Experiment No.: 2

Title: Preparing Software Project Management Plan (SPMP)

Batch: B–1 **Roll No.: 16010422234** **Experiment No.: 2**

Aim: To Prepare Software Project Management Plan (SPMP)

Resources needed: Internet Explorer, GanttProject, LaTex Editor

Theory

The software project management plan specifies:

- the objectives of the project,
- the project dependencies and constraints,
- the project deliverables, and
- the project timetable.

This document describes how you intend to develop your software; how you expect to approach the development process; the times and deliverables involved; problems foreseen and constraints imposed. The plan accompanies the project life span and is usually subject to a number of revisions.

SPMP Template:

1 INTRODUCTION

1.1 Project Overview
Provide a description of the purpose of the software, who the project is being executed for and the expected delivery date.

1.2 Project Deliverables
List the project deliverables/"work products" (documents, source code, library files, executable code, databases), and the delivery date for each (the date that each is to be complete).

2 PROJECT ORGANIZATION

2.1 Software Process Model
Identify and describe the software process model you will use in terms of the following:
? ? the flow of information and work products,
? ? reviews to be conducted,
? ? major milestones to be achieved,

? ? versions to be established,
? ? project deliverables to be completed,

To describe the process model, a combination of graphical and textual notations may be used.

2.2 Roles and Responsibilities

In the case of a group project, identify the various roles, the project team members, and their assignments. Take care to identify how information will be communicated between the different roles – for example: how will the design be passed to the coder.

Use diagrams to show the project structure, and the lines of communication within the project.

2.3 Tools and Techniques

Specify the development methodologies, notations, programming languages, techniques, and tools you plan to use at each stage of the project.

For example: you might use OO for the analysis phase, using the UML notation, and the Rational Rose Tool. In order to track changes to the analysis, you might use version control, and the CS-RCS tool.

This section should include details of the coding standards to be used.

3 PROJECT MANAGEMENT PLAN

3.1 Tasks

Identify the tasks involved in executing the project. Some sample tasks are:

? ? requirements analysis/clarify the requirements

? ? requirements analysis/develop an exploratory prototype and get feedback from the user

? ? requirements analysis/write the SRS

? ? system design/develop the design using Rational Rose

? ? system design/write the SDD

? ? development/develop version 1

? ? development/develop version 2

? ? system test/test version 1

? ? system test/test version 2

Tasks should be specified in enough detail to allow estimation of the time required, and to allow tracking of project status. As a very rough guide, tasks should take about one week.

3.1.n Task-n

Provide a name for the task, and a unique identifier of the form SPMP.

3.1.n.1 Description

A brief description of the task.

3.1.n.2 Deliverables and Milestones

A list of the deliverables of the task (documentation, code, other), and the milestones associated with these. For example, milestone 1 of the project might be the delivery of the final revisions of the SRS, STD, and SPMP; milestone 2 might be delivery of the STD, milestone 3 might be completion of version 1 (development & test), etc.

3.1.n.3 Resources Needed

Identify the resources needed to execute the task (equipment, access to papers/books, etc.)

3.1.n.4 Dependencies and Constraints

Identify the dependencies that must be met before the task can be started or completed, and any constraints placed on the task.

For example: detailed design can't start until the requirements are complete, or there is a constraint that Java be used as the programming language.

3.1.n.5 Risks and Contingencies

Identify the risks associated with getting this task completed on time, and the contingency plans to cope with this risk. You can never identify all the risks; and some events would just stop the project completely. Limit this section to reasonable risks.

For example: what is the probability of you being unable to get the programming language you plan on using running on your computer? What would you do to recover?

3.2 Assignments

For group projects, identify the assignment of team members to tasks.

3.3 Timetable

Provide a timetable showing the estimated start and completion dates for each task. This is best illustrated using a **Gantt chart diagram**.

This timetable should be updated throughout the project to reflect changes in: the tasks, the execution time for tasks, the completion dates, etc.

