UNIT-6
Planning a Software Project

Management Spectrum

- **The management spectrum describes the management of a software project or how to make a project successful.**
- It focuses on the four P's; people, product, process and project.
- Here, the manager of the project has to control all these P's to have a smooth flow in the project progress and to reach the goal.
- 4 P's are:

1. People
2. Product
3. Process
4. Project

Management Spectrum

1. People:

- **The "people factor" is so important that the Software Engineering Institute has developed a People Capability Maturity Model(People-CMM),** in recognition of the fact that "every organization needs to continually improve its ability to attract, develop, motivate, organize, and retain the workforce needed to accomplish its strategic business objectives.
- **The people capability maturity model defines the following key practice areas for software people:** staffing, communication and coordination, work environment, performance management, training, compensation, competency analysis and development, career development, workgroup development, team/culture development, and others.

Management Spectrum

- Organizations that achieve high levels of People-CMM maturity have a higher likelihood of implementing effective software project management practices.
- There are different types of people:

I. Stakeholders
II. Team Leaders
III. Software Team
IV. Agile Team

2. Product:

- Before a project can be planned, product objectives and scope should be established, alternative solutions should be considered, and technical and management constraints should be identified.

Management Spectrum

- **Software Scope:** The first software project management activity is the determination of software scope. Scope is defined by answering the following questions: context, information objectives, function and performance.

3. Process:

- **A software process provides the framework from which a comprehensive plan for software development can be established**.
- A small number of framework activities are applicable to all software projects, regardless of their size or complexity.

Management Spectrum

- A number of different task sets—tasks, milestones, work products, and quality assurance points—enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team.
- **Finally, umbrella activities**—such as software quality assurance, software configuration management, and measurement—overlay the process model.
- Umbrella activities are independent of any one framework activity and occur throughout the process.

Management Spectrum

- Team must decide which process model is most appropriate for:

(1) the customers who have requested the product and the people who will do the work,

(2) the characteristics of the product itself, and

(3) the project environment in which the software team works. When a process model has been selected, the team the defines a preliminary project plan based on the set of process framework activities.

4. Project: Five points to overcome the problems in the software projects.

i) Start on the right foot.

- This is accomplished by working hard to understand the problem that is to be solved and then setting realistic objects and expectations for everyone who will be involved in the project.

Management Spectrum

ii) Maintain momentum.

- Many projects get off to a good start and then slowly disintegrate.
- The project manager must provide reasons to keep turnover of personnel to an absolute minimum, the team should emphasize quality in every task it performs, and senior management should do everything possible to stay out of the team's way.

iii) Track progress.

- For a software project, progress is tracked as work products are produced and approved as part of a quality assurance activity.

iv) Make smart decisions.

- In essence, the decisions of the project manager and the software team should be to "keep it simple."
- Whenever possible decide to use commercial off the shelf software or existing software compo-available, decide to identify and then avoid obvious risks.

Management Spectrum

v) Conduct a post-mortem analysis.

- Establish a consistent mechanism for extracting lessons learned for each project.
- Evaluate the planned and actual schedules, collect and analyze software project metrics, get feedback from team members and customers, and record findings in written form.

W^{5}HH PRINCIPLE

- Boehm suggests an approach that addresses project objectives, milestones and schedules, responsibilities, management and technical approaches, and required resources.
- It is known as W5HH Principle, after a series of questions that lead to a definition of key project characteristics and the resultant project plan:
- **Why is the system being developed?** All stakeholders should assess the validity of business reasons for the software work.
- **What will be done?** The task set required for the project is defined.
- **When will it be done?** The team establishes a project schedule by identifying when project tasks are to be conducted and when milestones are to be reached.

W^{5}HH PRINCIPLE

- **Who is responsible for a function?** Earlier in this chapter, we noted that the role and responsibility of each member of the software team must be defined. The answer to this question helps accomplish this.
- **Where they are organizationally located?** Not all roles and responsibilities reside within the software team itself. The customer, users, and other stakeholders also have responsibilities.
- **How will the job be done technically and managerially?** Once product scope is established, a management and technical strategy for the project must be defined.
- **How much of each resource is needed?** The answer to this question is derived by developing estimates based on answers to earlier questions.

