Terna Engineering College

Computer Engineering Department

Program: Sem VI

Course: Software Engineering Lab

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No.08

A.1 Aim:

Study of project management tool and to schedule project plan using any tool (Gantt chart, Critical Path Method, and Earn Value Analysis).

A.2 Prerequisite:

Knowledge about preparing Project Schedule and the use of various Monitoring Tools.

A.3 Outcome:

After successful completion of this experiment students will be able to schedule the system using various project scheduling tools.

A.4 Theory:

1. Project Scheduling

Project Scheduling in a project refers to the roadmap of all activities to be done in the specified order and within the time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and then arrange them keeping various factors in mind. They look for tasks that lie in a critical path in the schedule, which are necessary to complete in a specific manner (because of task interdependence) and strictly within the time allocated. Arrangement of tasks that lies out of the critical path is less likely to impact the overall schedule of the project.

For scheduling a project, it is necessary to -

1. Break down the project tasks into smaller, manageable form
2. Find out various tasks and correlate them
3. Estimate the time frame required for each task
4. Divide time into work-units
5. Assign the adequate number of work-units for each task
6. Calculate the total time required for the project from start to finish

Project Execution & Monitoring

- In this phase, the tasks described in project plans are executed according to their schedules.
- Execution needs monitoring to check whether everything is going according to the plan. Monitoring is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.

These measures include -

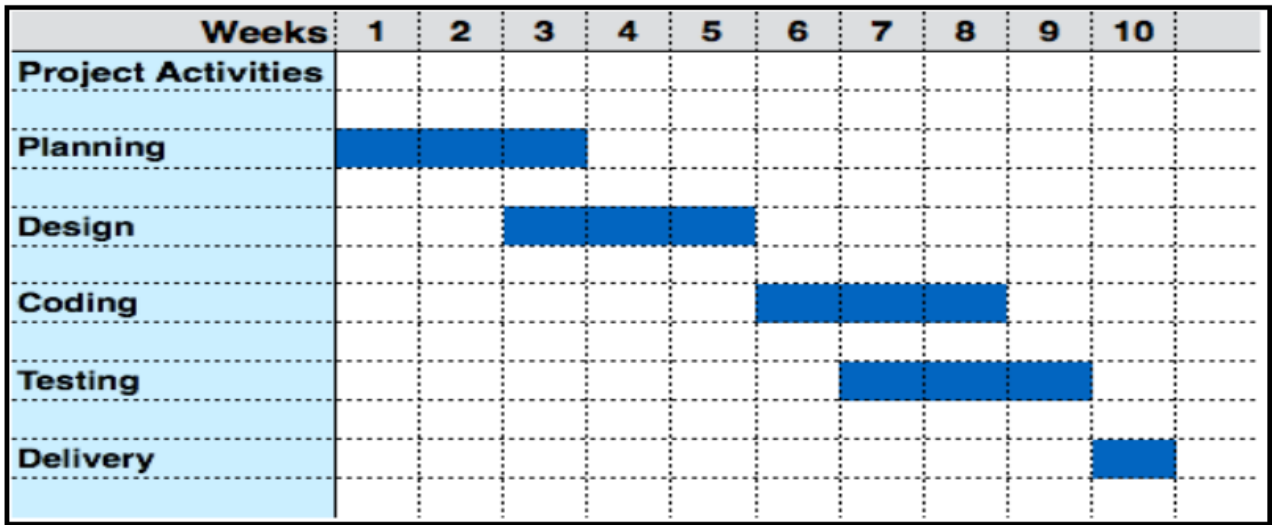
1. **Activity Monitoring** - All activities scheduled within some task can be monitored on a day-to-day basis. When all activities in a task are completed, it is considered complete.
2. **Status Reports** - The reports contain the status of activities and tasks completed within a given time frame, generally a week. Status can be marked as finished, pending or work-in-progress etc.
3. **Milestones Checklist** - Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC. This milestone checklist is prepared once every few weeks and reports the status of milestones.