Procedure:

1) Prepare a Gantt chart using Gantt Project.
2) Prepare SPMP document for chosen problem definition in LaTeX.

Results: SPMP Document in given format

LaTeX Code:

```
\documentclass[a4paper,12pt]{article}
\usepackage{graphicx}
\usepackage{geometry}
\geometry{margin=0.5in}

\title{Software Project Management Plan (SPMP)}
\author{Project Team: Chandana Galgali, Prachi Gandhi, Mahek Thakkar, Harsh Singwi}
\date{\today}

\begin{document}
```

```latex
\maketitle

\section{Introduction}
\subsection{Project Overview}
This project aims to develop a text-to-VR environment generator enabling users to create immersive VR experiences through textual descriptions. The system integrates Generative AI and VR technologies to achieve this. Expected delivery date: 3.5 months from project initiation.

\subsection{Project Deliverables}
\begin{itemize}
   \item SPMP Document
   \item Source Code (AI and VR modules)
   \item Prototype Application
   \item User Guide
   \item Testing Reports
\end{itemize}

\section{Project Organization}
\subsection{Software Process Model}
The Incremental Model will be used, involving:
\begin{itemize}
   \item Weekly progress reviews.
   \item Milestones: Requirements, Prototype, Testing, and Deployment.
   \item Versions: Alpha, Beta, and Final.
\end{itemize}

\subsection{Roles and Responsibilities}
\begin{itemize}
   \item \textbf{Mahek Thakkar}: Requirements Analysis
   \item \textbf{Chandana Galgali}: AI Model Development
   \item \textbf{Prachi Gandhi}: VR Prototype Development
   \item \textbf{Harsh Singwi}: Integration and Testing
\end{itemize}

\subsection{Tools and Techniques}
\begin{itemize}
   \item \textbf{Languages}: Python (AI), Unity/C# (VR)
   \item \textbf{Tools}: TensorFlow, Unity, GitHub, GanttProject
   \item \textbf{Methodology}: Agile
\end{itemize}

\section{Project Management Plan}
\subsection{Tasks and Timetable}
```

```
\begin{itemize}
    \item \textbf{Requirements Analysis}: 2 weeks (Mahek Thakkar)
    \item \textbf{AI Model Development}: 30 days (Chandana Galgali)
    \item \textbf{VR Prototype Development}: 30 days (Prachi Gandhi)
    \item \textbf{Integration and Testing}: 2 weeks (Harsh Singwi)
    \item \textbf{Final Deployment}: 2 weeks (Team Collaboration)
\end{itemize}

\subsection{Gantt Chart}
\begin{figure}[h!]
    \centering
    \includegraphics[width=\textwidth]{gantt_chart.png}
    \caption{Gantt Chart for Project Tasks}
\end{figure}

\end{document}
```

Software Project Management Plan (SPMP)

Project Team: Chandana Galgali, Prachi Gandhi, Mahek Thakkar, Harsh Singwi

January 22, 2025

1 Introduction

1.1 Project Overview

This project aims to develop a text-to-VR environment generator enabling users to create immersive VR experiences through textual descriptions. The system integrates Generative AI and VR technologies to achieve this. Expected delivery date: 3.5 months from project initiation.

1.2 Project Deliverables

- SPMP Document
- Source Code (AI and VR modules)
- Prototype Application
- User Guide
- Testing Reports

2 Project Organization

2.1 Software Process Model

The Incremental Model will be used, involving:

- Weekly progress reviews.
- Milestones: Requirements, Prototype, Testing, and Deployment.
- Versions: Alpha, Beta, and Final.

