

Topics Covered:

- ❖ Software engineering problem and software product
- ❖ Software product attributes
- ❖ Definition of a Software Project (SP)
- ❖ SP Vs. other types of projects activities covered by SPM,
- ❖ Categorizing SPs
- ❖ Project management cycle
- ❖ SPM framework
- ❖ Types of project plan

SOFTWARE ENGINEERING PROBLEM AND SOFTWARE PRODUCT

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product**.

Engineering on the other hand, is all about developing products, using well-defined, scientific principles and methods.

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

PROBLEMS RELATED TO SOFTWARE ENGINEERING

1. **The Problem of scale:** A fundamental problem of software engineering is the problem of scale; development of a very large system requires a very different set of methods compared to developing a small system. While dealing with a small software project, the technology requirement is low and the project management requirement is also low. However, when the scale changes to large systems, to solve such problems properly, it is essential that, and the project management for the development project also needs to be more formal.
2. **Cost, schedule and quality:** The cost of developing a system is the cost of the resources used for the system, which, in the case of software, are the manpower, hardware, software, and the other support resources. Generally, the manpower component is predominant, as software development is largely labor-intensive and the cost of the computing systems is now quite low. Hence, the cost of software project is measured in terms of person-months, i.e. the cost is considered to be the total number of person-months spent in the project. Schedule is an important factor in many projects. Business trends are dictating that the time to market of a product should be reduced; that is, the cycle time from concept to delivery should be small. Any business with such a



requirement will also require that the cycle time for building a software needed by the business be small.

One of the major factors driving any production discipline is quality. We can view quality of a software product as having three dimensions:

- a. Product Operation
 - b. Product Transition
 - c. Product Revision
3. **The Problem of consistency:** Though high quality, low cost and small cycle time are the primary objectives of any project, for an organization there is another goal: consistency. An organization involved in software development does not just want low cost and high quality for a project, but it wants these consistently.

SOFTWARE PRODUCT ATTRIBUTES



1. **Performance** shows the response of the system to performing certain actions for a certain period of time. There are two ways how to measure performance:
 - a. **Latency:** Time spent on responding to an event
 - b. **Channel capacity.** The number of events that occur at a certain point in time.

In practice, the possible performance indicators include, for example:

- c. Average/maximum number of system users per time unit.
 - d. Average page load time.
 - e. Average method execution time.
2. **Interoperability** is an attribute of the system or part of the system that is responsible for its operation and the transmission of data and its exchange with other external systems. A well-designed system facilitates integration with third-party systems. To improve the interoperability, you can use well-designed external interfaces, standardization systems, etc.

Naturally, there are a lot of problems for interaction:

- a. Outdated external systems.
 - b. Different formats of data in similar external systems.
 - c. Different versions of the API in external systems.
 - d. Backwards compatibility of the API for integration.
 - e. Poor quality and lack of standards of external systems.
3. **Usability** is one of the most important attributes, because, unlike in cases with other attributes, users can see directly how well this attribute of the system is worked out. One of the key problems of usability is too much interaction or too many actions necessary to accomplish a task. Incorrect sequences of steps in multistage interfaces are also a problem of usability. Data elements and controls may be designed not according to the accepted patterns of user experience, which also complicates the interaction. For example, if you are developing an iOS application, then it is important to use the guidelines from Apple, or the guidelines from Microsoft—for Windows desktop applications.

Examples of important indicators for this attribute are:

- a. List of supported devices, OS versions, screen resolutions, and browsers and their versions.

- b. Elements that accelerate user interaction, such as “hot keys”, “lists of suggestions”, and so on.
 - c. Average time a user needs to perform individual actions.
 - d. Support of accessibility for people with disabilities.
4. **Reliability** is an attribute of the system responsible for the ability to continue to operate under predefined conditions. Most often, the system fails due to the inaccessibility of external elements, such as databases, systems, and network connections.
5. **Availability** is part of reliability and is expressed as the ratio of the available system time to the total working time. Important indicators for this attribute are:
- a. Availability.
 - b. Planned downtime.
 - c. Time needed to update the software, and so on.
6. **Security** is responsible for the ability of the system to reduce the likelihood of malicious or accidental actions as well as the possibility of theft or loss of information. There are a number of measures that are used to protect systems: authentication, encryption, audit, and others. Examples of this attribute in the work of the system are:
- a. The ability of the system to detect DDoS attacks and respond to them.
 - b. Restrictions of user access in accordance with authentication/authorization.
 - c. Prevention of SQL injection.
 - d. Encryption of passwords and content.
 - e. Secure connection.
7. **Maintainability** is the ability of the system to support changes. Changes can be related to new business requirements or correction of old errors and affect system components or separate methods. Also, maintainability affects the time needed to restore the system after a failure. Excessive dependencies between components have a very negative effect on maintainability.
8. **Modifiability** determines how many common changes need to be made to the system to make changes to each individual item. Ideal is the case where each change affects only one element.
9. **Testability** shows how well the system allows performing tests, according to predefined criteria. In addition to testing performance, testability makes it possible to effectively divide the system into subsystems. The main indicators for this attribute are:
- a. Percentage of coverage with modular, integration, or unit tests.
 - b. The final list of required test environments as well as the final list of used approaches to testing (manual/automatic, regression, integration, etc.)
10. **Scalability** is the ability of the system to handle load increases without decreasing performance, or the possibility to rapidly increase the load.
- There are **two ways to improve scalability**:
- a. **Vertical:** To increase, we add more resources, such as memory, disks or processors into one system.
 - b. **Horizontal:** We increase the number of computing units and divide the load.
11. **Reusability** is a chance of using a component or system in other components/systems with small or no change. Allocation of responsibilities, modularization, decreasing of copy-paste are all about reusability. Copying code, or worse, using different components for the same result in different places, is one of the biggest problems of reusability.
12. **Supportability** is the ability of the system to provide useful information for identifying and solving problems. The main problems in ensuring supportability can be addressed with the following means:
- a. **No diagnosis:** How the activity and performance of the system are controlled. This includes various types of logging.

- b. **No tools for troubleshooting:** This includes backups, various systems for creating snapshots of the system, and tools for auditing the system. When the system fails, it is always more pleasant to wait for an automatic restart than to solve the issue manually.
- c. **No health checking:** This includes a variety of systems for measuring compilation time, deployment time, database size, or mobile application size.

A **Software Project** is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

Software Project Management is the art and science of planning and leading software projects. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

Fundamentals of Software Project Management

- clearly defining the vision for the end result
- gathering the requirements in detail
- breaking down the work effort into tasks
- establishing measurable milestones that the team accepts
- obtaining commitment on meeting the project objectives
- tracking progress using metrics that are simple and fair
- implementing quality controls that everyone understands
- staffing the project appropriately to meet the project goals
- selecting adequate tools and training to get the job done
- assessing risks and mitigating them

STAKEHOLDERS: These are people who have a stake or interest in the project. Stakeholders can be categorized as:

- ❖ Internal to the project team: They will be under the direct managerial control of the project leader.
- ❖ External to the project team but within the same organization: The project leader might need the assistance of the users to carry out systems testing.
- ❖ External to both the project team and the organization: External stakeholders may be customers (users) who will benefit from the system that the project implements. The relation here is usually based on contract.

Stakeholders include

- The project sponsor and project team
- Support staff
- Customers
- Users
- Suppliers

SP VS. OTHER TYPES OF PROJECTS ACTIVITIES COVERED BY SPM

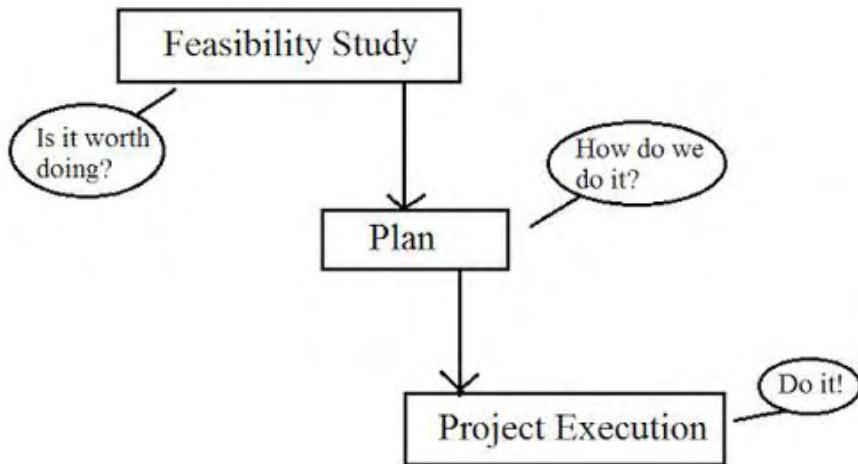
Many techniques in general project management also apply to software project management, but Fred Brooks identified some characteristics of software projects which make them particularly difficult:

1. **Invisibility:** When a physical artifact such as a bridge is constructed the progress can actually be seen. with software, progress is not immediately visible. Software project management can be the process of making the invisible visible.
2. **Complexity:** Per dollar, pound or euro spent, software products contain more complexity than other engineered artifacts.
3. **Conformity:** The 'traditional' engineer usually works with physical systems and materials like cement and steel. These physical systems have complexity, but are governed by consistent physical laws. Software developers have to conform to the requirement of human clients. It is not just that individuals can be inconsistent. Organizations, because of lapses in collective memory, in internal communication or in effective decision making, can exhibit remarkable, 'organizational stupidity'.

4. **Flexibility:** That software is easy to change is seen as a strength. However, where the software system interfaces with a physical or organizational system, it is accommodating the other components rather than vice versa. Thus software systems are particularly subject to change.

ACTIVITIES COVERED BY SOFTWARE PROJECT MANAGEMENT

A software project is not only concerned with the actual writing of software. In fact, where a software application is bought "off the shelf", there may be no software writing as such, but this is still fundamentally a software project because so many of the other activities associated with software will still be present.



The feasibility Study: - assesses whether a project is worth starting—that it has a valid 'business case'. Information is gathered about the requirements of the proposed application. Requirements elicitation can, at least initially, be complex and difficult. The stakeholders may know the aim they wish to pursue, but not be sure about the means of achievement. The developmental and the operational costs, and the value of the benefits of the new system, will also have to be estimated.

Planning: - If the feasibility study indicates that the prospective project appears viable, then project planning can start. We create an outline plan for the whole project and a detailed one for the first stage. Because we will have more detailed and accurate project information after the earlier stages of the project have been completed, planning of the later stages is left to nearer their start.

Project execution: - The project can now be executed. The execution of a project often contains design and implementation sub-phases. Design is making decision about the form of the products to be created. This could relate to the external appearance of the software, that is, the user interface, or the internal architecture. The plan details the activities to be carried out to create these products. Planning and design can be confused because at the most detailed level, planning decisions are influenced by design decisions.

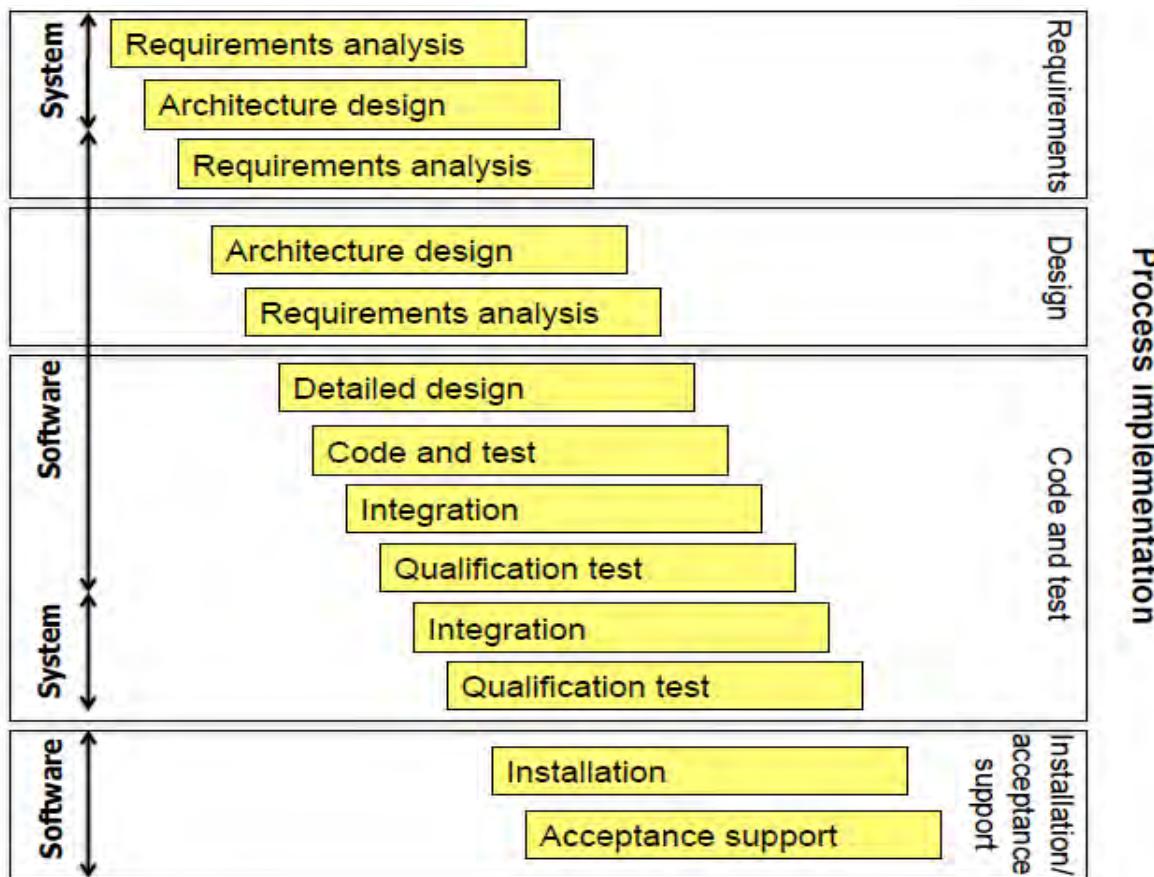


Fig: Project Management with software engineering approach

CATEGORIES SOFTWARE PROJECTS

1. **Compulsory versus voluntary users:** In workplace there are systems that staff have to use if they want to do something such as recording a sale. However, use of a system is increasingly voluntary as in the case of computer game.
2. **Information systems versus embedded systems:** Information system enable staff to carry out office processes e.g. Stock control system. Embedded system control machines. E.g. System might control air conditioning systems in a building.
3. **Outsourced Projects:** While developing a large project, sometimes, it makes good commercial sense for a company to outsource some parts of its work to other companies.
4. **Objectives versus products:** Project 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 project.
 - The project requirement might be meet certain objectives which could be met in a no of ways. An organization might have a problem and ask a specialist to recommend a solution.
 - Many software projects have 2 stages
 - An objective driven project resulting in recommendations, it identifies the need for a new system.
 - A project actually to create a software project.

PROJECT MANAGEMENT CYCLE

The project management cycle consists of five phases: initiation, planning, implementation, monitoring and closure. The cycle represents a continuous process in which each phase provides the foundation for the next.



- Initiation:** A solid initiation will set a project up for success and lay the foundation for all the other stages in its life cycle. During this phase, PMs first measure the feasibility and value of a project in order to determine whether or not it is worth pursuing. Once a project is given the green light, team members will be briefed on the project's objective and assigned goals during the Initiation Phase. PMs should be working closely with their clients and execs to properly prepare for the upcoming planning process. It is also during this phase that PMs should be rallying the team together and building support for the project. One way to do this is to pull the team tighter and to present the project significance and value to them. It will be important to have everyone on board.
- Planning:** The planning stage focuses on building a blueprint for achieving the project's goals, on time and on budget. This roadmap will be used to guide the team through the execution of the project. It is in this phase that the scope is defined and a solid project management plan will be developed. The plan involves identifying costs, available resources, potential financing options, and risks, as well as setting a realistic timeframe. Moreover, it should also include performance measures or baselines to measure progress and determine if the project is on track. During the planning stage, project managers clearly define roles, responsibilities and tasks, so that all team members are aware of what they're accountable for. Here are a few of the essential documents PMs typically create to ensure that everyone knows what needs to be done and that the project progresses properly:
 - Scope statement – a document that clearly describes the project's benefits, objectives, key milestones and deliverables.
 - Work breakdown structure – a diagram that breaks down the project's scope into manageable sections
 - Gantt chart – a project management visual used to illustrate the project timeline and to plan out the tasks identified in the work breakdown diagram.
 - Risk management plan – a document that identifies all foreseeable risks and possible strategies to mitigate them.
 - Communication plan – an essential plan if the project involves outside stakeholders. It should include communication objectives, frequency and methods, as well key content to share with the parties involved in the project. When planning project communications, the best PMs ensure their message will get across by adapting their approach to fit each particular audience. For instance, using simple, familiar PowerPoint visuals when reporting to stakeholders who may not understand PM jargon can be an effective way to share key data.
- Execution:** Execution is the stage that is most commonly associated with actual project management. PMs should expect intensive activity during this time, from allocating

resources and building deliverables, to creating development updates, status reviews and performance reports. Project Managers should arrange a kick-off meeting to officially mark the onset of the execution phase, get the team started on the right track, and ensure everything is properly prepared for team members to begin executing their assignments. The execution phase is active and PMs will be required to leverage their management skills and their soft skills to keep the project team motivated, performing and on track. PMs may need to:

- eliminate all unnecessary distractions or activities
- get underperformers back on track
- manage morale to prevent burnout
- find needed resources to overcome stalls
- solve conflicts that may occur

4. Monitoring and Control: Although it is sequenced as the 4th stage in the project management life cycle, the monitoring phase is actually most often implemented during the execution stage, not afterwards. While the team executes the project plan, PMs begin monitoring and controlling it to ensure progression is on track with the schedule. To achieve this, PMs will be:

- monitoring the tasks that are on the critical path
- verifying and controlling scope creep and taking measures to counter it
- updating stakeholder with status reviews according to the pre-established communication plan
- comparing planned costs versus actual costs
- seeking ways to optimize performance

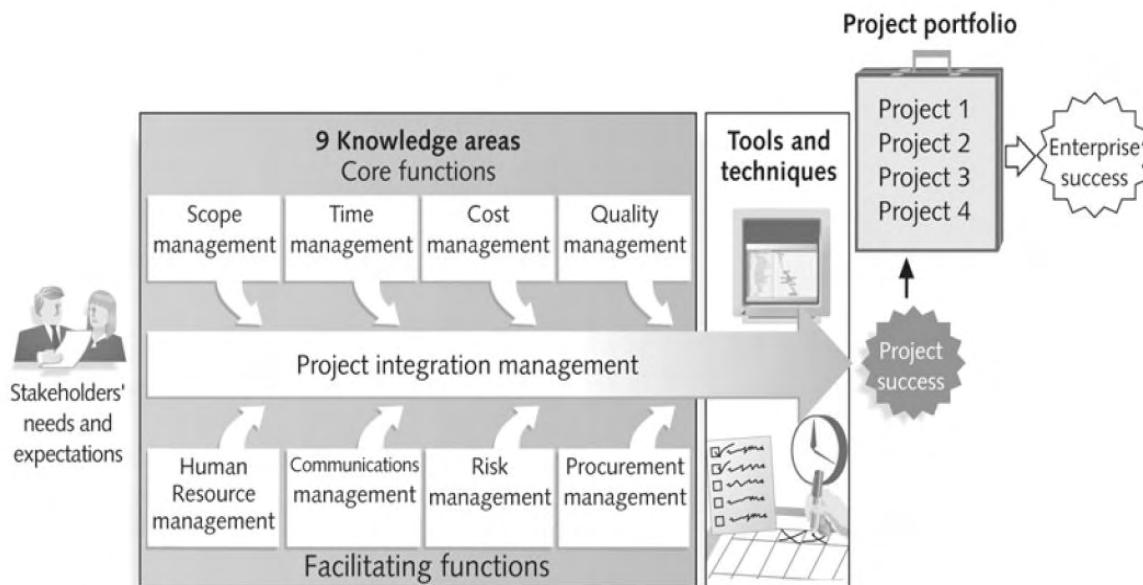
5. Closure: The final stage of the project management life cycle is the closure phase, which requires a series of essential tasks and activities, such as delivering the finished project to the client, communicating its completion to stakeholders, releasing resources, and terminating contractors hired specifically for the project. During the closure stage, PMs also hold a post-mortem meeting to evaluate what went wrong, highlight successes, and learn what improvements can be made for future projects. Using this meeting to recognize and appreciate valuable team members is a best practice that can help build a PM's credibility and brand.

Managing a project, regardless of its magnitude or complexity, can become overwhelming at times. Breaking it down into these 5 phases and mastering each stage can help PMs and their teams handle even the most complex projects successfully.