W^5HH PRINCIPLE

- Boehm's W5HH principle is applicable regardless of the size or complexity of a software project. The questions noted provide an excellent planning outline for the project manager and the software team.

SOFTWARE SCOPE AND FEASIBILITY

- **Software scope:** describes the functions and features that are to be delivered to end users; the data that are input and output; the "content" that is presented to users as a consequence of using the software; and the performance, constraints, interfaces, and reliability that bound the system.

Scope is defined using one of **two** techniques:

- 1. A narrative description of software scope is developed after communication with all stakeholders.
- 2. A set of use cases is developed by end users

SOFTWARE SCOPE AND FEASIBILITY

- **Software feasibility has four solid dimensions:**

1. **Technology**—Is a project technically feasible? Is it within the state of the art? Can defects be reduced to a level matching the application's needs?
2. **Finance**—Is it financially feasible? Can development be completed at a cost the software organization, its client, or the market can afford?
3. **Time**—Will the project's time-to-market beat the competition?
4. **Resources**—Does the organization have the resources needed to succeed?

Effort Estimation

- For a software development project, overall effort and schedule estimates are essential prerequisites for planning the project. These estimates are needed before development is initiated, as they establish the cost and schedule goals of the project.
- Use of these estimates is in bidding for software projects, where cost and schedule estimates must be given to a potential client for the development contract.
- Effort and schedule estimates are also required for determining the staffing level for a project during different phases, for the detailed plan, and for project monitoring.
- **The accuracy with which effort can be estimated clearly depends on the level of information available about the project. The more detailed the information, the more accurate the estimation can be.**

Effort Estimation

- There are TWO types of approaches:

1) Top-down estimation:

- **The primary factor that controls the effort is the size of the project. That is, the larger the project, the greater is the effort requirement.**
- It should be noted that to use the top-down approach for estimation, even if we have a suitable function, we need to have an estimate of the project size. In other words, we have replaced the problem of effort estimation by size estimation.

2) Bottom-up estimation:

- **In this approach, the project is first divided into tasks and then estimates for the different tasks of the project are obtained.**

EFFORT ESTIMATION

- The overall estimate of the project is derived from the estimates of its parts. This type of approach is also called **activity-based estimation.**
- **The procedure for estimation can be summarized as the following sequence of steps:**

  1. **Identify modules** in the system and classify them as **simple, medium, or complex.**
  2. **Determine** the average coding effort for **simple/medium/complex modules.**
  3. **Get the total coding effort** using the coding effort of different types of modules and the counts for them.
  4. **Using the effort distribution** for similar projects, estimate the effort for other tasks and the total effort.
  5. **Refine the estimates** based on project-specific factors.

Schedule and Staffing

- **With the effort estimate (in person-months), it may be tempting to pick any project duration and then fix a suitable team size to ensure that the total effort matches the estimate**.
- The schedule and people are not fully interchangeable in the software project.
- For instance if the effort required is 36 persons month then it is hard to say that within 6 months and with the help of 6 people one can complete the project or with the help of 9 people the same project can be completed within 4 months.
- In **project the scheduling** can be done using **two simple activities. Determining** overall schedule by using different important milestones. **Developing** detailed schedule for different tasks.

Schedule and Staffing

- **Overall Scheduling**
- The overall schedule for the project can be obtained using the effort as the function. This function is designed from the data available from the past similar projects.
- The IBM federal systems division has found that duration M is estimated in terms of month as

$$\mathbf{M = 4.1E^{0.36}}$$

- In COCOMO, the equation for schedule for an unrefined type of software is,

$$\mathbf{M= 2.5\ E^{0.38}}$$

- From these equations it is clear that, the schedule is not entirely dependent upon the effort estimate.

Schedule and Staffing

- Square root check is another method which can be used to obtain the estimate for the schedule. For the **medium sized projects the proposed schedule can be around square root of the total effort in terms of person-months.**
- That is if the effort estimate is 36 person months then the schedule is about 6 to 7 months. From this estimate, we can determine the schedule for the major milestones in the project.
- **The requirement of people for system testing and integration is less. But for coding and unit testing large number of people are required.**
-