2.2 Roles and Responsibilities

- **Mahek Thakkar**: Requirements Analysis
- **Chandana Galgali**: AI Model Development
- **Prachi Gandhi**: VR Prototype Development
- **Harsh Singwi**: Integration and Testing

2.3 Tools and Techniques

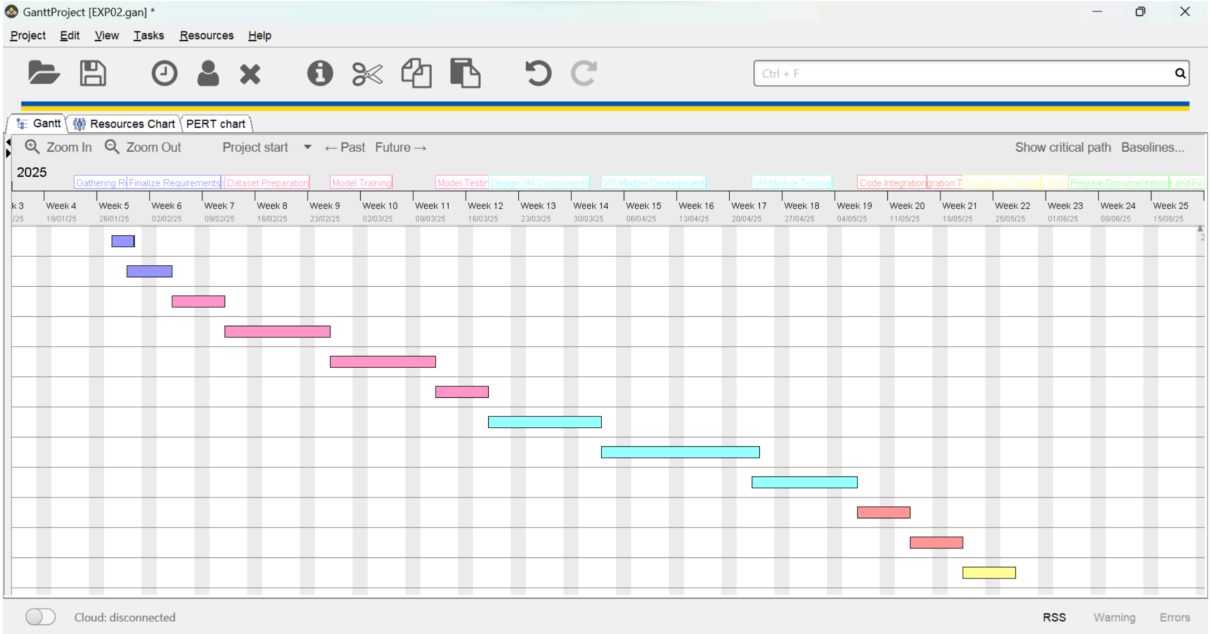
- **Languages**: Python (AI), Unity/C (VR)
- **Tools**: TensorFlow, Unity, GitHub, GanttProject
- **Methodology**: Agile

