"Software Project Management"

1. **Define software project management?**
   Software project management is the art and science of planning and leading software projects. It is a sub-discipline of project management.
2. **What is RMMM?**
   RMMM stands for Risk Mitigation Monitoring and Management. The goal of RMMM is to identify as many potential risks as possible.
3. **Differentiate between Project, Process, Product, and portfolio?**
   Project: A project is a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end.
4. **Process:** A process is a set of interrelated actions and active performed to create a pre-specified product, service, or Result. Each process is characterized by its inputs, the tools and techniques that can be applied, and the resulting outputs.
   Product: An artifact that is produced, is quantifiable, and can be either an end item in itself or a component item.
   Additional words for products are material and goods. Contrast with *result*. See also *deliverable.*
5. **Define quantitative approach?**

6. **Differentiate between Schedule variance and Cost variance?**

Schedule variance	Cost variance
Schedule variance (SV) is a measure of schedule performance expressed as the difference between the earned value and the planned value. It is the amount by which the project is ahead or behind the planned delivery date, at a given point in time. It is a measure of schedule performance on a project. It is equal to the earned value (EV) minus the planned value (PV).	Cost variance (CV) is the amount of budget deficit or surplus at a given point in time, expressed as the difference between earned value and the actual cost. It is a measure of cost performance on a project. It is equal to the earned value (EV) minus the actual cost (AC).

7. **What is Portfolio management?**
   Portfolio management refers to the centralized management of one or more portfolios to achieve strategic objectives. Portfolio management focuses on ensuring that projects and programs are reviewed to prioritize resource allocation, and that the management of the portfolio is consistent with and aligned to organizational strategies.
8. **What is the basic structure of the V-model?**
   The V-Model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.
9. **List different stages of project life cycle?**
   - Initiation
   - Planning
   - Execution
   - Closure

10. What are the different types of contract?

- Fixed-price contracts
- cost-reimbursable contracts
- time and Material contracts (T&M)
- Cost Plus Fixed Fee Contracts (CPFF)
- Cost Plus Incentive Fee Contracts (CPIF)
- Cost plus Award Fee Contracts (CPAF).

11. What is Brainstorming?

A general data gathering and creativity technique that can be used to identify risks, ideas, or solutions to issues by using a group of team members or subject matter experts.

12. Define RAG?

In project management, RAG stands for Red Amber Green and relates to project status reporting which is utilized by project manager to indicate how well a certain project is performing.

13. What is objective driven or product driven?

In object driven projects the main objective of the final outcome is considered. But doesn't take much effort to build the finalized fully functioning expected version at the initial iteration.

14. Explain the dependency diagram?

The dependency diagram is built using the lowest level of decomposition in the work breakdown structure (WBS).

15. Define project success and project failure?

Project success: The achievement of something desired, planned or attempted. It is also said that success is an event that accomplishes its intended purpose.

Project failure: A project that fails to perform a duty or an expected action, non-occurrence or non-performance.

16. What is management? And also define management control?

Management: Management is a set of principles relating to the functions of planning, organization, directing, and controlling, and the application of these principles in harnessing physical, financial, human, and informational resources efficiently and effectively to achieve organizational goals.

Management control: Management control can be defined as a systematic effort by business management to compare performance to predetermined standards, plans, or objectives in order to determine whether performance is in line with these standards and presumable in order to take any remedial action required to see that human and other.

17. Define term payback period?

Payback period refers to the amount of time it takes to recover the cost of an investment. **OR** Payback period is the length of time an investment reaches a break even point.

18. What is IRR? How we can calculate PV?

IRR stands for internal rate of return. The internal rate of return of a project is the expected growth rate of a project investment. Organizations typically calculate IRR to make decision between several investment alternatives.

PV: PV stands for planned value. The formula to calculate PV is simple. Multiply the planed percentage of the completed work by the project budget.

19. What is cash flow forecasting?

The cash flow forecasting is a financial planning tools that shows the predicted flow of cash in and out of project or organization each month. Forecasting will enable you to plan ahead so that you can anticipate periods of cash shortage and take corrective action.

20. **Define version control?**
    Version control is a method of tracking changes to document and files so that you can always know which version is the current iteration. Projects typically result in the creation of a lot of documents, from project reports to deliverables.
21. **What is project mitigation?**
    Mitigation is a strategic risk response wherein a project team takes active steps to reduce the probability or impact of a negative risk to a project.
22. **What is resource scheduling?**
    Resource scheduling refers to the set of actions and methodology used by organizations to efficiently assign the resources they have to jobs, tasks or project they need to complete, and schedule start and end dates for each tasks or project based on resource availability.
23. **Define statement of work?**
    A statement of work (SOW) is a document routinely employed in the field of project management. It defines project-specific activities, deliverables and timelines for a vendor providing services to the client.
24. **Write name for any two resource scheduling techniques?**
    I. Critical path method (CPM)
    II. Program evaluation review technique (PERT)
    You can use these methods to calculate the assumed start and finish dates, based on the known scope of the project.
25. **How we find the time estimation for a project?**
    Step 1: Understand what's required. Start by identifying all of the work that needs to be done within the project.
    Step 2: Order these activities. Now, list all of the activities you identified in the order in which they need to happen.
26. **What is incremental model?**
    The incremental model is a method of software development where the product is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance.
27. **What are stake holders?**
    The individuals whose interest may be positively or negatively affected as a result of project execution or successful project completion.
28. **Define activity?**
    A distinct, scheduled portion of work performed during the course of a project is called activity.
29. **Write name for any two software process model?**
    - Waterfall model
    - V model
    - RAD model
    - Spiral model
    - Agile model

30. **How we control the quality of project?**

    10 ways to maintain the quality of a project:

    i. Define quality
    ii. Commit to quality
    iii. Stick to project requirements
    iv. Manage quality
    v. Perform quality assurance
    vi. Control the quality
    vii. Focus on requirement
    viii. Follow project processes

31. **What is extreme programming?**

    Extreme programming (XP) is an agile project management framework used in software development. It prescribes everything, from how to organize projects and develop software, to how to increase developer's productivity and what's the best way to collaborate on code.

32. **What are software metrics?**

    A software metric is a standard of measure of a degree to which a software system or process possesses some property.

33. **What do you mean by qualitative risk analysis?**

    Qualitative risk analysis is a technique used to quantify risk associated with a particular hazard. Risk assessment is used for uncertain events that could have many outcomes and for which there could be significant consequences.

34. **Explain net present value?**

    In finance, the net present value applies to a series of cash flows occurring at different times. The present value of a cash flow depends on the interval of time between now and the cash flow. It also depends on the discount rate. NPV for the time value of money.

35. **Why decision tree is used?**

    The decision tree shows how to make a decision between alternative capital strategies (represented as "decision
    nodes") when the environment contains uncertain elements (represented as "chance nodes").

36. **What do you mean by scope creep?**

    The uncontrolled expansion to product or project scope without adjustments to time, cost, and resources is called scope creep.

37. **What are positive risks in any IT project?**

    Positive risk is any condition, event, occurrence or situation that provides a possible positive impact for a project or environment. A positive risk element can positively affect your project and its objectives.

38. **What do you mean by software release management?**

    Release management is a process that entails the management, planning, scheduling, and controlling of an entire software build through every stage and environment involved, including testing and deploying software releases.

39. **What is program management?**

    The application of knowledge, skills, tools, and techniques to a program to meet the program requirements and to obtain benefits and control not available by managing projects individually.

40. **What is procurement management?**

The documents utilized in bid and proposal activities, which include the buyer’s Invitation for Bid, Invitation for Negotiations, Request for Information, Request for Quotation, Request for Proposal, and seller’s responses.

41. What is critical path?

The critical path is the sequence of activities that represents the longest path through a project, which determines the shortest possible project duration.

42. What is project outsourcing?

Outsourcing allows a company to subcontract a particular area within the organization. A company may outsource project management or any other task or department for one or more reason.

43. What is resource acquisition?

Resource acquisition focuses on defining the needs for the project, and obtaining the right resources for the team and other resources and tools available to manage the effort.

44. Define PERT?

PERT stands for Program Evaluation Review Technique breaks down the individuals tasks of a project for analysis.

45. Define project charter?

The project charter establishes a partnership between the performing and requesting organizations .It is used to establish internal agreements within an organization to assure proper delivery under the contract. The approved project charter formally initiates the project.

46. Explain business case? Write four dimension of a project?

The business case or similar document describes the necessary information from a business standpoint to determine whether or not the project is worth the required investment. It is commonly used for decision making by managers or executives above the project level.

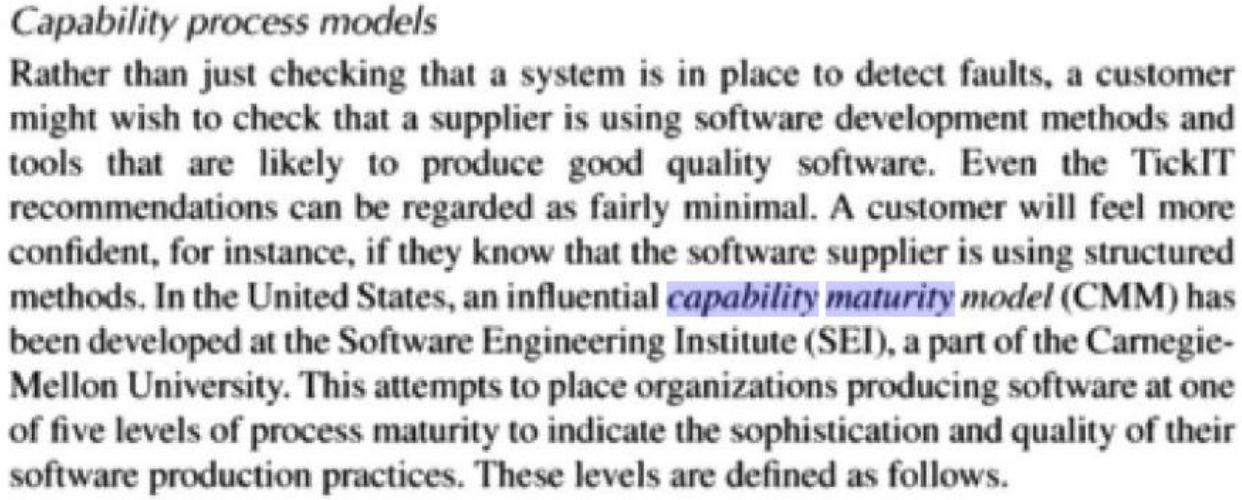
i. Project Objectives. Having a set of objectives, expectations of the outcomes is why projects are developed, planned, funded, and executed. Being explicit about the objectives, and having everyone involved, and who may be affected, is essential.

ii. Project Scope. The scope describes what will and will not be done as a part of the project. Failing to have an explicit scope encourages ‘project creep’ and lessens the accountability. In the ERP implementations I have been involved in, project creep is an ever present cancer on the project, and those that failed to be absolutely explicit about the scope, and enforced it ruthlessly, failed to meet expectations in numerous ways.

iii. Project Budget. How much the project is expected to cost. Pretty basic, but ignored often, and subject to blow-out as the scope creeps out of control. The only ones who benefit are the consultants who either fix the problems (often they are a part of the problem) and your competitors.

iv. Project Timetable. Every project needs a timetable, with milestones connected to the scope and costs, as well as performance.

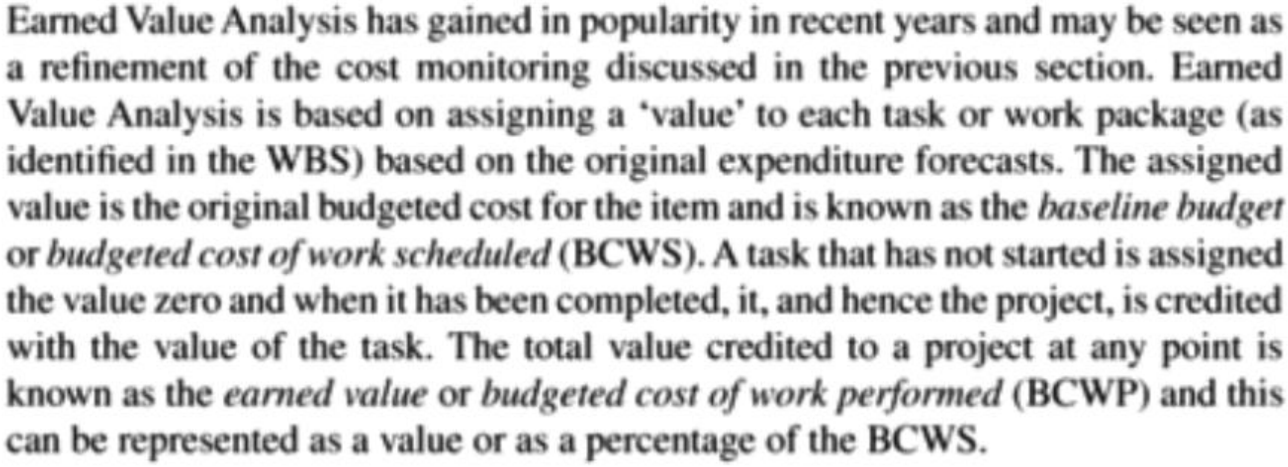
47. What is CMMI?



Capability process models

Rather than just checking that a system is in place to detect faults, a customer might wish to check that a supplier is using software development methods and tools that are likely to produce good quality software. Even the TickIT recommendations can be regarded as fairly minimal. A customer will feel more confident, for instance, if they know that the software supplier is using structured methods. In the United States, an influential *capability maturity model* (CMM) has been developed at the Software Engineering Institute (SEI), a part of the Carnegie-Mellon University. This attempts to place organizations producing software at one of five levels of process maturity to indicate the sophistication and quality of their software production practices. These levels are defined as follows.

48. What is earned value analysis?



Earned Value Analysis has gained in popularity in recent years and may be seen as a refinement of the cost monitoring discussed in the previous section. Earned Value Analysis is based on assigning a 'value' to each task or work package (as identified in the WBS) based on the original expenditure forecasts. The assigned value is the original budgeted cost for the item and is known as the *baseline budget* or *budgeted cost of work scheduled* (BCWS). A task that has not started is assigned the value zero and when it has been completed, it, and hence the project, is credited with the value of the task. The total value credited to a project at any point is known as the *earned value* or *budgeted cost of work performed* (BCWP) and this can be represented as a value or as a percentage of the BCWS.

49. What do you mean by team development?

Teams are becoming a key tool for organizing work in today's corporate world. Teams have the potential to immediately amass, organize, relocate, and disperse. But, teams are an effective tool of employee motivation. It is essential to consider the fact that teams develop and get mature over a period of time. Team development creates a captivating atmosphere by encouraging co-operation, teamwork, interdependence and by building trust among team members.

50. How a project can be success and fail?

A project's success or failure depends on following factors:

Success factors:

1 User involvement

2 Executive management support

3 Clear statement of requirements

4 Proper planning

Failure factors:

1 Planning and Estimation factor

2 Implementation factor

3 Human factor

51. Differentiate between project monitoring and project controlling?

Project Monitoring refers to collecting, recording, and reporting information concerning project performance that project manager and others wish to know.

Project Controlling uses data from monitor activity to bring actual performance to planned performance.

52. **What do you mean by software configuration management?**

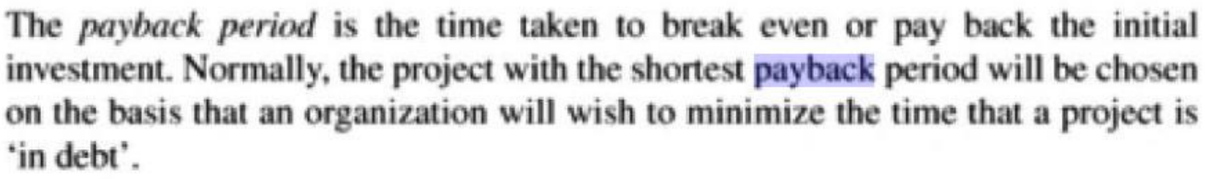
Software Configuration Management(SCM) is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle. The primary goal is to increase productivity with minimal mistakes. SCM is part of cross-disciplinary field of configuration management and it can accurately determine who made which revision.

53. **What do you mean by resource planning in software in deployment project of any software?**

Resource planning is a process of allocating tasks to human and non-human resources so that they're maximized for efficiency. Resource planning helps project managers manage resource utilization and track resource capacity, to keep projects on budget.

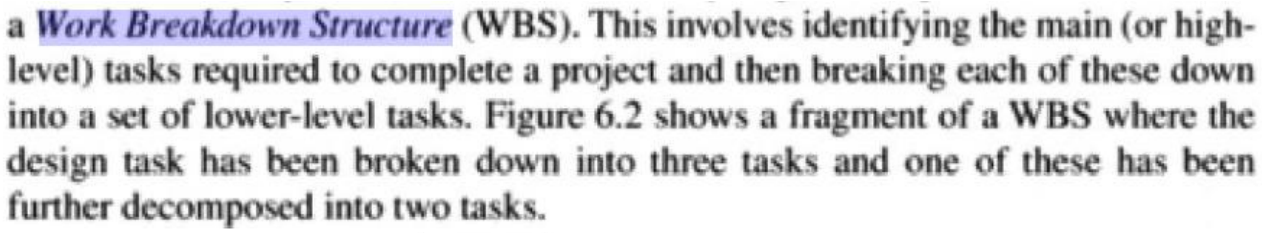
Resource planning helps you to organize your team so that they know exactly what projects they're working on. More importantly a resource planner that accurately tracks and manages your resource capacity ensures that you're managing your team's time effectively and not burning anybody out.

