# UNIT - 1

# PROJECT EVALUATION AND PROJECT PLANNING

## PART - A

**1. What is a Project?**

The dictionary definitions put a clear emphasis on the project being a planned activity. A project is a unique venture with a beginning and an end, conducted by people to meet established goals within parameters of cost, schedule and quality.

**2. What are the characteristics of a project?(Nov/Dec2011)(Nov/Dec2012)**

- Non-routine tasks are involved
- Planning is required
- Specific objects are to be met or a specified product is to be correct
- The project has a predetmined time span.

**3. What is the differentsoftware projects and other types of project?(May/June2012)**

- İnvisibility- Software can't be rep-resented with geometric models
- Complexity- The proposed model is based on the widely known and accepted
- Confirmity- The controlling document for a software
- Flexibility- project management performance

**4. Why organize an activity or job as a project?**

- It allows you to better structure and organize the tasks that need to be performed
- Well developed approaches and tools are available for managing projects
- Easy-to-use software is available for scheduling and budgeting projects.

**5. Define Contract Management. (May/Jun2013)(Apr2014).**

Contract management or contract administration is the management of contracts made with customers, vendors, partners, or employees. Contract management includes negotiating the terms and conditions in contracts and ensuring compliance with the terms and conditions, as well as documenting and agreeing on any changes that may arise during its implementation or execution. It can be summarized as the process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk.

**6. What are the Technical Project Planning Methodologies?**

- Identify different approaches to planning technical projects: rolling wave
- Planning…stage gate process…critical chain project management
- Common construction project life cycle
- Common pharmaceutical project life cycle

**7. What are the three successive processes that bring a new system? (Nov/Dec2012)**

- The feasibility study- Evaluate the cost of the software development against the Software Engineering Planning-outline the structure of the project
- Project Execution- Product Implementation Product implementation activities

**8. Define Feasibility Study.**

It is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc.

**9. What is meant by planning?**

Planning as a process involves the determination of future course of action, that is, why an action, what action, how to take action, and whento take action. These why, what, how, and when are related with different aspects of planning process.

**10. What are the phases in software development life cycle?**

- Requirement analysis
- Architecture design
- Detailed design
- Code and test
- Integration
- Qualification testing.
- Installation.
- Acceptance support

**11. Define Requirement Analysis.**

This investigates what the potential users and their managers and employers require as features and qualities of the new system.

**12. What is meant by qualification testing?**

The system, including the software components, has to be tested carefully to ensure that all the requirements have been fulfilled.

**13. What is the difference between Information systems and embedded systems?**

**Information systems:-**

Information System includes databases that include useful "information". Information Systems is the discipline concerned with the development, use, application and influence of information systems. An information system, following a definition of Lange fors, is a technologically implemented medium for recording, storing, and disseminating linguistic expressions, as well as for drawing conclusions from such expressions.

The technology used for implementing information systems by no means has to be computer technology. A notebook in which one lists certain items of interest is, according to that definition, an information system. Likewise, there are computer applications that do not comply with this definition of information systems. Embedded systems are an example.

**Embedded Systems**: Embedded systems include small computers that make things work, such as the computer in your radio, television or the computer that controls your vehicle engine. An embedded system is a computer systems that is part of a larger system.

**Examples:**

- Washing machine
- Car engine control
- Mobile phone

**14. Differentiate Objectives Vs products.**

**Objectives** are goals or aims which the management wishes the organization to achieve.

These are the end points or pole-star towards which all business activities like organizing, staffing, directing and controlling are directed.

A project might be to create a **product**, the details of which have been specified by the client.

The client has the responsibility for justifying the product.

**15. What is management?**

Management can be defined as all activities and tasks undertaken by one or more Persons for the purpose of planning and controlling the activities of others in order to achieve objectives or complete an activity that could not be achieved by others acting independently.

**16. What are the activities of management? (Apr2014)**

- Planning –Deciding what is to be done.
- Organizing – making arrangements.
- Staffing-selecting the right people for the job
- Directing-giving instructions.
- Monitoring – checking on progress
- Controlling- taking action to remedy hold-ups
- In Novating-coming up with new solutions.
- Representing – liaising with clients, users , developers , suppliers

**17. What are the problems with software project from manager's point of view? (May/Jun2013)**

- Poor estimates and plans.
- Lack of quality standards and measures.
- Lack of techniques to make progress visible.
- Lack of guidance about organizational Decisions.
- Poor role definition. 6.Incorrect success criteria

**18. What are the problems with software project from student's point of view? (May/Jun2013)**

- Inadequate specification of work.
- Lack of knowledge of application area.
- Lack of standards.
- Narrow scope of technical expertise.

**19. What is meant by management control?**

The process of setting objectives for a system and then monitoring the systems to see what is true performance, A change is proposed by anyone evaluating the software.

**20. What are the steps involved in step wise planning?**

- Identify project scope and objectives.
- Identify project infrastructure.3.Analyze project characteristics.
- Identify project products and activities.
- Estimate effort for each activity.
- Identify activity risks.
- Allocate resources.
- Review / publicize plan9. Execute plan/ lower levels of planning.

**21. How to identify project infrastructure?**

- Establish relationship between project and strategic planning.
- Identify installation standards and procedures.
- Identify project team organization.

**22. How to manage activity risks?**

- Identify and quantify activity-based risks.
- Plan risk reduction and contingency measures where appropriate
- Adjust plans and estimates to take account of risks.

**23. Define project stake holders.**

Stakeholders are the people involved in or affected by the project actives **Stake holders power**-Integrate all expectations of several people.

**24. How to review and publicize plan?**

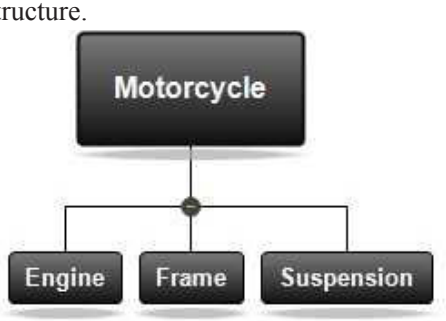
- Review quality aspects of project plan
- Document plans and obtain agreement.

**25. Define process. (Nov/Dec2011)**

A software process provides the framework from which a comprehensive plan for software development can be established.

**26. What is a product breakdown structure (PBS) ? Show the hierarchical diagram of sample PBS. (May/Jun2012)**

A product breakdown structure is an effective tool that details the physical components of a particular product, or system, under consideration. The formal PBS comes in the form of a hierarchy. It begins with the final product at the top of the hierarchy followed by the subcategorized elements of the product. The product breakdown structure is similar to the work breakdown structure (WBS). Like WBS, a product breakdown structure serves to reduce a complex project, or product, into manageable components. As a result, teams can obtain a clear understanding of a product, its components, and what is required to provide those components. Figure 1 (below) is a sample product breakdown structure.



**27. Define project Evaluation.**

**Project evaluation** is a systematic method for collecting, analyzing, and using information to answer questions about projects, policies and programs, particularly about their effectiveness and efficiency.

**28. What is meant by programme?**

D.C. Ferns defined a programme as " a group of project that are managed in a coordinated way to gain benefits that would not be possible were the projects to be managed independently".

**29. What is the concept of strategic programmes?**

Several projects together can implement a single strategy. For example the merging of two organizations could involve the creation of unified payroll and accounting applications.

**30. Define business cycle programmes.**

The collection of projects that an organization undertakes within a particular planning cycle is sometimes refer to portfolio. Decisions have to be made about which projects to implement within that budget within the accounting period.

**31. Define Infrastructure programmes.**

Some organizations have very integrated information systems. The distinct activities can be integrated.

**32. Define Research and development programmes**

Truly inNovative companies especially those that are trying to develop new product for the market, are well aware that projects will vary in terms of their risk of failure and the potential returns.

**33. Write the difference between programme managers and project managers.**

| Programme manager | Project manager |
|---|---|
| Many simultaneous projects | One project ata time |
| Personal relationship with skilled resources | Impersonal relationship with resource type |
| Need to maximize utilization of resources | Need to minimize demand for resources |
| Projects tend to be similar | Projects tend to be dissimilar |

**34. Define programme mandate.**

- This should include the new services or capabilities the programme should deliver.
- How the organization will be improved by use of the new services or capability.
- How the programme fits with corporate goals and any other initiatives

**35. How the programme will brief?**

A programme brief is now produced which would be the equivalent of a feasibility study for the programme,used by achievers in all fields.

**36. Define vision statement.**

A preliminary vision statement which describes the new capacity that the organization seeks.

**Significance-**When the project begins, the project **...** The goal of the *vision statement* is to describe what the project is expected.

**37. What is meant by blueprint?**

The achievement of the improved capability described in the vision statement can only come about when changes have been made to the structure and operations of the organizations. These are detailed in the blueprint.

**38. What are things to be considered in the blueprint?**

- Business models outlining the new process required.
- Organization structure-The information systems
- Data and information requirements
- Costs, performance and service level requirements.

**39. What are the benefits of management?**

1) Mandatory compliance
2) Quality of service
3) Productivity
4) More motivated force

5) Internal management benefits

6) Risk reduction

**40. Define technical assessment. (May/Jun2013)**

Technical assessment of a proposed system consists of evaluating the required functionality against the hardware and software available. Organizational policy aimed at the provision of a uniform and consistent hardware/software infrastructure is likely to place limitations on the nature of technical solutions that might consider.

**41. What are the steps in cost-benefit analysis?**

- Identifying and estimating all of the costs and benefits of carrying out the project and operating the delivered application.
- Expressing these costs and benefits in common units.

**42. Define development costs.**

Development costs include the salaries and other employment costs of the staff involved in the development project and all associated costs.

- TDEV = 3 ´ $(PM)^{(0.33+0.2*(B-1.01))}$
- PM is the effort computation and B is the exponent computed as discussed above (B is 1 for the early prototyping model). This computation predicts the nominal schedule for the project.

**43. Define setup costs.**

Setup costs include the costs of putting the system into place. These consists of mainly the costs of the new hardware

- ESLOC = ASLOC * (1-AT/100) * AAM.
- ASLOC and AT as before.
- AAM is the adaptation adjustment multiplier computed from the costs of changing the reused code, the costs of understanding how to integrate the code and the costs of reuse Decision making.

**44. Define operational costs.**

It consists of the costs of operating the system once it has been installed. EAC = AC + ETC. Current variances are seen as a typical and the Fixed Costs, Costs do not change.

**45. What is meant by cost flow forecasting? (Apr 2014)**

As important as estimating the overall costs and benefits of a project is the forecasting of the cash flow that will take place and their timing. A cash flow forecast will indicate when expenditure and income will take place.

**46. What are the cost-benefit evaluation techniques?**

- Net profit- net profit and discounted cash flow automatically
- Payback period- projects will provide a true return-on-investment while meeting an acceptable Return of investment- successfully complete projects and receive a return on investment.
- Net present value- Successful Projects Fortunately for project managers
- Internal rate of return- delegation of general management authority to the Project Leader

**47. Give the formula of Net Present Value (Nov/Dec2011)**

$C_n$= net cash flow for period $NPV = \sum_{n=0}^{N} \frac{C_n}{(1+r)^n} = 0$

r = required rate of return

**48. Give the formula of payback period.**

$$\text{Payback Period} = \frac{\text{Investment}}{\text{Annual Cash Savings}}$$

**Significance**

Creating a project charter to formally initiate projects

**49. Define Decision tree. (May/Jun2013)**

Decision tree provide tools for evaluating expected outcomes and choosing between alternate strategies.

**Advantages**

Assistance in upgrading, designing and developing a software.

**50. What is IRR ? How is it calculated?(Nov/Dec2011)(May/Jun2012)**

The internal rate of return on an investment or project is the "annualized effective compounded return rate" orrate of return that makes

the net present value(NPV as NET*1/(1+IRR)^year) of all cash flows (both positive and negative) from a particular investment equal to zero . It can also be defined as the discount rate at which the present value of all future cash flow is equal to the initial investment or in other words the rate at which an investment breaks even.

Given a collection of pairs (time, cash flow) involved in a project, the internal rate of return follows from the net present value as a function of the rate of return. A rate of return for which this function is zero is an internal rate of return.

Given the (period, cash flow) pairs ( periods N, and the net present value N

$$\mathrm{NPV} = \sum_{n=0}^{N} \frac{C_n}{(1+r)^n} = 0$$

n,$C_n$ where n is a positive integer, the total number of PV, the internal rate of return is given by T in:

The period is usually given in years, but the calculation may be made simpler if T is calculated using the period in which the majority of the problem is defined (e.g., using months if most of the cash flows occur at monthly intervals) and converted to a yearly period thereafter.

Any fixed time can be used in place of the present (e.g., the end of one interval of an annuity); the value obtained is zero if and only if the NPV is zero.

In the case that the cash flows are random variables, such as in the case of a life annuity, the expected values are put into the above formula. Often, the value of T cannot be found analytically. In this case, numerical methods or graphical methods must be used.

**51. What is the significance of a "project risk matrix"? Give an example (May/Jun2012)**

- Identify the risk and give priority.
- Could draw up draw a project risk matrix for each project to assess risks
- Project risk matrix used to identify and rank the risk of the project

| Risk | Importance | Likeihood |
|---|---|---|
| Sofrware never completed or delivered | H | – |
| Project cancelled after design stage | H | – |
| Sofware delivered late | M | M |
| Development budget exceeded ≤ 20% | L | M |
| Development budget exceeded >20% | M | L |
| Maintenance cost higher than estimated | L | L |
| Response time targets not met | L | H |

**52. When Net present value is calculated for a project.(Nov/Dec2012)**

The net present value(NPV) or net present worth(NPW)is defined as the sum of the present value s(PVs) of incoming and outgoing cash flows over a period of time. Incoming and outgoing cash flows can also be described as benefit and cost cash flows, respectively.

## PART B

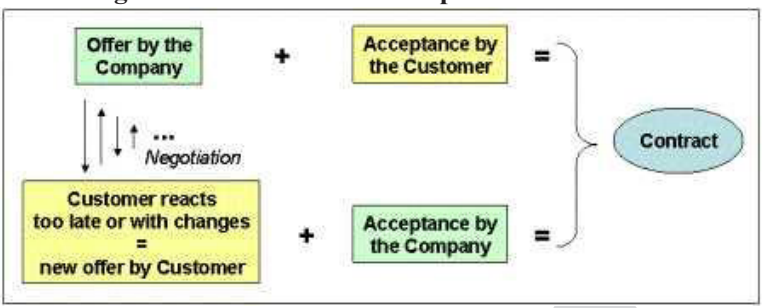
**1. Explain the difference between software projects and other projects in detail.**

- Invisibility- Software can't be rep-resented with geometric models,
- Complexity- The proposed model is based on the widely known and accepted
- Confirmity- The controlling document for a software
- Flexibility- project management performance

**2. Explain contract management and technical project management.**

A contract is any agreement between two or more parties where one party agrees to provide certain deliveries or services, and the other party agrees to pay for those deliveries or services.

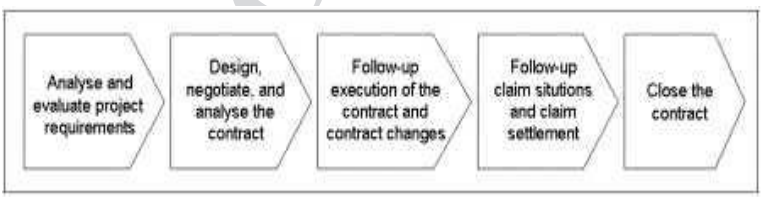
**How do we get a contract between two parties?**




It just takes an offer by a company and the simple acceptance of that offer by the customer, and we have a contract. Typically, we will see some negotiation going on between the two parties before one of them accepts the last offer of the other party. However, since it is so easy to end up in a legally binding contract situation, the first step, generally the offer by the company has to be prepared very carefully.

Even for smaller projects we usually need more than two parties to contribute. So, another important aspect is, how many different parties we need and how the contractual structure should look like. In sub-section Project Contract Structure, we summarize the basic structures to choose.

Contract management is a continuous process, starting with analysis and evaluation of the customer's inquiry, and carrying on until contract closure, upon fulfillment of all contractual obligations.




This process overview indicates that contract management activities seem to belong to the responsibilities of the project manager and the whole project team. In fact, they do; however, in larger projects where we have large contracts it is best practice to involve a full-time contract manager who brings in his professional experience, takes responsibility for that process, and ensures the contribution of all team members.

Contract preparation comprises analysis and evaluation of the other parties' requirements, a clear statement of our own requirements, and

negotiation in order to reach agreement between the involved parties. After signing the contract, upon handover, the implementation team needs to analyze the contract in order to ensure that they understand what has been signed and needs to be implemented. When preparing and signing a contract in definition and planning phase, we anticipate how we want to implement the required project results, and fix this anticipation in our planning documents. This means that all our project planning is based on assumptions on how the project environment will develop over implementation and closure phase. As a simple matter of life, these assumptions can turn out to be wrong: certain conditions can change, or certain events can happen so that changes or deviations of the plans and of the contract become necessary. Thus, it would be helpful to prepare the project plans and the contract in a way so that those necessary changes can be implemented with mutual agreement of all involved parties.

**Technical Project Planning Methodologies**

- Identify different approaches to planning technical projects: rolling wave
- Planning…stage gate process…critical chain project management.
- Common construction project life cycle.
- Common pharmaceutical project life cycle.

**3. Explain activities covered by the software project management. (Nov/Dec2011)(Nov/Dec2012) (May/Jun2013)(Apr 2014)**

- Project identification
- Project definition
- Project planning
- Project organization
- Resource allocation
- Tracking, monitoring ,control
- Project termination

**4. What is management? Explain the problems with software projects. (Nov/Dec2011)(Nov/Dec2012)**

Management can be defined as all activities and tasks undertaken by one or more Persons for the purpose of planning and controlling the activities of others in order to achieve objectives or complete an activity that could not be achieved by others acting independently.

The activities of Management control are following:

- **Planning –**Deciding what is to be done.
- **Organizing –** making arrangements.
- **Staffing-**selecting the right people at right time for the job.
- **Directing-**giving instructions.
- **Monitoring –** checking on progress.
- **Controlling-** taking action to remedy hold-ups.
- **Innovating-**coming up with new solutions.
- **Representing –** liaising with clients, users, developers, suppliers.

**Problems with Software Project from Manager's point of view:**

- Poor estimates and plans.
- Lack of quality standards and measures.
- Lack of techniques to make progress visible.
- Lack of guidance about organizational decisions.
- Poor role definition.
- Incorrect success criteria.

**Problems with software project from student's point of view**

- Inadequate specification of work.
- Lack of knowledge of application area.
- Lack of standards.
- Narrow scope of technical expertise.

**5. Explain stakeholders and business case.**

The importance of stakeholder management is to support an organization in achieving its strategic objectives by interpreting and influencing both the external and internal environments and by creating positive relationships with stakeholders through the appropriate management of their expectations and agreed objectives.