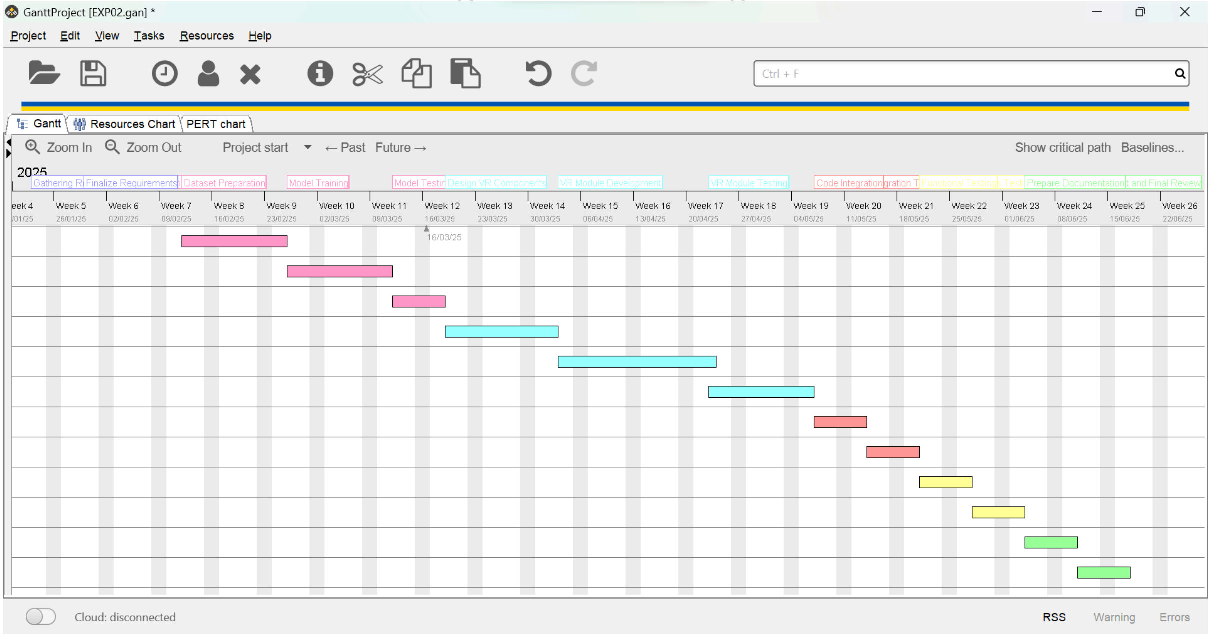
3 Project Management Plan

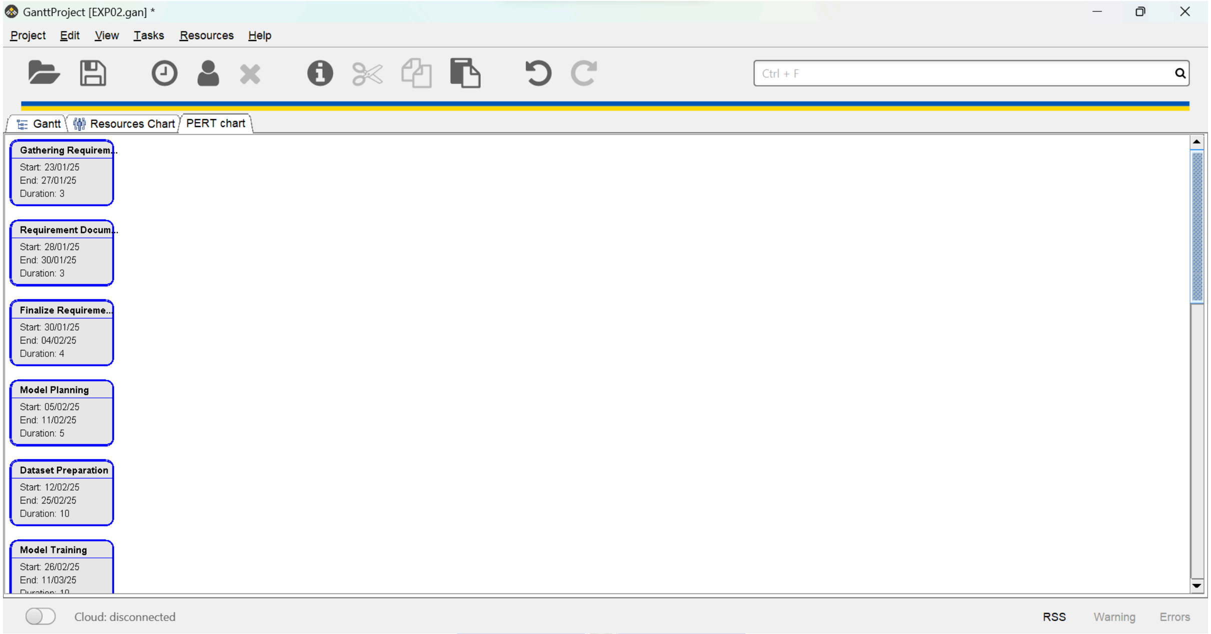
3.1 Tasks and Timetable

- **Requirements Analysis**: 2 weeks (Mahek Thakkar)
- **AI Model Development**: 30 days (Chandana Galgali)
- **VR Prototype Development**: 30 days (Prachi Gandhi)
- **Integration and Testing**: 2 weeks (Harsh Singwi)
- **Final Deployment**: 2 weeks (Team Collaboration)