54. **What is payback period with reference to cost benefit analysis?**



The *payback period* is the time taken to break even or pay back the initial investment. Normally, the project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in debt'.

55. **What do you mean by work breakdown structure?**



a *Work Breakdown Structure* (WBS). This involves identifying the main (or high-level) tasks required to complete a project and then breaking each of these down into a set of lower-level tasks. Figure 6.2 shows a fragment of a WBS where the design task has been broken down into three tasks and one of these has been further decomposed into two tasks.

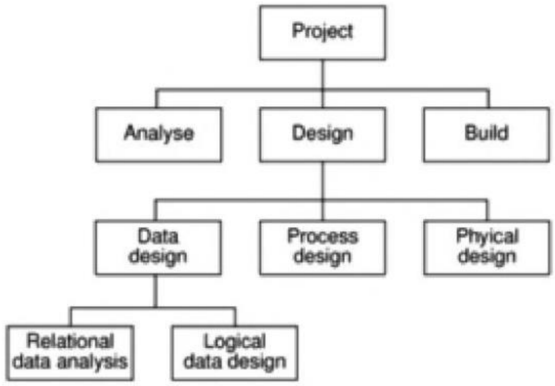


Figure 6.2 *A fragment of an activity-based Work Breakdown Structure.*

56. **What is project portfolio management?**

Project portfolio management (PPM) refers to a process used by project managers and project management organizations (PMOs) to analyze the potential return on undertaking a project. By organizing and consolidating every piece of data regarding proposed and current projects, project portfolio managers provide forecasting and business analysis for companies looking to invest in new projects.

Project portfolio management gives organizations and managers the ability to see the big picture.

- Executives – know what project managers to reach
- Project Managers – easy access to team members
- Team Members – improved communication with leadership and other teammates
- Stakeholders – kept in the loop with reliable and consistent feedback

57. Write down different techniques used for time scheduling?

Time scheduling is a collection of techniques used to develop and present schedules that show when work will be performed. The results of all these techniques are usually presented as activities or bars on a timeline, known as a Gantt chart.

Scheduling comes into project planning and there are principally two types of scheduling: critical path and critical chain.

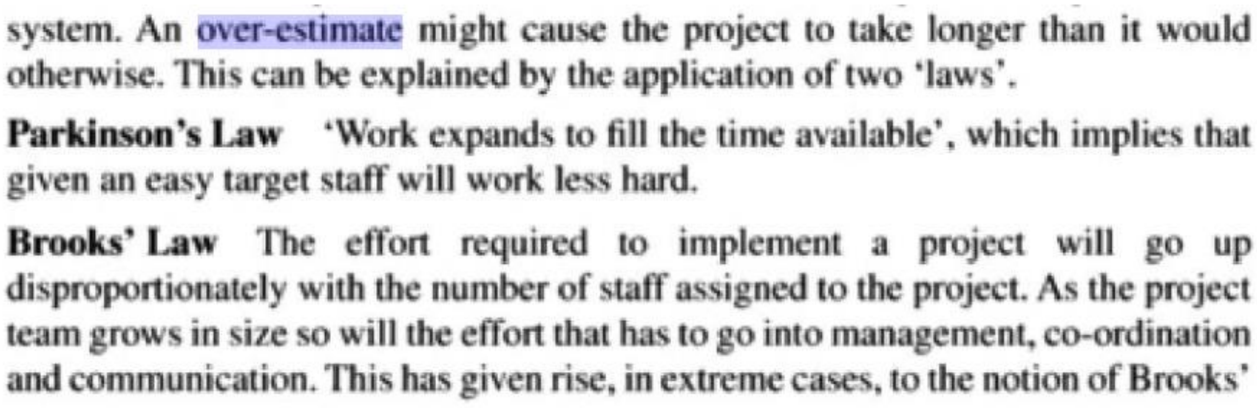
The **critical path** approach places emphasis on the activities in a project and understanding the shortest time to complete all activities in a logical order. It is a sequence of activities through a precedence network from start to finish, the sum of whose durations determines the overall duration.

Critical path analysis is an activity- based scheduling technique that determines the overall duration of the identified work based on estimates and logical dependencies.

The **critical chain** is resource based approach to scheduling, useful when time is critical. It is derived from the critical path that protects critical chains of activities with buffers. The critical chain approach attempts to keep resources at a constant utilisation, avoiding common working practices such as:

- multitasking between activities;
- not starting planned work at the earliest start date and committing time until it is finished.

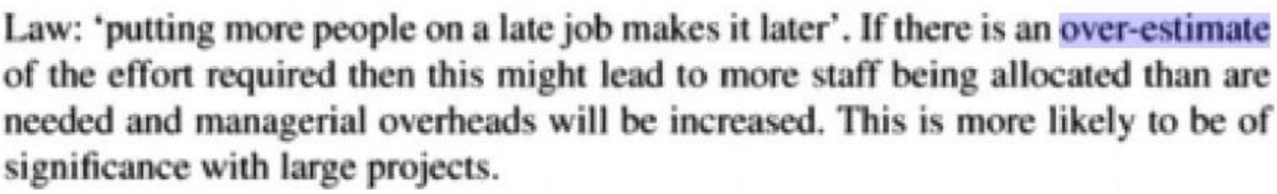
58. Write two problem of over estimates?



system. An over-estimate might cause the project to take longer than it would otherwise. This can be explained by the application of two 'laws'.

Parkinson's Law 'Work expands to fill the time available', which implies that given an easy target staff will work less hard.

Brooks' Law The effort required to implement a project will go up disproportionately with the number of staff assigned to the project. As the project team grows in size so will the effort that has to go into management, co-ordination and communication. This has given rise, in extreme cases, to the notion of Brooks'



Law: 'putting more people on a late job makes it later'. If there is an over-estimate of the effort required then this might lead to more staff being allocated than are needed and managerial overheads will be increased. This is more likely to be of significance with large projects.

59. Write names down any two mistakes related to project development?

Some of the more significant mistakes along the development-speed dimensions of people, process, product, and technology are as follows.

1: Undermined motivation

2: Weak personnel.

3: Uncontrolled problem employees

5: Adding people to a late project

6: Noisy, crowded offices
7: Contractor failure
8: Insufficient planning
9: Wasted time during the fuzzy front end

60. Define backward pass?

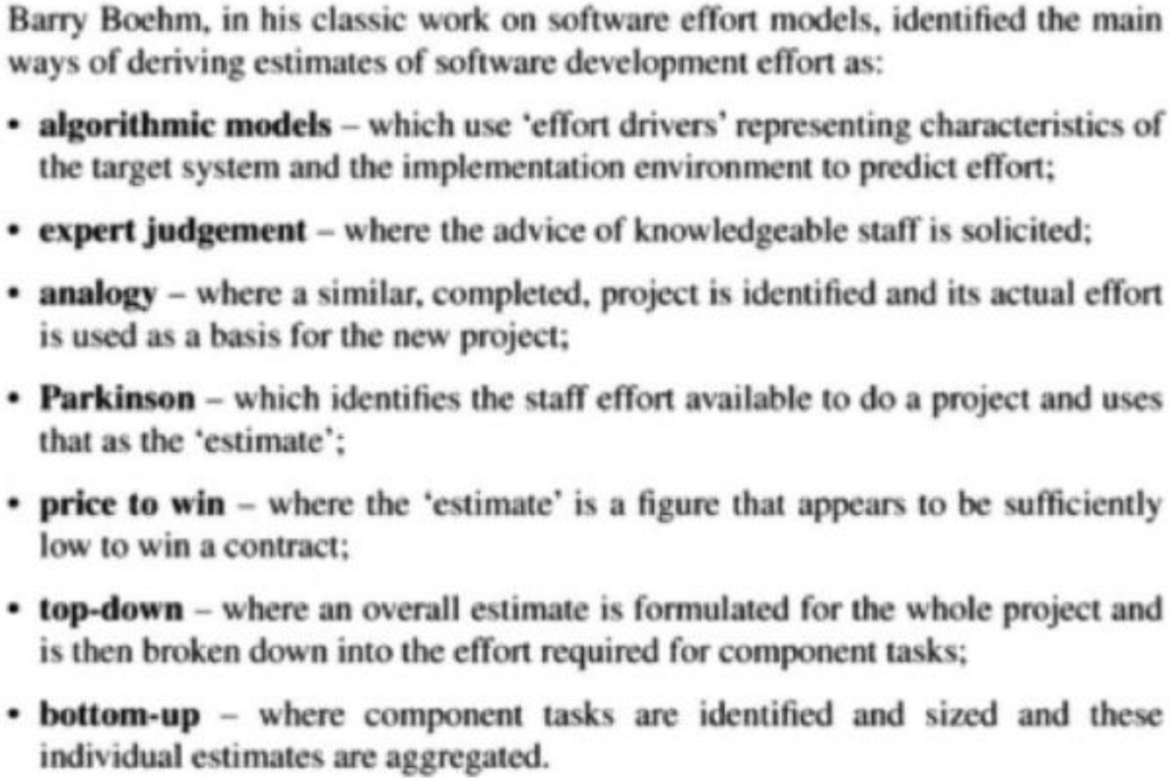
The backward pass starts from the right side of the network diagram and proceeds to the left. It determines the Latest Start (LF) and Latest Finish (LF) of each task. These are defined as follows:

- Latest Start (LS): The latest date that the task can start.
- Latest Finish (LF): The lates date that the task can finish.

61. What is Quadruple constraints?

In project management, there is a well-known theory called Triple Constraint, but many people think that it should add one more constraint to be the Quadruple Constraint. Quadruple Constraint consists of SCOPE, SCHEDULE, COST and QUALITY. The last one which make the Triple Constraint to become the Quadruple Constraint.

62. Write names of any two software estimation techniques?



Barry Boehm, in his classic work on software effort models, identified the main ways of deriving estimates of software development effort as:

- **algorithmic models** – which use 'effort drivers' representing characteristics of the target system and the implementation environment to predict effort;
- **expert judgement** – where the advice of knowledgeable staff is solicited;
- **analogy** – where a similar, completed, project is identified and its actual effort is used as a basis for the new project;
- **Parkinson** – which identifies the staff effort available to do a project and uses that as the 'estimate';
- **price to win** – where the 'estimate' is a figure that appears to be sufficiently low to win a contract;
- **top-down** – where an overall estimate is formulated for the whole project and is then broken down into the effort required for component tasks;
- **bottom-up** – where component tasks are identified and sized and these individual estimates are aggregated.

63. What is agile process modeling?

Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

64. What is procurement management?

Project procurement management is a section of the Implementation Plan to determine how "the ordered products necessary for producing deliverables can be delivered on time and within the allocated budget".

Project Procurement Management is part of the project management process in which products or services are aquired or purchased from outside the existing employee base (which would work on the project) in order to complete the task or project. There are essentially two different types of procurements, one in which the company is responsible for the particular product or service under a legal contract, this PPM includes contract management responsibilities that issue specific tasks to various team members. Both of these project management processes are imperative to a company's success.

65. **What is the difference between testing and debugging?**

Testing:

Testing is the process of verifying and validating that a software or application is bug free, meets the technical requirements as guided by its design and development and meets the user requirements effectively and efficiently with handling all the exceptional and boundary cases.

Debugging:

Debugging is the process of fixing a bug in the software. It can defined as the identifying, analyzing and removing errors. This activity begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software. It is considered to be an extremely complex and tedious task because errors need to be resolved at all stages of debugging.

66. **Define preliminary investigation?**

Preliminary investigation is the first phase. In this phase, the system is investigated. The objective of this phase is to conduct an initial analysis and findings of the system.

67. **Define change control?**

Change Control is focused on identifying, documenting and controlling changes to the project and the project baselines. In the change management system, you manage the changes related to the project scope, planning, and baselines.

68. **What is Gantt Chart?**

A Gantt chart can be helpful to visualize the project timeline and whether they are tracking to the proper constraints.

69. **Write name of any two cost benefit analysis techniques?**

i. Risk profile analysis

ii. Using decision trees

70. **Differentiate between Schedule Variance and Cost Variance?**

Schedule variance	Cost variance
Schedule variance (SV) is a measure of schedule performance expressed as the difference between the earned value and the planned value. It is the amount by which the project is ahead or behind the planned delivery date, at a given point in time. It is a measure of schedule performance on a project. It is equal to the earned value (EV) minus the planned value (PV).	Cost variance (CV) is the amount of budget deficit or surplus at a given point in time, expressed as the difference between earned value and the actual cost. It is a measure of cost performance on a project. It is equal to the earned value (EV) minus the actual cost (AC).

71. **Define the term “Statement of work”?**
    A statement of work (SOW) is a document routinely employed in the field of project management. It defines project-specific activities, deliverables and timelines for a vendor providing services to the client.
72. **What is the difference between BCWP and BCWS?**
    Budgeted Cost of Work Performed (BCWP) is the budgeted cost of the value of work that has actually been accomplished or completed to date. It can be used to address the entire project, individual task or work packages. It’s compared against Actual Cost of Work Performed (ACWP). BCWP is a tool used in Earn Value Management (EVM) and is also called Earned Value.
    Budgeted Cost of Work Scheduled (BCWS) is the sum of the budgets for all work scheduled to be accomplished with a given time period. It also includes the cost of previous work completed and can address a specific period of performance or a date in time.
73. **How can slack be negative?**
    Negative slack indicates that there is not enough time scheduled for the task and is usually caused by constraint dates.
74. **What is quantitative risk analysis?**
    The qualitative risk analysis is the process of numerically analyzing the effect of identified risks on overall project objectives.
75. **Write names of any two project related risks?**
    Cost risk, typically escalation of project costs due to poor cost estimating accuracy and scope creep.
    Schedule risk, the risk that activities will take longer than expected. Slippages in schedule typically increase costs and, also, delay the receipt of project benefits, with a possible loss of competitive advantage.
    Performance risk, the risk that the project will fail to produce results consistent with project specifications.
76. **What is configuration management tool?**
    Configuration Management is a subset of systems management. Configuration management tools perform various roles to ensure consistency among physical and logical assets. These tools identify and track configuration items and document functional dependencies. These tools are invaluable for understanding the impact of changing one configuration item on all the others. Configuration management data is usually stored in a configuration management database.
77. **Write down the name of any framework used for process improvement in information technology?**
78. **What are risk involving in project estimation?**

“Subjective”

1. **Write a detailed note on Step Wise project planning with diagram?**

- **Step 0: Select project**
- **Step 1: Identify project scope and objectives**
- **Step 2: Identify project infrastructure**
- **Step 3: Analyse project characteristics**
- **Step 4: Identify project products and activities**
- **Step 5: Estimate effort for each activity**
- **Step 6: Identify activity risk**
- **Step 7: Allocate resources**
- **Step 8: Review/publicize plan**
- **Step 9 and 10: Execute plan and lower level of planning**

� Step 0 : Select project:

This is called step 0 because in a way of project planning, it is outside the main project planning process. Possibility study suggests us that the project is worthwhile or not.

� Step 1 : Identify project scope and objectives:

The activities in this step ensure that all parties to the project agree on the objectives and are dedicated to the success of the project.

1: Identify objectives and practical measures of the effectiveness in meeting those objectives

2: Establish project authority.

3: Stakeholders analysis – Identify all stakeholders in the project and their interest.

4: Modify objectives in the light of stakeholder analysis.

5: Establish method of communication

� Step 2 : Identify project Infrastructure:

Projects are rarely carried out in a vacuum. There is usually some kind of infrastructure into which the project must fit. Where the project managers are new to the organization, they must find out the precise nature of this infrastructure.

1: Identify relationship between the project and strategic planning

2: Identify installation standards and procedures.

3: Identify project team organization.

� Step 3: Analyze project Characteristics:

The general purpose of this part of planning operation is to ensure that the appropriate methods are used for the project.

1: Distinguish the project as either objective- product driven

2: Analyze other project characteristics (including quality –based ones)

3: Identify high level project risks

3: Take into account user requirement concerning implementation.

4: Select development methodology and life cycle approach.

5: Review overall resources estimates

� Step 4 : Identify project products and activities:

The more detailed planning of the individual activities now takes place. The longer term planning is broad and in outline, while the more immediate tasks are planned in some detail.

