk Identification:**

a. **Brainstorming With the Project Team**

Brainstorming with the project team is a structured or informal group activity used during *project planning to generate ideas, identify potential risks, define project requirements, or solve problems collaboratively*.

It promotes team participation, creativity, and shared ownership of project outcomes.

b. **SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats)**

SWOT Analysis is a strategic planning tool used to identify and analyze the internal and external factors that can impact a project, product, team, or organization.

It helps project managers and stakeholders make *informed decisions* by understanding *Strengths, Weaknesses, Opportunities*, and *Threats* related to the project or initiative.

Example:

- **SWOT Analysis for a Web App Development Project**

Strengths	**Weaknesses**
▪ *Experienced developers*	▪ *Tight budget*
▪ *Clear project requirements*	▪ *Limited testing time*
Opportunities	**Threats**
▪ *Rising demand for digital tools*	▪ *New government data regulations*
▪ *Potential to scale product*	▪ *Competitors with similar apps*

c. **Expert Interviews**

Expert Interviews are project estimation and information-gathering techniques where project managers and planners consult individuals with specialized knowledge, skills, or experience to obtain insights on the scope, risks, costs, schedule, or other aspects of a project.

Sample Questions in an Expert Interview:

- *"What are common pitfalls in projects like this?"*
- *"How long does it usually take to implement this module?"*
- *"What team composition would you recommend?"*
- *"Are there any hidden costs or effort areas we should plan for?"*

d. **Checklist-Based Analysis**

Checklist-Based Analysis is a systematic project planning and risk identification technique where a project team uses a *predefined checklist of common items* (e.g., risks, activities, deliverables, or requirements) to ensure nothing important is overlooked during planning or analysis.

It is a simple but effective method for improving the completeness and quality of project plans, especially in the early stages.

Example:

- **Risk Management Checklist (Partial)**

- ☐ Have technical risks been identified?
- ☐ Are supplier and vendor risks considered?
- ☐ Are there legal or compliance risks?
- ☐ Has the team assessed cybersecurity risks?

e. **Historical Data and Lessons Learned**

Historical Data and Lessons Learned are **project management tools** used during **project planning and estimation** to improve accuracy, reduce risks, and avoid repeating past mistakes. These come from **past projects** and are part of a continuous improvement process.

- **Historical Data**

Historical data refers to documented information from *previous, similar projects*, such as:

1. Actual vs. estimated time, cost, and effort
2. Resource utilization rates
3. Risks encountered and how they were addressed
4. Performance of vendors or tools
5. Changes and issues logged

Example of Historical Data Use:

- *If a similar web app project took 6 months and cost $120,000 last year, that information can be used to estimate the scope, timeline, and budget of a new project with similar features.*

- **Lessons Learned**

Lessons Learned are the *insights and reflections* from a completed project — what went well, what went wrong, and why.

These are typically documented in a **Lessons Learned Report** at the end of a project and may include:

Category	Examples
Successes	▪ Early stakeholder engagement avoided scope creep.
Failures	▪ Underestimated testing time by 30%.
Recommendations	▪ Use Agile for future updates to handle changing requirements.

Example Entry in Lessons Learned Document:

- **Issue**: Inadequate stakeholder involvement
- **Impact**: Rework of 25% of requirements
- **Lesson**: Engage stakeholders biweekly in review sessions

- **Recommendation**: Include client demo checkpoints in the schedule

➢ **Risk Register:**

A **Risk Register** (also called a **Risk Log)** is a *formal document* used to **identify**, **assess**, and **track risks** throughout the life of a project. It is a *central tool* in risk management that helps teams *document risks, monitor their status*, and plan appropriate responses.