SOFTWARE PROJECT MANAGEMENT FRAMEWORK

1. **Managing integration:** Projects have all types of activities going on and there is a need to keep the “whole” thing moving collectively – integrating all of the dynamics that take place. Managing integration is about developing the project charter, scope statement, and plan to direct, manage, monitor, and control project change.
2. **Managing scope:** Projects need to have a defined parameter or scope, and this must be broken down and managed through a work breakdown structure or WBS. Managing scope is about planning, definition, WBS creation, verification, and control.
3. **Managing time/schedule:** Projects have a definite beginning and a definite ending date. Therefore, there is a need to manage the budgeted time according to a project schedule. Managing time/schedule is about definition, sequencing, resource and duration estimating, schedule development, and schedule control.

4. **Managing costs:** Projects consume resources, and therefore, there is a need to manage the investment with the realization of creating value (i.e., the benefits derived exceed the amount spent). Managing costs is about resource planning, cost estimating, budgeting, and control.
5. **Managing quality:** Projects involve specific deliverables or work products. These deliverables need to meet project objectives and performance standards. Managing quality is about quality planning, quality assurance, and quality control.
6. **Managing human resources:** Projects consist of teams and you need to manage project team(s) during the life cycle of the project. Finding the right people, managing their outputs, and keeping them on schedule is a big part of managing a project. Managing human resources is about human resources planning, hiring, and developing and managing a project team.
7. **Managing communication:** Projects invariably touch lots of people, not just the end users (customers) who benefit directly from the project outcomes. This can include project participants, managers who oversee the project, and external stakeholders who have an interest in the success of the project. Managing communication is about communications planning, information distribution, performance reporting, and stakeholder management.



8. **Managing integration:** Projects have all types of activities going on and there is a need to keep the “whole” thing moving collectively – integrating all of the dynamics that take place. Managing integration is about developing the project charter, scope statement, and plan to direct, manage, monitor, and control project change.
9. **Managing scope:** Projects need to have a defined parameter or scope, and this must be broken down and managed through a work breakdown structure or WBS. Managing scope is about planning, definition, WBS creation, verification, and control.
10. **Managing time/schedule:** Projects have a definite beginning and a definite ending date. Therefore, there is a need to manage the budgeted time according to a project

schedule. Managing time/schedule is about definition, sequencing, resource and duration estimating, schedule development, and schedule control.

11. **Managing costs:** Projects consume resources, and therefore, there is a need to manage the investment with the realization of creating value (i.e., the benefits derived exceed the amount spent). Managing costs is about resource planning, cost estimating, budgeting, and control.
12. **Managing quality:** Projects involve specific deliverables or work products. These deliverables need to meet project objectives and performance standards. Managing quality is about quality planning, quality assurance, and quality control.
13. **Managing human resources:** Projects consist of teams and you need to manage project team(s) during the life cycle of the project. Finding the right people, managing their outputs, and keeping them on schedule is a big part of managing a project. Managing human resources is about human resources planning, hiring, and developing and managing a project team.
14. **Managing communication:** Projects invariably touch lots of people, not just the end users (customers) who benefit directly from the project outcomes. This can include project participants, managers who oversee the project, and external stakeholders who have an interest in the success of the project. Managing communication is about communications planning, information distribution, performance reporting, and stakeholder management.
15. **Managing risk:** Projects are a discovery-driven process, often uncovering new customer needs and identifying critical issues not previously disclosed. Projects also encounter unexpected events, such as project team members resigning, budgeted resources suddenly changing, the organization becoming unstable, and newer technologies being introduced. There is a real need to properly identify various risks and manage these risks. Managing risk is about risk planning and identification, risk analysis (qualitative and quantitative), risk response (action) planning, and risk monitoring and control.
16. **Managing procurement:** Projects procure the services of outside vendors and contractors, including the purchase of equipment. There is a need to manage how vendors are selected and managed within the project life cycle. Managing procurement is about acquisition and contracting plans, sellers' responses and selections, contract administration, and contract closure.
17. **Managing stakeholders:** Every project impacts people and organizations and is impacted by people and organizations. Identifying these stakeholders early, and as they arise and change throughout the project, is a key success factor. Managing stakeholders is about identifying stakeholders, their interest level, and their potential to influence the project; and managing and controlling the relationships and communications between stakeholders and the project.

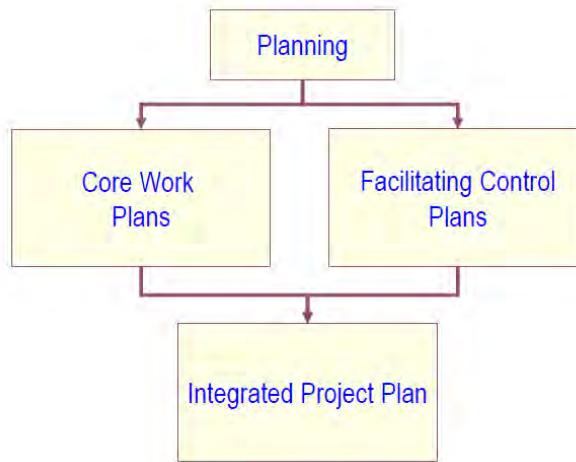
TYPES OF PROJECT PLAN

There are several types of planning involved in any project. That may depend upon the project manager how he defines / sees /fits the project into the overall planning process. Generally, there are three major types of project planning:

1. **Strategic planning (corporate planning):** It involves the high level selection of the project objectives. It is the long term vision of what a project wants to achieve and it is for long term planning. This type of strategic planning is prepared by the top level management.

2. **Tactical planning (divisional planning):** It identifies the divisional activities required to achieve goal. It is prepared for each department or division. It should be consistent with strategic planning. This type of planning is prepared by the middle manager and generally for one year.
3. **Operational planning:** It involves the detailed planning required to meet the strategic objectives. It is the action plan of each activity of the unit. This type of planning is prepared by lower level management. These are short term, monthly or quarterly. Operational planning should be consistent with the tactical planning.

Planning Process Overview: The project planning process is the main tool used to ensure that tasks are completed in timely manner. A project may best be defined as a venture taken to ensure that a deliverable is completed within a specific timeframe and that certain criteria or objectives are met. In order to make certain that a project has the best chance of success and risk of failure has been minimized, a plan is devised to determine the most effective strategy for completion. Simply put, project planning involves the processes used to ensure that tasks are completed in an efficient manner. Those in positions of leadership often need to understand various project planning process steps and tools in order to facilitate communication between team members and ensure that all responsibilities are handled.



Structure of a Software Project Management plan: Projects require a plan of what is intended and a description of what happens. Following is pro-forma that can be used for a project plan.

Project	What is the name of the project ?
Background	How did this project arise ? How was it identified ? How has it evolved ?
Description	What is a brief description of the project ?
Values	What values underpin this project ?
Target group & their needs	Who will benefit ? What are their needs ?
Aims	What are the overall aims this project is trying to achieve ?
Outcomes hierarchy	What is the outcomes hierarchy that will lead to the achievement of these aims ?
Objectives	What are the specific objectives this project will achieve ?
Strategies/ steps	What are the principal strategies or steps that will be required to make this project happen ? How will the project work ?
Management	Who is responsible for the management of the project ?
Funding	What financial resources are required and where are they coming from ?
Time frame	What is the time frame ?
Action	How has to do what to make the project happen ? By when ?
Evaluation	How will the project be evaluated ?

Topics Covered:

- ❖ Project life cycle and product life cycle
- ❖ Project planning and scheduling
- ❖ Resource allocation

Product management and project management are both essential concepts for companies aiming to stay competitive in today's business landscape. However, project management and product management are often confused by even the most experienced business professionals.

A **product** is anything that can be offered to a market to solve a problem, or to satisfy a want or need. Products have a life cycle that consists of multiple stages. First the product is conceived, then developed, then introduced and managed in the market, and finally the product is retired when the need for it diminishes.

A **project** is a temporary endeavor that is undertaken to create a unique product or service. With a project, there is a clear definition of what needs to be delivered by a specified date in time.

It's important to note that a product can only be developed within the context of a project, and multiple projects can occur within a product's life cycle.

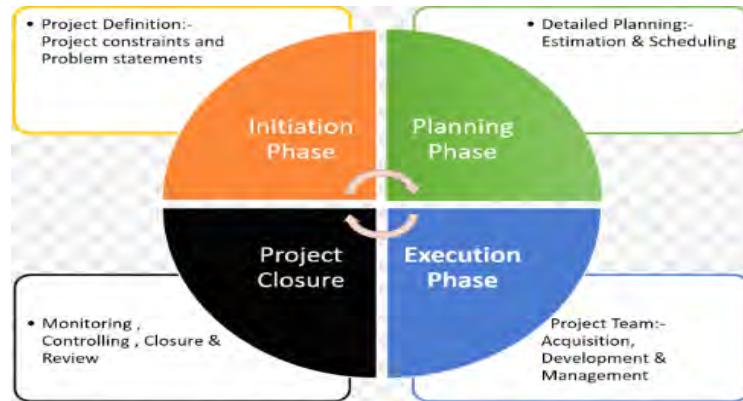
Product management is an organizational function within a company that deals with the planning, forecasting, and production or marketing of a product or products at all stages of the product life cycle.

Project management, on the other hand, consists of the application of processes, methods, knowledge, skills, and experience to achieve the objectives of a specific project.

Project Management Life Cycle

Whether you're working on a small project with modest business goals or a large, multi-departmental initiative with sweeping corporate implications, understanding the project management life cycle is essential. Every project has essential milestones at the beginning, in the middle, and at the end, following a path from initiation to completion to evaluation.

The project management life cycle is usually broken down into four phases: initiation, planning, execution, and closure—these make up the path that takes your project from the beginning to the end. Some methodologies also include a fifth phase, controlling or monitoring. For our purposes, this phase is covered under the execution and closure phases.



1. Initiation Phase

During the first of these phases, the initiation phase, the project objective or need is identified; this can be a business problem or opportunity. An appropriate response to the need is documented in a business case with recommended solution options. A feasibility study is conducted to investigate whether each option addresses the project objective and a final recommended solution is determined. Issues of feasibility ("can we do the project?") and justification ("should we do the project?") are addressed.

Once the recommended solution is approved, a project is initiated to deliver the approved solution and a project manager is appointed. The major deliverables and the participating work groups are

identified, and the project team begins to take shape. Approval is then sought by the project manager to move onto the detailed planning phase.

2. Planning Phase

The next phase, the planning phase, is where the project solution is further developed in as much detail as possible and the steps necessary to meet the project's objective are planned. In this step, the team identifies all of the work to be done. The project's tasks and resource requirements are identified, along with the strategy for producing them. This is also referred to as "scope management." A project plan is created outlining the activities, tasks, dependencies, and timeframes. The project manager coordinates the preparation of a project budget by providing cost estimates for the labor, equipment, and materials costs. The budget is used to monitor and control cost expenditures during project implementation.

Once the project team has identified the work, prepared the schedule, and estimated the costs, the three fundamental components of the planning process are complete. This is an excellent time to identify and try to deal with anything that might pose a threat to the successful completion of the project. This is called risk management. In risk management, "high-threat" potential problems are identified along with the action that is to be taken on each high-threat potential problem, either to reduce the probability that the problem will occur or to reduce the impact on the project if it does occur. This is also a good time to identify all project stakeholders and establish a communication plan describing the information needed and the delivery method to be used to keep the stakeholders informed.

Finally, you will want to document a quality plan, providing quality targets, assurance, and control measures, along with an acceptance plan, listing the criteria to be met to gain customer acceptance. At this point, the project would have been planned in detail and is ready to be executed.

3. Implementation (Execution) Phase

During the third phase, the implementation phase, the project plan is put into motion and the work of the project is performed. It is important to maintain control and communicate as needed during implementation. Progress is continuously monitored and appropriate adjustments are made and recorded as variances from the original plan. In any project, a project manager spends most of the time in this step. During project implementation, people are carrying out the tasks, and progress information is being reported through regular team meetings. The project manager uses this information to maintain control over the direction of the project by comparing the progress reports with the project plan to measure the performance of the project activities and take corrective action as needed. The first course of action should always be to bring the project back on course (i.e., to return it to the original plan). If that cannot happen, the team should record variations from the original plan and record and publish modifications to the plan. Throughout this step, project sponsors and other key stakeholders should be kept informed of the project's status according to the agreed-on frequency and format of communication. The plan should be updated and published on a regular basis.

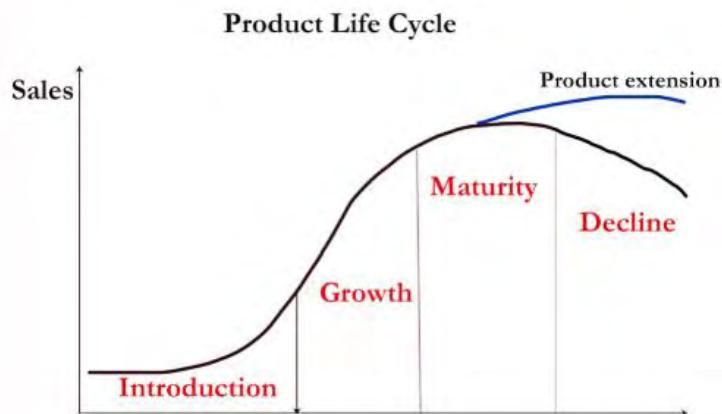
Status reports should always emphasize the anticipated end point in terms of cost, schedule, and quality of deliverables. Each project deliverable produced should be reviewed for quality and measured against the acceptance criteria. Once all of the deliverables have been produced and the customer has accepted the final solution, the project is ready for closure.

4. Closure Phase

During the final closure, or completion phase, the emphasis is on releasing the final deliverables to the customer, handing over project documentation to the business, terminating supplier contracts, releasing project resources, and communicating the closure of the project to all stakeholders. The last remaining step is to conduct lessons-learned studies to examine what went well and what didn't. Through this type of analysis, the wisdom of experience is transferred back to the project organization, which will help future project teams.

Product Life Cycle

The product life cycle describes the period of time over which an item is developed, brought to market and eventually removed from the market. The cycle is broken into four stages: introduction, growth, maturity and decline. The idea of the product life cycle is used in marketing to decide when it is appropriate to advertise, reduce prices, explore new markets or create new packaging.



1. **Introduction Stage** – This stage of the cycle could be the most expensive for a company launching a new product. The size of the market for the product is small, which means sales are low, although they will be increasing. On the other hand, the cost of things like research and development, consumer testing, and the marketing needed to launch the product can be very high, especially if it's a competitive sector.
2. **Growth Stage** – The growth stage is typically characterized by a strong growth in sales and profits, and because the company can start to benefit from economies of scale in production, the profit margins, as well as the overall amount of profit, will increase. This makes it possible for businesses to invest more money in the promotional activity to maximize the potential of this growth stage.
3. **Maturity Stage** – During the maturity stage, the product is established and the aim for the manufacturer is now to maintain the market share they have built up. This is probably the most competitive time for most products and businesses need to invest wisely in any marketing they undertake. They also need to consider any product modifications or improvements to the production process which might give them a competitive advantage.
4. **Decline Stage** – Eventually, the market for a product will start to shrink, and this is what's known as the decline stage. This shrinkage could be due to the market becoming saturated (i.e. all the customers who will buy the product have already purchased it), or because the consumers are switching to a different type of product. While this decline may be inevitable, it may still be possible for companies to make some profit by switching to less-expensive production methods and cheaper markets.

Project Planning

Project planning is a procedural step in project management, where required documentation is created to ensure successful project completion. Documentation includes all actions required to define, prepare, integrate and coordinate additional plans. The project plan clearly defines how the project is executed, monitored, controlled and closed. Project planning is never truly finished until a project is completed.

A well-prepared plan should address and consequently answer the following questions:

Why? -This question is about the reasons why a project is sponsored and the problem it addresses.

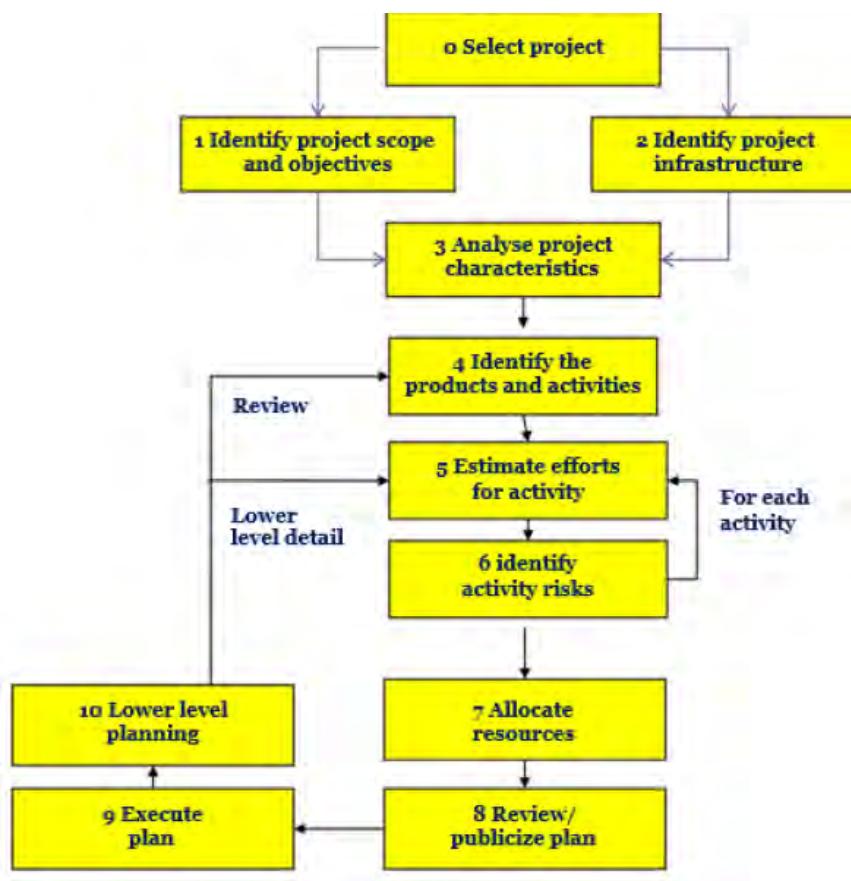
What? -This question is about the work that is to be done to deliver products/results, etc. and deliverables themselves.

Who? - This is about involved people, their roles and responsibilities as well as the way they are supposed to be organized.

When? - This is about project's schedule/timeline.

Step in Project Planning Process

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



Step 1: Identify project scope and objectives

- ❖ Identify objectives and practical measures of the effectiveness in meeting those objectives
- ❖ Establish a project authority
 - To ensure the unity of purpose among all persons concerned
- ❖ Identify all stakeholders in the project and their interests
- ❖ Modify objectives in the light of stakeholder analysis
- ❖ Establish methods of communication between all parties

Step 2: Identify Project Infrastructure

- ❖ Identify relationship between the project and strategic planning
 - To determine the order of related projects (in the organization) being carried out
 - To establish a framework within which the system fits
 - To ensure the hardware and software standards are followed
- ❖ Identify installation standards and procedures
 - more appropriate name: "Identify standards and procedures related to the software project"
- ❖ Identify project team organization

Step 3: Analyze Project Characteristics

- ❖ Distinguish the project as either objective-driven or product-driven
- ❖ Analyze other project characteristics (including quality-based ones)
- ❖ Identify high level project risks

- ❖ Take into account user requirements concerning implementation
- ❖ Select general lifecycle approach in the light of the above
- ❖ Review overall resource estimates

Up to this stage,

- the major risks of the project are identified
- the overall approach of the project is decided

So, it is a good place to re-estimate the required effort and other resources for the project

Step 4: Identify Project Products and Activities

- ❖ Identify and describe project products
 - **Identify all the products related to the project**
 - **Account for the required activities**
- ❖ Document generic product flows
 - To document the relative order of the products
- ❖ Recognize product instances
- ❖ Produce an ideal activity network
 - **Activity network shows the tasks that have to be carried out as well as their sequence of execution for the creation of a product from another**
- ❖ Modify the ideal to take into account need for stages and checkpoints
 - To check compatibility of products of previous activities

Step 5: Estimate Effort for Each Activity

- ❖ Carry out bottom-up estimates
 - need to estimate staff effort, time for each activity, and other resources
- ❖ Revise plan to create controllable activities
 - **need to break a task into a series of manageable sub-tasks**

Step 6: Identify Activity Risks

- ❖ Identify and quantify the risks of each activity
- ❖ Plan risk reduction and contingency measures where appropriate
- ❖ Adjust overall plans and estimates to take account of risks

Step 7: Allocate Resources (Staffing)

- ❖ Identify and allocate resources
 - **type of staff needed for each activity**
 - **staff availabilities are identified**
 - **staff are provisionally allocated to task**
- ❖ Revise plans and estimates to take into account resource constraints
 - **staffing constraints**
 - **staffing issues**

Step 8: Review/publicize Plan

- ❖ Review quality aspects of the project plan
 - **To ensure each activity is completed with a quality product**
 - **Each activity should have ‘exit requirements’.**
 - **This ensures the quality of the product on each activity.**
- ❖ Document plans and obtain agreement
 - all parties understand and agree to the commitments in the plan

Step 9 and 10: Execute Plan/Lower Level of Planning

- ❖ Build the physical project deliverables and present them to the customer for signoff.

Once the project is under way, 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 later stages will need to be delayed because more information will be available nearer the start of the stage.

