**Chapter 8**

**Project Planning II**

**8.1 Introduction**

Project planning is one of the most important jobs of a software project manager. The project plan, which is created at the start of a project, is used to communicate how the work will be done to the project team and customers, and to help assess progress on the project.

Project planning is an iterative process that starts when you create an initial project plan during the project startup phase. Plan changes are inevitable. As more information about the system and the project team becomes available during the project, you should regularly revise the plan to reflect requirements, schedule, and risk changes.

Project planning takes place at three stages in a project life cycle:

1. ***At the proposal stage,*** you need a plan at this stage to help you decide if you have the resources to complete the work.
2. ***During the project startup phase,*** when you have to plan who will work on the project, how the project will be broken down into increments, how resources will be allocated across your company, etc.
3. ***Periodically throughout the project,*** when you modify your plan in light of experience gained and information from monitoring the progress of the work.

**8.2 Software Pricing**

*In principle, the price of a software product to a customer is simply the cost of development plus profit for the developer*. In practice, however, the relationship between the project cost and the price quoted to the customer is not usually so simple. When calculating a price, you should take ***broader organizational*,** ***economic*, *political*, and** ***business considerations*** into account.

**8.3 Plan-Driven Development**

Plan-driven or plan-based development is an approach to software engineering where the development process is planned in detail.

A project plan is created that record:

***the work to be done, who will do it, the development schedule, and the work products*.**

Managers use the plan to support project decision making and as a way of measuring progress. Although the specific details of project plans vary depending on the type of project and organization, plans normally include the following sections:

1. **Introduction** This briefly describes the objectives of the project and sets out the constraints (e.g., budget, time, etc.) that affect the management of the project.
2. **Project organization** this describes the way in which the development team is organized, the people involved, and their roles in the team.
3. **Risk analysis** this describes possible project risks, the likelihood of these risks arising, and the risk reduction strategies that are proposed.
4. **Hardware and software resource requirements** this specifies the hardware and support software required to carry out the development.
5. **Work breakdown** this sets out the breakdown of the project into activities and identifies the \***milestones** and \***deliverables** associated with each activity..
6. **Project schedule** this shows the dependencies between activities, the estimated time required to reach each milestone, and the allocation of people to activities.
7. **Monitoring and reporting mechanisms** this defines the management reports that should be produced, when these should be produced, and the project monitoring mechanisms to be used.

**\*Milestones:** indicate the completion of key project tasks, they identify when one or multiple groups of activities have been completed thus implying that a notable point has been reached in the project are key stages in the project where progress can be assessed.

**\*Deliverables**: are work products that are delivered to the customer.

**8.4 Project Scheduling**

Project scheduling is the process of deciding:

- How the work in a project will be organized as separate tasks.

- When and how these tasks will be executed.

- Estimate the calendar time needed to complete each task, the effort required,

- Who will work on the tasks that have been identified.

- Estimate the resources needed to complete each task, such as the disk space required on a server, the time required on specialized hardware, such as a simulator.

Scheduling in plan-driven projects (Figure 8.1) involves breaking down the total work involved in a project into separate tasks and estimating the time required to complete each task.

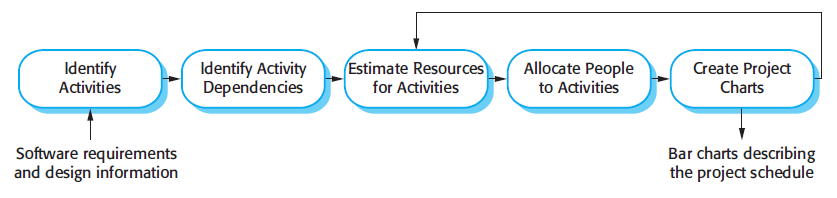


Figure (8.1) The project scheduling process

* Tasks should normally last at least a week.
* The maximum amount of time for any task should be around 8 to 10 weeks.
* If it takes longer than this, the task should be subdivided for project planning and scheduling.
* Some of these tasks are carried out in parallel, with different people working on different components of the system.