Project Management Tools

- The risk and uncertainty rise multifold concerning the size of the project, even when the project is developed according to set methodologies.
- There are tools available, which aid in effective project management. A few are described –

Gantt Chart:

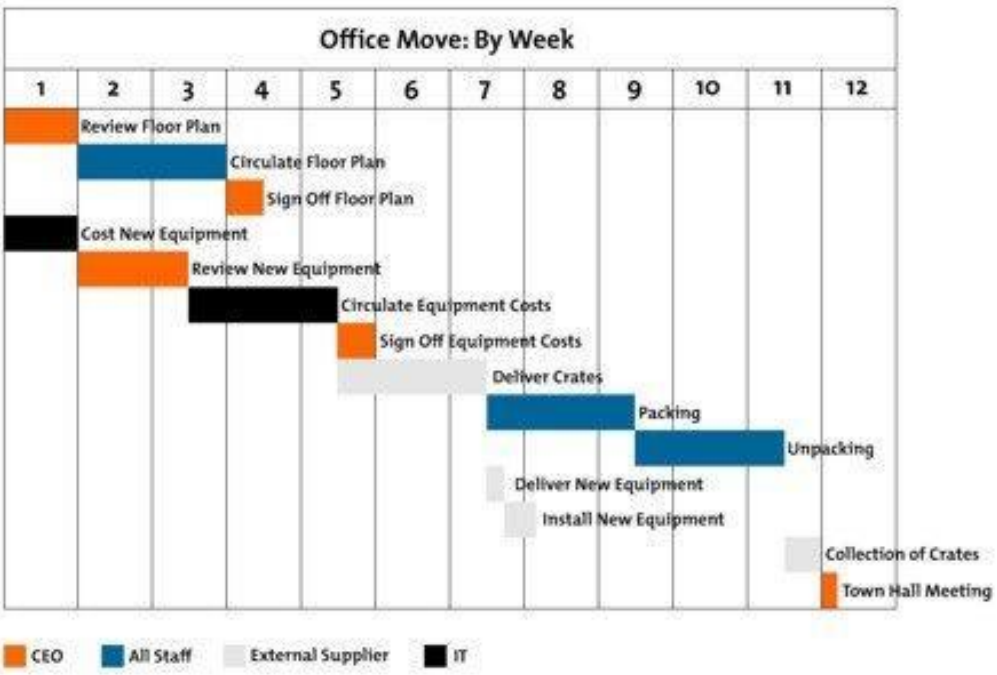
Gantt charts were devised by Henry Gantt (1917). It represents a project schedule concerning periods. It is a horizontal bar chart with bars representing activities and time scheduled for the project activities.



Creating a Gantt Chart

You can see an example in figure 1, below:

Figure 1 – A Gantt chart



To create one for your project, follow these steps, using our example as a guide.

Step 1: Identify Essential Tasks

Gantt charts don't give useful information unless they include all of the activities needed for a project or project phase to be completed.

So, to start, list all of these activities. Use a work breakdown structure if you need to establish what the tasks are. Then, for each task, note its earliest start date and its estimated duration.

Example

Your organization has won a tender to create a new "Software as a Service" product, and you're in charge of the project.

You decide to use a Gantt chart to organize all of the necessary tasks, and to calculate the likely overall timescale for delivery.

You start by listing all of the activities that have to take place, and you estimate how long each task should take to complete. Your list looks as follows:

Task	Length
A. High-level analysis	1 week
B. Selection of server hosting	1 day
C. Configuration of server	2 weeks
D. Detailed analysis of core modules	2 weeks

E. Detailed analysis of supporting modules	2 weeks
F. Development of core modules	3 weeks
G. Development of supporting modules	3 weeks
H. Quality assurance of core modules	1 week
I. Quality assurance of supporting modules	1 week
J. Initial client internal training	1 day
K. Development and QA of accounting reporting	1 week
L. Development and QA of management reporting	1 week
M. Development of management information system	1 week
N. Client internal user training	1 week

Step 2: Identify Task Relationships

The chart shows the relationship between the tasks in a project. Some tasks will need to be completed before you can start the next one, and others can't end until the preceding ones have ended. For example, if you're creating a brochure, you need to finish the design before you can send it to print.

These dependent activities are called "sequential" or "linear" tasks.

Other tasks will be "parallel"– i.e. they can be done at the same time as other tasks.

You don't have to do these in sequence, but you may sometimes need other tasks to be finished first. So, for example, the design of your brochure could begin before the text has been edited (although you won't be able to finalize the design until the text is perfect.)

Identify which of your project's tasks are parallel, and which are sequential. Where tasks are dependent on others, note down the relationship between them. This will give you a deeper understanding of how to organize your project, and it will help when you start scheduling activities on the chart.

Note:

In the Gantt charts, there are three main relationships between sequential tasks:

Finish to Start (FS) – FS tasks can't start before a previous (and related) task is finished. However, they can start later.

Start to Start (SS) – SS tasks can't start until a preceding task starts. However, they can start later.

Finish to Finish (FF) – FF tasks can't end before a preceding task ends. However, they can end later.

A fourth type, Start to Finish (SF), is very rare.

Tip 1:

Tasks can be sequential and parallel at the same time – for example, two tasks (B and D) may be dependent on another one (A), and maybe completed at the same time. Task B is sequential in that it follows on from A, and it is parallel, concerning D.