SCHEDULING

Software Project Scheduling is an activity that distributes Estimated Effort across the Planned Project Duration by allocating the Effort to specific Software Engineering Task. Project Scheduling is the culmination of a Project Planning activity that is a primary component of Software Project Management. Project Scheduling establishes a “Road Map” for the Project Manager When combined with Estimation Methods and Risk Analysis.

Scheduling begins with Process Decompositions. The characteristics of the Project are used to adapt an appropriate task set for the Work to be done. A Task Network depicts each engineering task, its dependency on other tasks and its projected duration. The Task Network is used to compute the Critical Path, a Timeline Chart and a variety of project information. Using the Project Schedule as a guide, the Project Manager can track and control each step in the Software process.

Project scheduling

The reality of a Technical Project is that hundreds of small tasks must occur in order to accomplish a large goal. Some of these tasks lie outside the main stream and may be completed without worry about impact On Project Completion Date. Other tasks lie on the “Critical Path”. These tasks are called “Critical Tasks” fall behind schedule the completion date on the entire project is put into jeopardy. The Project Managers Objectives with regards to Project Scheduling are:

1. Define all Tasks.
2. Build a Network that depicts that's dependencies.
3. Identify the tasks that are Critical within the Network.
4. Track task progress to ensure that delay is recognized. “One Day at a time “.

Scheduling Techniques

- ❖ Program Evaluation and Review Technique (PERT)
- ❖ Gantt Chart
- ❖ Critical Path Method (CPM)
- ❖ Work Breakdown Structure (WBS)
- ❖ Bar Chart
- ❖ Linked Bar Chart
- ❖ Network Diagram

RESOURCE ALLOCATION

Resource allocation in project management is concerned with creating a plan which can help achieve future goals. There are many resources which have to be allocated when managing a project, *beginning from budget to equipment and tools, to data and the project's plan*.

Resource allocation in project management is so important because it gives a **clear picture on the amount of work that has to be done**. It also helps to schedule ahead and have an insight into the team's progress, including allocating the right amount of time to everyone on the team.

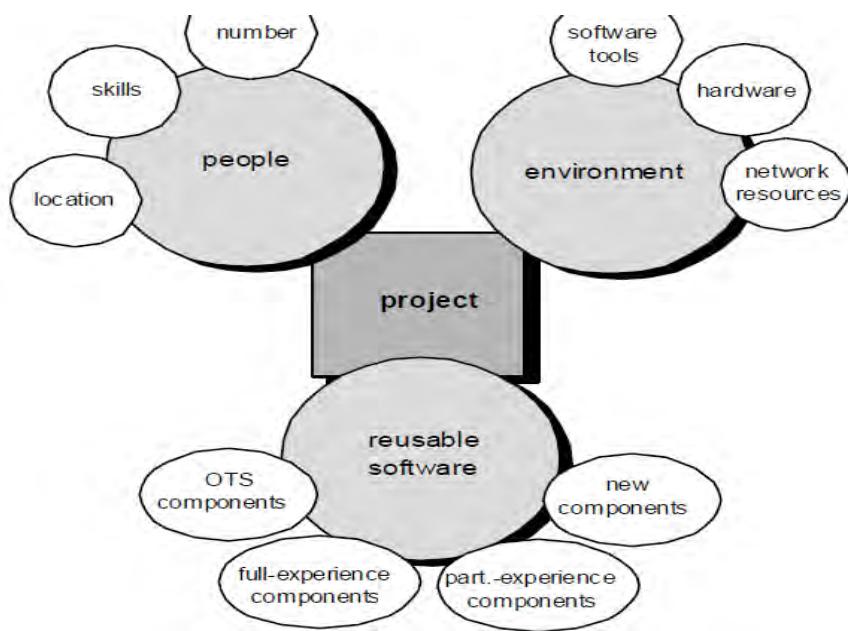
Resource allocation allows to plan and prepare for the project's implementation or achieving goals. It is also possible to analyze existing threats and risks to the project.

But above all, resource allocation in project management helps to **control all the workload**. This, as a result, contributes to team's effectiveness at work and what follows later is a **satisfying and exhaustive project**.

How to Allocate Resources

Since resources refer to quite an extensive list of things and aspects of project management, it may be confusing when trying to properly allocate them all. But it's not difficult and requires certain **dedication and involvement**. Here's what you can do to allocate resources in the right way when managing a project:

- ❖ **know the scope** – to know what is project about, what will be needed to achieve it, and to be able to properly allocate resources;
- ❖ **identify resources** – to know which tools, equipment, etc. will be needed for completing the project;
- ❖ **track time** – to have a deep analysis of the progress and current situation as well as be able to control it in the real-time;
- ❖ **Don't look only at the big picture** – the process of working on a project is not done with task allocation. Once resources are allocated keep track of all of them. If you lose at least one tiny detail, your project may fail;
- ❖ **Don't over-allocate** – because your team will experience burnout and their productivity will significantly drop.



Topics Covered:

- Cost benefit analysis
- Cash flow forecasting
- Cost benefit evaluation techniques
- Risk evaluation
- Selection of appropriate project report
 - Choosing technologies
 - Choice of Process Model
- Structured Method
- Rapid Application Development
- Waterfall model
- Spiral Model
- Prototyping Model

Cost benefit analysis: Cost-benefit analysis is a simple technique for comparing the business value a project will produce with the cost of producing it. Project managers use cost-benefit analysis in the project initiation phase to show the value of doing a project. During project initiation, the sponsor and project manager must justify the project to get the organization's approval to spend the money. The cost-benefit analysis compares the project's costs to the business value it will deliver. Few organizations want to go ahead with projects that will cost more than the value they will produce. So project managers conduct the cost-benefit analysis by gathering data on the value of the benefits and the cost of the project.

The Purpose of Cost Benefit Analysis

The purpose of cost benefit analysis in project management is to have a systemic approach to figure out the pluses and minuses of various paths through a project, including transactions, tasks, business requirements and investments. Cost benefit analysis gives you options, and it offers the best approach to achieve your goal while saving on investment.

There are two main purposes in using CBA:

1. To determine if the project is sound, justifiable and feasible by figuring out if its benefits outweigh costs.
2. To offer a baseline for comparing projects by determining which project's benefits are greater than its costs.

The **benefits** of a software project include four types:

- ❖ Cost-savings benefits
- ❖ Cost-avoidance benefits
- ❖ Improved-service-level benefits
- ❖ Improved Information benefits

The **cost** of a software project include five types:

- ❖ Hardware Costs
- ❖ Personnel Costs
- ❖ Facility Costs
- ❖ Operating Costs
- ❖ Supply Costs

Classification of Costs and Benefits: Costs and benefits can be classified as follows:

- (a) Tangible or Intangible Costs and Benefits
- (b) Fixed or Variable Costs and Benefits
- (c) Direct or Indirect Costs and Benefits

- ❖ **Tangible Cost:** An out lay of cash for any specific item or activity is referred to as a tangible cost. These costs are known and can be estimated quite accurately. *Hardware costs, salaries for professionals, software cost* are all tangible costs. Included tangible costs are the cost of equipment such as computers and terminals, the cost of resources, the cost of systems analysts' time, the cost of program's time, and other employees' salaries.
- ❖ **Intangible Cost:** Costs that are known to exist but their financial value cannot be exactly measured are referred to as intangible costs. The estimate is only an approximation. It is difficult to fix exact intangible costs. For example, *employee morale problem* because of installing new system is an intangible cost. The *cost of breakdown of an online system* during banking hours will cause the bank lose deposits, loss of customer goodwill etc.

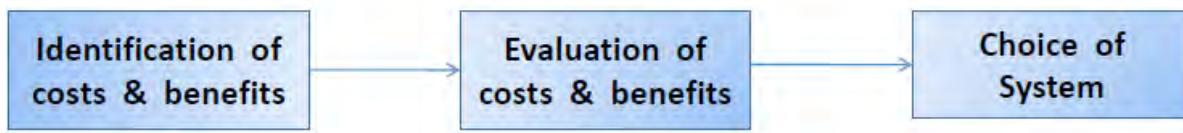
Tangible Benefits	Intangible Benefits
Increased Sales	Increased market share
Reduction in staff	Increased brand recognition
Reduction in inventory	Higher quality products
Reduction in IT cost	Improved customer service
Better supplier price	Better supplier relation

- ❖ **Direct Costs:** Direct costs are those which are directly associated with a system. Direct costs are having rupee value associated with it. They are applied directly to the operator. *For example, the purchase of pen drive for NRs.2, 000/- is a direct cost* because we can associate the pen drive with money spent.
- ❖ **Indirect Costs:** Indirect costs result from the operations that are not directly associated with the system. They are often referred to as overhead expenses. *For example, cost of space to install a system, insurance, maintenance of computer center, heat, light and air-conditioning are all indirect costs.*
- ❖ **Direct Benefits:** Direct benefits also can be specifically attributable to a given project. *For example, if the proposed system that can handle more transactions say 25% more than the present system then it is direct benefit.*
- ❖ **Indirect Benefits:** Indirect benefits are realized as a by-product of another system. *For example, a system that tracks sales-calls on customers provides an indirect marketing benefit* by giving additional information about competition. In this case, competition information becomes an indirect benefit although its work in terms of money cannot be exactly measured.
- ❖ **Fixed Costs:** Fixed costs are considered as sunk costs. They don't change. Once encountered, they will not recur. *For example, the purchase of equipment for a computer center is called as fixed cost as it remains constant whether in equipment is being used extensively or not.* Similarly, the insurance, and purchase of software, etc.
- ❖ **Variable Costs:** Variable costs are incurred on a regular basis. Recurring period may be weekly or monthly depending upon the system. They are generally proportional to work volume and continue as long as the system is in operation. *For example, the cost of computer forms vary in proportion to the amount of processing or the length of the reports desired.*
- ❖ **Fixed Benefits:** Fixed benefits also remain constant by using a new system. They also don't change. *If 20% of staff member are reduced, we can call it a fixed benefit.*
- ❖ **Variable Benefits:** Variable benefits, on the other hand, are realized on a regular basis. *For example the library information system that saves two minutes in providing information about particular*

book whether it is issued or not, to the borrower compared with the manual system. The amount of time saved varies with the information given to the number of borrowers.

The Process of Cost Benefit Analysis:

CBA is a process that presents a picture of various costs & benefits associated with an information system.
CBA as a process consists of 3 basic activities:



1. **Identification of Costs & Benefits:** Identify the costs & benefits associated with a task / operation, or investment project.
2. **Evaluation of costs** (tangible, intangible, fixed, variable, direct, & indirect) & benefits associated with each task / operation / investment project.
3. **Choice of system:** Select the best system which involves least cost & yields maximum benefits.

Steps in Cost Benefit Analysis:

Step-1: Identifying and estimating all of the costs and benefits of carrying out the project and expressing these costs and benefits in common units. It includes

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

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

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

Step-2: Expressing these costs and benefits in common units. Calculates net benefit.

Net benefit=total benefit – total cost. (Cost should be expressed in monetary terms)

Three types of cost and benefits

- **Development costs:** includes salary and other employment cost of staff involved.
- **Setup costs:** includes the cost of implementation of system such as hardware, and also file conversion, recruitment and staff training.
- **Operational cost:** cost require to operate system, after it is installed.
- **Direct benefits:** directly obtained benefit by making use of/operating the system.
 - Example: reduction of salary bills, through the introduction of a new, computerized system.
- **Assessable indirect benefits:** these benefits are obtained due to updating / upgrading the performance of current system. It is also referred as “secondary benefits”.
 - Example: “use of user – friendly screen”, which promotes reduction in errors, thus increases the benefit.
- **Intangible benefits:** these benefits are longer term, difficult to quantify. It is also referred as “indirect benefits”.
 - Example: enhanced job interest leads reduction of staff turnover, in turn leads lower recruitment costs.

Cost benefit evaluation techniques

Cost-benefit analysis (CBA) is a technique used to compare the total costs of a program/project with its benefits, using a common metric (most commonly monetary units). This enables the calculation of the net cost or benefit associated with the program.

As a technique, it is used most often at the start of a program or project when different options or courses of action are being appraised and compared, as an option for choosing the best approach. It can also be used, however, to evaluate the overall impact of a program in quantifiable and monetized terms.

CBA adds up the total costs of a program or activity and compares it against its total benefits. The technique assumes that a monetary value can be placed on all the costs and benefits of a program, including tangible and intangible returns to other people and organizations in addition to those immediately impacted. As such, a major advantage of cost-benefit analysis lies in forcing people to explicitly and systematically consider the various factors which should influence strategic choice.

Decisions are made through CBA by comparing the net present value (NPV) of the program or project's costs with the net present value of its benefits. Decisions are based on whether there is a net benefit or cost to the approach, i.e. total benefits less total costs. Costs and benefits that occur in the future have less weight attached to them in a cost-benefit analysis. To account for this, it is necessary to 'discount' or reduce the value of future costs or benefits to place them on a par with costs and benefits incurred today. The 'discount rate' will vary depending on the sector or industry, but public sector activity generally uses a discount rate of 5-6%. The sum of the discounted benefits of an option minus the sum of the discounted costs, all discounted to the same base date, is the 'net present value' of the option.

We would consider proceeding with a project only where the benefits outweigh the costs. However, in order to choose among projects, we need to take into account the timing of the costs and benefits as well as the benefits relative to the size of the investment.

Net profit: The net profit of a project is the difference between the total costs and the total income over the life of the project. It shows what the company has earned (or lost) in a given period of time (usually one year). It is also called net income or net earnings.

Net profit=total costs-total incomes

Calculate net profit for the following

Year	Project 1	Project 2	Project 3
0	-100000	-1000000	-120000
1	10000	200000	30000
2	10000	200000	30000
3	10000	200000	30000
4	20000	200000	30000
5	100000	300000	75000

Solution

Year	Project 1	Project 2	Project 3
0	-100000	-1000000	-120000
1	10000	200000	30000
2	10000	200000	30000
3	10000	200000	30000
4	20000	200000	30000
5	100000	300000	75000
Net Profit	50000	100000	75000

Payback period: 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'.

Advantages

- simple and easy to calculate.
- It is also a seriously flawed method of evaluating investments

Disadvantages

- It attaches no value to *cash flows* after the end of the payback period.
- It makes no adjustments for risk.
- It is not directly related to wealth maximization as NPV is.
- It ignores the time value of money.
- The "cut off" period is arbitrary.

Calculate Payback period if original investment is Rs. 12000.

Year	Return Benefits	
	Project X	Project Y
1	3,000	3,000
2	3,000	3,500
3	3,000	4,000
4	3,000	4,500
5	3,000	5,000

Solution:

Year	Return Benefits		Cumulative benefits	
	Project X	Project Y	Project X	Project Y
1	3,000	3,000	3,000	3,000
2	3,000	3,500	6,000	6,500
3	3,000	4,000	9,000	10,500
4	3,000	4,500	12,000 ←	15,000
5	3,000	5,000	15,000	20,000

Therefore, the cumulative benefit of project X at year 4 is equal to the cost (i.e. NRs. 12,000) and that of project Y lies between year 3 and year 4. So, the payback period is 4 years for the project X and for project Y is calculated below:

$$\begin{aligned}
 \text{Payback Period} &= (\text{Amount not covered in lower year} / \text{Difference in cumulative amounts of HY \& LY}) \\
 &+ \text{LY} \\
 &= (1500/4500) + 3 \\
 &= 3.33 \text{yr.}
 \end{aligned}$$

Return on investment: The return on investment (ROI), also known as the accounting rate of return (ARR), provides a way of comparing the net profitability to the investment required. There are some variations on the formula used to calculate the return on investment but a straightforward common version is

$$\text{ROI} = \frac{\text{Average Annual Profit}}{\text{Total Investment}} * 100$$

Total Investment

$$\text{Average Annual Profit} = \frac{\text{Net Profit}}{\text{Total no. of Years}}$$

Disadvantages of ROI

- Takes no account of the timing of the cash flows
- It is tempting to compare the rate of return with current interest rates. (No compound interest calculated)

Calculate ROI for following.

Year	Project 1
0	-100000
1	10000
2	10000
3	10000
4	20000
5	100000
Net Profit	50000

Answer: Total Investment = 1,00,000

Net Profit = 50,000

Total no. of Year = 5

Average Annual Profit = 50,000/5
= NRs. 10,000

$$\text{ROI} = \frac{(10,000/1,00,000) * 100}{10\%}$$

Note: The higher the ROI, the better.

Discounted Cash Flow (DCF) is a cash flow summary adjusted to reflect the time value of money. DCF can be an important factor when evaluating or comparing investments, proposed actions, or purchases. Other things being equal, the action or investment with the larger DCF is the better decision. When discounted cash flow events in a cash flow stream are added together, the result is called the **Net Present Value (NPV)**.

When the analysis concerns a series of cash inflows or outflows coming at different future times, the series is called a **cash flow stream**. Each future cash flow has its own value today (its own present value). The sum of these present values is the **Net Present Value** for the cash flow stream. The size of the discounting effect depends on two things: the amount of time between now and each future payment (the number of discounting periods) and an interest rate called the **Discount Rate**.

- As the number of discounting periods between now and the cash arrival increases, the present value decreases.
- As the discount rate (interest rate) in the present value calculations increases, the present value decreases.

$$\text{Discount factor} = \frac{1}{(1 + r\%)}^t$$

Net present value (NPV): It is the sum of the present values of all future amounts. *Present value* is the value which a future amount is worth at present. It takes into account the profitability of a project and the timing of the cash flows.

Discount rate: Is the annual rate by which we discount future earnings. e.g. If discount rate is 10% and the return of an investment in a year is \$110, the present value of the investment is \$100.

Let n be the number of year and r be the discount rate, the present value (PV) is given by

$$PV = \frac{\text{value in year } n}{(1+r)^n}$$

Disadvantage: May not be directly comparable with earnings from other investments or the costs of borrowing capital.

Issues in NPV

- Choosing an appropriate discount rate is difficult
- Ensuring that the rankings of projects are not sensitive to small changes in discount rate

Guidelines:

- Use the standard rate prescribed by the organization
- Use interest rate + premium rate
- Use a target rate of return
- Rank the projects using various discount rates

Consider two competing investments in computer equipment. Each calls for an initial cash out lay of \$100, and each returns a total a \$200 over the next 5 years making net gain of \$100. But the timing of the returns is different, as shown in the table below (Case A and Case B), and therefore the present value of each year's return is different. The sum of each investments present values is called the Discounted Cash flow (DCF) or Net Present Value (NPV). Using a 10% discount rate.

Timing	Discount rate (10%)	Case A		Case B	
		Net cash flow	Present value	Net cash flow	Present value
now 0	1	100	100	100	100
year 1	0.9091	60	54.55	20	18.18
year 2	0.8264	60	49.59	20	16.53
year 3	0.7513	40	30.05	40	30.05
year 4	0.6830	20	13.66	60	40.98
year 5	0.6209	20	12.42	60	37.26
Total		100	62.67	100	33.83

Using NPV function of excel

Assuming a 10% discount rate, the NPV for project 1 would be calculated as in the following table. Using a 10% discount rate, calculate the net present values for projects 2, 3 and 4 and decide which, on the basis of this, is the most beneficial to pursue.

Table Applying the discount factors to project I

Year	Project I cash flow (£)	Discount factor @ 10%	Discounted cash flow (£)
0	-100,000	1.0000	-100,000
1	10,000	0.9091	9,091
2	$\frac{1}{(1 + r\%)^t}$	0.8264	8,264
3	10,000	0.7513	7,513
4	20,000	0.6830	13,660
5	100,000	0.6209	62,090
Net Profit:	£50,000		NPV: £618

Calculate the NPV for each of the projects A, B and C shown in the table below using each of the discount rates 8%, 10% and 12%. For each of the discount rates, decide which is the best project. What can you conclude from this result?

Year	Cash flow values (£)		
	Project A	Project B	Project C
0	-8,000	-8,000	-10,000
1	4,000	1,000	2,000
2	4,000	2,000	2,000
3	2,000	4,000	6,000
4	1,000	3,000	2,000
5	500	9,000	2,000
6	500	-6,000	2,000
Net Profit	£ 4,000	£ 5,000	£ 6,000

Table F.3 The effect on net present value of varying the discount rate

Year	Cash flow values (£)		
	Project A	Project B	Project C
0	-8,000	-8,000	-10,000
1	4,000	1,000	2,000
2	4,000	2,000	2,000
3	2,000	4,000	6,000
4	1,000	3,000	2,000
5	500	9,000	2,000
6	500	-6,000	2,000
Net Profit	£ 4,000	£ 5,000	£ 6,000
NPV @ 8%	£ 2,111	£ 2,365	£ 2,421
NPV @ 10%	£ 1,720	£ 1,818	£ 1,716
NPV @ 12%	£ 1,356	£ 1,308	£ 1,070

NPV –Decision Rule (If the NPV is positive, accept the project)

A positive NPV means that the project is expected to add value to the firm and will therefore increase the wealth of the owners. Since our goal is to increase owner wealth, NPV is a direct measure of how well this project will meet our goal.

If $NPV > 0$, **accept** the Project.

