

1 UNIT

Introduction and Software Project Planning

Part-1 (5C - 18C)

- Fundamentals of Software Project Management
- Need Identification
- Vision and Scope Document
- Project Management Cycle
- SPM Objectives
- Management Spectrum
- SPM Framework

A. Concept Outline : Part-1 5C
B. Long and Medium Answer Type Questions 5C

Part-2 (19C - 25C)

- Software Project Planning
- Planning Objectives
- Types of Project Plan
- Structure of Software Project
- Management Plan

A. Concept Outline : Part-2 19C
B. Long and Medium Answer Type Questions 19C

Part-3 (28C - 40C)

- Software Project Estimation
- Estimation Methods
- Estimation Models
- Decision Process

A. Concept Outline : Part-3 28C
B. Long and Medium Answer Type Questions 29C

Software Project Management

5 (CS/IT-7) C

PART-1

Fundamentals of Software Project Management, Need Identification, Vision and Scope Document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM Framework.

CONCEPT OUTLINE : PART-1

- Software project management is perhaps the most important factor in the outcome of a project.
- Project management cycle comprises of four phases :
 - a. Initiation
 - b. Planning
 - c. Execution
 - d. Closure
- Need and identification is the initial phase of project life cycle.
- The vision/scope document represents the ideas and decisions developed during envisioning phase.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.1. What is project ? Explain the characteristics of project.

Answer

1. In general, a project is understood to be a group of activities that are carried out to achieve a certain outcome.
2. Each project, thus, will have well defined objectives, which would measure success or failure of the project.
3. In IT industry, software projects involve the implementation of applications using new technology or improving upon the functionalities of existing applications using technology.
4. In essence, software projects use technology to develop or improve existing applications for meeting business requirements.
5. In software projects, the users will have a general idea about how the application would look like and its applicability.
6. The software projects sometimes deal with the abstract nature of the end result and get modified as time elapses.
7. Software projects should have certain activities and need to have minimum attributes for its successful completion.

Characteristics of project :

The characteristics which distinguish projects can be summarized as follows :

1. Non-routine tasks are involved.
2. Planning is required.
3. Specific objectives are to be met or a specified product is to be created.
4. The project has a pre-determined time span.
5. Work is carried out for someone other than yourself.
6. Work involves several specialism's.
7. Work is carried out in several phases.
8. The resources that are available for use on the project are constrained.
9. The project is large or complex.

Que 1.2. What is project management ?**Answer**

1. Project management is the art of maximizing the probability that a project delivers its goals on time, within budget and at the required quality.
2. The art of planning for the future has always been a human trait.
3. In essence, a project can be captured on paper with a few simple elements : a start date, an end date, the tasks that have to be carried out and when they should be finished, and some idea of the resources (people, machines etc.) that will be needed during the course of the project.
4. Project management is the application of knowledge, skills, tools, and techniques to project activities to meet project requirements.
5. Project management is accomplished through the use of the processes such as : initiating, planning, executing, controlling, and closing.
6. It is important to note that many of the processes within project management are iterative in nature.

Que 1.3. Discuss various types of software projects.**Answer**

The following are main application areas of software :

1. **System software :** System software is a collection of programs that are written to provide services to other programs. The examples of system software are compiler, editor, file management utilities etc.
2. **Business software :** Business information processing is the largest single software application area. The examples of business application area are payroll, inventory management, marketing, purchase etc.
3. **Embedded software :** Intelligent consumer products are becoming very popular in industrial market. Embedded software resides in read

only memory and used to control products and systems for the consumer and markets. Examples of embedded software are washing machine, microwave oven, air conditioner, sleeping timer in television remote, fuel control keypad etc.

4. **Engineering and scientific software :** Engineering and scientific software have been categorized by "number crunching" algorithms. There are number of software which are used for scientific use. Examples of such types of software are Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), simulation etc.

5. **Personal computer software :** Personal computer software may be said as small business applications which are used by single user. The examples of this type of software are word processor, spreadsheet, multimedia, entertainment, database management, small business financial application etc.

6. Real time software :

- a. Software that monitors/analyzes/controls real world events as they occur is known as real time software.
- b. Elements of real time software includes :
 - i. data gathering components that collects and formats information from an external environment,
 - ii. environment, an analysis component that transform information as required by the application,
 - iii. a control/output component that respond to the external environment, and
 - iv. a monitoring component that coordinates all other components so that real time response (typically all ranging from 1 millisecond to 1 second) can be maintained.

7. **Artificial intelligence software :** Artificial intelligence software makes use of non-numerical algorithms to solve the complex problem that cannot solve by straight forward analysis. The examples of A.I. software are expert system, pattern recognition (image or voice), artificial neural network, theorem proving, game playing etc.

Que 1.4. How software projects are different from other types of project ?**Answer**

Many of the techniques of general project management are applicable to software project management, but Fred Brooks pointed out that the products of software projects have certain characteristics which make them different.

1. Invisibility :

- a. When a physical artefact such as a bridge or road is being constructed the progress being made can actually be seen. With software, progress is not immediately visible.

- b. One way of perceiving software project management is the process of making visible that which is invisible.
- 2. Complexity :** Per dollar, pound or euro spent software products contain more complexity than other engineered artefacts.
- 3. Conformity :**
- The traditional engineer is usually working with physical systems and physical materials like cement and steel.
 - These physical systems can have some complexity, but are governed by physical laws that are consistent.
 - Software developers have to conform to the requirements of human clients. It is not just that individuals can be inconsistent.
 - Organizations, because of lapses in collective memory, in internal communication or ineffective decision-making, can exhibit remarkable organizational stupidity that developers have to cater for.
- 4. Flexibility :**
- The ease with which software can be changed is usually seen as one of its strengths.
 - However, this means that where the software system interfaces with a physical or organizational system, it is expected that, where necessary, the software will change to accommodate the other components rather than vice versa.
 - This means the software systems are likely to be subjected to a high degree of change.

Que 1.5. What are the activities or knowledge area involved in management/project management ?

Answer

The open university software project management module suggested that management involves the following activities/knowledge area :

- Planning :** Deciding what is to be done.
- Organizing :** Making arrangements.
- Staffing :** Selecting the right people for the job, etc.
- Directing :** Giving instructions.
- Monitoring :** Checking on progress.
- Controlling :** Taking action to remedy hold-ups.
- Innovating :** Coming up with new solutions.
- Representing :** Liaising with users, etc.

Que 1.6. Write a short note on software project management.