Tip 2:

To minimize delivery times, you'll need to do as much work in parallel as you sensibly can. You also need to keep the scope of the project as small as possible.

Example

Task	Length	Type*	Dependent on...
A. High-level analysis	1 week	S	
B. Selection of server hosting	1 day	S	A
C. Configuration of server	2 weeks	S	B
D. Detailed analysis of core modules	2 weeks	S, P to B, C	A
E. Detailed analysis of supporting modules	2 weeks	S, P to F	D
F. Development of core modules	3 weeks	S, P to E	D

G. Development of supporting modules	3 weeks	S, P to H, J	E
H. Quality assurance of core modules	1 week	S, P to G	F
I. Quality assurance of supporting modules	1 week	S	G
J. Initial client internal training	1 day	S, P to G	C, H
K. Development and QA of accounting reporting	1 week	S	E
L. Development and QA of management reporting	1 week	S	E
M. Development of Management Information System	1 week	S	L
N. Client internal user training	1 week	S	I, J, K, M

* P: Parallel, S: Sequential

Step 3: Input Activities Into Software Or a Template

You can draw your charts by hand or use specialist software, such as Gatto, Matchware, or Microsoft Project. Some of these tools are cloud-based, meaning that you and your team can access the document simultaneously, from any location. (This helps a lot when you're discussing, optimizing, and reporting on a project.)

Several Gantt templates have been created for Microsoft Excel, and you can also find free templates with a quick search online.

Step 4: Chart Progress

As your project moves along, it will evolve. For example, in our scenario, if quality assurance of core modules revealed a problem, then you may need to delay training and halt the development of the management information system until the issue is resolved.

Update your chart to reflect changes as soon as they occur. This will help you to keep your plans, your team, and your sponsors up to date.

Critical Path Analysis

This tool is useful in recognizing interdependent tasks in the project. It also helps to find out the shortest path or critical path to complete the project successfully. Like a PERT diagram, each event is allotted a specific time frame. This tool shows the dependency of the event assuming an event can proceed to the next only if the previous one is completed.

The events are arranged according to their earliest possible start time. The path between the start and end node is a critical path that cannot be further reduced and all events require to be executed in the same order.

The software requirements are a description of the features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from the client's point of view.

This program is a tool to assist in project management in general and project scheduling in particular. It uses a critical path method (CPM) and applies it to software development. The technique is driven by information already developed in earlier project planning activities like effort, product function, tasks decomposition, process model and task set. CPM provide a quantitative tool that allows the software planner to (1) determine the critical path - the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models, and (3) calculate boundary times that define a time window for a particular task.

Earned Value Analysis (EVA)

Earned Value Management (EVM) is a technique that is used to track the progress and status of a project and forecast its likely future performance.

This is a brief tutorial that acquaints the reader with the basics of EVM and explains how to utilize it for better project management.