If $NPV = 0$, **accept or reject** the Project.

If $NPV < 0$, **reject** the Project.

Internal rate of return: One disadvantage of NPV as a measure of profitability is that, although it may be used to compare projects, it might not be directly comparable with earnings from other investments or the costs of borrowing capital. Such costs are usually quoted. It is based entirely on the estimated cash flows and is independent of interest rates found elsewhere. IRR is calculated by following formula here CF stands for Cash Flow and r is the internal rate of return

$$\left[\frac{CF_1}{(1+r)^1} + \frac{CF_2}{(1+r)^2} + \frac{CF_3}{(1+r)^3} + \dots \right] - \text{Initial Investment} = 0$$

The IRR compares returns to costs by asking: "What is the discount rate that would give the cash flow stream a net present value of 0?"

Timing	Discount Rate(10%)	CASE A		CASE B	
		Net Cash Flow	Present Value	Net Cash Flow	Present Value
Now 0	1	-\$100.00	-\$100.00	-\$100.00	-\$100.00
Year 1	0.9091	\$60.00	\$54.54	\$20.00	\$18.18
Year 2	0.8264	\$60.00	\$49.59	\$20.00	\$16.52
Year 3	0.7513	\$40.00	\$30.05	\$40.00	\$30.05
Year 4	0.6830	\$20.00	\$13.70	\$60.00	\$41.10
Year 5	0.6209	\$20.00	\$12.42	\$60.00	\$37.27
Total		Net CF _A = \$100.00	NPV _A = \$60.30	Net CF _B = \$100.00	NPV _B = \$43.12

Decision Rule: Accept the project if the IRR is greater than the required return

Internal Rate of Return Rules: In IRR decisions, if we have only one project, most of the time we need the basic rule «independent project»:

- IRR > Cost of capital (should be accepted)
- IRR = Cost of capital (provides the minimum return)
- IRR < Cost of capital (shouldn't be accepted)

In addition, we need to take other situations into account too. Especially, even though NPV and IRR will generally give us the same decision, there are some exceptions:

- Nonconventional cash flows—cash flow signs change more than once
- Mutually exclusive projects

Cash flow forecasting: A cash flow forecast is an estimate of the amount of money you expect to flow in and out of your business. It includes all your projected income and expenses and usually covers the next

year, though it can also cover a shorter period such as a week or month. It estimate overall cost and benefits of a product with respect to time. Cash flow is negative during development stage where as it is positive during operating life. How often management should forecast cash flow is dependent on the financial security of the business. If the business is struggling, or is keeping a watchful eye on its finances, the business owner should be forecasting and revising his or her cash flow on a daily basis. However, if the finances of the business are more stable and 'safe', then forecasting and revising cash flow weekly or monthly is enough. Here are the key reasons why a cash flow forecast is so important:

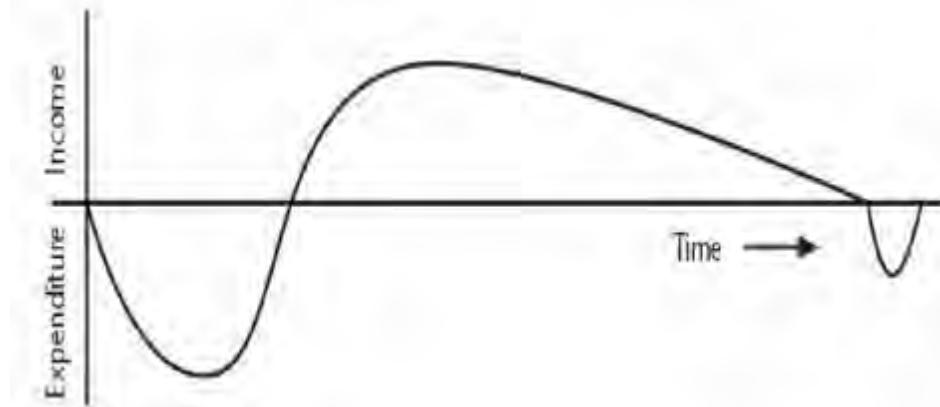
- Identify potential shortfalls in cash balances in advance—think of the cash flow forecast as an "early warning system". This is, by far, the most important reason for a cash flow forecast.
- Make sure that the business can afford to pay suppliers and employees. Suppliers who don't get paid will soon stop supplying the business; it is even worse if employees are not paid on time.
- Spot problems with customer payments—preparing the forecast encourages the business to look at how quickly customers are paying their debts. Note—this is not really a problem for businesses (like retailers) that take most of their sales in cash/credit cards at the point of sale.
- As an important discipline of financial planning—the cash flow forecast is an important management process, similar to preparing business budgets.
- External stakeholders such as banks may require a regular forecast. Certainly, if the business has a bank loan, the bank will want to look at the cash flow forecast at regular intervals.

The cash flow **during development stage** includes

- Staff wages
- Borrowing money from bank
- Paying interest to bank
- Payment of salaries
- Amount spent for installation, buying hw and sw

Cash flow expected after completion of project is also termed as **income** and income is expected by 2 ways.

- Payment on completion
- Stage payment



Risk evaluation

Software development is activity that uses a variety of technological advancements and requires high levels of knowledge. Because of these and other factors, every software development project contains elements of uncertainty. This is known as project risk. The success of a software development project depends quite heavily on the amount of risk that corresponds to each project activity. As a project manager, it's not enough

to merely be aware of the risks. To achieve a successful outcome, project leadership must identify, assess, prioritize, and manage all of the major risks.

Risk evaluation is meant to decide whether to proceed with the project or not, and whether the project is meeting its objectives. Risk occurs, When the project exceed its original specification and deviations from achieving it objectives and so on.

- **Risk identification and ranking:** In any project evaluation we should attempt to identify the risks and quantify their potential effects. One common approach to risk analysis is to construct a project risk matrix utilizing a checklist of possible risks and to classify each risk according to its relative importance and likelihood. Note that the importance and likelihood need to be separately assessed - we might be less concerned with something that, although serious, is very unlikely to occur than with something less serious that is almost certain.
- **Risk and net present value:** Where a project is relatively risky it is common practice to use a higher discount rate to calculate net present value. This addition or risk premium, might, for example, be an additional 2% for a reasonably safe project or 5% for a fairly risky one. Projects may be categorized as high, medium or low risk using a scoring method and risk premiums designated for each category. The premiums, even if arbitrary, provide a consistent method of taking risk into account.
- **Cost-benefit analysis:** A rather more sophisticated approach to the evaluation of risk is to consider each possible outcome and estimate the probability of its occurring and the corresponding value of the outcome. Rather than a single cash flow forecast for a project, we will then have a set of cash flow forecasts, each with an associated probability of occurring. The value of the project is then obtained by summing the cost or benefit for each possible outcome weighted by its corresponding probability.
- **Risk profile analysis:** An approach which attempts to overcome some of the objections to cost-benefit averaging is the construction of risk profiles using sensitivity analysis.

	Probability P	Expected Value (£) i xp	
High	800,000	0.8	80,000
Medium	650,000	0.6	390,000
Low	100,000	0.3	30,000
Expected Income			500,000

This involves varying each of the parameters that affect the project's cost or benefits to ascertain how sensitive the project's profitability is to each factor. We might, for example, vary one of our original estimates by plus or minus 5% and recalculate the expected costs and benefits for the project. By repeating this exercise for each of our estimates in turn we can evaluate the sensitivity of the project to each factor.

By studying the results of a sensitivity analysis we can identify those factors that are most important to the success of the project. We then need to decide whether we can exercise greater control over them or otherwise mitigate their effects. If neither is the case, then we must live with the risk or abandon the project. For an explanation of the Sensitivity analysis demands that we vary each factor one at a time. It does not Monte Carlo technique easily allow us to consider the effects of

combinations of circumstances, neither see any textbook on does it evaluate the chances of a particular outcome occurring.

- **Using decision trees:** The approaches to risk analysis discussed previously rather assume that we are passive bystanders allowing nature to take its own course - the best we can do is to reject over-risky projects or choose those with the best risk profile. There are many situations, however, where we can evaluate whether a risk is important and, if it is, indicate a suitable course of action. Many such decisions will limit or affect future options and, at any point, it is important to be able to see into the future to assess how a decision will affect the future profitability of the project.

Selection of appropriate project Approach: To develop a project an appropriate technology as well as model needs to be selected. The appropriate technology and process model leads to a successful project management, resource management and project is completed with in deadline.

Choosing technologies:

- Is a data oriented or process oriented system to be implemented?
- Will the software that is to be produced be a general package or application specific?
- **Is the system to be implemented of a particular type for which specific tools have been developed?**
- Is the system to be created safety critical?
- **What is the nature of the hardware/software environment in which the system will operate?**

Uncertainty: The greater the uncertainties at the beginning of the project, the greater the risk that the project will be unsuccessful.

The **product** becomes uncertain if the requirements are not understood properly. And the users themselves could be uncertain about what a proposed Information System is to do.

The **process** becomes uncertain if the project under consideration is the first where an organization has tried to use a method that is new to them. As well as any changes in the way that the systems are developed is going to introduce uncertainty.

The **resource** become uncertain mainly because of unavailability of staff of the right ability and experience. As well as the larger the number of resources need or the longer the duration of the project, the more inherently risky it is likely to be.

Choice of Process Model: A software process (also knowns as software methodology) is a set of related activities that leads to the production of the software. These activities may involve the development of the software from the scratch, or, modifying an existing system. Any software process must include the following four activities:

1. **Software specification** (or requirements engineering): Define the main functionalities of the software and the constrains around them.
2. **Software design and implementation:** The software is to be designed and programmed.
3. **Software verification and validation:** The software must conform to its specification and meets the customer needs.
4. **Software evolution** (software maintenance): The software is being modified to meet customer and market requirements changes.

In practice, they include sub-activities such as requirements validation, architectural design, unit testing, ...etc.

There are also **supporting activities** such as configuration and change management, quality assurance, project management, user experience. Along with **other activities** aim to **improve** the above activities by introducing new techniques, tools, following the best practice, process standardization (so the diversity of software processes is reduced), etc.

When we talk about a process, we usually talk about the activities in it. However, a process also includes the process description, which includes:

1. **Products:** The outcomes of an activity. For example, the outcome of architectural design maybe a model for the software architecture.
2. **Roles:** The responsibilities of the people involved in the process. For example, the project manager, programmer, etc.
3. **Pre and post conditions:** The conditions that must be true before and after an activity. For example, the pre-condition of the architectural design is the requirements have been approved by the customer, while the post condition is the diagrams describing the architectural have been reviewed.

Software process is complex; it relies on making decisions. There's no ideal process and most organizations have developed their own software process.

Rapid Application Development: Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product. In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process.

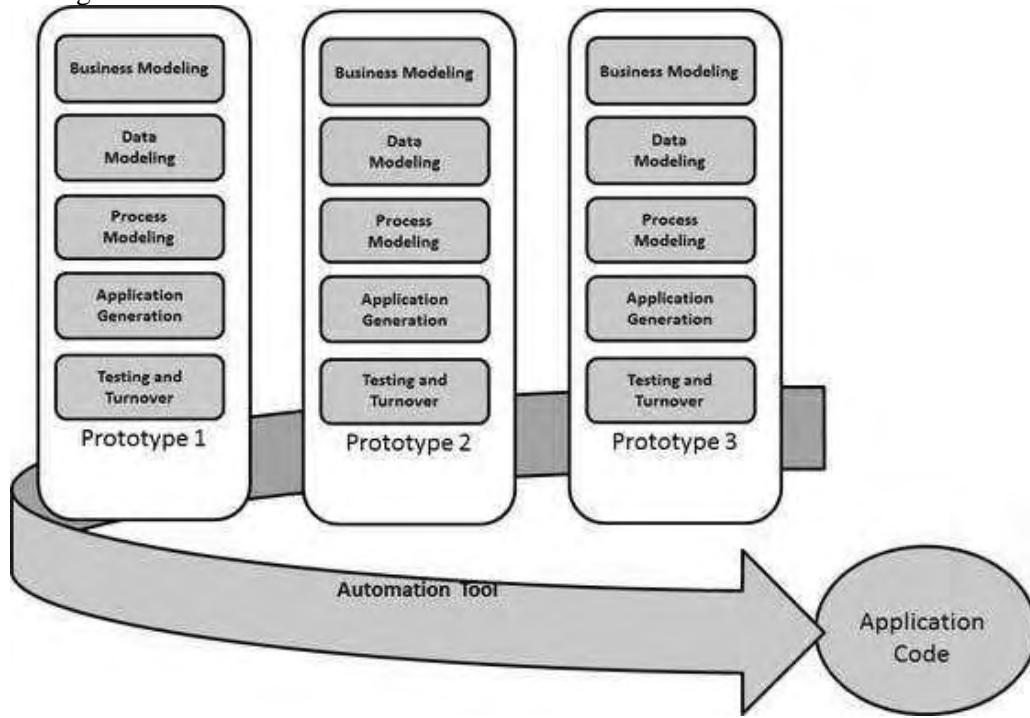
RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

RAD Model Design: RAD model distributes the analysis, design, build and test phases into a series of short, iterative development cycles. Following are the various phases of the RAD Model –

- **Business Modeling:** The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels. A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.
- **Data Modeling:** The information gathered in the Business Modeling phase is reviewed and analyzed to form sets of data objects vital for the business. The attributes of all data sets is identified and defined. The relation between these data objects are established and defined in detail in relevance to the business model.
- **Process Modeling:** The data object sets defined in the Data Modeling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model. The process model for any changes or enhancements to the data object sets is defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.

- **Application Generation:** The actual system is built and coding is done by using automation tools to convert process and data models into actual prototypes.
- **Testing and Turnover:** The overall testing time is reduced in the RAD model as the prototypes are independently tested during every iteration. However, the data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage. Since most of the programming components have already been tested, it reduces the risk of any major issues.

The following illustration describes the RAD Model in detail.

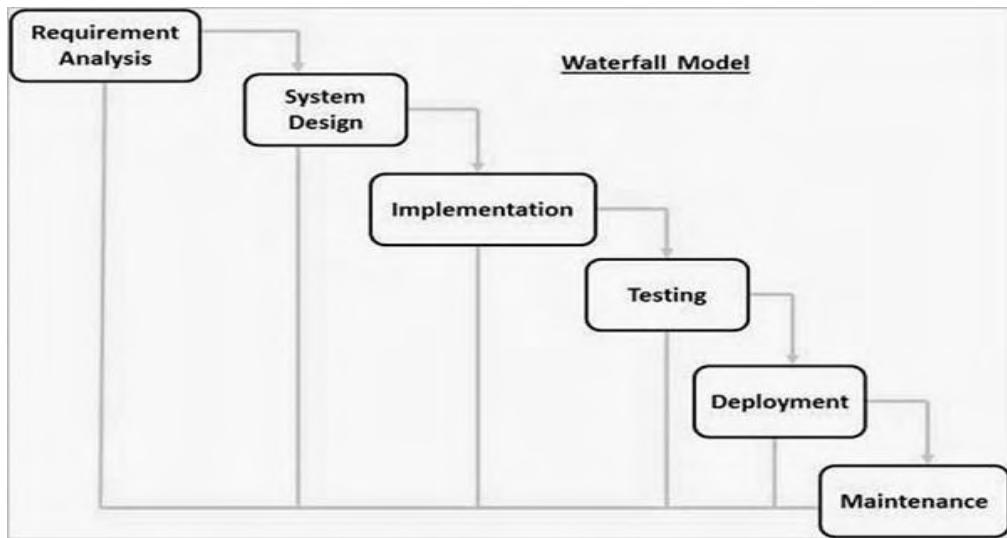


Waterfall model: The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model – Design: Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** – With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** – All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** – Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** – There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

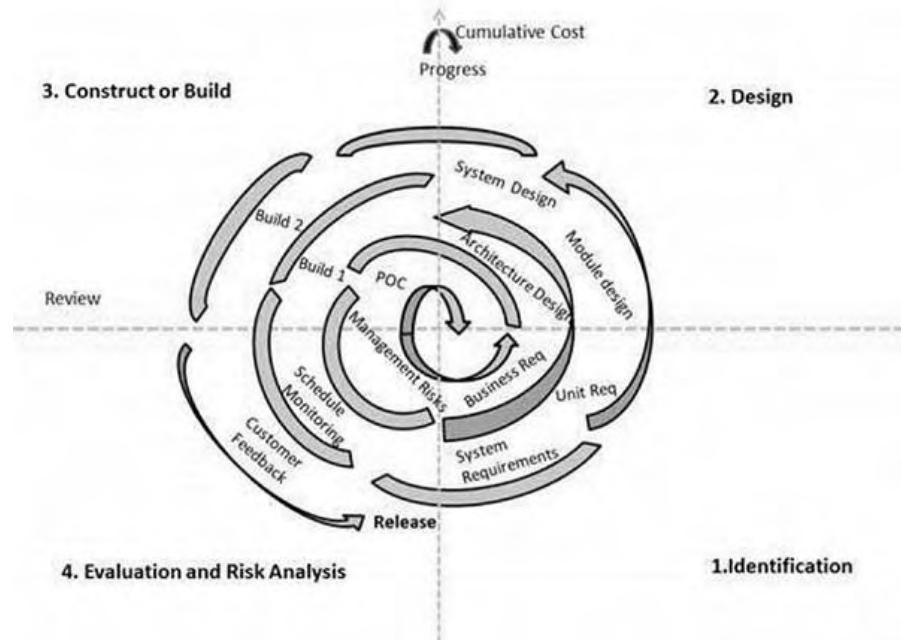
Spiral Model: The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk

analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

Spiral Model – Design: The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

- **Identification:** This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase. This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.
- **Design:** The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.
- **Construct or Build:** The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback. Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.
- **Evaluation and Risk Analysis:** Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

The following illustration is a representation of the Spiral Model, listing the activities in each phase.



Based on the customer evaluation, the software development process enters the next iteration and subsequently follows the linear approach to implement the feedback suggested by the customer. The process of iterations along the spiral continues throughout the life of the software.

Prototyping Model: Prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation. Prototyping is used to allow the users evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

Following is a stepwise approach explained to design a software prototype.

- **Basic Requirement Identification:** This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.
- **Developing the initial Prototype:** The initial Prototype is developed in this stage, where the very basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed. While, the workarounds are used to give the same look and feel to the customer in the prototype developed.
- **Review of the Prototype:** The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an organized manner and used for further enhancements in the product under development.

- **Revise and Enhance the Prototype:** The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like – time and budget constraints and technical feasibility of the actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until the customer expectations are met.

Prototypes can have horizontal or vertical dimensions. A Horizontal prototype displays the user interface for the product and gives a broader view of the entire system, without concentrating on internal functions. A Vertical prototype on the other side is a detailed elaboration of a specific function or a sub system in the product.

The purpose of both horizontal and vertical prototype is different. Horizontal prototypes are used to get more information on the user interface level and the business requirements. It can even be presented in the sales demos to get business in the market. Vertical prototypes are technical in nature and are used to get details of the exact functioning of the sub systems. For example, database requirements, interaction and data processing loads in a given sub system.

Software Prototyping – Types: There are different types of software prototypes used in the industry. Following are the major software prototyping types used widely –

- **Throwaway/Rapid Prototyping:** Throwaway prototyping is also called as rapid or close ended prototyping. This type of prototyping uses very little efforts with minimum requirement analysis to build a prototype. Once the actual requirements are understood, the prototype is discarded and the actual system is developed with a much clear understanding of user requirements.
- **Evolutionary Prototyping:** Evolutionary prototyping also called as breadboard prototyping is based on building actual functional prototypes with minimal functionality in the beginning. The prototype developed forms the heart of the future prototypes on top of which the entire system is built. By using evolutionary prototyping, the well-understood requirements are included in the prototype and the requirements are added as and when they are understood.
- **Incremental Prototyping:** Incremental prototyping refers to building multiple functional prototypes of the various sub-systems and then integrating all the available prototypes to form a complete system.
- **Extreme Prototyping:** Extreme prototyping is used in the web development domain. It consists of three sequential phases. First, a basic prototype with all the existing pages is presented in the HTML format. Then the data processing is simulated using a prototype services layer. Finally, the services are implemented and integrated to the final prototype. This process is called Extreme Prototyping used to draw attention to the second phase of the process, where a fully functional UI is developed with very little regard to the actual services.

Topics Covered:

- Objectives of activity planning
- Project schedule
- Projects and activities
- Sequencing and scheduling activities
- Network planning model
- Representation of Lagged activities
- Adding the time dimension
- Backward and Forward Pass
- Identifying Critical Path
- Activity Throat
- Shortening Project
- Precedence Network

The next step in project planning is working on the schedule of the project. Several automated tools are available so project managers can schedule the work with efficiency. A Work Breakdown Structure is needed to determine the activity plan. A thoroughly outlined Work Breakdown Structure helps minimize task redundancy and makes certain the work is within the scope of the project.