Stakeholder Management is a process and control that must be planned and guided by underlying Principles.

Stakeholder Management, within business or projects, prepares a strategy utilizing information (or intelligence) gathered during the following common processes:

**Stakeholder Identification** - Interested parties either internal or external to organization/project.

**Stakeholder Analysis** - Recognizes and acknowledges stakeholder's needs concerns, wants, authority, common relationships, and inter faces and align this information within the Stakeholder Matrix.

**Stakeholder Matrix** - Positioning stakeholders according to the level of influence, impact or enhancement they may provide to the business or its projects.

**Stakeholder Engagement** - Different to Stakeholder Management in that the engagement does not seek to develop the project/business requirements, solution or problem creation, or establishing roles and responsibilities.

It is primarily focused at getting to know and understand each other, at the Executive level.

Engagement is the opportunity to discuss and agree expectations of communication and, primarily, agree a set of Values and Principles that all stakeholders will abide by.

**Communicating Information** - Expectations are established and agreed for the manner in which communications are managed between stakeholders – who receives communications, when, how and to what level of detail.

Protocols may be established including security and confidentiality classifications.

**Stakeholder Agreements:** A collection of agreed Decisions between stakeholders.

This may be the lexicon of an organization or project, or the Values of an initiative, the objectives, or the model of the organization, etc.

These should be signed by key stakeholder representatives.

Contemporary or modern business and project practice favours transparent, honest and open stakeholder management processes.

**6. Explain Management Control in detail.**

- Briefly explain what management is.
- Explains the process of setting objectives for a system and then monitoring the systems to see what true performance is.
- A change is proposed by any one evaluating the software.

**7. Explain the step-wise project planning in detail.**
**(Nov/Dec 2011)(May/Jun 2012)(Apr 2014)**

- Identify project scope and objectives.
- Identify project infrastructure.
- Analyze project characteristics.
- Identify project products and activities.
- Estimate effort for each activity.
- Identify activity risks.
- Allocate resources.
- Review / publicize plan
- Execute lower levels of planning.
- Execute higher levels of planning.

**8. How to analyze the project characteristics? (May/Jun 2012)**

**Explain the following characteristics:**

- A positive relationship with an active, intelligent client.
- Strong project management.

- Clear requirements, well managed.
- Ruthless change management.
- Pervasive process focus.
- Effective controls and communication.
- Technical leadership and excellence.

**9. Explain the steps involved in to identify activity risks.**

The first thing is to understand the risk management is that it's an on-going activity.

It's not about identifying risks upfront and then forging ahead regardless.

It's too easy to forget the risks once the project is started and fail to recognize and raise new risks when the project is underway.

The key steps to risk management are summarized below.

- Risk Assessment.
- Risk Reduction.
- Risk Minimization.
- Risk Containment.
- Risk Monitoring.
- Risk Reporting.
- Risk Evaluation.

A key part to project management is a common language.

The diagram below shows the key steps of risk management in the overall context of analysis and control.

As outlined in the introduction, there are two key outcomes for risk management, action and awareness.

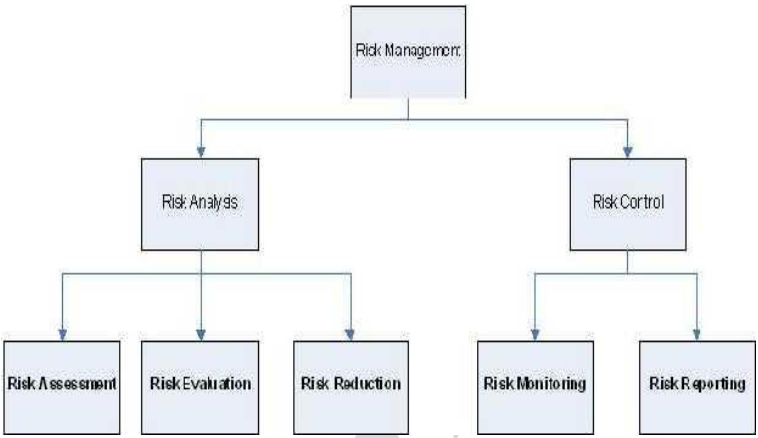
Through risk control we manage action, through risk analysis we manage awareness.

The diagram below shows the breakdown of risk management. On the left we have risk analysis which comprises of reviewing risks, evaluating risks and reducing risks.

This is the "action" side of risk management.

On the right we have risk control which comprises of risk monitoring and reporting.

This is the awareness and prevention side of risk reporting.




**10. Explain the steps in project planning with case studies example. (Nov/Dec 2012)**

**Define**

- Project goal.
- Project deliverables.
- Project schedule.
- Supporting plans.

**11. Explain the various SDLC activities as outlined by ISO 12207 with a neat diagram. (May/Jun 2012)**

ISO/IEC 12207 Systems and software engineering — Software life cycle processes is an International Standard for software life cycle processes.

It aims to be the standard that defines all the tasks required for developing and maintaining software.

The standard has the main objective of supplying a common structure so that the buyers, suppliers, developers, maintainers, operators, managers and technicians involved with the software development use a common language.

This common language is established in the form of well-defined processes.

The structure of the standard was intended to be conceived in a flexible, modular way so as to be adaptable to the necessities of whoever uses it.

The standard is based on two basic principles: modularity and responsibility.

Modularity means processes with minimum coupling and maximum cohesion.

Responsibility means to establish a responsibility for each process, facilitating the application of the standard in projects where many people can be legally involved.

The set of processes, activities and tasks can be adapted according to the software project.

These processes are classified in three types: basic, for support and organizational.

The support and organizational processes must exist independently of the organization and the project being executed. The basic processes are instantiated according to the situation.

**Activities:**

- Acquisition
- Supply
- Development
- Operation
- Maintenance
- Destruction

**12. What are the steps involved in project evolution?**

**Project evaluation** is a systematic method for collecting, analyzing, and using information to answer questions about projects, policies and programs, particularly about their effectiveness and efficiency **Develop program logic and review needs**

- Develop the evaluation brief
- Commission the evaluation project

- Manage development of the evaluation design
- Manage development of the evaluation work plan
- Manage implementation of the work plan, including production of report(s)
- Disseminate report and support use of the evaluation

**13. Write in detail for project management with strategic assessment. (Nov/Dec 2011)**

Strategic planning is defined as an organization's process of defining its strategy, or direction and making Decisions on allocating its resources to pursue this strategy.

- Briefly explain what it deals with?
- What do we do?
- For whom do we do it?
- How to we excel ?
- For successful strategic assessment, there should be a strategic plan which defines:
- Organization's objectives.
- Provides context for defining programme
- Provides context for defining programme goals.
- Provide context for accessing individual project.

**14. How to manage the allocation of resources within programmes with examples. (Apr 2014)**

In strategic planning, resource allocation is a plan for using available resources.

Example: Human resources, especially in the near term, to achieve goals for the future.

It is the process of allocating scarce resources among the various projects or business units.

There are a number of approaches to solving resource allocation problems e.g. resources can be allocated using a manual approach, an algorithmic approach or a combination of both.

There may be contingency mechanisms such as a priority ranking of items excluded from the plan, showing which items to fund if more resources should become available and a priority ranking of some items included in the plan, showing which items should be sacrificed if total funding must be reduced.

Resource allocation may be decided by using computer programs applied to a specific domain to automatically and dynamically distribute resources to applicants.

This is especially common in electronic devices dedicated to routing and communication.

For example, channel allocation in wireless communication may be decided by a base transceiver station using an appropriate algorithm.

**15. What re the steps involves in creating a programme?**

In large organization, programme management is taken care by programme director and programme executive, rather than, project manager, who will be responsible for the strategic assessment of project.

Any potential software system will form part of the user organization's overall information system and must be evaluated within the context of existing information system and the organization's information strategy.

If a well – defined information system does not exist then the system development and the assessment of project proposals will be based on a more **"piece meal approach".**

Piece meal approach is one in which each project being individually early in its life cycle.

Typical issues and questions to be considered during strategic assessment

**Issue – 1: objectives:**

- How will the proposed system contribute to the organization's stated objectives?
- How, for example, might it contribute to an increase in market share?

**Issue – 2: is plan**

- How does the proposed system fit in to the IS plan?

- Which existing system (s) will it replace/interface with? How will it interact with systems proposed for the later development?

**Issue – 3: organization structure:**

- What effect will the new system have on the existing departmental and organization structure?
- For example, a new sales order processing system overlap existing sales and stock control functions?

**Issue – 4: MIS:**

- What information will the system provide and at what levels in the organization?
- In what ways will it complement or enhance existing management information system?

**Issue – 5: personnel:**

- In what way will the system proposed system affect manning levels and the existing employee skill base?
- What are the implications for the organization's overall policy on staff development.

**Issue – 6: image:**

- What, if any, will be the effect on customer's attitudes towards the organization?
- Will the adoption of, say, automated system conflict with the objectives of providing a friendly service?

**16. Explain cost-benefit evaluation techniques.**

**(Nov/Dec 2011)(May/Jun 2013)(Apr 2014)**

**a)** It is one of the important and common way of carrying **"economic assessment"** of a proposed information system.

**b)** This is done by comparing the expected costs of development and operation of the system with its benefits.

**c)** So it takes an account:

i. Expected cost of development of system
ii. Expected cost of operation of system

iii. Benefits obtained

d)Assessment is based on:

i) Whether the estimated costs are executed by the estimated income.

ii) And by other benefits

e) For achieving benefit where there is scarce resources, projects will be prioritized and resource are allocated effectively.

**f)** The standard way of evaluating economic benefits of any project is done by "cost benefit analysis"

**Cost benefit analysis comprises of two steps:**

- **Step-1:** Identifying and estimating all of the costs and benefits of carrying out the project.
- **Step-2:** Expressing these costs and benefits in common units.

**Step-1:**

It includes

- Development cost of system.
- Operating cost of system.
- Benefits obtained by system.

When new system is developed by the proposed system, then new system should reflect the above three as same as proposed system.

**Example:** Sales order processing system which gives benefit due to use of new system.

**Step-2:**

Calculates net benefit.

**Net benefit = total benefit = total cost.**

(cost should be expressed in monetary terms).

Three types of cost

- **Development costs:** includes salary and other employment cost of staff involved.
- **Setup costs :** includes the cost of implementation of system such as hardware, and also file conversion, recruitment and staff training.
- **Operational cost :** cost require to operate system, after it is installed.

**17. Explain Decision trees with examples.**

Decision tree provide tools for evaluating expected outcomes and choosing between alternate strategies.

**Advantages**

Assistance in upgrading, designing and developing a software.

- Identify over risky projects
- Choose best from risk
- Take suitable course of action
- Decision tree of analysis risks helps us to
- Extend the existing system
- increase sales
- improve the management information
- Replace the existing system
- Not replacing system leads in loss
- Replace it immediately will be expensive.

**18. Explain Risk Evaluation (May/June 2012)**

**Evaluation**

Risk evaluation is meant to decide whether to proceed with the project or not, and whether the project is meeting its objectives.

**Risk Occurs:**

- When the project exceed its original specification.
- Deviations from achieving it objectives and so on.
- Risk Identification and ranking.
- Risk and Net Present Value

For riskier projects could use higher discount rates

**Example:** Can add 2% for a Safe project or 5 % for a fairly risky one.

- Cost benefit Analysis.
- Risk profile analysis.
- Decision trees.

**19. What is meant by cash flow forecasting? Explain with example. (May/Jun 2012)(Nov/Dec 2012)**

As important as estimating the overall costs and benefits of a project is the forecasting of the cash flow that will take place and their timing.

A cash flow forecast will indicate when expenditure and income will take place.

- It estimates overall cost and benefits of a product with respect to time.
- Negative cash flow during development stage.
- Positive cash flow during operating life.
- During development stage
- Staff wages
- Borrowing money from bank
- Paying interest to bank
- Payment of salaries
- Amount spent for installation, buying H/W and S/W.
- Income is expected by 2 ways.
- Payment on completion
- Stage payment

**20. Explain the "internal rate of return" method for measuring the profitability of a project. Also mention its advantage over the NPV method. (May/Jun 2012).**

The internal rate of return on an investment or project is the "annualized effective compounded return rate" or rate of return that makes the net present value (NPV)

**NPV = NET*1/(1+IRR)^year)**

of all cash flows (both positive and negative) from a particular investment equal to zero.

It can also be defined as the discount rate at which the present value of all future cash flow is equal to the initial investment or in other words the rate at which an investment breaks even.

Given a collection of pairs (time, cash flow) involved in a project, the internal rate of return follows from the net present value as a function of the rate of return.

A rate of return for which this function is zero is an internal rate of return.

Given the (period, cash flow) pairs $(n,C_n)$ where n is a positive integer, the total number of periods N, and the net present value NPV, the internal rate of return is given by 'i' in:

$$NPV = \sum_{n=0}^{N} \frac{C_n}{(1+r)^n} = 0$$

The period is usually given in years, but the calculation may be made simpler if 'i' is calculated using the period in which the majority of the problem is defined (e.g., using months if most of the cash flows occur at monthly intervals) and converted to a yearly period thereafter.

Any fixed time can be used in place of the present (e.g., the end of one interval of an annuity); the value obtained is zero if and only if the NPV is zero.

In the case that the cash flows are random variables, such as in the case of a life annuity, the expected values are put into the above formula.

Often, the value of 'i' cannot be found analytically.

In this case, numerical methods or graphical methods must be used.

# UNIT - 2

## PROJECT LIFE CYCLE AND EFFORT ESTIMATION

### PART – A

**1. What are the characteristics of Software Project Management?**

- Software Project Management is engineered or developed.
- It is not manufactured in the classical sense.
- Software Project Management doesn't wear out.
- Although the industry is moving toward component based assembly, most software Project continues to be custom built.

**2. Write down the generic process framework that is applicable to any software project. (NOV/DEC 2010)(APR/MAY 2015)**

The following generic process framework is applicable to vast majority of software projects:

**Communication:** This framework activity involves heavy communication and collaboration with the customer and encompasses requirements gathering and other related activities.

**Planning:** This activity establishes a plan for the software engineering work that follows. It describes the technical tasks to be conducted, the risks that are likely, the resources that will be required, the work products to be produced, and a work schedule.

**Modeling:** The activity encompasses the creation of models that allow the developer and the customer to better understand software requirements and the design that will achieve those requirements.

**Construction:** This activity combines code generation and the testing that is required to uncover errors in the code.

**Deployment:** The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation.

**3. List the goals of Software Project Management? (APR/MAY 2011)**

Software Project Management is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

The goals Software Project Management are:

- Software production which consists of developed programs and associated documentation.
- The software product should have the essential product attributes maintainability, dependability, efficiency and acceptability.
- It should also include suggestions for the process to be followed, the notations to be used, system models to be developed and rules governing these models and design guidelines.

**4. What are the difference levels of Software CMMI?**

- Level 0 – Incomplete.
- Level 1 – Performed.
- Level 2 – Managed.
- Level 3 – Defined.
- Level 4 – Quantitatively Managed.
- Level 5 – Optimized.

**5. Give two reasons why system engineers must understand the environment of a system. (MAY/JUNE 2012)**

The reason why system engineers must understand the environment of the system:

- **Limited scope for rework during system development:** Once some system engineering decisions have been made, they are very expensive to change.
- Reworking the system design to solve these problems is rarely possible.
- **Interdisciplinary involvement:** Many engineering disciplines are involved in system engineering.
- There is a lot of scope for miss-understanding because different engineers use different terminology and conventions.

**6. What are the two types of software products? (MAY/JUNE 2012)**

The two fundamental types of software product are

- **Generic products:** These are standalone systems developed by organizations and sold on open market to any customer who is able to buy them.
- **Customized products:** These are systems which are commissioned by a particular customer. A software contractor develops the software especially for that customer.

**7. What is Software Project Management?**

**(NOV/DEC 2013)(NOV/DEC 2014)**

Software Project Management is a discipline in which theories, methods and tools are applied to develop professional software.

(Or)

Software Project Management is the systematic approach to develop and maintain a software product in a cost effective and efficient way.

**8. What is RAD?**

- Rapid Application Development (RAD) is an incremental software development process.
- When tools are interpreted, the information created by one tool can be used by another.
- A system for the support of software development called computer aided software engineering is established.

**9. List out Evolutionary Software Process Model.**

- Incremental model
- Spiral model
- WIN-WIN spiral model
- Concurrent development model
- Object oriented model
- Embedded model
- Synchronize and stabilize model

**10. What are the difference between product and process?**

| PROCESS | PRODUCT |
|---|---|
| It is a frame work which has a set of rules to be followed in key processing areas (KPA), rules for framing task sets, setting a milestone for it and applying s/w quality assurance points. | It is the initial shipment outcome of the process. |
| It is used to obtain quality product. | Various process paradigms/ models are used to build a quality product. |

**11. What is CPF?**

A Common Process Framework (CPF) is established by defining a small number of framework activities that are applicable to all software projects, regardless of their size or complexity.

**12. What are the advantages and disadvantages of Waterfall Model?**

**Advantages:**

- It provides a template into which methods for analysis, design, and other phases can be placed.
- It provides for baseline management.
- It is better than any haphazard approach to software development.

**Disadvantages:**

- It lacks the perception for a reverse engineering on how to engineer an existing legacy system.
- The client has to wait until the installation and checkout phase to see how a system works.
- Thus a complex system requires considerable time and effort.
- There is no rapid prototyping and incremental development.
- Real time software cannot follow this model.
- Customer satisfaction is not full filled.

**13. What are the advantages disadvantages of Incremental Model ?**

**Advantages:**

- The software development activities are repeated each time there is a new release of software.
- It provides a platform for evaluation by the user.
- It can be planned to manage technical risks.
- It enables partial functionality to be delivered to end users without ordinary delay.

**Disadvantage:**

- It makes the unrealistic assumptions that system as well as software requirements remain stable which is not true.

**14. What are the advantages and disadvantages of Spiral Model?**

**Advantages:**

- It provides the potential for rapid development for incremental versions of the software.
- It can be applied throughout the life of the computer software.
- It allows the developer to apply the prototyping approach at any stage.
- It demands a direct consideration of technical risks at all stages.

**Disadvantages:**

- It may be difficult to convince the customers at times especially in contrast solutions.
- It demands considerable risks assessment expertise and relies on them for success.
- It takes time for determining the efficiency and thus the model cannot be used as widely as others.

**15. What are the advantages and disadvantage of WIN-WIN SPIRAL MODEL?**

**Advantages:**

- It has a provision for system stakeholders to negotiate mutually satisfactory specifications.

- Customer satisfaction is fulfilled.
- It overcame the problem of lack of anchor points to correlate the completion of the spiral cycles and organization major milestones.

**Disadvantage:**

- The model does not specifically address the issues of how developers specify, design and test the conceptual construct software.