Answer

- The essence of project management is to make sure that all the agreed objectives are accepted, executed, and shared among all the stakeholders.
- An effective project management will ensure that all the stakeholders are identified at the initial stage itself and then "buy in" the objectives.
- The first step in project management would be to start planning.
- A detailed planning will help identify minute details and plan for these details at that granular level.
- The success of project management is entrusted to project manager (PM).
- Some of the elements that contribute to the success of project management are as follows :
 - clearly stated and understood objectives,
 - role and responsibilities of each team member,
 - communication protocol,
 - escalation process,
 - clearly defined project organization structure, which will show process of information flow,
 - periodic status report review, and
 - changed management process.
- Predictability is an important aspect of project management; this ensures that customers as well as team members can predict the quality and the schedule of the deliverables.

Que 1.7. What are the various activities covered by software project management ? Also, explain the life cycle of a software project.

Answer

Various activities covered by software project management are :

- The feasibility study :**
 - This is an investigation into whether a prospective project is worth starting.
 - Information is gathered about the requirements of the proposed application.
 - The probable developmental and operational costs, along with the value of the benefits of the new system, are estimated.
 - With a large system, the feasibility study could be treated as a project in its own right and have its own planning sub-phase.

2. Planning :

- a. If the feasibility study produces results which indicate that the prospective project appears viable, then planning of the project can take place.
- b. However, for a large project, we would not do all our detailed planning right at the beginning.
- c. We would formulate an outline plan for the whole project and a detailed one for the first stage.

3. Project execution : The project can now be executed. The execution of a project often contains design and implementation sub-phases.

Stages involved in the life cycle of a software project are :

1. **Requirements analysis :** The user's requirements from the system are analyzed during the implementation of the project.
2. **Specification :** Detailed documentation of what the proposed system is to do.
3. **Design :** A design has to be drawn up which meets the specification. This design will be in two stages. One will be the external or user design concerned with the external appearance of the application. The other produces the physical design.
4. **Coding :** This may refer to writing code in a procedural language such as C or Ada, or could refer to the use of an application-builder such as Microsoft Access.
5. **Verification and validation :** Whether software is developed specially for the current application or not, careful testing will be needed to check that the proposed system meets its requirements.
6. **Implementation/Installation :** Some system development practitioners refer to the whole of the project after design as 'implementation' (that is, the implementation of the design) while others insist that the term refers to the installation of the system after the software has been developed.
7. **Maintenance and support :** Once the system has been implemented there is a continuing need for the correction of any errors that may have crept into the system and for extensions and improvements to the system.

Que 1.8. Explain various problems with software projects in detail.

UPTU 2014-15, Marks 05

Answer

A survey of managers published by Thayer, Pyster and Wood identified the following commonly experienced problems :

1. Poor estimates and plans.
2. Lack of quality standards and measures.

3. Lack of guidance about making organizational decisions.
4. Lack of techniques to make progress visible.
5. Poor role definition who does what ?
6. Incorrect success criteria.

The above list looks at the project from the manager's point of view. What about the staff who make up the members of the project team? Below is a list of the problems identified by a number of students on a computing and information systems course who had just completed a year's industrial placement :

1. Inadequate specification of work.
2. Management ignorance of ICT.
3. Lack of knowledge of application area.
4. Lack of standards.
5. Lack of up-to-date documentation.
6. Preceding activities not completed on time - including late delivery of equipment.
7. Lack of communication leading to duplication of work.
8. Lack of communication between users and technicians.
9. Lack of commitment especially when a project is tied to one person who then moves.
10. Narrow scope of technical expertise.
11. Changing statutory requirements.
12. Changing software environment.
13. Deadline pressure.
14. Lack of quality control.
15. Remote management.
16. Lack of training.

Que 1.9. Explain need and identification of SPM. Also, write a vision and scope document according to IEEE standard in respect of SPM.

UPTU 2013-14, Marks 10

OR

What are the various factors which describe the scope of software ? Explain.

UPTU 2012-13, Marks 05

Answer

Need and identification of SPM :

1. Need and identification is the initial phase of the project life cycle.

2. It starts with the recognition of a need, problem, or opportunity and ends with the issuance of a Request For Proposal (RFP).
3. The customer identifies a need, a problem, or an opportunity for a better way of doing something and therefore sees some benefit to undertaking a project that will result in an improvement or advantage over the existing condition.
4. For example, suppose a company's management recognizes that the time the company takes to issue invoices and collect payments from its customers is too long.
5. Furthermore, the fact that the company payment records are not up to date has caused second invoices to be sent to customers who have already paid, thus upsetting some good customers.
6. Management recognizes several problems and opportunities for improvement, so it develops an RFP asking contractors to submit proposals for implementing an automated billing and collection system.
7. In a different scenario, the company's management might request a proposal from an in-house individual or project team rather than from an external contractor.

The outline of vision and scope documents is as follows :

1. Problem statement
 - a. Project background
 - b. Stakeholders
 - c. Users
 - d. Risks
 - e. Assumptions
2. Vision of the solution
 - a. Vision statement
 - b. List of features
 - c. Scope of phased release (optional)
 - d. Features that will not be developed

Que 1.10. Describe project management life cycle.

OR

Describe the terms project initiation, planning, execution and closure.

Answer

The project management life cycle comprises of four phases :

1. Initiation involves starting up the project, by documenting a business case, feasibility study, terms of reference, appointing the team and setting up a project office.

2. Planning involves setting out the roadmap for the project by creating the following plans : project plan, resource plan, financial plan, quality plan, acceptance plan and communications plan.
3. Execution involves building the deliverables and controlling the project delivery, scope, costs, quality, risks and issues.
4. Closure involves winding-down the project by releasing staff, handing over deliverables to the customer and completing a post implementation review.

A more detailed description of the MPMM project management methodology and life cycle is given below :

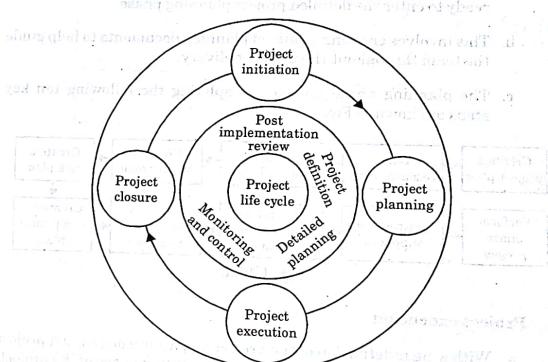


Fig. 1.10.1.

1. Project initiation :

- a. Project initiation is the first phase in the project life cycle and essentially involves starting up the project.
- b. Initiate a project by defining its purpose and scope, the justification for initiating it and the solution to be implemented.
- c. We will also need to recruit a suitably skilled project team, set up a project office and perform an end of phase review.
- d. The project initiation phase involves the following six key steps which are shown in Fig. 1.10.2.

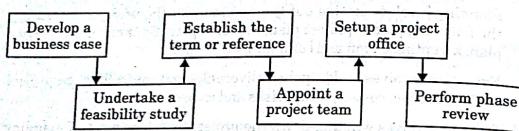


Fig. 1.10.2.