Activity Planning breaks down into two processes:

- Defining Activities
- Sequencing Activities

Defining Activities: The first task is defining the term activity. In reference to project management, the term activity means a specific event or occurrence. Team members are required to participate in an activity. This can also mean the project as a whole. Activity planning begins with outlining the structure of work breakdown. The goal of activity planning is pinpointing activities required to achieve deliverables. It addresses this question: “What activities are to be performed to meet the requirements of the work package?” The data gathered to answer this question is the lineup of activities arranged in a sequence. This sequence is known as activity planning in project management.

Every project cycles through a process known as progressive elaboration. As work progresses, corresponding knowledge is acquired. Additional knowledge indicates an adjustment in planning which means details will need to be reviewed or reworked. Progressive elaboration is the unfolding of project progression and details worked through. Project plans should be revisited and reviewed as the project moves forward. This assures the project will meet expectations and plans remain flexible if details need to be changed. Ongoing planning as the project moves forward is termed rolling wave planning.

Sequencing Activities: Sequence activities is the process of identifying and documenting relationships among the project activities. In the project management, the key benefit of this type of process is that it defines the logical sequence of work to obtain the greatest efficiency given all project constraints.

In the project management process groups and knowledge area mapping the sequence activities fall under the planning process group and project time management knowledge areas. The project schedule development uses the outputs from the processes to define activities, sequence activities, estimate activity resources, and estimate activity durations in combination with the scheduling tool to produce the schedule model.

In the sequence activities data flow diagram, every activity and milestone except the first and last should be connected to at least one predecessor with a finish-to-start logical relationship and at least one successor with the finish-to-start or finish-to-finish logical relationship. Sequencing can be performed by using project management software or by using manual or automated techniques.

Moreover, the schedule management plan identifies the scheduling method and tool to be used for the project, which will guide how the activities may be sequenced.

Objectives of activity planning

In addition to providing project and resource schedules, activity planning aims to achieve a number of other objectives which may be summarized as follows.

- **Feasibility assessment** Is the project possible within required timescales and resource constraints? It is not until we have constructed a detailed plan that we can forecast a completion date with any reasonable knowledge of its achievability. The fact that a project may have been estimated as requiring two work-years effort might not mean that it would be feasible to complete it within, say, three months were eight people to work on it - that will depend upon the availability of staff and the degree to which activities may be undertaken in parallel.
- **Resource allocation** What are the most effective ways of allocating resources to the project and when should they be available? The project plan allows us to investigate the relationship between timescales and resource availability (in general, allocating additional resources to a project shortens its duration) and the efficacy of additional spending on resource procurement.
- **Detailed costing** How much will the project cost and when is that expenditure likely to take place? After producing an activity plan and allocating specific resources, we can obtain more detailed estimates of costs and their timing.
- **Motivation** Providing targets and being seen to monitor achievement against targets is an effective way of motivating staff, particularly where they have been involved in setting those targets in the first place.
- **Co-ordination** When do the staff in different departments need to be available to work on a particular project and when do staff need to be transferred between projects? The project plan, particularly with large projects involving more than a single project team, provides an effective vehicle for communication and co-ordination among teams. In situations where staff may need to be transferred between project teams (or work concurrently on more than one project), a set of integrated project schedules should ensure that such staff are available when required and do not suffer periods of enforced idleness.

Activity planning and scheduling techniques place an emphasis on completing the project in a minimum time at an acceptable cost or, alternatively, meeting an arbitrarily set target date at minimum cost. These are not, in themselves, concerned with meeting quality targets, which generally impose constraints on the scheduling process.

One effective way of shortening project durations is to carry out activities in parallel. Clearly we cannot undertake all the activities at the same time - some require the completion of others before they can start and there are likely to be resource constraints limiting how much may be done simultaneously. Activity scheduling will, however, give us an indication of the cost of these constraints in terms of lengthening timescales and provide us with an indication of how timescales may be shortened by relaxing those constraints. It is up to us, if we try relaxing precedence constraints by, for example, allowing a program

coding task to commence before the design has been completed, to ensure that we are clear about the potential effects on product quality.

When to Plan?

Planning is the ongoing process of refinement, each iteration becoming more detailed and more accurate than the last. Over successive iterations, the emphasis and purpose of planning will shift. The main purpose of planning is to estimate the time scales and the risks of non-achieving target completion dates or keeping within the budget.

Project schedule

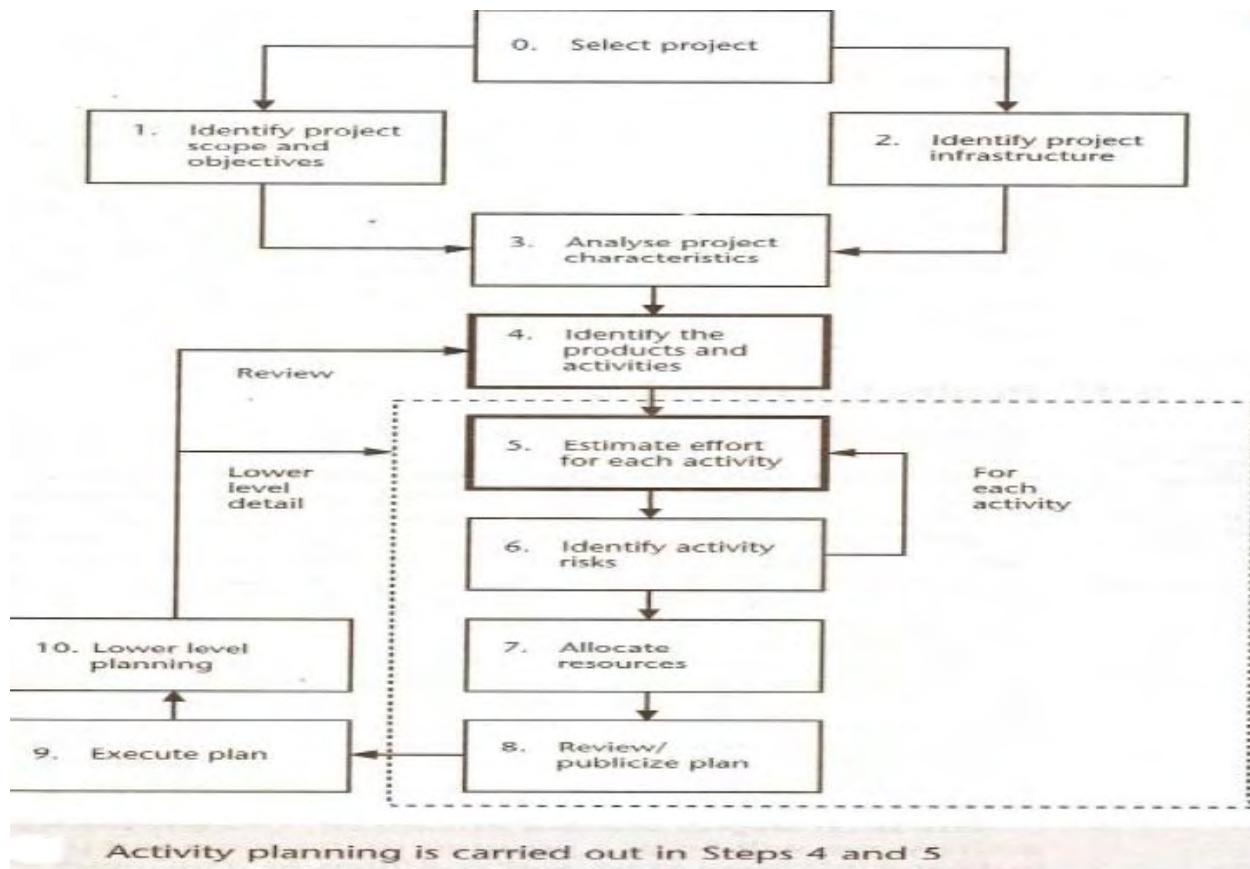
Project scheduling is a mechanism to communicate what tasks need to get done and which organizational resources will be allocated to complete those tasks in what timeframe. A project schedule is a document collecting all the work needed to deliver the project on time.

A project is made up of many tasks, and each task is given a start and end (or due date), so it can be completed on time. Likewise, people have different schedules, and their availability and vacation or leave dates need to be documented in order to successfully plan those tasks. Project scheduling occurs during the planning phase of the project. Following three questions need to be answered before start:

1. What needs to be done?
2. When will it be done?
3. Who will do it?

Creating a project schedule comprises four main stages.

- **First step** in producing the plan is to decide what activities need to be carried out and in what order they are to be done. From this we can construct an ideal activity plan i.e., a plan of when each activity would ideally be undertaken were resources not a constraint.
- **Second step:** Ideal activity plan will then be the subject of an activity risk analysis, aimed at identifying potential problems. This might suggest alterations to the ideal activity plan and will almost certainly have implications for resource allocation.
- **Third step** is resource allocation. The expected availability of resources might place constraints on when certain activities can be carried out.
- **Final Step** is schedule production. Once resource have allocated to each activity, one can draw and publish the project schedule. Planned start and completion dates and a resource requirement statement for each activity. Activity planning is normally carried out in step 4 and 5.



Projects and Activities

Defining activities: Some assumptions (criteria) need to make that will be relevant when we start to produce an activity plan. Activities must be defined so that they meet these assumptions. Any activity that does not meet these criteria must be redefined.

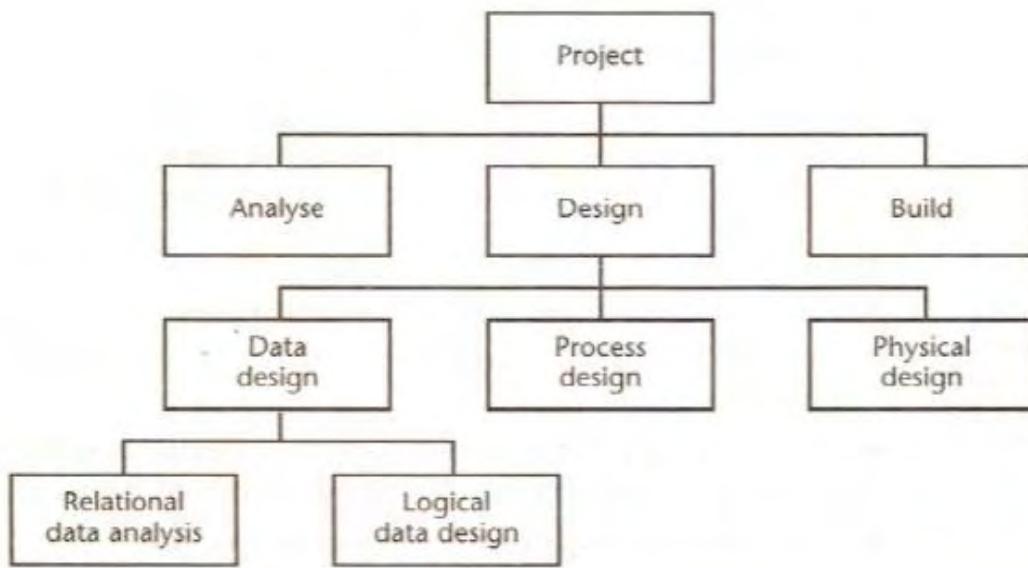
A project is composed of a number of interrelated activities. A project may start when at least one of its activities is ready to start. A project will be completed when all of the activities it encompasses have been completed. An activity must have a clearly defined start and a clearly defined end-point, normally marked by the production of a tangible deliverable. An activity requires a resource (as most do) then that resource requirement must be forecastable and is assumed to be required at a constant level throughout the duration of the activity. The duration of an activity must be forecastable — assuming normal circumstances, and the reasonable availability of resources. Some activities might require that others are completed before they can begin (these are known as precedence requirements).

Identifying activities: Essentially there are three approaches to identifying the activities or tasks that make up a project

- The activity-based approach,
 - The product-based approach
 - The hybrid approach.

The activity-based approach: The activity-based approach consists of creating a list of all the activities that the project is thought to involve. This might involve a brainstorming session involving the whole project team or it might stem from an analysis of similar past projects. When listing activities, particularly for a large project, it might be helpful to subdivide the project into the main life-style stages and consider each of these separately. Generating a task list is to create 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.

Activities are added to a branch in the structure if they directly contribute to the task immediately above — if they do not contribute to the parent task, then they should not be added to that branch. The tasks at each level in any branch should include everything that is required to complete the task at the higher level — if they are not a comprehensive definition of the parent task, then something is missing. When preparing a WBS, consideration must be given to the final level of detail.



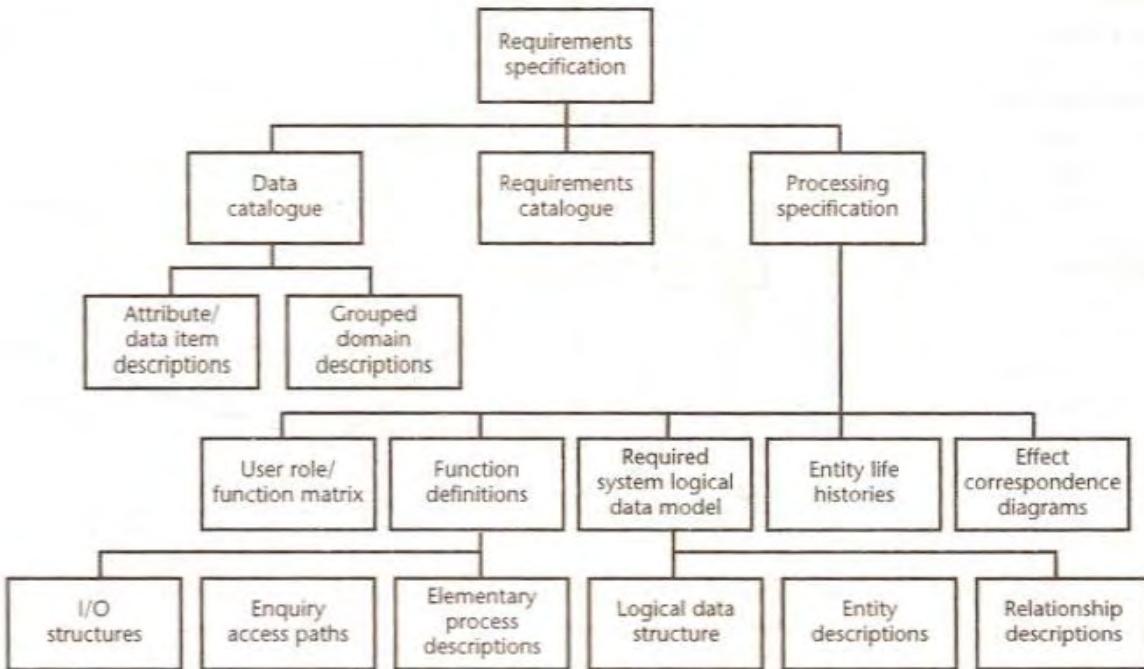
A fragment of an activity-based Work Breakdown Structure

Advantages

- The WES approach includes the belief that it is much more likely to result in a task catalogue
- That is complete and is composed of non-overlapping activities.
- Project's activities need to be sequenced in the sense of deciding which activities need to be completed before others can start.

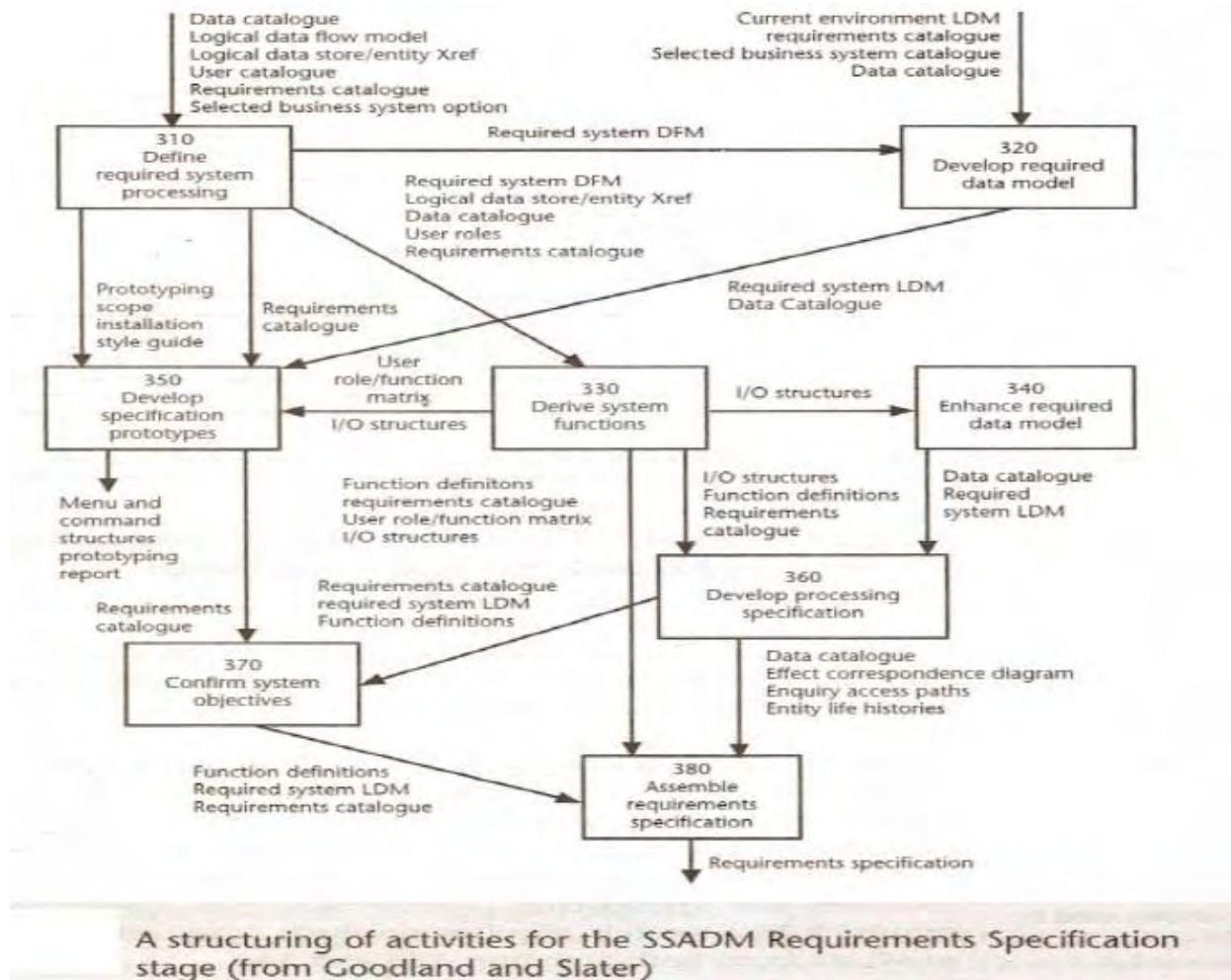
The product based approach: It consists of producing a Product Breakdown Structure and a Product Flow Diagram. The PFD indicates, for each product, which other products are required as inputs. The PFD can therefore be easily transformed into an ordered list of activities by identifying the transformations that turn products into others.

This approach is particularly appropriate if using a methodology such as Structured Systems Analysis and Design Method (SSADM), which clearly specifies, for each step or task, each of the products required and the activities required to produce it.



**SSADM Product Breakdown Structure for Requirements Specification
(adapted from Goodland and Slater)**

The SSADM Reference Manual also supplies generic activity networks and, using the project-specific PBS and derived PFD, these may be used as a basis for developing a project-specific activity network. Figure below illustrates an activity network for the activities required to create the products in above Figure. Notice how the development of a PFD leads directly to an activity network that indicates the sequencing of activities — in Figure below, activity 340 (Enhance required data model) requires products from activity 330 and activity 360 needs products from both activities 330 and 340.



The hybrid approach: The WBS is based entirely on a structuring of activities. WBS may be based upon the project's products as illustrated in Figure below, which is in turn based on a simple list of final deliverables and, for each deliverable, a set of activities required to produce that product. Figure below illustrates a flat WBS and it is likely that, in a project of any size, it would be beneficial to introduce additional levels — structuring both products and activities.

The degree to which the structuring is product-based or activity-based might be influenced by the nature of the project and the particular development method adopted. As with a purely activity-based WBS, having identified the activities we are then left with the task of sequencing them.