3.2 Gantt Chart:







Gathering Requirem...
Start: 23/01/25
End: 27/01/25
Duration: 3

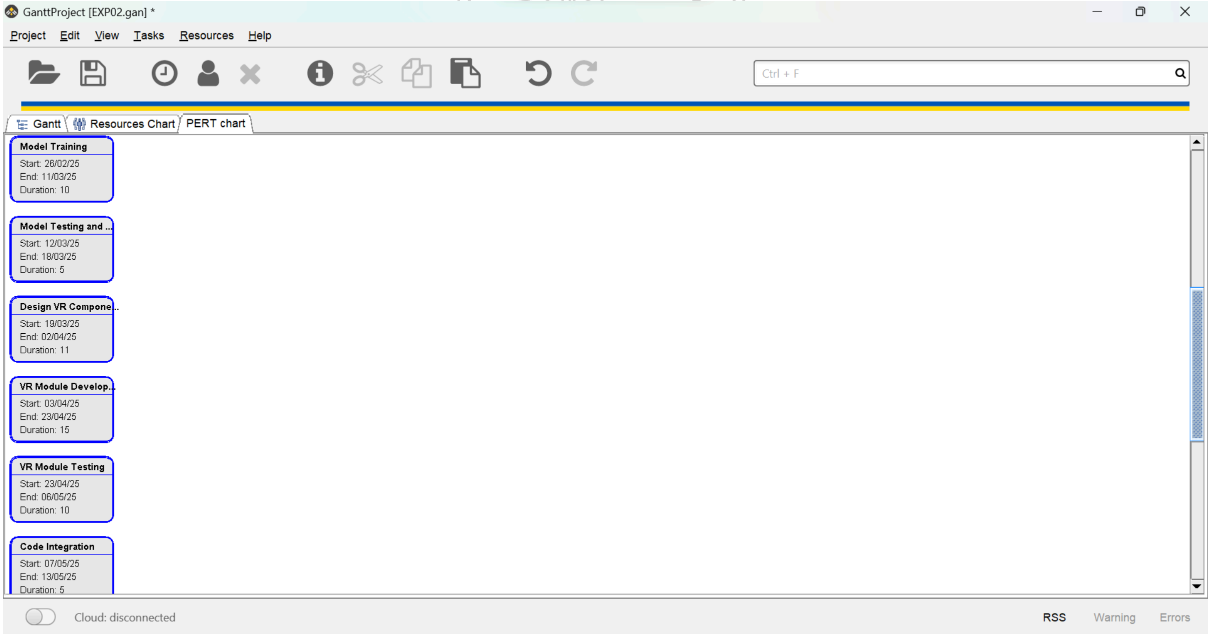
Requirement Docum...
Start: 28/01/25
End: 30/01/25
Duration: 3