❖ **Purpose of a Risk Register**

a) To record all identified risks systematically
b) To track risk probability, impact, and priority
c) To plan and document risk responses and ownership
d) To monitor changes in risk status over time

❖ **A document that records:**

a) Identified risks
b) Risk category/source
c) Potential impact
d) Owner/responsible party

❖ **Example of Risk Register Entry**

Risk ID	Description	Category	Prob.	Impact	Priority	Response	Owner	Status
R-01	Delay in supplier delivery	Schedule	High	High	Critical	Mitigate: Source backup vendor	Procurement Team	Open
R-02	The developer may resign	HR	Medium	High	High	Mitigate: Crosstrain team	PM	Open
R-03	Regulatory change	External	Low	Medium	Medium	Monitor: Regular compliance checks	Legal Dept.	Open

2. **Risk Analysis Techniques**

Risk Analysis Techniques are methods used to evaluate the potential impact of risks on a project's success. These techniques help project managers identify which risks are most critical, understand how they may impact the *project's scope, time, cost, or quality*, and determine the necessary *actions to take.*

- **Qualitative Risk Analysis**

Qualitative Risk Analysis is the process of evaluating identified risks based on their *probability of occurrence* and *potential impact—without using complex numerical data*. It helps project managers prioritize risks quickly so that they can focus on the most significant ones.

- Subjective analysis based on likelihood and impact; quick, uses expert judgment
- It is typically done after risk identification and before (or instead of) quantitative analysis.
- Prioritize risks based on:
  a) **Probability** (Likelihood of occurrence)
  b) **Impact** (Severity of consequences)
- Often visualized using a **Probability-Impact Matrix**

	LOW IMPACT	MEDIUM IMPACT	HIGH IMPACT
Low Prob.	Low Risk	Low Risk	Moderate Risk
Med Prob.	Low Risk	Moderate Risk	Critical Risk
High Prob.	Moderate Risk	High Risk	Critical Risk

Example:

- *Entry in Risk Register*

RISK ID	DESCRIPTION	PROBABILITY	IMPACT	PRIORITY	CATEGORY	NOTES
R-01	Key team member quits	High	High	Critical	HR	Requires a mitigation plan
R-02	Minor spec change	Medium	Low	Low	Scope	Acceptable risk
R-03	New tool integration	Low	High	Medium	Technical	Monitor during implementation

- **Quantitative Risk Analysis**

Quantitative Risk Analysis is a data-driven process used to numerically evaluate the impact of identified project risks on overall objectives, such as *cost, time*, and *scope*. Unlike qualitative analysis (which is subjective), quantitative analysis uses statistical and mathematical techniques to estimate the probability of different outcomes.

- Data-driven uses numerical models and simulations to measure risk effects
- Use numerical methods to estimate impact (e.g., Monte Carlo Simulation, EMV)

❖ **Expected Monetary Value (EMV)**:

EMV=Probability × Impact

- It helps estimate cost or schedule buffers needed for specific risks

Example:

- *If the probability of a server crash is 20% and it would cost $10,000, then:*

EMV = 0.2 x 10,000 = **$2,000**

Example:

- *A construction project has identified 3 cost risks. Using EMV:*

Risk	Probability	Impact (₱)	EMV (₱)
Equipment Failure	30%	100,000	30,000
Weather Delays	20%	80,000	16,000
Design Rework	10%	150,000	15,000
Total Contingency			**₱61,000**

3. **Risk Response Planning**

Risk Response Planning is the process of *developing strategies and actions* to address project risks — both *threats* (negative risks) and *opportunities* (positive risks) — to *minimize impact on objectives* or *maximize potential gains*.

It is part of the Project Risk Management process group and follows risk identification and analysis (qualitative and/or quantitative).

➢ **Negative Risks (Threats):**

Strategy	Description
Avoid	▪ Eliminate the cause of the risk
Mitigate	▪ Reduce probability/impact
Transfer	▪ Shift impact to a third party (e.g., insurance)
Accept	▪ Acknowledge and monitor (no action)

➢ **Positive Risks (Opportunities):**

Strategy	Description
Exploit	▪ Ensure opportunity occurs
Enhance	▪ Increase likelihood/benefit
Share	▪ Allocate to a partner/third party
Accept	▪ Take advantage if it arises

Example:

- *Risk Response Plan*

Risk ID	Description	Strategy	Response Action	Risk Owner	Status
R-01	Server crash risk	Mitigate	Install redundancy and auto-backup	IT Manager	Planned
R-02	Faster supplier lead	Exploit	Prioritize the early purchase order	Procurement	In progress
R-03	Delay due to rain	Accept	Add 5 days to the schedule buffer	Project Leader	Implemented
R-04	Code reuse opportunity	Enhance	Reuse tested modules from past apps	Dev Leader	Approved

4. **Risk Monitoring and Control**

Risk Monitoring and Control is the process of *tracking identified risks, monitoring residual risks, identifying new risks,* and *evaluating the effectiveness of risk response strategies* throughout the project life cycle.