A framework dictating the number of levels and the nature of each level in the structure may be imposed on a NVBS. For example, in their Master Intern Training Plan (MITP) methodology, IBM recommend that the following five levels should be used in a WBS:

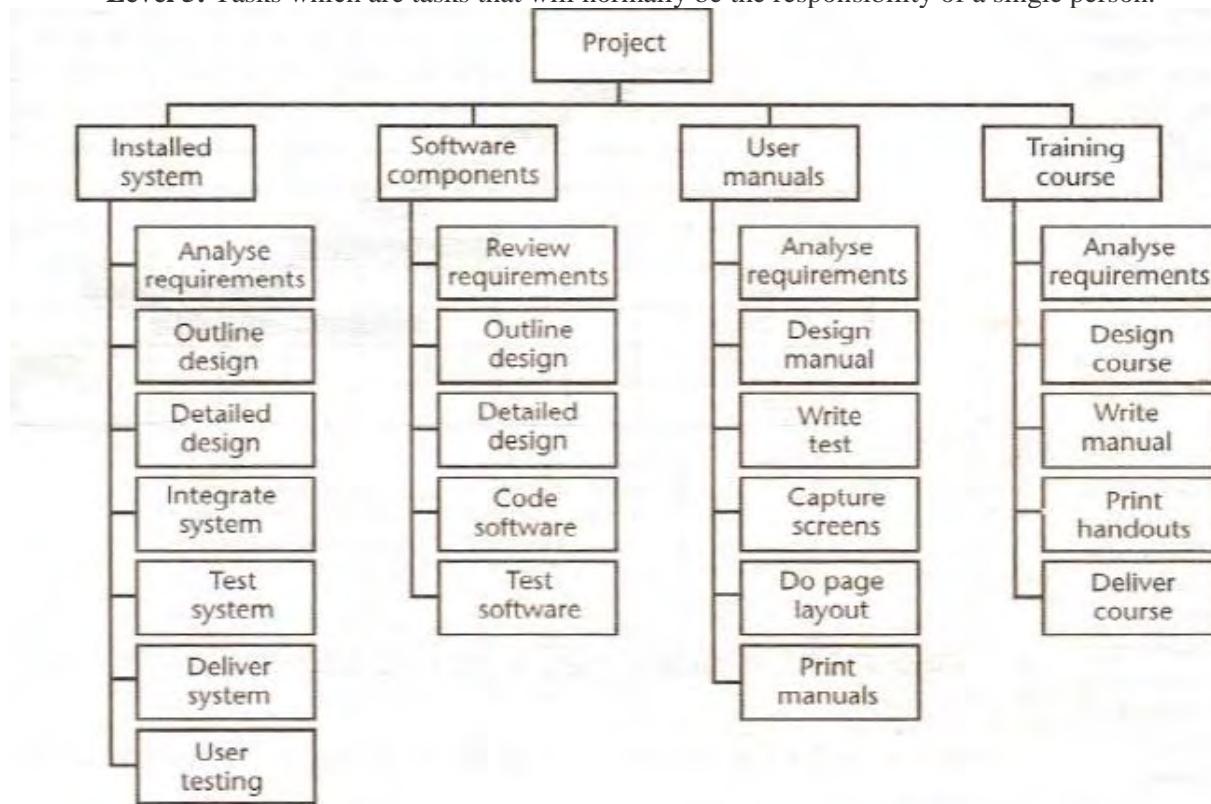
Level 1: Project.

Level 2: Deliverables such as software, manuals and training courses.

Level 3: Components which are the key work items needed to produce deliverables, such as the modules and tests required to produce the system software.

Level 4: Work-packages which are major work items, or collections of related tasks, required to produce a component.

Level 5: Tasks which are tasks that will normally be the responsibility of a single person.



A work breakdown structure based on deliverable

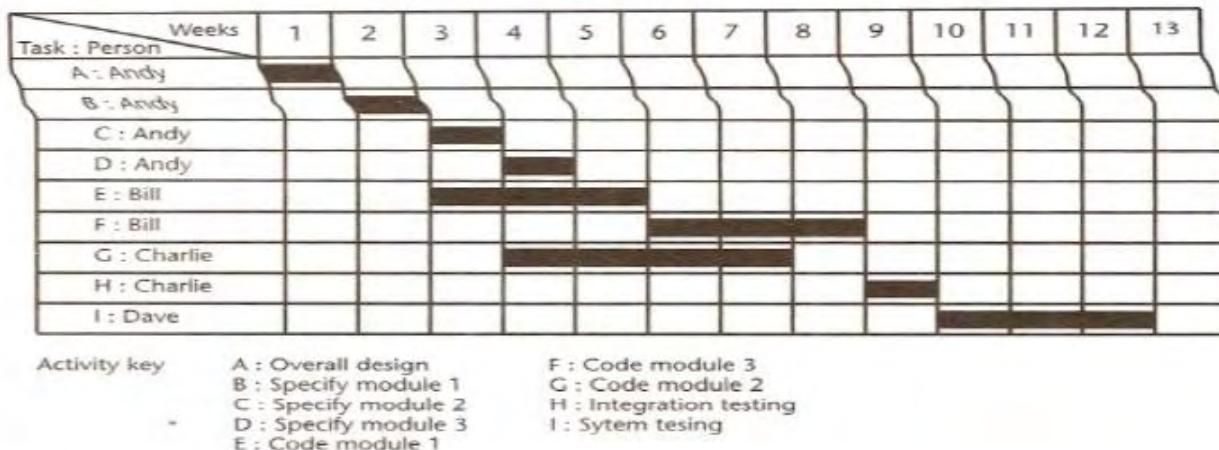
Sequencing and scheduling activities

Throughout a project, we will require a schedule that clearly indicates when each of the project's activities is planned to occur and what resources it will need. The chart shown has been drawn up taking account of the nature of the development process that is, certain tasks must be completed before others may start and the resources that are available (for example, activity C follows activity B because Andy cannot work on both tasks at the same time).

In drawing up the chart, we have therefore done two things — we have sequenced the tasks (that is, identified the dependencies among activities dictated by the development process) and scheduled them (that is, specified when they should take place). The scheduling has had to take account of the availability of staff and the ways in which the activities have been allocated to them. The schedule might look quite different were there a different number of staff or were we to allocate the activities differently.

In the case of small projects, this combined sequencing—scheduling approach might be quite suitable, particularly where we wish to allocate individuals to particular tasks at an early planning stage.

However, on larger projects it is better to separate out these two activities: to sequence the tasks according to their logical relationships and then to schedule them taking into account resources and other factors.

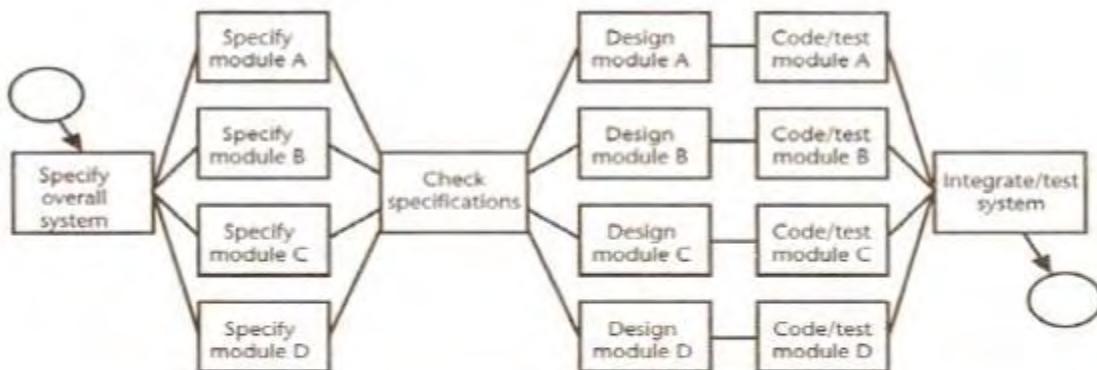


Network planning models: These project scheduling techniques model the project's activities and their relationships as a network. In the network, time flows from left to right. The two best known being CPM (Critical Path Method) and PERT (Program Evaluation Review Technique).

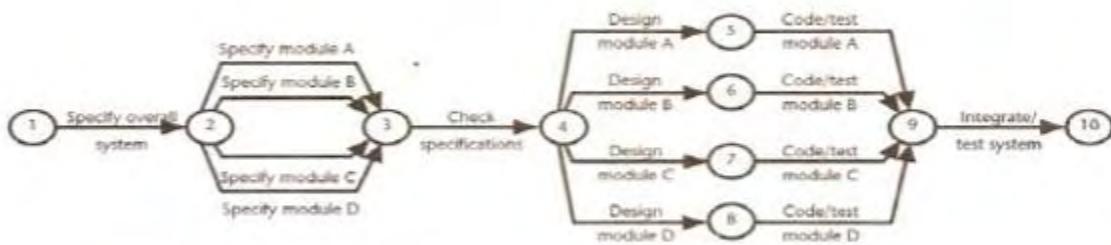
Both of these techniques used an **activity-on-arrow** approach to visualizing the project as a network where activities are drawn as arrows joining circles, or nodes which represent the possible start and/or completion of an activity or set of activities.

More recently a variation on these techniques, called **precedence network**, has become popular. This method uses activity-on-node networks where activities are represented as nodes and the links between nodes represent precedence (or sequencing) requirements.

This latter approach avoids some of the problems inherent in the activity-on-arrow representation and provides more scope for easily representing certain situations. It is this method that is adopted in the majority of computer applications currently available. These three methods are very similar and it must be admitted that many people use the same name (particularly CPM) indiscriminately to refer to any or all of the methods.



The IOE maintenance group accounts project activity network fragment with a checkpoint activity added

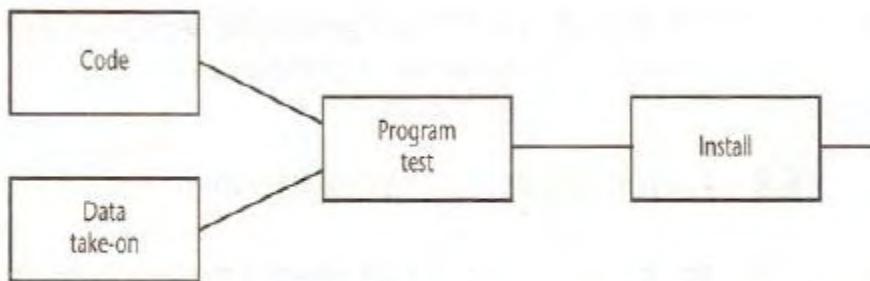


The IOE maintenance group accounts project activity network fragment represented as a CPM network

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

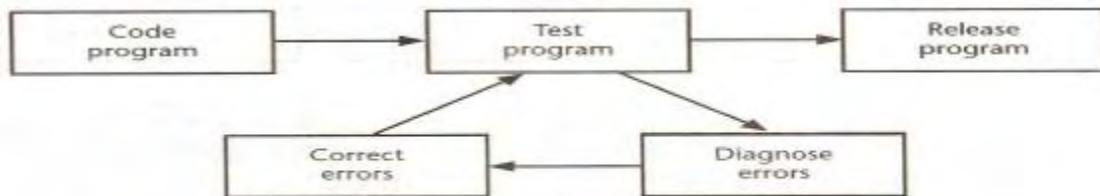
Constructing precedence networks

- A project network should have only one start node
- A project network should have only one end node
- A node has duration
- A node represents an activity and, in general, activities take time to execute.
- Links normally have no duration
- Precedents are the immediate preceding activities in Figure below, the activity 'Program test' cannot start until both 'Code' and 'Data take-on' have been completed and activity 'Install' cannot start until 'Program test' has finished. 'Code' and Data take-on can therefore be said to be precedents of 'Program test', and 'Program test' is a precedent of 'Install'. Note that we do not speak of 'Code' and 'Data take-on' as precedents of 'Install' - that relationship is implicit in the previous statement.
- Time moves front left to right



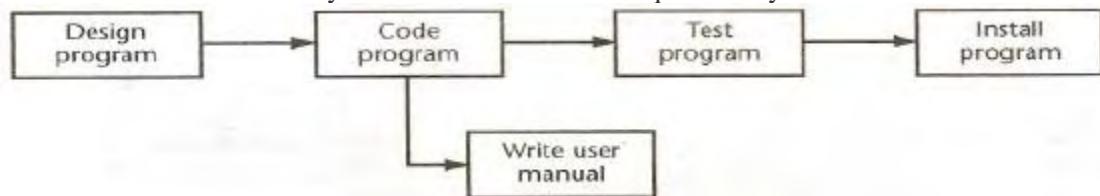
Fragment of a precedence network

A network may not contain loops. Figure below demonstrates a loop in a 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.

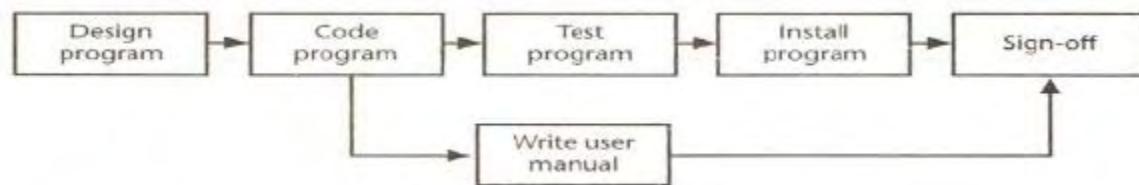


A loop represents an impossible sequence

A network should not contain dangles. A dangling activity such as 'Write user manual' in Figure below should not exist as it is likely to lead to errors in subsequent analysis.

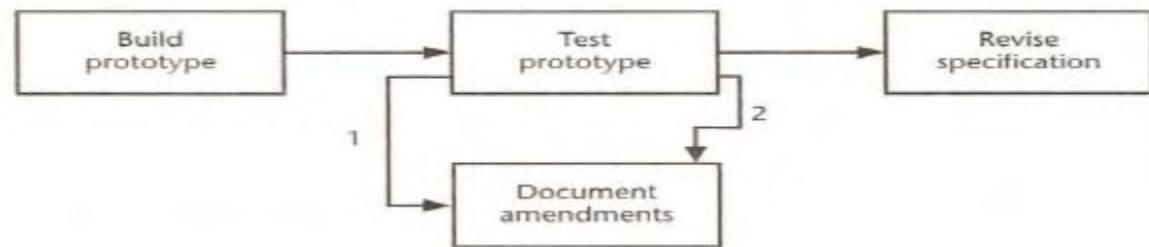


Redraw the network with a final completion activity — which, at least in this case, is probably a more accurate



Representing lagged activities

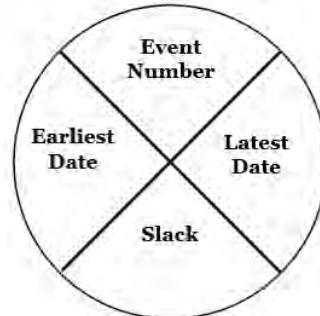
We might come across situations where we wished to undertake two activities in parallel so long as there is a lag between the two. We might wish to document amendments to a program as it was being tested - particularly if evaluating a prototype. Where activities can occur in parallel with a time lag between them we represent the lag with a duration on the linking arrow as shown in Figure below. This indicates that documenting amendments can start one day after the start of prototype testing and will be completed two days after prototype testing is completed.



Labelling conventions: There are a number of differing conventions that have been adopted for entering information on an activity-on-node network. One of the more common conventions for labelling nodes, and the one adopted here, is shown on the left. The activity label is usually a code developed to uniquely identify

the activity and may incorporate a project code. The activity description will normally be a brief activity name such as 'Test take-on module'.

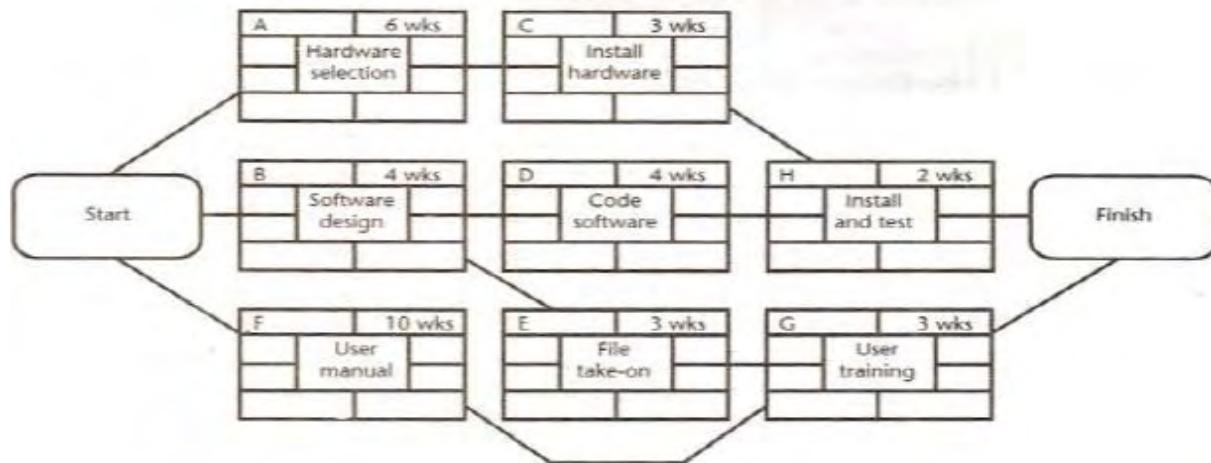
Activity Label		Duration
Earliest Start	Activity Description	Earliest Finish
Latest Start		Latest Finish
Activity Span		Float



Adding the time dimension

Having created the logical network model indicating what needs to be done and the interrelationships between those activities, we are now ready to start thinking about when each activity should be undertaken. The critical path approach is concerned with two primary objectives: planning the project in such a way that it is completed as quickly as possible; and identifying those activities where a delay in their execution is likely to affect the overall end date of the project or later activities' start dates.

The method requires that for each activity we have an estimate of its duration. The network is then analyzed by carrying out a forward pass, to calculate the earliest dates at which activities may commence and the project be completed, and a backward pass, to calculate the latest start dates for activities and the critical path. In practice we would use a software application to carry out these calculations for anything but the smallest of projects. It is important, though, that we understand how the calculations are carried out in order to interpret the results correctly and understand the limitations of the method. The description and example that follow use the small example project outlined below — a project composed of eight activities whose durations have been estimated as shown in the table



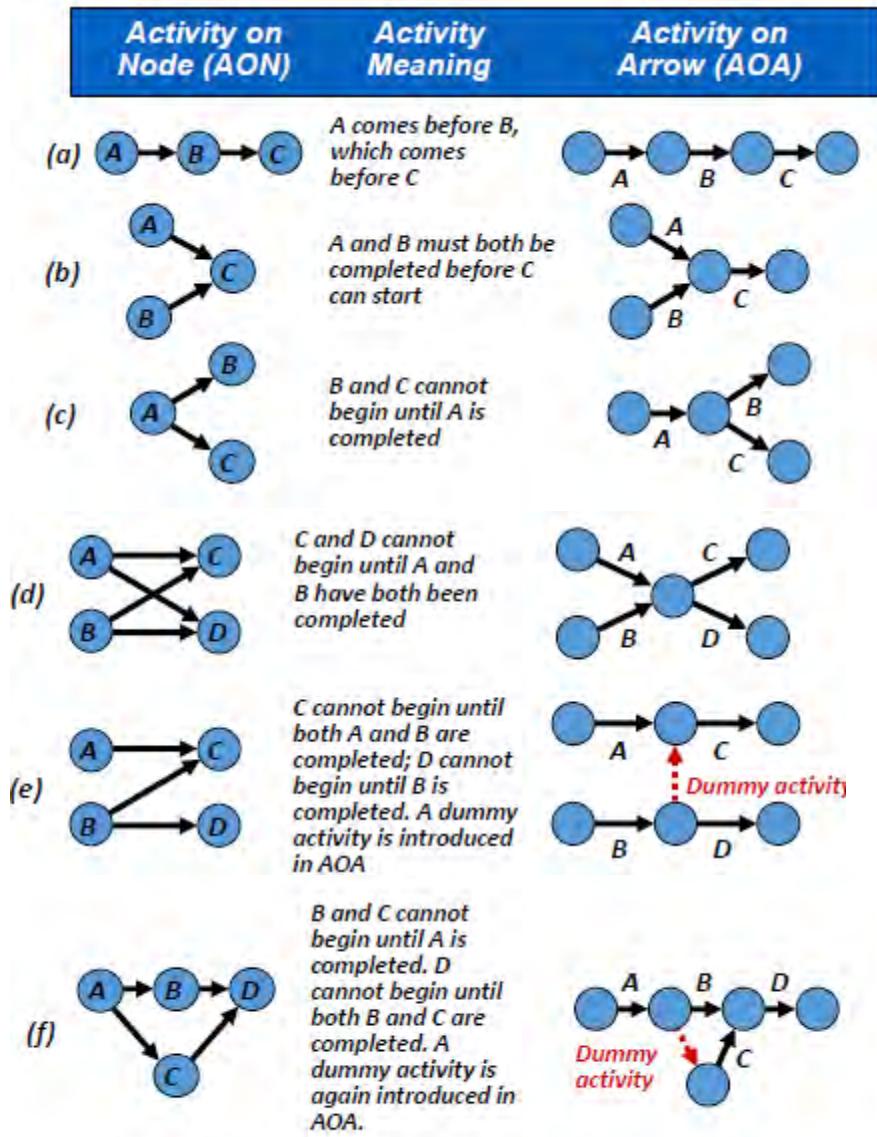
The precedence network for the example project

Activity in the Box (AIB)

- Also called as Activity-on-node (AON)
- nodes represent activities, and arrows show precedence relationships
- Each activity is represented by a box.
- The activity description is written in the box.
- Activity consume time, and their description usually starts with a verb
- Each activity is represented by one and only one box.
- Each box is assigned a unique activity number.
- Activities have a precedential relationship.
- Some activities may be done concurrently.