**16. What are the advantages and disadvantage of Object Oriented Model?**

**Advantages:**

- Object oriented concepts like encapsulation can be improvised in this model.
- It simplifies software development because it hides complexity.
- Reusability is enhanced.

**Disadvantage:**

- In safety critical conditions they require a design by contract in the construction of reliability software.

**17. Define Computer based system and specify its components.**

- A set of arrangement of elements that are organized to accomplish some predefined goals by processing information.
- Its components are software, hardware, people, database, documentation and procedure.

**18. Give the restraining factors that are to be considered to construct a system model.**

**Assumptions** that reduce the number of possible permutations and variations thus enabling the model to reflect the problem in a reasonable manner.

**Simplifications** that enable the model to be created in a timely manner.

**Limitations** that help to bound the system.

**Constraints** that will guide the manner in which the model is created.

**Approach** taken when the model is implemented.

**Preferences** that indicate the preferred architecture for all data, functions, and technology.

**19. What are the advantages and disadvantages of Prototyping Model?**

**Advantages:**

- It produces the products quickly and thus saves the time.
- Solves the waiting problem in waterfall model.
- It minimizes the cost and product failure.
- It is possible for the developers and client to check the function of preliminary implementations of system models before committing to a final system.
- It obtains feedback from clients and changes in system concept.

**Disadvantages:**

- It ignores quality, reliability maintainability and safety requirements.
- Customer satisfaction is not achieved.

**20. What does a System Engineering Model accomplish?**

- Define processes that serve needs of view.
- Represent behavior of process and assumption.
- Explicitly define Exogenous and Endogenous Input.
- Represent all Linkages that enable engineer to better understand view.

**21. What are the different considerations that have to be followed for project estimation?**

To achieve reliable cost and effort estimation, the following options are considered,

- Delay estimation until late in the project.

- Base estimates on similar projects that have already been completed.
- Use relatively simple decomposition techniques to generate project cost and effort estimates.
- Use one or more empirical models for software cost and effort estimation.

**22. Define process-based estimation. State the advantages and disadvantages in LOC based Cost Estimation. (APR/MAY 2015)**

The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated.

**Advantages of LOC:**

- Simple to measure.
- An intuitive metric for measuring the size of software due to the fact that it can be seen and the effect of it can be visualized.

**Disadvantages of LOC:**

- Difficult to measure LOC in the early stages of a new product.
- Source instructions vary with coding languages, design methods and with programmer's ability.
- No industry standard for measuring LOC.
- LOC cannot be used for normalizing if platforms and languages are different.
- The only way to predict LOC for a new app to be developed is through analogy based on similar software application.
- Programmers may be rewarded for writing more LOC based on a misconception of higher management by thinking that more the LOC, means more the productivity of the programmer.

**23. What are the different hierarchies of estimation models?**

- Application Composition Models
- Early Design Stage Models.
- Post-Architecture Stage Models.

**24. What are the guidelines for Project Scheduling? (APR/MAY 2015)**

- Compartmentalization.
- Interdependency.
- Time allocation.
- Effort validation.

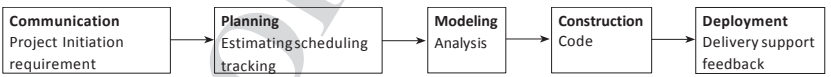
**25. What are the characteristics of Risks?**

- **Uncertainty** – the risk may or may not happen
- **Loss** – if the risk becomes reality, unwanted consequences or losses will occur

## PART – B

**1. The Linear Sequential Model (Water Fall Model) (NOV/DEC 2010 & 2013) (MAY/JUN 2014) (APR/MAY 2015)**

Sometimes called the classic life cycle or the waterfall model the linear sequential model suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing and support.




**Communication**: The basic requirement of the system must be understood by software engineer. The information domain function behavioral requirements of the system are understood. Communication and collaboration with the customer and encompasses requirement gathering

**Planning:** It establishes a plan for the software engineering work that follows. It describes the technical tasks to be conducted, the risks that are likely

1. The resources that we require
2. The work products to be produced
3. Work Schedule

**Modeling:** This activity involves with the creation of models that allow the developer and the customer to better understand software requirements and the design that will achieve those requirements

**Construction:** This activity combines code generation (either manual or automated) and the testing that is required to uncover errors in the code.

**Deployment:** The software is delivered to the customer who evaluates the delivered product and provides feedback based on the evaluation

**Advantages:**

- Easy to use and implement.
- Understood by non-technical persons also.
- Proper planning due to proper definitions of the problem.

**Disadvantages:**

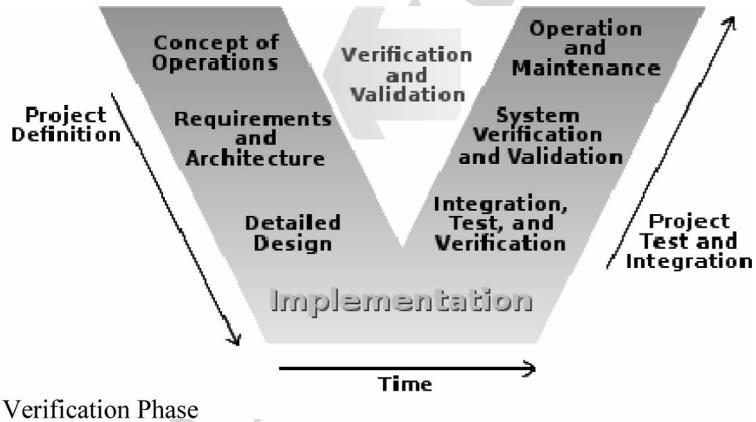
- Difficult to follow the sequential low in software development process.
- If some changes are made at some phases then it may cause some confusion.
- Changes can cause confusion as the project team proceeds.
- The Requirement analysis is done initially and sometimes it is not possible to state all the requirements explicitly in the beginning. This causes difficulty in the project.
- The customer can see the working model of the project only at the end.
- After reviewing of the working model, if the customer gets dissatisfied then it causes serious problems
- Linear Nature of waterfalls model induces blocking stages, because certain tasks may be dependent on some previous tasks.
- Hence it is necessary to accomplish all the dependent tasks first.
- It may cause long waiting time.
- Not suitable for complex, large and new projects.

**2. Explain V Model** **(NOV/DEC 2010 & 2013)**

**(MAY/JUNE 2014) (APRIL/MAY 2015)**

The V-model represents a software development process (also applicable to hardware development) which may be considered an extension of the waterfall model. Instead of moving down in a linear way, the process steps are bent upwards after the coding phase, to form the typical V shape.

The V-Model demonstrates the relationships between each phase of the development life cycle and its associated phase of testing. The horizontal and vertical axes represents time or project completeness (left-to-right) and level of abstraction (coarsest-grain abstraction uppermost), respectively.




Verification Phase

**Requirements analysis**

In the Requirements analysis phase, the requirements of the proposed system are collected by analyzing the needs of the user(s). This phase is concerned about establishing what the ideal system has to perform.

However it does not determine how the software will be designed or built. Usually, the users are interviewed and a document called the user requirements document is generated.

The user requirements document will typically describe the system's functional, interface, performance, data, security, etc., requirements as expected by the user. It is used by business analysts to communicate their understanding of the system to the users.

The users carefully review this document as this document would serve as the guideline for the system designers in the system design phase. The user acceptance tests are designed in this phase. See also functional requirements. This is parallel processing.

There are different methods for gathering requirements of both soft and hard methodologies including; interviews, questionnaires, document analysis, observation, throw-away prototypes, use cases and status and dynamic views with users.

**System Design**

Systems design is the phase where system engineers analyze and understand the business of the proposed system by studying the user requirements document.

They figure out possibilities and techniques by which the user requirements can be implemented. If any of the requirements are not feasible, the user is informed of the issue. A resolution is found and the user requirement document is edited accordingly.

The software specification document which serves as a blueprint for the development phase is generated. This document contains the general system organization, menu structures, data structures etc.

It may also hold example business scenarios, sample windows, reports for the better understanding. Other technical documentation like entity diagrams, data dictionary will also be produced in this phase. The documents for system testing are prepared in this phase.

**Architecture Design**

The phase of the design of computer architecture and software architecture can also be referred to as high-level design.

The baseline in selecting the architecture is that it should realize all which typically consists of the list of modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology details etc. The integration testing design is carried out in the particular phase.

**Module Design**

The module design phase can also be referred to as low-level design. The designed system is broken up into smaller units or modules and

each of them is explained so that the programmer can start coding directly. The low level design document or program specifications will contain a detailed functional logic of the module, in pseudo code:

- Database tables, with all elements, including their type and size
- All interface details with complete API references
- All dependency issues
- Error message listings
- Complete input and outputs for a module. The unit test design is developed in this stage

**Unit Test**

In computer programming, unit testing is a method by which individual units of source code are tested to determine if they are it for use. A unit is the smallest testable part of an application.

In procedural programming a unit may be an individual function or procedure. Unit tests are created by programmers or occasionally by white box testers. The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage.

Static analysis tools are used to facilitate in this process, where variations of input data are passed to the function to test every possible case of execution.

**Integration Testing**

In integration testing the separate modules will be tested together to expose faults in the interfaces and in the interaction between integrated components. Testing is usually black box as the code is not directly checked for errors.

**System Testing**

System testing will compare the system specifications against the actual system. After the integration test is completed, the next test level is the system test.

System testing checks if the integrated product meets the specified requirements. Why is this still necessary after the component and integration tests? The reasons for this are as follows:

**Reasons for System Test**

In the lower test levels, the testing was done against technical specifications, i.e., from the technical perspective of the software producer.

The system test, though, looks at the system from the perspective of the customer and the future user.

The testers validate whether the requirements are completely and appropriately met.

- **Example:** The customer (who has ordered and paid for the system) and the user (who uses the system) can be different groups of people or organizations with their own specific interests and requirements of the system.
- Many functions and system characteristics result from the interaction of all system components, consequently, they are only visible on the level of the entire system and can only be observed and tested there.

**User Acceptance Test**

Acceptance testing is the phase of testing used to determine whether a system satisfies the requirements specified in the requirements analysis phase.

The acceptance test design is derived from the requirements document. The acceptance test phase is the phase used by the customer to determine whether to accept the system or not.

**Acceptance testing helps**

- To determine whether a system satisfies its acceptance criteria or not.
- To enable the customer to determine whether to accept the system or not.
- To test the software in the "real world" by the intended audience.

**Purpose of acceptance testing:**

To verify the system or changes according to the original needs.

**3. Explain the Spiral Model**
**(NOV/DEC 2010 & 2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

The spiral model is an evolutionary software process that couples the iterative nature of prototyping with the controlled and systematic aspects of the linear sequential model.

Software is developed in a series of incremental releases. During early iterations, the incremental release might be a paper model or prototype. During later iterations, increasingly more complete versions of the engineered system are produced.

**Spiral Model contains six task regions:**

**Customer communication**-tasks required to establish effective communication between develop and customer.

**Planning:** tasks required to define resources, timelines and other project related information.

**Risk analysis**-tasks required to asses both technical and management risks.

**Engineering:** tasks required to assess both technical and management risks.

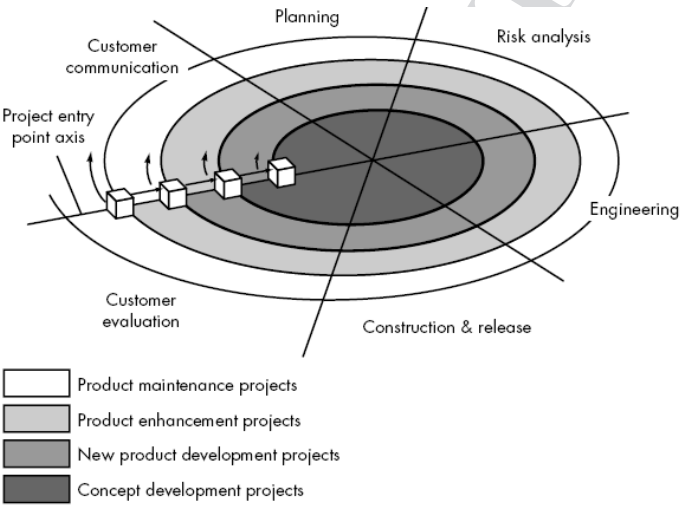
**Construction and release**-tasks required to construct, test, install and provide user support (e.g documentation and training).

**Customer evaluation:** tasks required to obtain during the engineering stage and implemented during the installation stage.

- Each of the regions is populated by a set of work tasks, called a task set for small projects.
- The number of work tasks and their formality is low for larger, more critical projects.
- Each task region contains more work tasks that are defined to achieve a higher level of formality.
- As the process begins, the software engineering team moves around the spiral in a clockwise directions beginning at the center.
- The first circuit around the spiral might result in the development of a product specification.
- Subsequent passes around the spiral might be used to develop a prototype.

- Then progressively develops more sophisticated versions of the software.
- Each pass through the planning region results in adjustments to the project plan.
- Cost and schedule are adjusted based on feedback derived from customer evaluation.
- In addition, the project manager adjusts the planned number of iterations required to complete the software.

Unlike classical process models that end when software is delivered, the spiral model can be adapted to apply throughout the life of the computer software.




- An alternative view of the spiral model can be considered by examining the project entry point axis.
- Each cube placed along the axis can be used to represent the starting point for different types of projects.

i. Product maintenance projects

ii. Product enhancement projects

iii. New product development projects

iv. Concept development projects

- A "concept development project" starts at the core of the spiral and will continue until concept development is complete.
- If the concept is to be developed into an actual product, the process proceeds through the next cube (new product development project entry point) and a "new development project" is initiated.
- The new product will evolve through a number of iterations around the spiral, following the path that bounds the region that has somewhat lighter shading than the core.
- In essence, the spiral, when characterized in this way, remains operative until the software is retired.
- There are times when the process is dormant, but whenever a change is initiated, the process starts at the appropriate entry point (e.g., product enhancement).
- The spiral model is a realistic approach to the development of large-scale systems and software.
- Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level.
- The spiral model demands a direct consideration of technical risks at all stages of the project and, if properly applied, should reduce risks before they become problematic.

**Advantages:**

- User is able to see and perceive the project
- Risk analyzer which is used to resolves higher priority errors
- Project is very much defined
- Reusability of the software

**Disadvantages:**

- It is only suitable for larger size projects
- Model is complex to use
- Management skill is necessary so as to analyze the risk factor.

**4. Discuss RAD Model**

**(NOV/DEC 2010 & 2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

Rapid application development (RAD) is an incremental software development process model that emphasizes an extremely short development cycle.

The RAD model is a "high-speed" adaptation of the linear sequential model. If requirements are well understood and project scope is constrained, the RAD process enables a development team to create a **"fully functional system"** within very short time periods **(e.g. 60 to 90 days).**

**Phases: Communication**

To understand the business problem and the information characteristics that the software must accommodate.

The information low defined as part of the business modeling phase is reined into a set of data objects that are needed to support the business.

The characteristics (called attributes) of each object are identified and the relationships between these objects defined.

**Planning**

It is essential because multiple software teams work in parallel on different system functions

**Modeling:** the data objects defined in the data modeling phase are transformed to achieve the information low necessary to implement a business function.

**Processing** descriptions are created for adding, modifying, deleting or retrieving a data object.

Modeling encompasses three major phases

- Business Modeling
- Data Modeling
- Process Modeling

**Construction**

It uses the fourth generation techniques.

In all cases, automated tools are used to facilitate construction of the software.

Since the RAD process emphasizes reuse, many of the program components have already been tested .this reduces overall testing time.

However, new components must be tested and all interfaces must be fully exercised.

**Deployment**

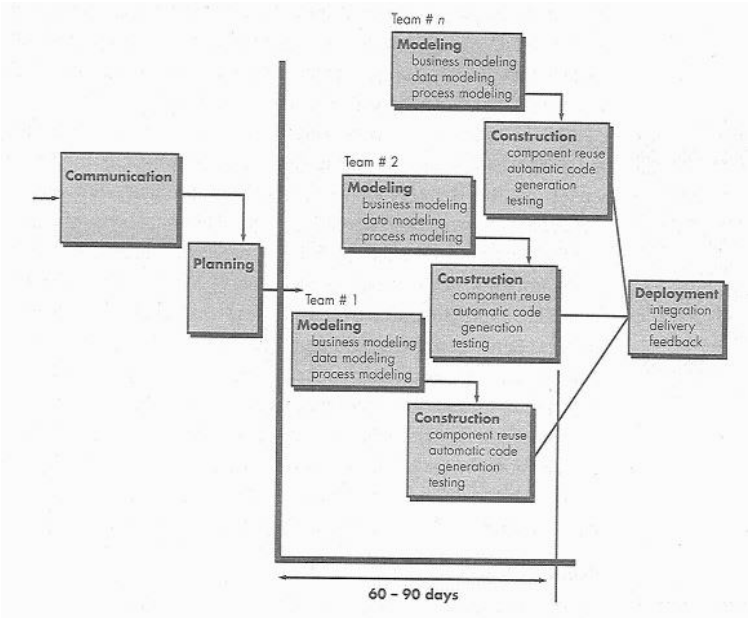
It establishes a basis for subsequent iterations if required.

**Advantages:**

**a)** Each major function to be completed in less than 3 months.
**b)** Each major function can be addressed by a separate RAD team.
**c)** Then integrated to form a whole.

**Disadvantages:**

a. For large but scalable projects.
b. RAD requires sufficient human resources to create the right number of RAD teams.
c. Developers and customers who are committed to the rapid-ire activities necessary to get a system complete in a much abbreviated time frame.
d. If commitment is lacking from either constituency. RAD projects will fail.
e. Not all types of applications are appropriate for RAD.
f. If a system cannot be properly modularized, building the components necessary RAD will be problematic.
g. RAD is not appropriate when technical risks are high.



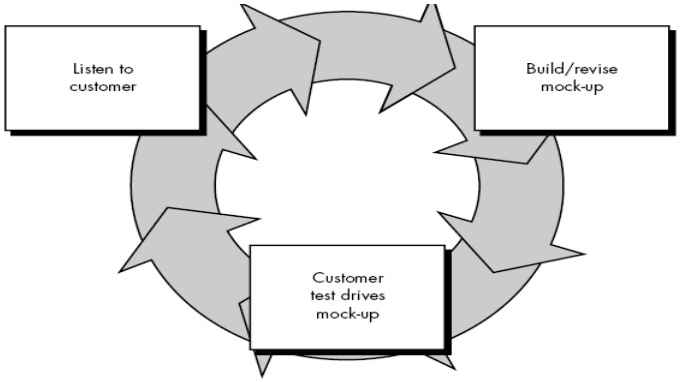

**5. How to develop Prototyping Model**
**(NOV/DEC2010 & 2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

A customer defines a set of general objectives for software, but does not identify detailed input, processing or output requirements.