**8.4.1 Schedule Representation**

Project schedules may simply be represented in a table or spreadsheet showing the ***tasks***, ***expected duration, and task dependencies***, as shown in table (8.1).

- Milestones may be associated with a single task or with groups of related activities. For example, in table (8.1), milestone M1 is associated with task T1 and milestone M3 is associated with a pair of tasks, T2 and T4.

- ***A deliverable*** is usually delivered at the end of some major project phase such as specification, design, etc.

- Deliverables are usually milestones but milestones need not be deliverables.

***“Milestones may be internal project results that are used by the project manager to check project progress but which are not delivered to the customer”.***

From table (8.1), you can see that task T3 is dependent on task T1. Task T1 must, therefore, be completed before T3 starts. For example, T1 might be the preparation of a component design and T3, the implementation of that design. Before implementation starts, the design should be complete.

|  |  |  |
| --- | --- | --- |
| Task | Duration (days) | Dependencies |
| T1 | 10 |  |
| T2 | 15 |  |
| T3 | 15 | T1 (M1) |
| T4 | 10 |  |
| T5 | 10 | T2, T4 (M3) |
| T6 | 5 | T1, T2 (M4) |
| T7 | 20 | T1 (M1) |
| T8 | 25 | T4 (M2) |
| T9 | 15 | T3, T6 (M5) |
| T10 | 15 | T7, T8 (M6) |
| T11 | 10 | T9 (M7) |
| T12 | 10 | T10, T11 (M8) |

Table (8.1) Tasks, durations, and dependencies

**8.4.2 Bar Charts and Activity Networks**

In table (8.1), this style of representation makes it difficult to see the relationships and dependencies between the different activities. For this reason, alternative graphical representations of project schedules have been developed that are often easier to read and understand. There are two types of representation that are commonly used **Bar charts (Gantt charts)**, and **Activity networks**:

1. **Bar charts (Gantt charts)**, which are calendar-based, show *who is responsible for each activity*, *the expected elapsed time*, and *when the activity is scheduled to begin and end*.

* A bar chart shows a project calendar and the start and finish dates of tasks.
* Reading from left to right.
* The milestones (M1, M2, etc.) are also shown on the bar chart.
* A bar chart allocates suitable staff to the suitable activity.
* Staff may be working on more than one task at the same time. If a task is delayed, this can obviously affect later tasks that are dependent on it. They cannot start until the delayed task is completed
* The process of creating a Gantt chart was manual. Nowadays, automated tools are available for creating Gantt charts.
* Notice that tasks that are independent are carried out in parallel (e.g., tasks T1, T2, and T4 all start at the beginning of the project - figure 8.2 -). To illustrate how bar charts are used, figure (8.2) takes the information in table (8.1) and presents the project schedule in a graphical format.