1: Identify and describes project products (or deliverables)

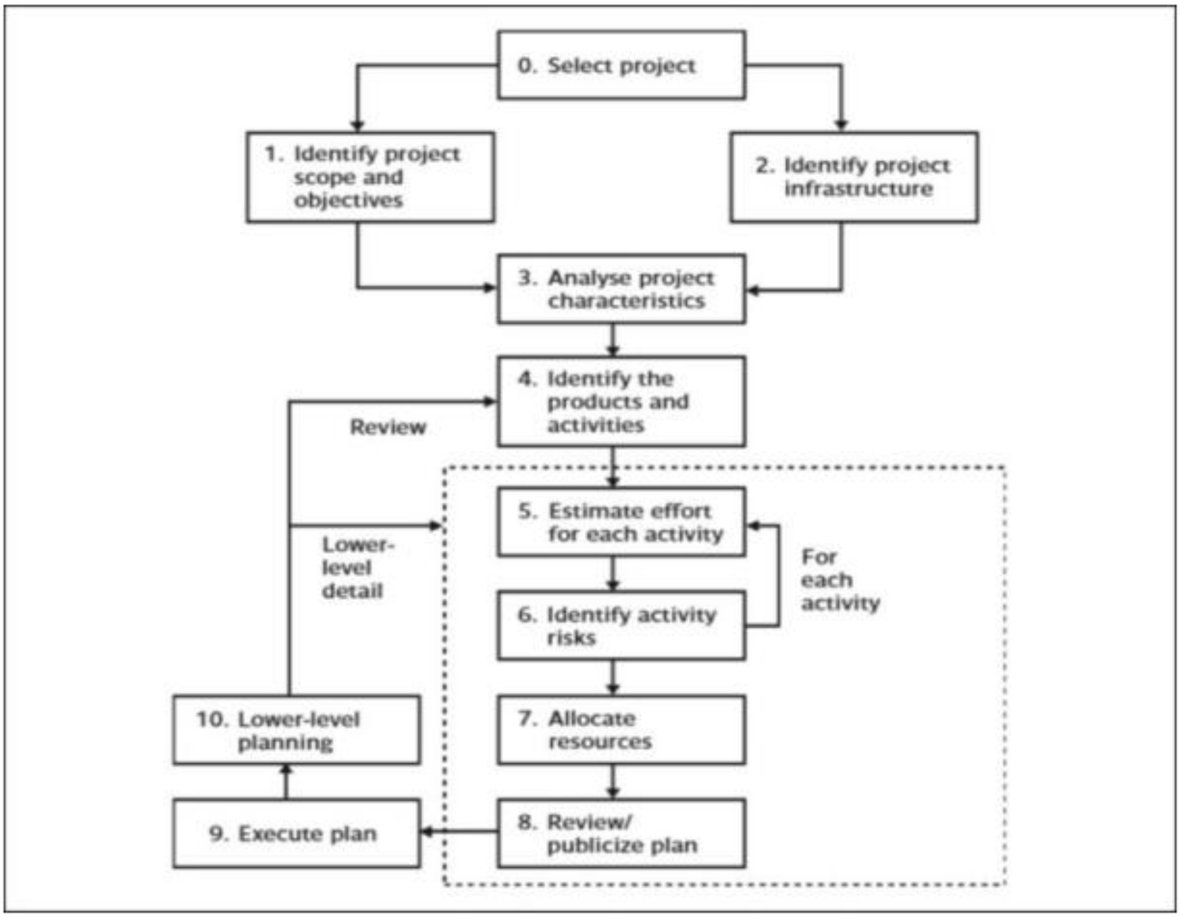
2: Document generic product flows

3: Record product instance

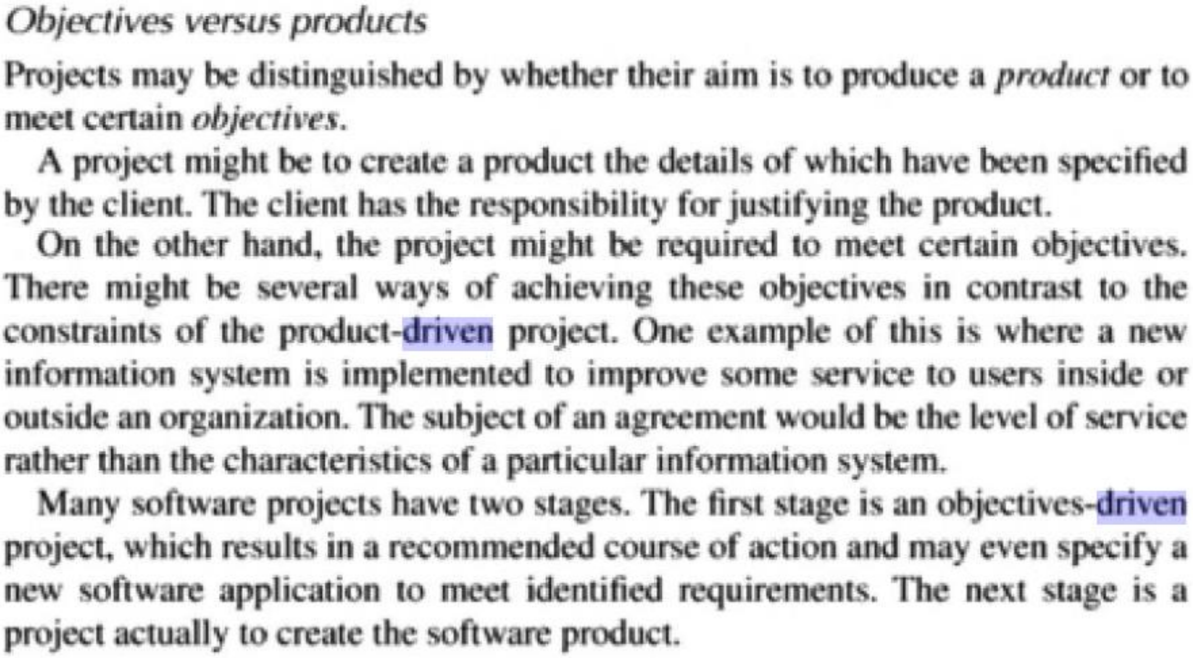
4: produce ideal activity network

5: Modify the ideal to take into account need for stages and checkpoints.

Step 5: Estimate effort for each activity:
1: Carry out bottom-up estimates
2: Revise plan to create controllable activities.
Step 6: Identify activity risks:
1: Identify and quantify activity based risks
2: Plan risk reduction and contingency measures where appropriate
3: Adjust overall plans and estimates to take account of the risks
Step 7: Allocate resources:
1: Identify and allocate resources
2: Revise plans and estimates to take into account resource constraints
Step 8: Review / Publicize plan:
1: Review quality aspects of the project plan.
2: Document plans and obtain agreement.
Step 9 & 10: Execute plan / lower level of planning:
Once the project is started, plans will need to be drawn up in greater detail for each activity as it becomes due. Detailed and lower level of planning of the soon stages will need to be delayed because more information will be available nearer the start of the stage.


0. Select project
1. Identify project scope and objectives
2. Identify project infrastructure
3. Analyse project characteristics
4. Identify the products and activities
Review
Lower-level detail
5. Estimate effort for each activity
For each activity
6. Identify activity risks
7. Allocate resources
10. Lower-level planning
9. Execute plan
8. Review/ publicize plan

2. Explain object driven and product driven?



Objectives versus products

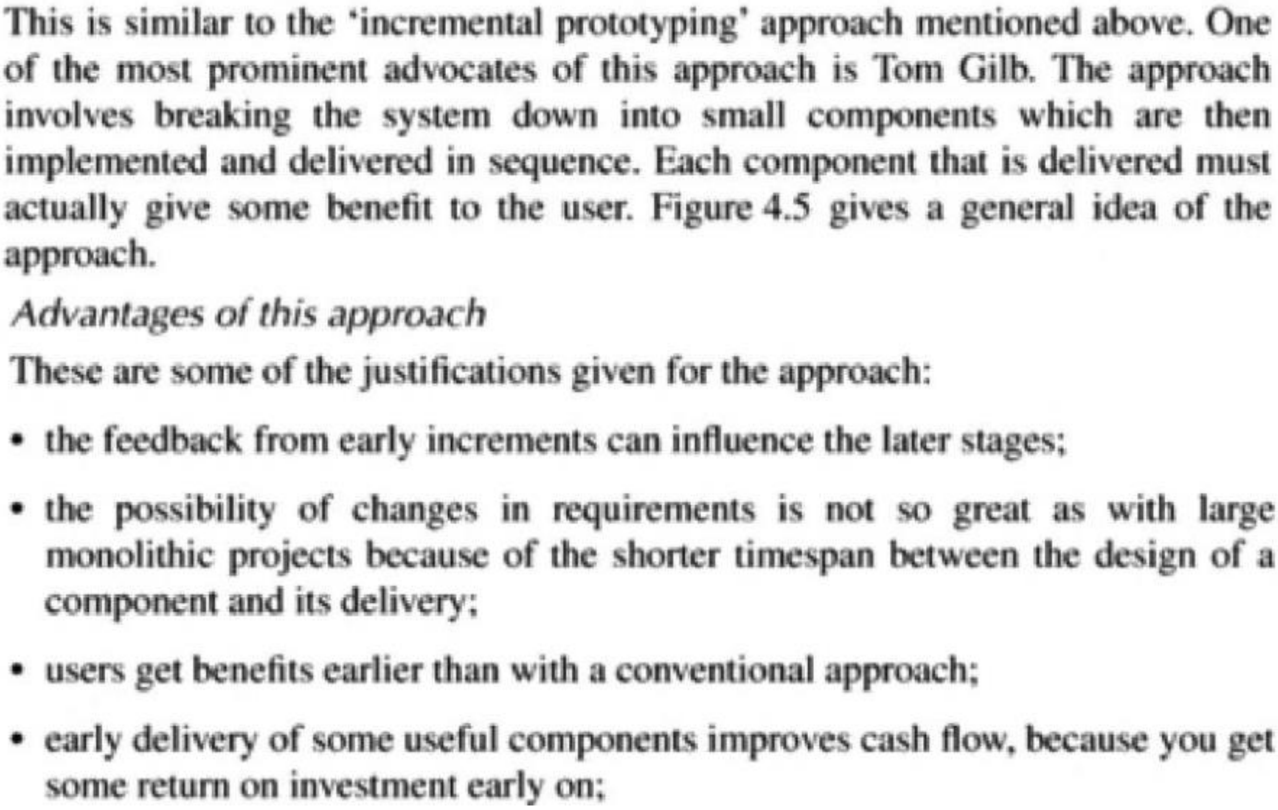
Projects may be distinguished by whether their aim is to produce a *product* or to meet certain *objectives*.

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.

On the other hand, the project might be required to meet certain objectives. There might be several ways of achieving these objectives in contrast to the constraints of the product-driven project. One example of this is where a new information system is implemented to improve some service to users inside or outside an organization. The subject of an agreement would be the level of service rather than the characteristics of a particular information system.

Many software projects have two stages. The first stage is an objectives-driven project, which results in a recommended course of action and may even specify a new software application to meet identified requirements. The next stage is a project actually to create the software product.

3. Advantages and disadvantages of incremental delivery?

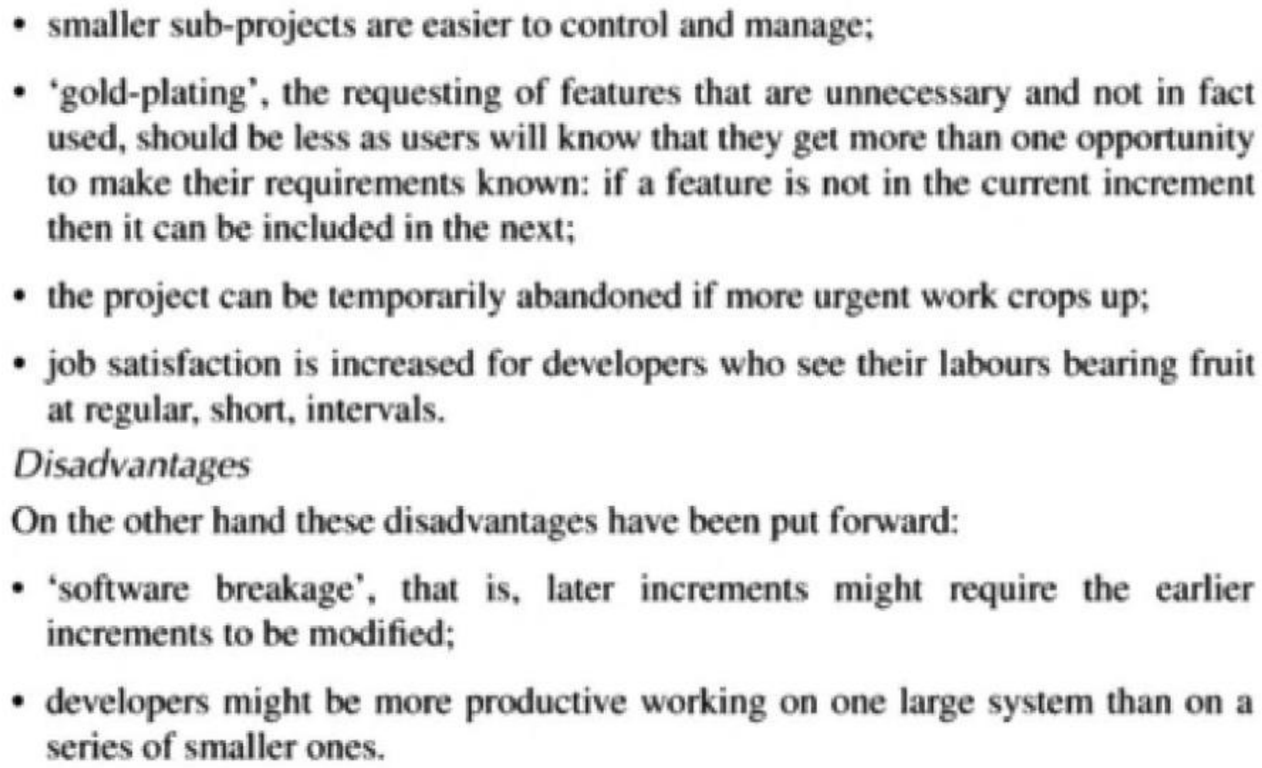


This is similar to the 'incremental prototyping' approach mentioned above. One of the most prominent advocates of this approach is Tom Gilb. The approach involves breaking the system down into small components which are then implemented and delivered in sequence. Each component that is delivered must actually give some benefit to the user. Figure 4.5 gives a general idea of the approach.

Advantages of this approach

These are some of the justifications given for the approach:

- the feedback from early increments can influence the later stages;
- the possibility of changes in requirements is not so great as with large monolithic projects because of the shorter timespan between the design of a component and its delivery;
- users get benefits earlier than with a conventional approach;
- early delivery of some useful components improves cash flow, because you get some return on investment early on;



- smaller sub-projects are easier to control and manage;
- 'gold-plating', the requesting of features that are unnecessary and not in fact used, should be less as users will know that they get more than one opportunity to make their requirements known: if a feature is not in the current increment then it can be included in the next;
- the project can be temporarily abandoned if more urgent work crops up;
- job satisfaction is increased for developers who see their labours bearing fruit at regular, short, intervals.

Disadvantages

On the other hand these disadvantages have been put forward:

- 'software breakage', that is, later increments might require the earlier increments to be modified;
- developers might be more productive working on one large system than on a series of smaller ones.

4. **How to define project scope? Explain project activities and work break down structure?**

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, features, functions, tasks, deadlines, and ultimately costs. In other words, it is what needs to be achieved and the work that must be done to deliver a project. Below is an overview of some of the key processes to follow in order to define scope correctly.

- Define the Product Requirements:
- Define the Process Requirements:
- Involve the correct stakeholders:
- Identify the limitations
- Change Management

<u>Activities:</u>

1. Project Planning: It is a set of multiple processes, or we can say that it a task that performed before the construction of the product starts.

2. Scope Management: It describes the scope of the project. Scope management is important because it clearly defines what would do and what would not. Scope Management create the project to contain restricted and quantitative tasks, which may merely be documented and successively avoids price and time overrun.

3. Estimation management: This is not only about cost estimation because whenever we start to develop software, but we also figure out their size(line of code), efforts, time as well as cost.

4. Scheduling Management: Scheduling Management in software refers to all the activities to complete in the specified order and within time slotted to each activity. Project managers define multiple tasks and arrange them keeping various factors in mind.

For scheduling, it is compulsory -

- Find out multiple tasks and correlate them.
- Divide time into units.
- Assign the respective number of work-units for every job.
- Calculate the total time from start to finish.
- Break down the project into modules.

5. Project Resource Management: In software Development, all the elements are referred to as resources for the project. It can be a human resource, productive tools, and libraries.
Resource management includes:

- Create a project team and assign responsibilities to every team member
- Developing a resource plan is derived from the project plan.
- Adjustment of resources.

6. Project Risk Management: Risk management consists of all the activities like identification, analyzing and preparing the plan for predictable and unpredictable risk in the project.
Several points show the risks in the project:

- The Experienced team leaves the project, and the new team joins it.
- Changes in requirement.
- Change in technologies and the environment.
- Market competition.

7. Project Communication Management: Communication is an essential factor in the success of the project. It is a bridge between client, organization, team members and as well as other stakeholders of the project such as hardware suppliers.

From the planning to closure, communication plays a vital role. In all the phases, communication must be clear and understood. Miscommunication can create a big blunder in the project.

8. Project Configuration Management: Configuration management is about to control the changes in software like requirements, design, and development of the product.

Work Breakdown Structure:

Dividing complex projects to simpler and manageable tasks is the process identified as Work Breakdown Structure (WBS).

Usually, the project managers use this method for simplifying the project execution. In WBS, much larger tasks are broken down to manageable chunks of work. These chunks can be easily supervised and estimated.

WBS is not restricted to a specific field when it comes to application. This methodology can be used for any type of project management.

Characteristics of the Work Breakdown Structure:

Not every breakdown of project deliverables can be classified as a WBS. For it to be called a work breakdown structure, it must have certain characteristics:

Hierarchy: The WBS is hierarchical in nature. Each “child” level exists in a strict hierarchical relationship with the parent level. The sum of all the child elements should give you the parent element.