2. Project planning :

- After defining the project and appointing the project team, we are ready to enter the detailed project planning phase.
- This involves creating a suite of planning documents to help guide the team throughout the project delivery.
- The planning phase involves completing the following ten key steps as shown in Fig. 1.10.3.

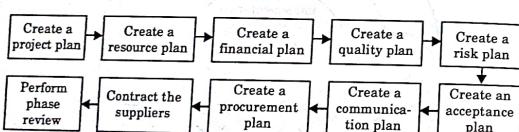


Fig. 1.10.3.

3. Project execution :

- With a clear definition of the project and a suite of detailed project plans, we are now ready to enter the execution phase of the project.
- This is the phase in which the deliverables are physically built and presented to the customer for acceptance.
- While each deliverable is being constructed, a suite of management processes are undertaken to monitor and control the deliverables being output by the project.
- These processes include managing time, cost, quality, change, risks, issues, suppliers, customers and communication.
- Once all the deliverables have been produced and the customer has accepted the final solution, the project is ready for closure.

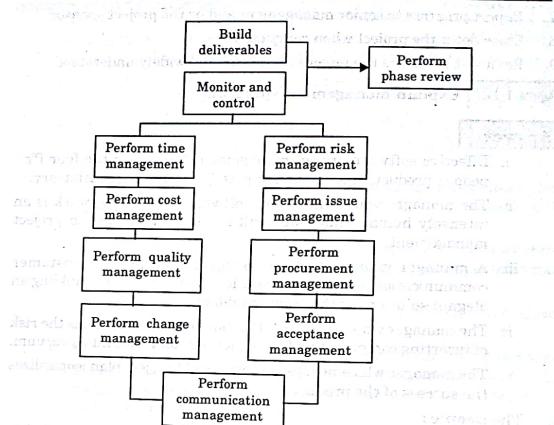


Fig. 1.10.4.

4. Project closure :

- Project closure involves releasing the final deliverables to the customer, handing over project documentation to the business, terminating supplier contracts, releasing project resources and communicating project closure to all stakeholders.
- The last remaining step is to undertake a post implementation review to identify the level of project success and note any lessons learned for future projects.

Ques 1.11. What are the objectives of SPM ?**Answer**

Basic objectives of SPM are as follows :

- Define the project.
- Reduce it to a set of manageable tasks.
- Obtain appropriate and necessary resources.
- Build a team or teams to perform the project work.
- Plan the work and allocate the resources to the tasks.
- Monitor and control the work.

7. Report progress to senior management and/or the project sponsor.
8. Close down the project when completed.
9. Review it to ensure the lessons are learnt and widely understood.

Que 1.12. Explain management spectrum.

Answer

- i. Effective software project management focuses on the four P's : people, product, process, and project. The order is not arbitrary.
- ii. The manager who forgets that software engineering work is an intensely human endeavour will never have success in project management.
- iii. A manager who fails to encourage comprehensive customer communication early in the evolution of a project risks, building an elegant solution for the wrong problem.
- iv. The manager who pays little attention to the process runs the risk of inserting competent technical methods and tools into a vacuum.
- v. The manager who embarks without a solid project plan jeopardizes the success of the product.

1. The people :

- a. The cultivation of motivated, highly skilled software people has been discussed since the 1960s.
- b. In fact, the people factor is so important that the Software Engineering Institute has developed a People Management Capability Maturity Model (PM-CMM), "to enhance the readiness of software organizations to undertake increasingly complex applications by helping to attract, grow, motivate, deploy, and retain the talent needed to improve their software development capability".
- c. The people management maturity model defines the following key practice areas for software people : recruiting, selection, performance management, training, compensation, career development, organization and work design, and team/culture development.
- d. Organizations that achieve high levels of maturity in the people management area have a higher likelihood of implementing effective software engineering practices.

2. The product :

- a. 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.
- b. Without this information, it is impossible to define reasonable (and accurate) estimates of the cost, an effective assessment of risk, a

realistic breakdown of project tasks, or a manageable project schedule that provides a meaningful indication of progress.

- c. The software developer and customer must meet to define product objectives and scope.

d. In many cases, this activity begins as part of the system engineering or business process engineering and continues as the first step in software requirements analysis.

- e. Objectives identify the overall goals for the product (from the customer's point of view) without considering how these goals will be achieved.

f. Scope identifies the primary data, functions and behaviours that characterize the product, and more important, attempts to bound these characteristics in a quantitative manner.

- g. Once the product objectives and scope are understood, alternative solutions are considered.

h. The alternatives enable managers and practitioners to select a best approach, given the constraints imposed by delivery deadlines, budgetary restrictions, personnel availability, technical interfaces, and myriad other factors.

3. The process :

- a. A software process provides the framework from which a comprehensive plan for software development can be established.
- b. A small number of framework activities are applicable to all software projects, regardless of their size or complexity.
- c. 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.
- d. Finally, umbrella activities - such as software quality assurance, software configuration management, and measurement - overlay the process model.
- e. Umbrella activities are independent of any one framework activity and occur throughout the process.

4. The project :

- a. We conduct planned and controlled software projects for one primary reason.
- b. It is the only known way to manage complexity. In 1998, industry data indicated that 26 % of software projects failed outright and 46 % experienced cost and schedule overruns.
- c. Although the success rate for software projects has improved somewhat, our project failure rate remains higher than it should be.

18 (CS/IT-7) C**Introduction and Software Project Planning**

- d. In order to avoid project failure, a software project manager and the software engineers who build the product must avoid a set of common warning signs, understand the critical success factors that lead to good project management, and develop a common sense approach for planning, monitoring and controlling the project.

Que 1.13. What is software project management framework ?

Answer

1. A project is a synchronized event where there is perfect harmony and understanding between the participants.
2. The participants are equipped with the essential skills of planning, cooperating, helping, and communicating.
3. However, the most important activity here is to arrange the movement of the participants.
4. The responsibility lies with the project manager to synchronize the activities of the project to result in a perfect presentation.
5. Although each, project manager has a unique style of functioning, there are some fundamental approaches that guide a project manager.
6. These approaches are traditional project management concepts and software engineering concepts.
7. To understand software projects and their dynamics, we must be aware of the environment in which a software project is executed.
8. This further requires an understanding of the larger framework of software project management.

Que 1.14. List the various responsibilities of software project manager.

Answer

1. Ownership of customer relationship and business.
2. Analysis of project health (productivity and profitability) and report to business manager.
3. Managing the onsite team.
4. Maintain the consolidated delivery and billing plan.
5. Identification and planning of new business with the customer.
6. Review of estimates and proposals.
7. Provide manpower requirements.
8. Maintain the project management plan.