It ensures that risk management remains active and responsive, not just a one-time task during planning.

➢ **Activities Involved**

Activity	Description
Track Identified Risks	▪ Review current risks and evaluate their status
Identifying New Risks	▪ Add newly discovered risks to the register
Reassess Existing Risks	▪ Update probability, impact, or ranking based on project changes
Evaluating Risk Responses	▪ Analyze if mitigation, avoidance, or contingency actions are working
Monitor Triggers & Thresholds	▪ Watch for early warning signs to initiate fallback or contingency responses
Control Residual Risks	▪ Monitor leftover risk after the response has been implemented
Update Risk Register	▪ Revise response plans, ownership, or status of each risk
Generate Reports	▪ Provide risk-related updates to stakeholders

- **Sample Risk Monitoring Log Entry**

Risk ID	Description	Status	Action Taken	Effectiveness	Owner	Next Steps
R-01	Supplier delay	Active	Changed supplier	Highly Effective	Procurement	Monitor the next batch
R-02	Team member turnover	Realized	Trained backup developer	Partially Effective	HR Manager	Conduct a team survey
R-03	Hardware failure	Closed	Installed backup system	Effective	IT Lead	Review quarterly

MANAGEMENT IN PLANNING

Management in Project Planning refers to the activities, decisions, and leadership actions that guide the development of the project plan and ensure alignment with project objectives.

- **Key Responsibilities:**

1. Leading planning meetings
2. Aligning with organizational goals
3. Balancing constraints (scope, time, cost)
4. Assigning roles and resources
5. Ensuring stakeholder engagement

Core Planning Areas That Require Management

1. **Scope Management**

Scope Management is the process of *defining, controlling*, and *validating* all the work required to complete a project successfully — and only that work. It ensures that the *project includes all the necessary tasks* to meet the objectives and *prevents scope creep*, which is the uncontrolled expansion of work.

- Define boundaries of work
- Validate requirements
- Prevent scope creep

Example:

- *A client adds extra features not included in the original contract during development without changing the deadline or budget limitation. This is **scope creep** and must be controlled.*

2. **Time Management**

Time Management in project management refers to the process of *planning, estimating, scheduling*, and *controlling the time* required to complete project activities. It ensures that the *project is delivered on time*, within deadlines, and with efficient use of resources.

- Develop schedules
- Monitor task dependencies
- Apply buffers if necessary

3. **Cost Management**

Cost Management is the process of *planning, estimating, budgeting, financing, funding, managing*, and *controlling* project costs so that the project can be completed within the approved budget.

It ensures that the *project stays financially viable* and does not exceed the allocated resources

- Approve budgets
- Ensure accurate estimates
- Plan for contingencies

4. **Resource Management**

Resource Management is the process of *planning, allocating, managing*, and *optimizing the use of resources* (people, tools, equipment, facilities, materials, etc.) needed to complete a project efficiently and successfully.

It ensures that the *right resources are available at the right time*, in the right amount, and with the right skills.

- Allocate team members
- Manage workload balance
- Adjust for availability

➢ **Types of Resources in Projects**

Resource Type	Examples
Human Resources	▪ Project manager, developers, testers, analysts
Physical Resources	▪ Equipment, tools, materials, facilities
Financial Resources	▪ Budget, cash, funding
Time Resources	▪ Availability of people and tools (e.g., work hours)

➢ **Example of Resource Allocation**

Activity	Required Resources	Duration
Design UI	1 UX Designer	5 days
Develop Frontend	2 Developers, 1 Laptop	7 days
Testing	1 QA, 1 Test Server	4 days

5. **Quality Planning**

Quality Planning is the process of identifying *quality requirements and standards* for the project and its deliverables and documenting how the project will demonstrate compliance with those standards.