100% rule: Every level of decomposition must make up 100% of the parent level. It should also have at least two child elements.

Mutually exclusive: All elements at a particular level in a WBS must be mutually exclusive. There must be no overlap in either their deliverables or their work. This is meant to reduce miscommunication and duplicate work.

Outcome-focused: The WBS must focus on the result of work, i.e. deliverables, rather than the activities necessary to get there. Every element should be described via nouns, not verbs. This is a big source of confusion for beginners to WBS.

Following are a few reasons for creating a WBS in a project:

- Accurate and readable project organization.

- Accurate assignment of responsibilities to the project team.
- Indicates the project milestones and control points.
- Helps to estimate the cost, time and risk.
- Illustrate the project scope, so the stakeholders can have a better understanding of the same.

Construction of a WBS:

Identifying the main deliverables of a project is the starting point for deriving a work breakdown structure.

This important step is usually done by the project managers and the subject matter experts (SMEs) involved in the project. Once this step is completed, the subject matter experts start breaking down the high-level tasks into smaller chunks of work.

In the process of breaking down the tasks, one can break them down into different levels of detail. One can detail a high-level task into ten sub-tasks while another can detail the same high-level task into 20 sub-tasks.

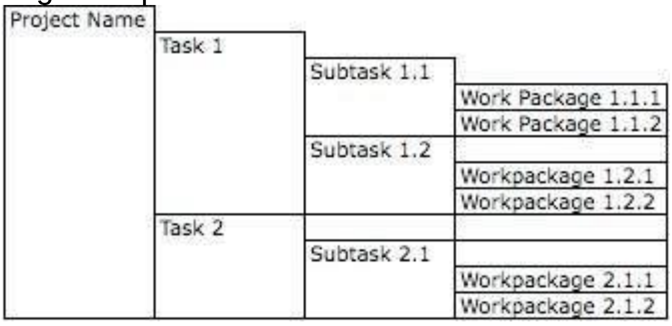
Therefore, there is no hard and fast rule on how you should breakdown a task in WBS. Rather, the level of breakdown is a matter of the project type and the management style followed for the project.

In general, there are a few "rules" used for determining the smallest task chunk. In "two weeks" rule, nothing is broken down smaller than two weeks worth of work.

This means, the smallest task of the WBS is at least two-week long. 8/80 is another rule used when creating a WBS. This rule implies that no task should be smaller than 8 hours of work and should not be larger than 80 hours of work.

One can use many forms to display their WBS. Some use tree structure to illustrate the WBS, while others use lists and tables. Outlining is one of the easiest ways of representing a WBS.

Following example is an outlined WBS:

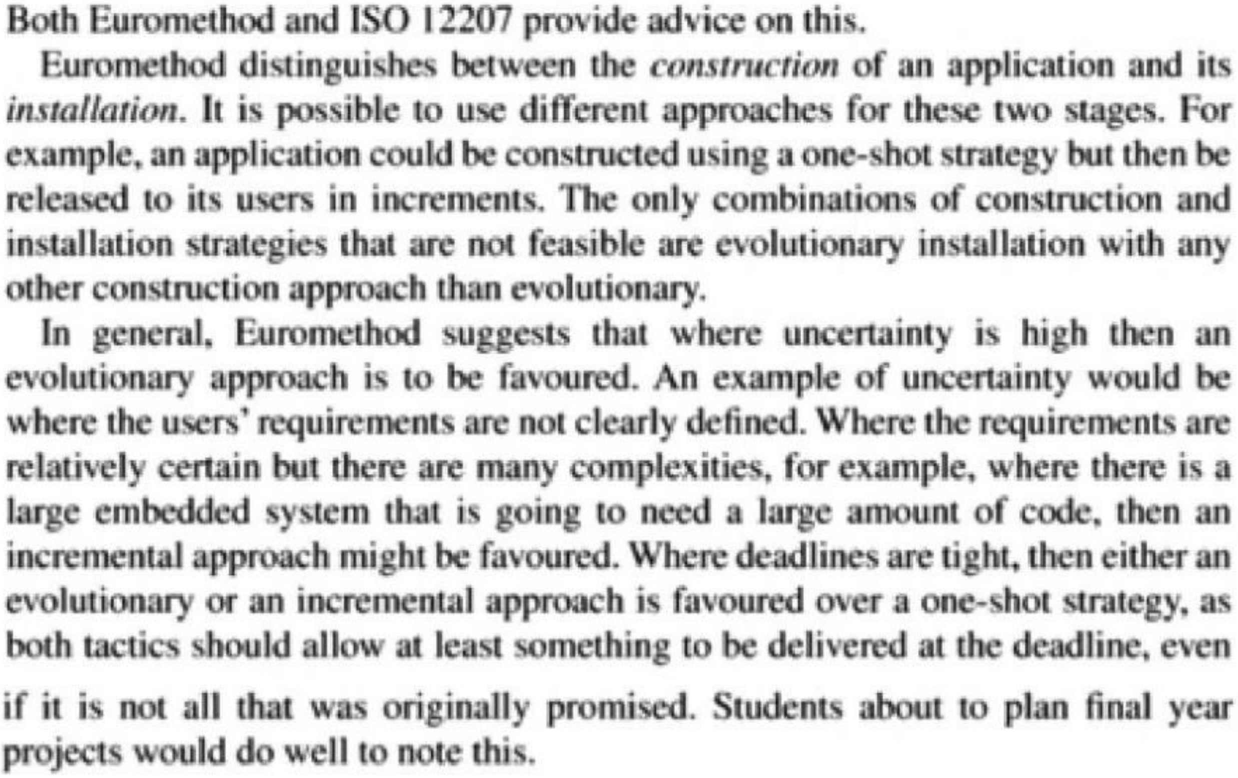


Project Name			
	Task 1		
		Subtask 1.1	
			Work Package 1.1.1
			Work Package 1.1.2
		Subtask 1.2	
			Workpackage 1.2.1
			Workpackage 1.2.2
	Task 2		
		Subtask 2.1	
			Workpackage 2.1.1
			Workpackage 2.1.2

There are many design goals for WBS. Some important goals are as follows:

- Giving visibility to important work efforts.
- Giving visibility to risky work efforts.
- Illustrate the correlation between the activities and deliverables.
- Show clear ownership by task leaders.

5. **How we can select an appropriate process model. Differentiate between V-model and water fall model?**



Both Euromethod and ISO 12207 provide advice on this.

Euromethod distinguishes between the *construction* of an application and its *installation*. It is possible to use different approaches for these two stages. For example, an application could be constructed using a one-shot strategy but then be released to its users in increments. The only combinations of construction and installation strategies that are not feasible are evolutionary installation with any other construction approach than evolutionary.

In general, Euromethod suggests that where uncertainty is high then an evolutionary approach is to be favoured. An example of uncertainty would be where the users' requirements are not clearly defined. Where the requirements are relatively certain but there are many complexities, for example, where there is a large embedded system that is going to need a large amount of code, then an incremental approach might be favoured. Where deadlines are tight, then either an evolutionary or an incremental approach is favoured over a one-shot strategy, as both tactics should allow at least something to be delivered at the deadline, even if it is not all that was originally promised. Students about to plan final year projects would do well to note this.

Difference between V-Model and WaterFall Model:

Both Waterfall and V Model are most extensively practiced type of development methodology in software industry. Both of these models are practice for better tracking and have application development in systematic way.

On the basis of type of steps or phases between both of the models, we can distinguish between V and WaterFall Model as follows –

Sr. No.	Key	V-Model	WaterFall Model
1	Definition	V-Model is the development model in which the entire model is divided into various sub development phase where corresponding testing phase for each development phase is practices. In other words we can say that for every stage in the development cycle, there is an associated testing phase and corresponding testing phase of the development phase is planned in parallel.	On other hand Waterfall model there is first development of the application and after which the different testing of application take place. In other words we can say that in WaterFall the complete process is divided into several phases among which one phase should be completed in order to reach the next phase and testing is almost at end phase of the development.

Sr. No.	Key	V-Model	WaterFall Model
2	Type/Nature	As mentioned above that in V-Model the execution of the phases i.e., development and testing happens in a sequential manner so type of V-Model is Sequential/Parallel in nature.	On other hand WaterFall Model is a relatively linear sequential design approach as each phase should be completed in order to reach the next phase. So type of this model is Continuous in nature.
3	Testing and Validation	In V-Model each development phase get tested at its own level and hence no pending testing occurs in this model also if any validation requires to be implemented then it could be implemented at that phase.	On other hand in case of WaterFall Model the testing occurs after development is completed and thus if any missing validation is identified to be implemented then first that phase of development needs to be recognized and then that validation get implemented.
4	Cost and Complexity	As sequential phases need to be functional in case of V-Model hence the cost is higher as compared to that of WaterFall Model also the complexity is more than WaterFall.	On other hand in WaterFall Model due to linear development only one phase of development is operational and hence cost and complexity is low as compared to that of V-Model.
5	Defects	In V-Model the probability of total number of defects in the development of application is low as testing is done in parallel to the development.	On other hand in WaterFall Model the probability of total number of defects in the development of application is high as testing is done post development.

6. **Explain any two classical mistakes while developing Software project?**

The set of mistakes that researchers have identified is known as “Classic Mistakes”. Those bad practices have been chosen so often, by so many people. And those mistakes have predictable bad-results on the development of the project.

Four categories of classic mistakes:

1) People related
2) Process related
3) Product related
4) Technology related

People related classic mistakes:

This kind of mistakes talks about how to avoid mistakes among team mates. This kind of mistakes affect directly to the development speed and it is crucial to rectify those.

Undermined motivation – Studies have shown that giving suspicious talks at the beginning, asking to work overtime reduces the motivation of the people. Sometimes team leaders take long vacations while team is working overnights. The researchers highlighted that team lead has to work along with other team members is a positive motivation.

Weak personnel – If a team need an efficient development throughout the project, the recruitment needs to hire talented developers. Also carefully filter people who could do most of the work until the end of the project.

Uncontrolled problem employees – Failure to take actions for problems with team members and team leads will eventually affect the development speed. Some higher management should actively look into those and sort out.

Heroics – Heroics within the team increases the risk and discourages cooperation among the other members of the team

Adding people to a late project – Adding new people when the project is behind schedule, can take more productivity away from team members.

Noisy, crowded offices

Friction between developers and customers – Need to increase the communication between customers and developers.

Unrealistic expectations – Setting deadlines earlier without any proper reasons can lengthen the development schedule.

Process related classic mistakes

This type of mistakes talks about issues that may arise in management and technical methodologies.

Overly optimistic schedules – This sort of scheduling will result in failure by under-scoping the project and hurt long-term morale and productivity of the developers.

Insufficient risk management – If projects risks are not actively managed, the project will lead in to slow-development mode.

Contractor failure – weak relationship with contractors can lead to slow-down the project

Insufficient planning

Short-changed upstream activities – Start coding without properly design the project plans will cost 10 or 100 times than doing it with properly designed plans.

Short-changed quality assurance – Eliminating design and code reviews, eliminating test planning and do only perfunctory testing will reduce the development of the project and ends up with major bugs.

Omitting necessary tasks from estimates – People forget about the less visible tasks and those tasks add up.

Code-like-hell programming – Developers should be sufficiently motivated rather forcing them to work hard.

Product related classic mistakes

This type of mistakes talks about which can affect the outcome of the project.

Requirements gold-planting – More requirements that are not really necessary, and pay less attention on complex features

Feature creep – On average 25% of requirements can be changed and affect the project schedule.

Developer gold planting – It is frequent that developers attempt to try new technologies that they saw in other projects, which is not actually necessary.

Technology related classic mistakes

This type of mistakes is about technologies use during the project.

Silver-bullet syndrome – Thinking that certain approach will solve every issue, and that approach has not already used by developers (eg: Object oriented design)

Overestimated savings from new tools or methods – New practices will introduce a new risk as team has to go through a learning-curve to become familiar.

Switching tools in the middle of a project – Using new tools will add a learning curve, rework and inevitable mistakes to project schedule

Lack of automated source-code control – If two or more developers are working on the same part of the project, it is necessary to adhere to source-code control practices. If not developers have to spend time on resolving conflicting changes.

7. **You have a project completing in 12 month and total cost of project is $100,000. Six months have been passed (and schedule says that 50% of work should be completed). Six months have been passed and $60,000 is spent but on closer look you find that only 40% of work is completed so far. Calculate SPI and CPI to determine whether the project is on-time and on-budget after 6 months.**

Actual Cost(AC) = 60, 000 dollar

Planned Value(PV) = 50% of 100, 000 dollar
= 50, 000 dollar

Earned Value(EV) = 40% of 100, 000 dollar
= 40, 000 dollar

Cost Performance Index(CPI) = Earned Value / Actual Cost
= 40,000 / 60,000
= 0.67
Hence, the Cost Performance Index is **0.67** So,
This means you are earning 0.67 USD for every 1 USD spent since the Cost Performance Index is less than one. This means you are over budget.

Schedule Performance Index (SPI) = Earned Value / Planned Value
= 40,000 / 50,000
= 0.8
Hence, the Schedule Performance Index is **0.8** So,
You are behind schedule since the Schedule Performance Index is less than one.

8. **Risk can be dealt by how many ways explain them in detail with examples?**
   When you're planning your project, risks are still uncertain: they haven't happened yet. But eventually, some of the risks that you plan for do happen, and that's when you have to deal with them. There are four basic ways to handle a risk.
   Avoid: The best thing you can do with a risk is avoid it. If you can prevent it from happening, it definitely won't hurt your project. The easiest way to avoid this risk is to walk away from the cliff, but that may not be an option on this project.
   Mitigate: If you can't avoid the risk, you can mitigate it. This means taking some sort of action that will cause it to do as little damage to your project as possible.
   Transfer: One effective way to deal with a risk is to pay someone else to accept it for you. The most common way to do this is to buy insurance.
   Accept: When you can't avoid, mitigate, or transfer a risk, then you have to accept it. But even when you accept a risk, at least you've looked at the alternatives and you know what will happen if it occurs. If you can't avoid the risk, and there's nothing you can do to reduce its impact, then accepting it is your only choice.
9. **Differentiate between Software prototyping model and agile process model?**
   Prototyping
   Prototyping begins before development is underway. Prototypes can range from a quick sketch on a whiteboard to a fully interactive mockup.
   The first step is an initial prototype of the project, which will be presented so you can give feedback. Each prototype is then refined, and the process uses a trial-and-error approach until the last prototype is finalised, and development can begin.
   Prototyping allows you to experiment and spot bugs early on in the process. You and your team can see how your product would be used, and how each change would impact the end-result. You can also easily change direction if you find that you have too many features or the design is not user-friendly.
   During the prototype model, you're constantly kept updated throughout the process, and your feedback is continually incorporated. This can lead to a better overall result for end users.
   Agile Development
   Agile is a development methodology. Basically, the developers will develop and test in different stages (iterations) instead of waiting until the whole product has been completed.
   Agile is popular due to its detailed, accurate, consistent, and simple nature. The software is delivered in small parts, which are considered to be miniature projects themselves. Just as the director of a play will work on each scene individually before putting it all together, the agile model is about breaking a large project down into multiple iterations.
   These iterations are called sprints, and they can help minimise risk. That's because the agile development methodology involves a cycle of planning, testing, integration, and risk evaluation. This reduces the chances that your project will fail.\
   When should you use prototyping?
   Prototyping is a good choice if you have an idea or plan for your project, but you're wondering whether it will be feasible. It can also provide you and your team with clarity if you're not entirely sure about the direction or requirements of your project.
   You may also find that you're ready to get started on a project, but investors or stakeholders are unable to see the value. In this case, you can provide decision-makers with an interactive wireframe so you can communicate key ideas.

Prototyping is also a solid choice if you have a small budget. The more time spent in development, the more your project is likely to cost. Instead, working with increasingly functional prototypes will mean that developers can spot issues and bugs early on. That way, large problems can be solved well before development has begun- when issues are much more time-consuming and expensive to fix.

When should you use agile development?

Agile is a good choice if your requirements are continually changing. You can pivot later in the development process, instead of following a predetermined structure or development plan.

With agile development, it's easy to measure your progress, since you can easily see how much work has been completed. It may also be the best option if you're on a tight deadline, since agile allows for a rapid delivery.

The key with agile? Ensure you have an experienced project manager, so that the project doesn't become a large series of sprints which can lead to a budget blowout.

10. Explain function point estimation, Cocomo estimation and Line of code software estimation techniques and example?

Estimation of the size of software is an essential part of Software Project Management. It helps the project manager to further predict the effort and time which will be needed to build the project. Various measures are used in project size estimation. Some of these are as follows:

1. Lines of Code (LOC):

Estimation is done on behalf of number of line of codes in the software product.

As the name suggest, LOC count the total number of lines of source code in a project. The units of LOC are:

- KLOC- Thousand lines of code
- NLOC- Non comment lines of code
- KDSI- Thousands of delivered source instruction

The size is estimated by comparing it with the existing systems of same kind. The experts use it to predict the required size of various components of software and then add them to get the total size.

Advantages:

- Universally accepted and is used in many models like COCOMO.
- Estimation is closer to developer's perspective.
- Simple to use.

Disadvantages:

- Different programming languages contains different number of lines.
- No proper industry standard exist for this technique.
- It is difficult to estimate the size using this technique in early stages of project.