Software Project Management**19 (CS/IT-7) C****PART-2**

Software Project Planning, Planning Objectives, Project Plan, Types of Project Plan, Structure of Software Project Management Plan.

CONCEPT OUTLINE : PART-2

- Project planning is an aspect of project management which comprises of various processes.
- Types of project plans:
 - a. Quality plan
 - b. Validation plan
 - c. Configuration management plan
 - d. Maintenance plan
 - e. Staff development plan

Questions-Answers**Long Answer Type and Medium Answer Type Questions**

Que 1.15. Write short note on software project planning.

Answer

1. Project planning is an aspect of project management, which comprises of various processes.
2. The aim of these processes is to ensure that various project tasks are well coordinated and they meet the various project objectives including timely completion of the project.
3. The project plan reflects the current status of all project activities and is used to monitor and control the project.
4. Project planning is an ongoing effort throughout the project life cycle.
5. The project planning tasks ensure that various elements of the project are coordinated and therefore guide the project execution.
6. Project planning helps in :
 - a. Facilitating communication.
 - b. Monitoring/measuring the project progress.
 - c. Provides overall documentation of assumptions/planning decisions.
7. The project planning phases can be broadly classified as follows :
 - a. Development of the project plan.

- b. Execution of the project plan.
- c. Change control and corrective actions.

Que 1.16. Write down the outline of stepwise planning activities.

Answer

The stepwise planning is discussed as follows :

0. Select project
1. Identify project scope and objectives :
 - a. Identify objectives and measures of effectiveness in meeting them.
 - b. Establish a project authority.
 - c. Identify stakeholders.
 - d. Modify objectives in the light of stakeholder analysis.
 - e. Establish methods of communications with all parties.
2. Identify project infrastructure :
 - a. Establish relationship between project and strategic planning.
 - b. Identify installation standards and procedures.
 - c. Identify project team organization.
3. Analyze project characteristics :
 - a. Distinguish the project as either objective or product driven.
 - b. Analyze other project characteristics.
 - c. Identify high-level project risks.
 - d. Take into account user requirements concerning implementation.
 - e. Select general life cycle approach.
 - f. Review overall resource estimates.
4. Identify project products and activities :
 - a. Identify and describe project products (including quality criteria).
 - b. Document generic product flows.
 - c. Recognize product instances.
 - d. Produce ideal activity network.
 - e. Modify idea to take into account need for stages and checkpoints.
5. Estimate effort for each activity :
 - a. Carry out bottom-up estimates.
 - b. Revise plan to create controllable activities.
6. Identify activity risks :
 - a. Identify and quantify activity-based risks.
 - b. Plan risk reduction and contingency measures where appropriate.

- c. Adjust plans and estimates to take account of risks.
- 7. Allocate resources :
 - a. Identify and allocate resources.
 - b. Revise plans and estimates to account for resource constraints.
- 8. Review/publish plan :
 - a. Review quality aspects of project plan.
 - b. Document plans and obtain agreements.
- 9. Execute plan/lower levels of planning :
 - a. This may require the reiteration of the planning process at a lower level.

Que 1.17. How do we identify the planning objectives ?

Answer

1. After all the stakeholders are identified, it is critical to identify what each stakeholder expects to gain from the project.
2. For instance, for an end user it might be that they are expecting a very user friendly and robust software with a number of features, while for a maintainer it is the quality of the documentation and the modifiability of the system that are important.
3. A manager would not want any overruns to the schedule, while the person who approves the budget might expect a low budget project.
4. By identifying each of the stakeholders "win" condition, the project's objectives are clear from the start.
5. In the negotiation with the customer, these objectives should be reinforced and documented.
6. Steps to identify the objectives are as follows :
 - a. Identify and allocate resources.
 - b. Understand how people will benefit from the project.
 - c. Prioritize objectives for the project.
 - d. Establish reasonable expectations on the parts of all the stakeholders.
 - e. Transform these objectives into project activities.
 - f. Identify and manage the risks.
7. Keep people involved. Keep senior management and the customer aware of the status of the project at all times.

Que 1.18. When does the software planning activity start and end in software life cycle ? List some important activities that a software project manager performs during software project planning ?

UPTU 2014-15, Marks 05

Q.1.15. OR
Write short note on software project planning and its scope.

UPTU 2013-14, Marks 05

Answer

Software project planning : Refer Q. 1.15, Page 19C, Unit-1.

- Project planning spans across the various aspects of the project. Generally project planning is considered to be a process of estimating, scheduling and assigning the projects resources in order to deliver an end product of suitable quality.
- However, it is much more as it can assume a very strategic role, which can determine the success of the project.
- A project plan is one of the crucial steps in project planning. Typically, project planning can include the following types of project planning:
 - Project scope definition and scope planning :**
 - In this step, we document the project work that would help us achieve the project goal.
 - We document the assumptions, constraints, user expectations, business requirements, technical requirements, project deliverables, project objectives and everything that defines the final product requirements.
 - This is the foundation for a successful project completion.
 - Quality planning :**
 - The relevant quality standards are determined for the project.
 - This is an important aspect of project planning.
 - Based on the inputs captured in the previous steps such as the project scope, requirements, deliverables, etc., various factors influencing the quality of the final product are determined.
 - The processes required to deliver the product as promised and as per the standards are defined.
 - Project activity definition and activity sequencing :**
 - In this step, we define all the specific activities that must be performed to deliver the product by producing the various product deliverables.
 - The project activity sequencing identifies the interdependence of all the activities defined.
 - Time, effort and resource estimation :**
 - Once the scope, activities and activity interdependence is clearly defined and documented, the next crucial step is to determine the effort required to complete each of the activities.

- The effort can be calculated using one of the many techniques available such as function points, lines of code, complexity of code, benchmarks, etc.
- This step clearly estimates and documents the time, effort and resource required for each activity.
- Risk factors identification :**
 - "Expecting the unexpected and facing it".
 - It is important to identify and document the risk factors associated with the project based on the assumptions, constraints, user expectations, specific circumstances, etc.
- Schedule development :**
 - The time schedule for the project can be arrived on the basis of the activities, interdependence and effort required for each of them.
 - The schedule may influence the cost estimates, the cost benefit analysis and so on.
 - Project scheduling is one of the most important task of project planning and also the most difficult task.
 - In very large projects, it is possible that several teams work on developing the project.
 - They may work on it in parallel. However, their work may be interdependent.
 - Popular tools can be used for creating and reporting the schedules such as Gantt charts.
- Cost estimation and budgeting :**
 - Based on the information collected in all the previous steps, it is possible to estimate the cost involved in executing and implementing the project.
 - A cost benefit analysis can be arrived at for the project.
 - Based on the cost estimates, budget allocation is done for the project.
- Organizational and resource planning :**
 - Based on the activities identified, schedule and budget allocation, resource types and resources are identified.
 - One of the primary goals of resource planning is to ensure that the project is run efficiently.
 - This can only be achieved by keeping all the project resources fully utilized as possible.
 - The success depends on the accuracy in predicting the resource demands that will be placed on the project.
 - Resource planning is an iterative process and necessary to optimize the use of resources throughout the project life cycle thus making the project execution more efficient.