Risk Management

- Software risk encompasses the probability of occurrence for uncertain events and their potential for loss within an organization.
- Risk always involves two characteristics: **uncertainty**—the risk may or may not happen; that is, there are no 100 percent probable risks—and **loss**—if the risk becomes a reality, unwanted consequences or losses will occur
- **Risk Identification:**
- **Risk identification is a systematic attempt to specify threats to the project plan (estimates, schedule, resource loading, etc.).**
- By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

Risk Management

- One method for identifying risks is **to create a risk item checklist.** The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:
- **Product size**—risks associated with the overall size of the software to be built or modified.
- **Business impact**—risks associated with constraints imposed by management or the marketplace.
- **Stakeholder characteristics**—risks associated with the sophistication of the stakeholders and the developer's ability to communicate with stakeholders in a timely manner.
- **Process definition**—risks associated with the degree to which the software process has been defined and is followed by the development organization.

Risk Management

- **Development environment**—risks associated with the availability and quality of the tools to be used to build the product.
- **Technology to be built**—risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- **Staff size and experience**—risks associated with the overall technical and project experience of the software engineers who will do the work.

Risk Management

- **Risk Assessment:**
- **The goal of risk assessment is to prioritize the risks so that attention and resources can be focused on the more risky items.**
- Risk identification is the first step in risk assessment, which identifies all the different risks for a particular project.
- **The next tasks are risk analysis and prioritization.**
- **In risk analysis,** the probability of occurrence of a risk has to be estimated, along with the loss that will occur if the risk does appear.

Risk Management

- **Risk Assessment:**
- Once the probabilities of risks materializing and losses due to materialization of different risks have been analyzed, **they can be prioritized.**
- One approach for **prioritization is through the concept of risk exposure (RE), which is sometimes called risk impact.**
- RE is defined by the relationship

$$\mathbf{RE = Prob(UO) * Loss(UO)}$$

- where Prob(UO) is the probability of the risk materializing (i.e., undesirable outcome) and Loss(UO)