2. Function Point Analysis:

Estimation is done on behalf of number of function points in the software product.

In this method, the number and type of functions supported by the software are utilized to find FPC(function point count). The steps in function point analysis are:

- Count the number of functions of each proposed type.
- Compute the Unadjusted Function Points(UFP).
- Find Total Degree of Influence(TDI).
- Compute Value Adjustment Factor(VAF).

- Find the Function Point Count(FPC).

The explanation of above points given below:

- **Count the number of functions of each proposed type:** Find the number of functions belonging to the following types:
  - External Inputs: Functions related to data entering the system.
  - External outputs:Functions related to data exiting the system.
  - External Inquiries: They leads to data retrieval from system but don't change the system.
  - Internal Files: Logical files maintained within the system. Log files are not included here.
  - External interface Files: These are logical files for other applications which are used by our system.
- **Compute the Unadjusted Function Points(UFP):** Categorise each of the five function types as simple, average or complex based on their complexity. Multiply count of each function type with its weighting factor and find the weighted sum. The weighting factors for each type based on their complexity are as follows:

FUNCTION TYPE	SIMPLE	AVERAGE	COMPLEX
External Inputs	3	4	6
External Output	4	5	7
External Inquiries	3	4	6
Internal Logical Files	7	10	15
External Interface Files	5	7	10

- **Find Total Degree of Influence:** Use the '14 general characteristics' of a system to find the degree of influence of each of them. The sum of all 14 degrees of influences will give the TDI. The range of TDI is 0 to 70. The 14 general characteristics are: Data Communications, Distributed Data Processing, Performance, Heavily Used Configuration, Transaction Rate, On-Line Data Entry, End-user Efficiency, Online Update, Complex Processing Reusability, Installation Ease, Operational Ease, Multiple Sites and Facilitate Change.
  Each of above characteristics is evaluated on a scale of 0-5.
- **Compute Value Adjustment Factor(VAF):** Use the following formula to calculate VAF
  VAF = (TDI * 0.01) + 0.65
- **Find the Function Point Count:** Use the following formula to calculate FPC
  FPC = UFP * VAF
  Advantages:
  - It can be easily used in the early stages of project planning.
  - It is independing on the programming language.
  - It can be used to compare different projects even if they use different technologies (database, language etc).

Disadvantages:

- It is not good for real time systems and embedded systems.
- Many cost estimation models like COCOMO uses LOC and hence FPC must be converted to LOC.

3. COCOMO:

COCOMO stands for COnstructive COst MOdel, developed by Barry W. Boehm. It divides the software product into three categories of software: organic, semi-detached and embedded.
The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics.
COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers). Intermediate COCOMO takes these Cost Drivers into account and Detailed COCOMO additionally accounts for the influence of individual project phases.

i. Basic COCOMO:

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of lines of code (KLOC).

COCOMO applies to three classes of software projects:

- Organic projects - "small" teams with "good" experience working with "less than rigid" requirements
- Semi-detached projects - "medium" teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of "tight" constraints (hardware, software, operational, ...)

The basic COCOMO equations take the form:

Effort Applied = ab(KLOC)bb [man-months]
Development Time = cb(Effort Applied)db [months]
People required = Effort Applied / Development Time [count]
Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware constraints, personnel quality and experience, use of modern tools and techniques, and so on.

ii. Intermediate COCOMO:

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers", each with a number of subsidiary attributes:-

Product attributes

- Required software reliability
- Size of application database
- Complexity of the product

Hardware attributes

- Run-time performance constraints
- Memory constraints
- Volatility of the virtual machine environment

- Required turnabout time

Personnel attributes

- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience

Project attributes

- Use of software tools
- Application of software engineering methods
- Required development schedule

The Intermediate Cocomo formula now takes the form:

$$E=a_i(\text{KLoC})^{(b_i)}\text{EAF}$$

where E is the effort applied in person-months, KLoC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above. The coefficient ai and the exponent bi are given in the next table.

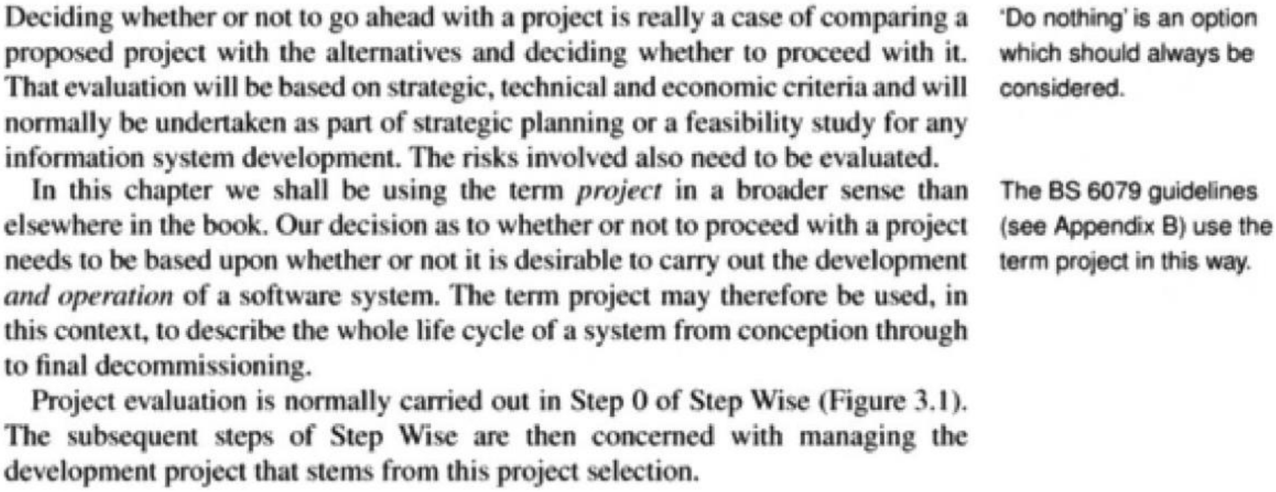
Software project	**ai**	**bi**
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

The Development time D calculation uses E in the same way as in the Basic COCOMO.

iii. Detailed COCOMO:

Detailed COCOMO - incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process 1. the detailed/ model uses different efforts multipliers for each cost drivers attribute these Phase Sensitive effort multipliers are each to determine the amount of effort required to complete each phase.

11. How projects are evaluated? Explain economic assessment and project portfolio management in detail?



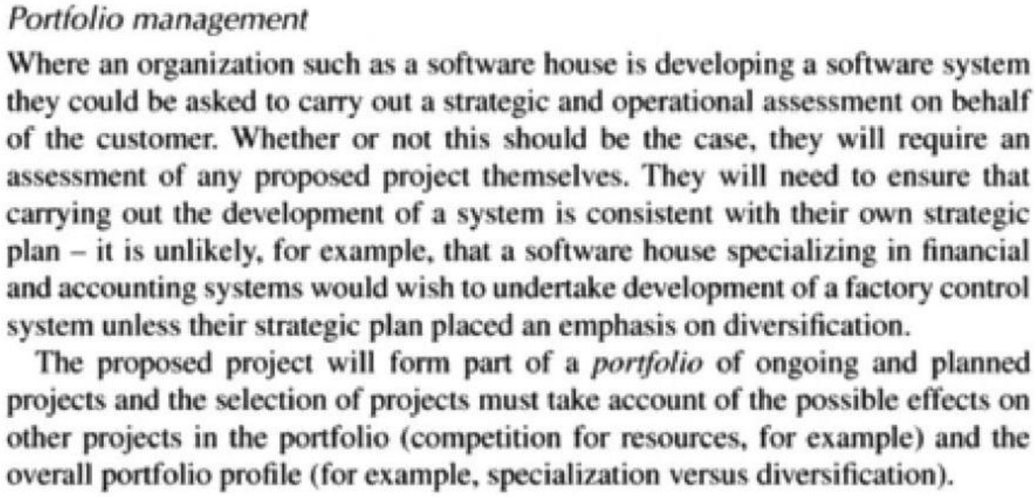
Deciding whether or not to go ahead with a project is really a case of comparing a proposed project with the alternatives and deciding whether to proceed with it. That evaluation will be based on strategic, technical and economic criteria and will normally be undertaken as part of strategic planning or a feasibility study for any information system development. The risks involved also need to be evaluated.

'Do nothing' is an option which should always be considered.

In this chapter we shall be using the term *project* in a broader sense than elsewhere in the book. Our decision as to whether or not to proceed with a project needs to be based upon whether or not it is desirable to carry out the development *and operation* of a software system. The term project may therefore be used, in this context, to describe the whole life cycle of a system from conception through to final decommissioning.

The BS 6079 guidelines (see Appendix B) use the term project in this way.

Project evaluation is normally carried out in Step 0 of Step Wise (Figure 3.1). The subsequent steps of Step Wise are then concerned with managing the development project that stems from this project selection.

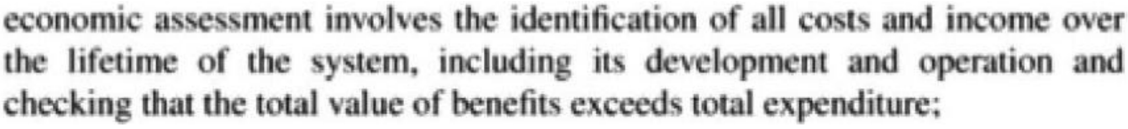


Portfolio management

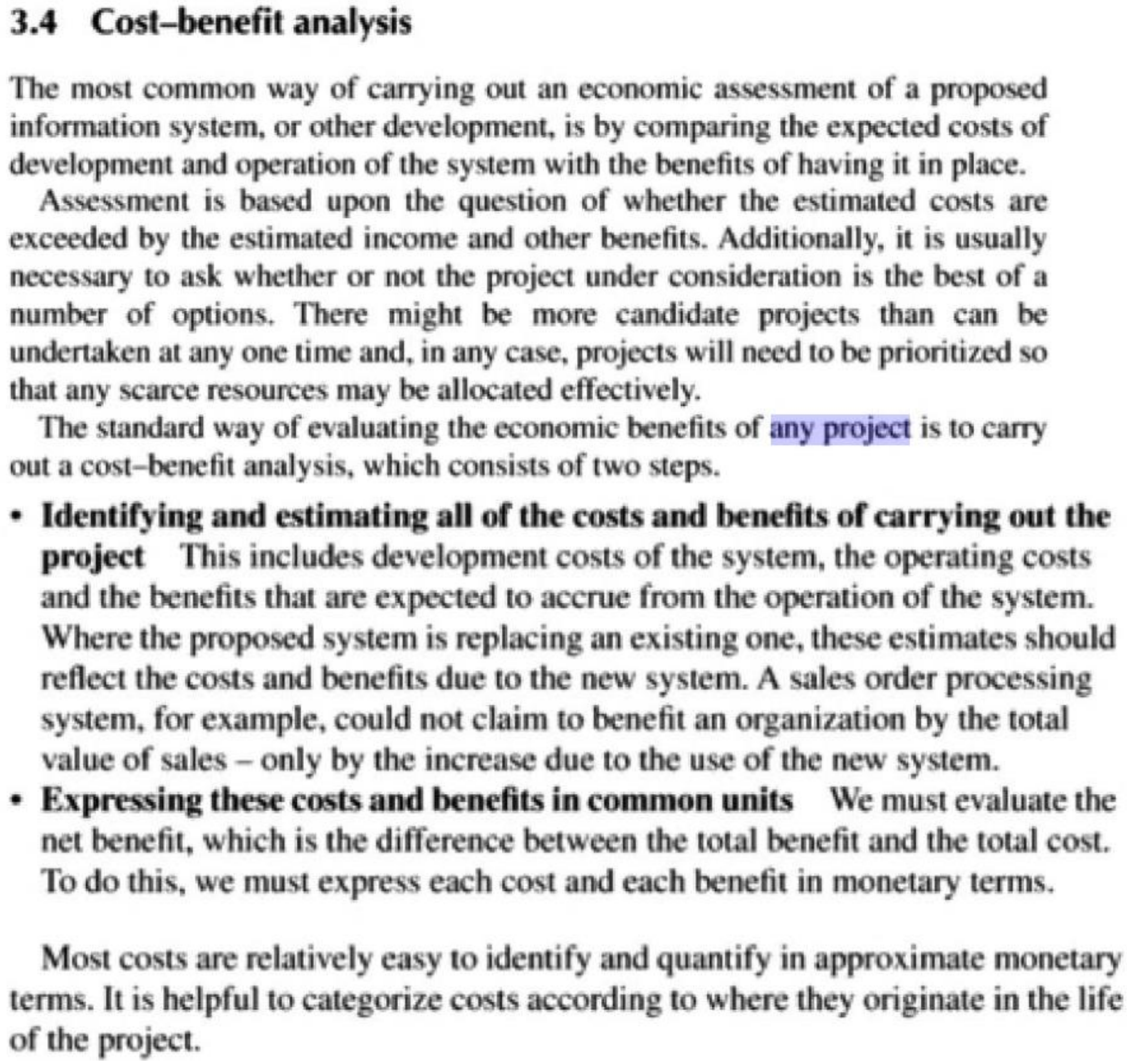
Where an organization such as a software house is developing a software system they could be asked to carry out a strategic and operational assessment on behalf of the customer. Whether or not this should be the case, they will require an assessment of any proposed project themselves. They will need to ensure that carrying out the development of a system is consistent with their own strategic plan – it is unlikely, for example, that a software house specializing in financial and accounting systems would wish to undertake development of a factory control system unless their strategic plan placed an emphasis on diversification.

The proposed project will form part of a *portfolio* of ongoing and planned projects and the selection of projects must take account of the possible effects on other projects in the portfolio (competition for resources, for example) and the overall portfolio profile (for example, specialization versus diversification).

Ecnomic Assessment:



economic assessment involves the identification of all costs and income over the lifetime of the system, including its development and operation and checking that the total value of benefits exceeds total expenditure;



3.4 Cost–benefit analysis

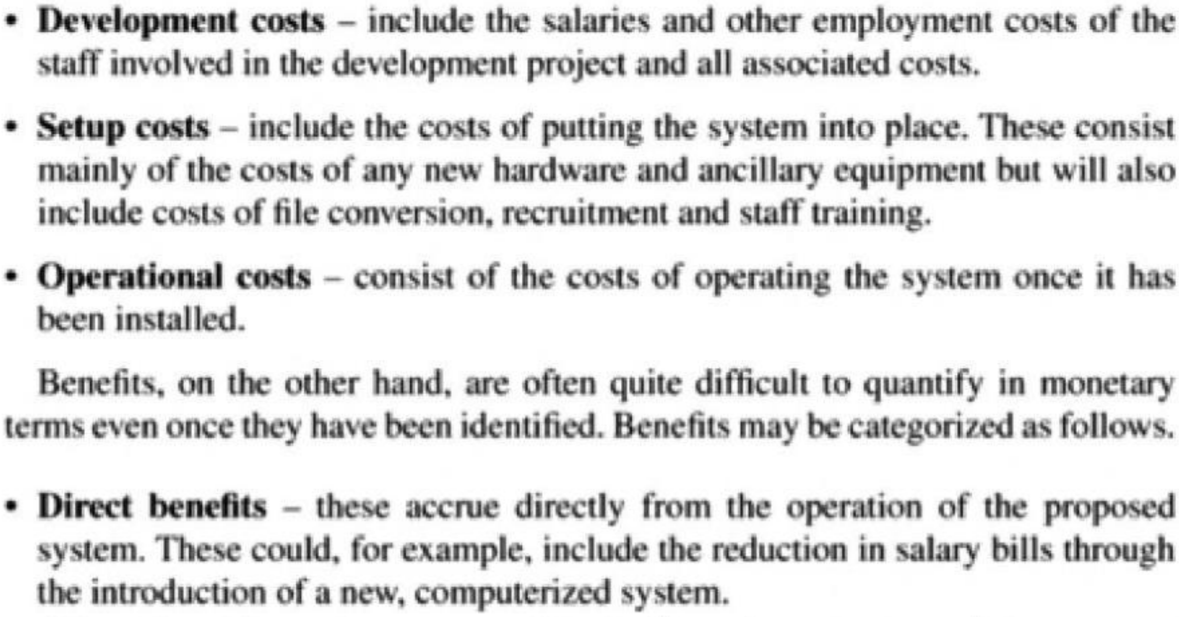
The most common way of carrying out an economic assessment of a proposed information system, or other development, is by comparing the expected costs of development and operation of the system with the benefits of having it in place.

Assessment is based upon the question of whether the estimated costs are exceeded by the estimated income and other benefits. Additionally, it is usually necessary to ask whether or not the project under consideration is the best of a number of options. There might be more candidate projects than can be undertaken at any one time and, in any case, projects will need to be prioritized so that any scarce resources may be allocated effectively.

The standard way of evaluating the economic benefits of any project is to carry out a cost–benefit analysis, which consists of two steps.

- **Identifying and estimating all of the costs and benefits of carrying out the project** This includes development costs of the system, the operating costs and the benefits that are expected to accrue from the operation of the system. Where the proposed system is replacing an existing one, these estimates should reflect the costs and benefits due to the new system. A sales order processing system, for example, could not claim to benefit an organization by the total value of sales – only by the increase due to the use of the new system.
- **Expressing these costs and benefits in common units** We must evaluate the net benefit, which is the difference between the total benefit and the total cost. To do this, we must express each cost and each benefit in monetary terms.