24 (CS/IT-7) C**Introduction and Software Project Planning**

- f. There are various types of resources : equipment, personnel, facilities, money, etc.
- 9. Risk management planning :**
- Risk management is a process of identifying, analyzing and responding to a risk.
 - Based on the risk factors, identified risk resolution plan is created.
 - The plan analyses each of the risk factors and their impact on the project. The possible responses for each of them can be planned.
 - Throughout the lifetime of the project, these risk factors are monitored and acted upon as necessary.
- 10. Project plan development and execution :**
- Project plan development uses the inputs gathered from all the other planning processes such as scope definition, activity identification, activity sequencing, quality management planning, etc.
 - A detailed work breakdown structure comprising of all the activities identified is used.
 - The tasks are scheduled based on the inputs captured in the steps previously described.
 - The project plan documents all the assumptions, activities, schedule, timelines and drives the project.
- 11. Performance reporting :**
- The progress of each of the tasks/activities described in the project plan is monitored.
 - The progress is compared with the schedule and timelines documented in the project plan.
 - Various techniques are used to measure and report the project performance such as EVM (Earned Value Management).
 - A wide variety of tools can be used to report the performance of the project such as PERT charts, Gantt charts, Logical bar charts, Histograms, Pie charts, etc.
- 12. Planning change management :**
- Analysis of project performance can necessitate that certain aspects of the project be changed.
 - The Requests for Changes need to be analyzed carefully and its impact on the project should be studied.
 - Considering all these aspects the project plan may be modified to accommodate this request for change.
- Activities performed during software project planning are :**
Refer Q. 1.14, Page 18C, Unit-1.

Software Project Management**25 (CS/IT-7) C**

Que 1.19. What are the various dimensions of software project planning ? Discuss the impact of each section of a software project planning on the successful completion of the software project.

UPTU 2012-13, Marks 10

Answer

The mnemonic SMART is sometimes used to describe well-defined objectives of software project planning such as :

- Specific :**
 - Effective objectives are concrete and well defined. Vague aspirations such as 'to improve customer relations' are unsatisfactory.
 - Objectives should be defined in such a way that it is obvious to all whether the project has been successful or not.
 - Measurable :**
 - Ideally there should be measures of effectiveness which tell us how successful the project has been.
 - For example 'to reduce customer complaints' would be more satisfactory as an objective than 'to improve customer relations'.
 - The measure can, in some cases, be an answer to simple yes/no question, for example 'Did we install the new software by 1 June ?'
 - Achievable :** It must be within the power of the individual or group to achieve the objective.
 - Relevant :** The objective must be relevant to the true purpose of the project.
 - Time constrained :** There should be a defined point of time by which the objective should have been achieved.
- Impact of each section of a software project planning on the successful completion of the software project :** Refer Q. 1.18, Page 21C, Unit-1.
- Que 1.20.** What do you mean by project plan ? Also, discuss types of project plans.

UPTU 2014-15, Marks 05

Answer

Project plan : Refer Q. 1.15, Page 19C, Unit-1.

There are various types of project plan :

- Quality plan :** It describes the quality procedures and standards that will be used in project.
- Validation plan :** It describes the approach, resources and schedule used for system validation.
- Configuration management plan :** It describes the configuration management procedures and structures to be used.

4. **Maintenance plan :** It predicts the maintenance requirements of the system, maintenance costs and effort required.
5. **Staff development plan :** It describes how the skills and experience of the project team members will be developed.

Que 1.21. Write a short note on pragmatic planning.

UPTU 2012-13, Marks 05

Answer

- a. In the pragmatic planning, the problem is solved in a practical and sensible way rather than by having ideas or theories.

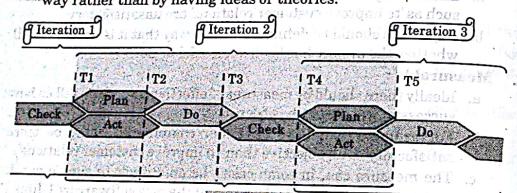


Fig. 1.21.1

- b. The set of activities are done in a particular period of time. The set of activities are called the iterations.
- c. These iterations are overlapped in time in order to optimize people allocation, use of resources and guarantee the feedback from previous iterations.
- d. Each iteration is performed in a fixed period of time. There are four set of activities : plan, do, check and act.
- e. In the planning process there are following five events :
- T1 : It represents the beginning of the iteration. In this, we do the initial planning. The scope of the planning covers only the period of the current iteration. All members of the team participate in the planning.
 - T2 : In this iteration, planned activities are executed. All members of the team must have something to do within the scope of the iteration. The activities for future iterations should not be planned in the current iteration. All sort of output may be produced such as documents, code, reports, meeting minutes, etc.
 - T3 : In this iteration, everything that is produced is checked. Documents are reviewed, codes are tested, user interfaces and integrations with other systems are tested, etc. All found issues must be registered to be solved in due time.

4. T4 :

- All issues found during the check phase are solved and all planned deliverables are released. Everybody should deliver something.
 - In case time is not enough to solve some issues, they are included in the planning of the next iteration with the highest priority.
 - The planning of the next iteration also starts at this point, taking advantage of the experience from the previous iteration.
5. T5 : Once everything is released, the iteration finishes. T2 of the next iteration immediately starts because most of people and resources are already available. T1 to T5 repeats several times, in a fixed period of time, until the end of the project.

Que 1.22. Discuss the structure of a Software Project Management Plan (SPMP) in detail.

UPTU 2014-15, Marks 05

Answer

The structure of software project plan is as follows :

1. Overview

- Project purpose, objectives, and success criteria
- Project deliverables
- Assumptions, dependencies, and constraints
- References
- Definitions and acronyms
- Evolution of the plan

2. Project organization

- External interfaces
- Internal structure
- Roles and responsibilities

3. Managerial process plans

- Start-up plans
 - Estimation plan
 - Staffing plan
 - Staff training plan
 - Resource acquisition plan
 - Project commitments

b. Work plan

c. Control plan

- i. Data control plan
 - ii. Requirements control plan
 - iii. Schedule control plan
 - iv. Budget control plan
 - v. Communication, tracking, and reporting plan
 - vi. Metrics collection plan
 - vii. Risk management plan
 - viii. Issue resolution plan
 - ix. Project close-out plan
- 4. Technical process plans**
- a. Process model
 - b. Methods, tools, and techniques
 - c. Configuration management plan
 - d. Quality assurance plan
 - e. Documentation plan
 - f. Process improvement plan

PART-3

Software Project Estimation, Estimation methods, Estimation models, Decision Process.