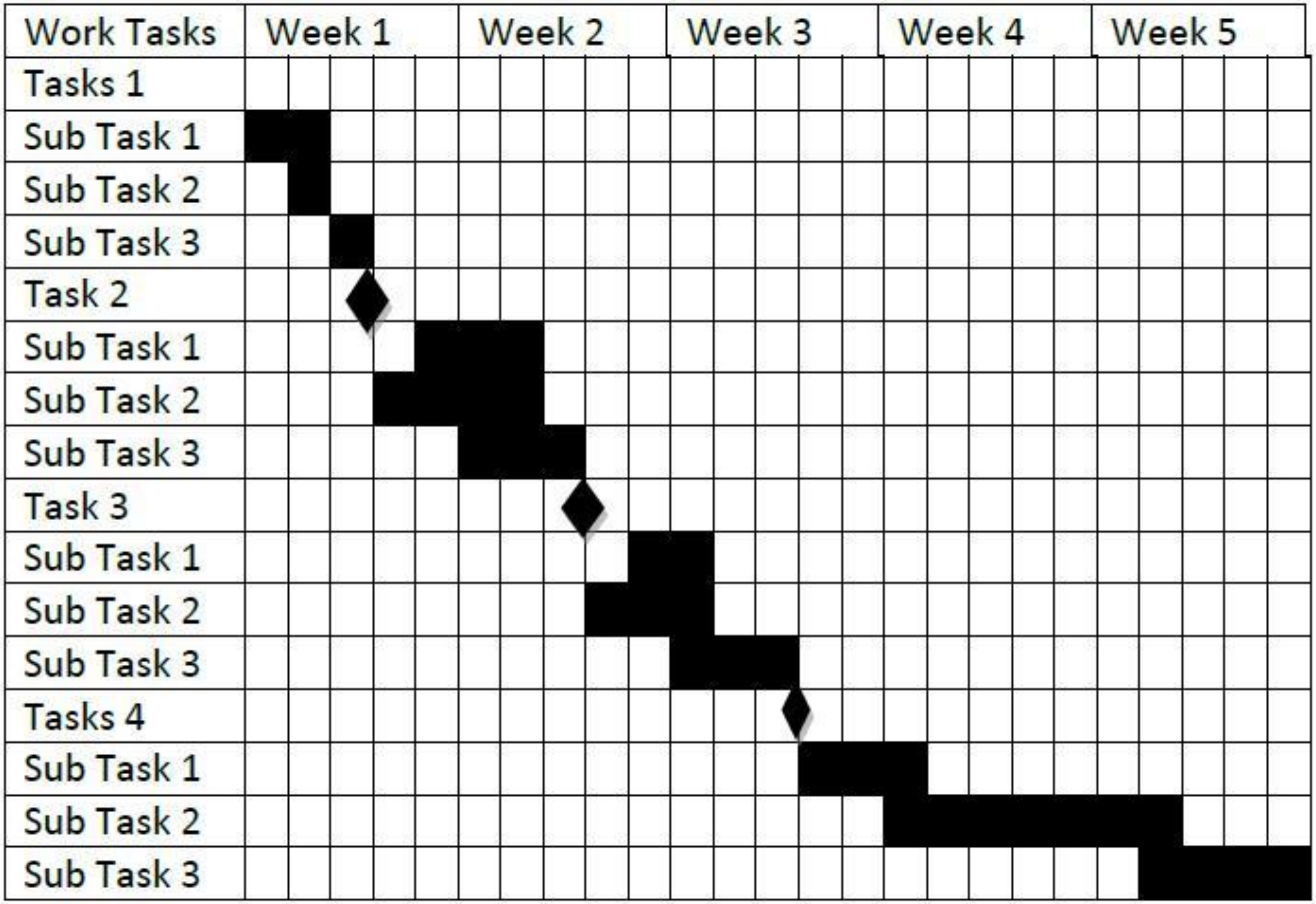
Risk Management

- **Risk Control:**
- For most risks, the strategy is to perform the actions that will either reduce the probability of the risk materializing or reduce the loss due to the risk materializing. **These are called risk mitigation[improvement] steps.**
- **Risk monitoring** is the activity of monitoring the status of various risks and their control activities. One simple approach for risk monitoring is to analyze the risks afresh at each major milestone, and change the plans as needed.

Detailed Scheduling

- Detailed scheduling is divided into two parts.
  - Time Line Chart.
  - Tracking the schedule.
- **1.Time Line Chart.**
- While creating a software project schedule the planner begins with a set of tasks. If automated tools are used then the work breakdown as a task network or task outline.
- In addition, tasks may be assigned to specific individuals.
- As a consequence a timeline chart, also called a Gantt chart, is generated.