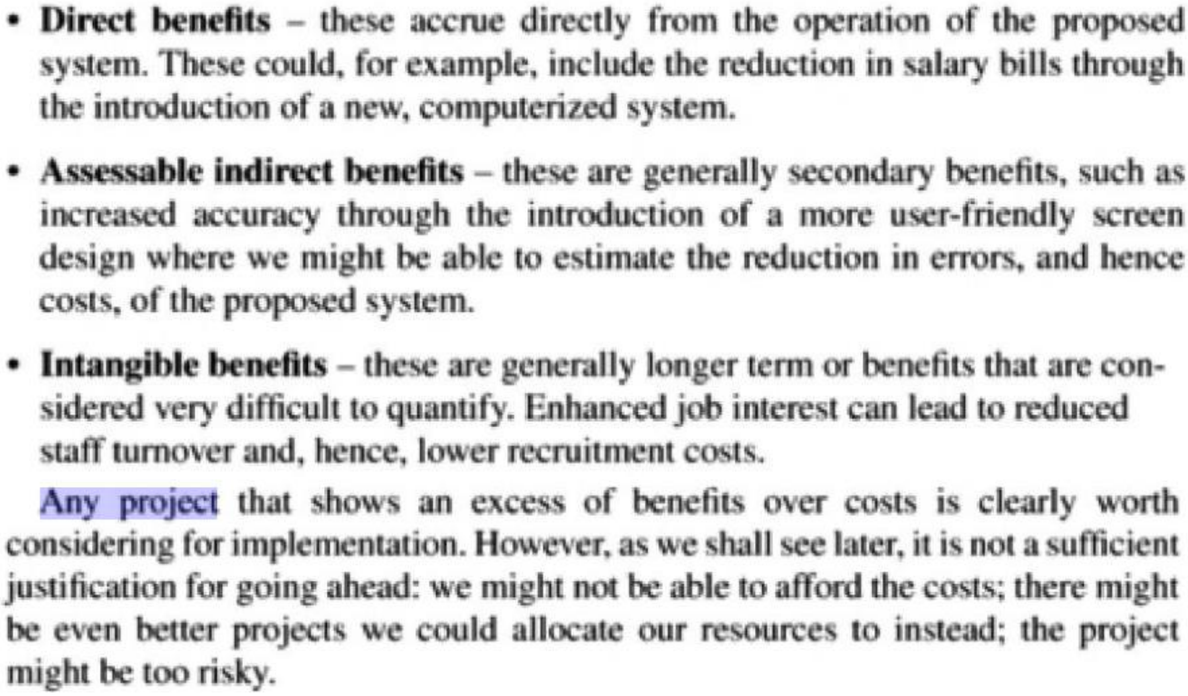
Most costs are relatively easy to identify and quantify in approximate monetary terms. It is helpful to categorize costs according to where they originate in the life of the project.



- **Development costs** – include the salaries and other employment costs of the staff involved in the development project and all associated costs.
- **Setup costs** – include the costs of putting the system into place. These consist mainly of the costs of any new hardware and ancillary equipment but will also include costs of file conversion, recruitment and staff training.
- **Operational costs** – consist of the costs of operating the system once it has been installed.

Benefits, on the other hand, are often quite difficult to quantify in monetary terms even once they have been identified. Benefits may be categorized as follows.

- **Direct benefits** – these accrue directly from the operation of the proposed system. These could, for example, include the reduction in salary bills through the introduction of a new, computerized system.



- **Direct benefits** – these accrue directly from the operation of the proposed system. These could, for example, include the reduction in salary bills through the introduction of a new, computerized system.

- **Assessable indirect benefits** – these are generally secondary benefits, such as increased accuracy through the introduction of a more user-friendly screen design where we might be able to estimate the reduction in errors, and hence costs, of the proposed system.
- **Intangible benefits** – these are generally longer term or benefits that are considered very difficult to quantify. Enhanced job interest can lead to reduced staff turnover and, hence, lower recruitment costs.

Any project that shows an excess of benefits over costs is clearly worth considering for implementation. However, as we shall see later, it is not a sufficient justification for going ahead: we might not be able to afford the costs; there might be even better projects we could allocate our resources to instead; the project might be too risky.

12. What is risk analysis? Explain any two types of risk in detail?

Risk Analysis:

Risk analysis involves examining how project outcomes and objectives might change due to the impact of the risk event.

Once the risks are identified, they are analysed to identify the qualitative and quantitative impact of the risk on the project so that appropriate steps can be taken to mitigate them. The following guidelines are used to analyse risks.

Types of risks:

Five Types of Risk In Software Project Management are as follows:

- New, unproven technologies
- User and functional requirements
- Application and system architecture
- Performance

  - Organizational
  - Estimation

- **New, unproven technologies.** The majority of software projects entail the use of new technologies. Ever-changing tools, techniques, protocols, standards, and development systems increase the probability that technology risks will arise in virtually any substantial software engineering effort. Training and knowledge are of critical importance, and the improper use of new technology most often leads directly to project failure.
- **User and functional requirements.** Software requirements capture all user needs with respect to the software system features, functions, and quality of service. Too often, the process of requirements definition is lengthy, tedious, and complex. Moreover, requirements usually change with discovery, prototyping, and integration activities. Change in elemental requirements will likely propagate throughout the entire project, and modifications to user requirements might not translate to functional requirements. These disruptions often lead to one or more critical failures of a poorly-planned software development project.
- **Application and system architecture.** Taking the wrong direction with a platform, component, or architecture can have disastrous consequences. As with the technological risks, it is vital that the team includes experts who understand the architecture and have the capability to make sound design choices.
- **Performance.** It’s important to ensure that any risk management plan encompasses user and partner expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the project to ensure that the work products are moving in the right direction.
- **Organizational.** Organizational problems may have adverse effects on project outcomes. Project management must plan for efficient execution of the project, and find a balance between the needs of the development team and the expectations of the customers. Of course, adequate staffing includes choosing team members with skill sets that are a good match with the project.
- **Estimation.** The time required to develop the software is underestimating. Or the rate of defect repair is underestimating. Or the size of software is underestimating.

13. **Briefly describe change management and configuration management?**

Project environments are dynamic and changes are constant in areas like process, planning, or scope. You can group these changes into two categories:

- Change Management
- Configuration Management

“Change Management” is the first category. Here you manage changes related to project management plans, processes, and baselines.

In the second category, you manage changes related to product scope, which is known as configuration management.

Change requests are required when baselines are established and you have to make changes to them. If the baselines are not set, no formal change request is required. Change requests and configuration requests are part of the integration management system.

Change Management System:

Change Control is focused on identifying, documenting and controlling changes to the project and the project baselines. In the change management system, you manage the changes related to the project scope, planning, and baselines.

For example, you run out of money and you need additional funding to complete the project, therefore, you will raise a change request for additional funds. Or, you may not be able to complete your project within the specified time and require a time extension.

In the change management system, the change request is analyzed for any possible impact on any other project objectives. Afterwards, the request is either approved or rejected.

To minimize disruption, a change management system must ensure that all parameters are identified and analyzed for any possible impact.

If the change request is approved, you will update the concerned baseline, update the project documents, and inform the concerned stakeholders.

Change Management Activities:

You do the following during change management:

- Identify the changes.
- Prepare a proper documentation for the changes.
- Review, analyze, and make a decision for the change request.
- Make sure that request is implemented, registered and communicated.

Configuration Management System:

Configuration Control focuses on the specifications of both the deliverables and the processes. In the configuration management system, you manage the changes related to the product specification and the process.

For example, suppose you are developing a product and the client requests the addition of some extra features.

Since this change is related to the configuration of the product, you will deal with this change using the configuration management system.

Configuration management documents how you will monitor and control changes. It; i a process of defining configurable items (product, service, result, and component) and controlling changes to such items.

The configuration management plan keeps version control of the product. Here you can keep a log of all the changes made to any version of the product for review.

Configuration Management Activities:

You do the following during configuration management:

- Identify the configurable items.
- Record and prepare a report for all configurable items.
- Verify and conduct an audit of all configurations are as per the requirements.

14. Discuss the conventional work break down structure issue?

Conventional work breakdown structures frequently suffer from three fundamental flaws.

1. They are prematurely structured around the product design.

2. They are prematurely decomposed, planned, and budgeted in either too much or too little detail.

3. They are project-specific, and cross-project comparisons are usually difficult or impossible.

1.Conventional work breakdown structures are prematurely structured around the product design:

Figure 10-1 shows a typical conventional WBS that has been structured primarily around the subsystems of its product architecture, then further decomposed into the components of each subsystem. A WBS is the architecture for the financial plan.

2. Conventional work breakdown structures are prematurely decomposed, planned, and budgeted in either too little or too much detail:

Large software projects tend to be over planned and small projects tend to be under planned. The basic problem with planning too much detail at the outset is that the detail does not evolve with the level of fidelity in the plan.

3. Conventional work breakdown structures are project-specific, and cross-project comparisons are usually difficult or impossible:

With no standard WBS structure, it is extremely difficult to compare plans, financial data, schedule data, organizational efficiencies, cost trends, productivity trends, or quality trends across multiple projects.

15. Explain traditional project management constraints?

With any project, there are limitations and risks that need to be taken into account and addressed to ensure the project's ultimate success. The three primary constraints that project managers should be familiar with are time, scope and cost. These are frequently known as the triple constraints or the project management triangle. Each constraint is connected to the other two; so, for example, increasing the scope of the project will likely require more time and money, while speeding up the timeline for the project may cut costs, but also diminish the scope.

The triple constraints of project management:

Time constraint: The time constraint refers to the project's schedule for completion, including the deadlines for each phase of the project, as well as the date for rollout of the final deliverable.

Scope constraint: The scope of a project defines its specific goals, deliverables, features and functions, in addition to the tasks required to complete the project.

Cost constraint: The cost of the project, often dubbed the project's budget, comprises all of the financial resources needed to complete the project on time, in its predetermined scope. Keep in mind that cost does not just mean money for materials—it encompasses costs for labor, vendors, quality control and other factors, as well.

1.Time constraint:

When it comes to the time constraint, proper scheduling is essential. There are following steps should be taken for effective time management:

✓**Planning:** This includes defining the main goal(s) of the project team, how the team intends to achieve the goal, and the equipment and/or steps that will be taken to do so.

✓**Scheduling:** The project management team must plot out the realistic timeframe for completion of each phase of the project.

✓**Monitoring:** This step occurs once the project is underway and requires the project team to analyze how the past stages of the project performed, noting trends and impacts on future plans, and communicating these findings to all relevant stakeholders.

✓**Control:** In the control step, the team must, upon communicating the results of each phase of the project, move forward accordingly. That means if things are running smoothly, the team must analyze the factors contributing to that positive outcome so that it can be continued and replicated. If there has been a derailment, the team must know how and why the derailment occurred and take steps to correct it for future actions.

A **Gantt chart** can be helpful to visualize the project timeline and whether they are tracking to the proper constraints.

2. Scope constraint:

Defined upfront, the scope of the project should be clearly and regularly communicated to all stakeholders to ensure that "scope creep"—the term used when changes are made to the scope mid-project, without the same levels of control—is avoided. To keep the scope in check, you can:

✓Provide clear documentation of the full project scope at the beginning of the project, including all requirements.

✓Set up a process for managing any changes, so if someone proposes a change, there is a controlled system in place for how that change will be reviewed, approved or rejected, and implemented if applicable.

✓Communicate the scope clearly and frequently with stakeholders.

3. Cost constraint:

A project's budget includes both fixed and variable costs, including materials, permits, labor and the financial impact of team members working on the project. A few of the ways to estimate the cost of a project include:

✓**Historical data:** Looking at what similar projects cost in the recent past

✓**Resources:** Estimating the rate of cost for goods and labor.

✓**Parametric:** Comparing historical data with updated, relevant variables

✓**Vendor bid:** Averaging the total charge of several solid vendor bids

16. What is configuration management? Differentiate between configuration control and version control?

Configuration management:

Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions and development of the product.

Configuration management is about to control the changes in software like requirements, design, and development of the product.

The Primary goal is to increase productivity with fewer errors.

Some reasons show the need for configuration management:

- Several people work on software that is continually update.
- Help to build coordination among suppliers.
- Changes in requirement, budget, schedule need to accommodate.
- Software should run on multiple systems.

Tasks perform in Configuration management:

- Identification
- Baseline
- Change Control
- Configuration Status Accounting
- Configuration Audits and Reviews

People involved in configuration management:

- Project manager
- Configuration manager
- Developer
- User

Difference between configuration control and version control:

Versioning is the creation and management of multiple releases of a product, all of which have the same general function but are improved, upgraded or customized. The term applies especially to operating systems (OSs), software and Web services. Version control is the practice of ensuring collaborative data sharing and editing among users of systems that employ different versions of a product. The terms “versioning” and “version control” are sometimes used interchangeably even though their technical meanings are different.

Configuration control refers to setting runtime dependencies and we often discuss “configuring” an application to run. An example would be a JMX control or even more basic – specifying whether you are accessing a QA/UAT or production database. There are lots of jobs out there where you focus on configuration management in the sense of configuring a package to run (actually customizing the runtime experience). This is often done through XML or properties files such as an application server(e.g. WebSphere).

Version control refers to checking in and storing specific versions of the source code and now there is a real difference between configuration control and version control. Years ago the terms were used almost interchangeably although back then (around the 80s and early 90s) we didn’t have too many real version control tools. On mainframes we had Pan valet and I think that CA Librarian came soon after.

Configuration control applies to service assets as a whole, of which, systems configuration is a subset. In places where configuration baselines are adopted, configurations of the service assets must be within the limits that are recommended in the baselines – so, in a way, configuration control also means that the configurations of the Configuration Items (CI) don’t cross above/below the limits defined in the baselines. That way, we ensure that all the assets follow uniform configurations. If and when there is a need to cross these limits, the concerned team must get the approval of Change Advisory Board (CAB) in order to make those changes.

17. What are the activities of software management team?

A software project is concerned not only with the actual writing of software. In fact, where a software application is bought in 'off-the-shelf', there might be no software writing as such. This is still fundamentally a software project because so many of the other elements associated with this type of project are present. Usually, there are three successive processes that bring a new system into being:

l. The feasibility study:

This is an investigation to decide whether a prospective project is worth starting. Information will be gathered about the general requirements of the proposed system. The probable developmental and operational costs, along with the value of the benefits of the new system are estimated. With a large system, the feasibility study could be treated as a project in its own right. This evaluation may be done as part of a strategic planning exercise where a whole range of potential software developments are evaluated and put into an order of priority. Sometimes an organization has a policy where a series of projects is planned as a programme of development.

2. Planning: If the feasibility study produces results that indicate that the prospective project appears viable, then planning of the project can take place. In fact, for a large

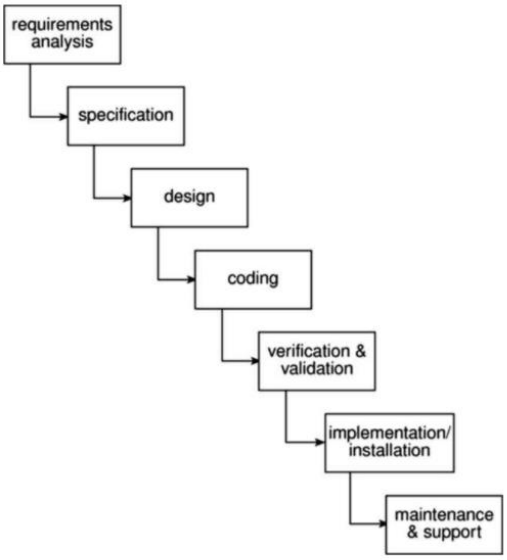


Figure 1.1 A typical project life-cycle.

project, we would not do all our detailed planning right at the beginning. We would formulate an outline plan for the whole project and a detailed one for the first stage. More detailed planning of the later stages would be done as they approached. This is because we would have more detailed and accurate information upon which to base our plans nearer to the start of the later stages.

3. Project execution The project can now be executed. Individual projects are likely to differ considerably but a classic project life-cycle is shown in Figure 1.1.

The stages in the life-cycle illustrated in Figure 1.1 above are described in a little more detail below:

Requirements analysis: This is finding out in detail what the users require of the system that the project is to implement. Some work along these lines will almost certainly have been carried out when the project was evaluated but now the original information obtained needs to be updated and supplemented. Several different approaches to the users' requirements may be explored. For example, a small system that satisfies some, but not all, of the users' needs at a low price may be compared to a system with more functions but at a higher price.

Specification: Detailed documentation of what the proposed system is to do.

Design: A design that meets the specification has to be drawn up. This design activity will be in two stages. One will be the external or user design. This lays down what the system is to look like to the users in terms of menus, screen and report layouts and so on. The next stage produces the physical design, which tackles the way in which the data and software procedures are be structured internally.

Coding: This might refer to writing code in a procedural language such as C or Ada, or might refer to the use of a high level application builder. Even where software is not being built from scratch, some modification to the base application might be required to meet the needs of the new application.

Verification and validation: Whether software is developed specially for the current application or not, careful testing will be needed to check that the proposed system meets its requirements.