CONCEPT OUTLINE : PART-3

- According to Kelkar, estimation is defined as process of reliably predicting the various parameters associated with making a project, i.e., size, effort, cost, time and quality.
- Various software estimating techniques are :
 - a. Parkinson's law
 - b. Function point
 - c. Top-down approach
 - d. Bottom-up approach
 - e. Estimation by analogy
- According to Boehm, software cost estimation should be done through three stages :
 - a. Basic COCOMO model
 - b. Intermediate COCOMO model
 - c. Complete COCOMO model

Questions-Answers**Long Answer Type and Medium Answer Type Questions**

Que 1.23. What do you mean by software project estimation ?

UPTU 2014-15, Marks 05

OR

Explain software project estimation. Why software project estimation is needed ?

Answer

1. Delivery of an error free product i.e., software within budget and schedule is the topmost priority of any organization.
2. This in turn calls for reasonably good estimates of effort, time and cost and of course quality which affects the schedule and effort.
3. It is seen that in order to produce good quality software, extra effort and longer schedule is required.
4. While computing the cost of project, one of the major components is the cost of effort applied by skilled software professionals in terms of person hours, person days or person-months.
5. Other factors which contribute to total cost are cost of hardware and software, traveling, cost of tools, training etc.
6. Hence, key to accurate estimation of cost is accurate estimation for the manpower that in turn also decides the team size.
7. Estimation of size of the software to be developed is also very important as it forms the basis for effort and schedule estimation.
8. A number of estimation methods to measure size in terms of number of lines of code, function points, object points etc. are proposed.
9. Software effort estimation is important because of following reasons :
 - a. Organizations have proper control over project and they can plan systematically.
 - b. There is a clear understanding of the product.
 - c. Estimation also determines the project feasibility in terms of budget and time constraints.
 - d. It helps in identification of resources to be used during the project.
 - e. Estimation also helps management in taking decision for current as well as future projects.
 - f. Estimation helps in quantifying the impact of risks and guides the project manager to take suitable decision.

Que 1.24. Is software project estimation essential for all kinds of software project? Discuss various software project estimation techniques, together with their merits and demerits.

Answer

Need of software project estimation : Refer Q. 1.23, Page 29C, Unit-1. Boehm has discussed seven techniques of software effort estimation. They are described briefly as follows:

1. Algorithmic cost modeling :

- a. This approach is based on historical cost information.
- b. In this, a model is developed which relates the project cost to some software metric which is usually the size in this case.
- c. The most common size metric is the number of Lines of Code (LOC) of the final product but that is not easy to predict in the beginning.
- d. Code size based estimation is also uncertain because number of factors contributes for computing the final number of LOC, for example, hardware used, software choice, type of DBMS used etc.

2. Expert judgment :

- a. In this approach, experience and judgment of one or more experts on that particular application domain is used for the project estimation by extrapolating expert's experience.
- b. Each of the expert estimates the project cost and final cost estimation is decided by consensus.
- c. The problem with this technique is the limited expertise of the experts and hence should be used when other options are not available.

3. Estimation by analogy :

- a. This technique is used when the organization in the past have completed the projects of the similar nature.
- b. For example, an organization for last few years has developed expertise in the banking domain, which can be used in future projects.
- c. In this technique, comparisons are made with the past projects and estimates are proposed.

4. Parkinson's law :

- a. According to Parkinson's law, work expands to fill the available time and budget.
- b. It means that the Parkinson's approach to cost estimation assumes that the time for completion and available resources are known prior to starting the project.

- c. As a result, in some cases it can also result into over estimates.

5. Pricing to win approach :

- a. In the pricing to win approach, cost of the project is proportional to the budget of the customer.
- b. As a result, the estimated effort does not depend upon the software functionality but also on the customer's capability to spend.
- c. This approach, therefore often results into poor quality product, schedule overrun and over budgeting.

6. Top-down estimation :

- a. In this approach, the whole project is decomposed into number of phases/tasks and estimation for each phase is done using different approach.
- b. In top-down estimation, whole functionality of the product is decomposed into sub-functions and cost estimation is done by using these logical sub-functions.
- c. It is therefore suitable for using early in software life cycle.

7. Bottom-up estimation :

- a. In this approach, instead of logical functions, components implementing these functions are used for cost estimation.
- b. Each component cost is estimated and then added to give the final cost estimate.
- c. Bottom-up estimation techniques are appropriate at detailed stages of project planning.
- d. In practice, both top-down as well as bottom-up approaches are used.

Que 1.25. Explain Bottom-up estimating.**Answer**

- i. Estimating methods can be generally divided into bottom-up and top-down approaches.
- ii. With the bottom-up approach, the estimator breaks the project into its component tasks and then estimates how much effort will be required to carry out each task.
- iii. With a large project, the process of breaking down into tasks would be a repetitive one: each task would be analyzed into its component subtasks and these in turn would be further analyzed.
- iv. This is repeated until we get to components that can be executed by a single person in about a week or two.
- v. The reader might wonder why this is not called a top-down approach: after all we are starting from the top and working down. Although this

32 (CS/IT-7) C**Introduction and Software Project Planning**

- top-down analysis is an essential precursor to bottom-up estimating, it is really a separate one -that of producing a Work Breakdown Structure (WBS).
- vi. The bottom-up part comes in adding up the calculated effort for each activity to get an overall estimate.
 - vii. The bottom-up approach is most appropriate at the later, more detailed, stages of project planning. If this method is used early on in the project cycle then the estimator will have to make some assumptions about the characteristics of the final system, for example, the number and size of software modules.
 - viii. These will be working assumptions that imply no commitment when it comes to the actual design of the system.
 - ix. Where a project is completely novel or there is no historical data available, the estimator would be forced to use the bottom-up approach.

Que 1.26. Explain the top-down approach and parametric models.

Answer

- i. The top-down approach is normally associated with parametric (or algorithmic) models. These may be explained using the analogy of estimating the cost of rebuilding a house.
 - ii. This would be of practical concern to a house-owner who needs sufficient insurance cover to allow for rebuilding their property if it were destroyed. Unless the house-owner happens to be in the building trade it is unlikely that he or she would be able to work out how many bricklayer-hours, how many carpenter hours, electrician-hours and so on would be required.
 - iii. Insurance companies, however, produce convenient tables where the house-owner can find an estimate of rebuilding costs based on such parameters as the number of storeys and the floor space that a house has.
 - iv. This is a simple parametric model.
 - v. The effort needed to implement a project will be related mainly to variables associated with characteristics of the final system.
 - vi. The form of the parametric model will normally be one or more formulae in the form :
- $$\text{effort} = (\text{system size}) \times (\text{productivity rate})$$
- vii. For example, system size might be in the form "thousands of lines of code" (KLOC) and have the specific value of 3 KLOC while the productivity rate 40 days per KLOC.
 - viii. The values to be used will often be matters of subjective judgment. A model to forecast software development effort therefore has two key components.