Finalize Requireme...
Start: 30/01/25
End: 04/02/25
Duration: 4

Model Planning
Start: 05/02/25
End: 11/02/25
Duration: 5

Dataset Preparation
Start: 12/02/25
End: 25/02/25
Duration: 10

Model Training
Start: 26/02/25
End: 11/03/25
Duration: 10



Model Training
Start: 26/02/25
End: 11/03/25
Duration: 10

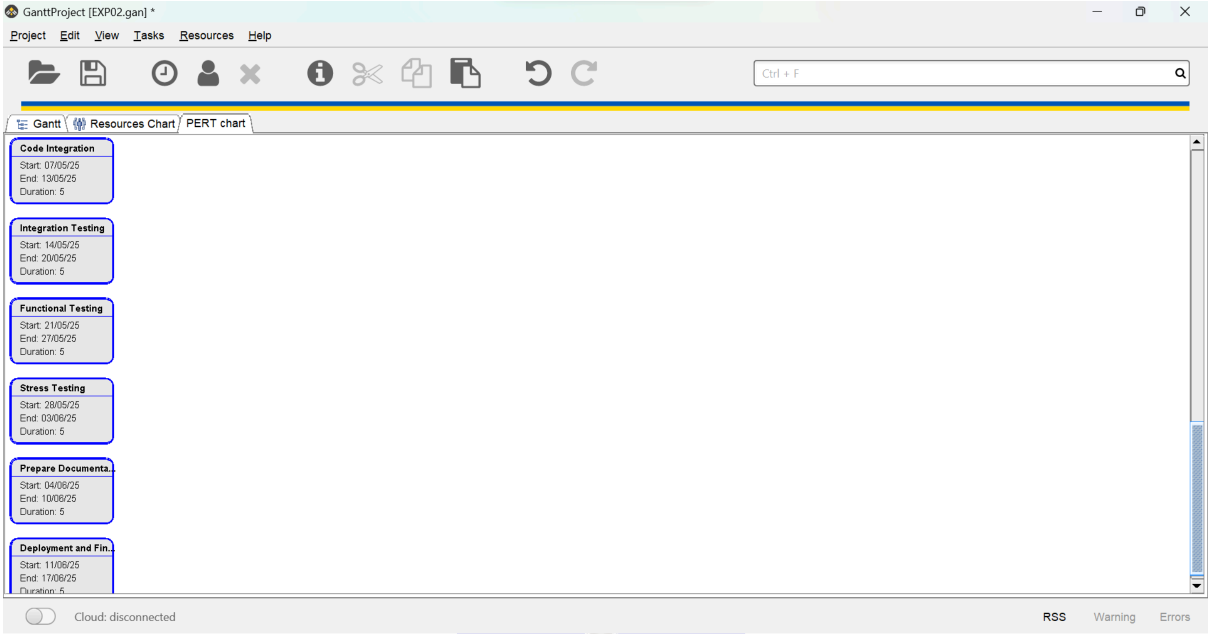
Model Testing and ...
Start: 12/03/25
End: 18/03/25
Duration: 5

Design VR Compone...
Start: 19/03/25
End: 02/04/25
Duration: 11

VR Module Develop...
Start: 03/04/25
End: 23/04/25
Duration: 15

VR Module Testing
Start: 23/04/25
End: 06/05/25
Duration: 10

Code Integration
Start: 07/05/25
End: 13/05/25
Duration: 5



Code Integration
Start: 07/05/25
End: 13/05/25
Duration: 5

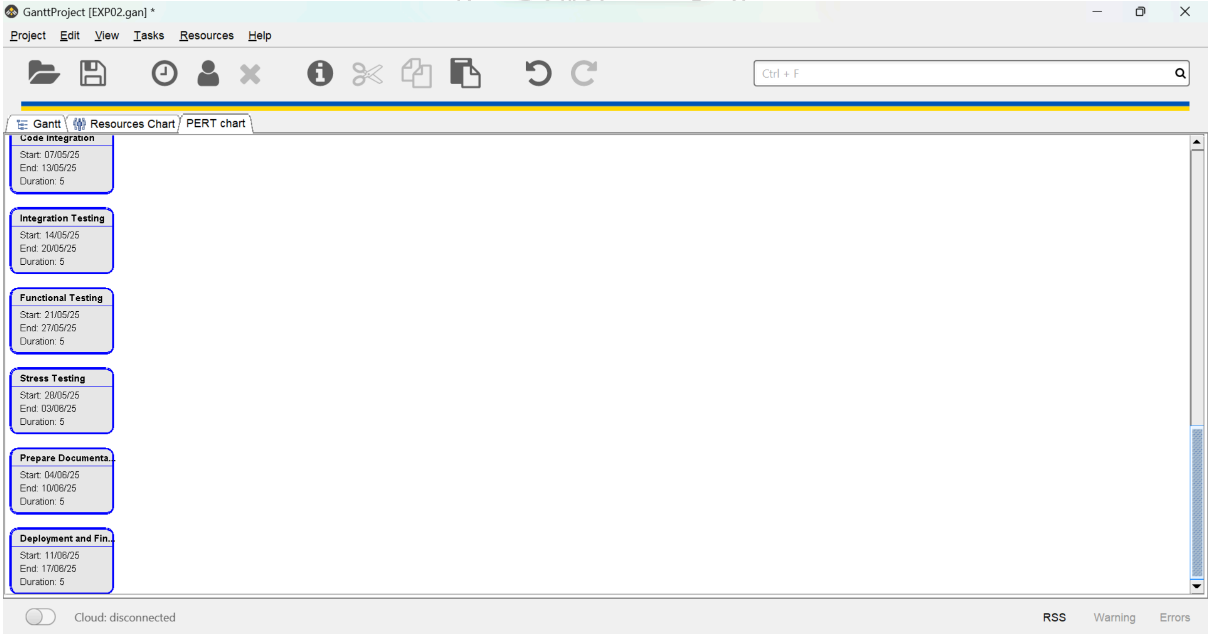
Integration Testing
Start: 14/05/25
End: 20/05/25
Duration: 5