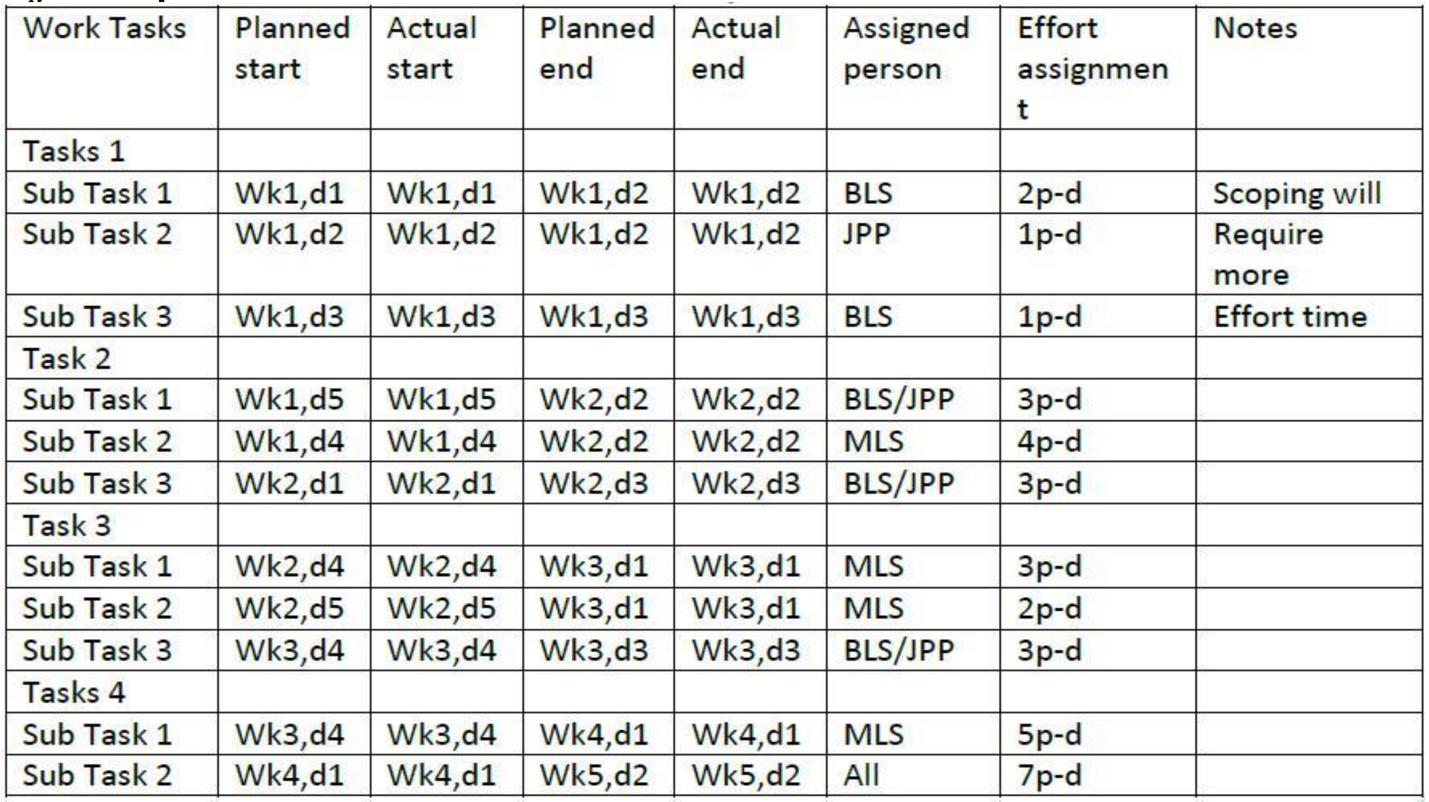
Detailed Scheduling



Work Tasks	Week 1					Week 2					Week 3					Week 4					Week 5				
Tasks 1																									
Sub Task 1	■	■																							
Sub Task 2		■																							
Sub Task 3			■																						
Task 2			◆																						
Sub Task 1					■	■	■																		
Sub Task 2				■	■	■	■																		
Sub Task 3						■	■	■																	
Task 3								◆																	
Sub Task 1										■	■														
Sub Task 2									■	■	■														
Sub Task 3												■	■												
Tasks 4													◆												
Sub Task 1														■	■	■									
Sub Task 2																■	■	■	■	■	■	■			
Sub Task 3																						■	■	■	■

DETAILED SCHEDULING

- **2.Tracking the schedule**
- The project schedule provides a road map for a software project manager. If it has been properly developed, the project schedule defines the tasks and milestones that must be tracked and controlled as the project proceeds.



Work Tasks	Planned start	Actual start	Planned end	Actual end	Assigned person	Effort assignmen t	Notes
Tasks 1							
Sub Task 1	Wk1,d1	Wk1,d1	Wk1,d2	Wk1,d2	BLS	2p-d	Scoping will
Sub Task 2	Wk1,d2	Wk1,d2	Wk1,d2	Wk1,d2	JPP	1p-d	Require more
Sub Task 3	Wk1,d3	Wk1,d3	Wk1,d3	Wk1,d3	BLS	1p-d	Effort time
Task 2							
Sub Task 1	Wk1,d5	Wk1,d5	Wk2,d2	Wk2,d2	BLS/JPP	3p-d	
Sub Task 2	Wk1,d4	Wk1,d4	Wk2,d2	Wk2,d2	MLS	4p-d	
Sub Task 3	Wk2,d1	Wk2,d1	Wk2,d3	Wk2,d3	BLS/JPP	3p-d	
Task 3							
Sub Task 1	Wk2,d4	Wk2,d4	Wk3,d1	Wk3,d1	MLS	3p-d	
Sub Task 2	Wk2,d5	Wk2,d5	Wk3,d1	Wk3,d1	MLS	2p-d	
Sub Task 3	Wk3,d4	Wk3,d4	Wk3,d3	Wk3,d3	BLS/JPP	3p-d	
Tasks 4							
Sub Task 1	Wk3,d4	Wk3,d4	Wk4,d1	Wk4,d1	MLS	5p-d	
Sub Task 2	Wk4,d1	Wk4,d1	Wk5,d2	Wk5,d2	All	7p-d	

THANK YOU