Software Project Management**33 (CS/IT-7) C**

- ix. The first is a method of assessing the size of the software development task to be undertaken. The second assesses the rate of work at which the task can be done.
- x. For example, Amanda at IOE might estimate that the first software module to be constructed is 2 KLOC.
- xi. She might then judge that if undertook the development of the code, with her expertise she could work at a rate of 40 days per KLOC per day and complete the work in 2×40 days, that is, 80 days, while Ken, who is less experienced would need 55 days per KLOC and take 2×55 that is, 110 days to complete the task.
- xii. Some parametric models, such as that implied by function points, are focused on system or task size, while others, such as COCOMO, are more concerned with productivity factors.
- xiii. Having calculated the overall effort required, the problem is then to allocate proportions of that effort to the various activities within that project.

Que 1.27. Why software project estimation is needed ? What are various estimation approaches and when these are used ? Explain any one estimation method with example.

UPTU 2012-13, Marks 10

OR
What do you understand by function point ? Discuss limitation and benefits of function point.

Answer

Need of software project estimation : Refer Q. 1.23, Page 29C, Unit-1.
Various estimation approaches : Refer Q. 1.24, Page 30C, Unit-1.
One estimation method with example :

1. Function point metric was proposed by Allan Albrecht.
2. This metric overcomes many of the shortcomings of the LOC metric.
3. One of the important advantages of using the function point metric is that it can be used to easily estimate the size of a software product directly from the problem specification.
4. This is contrast to the LOC metric, where the size can be accurately determined only after the product has been fully developed.
5. The conceptual idea behind the function point metric is that the size of a software product is directly dependent on the number of different functions or features it supports.
6. A software product supporting many features would certainly be of larger size than a product with less number of features. Each function when invoked reads some input data and transforms it to the corresponding output data.

7. Function point measures functionality from the user's point of view, that is, on the basis of what the user requests and receives in return from the system.
8. The principle of Albrecht's function point analysis (FPA) is that a system is decomposed into functional units.
- a. Inputs : information entering the system.
 - b. Outputs : information leaving the system.
 - c. Enquiries : requests for instant access to information.
 - d. Internal logical files : information held within the system.
 - e. External interface files : information held by other systems, that is, used by the system being analyzed.
9. The five functional units are divided in two categories :
- Data function type :**
 - Internal Logical Files (ILF)** : A user identifiable group of logically related data or control information maintained within the system.
 - External Interface Files (EIF)** : A user identifiable group of logically related data or control information referenced by the system, but maintained within another system. This means that EIF counted for one system, may be ILF in another system.
 - Transactional function types :**
 - External Input (EI)** : An EI processes data or control information that comes from outside the system. The EI is an elementary process, which is the smallest unit of activity, that is, meaningful to the end user in the business.
 - External Output (EO)** : An EO is an elementary process that generates data or control information to be sent outside the system.
 - External Inquiry (EQ)** : An EQ is an elementary process, that is, made up of an input output combination that results in data retrieval.

Advantages of function point :

1. It is not restricted to code.
2. Language independent.
3. More accurate than estimated LOC.

Drawbacks of function point :

1. Subjective counting.
2. Hard to automate and difficult to compute.
3. Ignores quality of output.
4. Oriented to traditional data processing application.
5. Efforts prediction using the unadjusted function count is often no worse than when the TCF is added.

Que 1.28. What do you mean by the software project estimation ? Give various estimation models. Describe any one of the estimation models using suitable examples.

UPTU 2013-14, Marks 10

OR

Give various estimation models. Describe any one of the estimation model using suitable examples.

UPTU 2014-15, Marks 05

Answer

Software project estimation : Refer Q. 1.23, Page 29C, Unit-1.

Various estimation models are :

COCOMO model :

1. COCOMO (COnstructive COst estimation MOdel) was proposed by Boehm in 1981.
2. Boehm postulated that any software development project can be classified into one of the following three categories based on the development complexity : organic, semidetached, and embedded.
3. In order to classify a product into the identified categories, Boehm requires us to consider not only the characteristics of the product but also those of the development team and development environment.
4. These three product classes correspond to application, utility, and system programs, respectively.
5. Normally, data processing programs are considered to be application programs.
6. Compilers, linkers, etc. are utility programs.
7. Operating systems and real time system programs are system programs.
8. System programs interact directly with the hardware and typically involve meeting timing constraints and concurrent processing.
9. Boehm's definitions of organic, semidetached, and embedded systems are elaborated as follows :

a. **Organic** : We can consider a development project to be of organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar types of projects.

b. **Semidetached** : A development project can be considered to be of semidetached type, if the development team consists of a mixture of experienced and inexperienced staff. Team members may have limited experience on related systems but may be unfamiliar with some aspects of the system being developed.

Introduction and Software Project Planning

36 (CS/IT-7) C

- c. **Embedded**: A development project is considered to be of embedded type, if the software being developed is strongly coupled to complex hardware, or if stringent regulations on the operational procedures exist.
- 10. According to Boehm, software cost estimation should be done through three stages : Basic COCOMO, Intermediate COCOMO, and Detailed/ Complete COCOMO.

a. Basic COCOMO model :

- i. The basic COCOMO model gives an approximate software development efforts and cost as function of program size expressed in estimated lines of code.
 - ii. The basic COCOMO estimation model is given by the following expressions :
- Effort (E) = $a * (KLOC)^b$
 Development Time (T_{dev}) = $c * (E)^d$
- Where E is effort applied in person-month, T_{dev} is development time in months.
- iii. The coefficients a , b , c , d are constant and can be calculated by given table :

Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

- iv. When effort and development time are known, the average staff size to complete the project may calculated as :
 $\text{Average staff size (SS)} = E/T_{dev}$ persons
 - v. When project size is known, the productivity level may be calculated as :
 $\text{Productivity (P)} = \text{KLOC}/E$
- ### b. Intermediate COCOMO :
- i. The basic COCOMO model assumes that effort and development time are functions of the product size alone.
 - ii. However, a host of other project parameters besides that product size affect the effort required to develop the product as well as the development time.
 - iii. Therefore, in order to obtain an accurate estimation of the effort and project duration, the effect of all relevant parameters must be taken into account.
 - iv. The intermediate COCOMO model recognizes this fact and refines the initial estimate obtained through the basic COCOMO

Software Project Management

37 (CS/IT-7) C

expressions by using a set of fifteen cost drivers (multipliers) based on various attributes of software development.