Activity on arrow

- Arrows represent activities and nodes are events for points in time
- Each activity is represented by an arrow.
- The activity description is written above the arrow.
- Each activity is represented by one and only one arrow
- The tail of the arrow designates the start of the activity.
- The head of the arrow designates the completion of the activity.
- The length and slope of the arrow are in no way indicative of the activity's duration or importance



Activity Scheduling

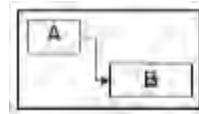
- It is necessary to select an estimated start time and a required completion time for the overall project.
- The project's required completion time is normally part of the project objective and stated in the contract.
- If a project must be completed in 130 working days, we define the project's estimated start time as 0 and its required completion time is day 130.
- A project schedule includes:
 - the earliest times (or dates) at which each activity can start and finish, based on the project's estimated start time (or date)

- the latest times (or dates) by which each activity must start and finish in order to complete the project by its required completion time (or date)

Task Dependency Relationships

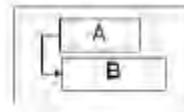
➤ **Finish-to-Start (FS)**

- B cannot start till A finishes
- A: Construct fence; B: Paint Fence



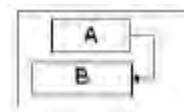
➤ **Start-to-Start (SS)**

- B cannot start till A starts
- A: Pour foundation; B: Level concrete



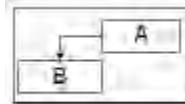
➤ **Finish-to-Finish (FF)**

- B cannot finish till A finishes
- A: Add wiring; B: Inspect electrical

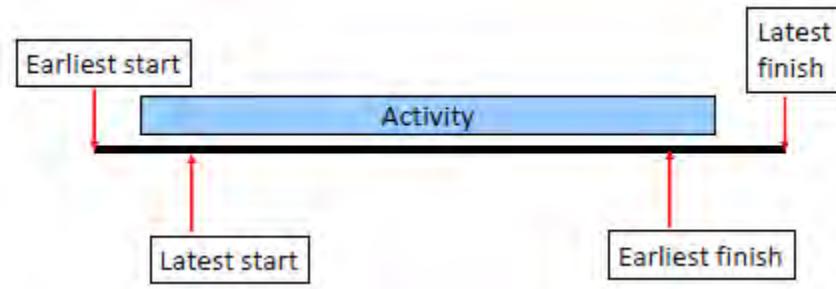


➤ **Start-to-Finish (SF)**

- B cannot finish till A starts (rare)



Start and finish times



- Earliest start time (ES) is the earliest time at which a particular activity can begin.
- Earliest finish time (EF) is the earliest time by which a particular activity can be completed.
- $EF = ES + \text{Duration Estimate}$
- Latest finish time (LF) is the latest time by which a particular activity must be finished in order for the entire project to be completed by its required completion time, calculated on the basis of the project's required completion time and the duration estimates for succeeding activities.
- Latest start time (LS) is the latest time by which a particular activity must be started in order for the entire project to be completed by its required completion time, calculated by subtracting the activity's duration estimate from the activity's latest finish time:
- $LS = LF - \text{Duration Estimate}$

Critical Path and Critical Activities

Consider all the paths in a project, beginning with the start event and stopping at the end event. For each path, calculate the time of execution, by adding the time for the individual activities in that path.

The path with the largest time is called the critical path and the activities along this path are called the critical activities or bottleneck activities. The activities are called critical because they cannot be delayed. However, a non-critical activity may be delayed to a certain extent. Any delay in a critical activity will delay the completion of the whole project. However, a certain permissible delay in a non-critical activity will not delay the completion of the whole project. It shall be noted that delay in a non-critical activity beyond a limit would certainly delay the completion the whole project. Sometimes, there may be several critical paths for a project. A project manager shall pay special attention to critical activities.

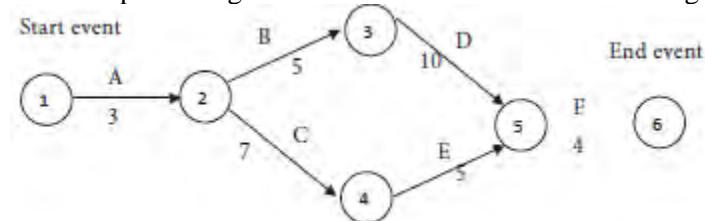
The following details are available regarding a project:

Activity	Predecessor Activity	Duration (Weeks)
A	-	3
B	A	5
C	A	7
D	B	10
E	C	5
F	D,E	4

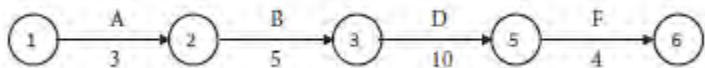
Determine the critical path, the critical activities and the project completion time.

Solution

First let us construct the network diagram for the given project. We mark the time estimates along the arrows representing the activities. We obtain the following diagram:

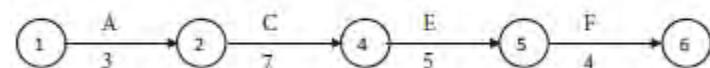


Path I



with a time of $3 + 5 + 10 + 4 = 22$ weeks.

Path II



with a time of $3 + 7 + 5 + 4 = 19$ weeks.

Compare the times for the two paths. Maximum of $\{22, 19\} = 22$.

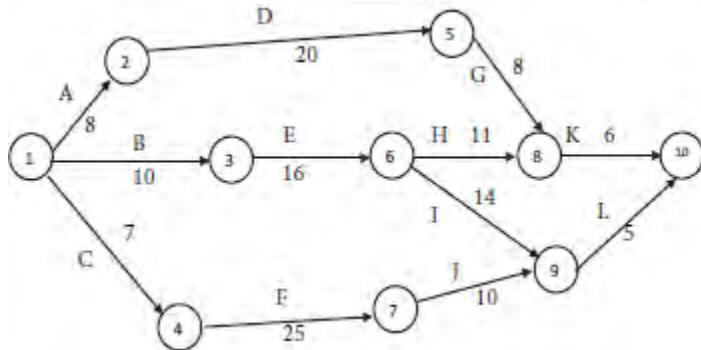
We see that path I has the maximum time of 22 weeks. Therefore, path I is the critical path. The critical activities are A, B, D and F. The project completion time is 22 weeks.

We notice that C and E are non-critical activities.

Time for path I - Time for path II = $22 - 19 = 3$ weeks.

Therefore, together the non-critical activities can be delayed up to a maximum of 3 weeks, without delaying the completion of the whole project.

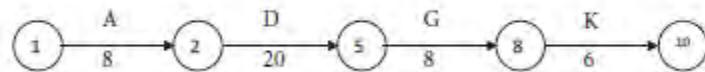
Find out the completion time and the critical activities for the following project:



Solution

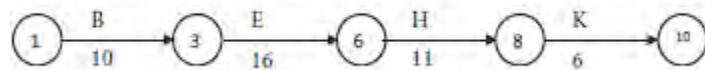
In all, we identify 4 paths, beginning with the start node of 1 and terminating at the end node of 10. They are as follows:

Path I



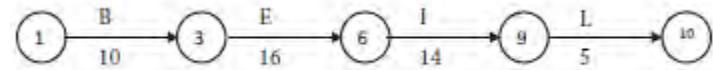
Time for the path = $8 + 20 + 8 + 6 = 42$ units of time.

Path II



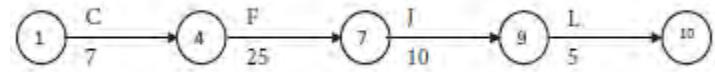
Time for the path = $10 + 16 + 11 + 6 = 43$ units of time.

Path III



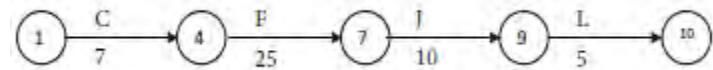
Time for the path = $10 + 16 + 14 + 5 = 45$ units of time.

Path IV



Time for the path = $7 + 25 + 10 + 5 = 47$ units of time.

Compare the times for the four paths. Maximum of $\{42, 43, 45, 47\} = 47$. We see that the following path has the maximum time and so it is the critical path:



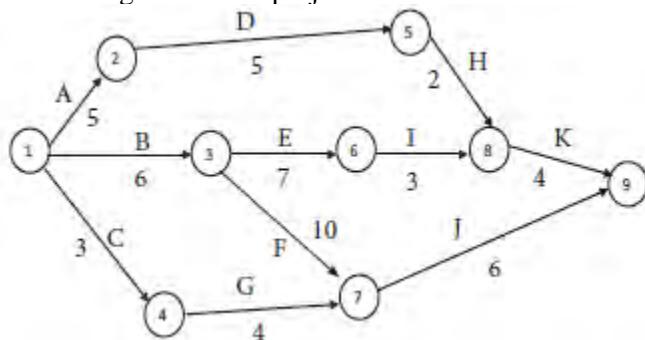
The critical activities are C, F, J and L. The non-critical activities are A, B, D, E, G, H, I and K. The project completion time is 47 units of time.

Draw the network diagram and determine the critical path for the following project:

Activity	Time estimate (Weeks)
1 - 2	5
1 - 3	6
1 - 4	3
2 - 5	5
3 - 6	7
3 - 7	10
4 - 7	4
5 - 8	2
6 - 8	5
7 - 9	6
8 - 9	4

Solution

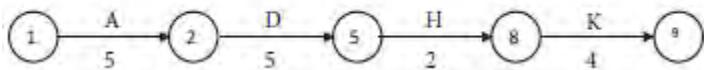
We have the following network diagram for the project:



Solution

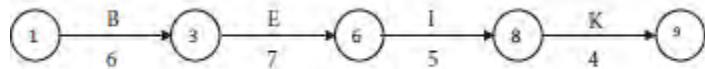
We assert that there are 4 paths, beginning with the start node of 1 and terminating at the end node of 9. They are as follows:

Path I



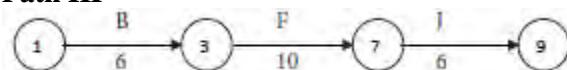
Time for the path = $5 + 5 + 2 + 4 = 16$ weeks.

Path II



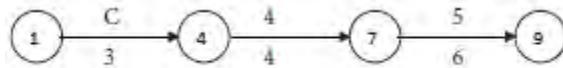
Time for the path = $6 + 7 + 5 + 4 = 22$ weeks.

Path III



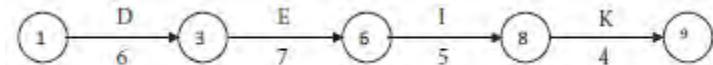
Time for the path = $6 + 10 + 6 = 16$ weeks.

Path IV



Time for the path = $3 + 4 + 6 = 13$ weeks.

Compare the times for the four paths. Maximum of $\{16, 22, 16, 13\} = 22$. We see that the following path has the maximum time and so it is the critical path:



The critical activities are B, E, I and K. The non-critical activities are A, C, D, F, G, H and J. The project completion time is 22 weeks.

Program Evaluation and Review Technique (PERT)

Program Evaluation and Review Technique (PERT) is a tool that would help a project manager in project planning and control? It would enable him in continuously monitoring a project and taking corrective measures wherever necessary. This technique involves statistical methods.

In PERT, we assume that it is not possible to have precise time estimate for each activity and instead, probabilistic estimates of time alone are possible. A multiple time estimate approach is followed here. In probabilistic time estimate, the following 3 types of estimate are possible:

Pessimistic time estimate (t_p)

Optimistic time estimate (t_o)

Most likely time estimate (t_m)

The optimistic estimate of time is based on the assumption that an activity will not involve any difficulty during execution and it can be completed within a short period. On the other hand, a pessimistic estimate is made on the assumption that there would be unexpected problems during the execution of an activity and hence it would consume more time. The most likely time estimate is made in between the optimistic and the pessimistic estimates of time. Thus the three estimates of time have the relationship

$$t_o \leq t_m \leq t_p$$

Practically speaking, neither the pessimistic nor the optimistic estimate may hold in reality and it is the most likely time estimate that is expected to prevail in almost all cases. Therefore, it is preferable to give more weight to the most likely time estimate. We give a weight of 4 to most likely time estimate and a weight of 1 each to the pessimistic and optimistic time estimates. We arrive at a time estimate (t_e) as the weighted average of these estimates as follows:

$$t_e = \frac{t_o + 4t_m + t_p}{6}$$

Since we have taken 6 units (1 for t_p , 4 for t_m and 1 for t_o), we divide the sum by 6. With this time estimate, we can determine the project completion time as applicable for CPM.

Since PERT involves the average of three estimates of time for each activity, this method is very practical and the results from PERT will be have a reasonable amount of reliability.

Measure of Certainty

The 3 estimates of time are such that

$$t_o \leq t_m \leq t_p$$

Therefore the range for the time estimate is $t_p - t_o$.

The time taken by an activity in a project network follows a distribution with a standard deviation of one sixth of the range, approximately.

i.e., The standard deviation = $\sigma = \frac{t_p - t_o}{6}$

$$\text{and the variance} = \sigma^2 = \left(\frac{t_p - t_o}{6} \right)^2$$

The certainty of the time estimate of an activity can be analyzed with the help of the variance. The greater the variance, the more uncertainty in the time estimate of an activity.

Two experts A and B examined an activity and arrived at the following time estimates.

Expert	Time Estimate		
	t_o	t_m	t_p
A	4	6	8
B	4	7	10

Determine which expert is more certain about his estimates of time:

Solution

$$\text{Variance } (\sigma^2) \text{ in time estimates} = \left(\frac{t_p - t_o}{6} \right)^2$$

$$\text{In the case of expert A, the variance} = \left(\frac{8-4}{6} \right)^2 = \frac{4}{9}$$

$$\text{As regards expert B, the variance} = \left(\frac{10-4}{6} \right)^2 = 1$$

So, the variance is less in the case of A. Hence, it is concluded that the expert A is more certain about his estimates of time.

Find out the time required to complete the following project and the critical activities:

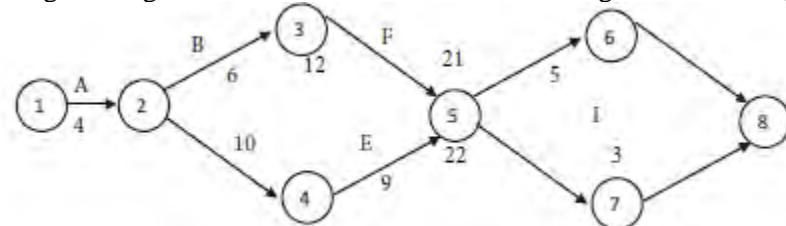
Activity	Predecessor Activity	Optimistic time estimate (to days)	Most likely time estimate (tm days)	Pessimistic time estimate (tp days)
A	-	2	4	6
B	A	3	6	9
C	A	8	10	12
D	B	9	12	15
E	C	8	9	10
F	D, E	16	21	26
G	D, E	19	22	25
H	F	2	5	8
I	G	1	3	5

Solution

From the three time estimates t_p , t_m and t_o , calculate t_e for each activity. We obtain the following table:

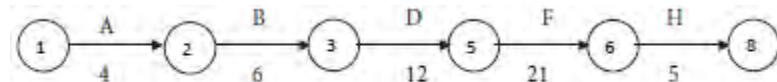
Activity	Optimistic time estimate (t_o)	4 x Most likely time estimate	Pessimistic time estimate (t_p)	$t_o + 4t_m + t_p$	Time estimate $t_e = \frac{t_o + 4t_m + t_p}{6}$
A	2	16	6	24	4
B	3	24	9	36	6
C	8	40	12	60	10
D	9	48	15	72	12
E	8	36	10	54	9
F	16	84	26	126	21
G	19	88	25	132	22
H	2	20	8	30	5
I	1	12	5	18	3

Using the single time estimates of the activities, we get the following network diagram for the project.



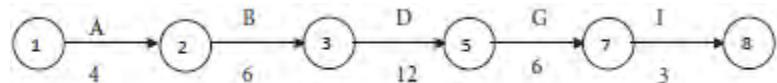
Consider the paths, beginning with the start node and stopping with the end node. There are four such paths for the given project. They are as follows:

Path I



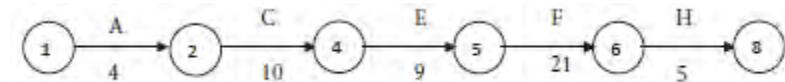
Time for the path: $4+6+12+21+5 = 48$ days.

Path II



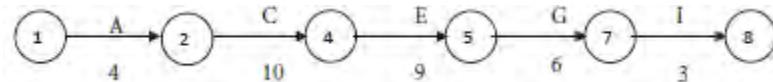
Time for the path: $4+6+12+6+3 = 31$ days.

Path III



Time for the path: $4+10+9+21+5 = 49$ days.

Path IV



Time for the path: $4+10+9+6+3 = 32$ days.

Compare the times for the four paths.

Maximum of $\{48, 31, 49, 32\} = 49$.

We see that Path III has the maximum time.

Therefore, the critical path is Path III. i.e., 1 2 4 5 6 8.

The critical activities are A, C, E, F and H.

The non-critical activities are B, D, G and I.

Project time (Also called project length) = 49 days.

Calculation of Standard Deviation and Variance for the Critical Activities:

Critical Activity	Optimistic time estimate (t_o)	Most likely time estimate (t_m)	Pessimistic time estimate (t_p)	Range ($t_p - t_o$)	Standard deviation = $\sigma = \frac{t_p - t_o}{6}$	Variance = $\sigma^2 = \left(\frac{t_p - t_o}{6}\right)^2$
A: 1 2	3	6	9	6	1	1
C: 2 3	6	12	18	12	2	4
E: 3 5	8	11	14	6	1	1
I: 5 8	2	4	6	4	2/3	4/9

Variance of project time (Also called Variance of project length) = Sum of the variances for the critical activities = $1+4+1+4/9 = 58/9$ Weeks.

Standard deviation of project time = $\sqrt{\text{Variance}} = \sqrt{58/9} = 2.54$ weeks.

A project consists of seven activities with the following time estimates. Find the probability that the project will be completed in 30 weeks or less.

Activity	Predecessor Activity	Optimistic time estimate (to days)	Most likely time estimate (tm days)	Pessimistic time estimate (tp days)
A	-	2	5	8
B	A	2	3	4
C	A	6	8	10
D	A	2	4	6
E	B	2	6	10
F	C	6	7	8
G	D, E, F	6	8	10

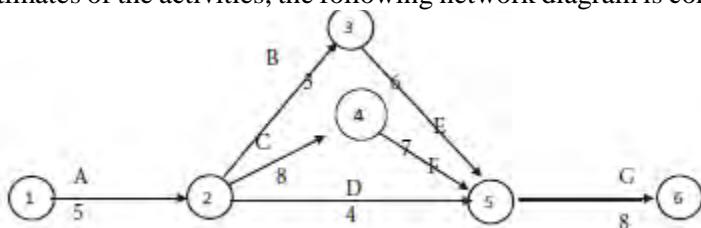
Solution

From the three time estimates t_p , t_m and t_o calculate for each activity.

The results are furnished in the following table:

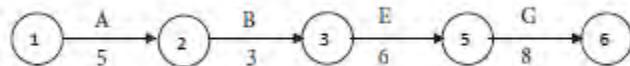
Activity	Optimistic time estimate (to)	4 x Most likely time estimate	Pessimistic time estimate (tp)	to + 4tm + tp	Time estimate $t_e = \frac{t_o + 4t_m + t_p}{6}$
A	2	20	8	30	5
B	2	12	4	18	3
C	6	32	10	48	8
D	2	16	6	24	4
E	2	24	10	36	6
F	6	28	8	42	7
G	6	32	10	48	8

With the single time estimates of the activities, the following network diagram is constructed for the project.



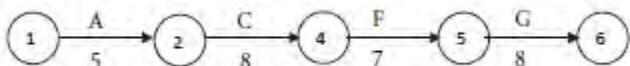
Consider the paths, beginning with the start node and stopping with the end node. There are three such paths for the given project. They are as follows:

Path I



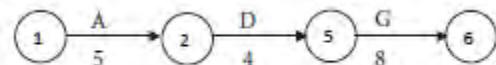
Time for the path: $5+3+6+8 = 22$ weeks.

Path II



Time for the path: $5+8+7+8 = 28$ weeks.

Path III



Time for the path: $5+4+8 = 17$ weeks.

Compare the times for the three paths.

Maximum of $\{22, 28, 17\} = 28$.

It is noticed that Path II has the maximum time.

Therefore, the critical path is Path II. i.e., 1 2 4 5 6.