It is a *proactive approach* that ensures quality is built into the product or service from the start—not inspected later.

- Define quality metrics and standards
- Set acceptance criteria
- Plan quality assurance activities

➢ **Example in a Software Project**

Quality Goal	Metric	Method
< 1% defect rate	▪ Defect rate during testing	▪ Automated & manual testing
100% requirement coverage	▪ Test coverage report	▪ Requirements traceability
90% customer satisfaction	▪ Post-launch survey	▪ Feedback forms and reviews

Stakeholder Management in Planning

Stakeholder Management in Planning is the process of identifying project stakeholders, analyzing their *needs and expectations*, and developing strategies to *engage and communicate* with them effectively throughout the project lifecycle — especially during planning.

It ensures that all key individuals or groups influencing or affected by the project are considered and managed to improve support and reduce resistance.

➢ **Key Processes in Stakeholder Management**

Strong stakeholder engagement improves planning accuracy and commitment.

Stakeholder Activity	Manager's Role
Identify stakeholders	▪ Conduct stakeholder analysis
Determine needs and influence	▪ Engage early and frequently
Communication plans and changes	▪ Ensure transparency
Address concerns and conflicts	▪ Use negotiation and mediation

Tools and Techniques Used in Planning Management

In Project Planning Management, various **tools and techniques** are used to help project managers and teams *define, organize, estimate, schedule*, and *control project* activities effectively. These tools help ensure that the plan is comprehensive, realistic, and achievable.

➢ Tools and Techniques

Tool/Technique	Purpose
Work Breakdown Structure (WBS)	▪ Decompose scope into tasks
Gantt Charts	▪ Schedule visualization
Responsibility Matrix (RACI)	▪ Clarify roles/responsibilities
Risk Register	▪ Document and track risks
Project Charter	▪ Align planning with project authority
Kickoff Meeting	▪ Align team on objectives and deliverables

Iteration and Change Management in Planning

Iteration and Change Management in Planning refers to the structured approach of *revisiting, revising*, and *adapting* project plans in response to feedback, evolving requirements, risks, or external factors. It is critical in *agile, adaptive*, and even *predictive* project environments to ensure that the plan remains realistic, relevant, and aligned with project goals.

Change Management is the process of identifying, evaluating, approving (or rejecting), and incorporating **changes** into the project plan in a **controlled** and **systematic** manner.

- Planning is not static; it evolves with new information
- Change requests must follow a formal process
- Maintain a change log and evaluate the impact on scope, time, and cost
- Update plans and communicate changes accordingly

➢ Change Management Tools

1. **Change Request Form**

A **Change Request Form** is a formal document used in project management to propose a change to any aspect of a project—such as its *scope, schedule, cost, quality*, or *resources*. It is a critical tool in Change Management processes, allowing changes to be *identified, evaluated, approved or rejected*, and then *tracked*.

➢ **Typical Contents of a Change Request Form**

Section	Description
Change Request ID	▪ Unique identifier for tracking
Date Submitted	▪ Date the request is made
Requested By	▪ Name and role of the person submitting the request
Project Name	▪ Name of the affected project
Change Description	▪ Clear explanation of the change requested
Reason for Change	▪ Why this change is needed
Type of Change	▪ Scope, Schedule, Cost, Quality, Resource, Risk, etc.
Priority Level	▪ High, Medium, Low
Impact Analysis	▪ Expected effects on time, cost, quality, risk, etc.
Alternatives Considered	▪ Other solutions or options evaluated
Approval Section	▪ For decision-makers (Approved, Rejected, Deferred)
Comments/Notes	▪ Additional details or justification
Implementation Plan	▪ (Optional) How the change will be integrated

2. **Change Log / Register**

A **Change Log** or **Change Register** is a formal *record-keeping tool* used in project management to *track all change requests* throughout the lifecycle of a project—whether they are *approved, rejected*, or *pending*. It ensures transparency and provides a single source of truth for project changes.