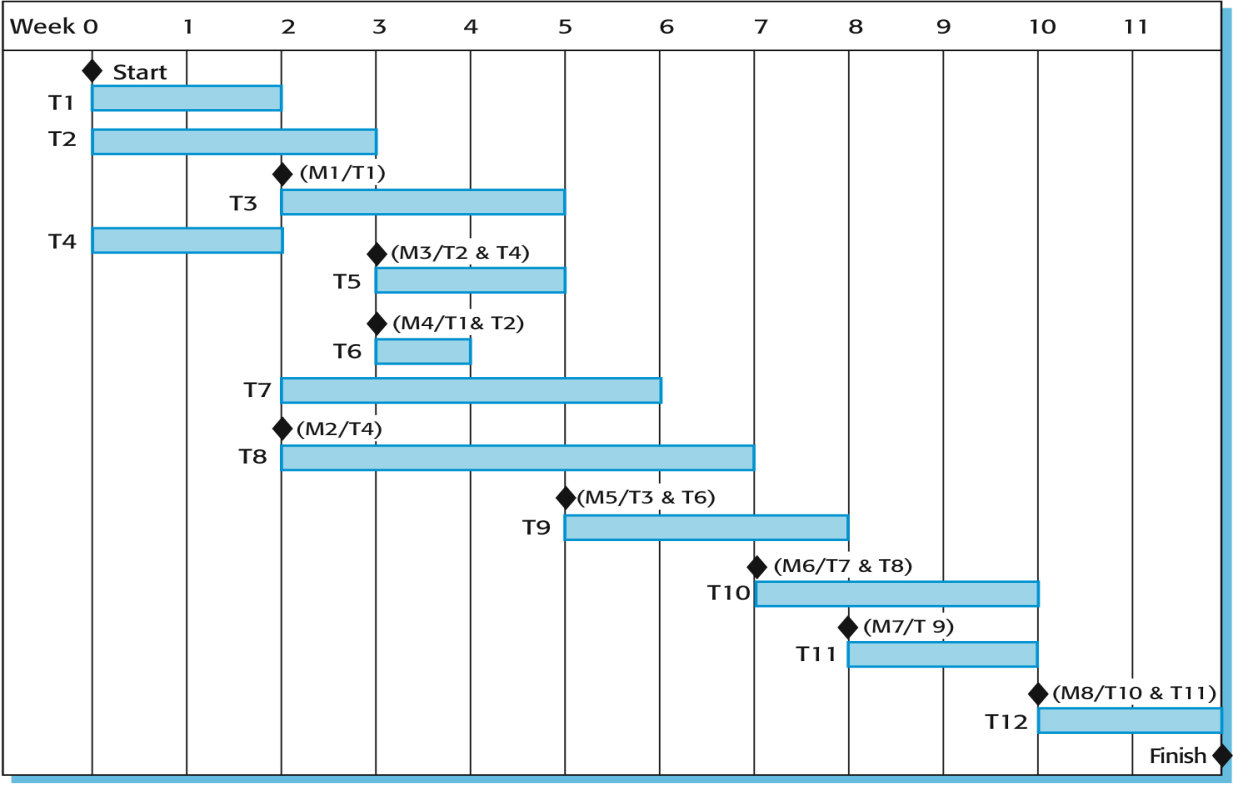


Figure (8.2) Bar chart

1. **Activity networks**, Ganttcharts are poor chart management when projects are lengthy and complex. Activity networksare more effective time management tool for *large* and *complex projects.* Activity networks *are graphic representation of the task flow for a project show the critical path and task dependencies between the different activities making up a project.*

- ***Critical path***: is sequence of activities that take the longest time to complete, those activities must be completed on schedule if the project as a whole is to be completed on schedule.

* + Any delay to an activity in the critical path will cause delays to the overall project.
  + Delays to activities not on the critical path need not necessarily cause overall delays.
  + Duration times of activities in the critical path will determine the overall project time, because there is no slack time for these activities.
* Break down project in to sequence of activities to be completed.
* You should read the network from left to right and from top to bottom.
* The activities are depicted as nodes (rectangles) labeled with estimated times.
* While the arrows denote precedence relations between activities
* Activities are linked if there is a dependency between them.
* Milestones are shown with rounded corners.

**Example -1-** The figure -8.3- below set out a number of tasks, their durations and their dependencies. Draw:

-An activity network and determine the critical path,

-Staff allocation against time chart for showing the project schedule.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Dependencies** | **Engineers** |
| T1 | 8 |  | Jane |
| T2 | 15 |  | Anne |
| T3 | 15 | T1 (M1) | Jane |
| T4 | 10 |  | Fred |
| T5 | 10 | T2, T4 (M2) | Mary |
| T6 | 5 | T1, T2 (M3) | Anne |
| T7 | 20 | T1 (M1) | Jim |
| T8 | 25 | T4 (M5) | Fred |
| T9 | 15 | T3, T6 (M4) | Jane |
| T10 | 15 | T5, T7 (M7) | Anne |
| T11 | 7 | T9 (M6) | Fred |
| T12 | 10 | T11 (M8) | Fred |

Figure (8.3) Tasks durations, dependencies

****

Figure (8.4) Activity network

The critical path is 55 days