In other cases the developer may be unsure of the deficiency of an algorithm, the adaptability of an OS or the form that human machine interaction should take. In these cases prototyping paradigm may offer best approach. The prototyping paradigm begins with requirements gathering.

Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known and outline areas where further definitions mandatory.

A **"quick design"** then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g.. input approaches and output formats).




The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and to collect the requirements for the software to be developed.

Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

The prototype is built the developer attempts to use existing program fragments or applies tools (e.g. report generators, window managers that enable working programs to be generated quickly.

**Advantages:**

- Customer satisfaction
- Flexible
- Partial products can be viewed by the customer

**Disadvantages**

- No optimal solution
- Time consuming if algorithm is inefficient

**6. Describe Waterfall, Incremental and Iterative models based SLCS and compare. (NOV/DEC 2012)**

**Waterfall Model:**

Refer

**Iterative Models:**

They are iterative.

They are characterized in a manner that enables softwareengineers to develop increasingly more complete versions of the software.

- **Prototyping**
- **The spiral model ,**
- **The WIN-WIN spiral Model**
- **Concurrent Development Model**

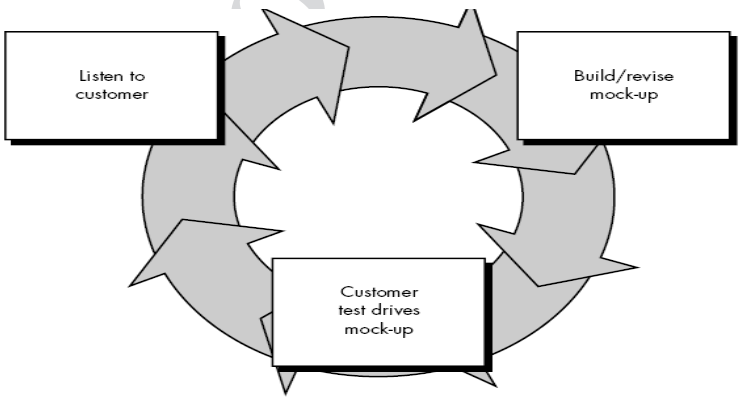
**Prototyping**

A customer defines a set of general objectives for software, but does not identify detailed input, processing or output requirements.

In other cases the developer may be unsure of the efficiency of an algorithm, the adaptability of an OS or the form that human machine interaction should take. In these cases prototyping paradigm may offer best approach.

The prototyping paradigm begins with requirements gathering. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known and outline areas where further definitions mandatory.

A **"quick design"** then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user (e.g.. input approaches and output formats).




The quick design leads to the construction of a prototype.The prototype is evaluated by the customer/user and used to define requirements for the software to be developed.

Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

The prototype is built the developer attempts to use existing program fragments or applies tools (e.g. report generators, window managers that enable working programs to be generated quickly.

**Advantages:**

- Customer satisfaction
- Flexible
- Partial products can be viewed by the customer

**Disadvantages**

- No optimal solution
- Time consuming if algorithm is inefficient
- Difficult to maintain

**The Spiral model**

Refer.

**7. Explain WIN-WIN spiral Model**
**(NOV/DEC 2010 &2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

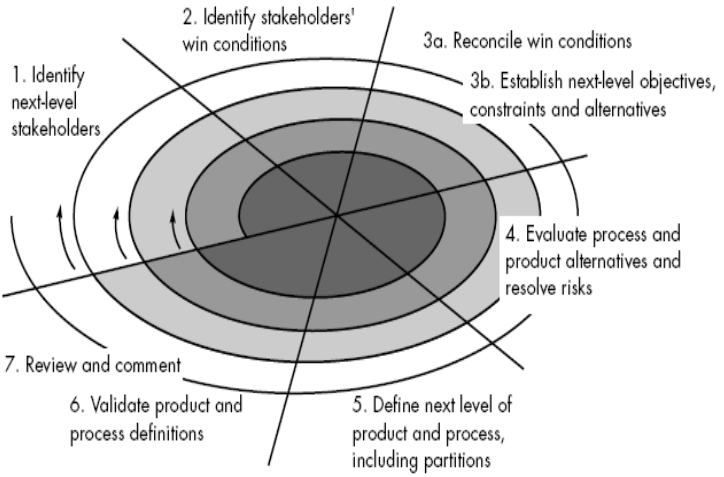
The objectivity of this activity is to elicit project requirements from the customer. Win-win result: the customer wins by getting the system of product that satisfies the majority of the customer needs and the developer wins by working to realistic and achievable budgets and deadlines.

A stakeholder is anyone in the organization that has a direct business interest in the system.The product to be built and will be rewarded for a successful outcome or criticized if the effort fails.

**The following activities are defined**

1. Identification of the system or subsystem's key **"stake holders."**
2. Determinations of the stakeholders' **"win conditions."**

3. Negotiations of the stakeholders' win conditions to reconcile them into a set of win-win conditions for all concerned (including the software project team).




The WIN-WIN spiral model introduces three process milestones, called anchor points.

**Life cycle Objectivities (LCO)** defines a set of objectives for each major software engineering activity.

**For example: A** set of objectives established the definition of top-level system /product requirements.

**Life cycle architecture (LCA),** established objectives that must be met as the system and software architecture is defined.

**Initial operational capability (IOC)** represents a set of objectives associated with the preparation of the software for installation / distribution, site preparation prior to installation and assistance required by all that will use or support the software.

**8. Describe Concurrent Development Model**
**(NOV/DEC 2010 & 2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

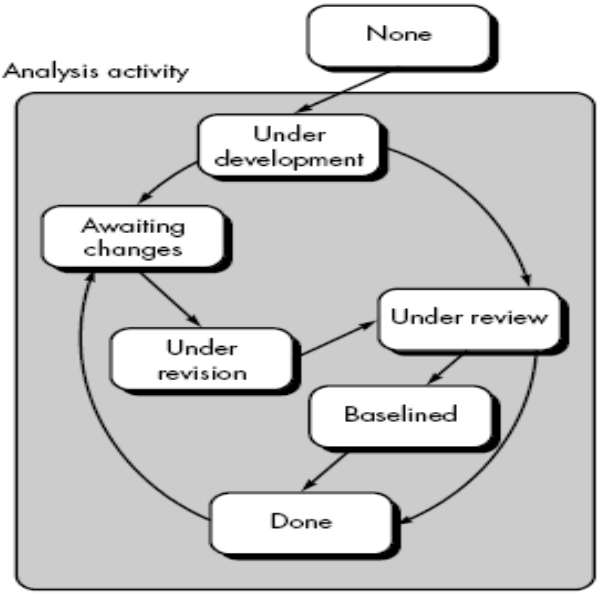
- It is also called as concurrent engineering.
- The concurrent process model defines a series of events that will trigger transitions from state to state for each of the software engineering activities

- The concurrent process model is often used as the paradigm for the development of client/server applications.
- A client/server system is composed of a set of functional components.
- When applied to client/server, the concurrent process model defines activities in two dimensions :
- A system dimension
- A component dimension.
- System level issues are addressed using three activities: design, assembly, and use.
- The component dimension is addressed with two activities: design and realization.
- Concurrency is achieved in two ways:
- (1) System and component activities occur simultaneously and can be modeled using the state-oriented approach described previously.
- (2) A typical client / server application is implemented with many components.
- Each of which can be designed and realized concurrently.
- It provides a schematic representation of one activity with the concurrent process model.
- The activity—analysis—may be in any one of the states noted at any given time.
- Similarly, other activities (e.g., design or customer communication) can be represented in an analogous manner.
- All activities exist concurrently but reside in different states.

**For example,** early in a project the customer communication activity (not shown in the figure) has completed its first iteration and exists in the **awaiting changes state**.

The analysis activity (which existed in the **none state** while initial customer communication was completed) now makes a transition into the under development state.

If, however, the customer indicates that changes in requirements must be made, the analysis activity moves from the **under development state** into the awaiting changes state.




**9. Discuss the various SDLC MODELS.**
**(APR/ MAY 2011) (NOV/DEC 2014).**

**Refer.**

**10. Evaluate Evolutionary Process Model (**
**NOV/DEC 2010 &2013) (MAY/JUNE 2014) (APRIL/MAY 2015)**

They are iterative. They are characterized in a manner that enables software engineers to develop increasingly more complete versions of the software.

1. **Prototyping**
2. **The spiral model , The WIN-WIN spiral Model**
3. **Concurrent Development Model**.

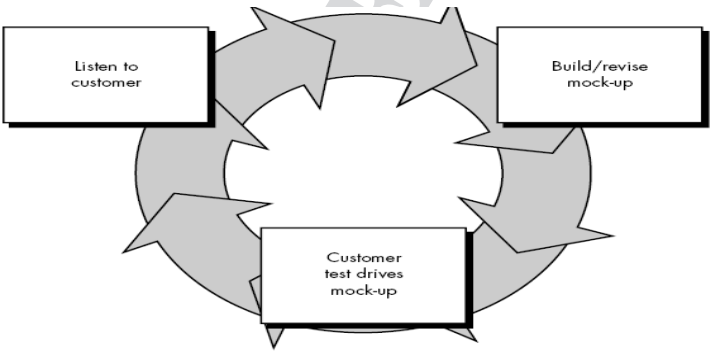
**1. Prototyping**

A customer defines a set of general objectives for software, but does not identify detailed input, processing or output requirements.

In other cases the developer may be unsure of the efficiency of an algorithm, the adaptability of an OS or the form that human machine interaction should take. In these cases prototyping paradigm may offer best approach.

The prototyping paradigm begins with requirements gathering. Developer and customer meet and define the overall objectives for the software, identify whatever requirements are known and outline areas where further definitions mandatory.

A **"quick design"** then occurs. The quick design focuses on a representation of those aspects of the software that will be visible to the customer/user(e.g.. input approaches and output formats).




The quick design leads to the construction of a prototype. The prototype is evaluated by the customer/user and used to refine requirements for the software to be developed.

Iteration occurs as the prototype is tuned to satisfy the needs of the customer, while at the same time enabling the developer to better understand what needs to be done.

The prototype is built the developer attempts to use existing program fragments or applies tools (e.g. report generators, window managers0that enable working programs to be generated quickly.

**Advantages:**

- Customer satisfaction
- Flexible
- Partial products can be viewed by the customer

**Disadvantages**

- No optimal solution
- Time consuming if algorithm is inefficient
- Difficult to maintain

**2. The spiral model**

Refer.

**3. The WIN-WIN spiral Model**

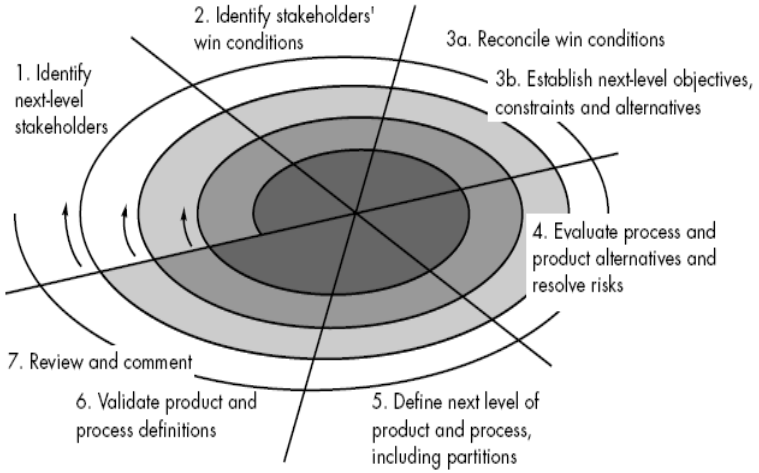
The objectivity of this activity is to elicit project requirements from the customer.

Win-win result: the customer wins by getting the system of product that satisfies the majority of the customer's needs and the developer wins by working to realistic and achievable budgets and deadlines.

A stakeholder is anyone in the organization that has a direct business interest in the system or product to be built and will be rewarded for a successful outcome or criticized if the effort fails.

The following activities are defined

1. Identification of the system or subsystem's key stake holders."
2. Determinations of the stakeholders' "win conditions."
3. Negotiations of the stakeholders' win conditions to reconcile them into a set of win-win conditions for all concerned (including the software project team).




The WIN-WIN spiral model introduces three process milestones, called anchor points.

**Life cycle Objectivities (LCO)** defines a set of objectives for each major software engineering activity. For example a set of objectives established the definition of top-level system /product requirements.

**Life cycle architecture (LCA),** established objectives that must be met as the system and software architecture is defined.

**Initial operational capability (IOC)** represents a set of objectives associated with the preparation of the software for installation/ distribution, site preparation prior to installation and assistance required by all that will use or support the software.

**4. Concurrent Development Model**

- It is also called as concurrent engineering.
- The concurrent process model defines a series of events that will trigger transitions from state to state for each of the software engineering activities
- The concurrent process model is often used as the paradigm for the development of client/server applications.
- A client/server system is composed of a set of functional components.
- When applied to client/server, the concurrent process model defines activities in two dimensions :

- A system dimension
- A component dimension.
- System level issues are addressed using three activities: design, assembly, and use.
- The component dimension is addressed with two activities: design and realization.
- Concurrency is achieved in two ways:

1 System and component activities occur simultaneously and can be modeled using the state-oriented approach described previously.

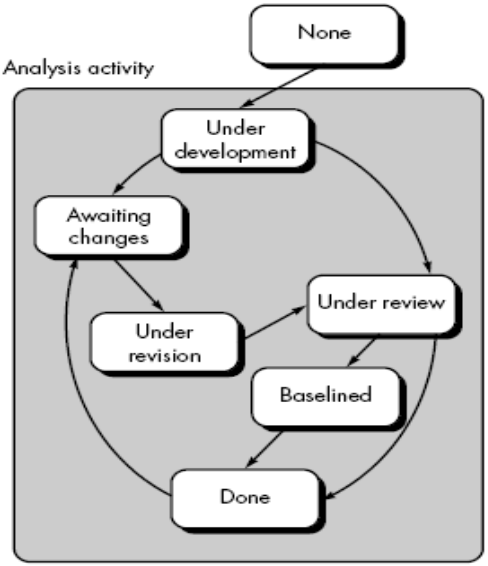
2. A typical client/ server application is implemented with many components, each of which can be designed and realized concurrently.

- It provides a schematic representation of one activity with the concurrent process model.

- The activity—analysis—may be in any one of the states noted at any given time.
- Similarly, other activities (e.g., design or customer communication) can be represented in an analogous manner.
- All activities exist concurrently but reside in diferent states.

For example, early in a project the customer communication activity (not shown in the figure) has completed its first iteration and exists in the **awaiting changes state**.

- The analysis activity (which existed in the **none state** while initial customer communication was completed) now makes a transition into the under development state.
- If, however, the customer indicates that changes in requirements must be made, the analysis activity moves from the **under development state** into the awaiting changes state.




**One element of the concurrent process model**

**11. Describe Specialized Process Model**

**1. Component-based development**

**2. Formal Methods Model**

**1. Component-based development**

The Component-Based Development (CBD) model incorporates many of the characteristics of the spiral model. It is evolutionary in nature, demanding an iterative approach to the creation of software.

However, the component-based development model composes applications from prepackaged software components (called classes). Object-oriented technologies provide the technical framework for a component-based process model for software engineering.

The object oriented paradigm emphasizes the creation of classes that encapsulate both data and the algorithms used to manipulate the data. The engineering activity begins with the identification of candidate classes.

This is accomplished by examining the data to be manipulated by the application and the algorithms that will be applied to accomplish the manipulation.

Corresponding data and algorithms are packaged into a class. Classes created in past software engineering projects are stored in a class library or repository.

Once candidate classes are identified, the class library is searched to determine if these classes already exist. If they do, they are extracted from the library and reused. If a candidate class does not reside in the library, it is engineered using object-oriented methods.

The first iteration of the application to be built is then composed, using classes extracted from the library and any new classes built to meet the unique needs of the application.

Process low then returns to the spiral and will ultimately re-enter the component assembly iteration during subsequent passes through the engineering activity.

The component-based development model leads to software reuse, and reusability provides software engineers with a number of measurable benefits.

It leads to a 70% reduction in development cycle time, an 84 % reduction in project cost and a productivity index of 26.2 compared to industry norm of 16.9.



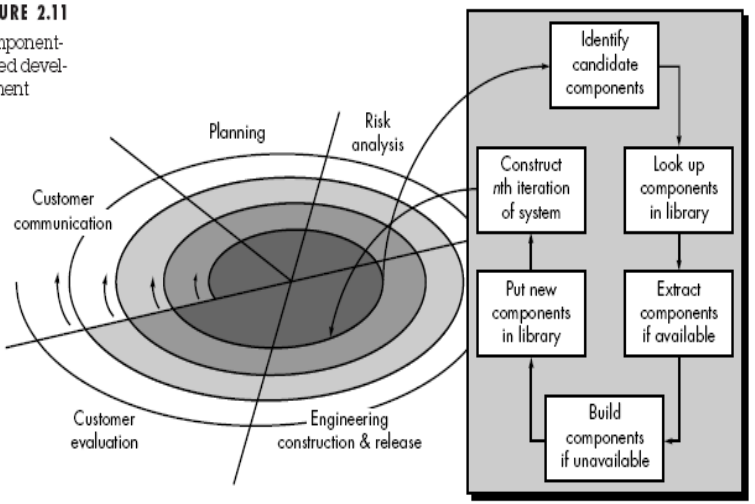



**FIGURE 2.11**
Component-based development

**Advantages:**

- Simplifies software development because it hides the complexity.
- Development of classes supports multiple instances of objects as well as encapsulation leads to reuse.

**Disadvantages:**

- Design by construct in the construction of reliable software.

**Formal Methods Model**

- The formal methods model encompasses a set of activities that leads to formal mathematical specification of computer software.
- Formal methods enable a software engineer to specify, develop, and verify a computer-based system by applying a rigorous, mathematical notation
- When formal methods are used during development, they provide a mechanism for eliminating many of the problems that are difficult to overcome using other software engineering paradigms
- Ambiguity, incompleteness, and inconsistency can be discovered and corrected more easily, not through ad hoc review but through the application of mathematical analysis.
- The formal methods model offers the promise of defect-free software.

**Disadvantages:**

1. The development of formal models is currently quite time consuming and expensive.
2. Because few software developers have the necessary background to apply formal methods, extensive training is required.
3. It is difficult to use the models as a communication mechanism for technically unsophisticated customers.

**12. Discuss about the COCOMO Models (Basic, Intermediate and Detailed) for cost estimation. (APRIL/MAY 2015).**

**COCOMO Model**

COCOMO (Constructive Cost Model) was proposed by Bochm (1981) and is the most complete and thoroughly documented model used in effort estimation. The model provides detailed formulas for determining the development time schedule, overall development effort, effort breakdown by phase and activity as well as maintenance effort.