➢ **Typical Fields in a Change Log**

Field	Description
Change ID	▪ Unique number (e.g., CR-001)
Date Requested	▪ When the change was proposed
Requested By	▪ Name and role of requester
Change Description	▪ Summary of the requested change
Category/Type	▪ Scope, Cost, Time, Quality, Resource, etc.
Impact Summary	▪ Effect on budget, timeline, scope, risk, etc.
Status	▪ Open, In Review, Approved, Rejected, Implemented
Decision Date	▪ When the change was approved/rejected
Approver	▪ The person or committee that made the decision
Comments	▪ Notes on progress or implementation

3. **Impact Assessment Matrix**

An **Impact Assessment Matrix** is a decision-support tool used to evaluate and prioritize the *effects of changes, risks*, or *issues* on a project. It helps project managers and stakeholders *assess the magnitude and likelihood* of impacts on critical project dimensions like *scope, schedule, cost, quality, resources*, or *risk*.

➢ **Typical Layout of an Impact Assessment Matrix**

- For Change Requests or Risks:

Impact Level	Scope	Time	Cost	Quality	Risk Level
Low	✗	☑	☑	☑	Low
Medium	☑	☑	✗	☑	Medium
High	☑	☑	☑	✗	High

Legend: ☑ = Some Impact, ✗ = Major Impact

4. **Change Control Board (CCB)**

A **Change Control Board (CCB)** is a formal *group of stakeholders* and *decision-makers* in a project who are responsible for *reviewing, evaluating, approving*, or *rejecting change requests*. It is a core component of change management in project planning.

➢ **Members of the CCB**

- Membership varies by project but often includes:

Role	Responsibility
Project Manager	▪ Facilitates CCB meetings, tracks changes
Product Owner/Client Rep	▪ Represents stakeholder priorities
Sponsor	▪ Provides funding decisions
Technical Lead/Architect	▪ Assesses technical feasibility
Quality Assurance	▪ Checks if the change meets standards
Risk Manager	▪ Evaluates risks introduced by change

➢ **Sample CCB Decision Table**

Change ID	Description	Impact Summary	CCB Decision	Date	Comments
CR-012	▪ Add login via Google	▪ +1 dev day	▪ Approved	▪ Aug 2	▪ Add to next sprint
CR-015	▪ Upgrade database engine	▪ Risk of downtime	▪ Rejected	▪ Aug 5	▪ Unnecessary risk
CR-018	▪ Delay launches by 2 weeks	▪ High client value	▪ Deferred	▪ Aug 10	▪ Wait for client approval

5. **Version Control Systems**

A **Version Control System (VCS)** is a software tool that helps teams *manage changes to source code, documents*, or *project files over time*. It tracks every modification made, who made it, and why—allowing teams to *collaborate, roll back errors*, and *maintain history*

➢ **Popular Version Control Tools**

Tool	Type	Use Case
Git	Distributed	▪ Code & document versioning
GitHub / GitLab / Bitbucket	Git hosting	▪ Online collaboration, CI/CD
Subversion (SVN)	Centralized	▪ Legacy or document-heavy projects
Azure DevOps	Mixed	▪ Enterprise-level planning and development
TFS (Team Foundation Server)	Centralized	▪ Microsoft enterprise project

6. **Project Management Information Systems (PMIS)**

A **Project Management Information System (PMIS)** is a software-based system used to support project planning, execution, monitoring, and closure. It integrates tools, processes, and data to help *project managers and teams make decisions, manage resources*, and *track progress* across the project life cycle.

➢ **Examples of PMIS Tools**

PMIS Tool	Description
Microsoft Project	▪ Popular for task scheduling, Gantt charts, and resource leveling
Primavera P6	▪ Enterprise-grade PMIS for large infrastructure projects
Smartsheet	▪ Spreadsheet-based tool for project tracking
Asana / Trello	▪ Simple, agile-friendly PM tools
Jira	▪ Widely used in software development for agile projects
ClickUp / Monday.com	▪ Visual, flexible platforms for collaboration and reporting

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.