Software Project Management Plan (SPMP)

Based on IEEE 1058 - Adapted for Student Projects

Project Title: [Enter Project Name]

Team Name & Members: [List Members]

Advisor: [Instructor Name]

Version: [e.g., 1.0]

Date: [dd-mm-yyyy]

Table of Contents

1. Project Overview

1.1 Purpose, Scope, and Objectives

• State the purpose of the project and why it is being developed.
• Define the project scope (what features are included and excluded).
• List project objectives clearly (e.g., deliver a working software prototype by Week 15).

1.2 Assumptions, Constraints, and Risks

• Mention assumptions (e.g., all team members have laptops, internet access).
• Note constraints (e.g., time limited to 15 weeks, only open-source tools allowed).
• Identify 3–5 top risks (e.g., missed deadlines, lack of technical skills).

1.3 Project Deliverables

• List all deliverables: Proposal, SPMP, SRS, SDS, STS, Code, Presentation.
• Include expected submission dates.
• Mention format (Word, PDF, Code repository).

1.4 Schedule Summary

• Provide a high-level project schedule.
• Use milestone chart or Gantt chart if possible.
• Show deadlines for major deliverables.

1.5 References

• List standards (IEEE 1058), course materials, textbooks, or research papers.
• Example: Sommerville, I. Software Engineering, 10th Edition.

1.6 Definitions and Acronyms

Define, or provide references to documents or annexes containing the definition of all terms and acronyms required to properly understand this Plan.

2. Project Organization

2.1 External Interfaces

• Identify external stakeholders such as client.
• Define their role in feedback, approval, or evaluation.

• Use organizational charts or diagrams to depict the project's external interfaces.

2.2 Internal Structure

• Describe the team's internal organization.
• Show team chart or list communication structure (leader, members, responsibilities).

•Use organizational charts or diagrams to depict the lines of authority, responsibility and communication within the project.

2.3 Roles and Responsibilities

•Identify and state the nature of each major work activity and supporting process.

•Assign clear roles to each member.
• Example roles: Project Manager, Developer, Tester, Documentation Lead.

Team Member	Role	Responsibility
Example: John Doe	Developer	Implements backend services

3. Managerial Process Plans

3.1 Estimates & Staffing

• Provide rough time estimates per project phase.
• Indicate number of team members working on each part.
• Example: 2 members for coding, 1 for testing, 1 for documentation.

3.2 Work Plan

• Work Breakdown Structure (WBS): List main activities (Requirements, Design, Coding, Testing).

•Decompose the work activities to a level that exposes all project risk factors, and that allows accurate estimation of resource requirements and schedule duration for each work activity.
• Specify the following factors for each work activity:

- necessary resources,
- estimated duration,
- products or deliverables of the activity,
- acceptance criteria for the work activity products, and
- predecessor and successor work activities.

3.3 Project Tracking Plan

• State how progress will be monitored.
• Mention how changes are approved.
• Metrics: % tasks completed, number of bugs fixed, etc.

3.4 Risk Management Plan

• Identify risks, their likelihood and impact, and how to mitigate them.
• Example: Risk = Team member unavailable → Mitigation = Reassign tasks.

Risk	Likelihood	Impact	Mitigation
Example: Delay in testing	Medium	High	Add buffer time and assign extra tester

3.5 Closeout Plan

• Explain how the project will end.

4. Technical Process Plans

4.1 Process Model

• Identify chosen process model (Agile, Waterfall, Spiral).
• Justify why it suits your project (e.g., If Agile ,then why?).

4.2 Methods, Tools, and Techniques

• List languages, frameworks, version control, testing tools.

4.3 Infrastructure

• Describe hardware/software environment.

4.4 Product Acceptance

• Define acceptance criteria (all features must pass testing).
• State who approves final product .

5. Supporting Process Plans

5.1 Documentation Plan

• List all required documents (Proposal, SPMP, SRS, SDS, STS, Reports).
• Assign responsibility for preparing and reviewing them.

5.2 Quality Assurance

• Define coding standards, peer reviews, and test coverage.

5.3 Configuration Management

• Define version control policy (branching, commits, pull requests).

5.4 Problem Resolution

• Define how bugs/issues will be reported and resolved.

6. Additional Plans (Optional)

- Training: If team members need new skills.
- Security: How sensitive data is handled.
- Maintenance: Future updates after submission.

7. Appendices

A. Glossary – Define project-specific terms.
B. Gantt chart – Attach project timeline.
C. WBS diagram – Visualize work breakdown.

Evaluation

To be completed by the instructor or supervisor.

3. Project SPMP (18 points / 3 marks) – Week 8

Section	Excellent (Full Points)	Good (75%)	Fair (50%)	Poor (25% or below)	Points
1. Project Overview (3 pts)	Clear purpose, well-defined scope, objectives measurable, risks realistic, deliverables & schedule complete.	Purpose/scope adequate, some objectives unclear, limited risks, minor gaps in deliverables/schedule.	Vague or incomplete scope, objectives weak, little risk identification.	Missing or irrelevant.	/3
2. Project Organization (2 pts)	Well-structured team org, clear roles/responsibilities, external interfaces defined.	Team roles mostly clear, some vagueness in structure or interfaces.	Roles incomplete or unclear, poor organization.	Missing or irrelevant.	/2
3. Managerial Process Plans (5 pts)	Estimates realistic, WBS & schedule clear, tracking plan defined, risks detailed with mitigation, closeout plan solid.	Most elements present but some weak (e.g., vague WBS or shallow risk analysis).	Several parts incomplete or poorly justified.	Very minimal or missing.	/5
4. Technical Process Plans (3 pts)	Process model justified, tools/methods listed, infrastructure described, acceptance criteria realistic.	Adequate but some parts generic or weak justification.	Vague process model, missing tools/infrastructure details.	Very poor or absent.	/3
5. Supporting Process Plans (3 pts)	Documentation plan complete, QA standards clear, version control explained, bug	Most parts covered but not in detail.	Minimal coverage (e.g., only documentation).	Missing or irrelevant.	/3

Section	Excellent (Full Points)	Good (75%)	Fair (50%)	Poor (25% or below)	Points
	tracking method defined.				
6. Additional Plans (1 pt)	At least one plan (training, security, maintenance) included.	Mentioned but vague.	Minimal or irrelevant.	Not included.	/1
7. Overall Quality (1 pt)	Well-written, well-structured, free of major grammar/formatting issues, follows template fully.	Clear but some formatting/clarity issues.	Hard to follow, weak formatting.	Poorly presented or incomplete.	/1

Note: If a student is listed as a member at the beginning of the SPMP **but not mentioned within the tasks, roles, or responsibilities throughout the document**, that student **will not receive the same score as active contributors**.

- Marks for that student may be reduced to reflect lack of contribution.
- Instructors should verify **role allocation tables, WBS assignments, and responsibilities** for evidence of individual involvement