PART B

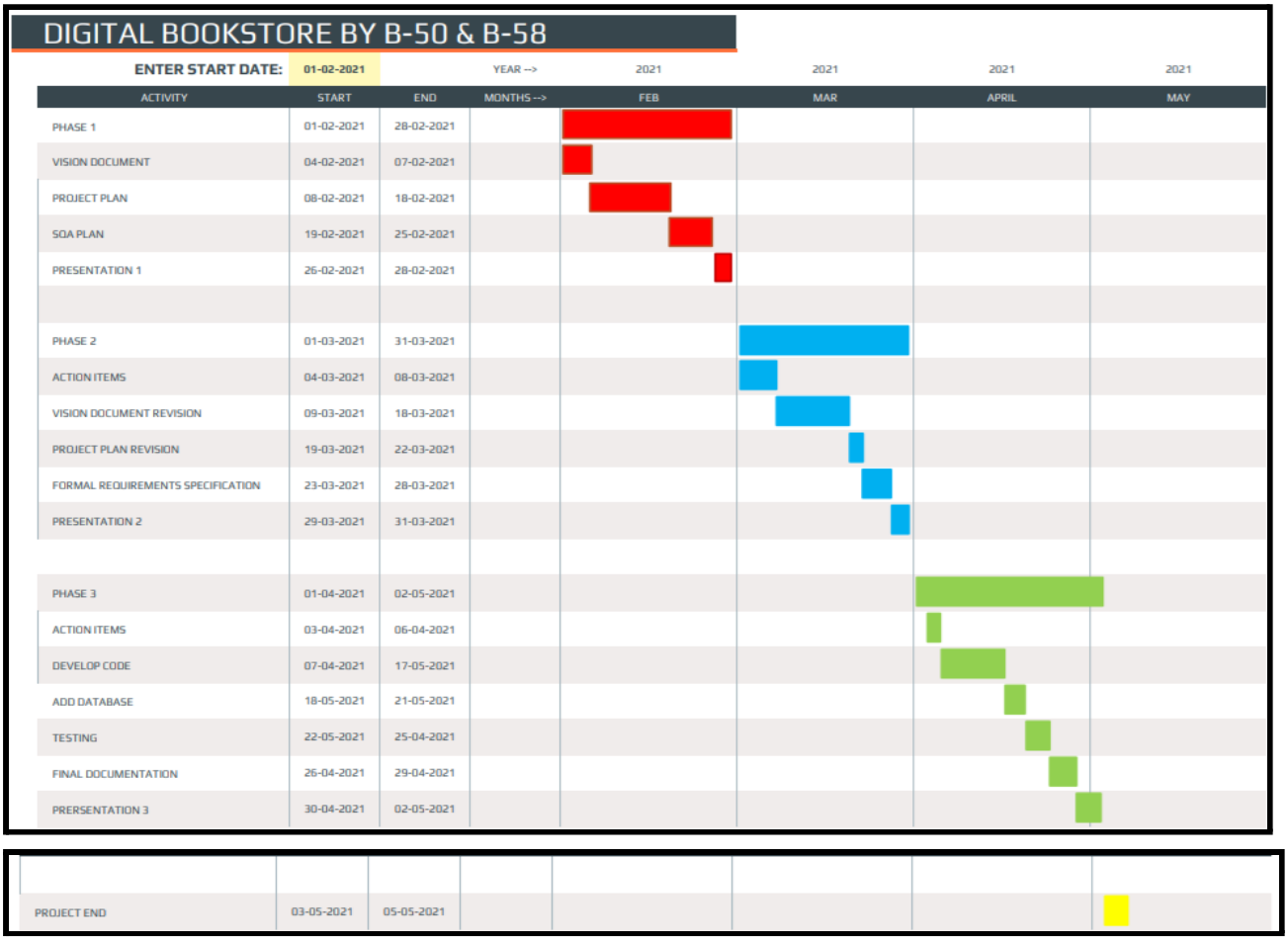
(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	**Name:** AMEY THAKUR
Class: Comps TE B	**Batch:** B3
Date of Experiment: 28/03/2021	**Date of Submission:** 28/03/2021
Grade:	

B.1 Draw Time Line chart or AON for your selected mini project and estimate the total duration of the project.

(Assume time from 1 Feb to 5 May 2021)



B.2 Conclusion:

(Students must write the conclusion)

From this experiment, we learn the project management tool and to schedule a project plan using any tool (Gantt chart, Critical Path Method, and Earn Value Analysis).

B.3 Question of Curiosity

1. Explain the principles of Project scheduling

Ans:

A software project schedule can be defined as an activity that distributes the estimated effort across the planned project duration by allocating the effort to specific software engineering tasks. Simply one can say that a project schedule is a tool that communicates

1. What works has to be performed.
2. Who will perform the work?
3. Time duration within which that work needs to be completed.

There are seven principles of software project scheduling :

1. **COMPARTMENTALIZATION:** A given software project is compartmentalized into several manageable activities. The project is divided into several small tasks.
2. **INTERDEPENDENCY:** Interdependent tasks are accomplished first. Certain tasks occur in sequence whereas other tasks occur in parallel. Therefore tasks that occur in the sequence has to be performed in sequential order since the output of one task will be the input of the next task. Other tasks can occur independently.
3. **TIME ALLOCATION:** Every task has to be assigned a specific period i.e a start date and a completion date based on whether the work will be performed on a full time or part-time basis.
4. **EFFORT VALIDATION:** Every project is assigned to a software team. The project manager has to make sure that the effort allocated should not be more than the number of people available to do the work.
5. **DEFINED RESPONSIBILITIES:** Each of the scheduled tasks is assigned to a specific member of the software team.
6. **DEFINED OUTCOMES:** Each task has a defined outcome. Work product is the outcome of a software project.
7. **DEFINED MILESTONES:** Every task is associated with a milestone. A milestone is an action or event marking a significant change in the development process.

2. For the given example draw Activity on Node diagram and estimate the critical path of the same.

Activity	Predecessor	Duration (in weeks)
A	–	2
B	A	3
C	–	2
D	A, C	3
E	D	4
F	A, D	3
G	B, E	2
H	G	3

Ans:

AMEY B 50 Amey

Activity	Predecessor	Duration (in weeks)
A	–	2
B	A	3
C	–	2
D	A, C	3
E	D	4
F	A, D	3
G	B, E	2
H	G	3