- v. For example, if modern programming practices are used, the initial estimates are scaled downwards by multiplication with a cost driver having a value less than one.
- vi. If there are stringent reliability requirements on the software product, this initial estimate is scaled upward.
- vii. Boehm requires the project manager to rate these fifteen different parameters for a particular project on a scale of one to three.
- viii. Then depending on these ratings, he suggests appropriate cost driver values which should be multiplied with the initial estimate obtained using the basic COCOMO.
- ix. In general, the cost drivers can be grouped into four categories :

1. **Product attributes** : The characteristics of the product that are considered include the inherent complexity of the product (CPLX), reliability requirements of the product (RELY) and database size (DATA).

2. **Computer attributes** : The characteristics of the computer that are considered include execution time constraints (TIME), main storage constraints (STOR), virtual machine volatility (VIRT) and computer turnaround time (TURN).

3. **Personnel** : The attributes of development personnel that are considered include the analyst capability (ACAP), application experience (AEXP), programmer capability (PCAP), virtual machine experience (VEXP) and programming language experience (LEXP).

4. **Project attribute** : The characteristics of the project that are considered includes modern programming practices (MODP), use of software tools (TOOL) and required development schedule (SCED).

x. The intermediate COCOMO equations are :

$$\text{Effort (E)} = a * (KLOC)^b * \text{EAF}$$

$$\text{Development Time (T}_{dev}\text{)} = c * (E)^d$$

xii. The EAF (Effort Adjustment Factor) is multiplication of different types of cost drivers. E is an effort applied in person-month, T_{dev} is development time in months.

xiii. The coefficients a , b , c , d are constant and can be calculated by given table :

Project	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semidetached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

c. Detailed/complete COCOMO model:

- i. A major shortcoming of both the basic and the intermediate COCOMO models is that they consider a software product as a single homogeneous entity.
- ii. However, most large systems are made up of several smaller subsystems. These subsystems may have widely different characteristics.
- iii. For example, some subsystems may be considered as organic type, some semidetached, and some embedded.
- iv. Not only that the inherent development complexity of the subsystems may be different, but also for some subsystems the reliability requirements may be high, for some the development team might have no previous experience of similar development, and so on.
- v. The complete COCOMO model considers these differences in the characteristics of the subsystems and estimates the effort and development time as the sum of the estimates for the individual subsystems.
- vi. The cost of each subsystem is estimated separately. This approach reduces the margin of an error in the final estimate.

2. COCOMO-II :

- i. COCOMO-II is the revised version of the original COCOMO and is developed at University of Southern California under the leadership of Dr. Barry Boehm.
- ii. The model is turned to the life cycle practices of the 21st century. It also provides a quantitative analytic framework, and set of tools and techniques for evaluating the effects of software technology improvements on software life cycle costs and schedules.
- iii. COCOMO-II provides three detailed cost estimation models. These can be used to estimate project costs at different phases of the software.
- iv. As the project progresses, these models can be applied at different stages of the same project.
 - a. **Application composition :** Here, the external features of the system that the users will experience are designed. Prototyping will typically be employed to do this with small applications that can be built using high-productivity application-building tools, development can stop at this point.
 - b. **Early design :** Here, the fundamental software structures are designed. With larger, more demanding systems, where, for example, there will be large volumes of transactions and performance is important, careful attention will need to be paid to the architecture to be adopted.

- c. **Post architecture :** Here, the software structures undergo final construction, modification and tuning to create a system that will perform as required.

Que 1.29. Discuss the role of cost estimation in a software development project. Briefly explain COCOMO model for cost estimation for all category of projects.

UPTU 2014-15, Marks 05

Answer In cost estimation, the number of estimation techniques have been developed and are having following attributes in common :

1. Project scope must be established in advance.
 2. Software metrics are used as a basis from which estimates are made.
 3. The project is broken into smaller pieces which are estimated individually.
- To achieve reliable cost and schedule estimates, a number of options arise :
1. Delay estimation until late in project.
 2. Use simple decomposition techniques to generate project cost and schedule estimates.
 3. Develop empirical models for estimation.
 4. Acquire one or more automated estimation tools.

COCOMO model : Refer Q. 1.28, Page 35C, Unit-1.

Que 1.30. Write short note on decision process.

UPTU 2013-14, Marks 05

Answer

1. Software project management (SPM) is basically defined by the ability of decision making.
2. It is the responsibility of managers to design this process and optimize it to minimize costs and maximize production.
3. Decision makings are based on resources and constraints which are planned for the target project and the plan could change hardly in line with new requirements during project progress.
4. But what could be done accordingly to confront changes which are contingent in every project, is to define an optimal plan with effective decisions.
5. Decision making is a cognitive process resulting in the selection of a course of action among several alternative scenarios.
6. This process finally leads the decision maker to take an action or make a choice.

7. Thereby, it is an ability based on experience and knowledge that enables a leader of a process to succeed.
8. The nature of software projects on the other side add other complexity in which development process has intangibility that makes it difficult for managers to design a suitable strategy for decision making.
9. SPM requires special mindset to make practitioners be able to conduct management process in an effective and efficient manner.
10. These mindsets are from any point of view considered as a high level experience and management capabilities, since it originates from a complex process and organizational understanding.
11. Therefore, management of this knowledge requires special strategies for an effective knowledge management.
12. It is evident that an effective approach for modeling the decision making process and redefining the decision structure over SPM is necessary.
13. This model should be able to deal with high level of intangibility and continuous change requests within the software project.

VERY IMPORTANT QUESTIONS

Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.

Q. 1. Write short note on :

- Project
- Characteristics of project
- Project management

ANS: *Project Management (PM) is a discipline involving application of knowledge, skills, tools and techniques to project activities to meet the specified requirements.*

Q. 2. Define SPM and life cycle of software project.

ANS: *Refer Q. 1.6 and 1.7.*

Q. 3. Write short note on : *minimum and maximum life cycle*

- Need and identification
- Vision and scope document

ANS: *Refer Q. 1.9.*

Q. 4. Explain project management cycle and SPM objectives.

ANS: *Refer Q. 1.10 and 1.11.*

Q. 5. What do you mean by software project planning and planning objectives ?

ANS: *Refer Q. 1.15 and 1.17.*

Q. 6. Discuss the following :

- Project plan and its types.
- Structure of project management plan.

ANS: *i. Refer Q. 1.20.*

ii. Refer Q. 1.22.

Q. 7. What do you mean by software project estimation ? Also discuss various techniques and models associated to it.

ANS: *Refer Q. 1.23, Q. 1.27 and 1.28.*

Q. 8. Define decision process.

ANS: *Refer Q. 1.30.*