Functional Testing
Start: 21/05/25
End: 27/05/25
Duration: 5

Stress Testing
Start: 28/05/25
End: 03/06/25
Duration: 5

Prepare Documenta...
Start: 04/06/25
End: 10/06/25
Duration: 5

Deployment and Fin...
Start: 11/06/25
End: 17/06/25
Duration: 5



Code Integration
Start: 07/05/25
End: 13/05/25
Duration: 5

Integration Testing
Start: 14/05/25
End: 20/05/25
Duration: 5

Functional Testing
Start: 21/05/25
End: 27/05/25
Duration: 5

Stress Testing
Start: 28/05/25
End: 03/06/25
Duration: 5

Prepare Documenta...
Start: 04/06/25
End: 10/06/25
Duration: 5

Deployment and Fin...
Start: 11/06/25
End: 17/06/25
Duration: 5

Questions:

1. Can a project be undertaken without a plan? What are the possible consequences?
Ans: No, a project cannot effectively be undertaken without a plan. The possible consequences of proceeding without a plan include:

1) Unclear Objectives: Without a clear plan, the project's goals and deliverables remain undefined, leading to confusion and misalignment among team members.

2) Missed Deadlines: A lack of planning results in poor time management, causing delays in task completion and failure to meet project deadlines.

3) Inefficient Resource Utilization: Without a plan, resources (time, tools, and manpower) may not be allocated efficiently, leading to waste or shortages during critical phases.

4) Increased Risks: Unanticipated issues may arise due to the absence of risk analysis and contingency planning, potentially derailing the project entirely.

5) Budget Overruns: An absence of financial planning can lead to uncontrolled expenditures, significantly increasing project costs.

6) Lack of Coordination: Without a structured approach, team roles and communication pathways remain undefined, leading to miscommunication and ineffective collaboration.

7) Quality Issues: A lack of focus on deliverables, standards, and milestones can compromise the overall quality of the final product or service.

In summary, planning is essential to ensure clarity, efficiency, and control throughout the project lifecycle, reducing the likelihood of failure.

Outcomes: CO2 – Describe software planning and management.

Conclusion:
The Software Project Management Plan (SPMP) is a vital document that outlines the project's objectives, deliverables, roles, and timelines, ensuring clarity and effective execution. By utilizing structured tools like Gantt charts and LaTeX, the plan facilitates efficient resource allocation, risk management, and team coordination. This experiment emphasizes the importance of thorough planning for delivering quality software projects on time.

Grade: AA / AB / BB / BC / CC / CD / DD

Signature of faculty in-charge with date

References:

Books:

1) Roger S. Pressman, Software Engineering: A practitioner's Approach, 7th Edition, McGraw Hill, 2010.
2) Ian Somerville, Software Engineering, 9th edition, Addison Wesley, 2011.
3) John M. Nicholas, “Project Management for Business and Technology”, 2nd edition, Pearson Education

---