The primary effort factor is the number of source line of code (SLOC) expressed in thousands of delivered source instructions (KDSI).

These instructions include all program instructions, format statements and job control language statements. They exclude comments and unmodified utility software.

The software development project can be classified into one of the following three categories based on the development complexity

1. Organic
2. Semi-detached
3. Embedded

Normally, data processing programs are considered as application programs and system programs interact directly with the hardware and typically involve meeting timing constraints.

The ratios of the relative levels of product development complexity for the three categories of products are 1:3:9 for application, utility and system programs.

**Embedded**

This class of system is characterized by tight constraints, changing environment, and unfamiliar surroundings.

Projects of the embedded type are novel to the company and usually exhibit temporal constraints.

Good examples of embedded systems are real-time software systems (medicine, aerospace).

**Organic**

This category encompasses all systems that are small relative to project size and team size and have a stable environment, familiar surroundings and relaxed interfaces.

These are simple business systems, data processing systems, and small software libraries.

**Semi-detached**

The software systems falling under this category are a mix of those of organic and embedded nature.

Some examples of software of this class are operating systems, database management systems and inventory management systems.

**Basic form of the COCOMO Model**

The basic form of the COCOMO model is based exclusively on program size expressed in KDSI.

The formula assumes the form

$$\text{Effort} = a*\text{KDLOC}^b$$

Where, a and b are two parameters whose specific values are selected upon the class of the software system.

Estimation of development Effort **For embedded systems**

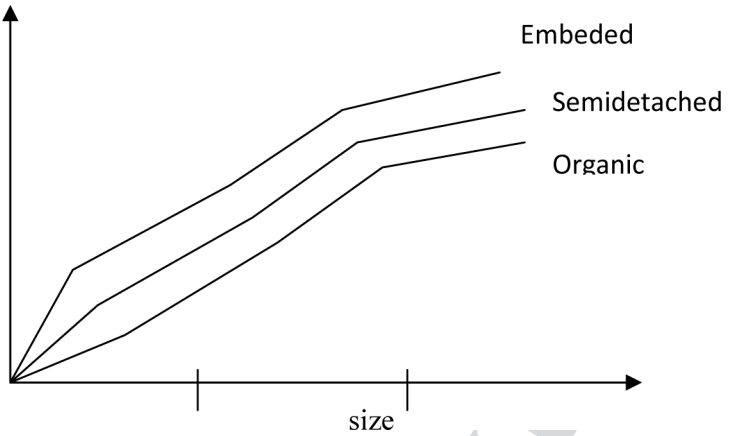
**For organic systems**

$$\textbf{Effort} = 3.6 * \textbf{KDLOC}^{1.20} \textbf{ per month}$$

$$\textbf{Effort} = 2.4 * \textbf{KDLOC}^{1.05} \textbf{ per month}$$

**For semidetached systems**

**Effort = 3.0 * KDLOC$^{1.12}$ per month** The plots of the effort viewed as a function of KDLOC (product size) for the three categories of software are illustrated in the following Figure.




**Estimation of Development Time**

The COCOMO Model determines the development schedule 'M' expressed in months

**For embedded systems**

**For organic systems**

$$\textbf{Effort} = 2.5 * \textbf{KDLOC}^{0.32} \textbf{ months}$$

$$\textbf{Effort} = 2.5 * \textbf{KDLOC}^{0.33} \textbf{ months}$$

**For semidetached systems**

$$\textbf{Effort} = 2.5 * \textbf{KDLOC}^{0.35} \textbf{ months.}$$

From the Figure below, the development time is a sub-linear function of the size of the product. (i.e) when the size of the product increases by two times, the time to develop the product does not double but rises moderately.

From the effort estimation, the project cost can be obtained by multiplying the required effort by the man power cost per month.

But, implicit in this project cost computation is the assumption that the entire project is incurred on account of the manpower cost alone.

In addition to manpower cost, a project would incur costs due to hardware software required for the project and the company overheads for administration, voice space etc.

COCOMO is also used to estimate software maintenance (support effort). The formula is based on the previous effort estimate

$$\mathbf{Effort_{maintenance} = ACT * Effort}$$

Where ACT is annual change traic that is a fraction of KDLOC undergoing change during the year.

**13. Describe Intermediate COCOMO Model**

The basic COCOMO Model assumes that effort and development time are functions of the product size alone.

However, a host of other project parameter besides the product size affects the effort required to develop the product as well as the development time.

The improvement comes in the form of 15 attributes of the product. For each of the attributes, the user of the model has to provide a rating using the following six point scale

**VL(Very Low) LO(Low)**

**NM(Nominal) HI(high) VH(Very High) XH(Extra High)**

The list of attributes is composed of several features of the software and includes product, computer, personnel and project attributes as follows product attributes.

The characteristics of the product that are considered include the inherent complexity of the product, reliability requirements of the product and data bytes per DSI (DATA). The lower rating comes with lower size of a database.

**Product Attributes:** Required software reliability, size & complexity of the product

**Computer Attributes:** The characteristics of the computer that are considered include the execution speed required, storage space required, development turna round time, virtual machine volatility etc.

**Personnel Attributes:**

The attributes of development personnel that are considered include the experience level of personnel, programming capability, analysis capability etc.

**Project Attributes:**

Development environment attributes capture the development facilities available to the developers.

An important parameter that is considered is the sophistication of the automation (CASE) tools used for software development.

## 14. Explain Complete COCOMO Model

A major drawback of both the basic and the intermediate COCOMO models is that they consider a software product as a single homogeneous entity. However, most large systems are made up of several sub systems.

These subsystems may have widely different characteristics. The cost of each subsystem is estimated separately and added. This approach reduces the margin of error in the final estimate.

For example, a distributed Management Information System (MIS) product for an organization having choices at several places across the country can have the following sub-components.

- Database part(Semi-detached software)
- Graphical User Interface part(Organic software)
- Communication part(Embedded software)

To improve the accuracy of their results, the different parameter values of the model can be fine-tuned and validated against an organizations project database to obtain more accurate estimations.

The COCOMO may not be accurate because of the lack of scientific justification but it is required for an engineering approach to Software Project Management.

## 15. Compare and contrast the different lifecycle models.

**(NOV/DEC 2011)**

| S.No | Life Cycle Model | When/Where to apply this model | Example |
|---|---|---|---|
| 1. | Water Fall Model | Simple & Small projects | Billing Software used in small retail shop/ small hotel |

| | | | |
|---|---|---|---|
| 2. | Increment Model | Starting the project with least man power | Word Processing Application – MS Word |
| 3. | Increment Model | Starting the project with least man power | Word Processing Application – MS Word |
| 4. | Rapid Application Model | To complete the Project within 60 – 90 days | Any application |
| 5. | Prototyping Model | If project requirements (input, processing, output )are not clearly defined | Any application |
| 6. | Spiral Model | For Large Scale projects, Follows Risk reduction mechanism like prototyping | ATM system |
| 7. | Concurrent Development Model | To Develop project in client server environment | Any Online System Ex: Online Ticket Reservation System, |
| 8. | Component Based Development | To achieve reusability thus minimizing development cycle time & project cost. | Any application |
| 9. | Unified Process | Popular method of object oriented software development using UML(Unified Modeling Lang) | Any System implemented in Object Oriented Environment |

**16. Discuss briefly about the techniques used for Project scheduling.**

Scheduling of software project does not differ greatly from scheduling of any multitask engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Project Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are two project scheduling methods that can be applied to software development.

Both techniques are driven by information already developed in earlier project planning activities.

- Estimates of effort
- A decomposition of the product function
- The selection of the appropriate process model and task set.
- Decomposition of tasks

Inter dependencies among tasks may be defined using a task network. Tasks, sometimes called the project work breakdown structure (WBS), are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to

(i) Determine the critical path – the chain of task that determines the duration of the project

(ii) Establish "most likely" time estimates for individual tasks by applying statistical models and

(iii) Calculate "boundary times" that define a time "window" for a particular task.

Boundary time calculations can be very useful in software project scheduling.

The important boundary times that may be discerned from a PERT or CPM network.

1. The earliest time that a task can begin when all proceeding tasks are completed in the shortest possible time.

2. The latest time for task initiation before the minimum project completion time is delayed.
3. The earliest finish – the sum of the earliest start and the task duration.
4. The latest finish – the latest start time added to task duration.
5. The total float – the amount of surplus time allowed in scheduling tasks so that the network critical path is maintained on schedule.
6. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as task are completed.

**17. Describe Time Line Charts**

When creating a software project schedule, the planner begin with a set of tasks. If automated tools are used, the work breakdown is input as a task network or task outline.

Effort, duration and start data are then input for each task. In addition, tasks may be assigned to specific individual.

As a consequence of this input, a timeline chart, also called as Gantt chart, is generated. A timeline chart can be developed for the entire project.

Alternatively, separate charts can be developed for each project function or for each individual working on the project. Figure illustrate the format of a timeline chart.

It depicts a part of a software project schedule that emphasizes the concept scoping task for a new word. Processing software product. All project tasks are listed in the left-hand column.

The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, tasks concurrency is implied. The diamond indicate milestones.

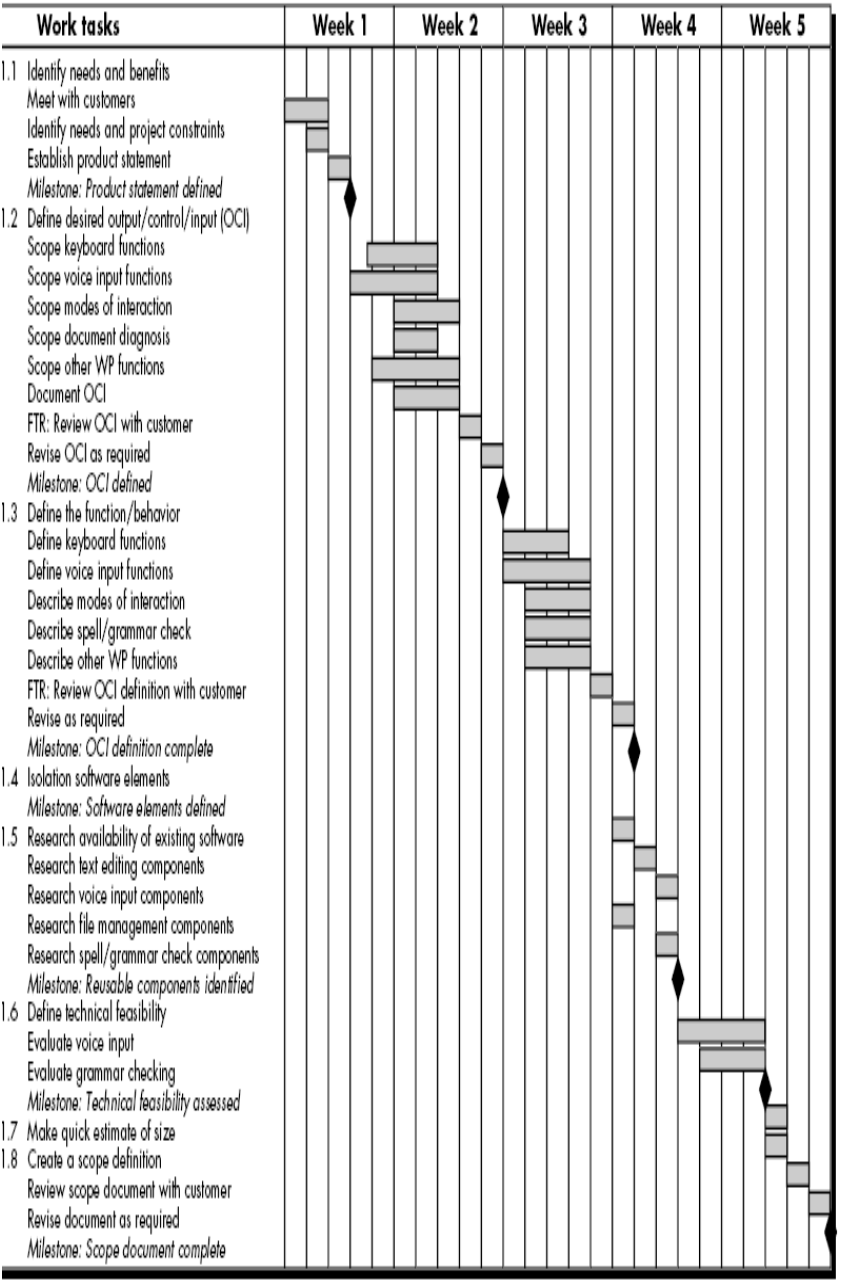



**Fig : An example timeline chart.**

The tasks that is to be solved and implemented. Thus the tasks can be scheduled and monitored properly.

Once the information necessary for the generation of a timeline chart has been input, the majority of the software project scheduling tools produce project tables – a tabular listing of all project tasks, their planned and actual start – and end - data, and a variety of related information's.

Used in conjunction with the timeline chart, project tables enable the project manager to track progress.

**18. How do you Track 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.

Tracking can be accomplished in a number of different ways.

- Conducting periodic project status meetings in which each team member reports progress and problems.
- Evaluating the results of all reviews
- Determining whether formal project milestones have been reached by the scheduled data.
- Comparing actual start-data to planned start-data for each project task listed in the resource table.
- Using earned value analysis to assess progress quantitatively.

Control is employed by a software project manager to administer project resources, cope with problems, and direct project staff.

It is the duty of the project manager to keep track of the project schedule.

When faced with severe deadline pressure, experienced project managers some-times use a project scheduling and control technique called time-boxing.

The time-boxing strategy recognizes that the complete product may not be deliverable by the predefined deadline. Therefore, an incremental software paradigm is chosen and a schedule is derived for each incremental delivery.

**19. Define Earned Value Analysis**

The number of qualitative approaches to project tracking are discussed earlier. Each provides the project manager with an indication of progress, but an assessment of the information provided is somewhat subjective.

A technique for performing quantitative analysis of progress is called earned value Analysis (EVA). The earned value system provides a common value scale for every task, regardless of the type of work being performed.

The total hours to do the whole project are estimated and every task is given an earned value based on its estimated percentage of the total. To determine the earned value, the following steps are performed.

**Step 1**

The budgeted cost of work scheduled (BCWS) is determined for each work task represented in the schedule.

BCWS i is the efort planned for work task i. To determine progress at a given point along the project schedule, the value of BCWS is the sum of BCWS I for all work tasks by that point in time on the project schedule.

**Step 2**

The BCWS values for all work tasks are summed to derive the budget at completion (BAC)

$$\textbf{BAC} = \sum (\textbf{BCWS}_k) \textbf{ for all tasks k.}$$

**Step 3**

The value for budgeted cost of work performed (BCWP) is computed. The value of BCWP is the sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.

SPI is an indication of the deficiency with which the project is utilizing scheduled resources.

An SPI value close to 1.0 indication efficient execution of the project schedule. SV is simply an absolute indication of variance from the planned schedule.

**Schedule Performance Index (SPI) = BCWP / BCWS**

**Schedule Variance (SV) = BCWP - BCWS**

**Percent scheduled for completion = BCWS / BAC**

Provides an indication of the percentage of work that should have been completed by time t.

Percent Complete = BCWP / BAC provides a quantitative indication of the percent of completeness of the project at a given point in time t. It is also possible to compute the actual cost of work performed, ACWP.

Value of ACWP is equal to the sum of the effort actually expanded on work tasks that have been completed by a point in time on the project schedule.

**Cost Performance Index (CPI) = BCWP / ACWP**

**Cost Variance (CV) = BCWP – ACWP**

Earned value analysis illuminates scheduling difficulties before they might otherwise be apparent.

**20. Discuss LOC and FP- Cost Estimation Models in detail.**

**Refer.**

**21. Perform Software Cost Estimation**

The software cost-estimation problem deserves special attention for the following reasons.

1. The development of any software product is usually a unique procedure.
2. There are no two identical systems or projects.
3. The new circumstances could be very different.
4. We are faced with new specification, new hardware and software platforms, new development tools and design methodologies.
5. When considered together, it becomes apparent that the resulting uncertainty about the required resources and associated costs is immense and cannot be ignored.

6. With the increased size of software projects, any estimation to the project.
7. Too limited resources may delay the project.
8. The use of excessive resources means genuine losses in terms of time and revenue.
9. The uncertainty about cost estimates is usually quite high.
10. The uncertainty becomes on inherent component of any software projects.
11. The level of uncertainty may eventually go down over the course of the projects.

The need for cost estimation models is justified for a number of common project requirements.

- Identify, prioritize and justify resource needs.
- Negotiate adequate budgets and establish starting plans.
- Make cost, productivity, and quality and functionality trade-offs.
- Need to quantify the impact of risk.
- Assess the impact of changes and permit preplanning to deal with them.
- Modify budgets to address unexpected events and contingencies.

There are two principal software process cost predictors- expected effort and elapsed time spent on the projects.

There are several factors such as Expert opinion, Analogy, Decomposition, PERT models and other mathematical models are important in cost estimation techniques.

**22. Compute Function Points Models**

Function point metric was proposed by Albrecht in 1979. This metric overcomes many of the shortcomings of the LOC metric.

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.

This is in contrast to the LOC metric, where the size can be accurately determined only after the product has fully been developed.

The function point metric can be used effectively as a means for predicting the size of a system that will be derived from the analysis model.

The conceptual idea underlying 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.

Each function when invoked reads some input data and transforms it to the corresponding output data.

Thus, a computation of the number of input and output data values to a system gives some indication of the number of functions supported by the system.

The number of basic functions that software performs, the size is also dependent on the number of files and the number of interfaces.

The various parameters involved can be explained as follows:

**Number of Inputs**

Each data item input by the user is counted Data inputs should be distinguished from user in quires.

Inquiries are user commands. Inquiries are counted separately.

It must be denoted that individual data items input by the user are not consideration the calculation of the number of inputs, but a group of related inputs are considered as a single input.

**Number of user outputs**

The outputs considered refer to reports printed, screen outputs, error messages produced etc.

While computing the number of outputs the individual data items within a report are not considered, but a set of related data items is counted as one output.

**Number of Inquiries**

Number of inquiries is the number of distinct interactive queries which can be made by the users.

These inquiries are the user commands which require specific action by the system.

**Number of Files**

Each logical file is counted.

A logical file means groups of logically related data.

Thus, logical files can be data structures or physical files.

**Number of Interfaces**

Here the interfaces considered are the interfaces used to exchange information with other systems.

Examples of such interfaces are data files on tapes, disks, communication links with other systems etc.