Implementation/installation: Some system development practitioners refer to the whole of the project after design as 'implementation' (that is, the implementation of the design) while others insist that the term refers to the installation of the system after the software has been developed. In this case it encompasses such things as setting up data files and system parameters, writing user manuals and training users of the new system.

Maintenance and support: Once the system has been implemented there will be a continuing need for the correction of any errors that may have crept into the system and for extensions and improvements to the system. Maintenance and support activities may be seen as a series of minor software projects. In many environments, most software development is in fact maintenance.

18. Explain the risks that are involved in delaying the project. Also explain the version control and configuration control management?

Risks involved in delaying the project:

There are number of risks that causes a project to be delayed are as follows:

1. **Expansion of functionality**
   The expansion of functionality is a phenomenon in which new functionalities continue to be conceived and requested as the project proceeds. The software can never be completed in this way.
2. **Gold plating**
   Gold plating is a phenomenon in which programmers and designers try to make many details of the software or design too elaborate. Much time is spent improving details, even though the improvements were not requested by the customer or client. The details often add little to the desired result.
3. **Neglecting quality control**
   Time pressure can sometimes cause programmers or project teams to be tempted to skip testing. This frequently causes more delays than it prevents. The time that elapses before an error is discovered in the software is associated with an exponential increase in the time that is needed to repair it.

4. **Overly optimistic schedules**
   Overly optimistic schedules place considerable pressure on the project team. The team will initially attempt to reach the (unrealistic) deadlines. These attempts lead to sloppy work and more errors, which cause further delays.
   In this regard, be particularly wary of schedules that are imposed from above. The desire to complete a project (more) quickly sometimes arises for primarily strategic reasons; if it is not feasible, however, it should not be attempted. The project will not proceed more quickly and the product will ultimately suffer.
5. **Working on too many projects at the same time**
   Dividing work across many different projects (or other tasks) causes waiting times that lead to many delays in projects.
6. **Poor design**
   The absence (or poor realisation) of designs leads to delays, as it requires many revisions at later stages.
7. **The 'one-solution-fits-all' syndrome**
   Using the right software for a project is important. Some software platforms are more suited to particular applications than others are. Thinking that the use of particular software will greatly improve productivity, however, is also a trap.
8. **Research-oriented projects**
   Projects in which software must be made and research must be conducted are difficult to manage. Research is accompanied by high levels of uncertainty. When or if progress will be achieved in research is unclear. When software development is dependent upon the results of research, the former frequently comes to a standstill.
9. **Mediocre personnel**
   Insufficiently qualified personnel can cause project delays. Technically substantive knowledge of the subject of the project plays a role, as do knowledge and skills in working together to play the game of the project.
10. **Customers fail to fulfil agreements**
    Customers are not always aware that they are expected to make a considerable contribution to the realisation of a project. When customers do not react in a timely manner to areas in which they must be involved, projects can come to a standstill. Worse yet, the team may proceed further without consulting the customer, which can lead to later conflicts.
11. **Tension between customers and developers**
    The tension that can arise between customers and developers (e.g. because the project is not proceeding quickly enough) can cause additional delays, as it disturbs the necessary base of trust and the working atmosphere.

Version Control:

Version control, also known as revision control or source control, is the management of changes to documents, computer programs, large websites, and other collections of information. Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and with some types of files, merged.

Version control systems (VCS) most commonly run as stand-alone applications, but may also be embedded in various types of software, including integrated development environments (IDEs). Any system that provides change tracking and control over programming source code and documentation can be considered version control software. The practice has been a part of creative processes almost as long as writing has existed.

Purpose of version control:

The purpose of version control is ensuring that content changes under development go as planned. While version control is often carried out by a separate application, it can also be embedded into programs such as integrated development environments (IDEs), word processors, spreadsheets and, especially, collaborative web documents and pages. Version control allows servers in multiple locations to run different versions on different sites, even while those versions are being updated simultaneously.

The most powerful and complex version control systems are used in software development. Version control often operates by locking files and using a check-out / check-in system for changed versions. Versions may be identified by labels or tags; approved versions or those that are especially significant may be designated baselines.

Another method used in version control is branching, in which programs in development are copied for development in parallel versions, retaining the original and working on the branch or making different changes to each. Each copy is considered a branch; the original program from which the branch is taken is referred to as the trunk, the baseline, the mainline or the master.

Version control is generally based on a client-server model. Another method is distributed version control, in which all copies are in a codebase repository and changes are synchronized through patches or changes shared from peer to peer.

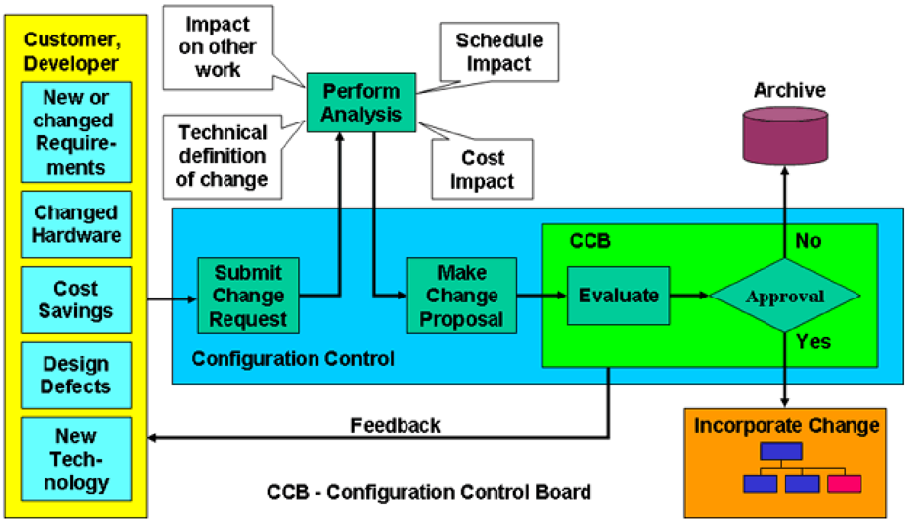
Configuration Control:

Configuration control is an important function of the configuration management discipline. Its purpose is to ensure that all changes to a complex system are performed with the knowledge and consent of management. The scope creep that results from ineffective or nonexistent configuration control is a frequent cause of project failure.

Configuration control tasks include initiating, preparing, analysing, evaluating and authorising proposals for change to a system (often referred to as "the configuration"). Configuration control has four main processes:

1. Identification and documentation of the need for a change in a change request
2. Analysis and evaluation of a change request and production of a change proposal
3. Approval or disapproval of a change proposal
4. Verification, implementation and release of a change.

The Configuration Control Process:



Importance of configuration control: Configuration control is an essential component of a project's risk management strategy. For example, uncontrolled changes to software requirements introduce the risk of cost and schedule overruns.

19. Explain the COCOMO and PMI's process group with a diagram?

COCOMO Model:

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Models of COCOMO:

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

i. Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

ii. Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.

iii. Embedded – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

Types of Models:

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. Any of the three forms can be adopted according to our requirements. These are types of COCOMO model:

i. Basic COCOMO Model
ii. Intermediate COCOMO Model
iii. Detailed COCOMO Model

The first level, **Basic COCOMO** can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations. **Intermediate COCOMO** takes these Cost Drivers into account and **Detailed COCOMO** additionally accounts for the influence of individual project phases, i.e in case of Detailed it accounts for both these cost drivers and also calculations are performed phase wise henceforth producing a more accurate result.

PMI's Process Groups:

Project Management has several knowledge areas. The project management processes are a part of those project management knowledge areas. All of these project management processes belong to five major project management process groups. These project management process groups are called initiating, planning, executing, monitoring and controlling and closing.

The five PMBOK process groups are:

Initiating: Processes required to launch a new project or a new project phase.

Planning: Processes related to defining and planning the extent of the project, as well as planning how it will be executed.

Executing: Processes related to the actual completion of project activities and tasks.

Monitoring & Controlling: Processes covering everything related to tracking, monitoring, reporting on, and controlling project performance and progress.

Closing: Processes required to finalize and complete a project or project phase.

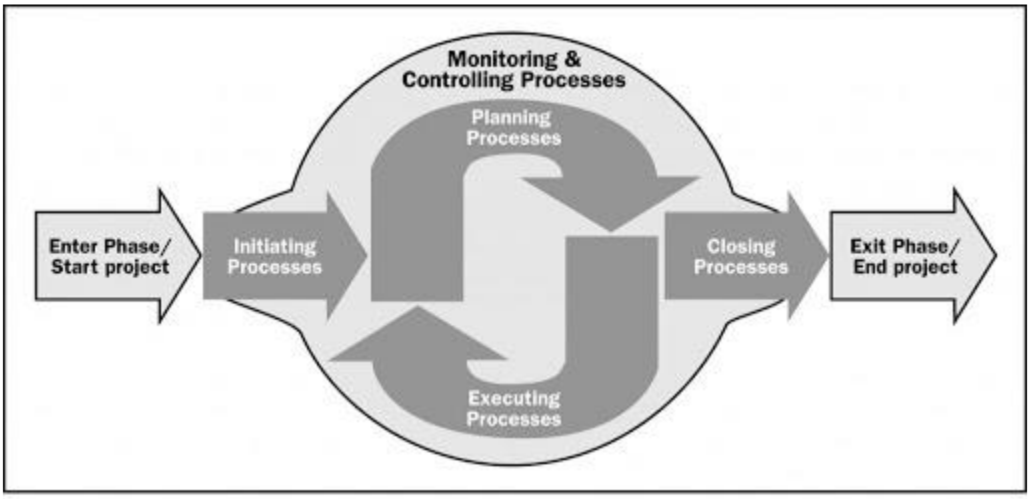


Figure 3-1. Project Management Process Groups

A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fourth Edition. ©2008 Project Management Institute, Inc. All Rights Reserved.

i. Initiating:

The initiating process group is generally when a project is formally approved and assigned a project manager. The group includes two primary processes: developing the project charter and identifying the project stakeholders.

The two outcomes of this process group are the project charter document and the stakeholder register. The stakeholder register lists who the project stakeholders are, what their stake in the project is, and what they expect in regards to frequency and form of communication.

The project charter should include the business case for the project (why it should be completed), as well as a high-level overview of the project's scope, deliverables, and objectives.

Typically, a project charter will also include:

- Resources required
- Key stakeholders
- A high-level timeline with key milestones
- A high-level cost estimate
- Any known risks, issues, or dependencies

ii. **Planning:**

The planning group is the largest of the five process groups, consisting of 24 processes in total. This group of processes is designed to help you plan your entire project in detail, from the scope, schedule, and budget, through to how you will manage the key stakeholders. The primary outcome of this planning stage is a project management plan (PMP).

For larger projects, the PMP may have sub-plans to further outline some of the critical areas, such as the project schedule or quality management. For smaller projects, processes may simply be covered in separate subsections or fleshed out in an appendix.

The PMP is a "living document" that is updated and revised throughout the project as changes occur.

iii. **Executing:**

The executing group is where most of the action happens on a project. It is also where most of the budget is spent and where the actual project deliverables are produced.

The executing process group includes ten project management processes. It is primarily focused around managing project activities and tasks to ensure progress is occurring, communications are happening, risk responses are being implemented, and stakeholders are being engaged.

The most significant role for the project manager during this phase is directing and managing the project work and managing the project knowledge (requirements documentation, meeting minutes, lessons learned). Other typical responsibilities of the project manager include acquiring project resources, developing and managing the project team, and managing communications.

iv. **Controlling and monitoring:**

The controlling and monitoring process group is the second largest, containing twelve project processes. These processes happen throughout the entire project and are in place to ensure there is sufficient oversight. This will also help identify and mitigate any potential issues.

Inevitably, something unexpected will come up during the project life cycle. The processes in this process group are designed to help you update the plan, modify your team's activities, and get everything back on track.

One of the essential processes in this group is monitoring the project work. This requires the tracking of the overall project and its key aspects. This process is critical in limiting overages and project errors. Often, project management software is used to monitor and report on progress.

v. **Closing:**

The closing process group only has one primary process: close out the project or phase. This process involves ensuring the customer has accepted all final phase or project deliverables. Documentation should also be completed and stored and any loose ends of the project or phase should be tied up.

20. Consider a project with the following functional units:

- **Number of user inputs=50**
- **number of user outputs=40**
- **number of user enquiries=35**
- **Number of user files=06**
- **Number of external**

Assuming all complexity adjustment factors and weighing factors as average, what is the function point for the project?

Explanation:

Step-1: As complexity adjustment factor is average (given in question), hence,

```
scale = 3.
F = 14 * 3 = 42
```

Step-2:

```
CAF = 0.65 + ( 0.01 * 42 ) = 1.07
```

Step-3: As weighting factors are also average (given in question) hence we will multiply each individual function point to corresponding values in TABLE.

```
UFP = (50*4) + (40*5) + (35*4) + (6*10) + (4*7) = 628
```

Step-4:

```
Function Point = 628 * 1.07 = 671.96
```

Counting Function Point (FP):

Step-1:

```
F = 14 * scale
```

Scale varies from 0 to 5 according to character of Complexity Adjustment Factor (CAF). Below table shows scale:

```
0 - No Influence
1 - Incidental
2 - Moderate
3 - Average
4 - Significant
5 – Essential
```

Step-2: Calculate Complexity Adjustment Factor (CAF).

```
CAF = 0.65 + ( 0.01 * F )
```

Step-3: Calculate Unadjusted Function Point (UFP).

TABLE (Required)

FUNCTION UNITS	LOW	AVG	HIGH
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

Multiply each individual function point to corresponding values in TABLE.

Step-4: Calculate Function Point.

```
FP = UFP * CAF
```

21. Write a note on organizational structures. Explain its types along with advantages and disadvantages for each type?

Organizational structure is an enterprise environmental factor, which can affect the availability of resources and influence how projects are conducted. Organizational structures range from functional to projectized, with a variety of matrix structures in between as follows:

1. Functional (Centralized) Organizational Structure:

This structure of organization is based on hierarchical structure of functions. A functional manger in this kind of organization has complete authority. All of the project work typically happens within a particular department, and that department's manager is completely in charge of everything.

- The project team members report to a functional manager.
- Project management decisions have to be cleared with functional managers.
- Project managers are assistants to the functional managers in getting the work done.
- Project managers spend a lot of time doing administrative tasks.
- They are called project expediters in functional organizations.

Advantages:

- Broad specialist base available
- Shared knowledge of latest technology
- Flexible resource scheduling

Disadvantages:

- Less Project Manager authority
- Complex coordination
- No one person responsible for an entire project

2. Matrix Organizational Structure:

Matrix organizations as the name suggests minimizes the difference between Functional and the Projectized Organizations. It also takes advantage of the strengths of these two set-ups. The gap however is so big that even the Matrix organization had to be divided in to three categories – Weak Matrix, Balanced Matrix and Strong Matrix:

- **Weak Matrix–** This structure is bent towards functional structure. The decisions making still happens with the functional manager's cooperation or approval. The project managers do have some authority; however, they hold close to no authority over the resources on a project. Project expediters and project coordinators can work in weak matrix organizations, too.
- **Balanced Matrix –** Clearly the authority level is balanced. Project managers share authority with the functional managers. Project manager, however, has to run his people management decisions by the functional manager, but vice versa, the functional manager too have to run the project decisions by the project manager. Resources working in a balanced matrix organization report to a project manager and a functional manager equally.
- **Strong Matrix –** A strong matrix organization has its focus majorly on the delivery of the project. In this set-up, the project managers have more authority than the functional managers. The people management decision making lies with the functional manager and the team still reports to both managers. The team is appraised based on their project performance as well as their functional expertise.

Advantage:

- Good control over projects
- Better coordination between the departments
- Identifying the problems early
- Change management becomes easier and in a timely manner

Disadvantage:

- Duplicate Reporting – team members have two different bosses, one their functional manager, another project manager of the project they have been assigned to.
- Sometimes conflicts would arise between departments.
- Information and workflow in multiple directions

3. Projectized Organizational Structure:

In a Projectized set-up, every entity is organized around projects. Once a project is complete, the team is released. The resources of that project are reassigned to another project. The project manager makes every crucial decision including that on budget. All of the decisions about a project’s schedule, quality, and resources are taken by the project manager. All said and done, the complete accountability lies with the project manager and he becomes responsible for the success or failure of the project. For example, a contractor or a consulting company is organized like this.

Advantage:

- Team reports to one manager
- No multi-direction flow, only single point focus on projects
- Change response happen rapidly
- High project commitment
- Accountability for strong and disciplined line of communication and status updates.

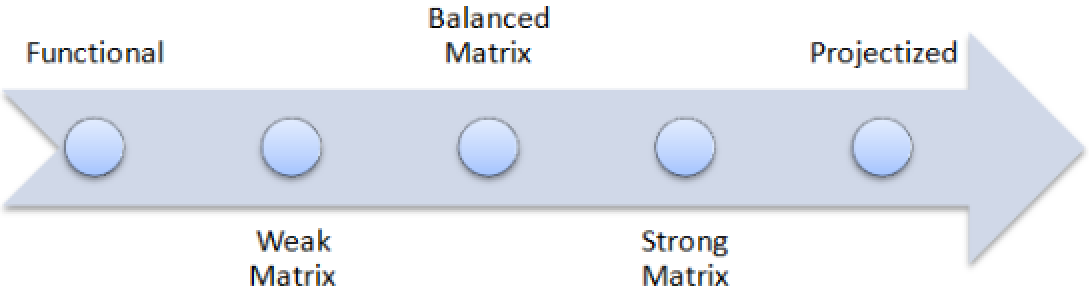
Disadvantage:

- High cost to the enterprise
- Team members cannot appraise their skill sets
- Duplication of resources
- Career insecurity

4. Composite Organizational Structure:

The last type of organization, a composite organization, is one that changes the authority level of the project manager from project to project. In one project, it might be more like that of Projectized organization, and in another project, more like a functional organization.