**Computing function points**

| | | | Weighting factor | | | |
|---|---|---|---|---|---|---|
| **Measurement parameter** | **Count** | | **Simple** | **Average** | **Complex** | |
| Number of user inputs | | x | 3 | 4 | 6 | = |
| Number of user outputs | | x | 4 | 5 | 7 | = |
| Number of user inquiries | | x | 3 | 4 | 6 | = |
| Number of files | | x | 7 | 10 | 15 | = |
| Number of external interfaces | | x | 5 | 7 | 10 | = |
| Count total → | | | | | | |

To compute function points (FP), the following relationship is used:

$$FP = \text{count total} \times \left[0.65 + 0.01 \times \Sigma(F_i)\right]$$

The $F_i$ (i = 1 to 14) are "complexity adjustment values" based on responses to the Following questions:

1. Does the system require reliable backup and recovery?
2. Are data communications required?

3. Are there distributed processing functions?
4. Is performance critical?
5. Will the system run in an existing, heavily utilized operational environment?
6. Does the system require on-line data entry?
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?
8. Are the master files updated on-line?
9. Are the inputs, outputs, files, or inquiries complex?
10. Is the internal processing complex?
11. Is the code designed to be reusable?
12. Are conversion and installation included in the design?
13. Is the system designed for multiple installations in different organizations?
14. Is the application designed to facilitate change and ease of use by the user?

Each of these questions is answered using a scale that ranges from 0 (not important or applicable) to 5 (absolutely essential).

Function point metric computes the size of a software product using the other characteristics of the product as shown in the following expression.

The size of a product in function points (FPS) can be expressed as the weighted sum of these characteristics.

The weights associated with the characteristics were proposed empirically and validated by observations over many projects. Function point is computed in two steps.

**Step 1**

**UFP = (no. of inputs)*4 + (no. of outputs)*5 + (no. of inquiries)*4 + (no.of files)*10 + (no. of interfaces)*10**

Once the unadjusted function point is computed, the technical complexity factor is computed next.

**Step 2**

The Technical Complexity Factor (TCF) refines the UFP measure by considering fourteen other factors such as high transaction rates, throughout and response time requirements etc.

Each of these 14 factors is assigned a value from 0 to 6. The resulting numbers are summed, yielding the total degree of influence (DI).

**TCF ranges from 0.65 to 1.35**

**DI ranges from 0 to 70 Finally FP=UFP*TCF**

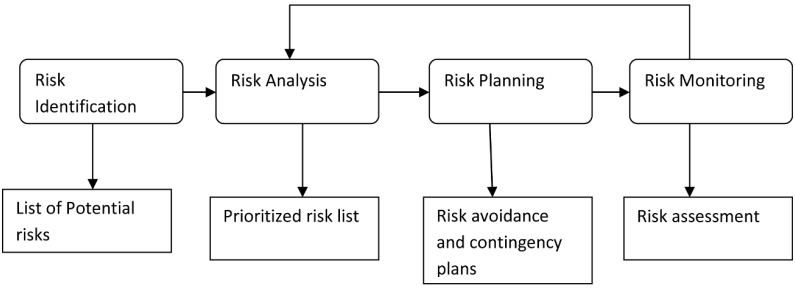
The function point approach to software cost estimation ignores effort reducing technologies such as integrated development environments, CASE tools and executable process descriptions.

**23. What is Risk Management**

- Risk management is concerned with identifying risks and drawing up plans to minimize their effect on project.
- A risk is a probability that some adverse circumstances will occur
- Project risks affect schedule or resources.
- Product risks affect the quality or performance of the software being developed.
- Business risks affect the organization developing or procuring the software.

The risk management process is illustrated in the fig. It involves several stages:

- **Risk identification:** Identify project, product and business risks.
- **Risk Analysis:** Assess the likelihood and consequences of these risks.
- **Risk planning:** Draw up plans to avoid or minimize the effects of the risk.
- **Risk monitoring:** Monitor the risks throughout the project.




**Risk Management Process**

| Risk | Affects | Description |
|---|---|---|
| Staff turnover | Project | Experienced staff will leave the project before it is finished. |
| Management change | Project | There will be a change of organizational management with different priorities. |
| Hardware unavailability | Project | Hardware that is essential for the project will not be delivered on schedule. |
| Requirements change | Project and product | There will be larger number of changes to the requirements than anticipated. |
| Specification delays | Project and product | Specification of essential interfaces is not available on schedule. |
| Size underestimate | Project and product | The size of the system has been underestimated. |
| CASE tool underperformance | Product | CASE tools which support the project do not perform as anticipated. |
| Technology change | Business | The underlying technology on which the system is built is superseded by new technology. |
| Product competition | Business | A competitive product is marketed before the system is completed. |

### Risk Identification

Risk identification may be carried out as a team progress using brainstorming approach or may simply be based on experience.

To help the process, a checklist of different types of risks may be used. There are at least six types of risks that can arise:

**Technology risks** : Risks that derive from the software or hardware technologies that are used to develop the system.

**People risks :** Risks that are associated with people in the development team.

**Organizational risks:** Risks that derive from the organizational environment where the software is being developed.

**Tools Risk :** Risks that derive from the CASE tools and other support software used to develop the system.

**Requirements risk:** Risks that derive from changes to the customer requirements and the process of managing the requirement change.

**Estimation risks:** Risks that derive from the management estimates of the system characteristics and the resources required to build the system.

| Risk type | Possible risks |
|---|---|
| Technology | The database used in the system cannot process as many transactions per second as expected. |
| | Software components that should be reused contains defects that limit their functionality. |
| People | It is impossible to recruit staff with the skills required. |
| | The Key staffs are ill and unavailable at critical times. |
| | Required training for staff is not available. |
| Organizational | The organization is restructured so that different management are responsible for the project. Organizational financial problems force reductions in the project budget. |

| | |
|---|---|
| Tools | The code generated by CASE tools is inefficient.<br>CASE tools cannot be integrated. |
| Requirements | Changes to requirements that require major design network are proposed.<br>Customers fail to understand the impact of requirement changes. |
| Estimation | The time required to develop the software is underestimated.<br>The rate of defect repair is underestimated.<br>The size of the software is underestimated. |

**Risk Analysis**

- Assess probability and seriousness of each risk.
- Probability may be very low, low, moderate, high or very high.
- Risk effects might be catastrophic, serious, tolerable or insignificant.

**Risk Planning**

The risk planning process considers each of the key risks that have been identified and identifies strategies to manage the risk.

These strategies fall into three categories.

1. Avoidance strategies: The probability that the risk will arise is reduced.
2. Minimization strategies: The impact of the risk on the project or product will be reduced.
3. Contingency plans: If the risk arises, contingency plans are plans to deal with that risk.

| Risk | Strategy |
|---|---|
| Organizational financial problems | Prepare a document for senior management showing how the project is making a very important contribution to the goals of the business. |

| | |
|---|---|
| Recruitment problems | Alert customer of potential difficulties and the possibility of delays, investigate buying-in components. |
| Staff illness | Reorganize team so that there is more overlap of work and people therefore understand each other's jobs. |
| Defective components | Replace potentially defective components with bought-in components of known reliability. |
| Organizational restructuring | Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business. |
| Database Performance | Investigate the possibility of buying a higher- performance database. |
| Underestimated development time | Investigate buying in components, investigate use of a program generator. |

**Risk Monitoring**

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.

| Risk type | Potential indicators |
|---|---|
| Technology | Late delivery of hardware or support software, many reported technology problems. |
| People | Poor staff morale, poor relationships amongst team member, job availability. |
| Organizational | Organizational gossip, lack of action by senior management. |

| | |
|---|---|
| Tools | Reluctance by team members to use tools, complaints about CASE tools, demands for higher- powered workstations. |
| Requirements | Many requirements change requests, customer complaints. |
| Estimation | Failure to meet agreed schedule, failure to clear reported defects. |

**24. Explain Specialized Process and Unified Process Models in detail.**

**Specialized Process Model**

a) Component-based development

b) Formal Methods Model

**a) Component-based development**

The Component-Based Development (CBD) model incorporates many of the characteristics of the spiral model. It is evolutionary in nature, demanding an iterative approach to the creation of software.

However, the component-based development model composes applications from prepackaged software components (called classes). Object-oriented technologies provide the technical framework for a component-based process model for software engineering.

The object oriented paradigm emphasizes the creation of classes that encapsulate both data and the algorithms used to manipulate the data. The engineering activity begins with the identification of candidate classes.

This is accomplished by examining the data to be manipulated by the application and the algorithms that will be applied to accomplish the manipulation. Corresponding data and algorithms are packaged into a class. Classes created in past software engineering projects are stored in a class library or repository.

Once candidate classes are identified, the class library is searched to determine if these classes already exist. If they do, they are extracted from the library and reused. If a candidate class does not reside in the library, it is engineered using object-oriented method.

The first iteration of the application to be built is then composed, using classes extracted from the library and any new classes built to meet the unique needs of the application.

Process low then returns to the spiral and will ultimately re-enter the component assembly iteration during subsequent passes through the engineering activity.

The component-based development model leads to software reuse, and reusability provides software engineers with a number of measurable benefits.

It leads to a 70% reduction in development cycle time, an 84 % reduction in project cost and a productivity index of 26.2 compared to industry norm of 16.9.

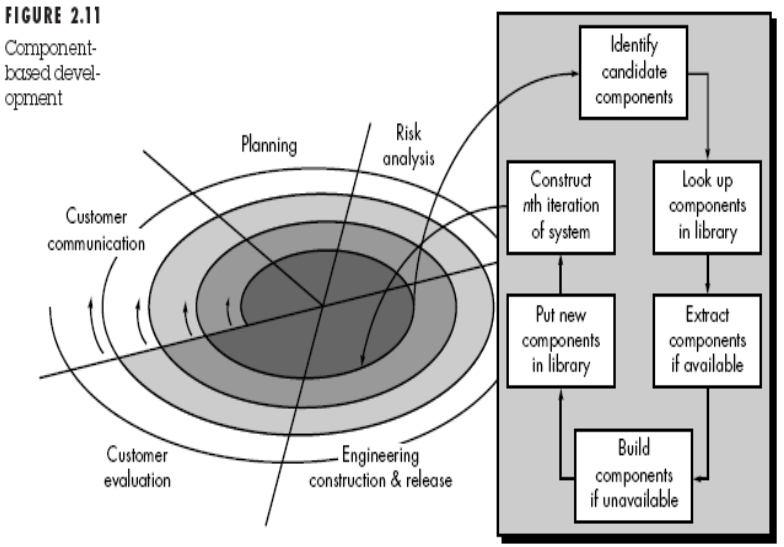



FIGURE 2.11 Component-based development

**Advantages:**

- Simplifies software development because it hides the complexity.
- Development of classes supports multiple instances of objects as well as encapsulation leads to reuse.

**Disadvantages:**

- Design by construct in the construction of reliable software.

**b) Formal Methods Model**

- The formal methods model encompasses a set of activities that leads to formal mathematical specification of computer software. Formal methods enable a software engineer to specify, develop, and verify a computer-based system by applying a rigorous, mathematical notation.
- When formal methods are used during development, they provide a mechanism for eliminating many of the problems that are difficult to overcome using other software engineering paradigms.
- Ambiguity, incompleteness, and inconsistency can be discovered and corrected more easily, not through ad hoc review but through the application of mathematical analysis.
- The formal methods model offers the promise of defect-free software.

**Disadvantages:**

1. The development of formal models is currently quite time consuming and expensive.
2. Because few software developers have the necessary background to apply formal methods, extensive training is required.
3. It is difficult to use the models as a communication mechanism for technically unsophisticated customers.

## 25. Define Unified Process Model

The Unified Process has emerged as a popular iterative software development process for building object-oriented systems. It is an attempt to draw on the best features and characteristics of conventional software process model.

The UP combines commonly accepted best practices, such as an iterative lifecycle and risk-driven development, into a cohesive and well-documented process description.

During the early 1990 James Rumbaugh, Grady Booch and Ivar Jacobson started working on a unified method that combine the best features of each of their individual methods and adopt additional

# UNIT - 3

## ACTIVITY PLANNING AND RISK MANAGEMENT

### PART-A

**1. What are the steps involved in Activity Planning?**

- Ensure that the appropriate resources will be available precisely when required.
- Avoid different activities competing for the same resources at the same time.
- Produce a detailed schedules showing which staff carry out each activity.
- Produce a timed cash flow forecast.

**2. What are the objectives of activity planning?**

**(Nov/Dec 2012)(May/Jun 2013)**

- Feasibility assessment
- Resource allocation
- Detailed costing
- Motivation
- Co-ordination

**3. Define resource allocation.**

- What are the most effective ways of allocating resources to the project?
- When should the resources be available?
- The project plan allows us to investigate the relationship between timescales and resource availability.

**4. How will define the activities?**

- A project is composed of a number of interrelated activities.
- A project may start when at least one of its activities is ready to start.
- A project will be completed when all of the activities it encompasses have been completed.
- If an activity must have a clearly defined start and a clearly defined end-point normally marked by the production of tangible deliverable.

**5. What are the three different approaches to identifying the activities?**

- **Activity-based approach-**constraints stemming from the relationships between projects
- **Product-based approach-**instructor becomes an active member of the project team
- Hybrid approach-Decision support system for software project management.

**6. Write short notes on WBS.**

This involves identifying the main tasks required to complete a project and then breaking each of these down into set of lower-level tasks.

**7. Mention the five levels of WBS.**

- **Project-** engineering resources has been developed by TASK
- **Deliverables-** term for the quantifiable goods or services
- **Components-** designing the floor plane
- **Work-packages-** Models for the description of *software* artifacts
- **Tasks-** Creation and distribution of organizing *software*

**8. How will formulate the network model?**

The first stage in creating a network model is to represent the activities and their interrelationships as a graph.

Then constructing the precedence networks.

**9. What are the rules for constructing precedence networks?**

- A project network should have only one start node.
- A project network should have only one end node.
- A node has duration. Links normally have no duration.
- Precedents are the immediate preceding activities.
- Times moves from left to right
- A network may not contain loops.
- A network should not contain dangles.

**10. Define Hammock activities.**

Hammock activities which, in themselves, have zero duration but are assumed to start at the same time as the first 'hammocked' activity and to end at the same time as the last one.

**11. What is meant by forward pass?**

The forward pass is carried out to calculate the earliest dates on which each activity may be started and completed.

Significance - calculation method used in Critical Path Method.

**12. What is meant by backward pass?**

- The second stage in the analysis of a critical path network is to carry out a backward pass to calculate the latest date at which each activity may be started and finished without delaying the end date of the project.
- The calculating the latest dates, we assume that the latest finish date for the project is the same as the earliest finish date- that is we wish to complete the project as early as possible.

**13. What are the rules of activity –on-arrow rules and conventions? (Nov/Dec2011)**

- A project network may have only one start node
- A project network may have only one end node
- A link has duration Nodes have no duration
- Times moves from left to right 5)Nodes are numbered sequentially
- A network may not contain loops.

**14. Define Risk. (Nov/Dec 2011)**

"An uncertain event or condition that, if it occurs has a positive or negative effect on a project objectives"

Include transferring the risk to another party, avoiding the risk, reducing the negative effect of the risk, and accepting some or all of the consequences of a particular risk.

**15. What are the risks to business impact?**

- Affect of this product on company revenue?
- Reasonableness of delivery deadline?
- number of customers who will use this product
- interoperability constraints
- Sophistication of end users?
- Costs associated with a defective product?

**16. What are things to be considered in risk management? (Nov/Dec 2012)**

- Risk Identification- Organizations and project teams
- Risk Analysis- Includes a download demo and other Decision analysis tools
- **Risk Planning-** assessment is an important part
- **Risk Monitoring-** identify Development Environment *Risks*.

**17. Define Risk Identification.**

- Risk management begins with analyzing the risks involved in the project.
- Risk identification is not a one-off initiative since projects are constantly evolving and new risks arise while other risks may dissipate or reduce in importance.

**18. Define risk analysis and risk monitoring.**

**Risk Analysis** considers each identified risk and makes a judgment about the probability and seriousness of it.

**Risk Monitoring** involves regularly assessing each identified risk to Decide whether that risk is becoming more or less probable and whether the effect of the risk have changed.

**19. Define Risk Planning.**

This project will develop the high- performance, computational technology infrastructure needed to analyze the past, present, and future geospatial distributions of living components of Earth environments**.**

**20. What are the steps in risk planning?**

- Risk identification
- Risk analysis and prioritization.
- Risk planning
- Risk monitoring.

**21. Define risk assessment.**

Using this formula the Risk assessment can be computed as follows:

**Risk exposure = (potential damage) * (probability of occurrence)**

**22. Define Hazard analysis.**

- A **hazard analysis** is a process used to assess risk.
- The results of a hazard analysis are the identification of unacceptable risks and the selection of means of controlling or eliminating them.
- The term is used in several engineering specialties, including avionics, chemical process safety, safety engineering and food safety.

**23. What are called "Free floats " and "interfering floats" ? How are they calculated? (May/Jun 2012)**

Total float is the amount of time by which an activity may be delayed without delaying the project completion.

**Caution:** interpret total floats of activities carefully - all cannot be used independently.

**Free float** is that part of total float which can be used without affecting floats of the succeeding activities.

The part of **total float** which is not free is called interfering float Independent float is the amount of time which can be used without affecting the head and the tail events.

**Total Float ≥ Free Float ≥ Independent Float**

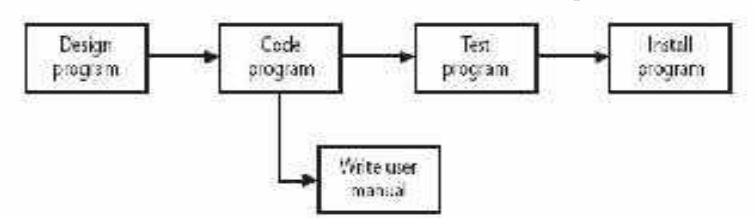
**Head event slack = Earliest start time of the next activity – Earliest completion time of the activity**

**Free float= Total float – Head event slack**

**Interfering float= Total float – Free float.**

**24. What is a "Dangle" in an activity Network ? Give an example? (May/Jun 2012)**

A dangling activity such as "write user manual" should not exist as it is likely to lead to errors in subsequent analysis.




## PART-B

**1. What are the objectives of activities of planning?**

- Is the project possible within required timescales and resource constraints?
- It is not until we have constructed a detailed plan that we can forecast a completion date with any reasonable knowledge of its achievability.

**Resource allocation**

What are the most effective ways of allocating resources to the project?

When should the resources be available?

The project plan allows us to investigate the relationship between timescales and resource availability

**Detailed costing**

How much will the project cost and when is that expenditure likely to take place?

After producing an activity plan and allocating specific resources, we can obtain more detailed estimates of costs and their timing.

**Motivation**

Providing targets and being seen to monitor achievement against targets is an effective way of motivating staff, particularly where they have been involved in setting those targets in the first place.

**Co-ordination**

When do the staff in different departments need to be available to work on a particular project and when do staff need to be transferred between projects?

The project plan, particularly with large projects involving more than a single project team, provides an effective vehicle for communication and coordination among teams.

Activity planning and scheduling techniques place an emphasis on completing the project in a minimum time at an acceptable cost or, alternatively, meeting a set target date at minimum cost.

One effective way of shortening project durations is to carry out activities in parallel.

**2. Write the steps involved in project schedule.**

- A stage of a larger project, the project plan must be developed to the level of showing dates when each activity should start and finish and when and how much of each resource will be required.
- Once the plan has been refined to this level of detail we call it a project schedule.
- Creating a project schedule comprises four main stages.

**First step**

- Step in producing the plan is to Decide what activities need to be carried out and in what order they are to be done.
- From this we can construct an ideal activity plan — that is, a plan of when each activity would ideally be undertaken were resources not a constraint
- This activity plan is generated by Steps 4 and 5 of Step Wise

**Second step**

- The ideal activity plan will then be the subject of an activity risk analysis, aimed at identifying potential problems.

- This might suggest alterations to the ideal activity plan and will almost certainly have implications for resource allocation.

**Third step**

- This is the resource allocation.
- The expected availability of resources might place constraints on when certain activities can be carried out.

**3. Explain the approaches for identifying the activities.**

**Defining activities**

- Some assumptions that will be relevant when we start to produce an activity plan.
- Activities must be defined so that they meet these criteria.
- Any activity that does not meet these criteria must be redefined.
- A project is composed of a number of interrelated activities. A project may start when at least one of its activities is ready to start.
- A project will be completed when all of the activities it encompasses have been completed.
- If an activity must have a clearly defined start and a clearly defined end-point, normally marked by the production of a tangible deliverable.
- An activity requires a resource (as most do) then that resource requirement must be forecast able and is assumed to be required at a constant level throughout the duration of the activity.
- The duration of an activity must be forecastable - assuming normal circumstances, and the reasonable availability of resources.
- Some activities might require that others are completed before they can begin these are known as precedence requirements.
- Identifying activities
- Essentially there are three approaches to identifying the activities or tasks that make up a project
- The activity-based approach
- The product-based approach and
- The hybrid approach.

**4. Explain in detail formulating a network model.**
**(May/Jun 2012)(Nov/Dec 2012)**

**Formulating a network model**

- The first stage in creating a network model is to represent the activities and their interrelationships as a graph.
- In activity-on-node we do this by representing activities as links (arrowed lines) in the graph — the nodes (circles) representing the events of activities starting and finishing.
- Constructing precedence networks
- A project network should have only one start node
- A project network should have only one end node
- A node has duration
- A node represents an activity and, in general, activities take time to execute.
- Links normally have no duration
- precedents are the immediate preceding activities.
- In Figure 6.9, the activity 'Program test' cannot start until both 'Code' and 'Data take-on' have been completed and activity 'Install' cannot start until 'Program test' has finished.
- 'Code' and Data take-on' can therefore be said to be precedents of 'Program test', and 'Program test' is a precedent of 'Install'.
- Note that we do not speak of 'Code' and 'Data take-on' as precedents of 'Install' - that relationship is implicit in the previous statement.
- Time moves front left to right
- A network may not contain loops.
- A loop is an error in that it represents a situation that cannot occur in practice.
- While loops, in the sense of iteration, may occur in practice, they cannot be directly represented in a project network.
- A network should not contain dangles.
- Redraw the network with a final completion activity — which, at least in this case, is probably a more accurate network model.

**5. What is the difference forward pass and backward pass explain with example.**

**Forward Pass**

- The forward pass is carried out to calculate the earliest dates on which each activity may be started and completed.
- Where an actual start date is known, the calculations may be carried out using actual dates.
- Alternatively we can use day or week numbers and that is the approach we shall adopt here.
- By convention, dates indicate the end of a period and the project is therefore shown as starting at the end of week zero (or the beginning of week 1).
- The forward pass and the calculation of earliest start dates is calculated according to the following reasoning.
- Activities A, B and F may start immediately, so the earliest date for their start is zero.
- Activity A will take 6 weeks, so the earliest it can finish is week 6.
- Activity B will take 4 weeks, so the earliest it can finish is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10.
- Activity C can start as soon as A has finished so its earliest start date is week 6.
- It will take 3 weeks so the earliest it can finish is week 9.
- Activities D and E can start as soon as B is complete so the earliest they can each start is week 4.
- Activity D, which will take 4 weeks, can therefore finish by week 8 and activity E, which will take 3 weeks, can therefore finish by week 7.
- Activity G cannot start until both E and F have been completed. It cannot therefore start until week 10 — the later of weeks 7 (for activity E) and 10 (for activity F).
- It takes 3 weeks and finishes in week 13.
- Similarly, Activity H cannot start until week 9 — the later of the two earliest finished dates for the preceding activities C and a

- The project will be complete when both activities H and G have been completed.
- Thus the earliest project completion date will be the later of weeks 11 and 13— that is, week 13.

**The Backward pass**

- The second stage in the analysis of a critical path network is to carry out a backward pass to calculate the latest date at which each activity may be started and finished without delaying the end date of the project.
- In calculating the latest dates, we assume that the latest finish
- date for the project is the same as the earliest finish date — that is, we wish to complete the project as early as possible.
- The latest activity dates are calculated as follows.
- The latest completion date for activities G and 1-1 is assumed to be week 13.
- Activity H must therefore start at week 11 at the latest (13-2) and the latest
- start date for activity G is week 10 (13-3).
- The latest completion date for activities C and D is the latest date at which activity H must start — that is week 11.
- They therefore have latest start dates of week 8 (11-3) and week 7 (11-4) respectively.
- Activities E and F must be completed by week 10 so their earliest start dates are weeks 7 (10-3) and0 (10-10 respectively.
- Activity B must be completed by week 7 the latest start date for both activities D and El so its latest start is week 3 (7-4).
- Activity A must be completed by week 8 (the latest start date for activity C) so its latest start is week2 (8-6).
- The latest start date for the project start is the earliest of the latest start dates for activities A, B and F.
- This is week zero.

- This is, of course, not very surprising since it tells $v_i$ that if the project does not start on time it won't finish on time.

**6. Explain activity-on-arrow networks. (May/Jun 2013)**

**Activity-on-arrow network rules and conventions**

- A project network may have only one start node
- A project network may have only one end node
- A link has duration
- Nodes have no duration
- Time moves from left to right
- Nodes are numbered sequentially
- A network may not contain loops

**7. Explain the categories of risk.**

- Information system
- Computer system
- Description

**8. What are the approaches in risk identification?**

Approaches to identifying risks include:

- **Use of checklists –** usually based on the experience of past projects
- **Brainstorming –** getting knowledgeable stakeholders together to pool concerns
- **Causal mapping –** identifying possible chains of cause and effect

**9. Explain the risk planning.**

**(May/Jun 2012)(Nov/Dec 2012)(Apr 2014)**

The planning for risk includes these steps:

- **Risk identification** – what risks might there be?
- **Risk analysis and prioritization** – which are the most serious risks?
- **Risk planning** – what are we going to do about them?
- **Risk monitoring** – what is the current state of the risk?

- Risks can be dealt with by:
- Risk acceptance
- Risk avoidance
- Risk reduction
- Risk transfer
- **Risk mitigation/contingency** measures

**10. How to evaluate the PERT techniques. (Nov/Dec 2011)(Apr 2014)**

**Network planning models**

These project scheduling techniques model the project's activities and their relationships as a network.

In the network, time flows from left to right.

- The two best known being CPM (Critical Path Method) and PERT (Program Evaluation Review Technique).
- Both of these techniques used an activity-on-arrow approach to visualizing the project as a network where activities are drawn as arrows joining circles, or nodes which represent the possible start and/or completion of an activity or set of activities.
- More recently a variation on these techniques, called precedence network, has become popular.
- This method uses activity-on-node networks where activities are represented as nodes and the links between nodes represent precedence (or sequencing) requirements.
- This latter approach avoids some of the problems inherent in the activity-on-arrow representation and provides more scope for easily representing certain situations.
- It is this method that is adopted in the majority of computer applications currently available.
- These three methods are very similar and it must be admitted that many people use the same name (particularly CPM) indiscriminately to refer to any or all of the methods.

**11. Explain with an example how critical path can be identified in precedence networks? (Nov/Dec 2011)(May/Jun 2013)**

A project usually consists of multiple activities that occur both simultaneously and sequentially.

To determine the flow of these activities, you'll need to create a Precedence Diagram.

After creating the Precedence Diagram, you can identify the activities that would, if delayed, cause your project to come in late. This is the Critical Path definition.

A delay in any of the critical path activities will delay the entire project, regardless of whether the other project activities are completed on or before time.

The act of determining the Critical Path is known as the Critical Path Method or the Critical Path Analysis.

To determine the Critical Path and conduct Critical Path Analysis, you need to:

- Define the duration of each activity.
- Identify all the paths.
- Calculate the duration of each path.
- Identify the longest path.

# UNIT - 4

# PROJECT MANAGEMENT AND CONTROL

## PART-A

**1. Write short notes on monitoring. (Apr 2014)**

Monitoring is collecting and reporting information concerning previously defined project performance elements.

**2. Write short notes on control.**

Control uses the information supplied by the monitoring techniques in order to bring project actual results in line with stated project performance standards.

**3. What are the three steps in project control? (May/Jun 2013)**

- Measuring & Monitoring
- Identifying/tracking key performance metrics
- Evaluating
- Analyzing causes of problems and potential corrective actions
- Correcting
- Taking corrective actions to bring project performance back in line with goals

**4. What are the functions in traffic light-method?**

- Identify the key
- Break these key elements into constituent
- Asses each of the second-level elements on the scale green for on target
- Review all the second-level assessments to arrive at first-level assessments.
- Review first- and second –level assessments to produce an overall Assessments.

**5. Define Gantt Chart**

One of the simplest and oldest techniques project progress is the Gantt -chart this is essentially an activity bar chart indicating scheduled activity dates and duration frequently augmented with activity floats.

**6. Define slip chart.**

A slip chart is a very alternative favored by some project managers who believe it provides a more striking visual indication of those activities that are not progressing to schedule the more the slip line bends, the greater variation from the plan.

**7. Write short notes on Earned Value Analysis. (Nov/Dec 2011)**

- It is a measure of progress
- It enables us to assess the "percent of completeness" of a project using quantitative analysis rather than rely on a gut feeling
- "Provides accurate and reliable readings of performance from as early as 15 percent into the project."
- A technique used to help determine and manage project progress and the magnitude of any variations from the planned values concerning cost, schedule, and performance.

**8. Define Scheduled variance.**

The schedule variance is measured in cost terms as EV-PV and indicates the degree to which the value of completed work differs from that planned.

**9. What are the deciding levels of monitoring? (May/Jun 2013)**

- Critical path activities
- Activities with no free float
- Activities with less than a specified float4)Activities using critical resources
- High risk activities.

**10. What are the steps in change control procedures? (Apr 2014)**

- One or more users might perceive a need for a modification to a system and ask for change request to be passed to the development staff.

- The user management consider the change request and, if they approve it, pass it to the development management.

**11. Define managing contracts.**

**Contract management** or **contract administration** is the management of contracts made with customers, vendors, partners, or employees. Contract management includes negotiating the terms and conditions in contracts and ensuring compliance with the terms and conditions, as well as documenting and agreeing on any changes that may arise during its implementation or execution.

It can be summarized as the process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk.

**12. What are the different types of contract?**

- Fixed price contracts.
- Time and materials contracts.
- Fixed price per delivered unit contracts.

**13. What is meant by fixed price contracts?**

- Involve a fixed total price for a well-defined product or service
- May include incentives for meeting certain performance objectives or penalties if those objectives are not met.

**14. Mention the advantages and disadvantages of fixed price contracts.**

**Advantages**

- Known customer expenditure
- Supply motivation
- Higher prices to allow for contingency

**Disadvantages**

- Difficulties in modifying requirements
- Upward pressure on the cost of changes
- Threat to system quality.

**15. Define time and materials contracts.**

- Hybrid of both fixed price and cost reimbursable, often used by consultants
- The buyer pays the seller for both the time and materials required to complete the work
- Resembles a cost-reimbursable contract because it is open-ended and full cost of project is not predetermined
- But can resemble a fixed-price contract if unit rates are set

**16. What are the advantages and disadvantages are time and materials contracts?**

**Advantages**

- Ease of changing requirements.
- Lack of price pressure

**Disadvantages**

- Customer liability
- Lack of incentives for supplier.

**17. Define fixed per unit delivered contracts.**

- Require the buyer to pay the seller a predetermined amount per unit of service
- Detailed requirements analysis done and frozen before starting the contract
- Any change after then, need renegotiating

**18. What the advantages and disadvantages are of fixed per unit delivered contracts?**

**Advantages**

- Customer understanding
- Comparability
- Emerging functionality
- Supplier efficiency
- Life-cycle range

**Disadvantages**

- Difficulties with software size measurements
- Changing requirements.

**19. What are the processes of evaluation need?**

- Security of the proposal documents
- Interviewing supplier's representatives.
- Demonstrations.
- Practical tests.

**20. What are the services to be provided in contracts?**

- Training
- Documentation
- Installation
- Conversion of existing files
- Maintenance agreements
- Transitional insurance agreements.

**21. Define Acceptance.**

When the work has been completed, the customer needs to take action to carry out acceptance testing.

The contract may put a time limit on how long acceptance testing can take, so the customer must be organized to carry out this testing before the time for requesting correction expires.

**22. Write any two advantages of function point analysis**

**(Nov/Dec 2011)**

- Improved project estimating;
- Understanding project and maintenance productivity;
- Managing changing project requirements;
- Gathering user requirements**.**

**23. List the important roles of the configuration librarian (May/Jun 2012).**

A configuration librarian is the owner of the configuration library and manager of all master copies of configuration items (CIs).

In a multi-customer environment, a configuration librarian is a super user for the accounts he or she is assigned to.

A configuration librarian has the following responsibilities:

- Make sure the CIs registered in the database are correct and up to date
- Configure discovery
- Create CIs
- Update a CI instance
- Delete a CI
- Register a new CI
- Transfer ownership of a CI
- Transition a CI state
- Assign or remove CIs to or from an organization
- Create extended attributes for a CI type
- View CIs
- Generate a configuration management report

**24. Name the popular visual tools used for monitoring and tracking the project progress. (May/Jun 2012).**

- PERT
- CPM

## PART-B

**1. Explain project control cycle in detail.**

- Measuring & Monitoring
- Identifying/tracking key performance metrics
- Evaluating
- Analyzing causes of problems and potential corrective actions
- Correcting
- Taking corrective actions to bring project performance back in line with goals

**2. Explain the method Earned value Analysis.**
**(Nov/Dec 2011)(Apr 2014)**

- It is a measure of progress
- It enables us to assess the "percent of completeness" of a project using quantitative analysis rather than rely on a gut feeling
- "Provides accurate and reliable readings of performance from as early as 15 percent into the project."
- A technique used to help determine and manage project progress and the magnitude of any variations from the planned values concerning cost, schedule, and performance.
- Planned value (PV) or Budgeted cost of work scheduled (BCWS) – original estimate of the effort/cost to complete a task (compare with idea of a 'price')
- Earned value (EV) or Budgeted cost of work performed (BCWP) – total of PVs for the work completed at this time.

**Earned value – an example**

- Tasks
- Specify module 5 days
- Code module 8 days
- Test module 6 days

- At the beginning of day 20, PV = 19 days
- If everything but testing completed, EV = 13 days
- Schedule variance = EV-PV i.e. 13-19 = -6
- Schedule performance indicator
- (SPI)=EV/PV i.e 13/19 = 0.68
- **Actual cost (AC)** is also known as Actual cost of work performed (ACWP)
- In previous example, if

'Specify module' actually took 3 days (planned 5 days)

'Code module' actually took 4 days (planned 8 days)

- Actual cost = 7 days
- Cost variance(CV) = EV-AC

i.e. 13-7 = 6 days

Cost performance indicator(CPI)=EV/AC i.e = 13/7 = 1.86

- Positive CV or CPI > 1.00 means project under budget or the work is completed better than planned.

**3. Explain the change in control procedures.**

**(Nov/Dec 2011)(May/Jun 2012)**

- Change control is a systematic approach to managing all changes made to a product or system.
- The purpose is to ensure that no unnecessary changes are made, that all changes are documented, that services are not unnecessarily disrupted and that resources are used efficiently.
- One or more users might perceive a need for a modification to a system and ask for change request to be passed to the development staff.
- The user management consider the change request and, if they approve it, pass it to the development management.
- The role of change control:
- Identifying items that need to be subject to change control

- Management of a central repository of the master copies of software and documentation
- Administering change procedures
- Maintenance of access records

**Typical change control process**

- One or more users might perceive the need for a change
- User management decide that the change is valid and worthwhile and pass it to development management
- A developer is assigned to assess the practicality and cost of making the change
- Development management report back to user management on the cost of the change; user management decide whether to go ahead
- One or more developers are authorized to make copies of components to be modified
- Copies modified. After initial testing, a test version might be released to users for acceptance testing
- When users are satisfied then operational release authorized – master configuration items updated

**4. Explain the different types of contract in detail.**
**(May/Jun 2012)(May/Jun 2013) (Apr 2014)**

- Fixed price contracts
- Time and materials contracts
- Fixed price per delivered unit

**5. Explain fixed price contracts with advantages and disadvantages.**
**(May/Jun 2012)**

- Involve a fixed total price for a well-defined product or service
- May include incentives for meeting certain performance objectives or penalties if those objectives are not met.

**Advantages to customer:**

- Known expenditure
- Supplier motivated to be cost-effective

**Disadvantages:**

- Supplier will increase price to meet contingencies
- Difficult to modify requirements
- Upward pressure on the cost of changes
- Threat to system quality

**6. Explain time and material contract with advantages and disadvantages**

- Hybrid of both fixed price and cost reimbursable, often used by consultants
- The buyer pays the seller for both the time and materials required to complete the work
- Resembles a cost-reimbursable contract because it is open-ended and full cost of project is not predetermined
- But can resemble a fixed-price contract if unit rates are set

**Advantages to customer:**

- Easy to change requirements
- Lack of price pressure can assist product quality

**Disadvantages:**

- Customer liability - the customer absorbs all the risk associated with poorly defined or changing requirements
- Lack of incentive for supplier to be cost-effective

**7. What are the stages in contract management?**
**(Nov/Dec 2011)(May 2013) (Apr 2014)**

- Requirement plan
- Evaluation plan
- Invitation to tender
- Evaluation of proposals.

**8. Explain Fixed price per deliver unit with advantages and disadvantages**

- Require the buyer to pay the seller a predetermined amount per unit of service
- Detailed requirements analysis done and frozen before starting the contract
- Any change after then, need renegotiating

**Advantages for customer**

- Customer understanding of how price is calculated
- Comparability between different pricing schedules
- Emerging functionality can be accounted for
- Supplier incentive to be cost-effective

**Disadvantages**

- Difficulties with software size measurement - may need independent FP counter
- Changing (as opposed to new) requirements: how do you charge?