The diagram below depicts how from Functional to Projectized, the authority of a Project Manager goes on increasing.



22. What is preliminary investigation? What’s the importance of this investigation? At which point during SDLC this investigation is being done? What is the end product of this investigation?

Preliminary Investigation:

Preliminary investigation is the first phase of the systems development life cycle in which a brief feasibility study is performed to assess whether or not a full-scale project should be undertaken.

The preliminary investigation is carried out to determine the scope and objectives of the new system and to investigate whether there is a feasible solution. New applications normally originate from end-user requests and are weighed against the other requests for IS resources before approval to develop the system is granted. At this stage an analyst or small project team is authorized to investigate the real potential of the new application. During this brief study the analyst must investigate the problem and the existing system sufficiently to be able to identify the true extent and purpose of the new application.

The purpose of the preliminary investigation is to determine whether the problem or deficiency in the current system really exists. The project team may reexamine some of the feasibility aspects of the project. At this point, the purpose is to make a “go” or “no-go” decision.

The output from this preliminary investigation is a statement of scope and objectives (often termed the project charter) together with a feasibility report. This document is submitted to management where a decision is made as to whether or not the development project should continue.

23. Write a note on any two of the following:

(a) Skill requirements for the project manager.

1. Communication

Project managers must have strong communication skills to be able to convey messages to clients and team members. They need this skill to effectively share their vision, goals, ideas and issues. They also need communication skills to produce presentations and reports.

2. Leadership

Strong leadership skills are critical for project managers. They allow leaders to oversee and coordinate tasks as well as motivate and encourage the team and define the road map to successfully complete the project.

3. Team management

Besides leading a team from a strategic perspective, project managers also need to manage from an operational point of view. An effective team manager excels at administering and coordinating groups of individuals by promoting teamwork, delegating tasks, resolving conflict, setting goals, and evaluating performance. Leadership is about inspiring others to walk with you; team management makes sure your team has the right shoes.

4. Organization

To ensure processes are running smoothly and in line with common goals, project managers must have strong organizational skills. While this includes the ability to multitask, it also includes prioritizing tasks, compartmentalizing projects and documenting everything for easy access and future reference.

5. Negotiation

A project manager must be effective at negotiating terms with suppliers, clients and other stakeholders. You must also employ negotiation skills when working with your team as well to bring everyone in line with strategic goals or manage interpersonal conflicts within the team.

6. Problem-solving

A project manager must be able to gather information, weigh the associated pros and cons and then formulate the best solution. Strong problem-solving skills will allow project managers to have a structured approach to solving problems to achieve a positive result.

(b) Process groups in project management.

Previously explained….

(c) Formulation of network model.

The first stage in creating a network model is to represent the activities and their interrelationships as a graph. In CPM we do this by representing activities as links (arrowed lines) in the graph - the nodes (circles) representing the events of activities starting and finishing.

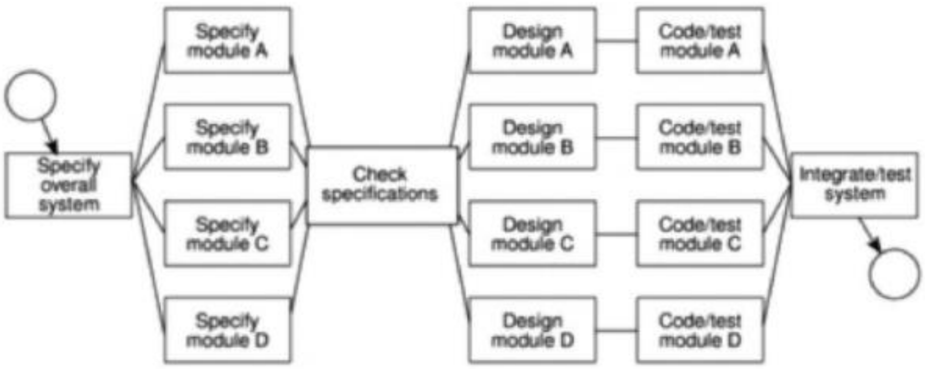


Figure 6.7 *The IOE maintenance group accounts project activity network fragment with a checkpoint activity added.*

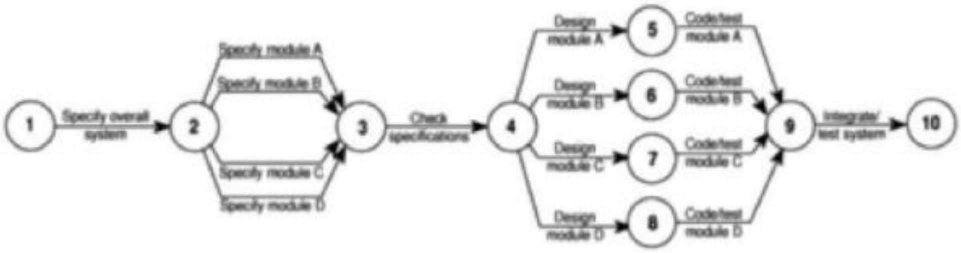


Figure 6.8 *The IOE maintenance group accounts project activity network fragment represented as a CPM network.*

Constructing CPM networks:

Before we look at how CPM networks are used, it is worth spending a few moments considering the rules for their construction.

i. **A project network may have only one start node:**
The start node (node 1 in Figure 6.8) designates the point at which the project may start. All activities coming from that node may start immediately resources are available - that is, they do not have to wait for any other activities to be completed.

ii. **A project network may have only one end node:**
The end node designates the completion of the project and a project may only finish once! The end node for the project fragment shown in Figure 6.8 is the one numbered 10.

iii. **A link has duration:**
A link represents an activity and, in general, activities take time to execute. Notice, however, that the network in Figure 6.8 does not contain any reference to durations. The links are not drawn in any way to represent the activity durations. The network drawing merely represents the logic of the project - the rules governing the order in which activities are to be carried out.

iv. **Nodes have no duration:**
Nodes are events and, as such, are instantaneous points in time. The source node is the event of the project becoming ready to start and the sink node is the event of the project becoming completed. Intermediate nodes represent two simultaneous events - the event of all activities

leading in to a node having been completed and the event of all activities leading out of that node being in a position to be started.

In Figure 6.9 node 3 is the event that both coding and data take-on have been completed and activity program testing is free to start. Installation may be started only when event 4 has been achieved, that is, as soon as program testing has been completed.

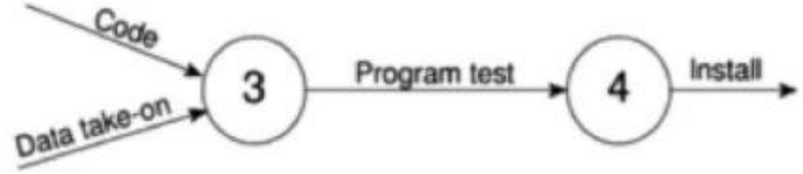


Figure 6.9 *Fragment of a CPM network.*

v. **Time moves from left to right:**
If at all possible, networks are drawn so that time moves from left to right. It is rare that this convention needs to be flouted but, in any case, the arrows on the activity lines give a strong visual indication of the time flow of the project.

vi. **Nodes are numbered sequentially:**
There are no precise rules about node numbering but nodes should be numbered so that head nodes (those at the 'arrow' end of an activity) always have a higher number than tail events (those at the 'non-arrow' end of an activity. This convention makes it easy to spot loops.

vii. **A network may not contain loops:**
Figure 6.10 demonstrates a loop in a CPM network. 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. Note that the logic of Figure 6.10 suggests that program testing cannot start until the errors have been corrected.

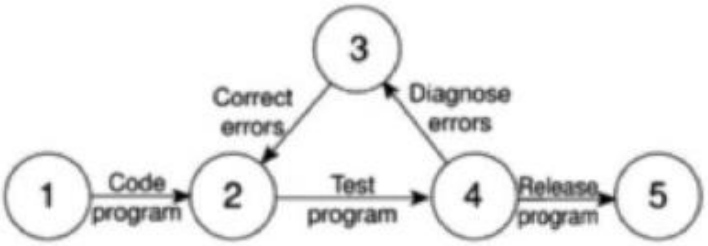


Figure 6.10 *A loop represents an impossible sequence.*

If we know the number of times we expect to repeat a set of activities, a test-diagnose-correct sequence, for example, then we can draw that set of activities as a straight sequence, repeating it the appropriate number of times. If we do not know how many times a sequence is going to be repeated then we cannot calculate the duration of the project unless we adopt an alternative strategy such as redefining the complete sequence as a single activity and estimating how long it will take to complete it.

viii. **A network may not contain dangles:**
A dangling activity such as Write user manual in Figure 6.11 cannot exist, as it would suggest there are two completion points for the project. If, in Figure 6.11 node 5 represents the true project completion point and there are no activities dependent on activity Write user manual, then the network should be redrawn so that activity Write user manual starts at node 2 and terminates at node 5 - in practice, we would need to insert a dummy activity between nodes 3

and 5 as described in Section 6.9. In other words, all events, except the first and the last, must have at least one activity entering them and at least one activity leaving them and all activities must start and end with an event.

ix. **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.

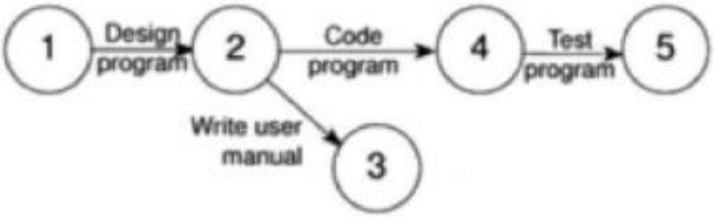


Figure 6.11 *A dangle.*

24. **What is difference between ROI (return or investigation) and payback period? Explain with example?**

Payback Period gives you the number of years would you be paid back the amount you invested. Example: I have invested 10,000$ and every year, I am getting say 1000$, so the payback period is 10 years. (1000$ * 10years = 10,000$)

Return on Investment gives you the AMOUNT you would get in return as a result of your investment.

Example: I have invested 15000$ and return on investment in terms of percentage is 5% after 3 years, so Return on Investment after 3 years= 0.05 * 15000 = 750$

The payback period is the period of time over which the return is received. The return on investment is the amount of money received from your investment.

25. **Suppose you are managing a software development project. The project is expected to be completed in 6 months at a cost of $15,000 per months, after 4 months, planned completion should have been 70%. But you realize that the project is 50% completed at a cost of $70,000.**

(A). Find BCWP, BCWS, and ACWP

(B). Calculate SPI and CPI and interpret these indexes?

(C) Calculate EAC for this project.

<u>SOL(A):</u>

ACWP (Actual cost of work performed) = $70,000

BCWP (Budget cost of work performed) = $45,000

BCWS (Budget cost of work scheduled) = $60,000

<u>SOL(B):</u>

CPI = Cost Performance Index = BCWP/ACWP

CPI = $45,000 / $70,000

CPI = $0.64

AS CPI = 0.64 < 1 so, project is over the budget.

SPI = Schedule Performance Index = BCWP/BCWS

SPI = $45,000 / $60,000

SPI = 0.75

As SPI = 0.75 < 1 so, project is behind the schedule.

SOL(C):

EAC = Estimated (cost) At Completion

EAC = BAC/CPI = BACxACWP/BCWPs

EAC =

EV = Earned Value = percent complete x corresponding budget

BCWS = Budgeted Cost of Work Scheduled, now more properly known as the Planned Value (PV)

ACWP = Actual Cost of Work Performed, i.e. the actual cost

BCWP = Budgeted Cost of Work Performed, now more properly known as the Earned Value (EV)

CV = Cost Variance = earned minus actual = BCWP-ACWP

SV = Schedule Variance = earned minus budget = BCWP-BCWS

BCWS-ACWP = Spending Variance = budget minus actual

BAC = Budget At Completion

EAC = Estimated (cost) At Completion

EAC = variance at completion

CPI = Cost Performance Index = BCWP/ACWP

SPI = Schedule Performance Index = BCWP/BCWS

Combined cost-schedule index = CPI x SPI

You can extrapolate performance to date to calculate the EAC as follows:

EAC = BAC/CPI = BACxACWP/BCWP

26. Consider the following scenario and answer the following questions:

Activities	Predecessor	Duration
A	-	10
B	-	15
C	A	5
D	B	12
E	C,D	14
F	B	8
G	D,F	15
H	E	10
I	E,G	6
J	F,I	9

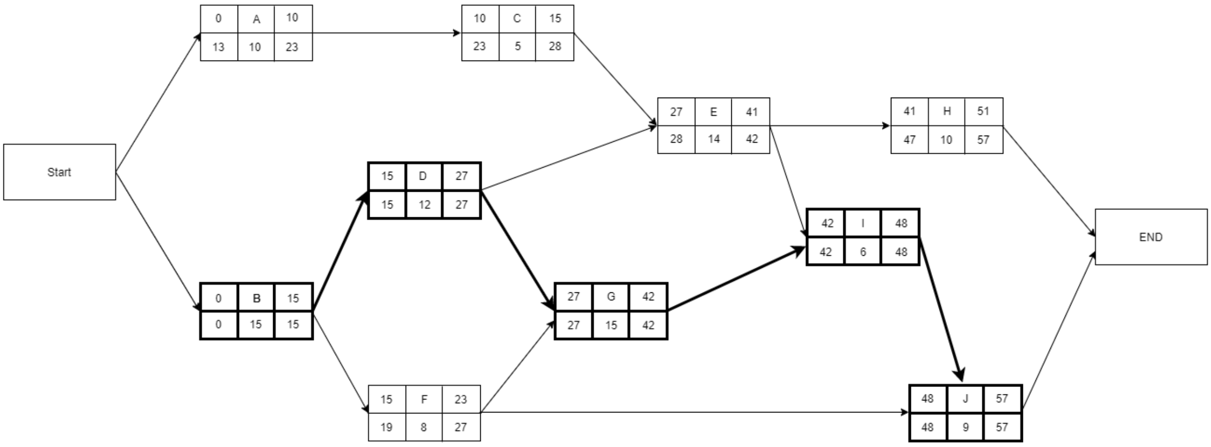
(A). Draw the project network diagram.

(B). Develop the project schedule(EST, EFT, LST, LFT, Slack).

(C). what are the critical activities and critical path?

(D). what is project completion duration? [57]

(E). if there is an option to delay one activity without delaying the entire merge project, which activity would you delay and why? [ ]



Activities	Predecessor	Duration	EST	EFT	LST	LFT	Slack
A	**-**	**10**	0	10	13	23	13
B	**-**	**15**	0	15	0	15	0
C	**A**	**5**	10	15	23	28	13
D	**B**	**12**	15	27	15	27	0
E	**C,D**	**14**	27	41	28	42	1
F	**B**	**8**	15	23	19	27	4
G	**D,F**	**15**	27	42	27	42	0
H	**E**	**10**	41	51	47	57	6
I	**E,G**	**6**	42	48	42	48	0
J	**F,I**	**9**	48	57	48	57	0

27. Calculate the early start, early finish, of the activities using backward pass. Also identify the critical path. Draw the activity network diagram.

Activity Id	Activity Description	Durations (in Weeks)	Precedents
1	**Requirement gathering**	**2**	
2	**Proposal Approval**	**2**	**1**
3	**Development**	**6**	**2**
4	**Integration**	**2**	**3**
5	**Testing**	**2**	**4**
6	**Revision**	**1**	**5**
7	**Deployment**	**1**	**6**

28. A project has following activities: Environment Permits, Traffic report, utility Locates and insertion design. To complete this project, the approval of these activities (Environment Permits, Traffic report, and insertion design) is given by project manager. The project manager is also informed the utility Locates. Environmental coordinator is responsible for environment permits and he is also consulted in intersection design. Traffic officer is responsible for traffic report and intersection design and also informed utility locates. Utility officer is responsible for utility locates and consulted for intersection design.

(a) Identify the role and responsibilities and draw RAM for this project?

29. Find critical path by performing complete forward and backward pass.

Activity	Description	Preceding Activity	Activity duration
A	Approval of application	None	5
B	Construction plan	A	15
C	Traffic study	A	10
D	Service availability check	A	5
E	Staff report	B, C	15
F	Commission Approval	B, C, D	10
G	Wait for construction	F	170
H	Occupancy	E, G	35

30. Draw an activity network using backward pass, for the activities mentioned below. Identify the critical path?

S/NO	Activity	Duration (Days)	Depends on
L	Planning	3	
M	Requirements Confirming	4	L
N	Project Designing	7	M
O	Implementation	6	L,N
P	Testing	5	L,N,O
Q	Integration	3	M,N,O,P

31. Draw an activity network using forward pass, for the activities mentioned below. Identify the critical path.

S/NO	Activity	Duration(Days)	Depends on
A	Requirement gathering	8	
B	Requirement checking	3	A
C	Project designing	8	B
D	Implementation	12	A, C
E	Testing	3	A, C, D
F	Integration	4	A, B, C, D