**9. Describe the various ways in visualizing the progress of the project.**
**(Nov/Dec 2012)(May/Jun 2013)**

- Gantt chart
- Slip chart
- Timeline

**10. Explain the process of prioritizing monitoring. Give example.**
**(Nov/Dec 2012)(May/Jun 2013)**

- Priority list of activity to monitor
- Critical activities
- Non-critical activities with no free float
- Non-critical activities with less than a specified float
- High risk activities
- Activities with critical resources

# UNIT - 5

## STAFFING IN SOFTWARE PROJECTS

### PART-A

**1. What are the objectives of managing people and organizing teams? (Apr 2014)**

- Identify some of the factors that influence people's behavior in project.
- Select and induct new staff into a project.
- Increase staff motivation.
- Improve group working.
- Use the most appropriate leadership styles.

**2. What are the three basic objectives of organizational behavior. (Apr 2014)**

- To select the best people for the job.
- To instruct them in the best methods.
- To give instructions in the form of higher wages to the best workers.

**3. What are the factors consider in X theory? (May/June 2013)**

- The average human has an innate dislike of work.
- There is a need therefore for correction, direction and control.
- People tend to avoid responsibility.

**4. What are the factors consider in Y theory?**

- Work is as natural as rest or play.
- External control and coercion are not the only ways of bringing about effort directed towards an organization's ends.
- The average human can learn to accept and further seek responsibility.

**5. Define Motivation.**

- Motivation is a general term applying to the entire class of drives, desires, needs, wishes, and similar forces.
- Managers, as a part of motivating their staff, do all such things which they hope will satisfy these drives and desires and induce the subordinates to act in a desired manner.

**6. What are the needs in Maslow's hierarchy theory? (May/Jun 2012)**

- **Physiological Needs** - attention turns to safety and security
- **Security or Safety Needs-** Calculation, Domain, Consulting,
- **Affiliation or Social Needs -** Developing New Programs
- **Esteem Needs-** needs for esteem can become dominant
- **Self-actualization Needs -** include symmetry

**7. Write short notes on Herzberg's Motivation-Hygiene theory**
**Herzberg's Motivation-Hygiene Theory**

Maslow's need approach has been considerably modified by Frederick Herzberg.

His research purports to find a two-factor theory of motivation.

In one group of needs are such things as company policy and administration, supervision, working conditions, interpersonal relations, salary, status, and job security.

These were found by Herzberg and his associates to be only *dissatisfies* and not motivators.

Their existence does not motivate in the sense of yielding satisfaction; their lack of existence would, however, result in dissatisfaction.

Herzberg called them maintenance, hygiene or job context factors.

**8. Write short notes on vroom's expectancy theory.**

Force = valence x expectancy

Where **force** is the strength of a person motivation,

**Valence** is the strength of an individual's preference for an outcome, and **expectancy** is the probability that a particular action will lead to a desired outcome.

**9. What are the factors to be considered in the Old Ham-Hackman job characteristic model?**

- **Skill variety-** one or more of the offerings available from a *variety* of organizations
- **Task variety-** enhance Key words
- **Task significance-** autonomy, and feedback from the job
- **Autonomy-** for Consulting & *Software* Companies
- **Feedback-** submit your comments and suggestions

**10. Mention the methods of improving motivation.**

Set specific tasks, provide feedback, and consider job design.

**11. How to becoming a team?**

- **Forming-** The members of the groups get to know each other and try to set up
- some ground rules about behaviour
- **Storming-** one nice packaging, all for publishing need
- **Norming-** Asset Management is a powerful and complete asset management solution
- **Performing-** Optimize project delivery across the *software*
- **Adjourning - added** a final stage

**12. Define Forming.**

The members of the groups get to know each other and try to set up some ground rules about behaviour.

**13. Define team worker.**

Skilled at creating a good working environment to manage all the people who are Developing Projects, team proposed to extend these concepts.

**14. What are the two categorized for Decision making?**

- **Structured-** generally relatively simple, routine Decisions where rules can be applied in a fairly straightforward way
- **Unstructured**- more complex and often requiring a degree of creativity.

**15. Mention some mental obstacles to good Decision making.**
**(May/Jun 2013)**

- **Faculty heuristics-** is an innovative effort by students and members of staff
- **Escalation of commitment-** behavior, sunk cost, risk propensity, risk perception,
- **Information overhead-** developers analyze, design, and develop *software*

**16. What are the measures to reduce the disadvantages of group Decision making?**

- The cooperation of a number of experts.
- The problem is presented to the experts.
- The experts record their recommendations.
- These recommendations are collated and reproduced.
- The collect responses are recirculated.

**17. Define Leadership.**

The ability of a superior to influence the behavior of his subordinates and persuade them to follow a particular course of action, do suggest here is that any analysis of project management.

**18. What are the functions of leader?**

- Goal-setter
- Planner,
- Executive,
- Expert,
- Spokesman,
- Controller of internal relationships,
- Administrator of rewards and punishments,
- Arbitrator and mediator,
- Role model,
- Symbol of the group, and
- Father figure.

**19. What are the leadership models/theories?**

- Trait theory,
- Leadership styles based on authority,
- Managerial grid,
- Continuum approach,
- Feidler's contingency model, and
- Path-goal theory.

**20. What are the leadership styles?**

- **Directive autocrat-** This manager makes all the Decisions unilaterally and manages learning to Lead
- **Permissive autocrat-** Concepts using simple and precise free downloadable
- **Directive democrat-** Management Styles Permissive Democrat Directive Autocrat document
- **Permissive democrat-** Makes Decisions participative subordinates have latitude

**21 Define Stress. (Nov/Dec 2011)(Nov/Dec 2012)**

- Projects are about overcoming obstacles and achieving objectives.
- Almost by definition both the project manager and team members will be under pressure.
- Once a project gets rolling, you should expect members to be putting in at least 60 hours a week.
- The project must except to put in as many hours as possible.

**22 Define Departmentation**

- The process of grouping activities is commonly known as departmentation.
- This is the first real task in designing an organization Project Methods.
- Staff provided course ware development and training on office automation *software.*
- *Staff* trying to escape poverty, and engaging in *democratic* reforms.

**23 What do you understand by "Egoless Programming".**

**(May/Jun 2012)**

Egoless programming is a style of computer programming in which personal factors are minimized so that quality may be improved.

**24 What is bespoke system. (Nov/Dec2012)**

- Bespoke is a term used in the United Kingdom and elsewhere for an individually- or custom-made product or service.
- Traditionally applied to custom-tailored clothing, the term has been extended to information technology, especially for software consulting services.
- Typically, software consulting company's offer packaged (already invented and generally applicable) software and bespoke software for client needs that can't be satisfied by packaged software.
- In the U.S., bespoke software is often called customer custom-designed software.

**21. What is the use of checkpoints in monitoring. (Nov/Dec 2012)**

- Based on regular time intervals
- Can be weekly or monthly or quarterly
- Depend on what to check and how to
- Based on a particular event
- At the end of each activity
- In the middle of a critical activity
- Should be set before the plan was published
- Make sure everyone knows when and what the check points are

## PART-B

**1. Explain the stepwise framework where staffing concerns are important.**

**Main Concerns**

- 4 main concerns:
- Staff Selection
- Staff Development
- Staff Motivation
- Well-being Staff during course of project ' Step Wise' - an overview
- Select project
- Identify project objectives
- Identify project infrastructure
- Analyse project characteristics
- Identify products and activities
- Estimate effort for activity
- Review/ publicize plan
- Identify activity risks
- Allocate resources
- Execute plan
- Lower level planning-Review Lower level detail For each activity

**2. Explain X theory and Y –theory. In detail.**

**Theory X**

- The average human has an innate dislike of work
- There is a need therefore for coercion, direction and control
- People tend to avoid responsibility

**Theory Y**

- Work is as natural as rest or play
- External control and coercion are not the only ways of bringing about effort directed towards an organization's end
- Commitment to objectives is a function of the rewards associated with their achievement
- The average human can learn to accept and further seek responsibility
- The capacity to exercise imagination and other creative qualities is widely distributed.

**3. Explain the recruitment process.**
**(Nov/Dec 2011 & 2012) (May/June 2013) (Apr 2014)**

- Selecting the right person for the job
- Besides the s/w tools the individuals selected for a job affect the programming productivity.
- Experience
- Person who can communicate well.
- Recruitment Process
- Recruitment is often an organizational responsibility.
- Eligible candidates- have a curriculum vitae which shows the right and required details
- Suitable candidates- who can actually do the job well.
- Assesses actual skills rather than experience.
- Create a job specification.
- Create a job holder profile.
- Obtain applicants.
- Examine CV's.
- Interviews.
- Other procedures.

**4. Define motivation. Explain Maslow's hierarchy of needs.**

Motivation and application can often make up for shortfalls in innate skills

Taylor's approach - financial incentives

Abraham Maslow (1908-1970)

- Motivations vary from individual to individual.
- People will be motivated by different things at different times.
- People will always feel dissatisfied, but the focus of the dissatisfaction changes over time.
- Hierarchy of needs – as lower ones fulfilled, higher ones emerge

**Physiological needs**

Physiological needs are the physical requirements for human survival.

If these requirements are not met, the human body cannot function properly and will ultimately fail.

Physiological needs are thought to be the most important; they should be met first.

Air, water, and food are metabolic requirements for survival in all animals, including humans.

Clothing and shelter provide necessary protection from the elements.

While maintaining an adequate birth rate shapes the intensity of the human sexual instinct, sexual competition may also shape said instinct.

**Safety needs**

With their physical needs relatively satisfied, the individual's safety needs take precedence and dominate behavior.

In the absence of physical safety – due to war, natural disaster, family violence, childhood abuse, etc. – people may (re-) experience post-traumatic stress disorder or trans generational trauma.

In the absence of economic safety – due to economic crisis and lack of work opportunities – these safety needs manifest
themselves in ways such as a preference for job
security, grievance procedures for protecting the individual from unilateral authority, savings accounts, insurance policies, reasonable disability accommodations, etc.

This level is more likely to be found in children because they generally have a greater need to feel safe.

**Safety and Security needs include:**

- Personal security
- Financial security
- Health and well-being
- Safety net against accidents/illness and their adverse impacts

**Love and belonging**

After physiological and safety needs are fulfilled, the third level of human needs is interpersonal and involves feelings of belongingness.

This need is especially strong in childhood and can override the need for safety as witnessed in children who cling to abusive parents.

Deficiencies within this level of Maslow's hierarchy – due to hospitalism, neglect, shunning, **ostracism,** etc. – can impact the individual's ability to form and maintain emotionally significant relationships in general, such as:

- Friendship
- Intimacy
- Family

**Esteem**

All humans have a need to feel respected; this includes the need to have self-esteem and self-respect. Esteem presents the typical human desire to be accepted and valued by others.

People often engage in a profession or hobby to gain recognition. These activities give the person a sense of contribution or value. Low self-esteem or an inferiority complex may result from imbalances during this level in the hierarchy.

People with low self-esteem often need respect from others; they may feel the need to seek fame or glory. However, fame or glory will not help the person to build their self-esteem until they accept who they are internally.

Psychological imbalances such as depression can hinder the person from obtaining a higher level of self-esteem or self-respect.

**5. Explain the Expectancy theory of Motivation.**

V room's expectancy theory assumes that behavior results from conscious choices among alternatives whose purpose it is to maximize pleasure and to minimize pain.

V room realized that an employee's performance is based on individual factors such as personality, skills, knowledge, experience and abilities.

He stated that effort, performance and motivation are linked in a person's motivation. He uses the variables Expectancy, Instrumentality and Valence to account for this.

**Expectancy** is the belief that increased effort will lead to increased performance i.e. if I work harder, then this will be better.

This is affected by such things as:

- Having the right resources available (e.g. raw materials, time)
- Having the right skills to do the job
- Having the necessary support to get the job done (e.g. supervisor support, or correct information on the job)

**Instrumentality** is the belief that if you perform well that a valued outcome will be received. The degree to which a first level outcome will lead to the second level outcome. i.e. if I do a good job, there is something in it for me. This is affected by such things as:

- Clear understanding of the relationship between performance and outcomes – e.g. the rules of the reward 'game'
- Trust in the people who will take the Decisions on who gets what outcome
- Transparency of the process that Decides who gets what outcome

**Valence** is the importance that the individual places upon the expected outcome. For the valence to be positive, the person must prefer attaining the outcome to not attaining it.

**For example**, if someone is mainly motivated by money, he or she might not value offers of additional time off.

The three elements are important behind choosing one element over another because they are clearly defined: effort-performance expectancy **(E>P expectancy)** and performance-outcome expectancy **(P>O expectancy).**

**>P expectancy:** our assessment of the probability that our efforts will lead to the required performance level.

**P>O expectancy:** our assessment of the probability that our successful performance will lead to certain outcomes.

Crucially, Vroom's expectancy theory works on **perceptions** – so even if an employer thinks they have provided everything appropriate for motivation, and even if this works with most people in that organization, it doesn't mean that someone won't perceive that it doesn't work for them.

At first glance expectancy theory would seem most applicable to a traditional-attitude work situation where how motivated the employee is depends on whether they want the reward on offer for doing a good job and whether they believe more effort will lead to that reward.

However, it could equally apply to any situation where someone does something because they expect a certain outcome.

For example, I recycle paper because I think it's important to conserve resources and take a stand on environmental issues (valence); I think that the more effort I put into recycling the more paper I will recycle (expectancy); and I think that the more paper I recycle then less resources will be used (instrumentality).

Thus, Vroom's expectancy theory of motivation is not about self-interest in rewards but about the associations people make towards expected outcomes and the contribution they feel they can make towards those outcomes.

**6. What the methods involved in motivation? (Nov/Dec 2011)(May/Jun 2013)**

- Fair pay and conditions
- A comfortable, safe, working environment
- Opportunities for employees to socialize and make friends
- Clearly defined work responsibilities and goals
- Education and training opportunities
- Career opportunities

**7. What are the steps needed to become a team? (Nov/Dec 2012)**

Becoming a team

Five basic stages of development:

- Forming
- Storming
- Norming
- Peforming
- Adjourning

**8. Explain the leadership style in detail. (Nov/Dec 2011)**

- **Position power**
- **Coercive power –** able to threaten punishment
- **Connection power** – have access to those who do have power
- **Legitimate power –** based on a person's title conferring a special status
- **Reward power –** able to reward those who comply
- Leadership styles
- **Task orientation** – focus on the work in hand
- **People orientation –** focus on relationships
- Where there is uncertainty about the way job is to be done or staff are inexperienced they welcome task oriented supervision
- Uncertainty is reduced – people orientation more important
- Risk that with reduction of uncertainty, managers have time on their hands and become more task oriented (interfering)

**9. Explain the organizational structures.**
**(May/Jun 2012)(Nov/Dec 2012)(May/Jun 2013)**

Formal organisational structures are categorised as:

- **Line** organisational structure.
- **Staff or functional** authority organisational structure.

- **Line and staff** organisational structure.
- **Committee** organisational structure.
- **Divisional** organisational structure.
- **Project** organisational structure.
- **Matrix** organisational structure and
- **Hybrid** organisational structure.

**10. How to improve group performance ? (Nov/Dec 2011)**

**Know Your Team**

As a leader, you need to guide the development of your group. So, start by learning about the phases that a group goes through as it develops.

When you understand these, you'll be able to preempt problems that could arise, including issues with poor group dynamics.

Next, use Benne and Sheats' Group Roles to identify positive and negative group roles, and to understand how they could affect the group as a whole.

This will also help you plan how to deal with potential problems.

**Tackle Problems Quickly**

If you notice that one member of your team has adopted a behavior that's affecting the group unhelpfully, act quickly to challenge it.

Providefeedback that shows your team member the impact of her actions, and encourage her toreflect on how she can change her behavior.

**Define Roles and Responsibilities**

Teams that lack focus or direction can quickly develop poor dynamics, as people struggle to understand their role in the group.

Create a team charter – defining the group's mission and objective, and everyone's responsibilities – as soon as you form the team.

Make sure that everyone has a copy of the document, and remind people of it regularly.

**Break Down Barriers**

Use team- building exercises to help everyone get to know one another, particularly when new members join the group.

These exercises ease new colleagues into the group gently, and also help to combat the **"black sheep effect,"** which happens when group members turn against people they consider different.

**Focus on Communication**

Open communication is central to good team dynamics, so make sure that everyone is communicating clearly.

Include all of the forms of communication that your group uses – emails, meetings, and shared documents, for example – to avoid any ambiguity.

If the status of a project changes, or if you have an announcement to make, let people know as soon as possible.

That way, you can ensure that everyone has the same information.

**Pay Attention**

Watch out for the warning signs of poor group dynamics. Pay particular attention to frequent unanimous decisions, as these can be a sign of groupthink, bullying, or free riding.

If there are frequent unanimous Decisions in your group, consider exploring new ways to encourage people to discuss their views, or to share them anonymously.

**11. Oldham-Hackman job characteristic model. (May/Jun 2012)**

Skill variety- one or more of the offerings available from a variety of organizations

Task variety- enhance Key words

Task significance- autonomy, and feedback from the job

Autonomy- for Consulting & Software Companies

Feedback- submit your comments and suggestions.

**12. Stress and its significance in IT Projects. (May/Jun 2012)**

Stress is a condition or feeling experienced when a person perceives that "demands exceed the personal and social resources the individual is able to mobilize.

"Occupational stress can be defined as the harmful physical and emotional responses that occur when the requirements of the job do not match the capabilities, resources, or needs of the worker.

Job stress can lead to poor health and even injury. The concept of Occupational stress is often confused with challenge, but these concepts are not the same. Challenge energizes us psychologically and physically, and it motivates us to learn new skills and master our Occupations.

When a challenge is met, we feel relaxed and satisfied (NIOSH,1999). Thus, challenge is an important ingredient for healthy and productive work. The importance of challenge in our work lives is probably what people are referring to when they say, "a little bit of stress is good for you.

Occupational stress is that which derives specifically from conditions in the work place. These may either cause stress initially or aggravate the stress already present from other sources.

In today's typical workplace, stress is seen as becoming increasingly more common.

People appear to be working longer hours, taking on higher level of responsibilities and exerting themselves even more strenuously to meet rising expectations about Occupational performance. Competition is sharp.

There is always someone else ready to "step into one's shoes" should one be found wanting.

**13. Explain the different ways of Decision making.**
**(Nov/Dec 2012)(May/Jun 2013)(Apr 2014)**

**Methods of Decision Making are 4, they are:**

According to Patterson, Grenny, McMillan, and Switzler, there's four common ways of making decisions:

- **Command** – Decisions are made with no involvement.
- **Consult** – invite input from others.
- **Vote** – discuss options and then call for a vote.
- **Consensus** – talk until everyone agrees to one decision.