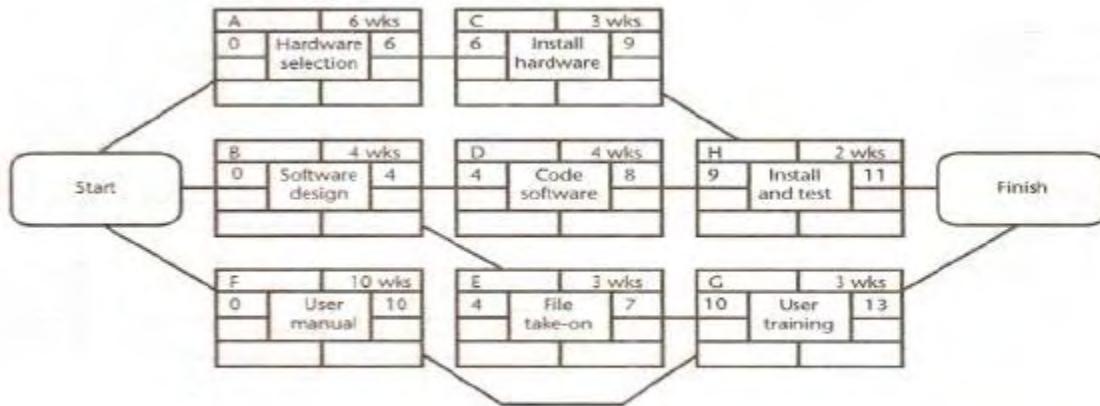
The critical activities are A, C, F and G.

The non-critical activities are B, D and E.

Project time = 28 weeks.

The forward pass: The forward pass is carried out to calculate the earliest dates on which each activity may be started and completed. Where an actual start date is known, the calculations may be carried out

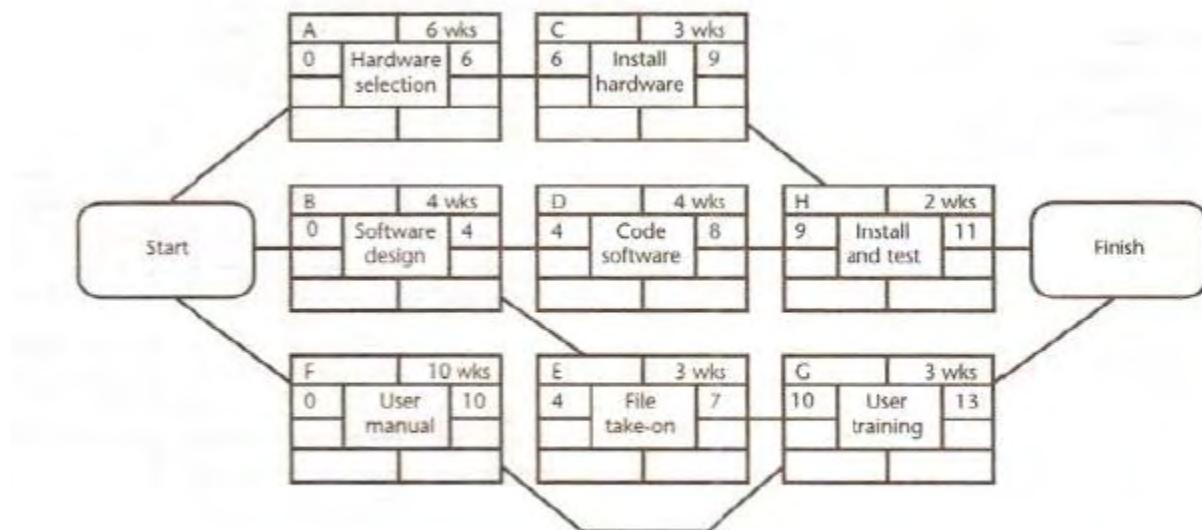
using actual dates. Alternatively, we can use day or week numbers and that is the approach we shall adopt here. By convention, dates indicate the end of a period and the project is therefore shown as starting at the end of week zero (or the beginning of week 1). The forward pass and the calculation of earliest start dates is calculated according to the following reasoning.



The network after the forward pass.

Activities A, B and F may start immediately, so the earliest date for their start is zero.

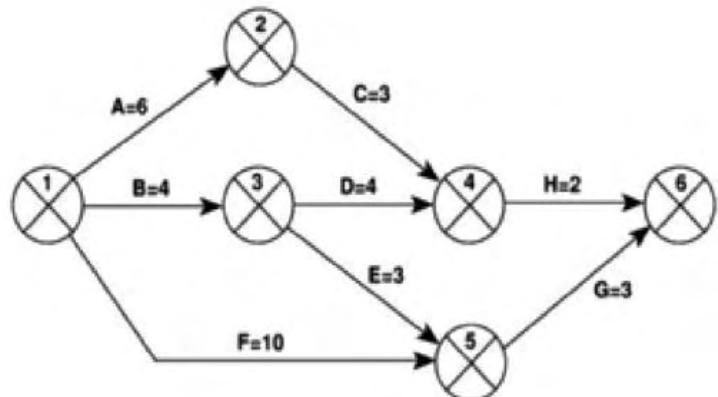
- Activity A will take 6 weeks, so the earliest it can finish is week 6.
- Activity B will take 4 weeks, so the earliest it can finish is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10.
- Activity C can start as soon as A has finished so its earliest start date is week 6. It will take 3 weeks so the earliest it can finish is week 9.
- Activities D and E can start as soon as B is complete so the earliest they can each start is week 4. Activity D, which will take 4 weeks, can therefore finish by week 8 and activity E, which will take 3 weeks, can therefore finish by week 7.
- Activity G cannot start until both E and F have been completed. It cannot therefore start until week 10 — the later of weeks 7 (for activity E) and 10 (for activity F). It takes 3 weeks and finishes in week 13.
- Similarly, Activity H cannot start until week 9 — the later of the two earliest finished dates for the preceding activities C and a
- The project will be complete when both activities H and G have been completed. Thus the earliest project completion date will be the later of weeks 11 and 13— that is, week 13.
- The results of the forward pass are shown in Figure below.



The network after the forward pass.

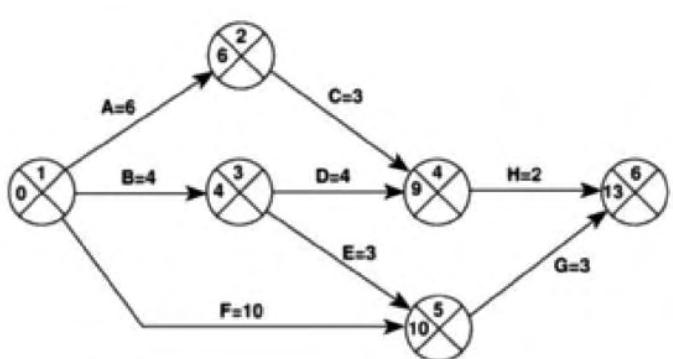
Forward pass example

Activity	Duration (Weeks)	Earliest Start Date (Weeks)	Earliest Finish Date (Weeks)
A	6	0	6
B	4	0	4
C	3	6	9
D	4	4	8
E	3	4	7
F	10	0	10
G	3	10	13
H	2	9	11



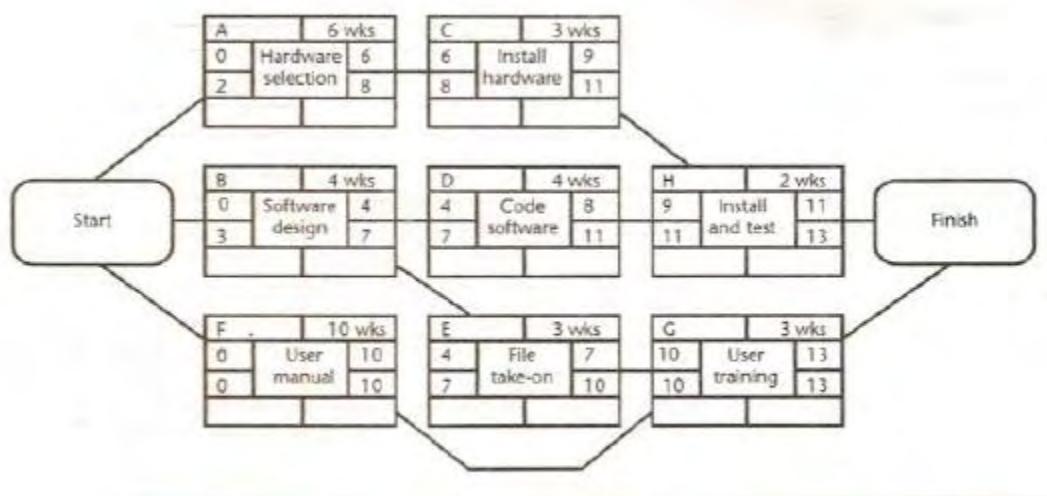
CPM Network after forward Pass

Activity	Duration (weeks)	Earliest Start Date	Latest Start Date	Earliest Finish Date	Latest Finish Date	Total Float
A	6	0		6		
B	4	0		4		
C	3	6		9		
D	4	4		8		
E	3	4		7		
F	10	0		10		
G	3	10		13		
H	2	9		11		



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

Figure below illustrates our network after carrying out the backward pass.



The network after the backward pass

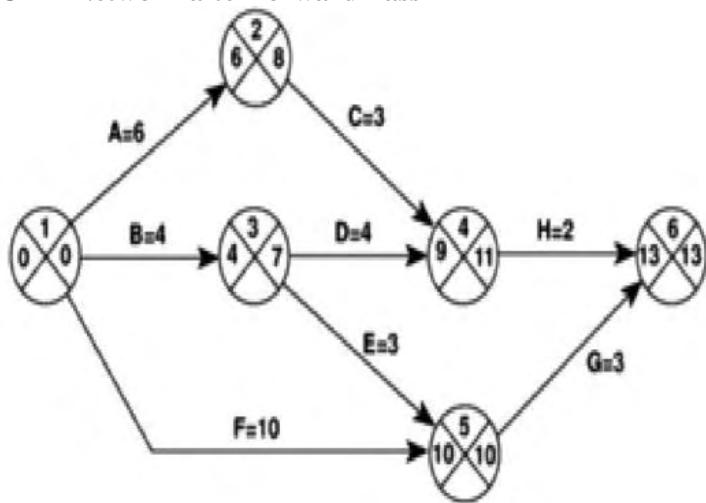
The latest activity dates are calculated as follows.

- The latest completion date for activities G and H is assumed to be week 13.
- Activity H must therefore start at week 11 at the latest (13-2) and the latest start date for activity G is week 10 (13-3).
- The latest completion date for activities C and D is the latest date at which activity H must start — that is, week 11. They therefore have latest start dates of week 8 (11-3) and week 7 (11-4) respectively.
- Activities E and F must be completed by week 10 so their earliest start dates are weeks 7 (10-3) and 0 (10-10) respectively.
- Activity B must be completed by week 7 the latest start date for both activities D and E is week 3 (7-4).
- Activity A must be completed by week 8 (the latest start date for activity C) so its latest start is week 2 (8-6).
- The latest start date for the project start is the earliest of the latest start dates for activities A, B and F. This is week zero. This is, of course, not very surprising since it tells us that if the project does not start on time it won't finish on time.

Example backward pass

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date
A	6	0	2	6	8
B	4	0	3	4	7
C	3	6	8	9	11
D	4	4	7	8	11
E	3	4	7	7	10
F	10	0	0	10	10
G	3	10	10	13	13
H	2	9	11	11	13

CPM Network after forward Pass



Activity float

Although the total float is shown for each activity, it really 'belongs' to a path through the network. Activities A and C in Figure above each have 2 weeks' total float. If, however, activity A uses up its float (that is, it is not completed until week 8) then activity B will have zero float (it will have become critical). In such circumstances it may be misleading and detrimental to the project's success to publicize total float!

There are a number of other measures of activity float, including the following:

- **Free float:** the time by which an activity may be delayed without affecting any subsequent activity. It is calculated as the difference between the earliest completion date for the activity and the earliest start date of the succeeding activity. This might be considered a more satisfactory measure of float for publicizing to the staff involved in undertaking the activities.
- **Interfering float:** the difference between total float and free float. This is quite commonly used, particularly in association with the free float. Once the free float has been used (or if it is zero), the interfering float tells us by how much the activity may be delayed without delaying the project end date - even though it will delay the start of subsequent activities.

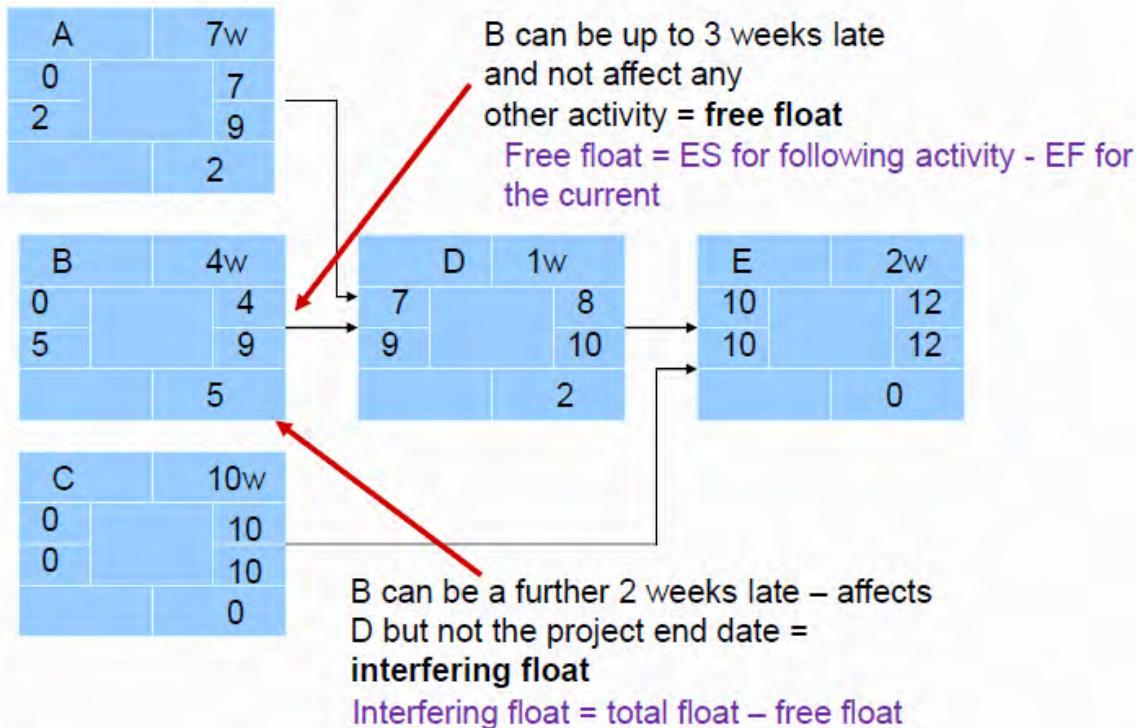
$$\text{Total float} = \text{LF} - \text{ES} - \text{duration} \text{ (or LS-ES or LF-EF)}$$

$$\text{Free float} = \text{ES for following activity} - \text{EF for the current}$$

$$\text{Interfering float} = \text{total float} - \text{free float}$$

Calculate the free float and interfering float for each of the activities shown in the activity network

Total float = LF – ES – duration



Shortening the project duration

If we wish to shorten the overall duration of a project, we would normally consider attempting to reduce activity durations. In many cases this can be done by applying more resources to the task - working overtime or procuring additional staff, for example. The critical path indicates where we must look to save time - if we are trying to bring forward the end date of the project, there is clearly no point in attempting to shorten non-critical activities. Referring to Figure above, it can be seen that we could complete the project in week 12 by reducing the duration of activity F by one week (to 9 weeks).

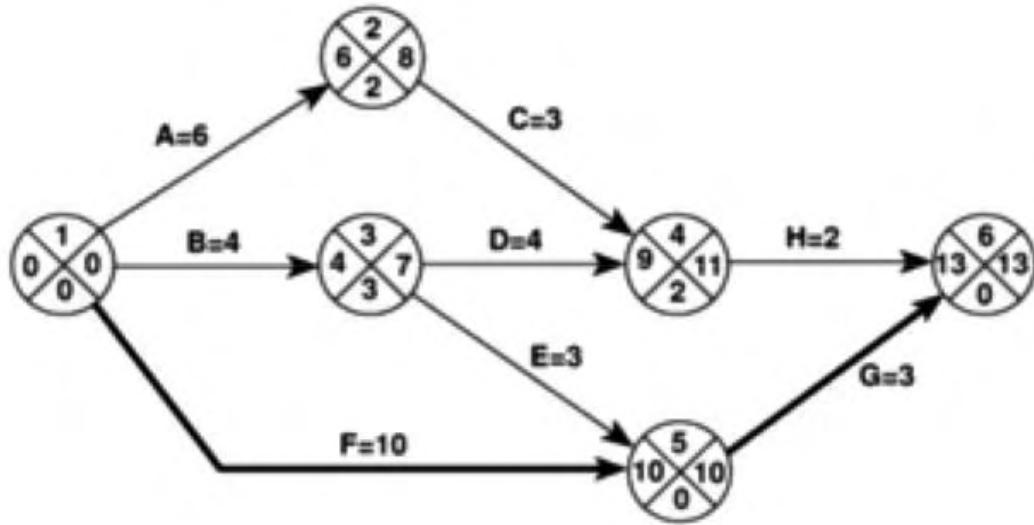
Referring to Figure above, suppose that the duration for activity F is shortened to 8 weeks. Calculate the end date for the project.

- What would the end date for the project be if activity F were shortened to 7 weeks? Why?
- As we reduce activity times along the critical path we must continually check for any new critical path emerging and redirect our attention where necessary.
- There will come a point when we can no longer safely, or cost-effectively, reduce critical activity durations in an attempt to bring forward the project end date. Further savings, if needed, must be sought in a consideration of our work methods and by questioning the logical sequencing of activities. Generally, time savings are to be found by increasing the amount of parallelism in the network and the removal of bottlenecks (subject always, of course, to resource and quality constraints).

Identifying the critical path

Slack: It is the difference between the earliest date and the latest date for an event-it is a measure of how late an event may be without affecting the end date of the project.

- The critical path is the path joining all nodes with a zero slack.
- The longest path in the overall network diagram is called the critical path.
- One way to determine which activities make up the critical path is to find which ones have the least slack.
- All the activities with this value are on the critical path of activities.
- The values of total slack for the consumer market study project are shown in Figure



Critical Path = F-G

Identifying the critical path

The activity schedule showing total float for each activity

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date	Total float
A	6	0	2	6	8	2
B	4	0	3	4	7	3
C	3	6	8	9	11	2
D	4	4	7	8	11	3
E	3	4	7	7	10	3
F	10	0	0	10	10	0
G	3	10	10	13	13	0
H	2	9	11	11	13	2

Unit: 4

Activity Planning

7 Hrs.

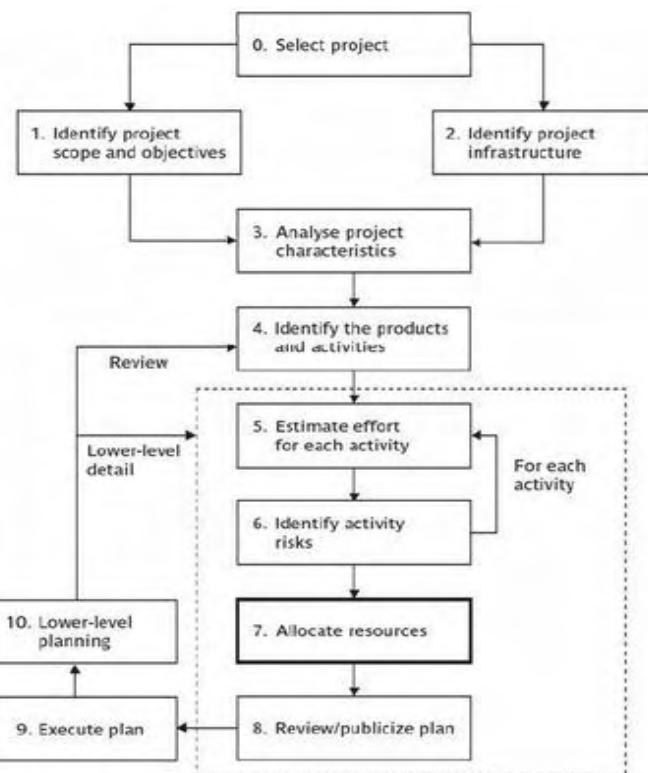
Activity Schedule showing total float for each activity

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish date	Latest finish date	Total float
A	6	0	2	6	8	2
B	4	0	3	4	7	3
C	3	6	8	9	11	2
D	4	4	7	8	11	3
E	3	4	7	7	10	3
F	10	0	0	10	10	0
G	3	10	10	13	13	0
H	2	9	11	11	13	2

Topics Covered:

- Introduction
 - The nature of resources
 - Identifying resource requirements
 - Scheduling resources
 - Creating Critical Paths
 - Counting the cost
 - Being Specific
 - Publishing the resource schedule
 - Cost schedule
 - The scheduling sequence

After the activities have been identified using various techniques and tabulated into a Work-Break-Down their sources need to be allocated to complete the identified tasks. Resource allocation in project management is concerned with creating a plan which can help achieve future goals. There are many resources which have to be allocated when managing a project, *beginning from budget to equipment and tools, to data and the project's plan*. This process is considered resource allocation



The final result of resource allocation will normally be a number of schedules including:

- **Activity schedule:** indicating start and completion dates for each activity
- **Resource schedule:** indicating dates when resources are needed and level of resources
- **Cost schedule:** shows accumulative expenditures.

Who allocates resources? Project manager

A project manager is a person who has the overall responsibility for the successful initiation, planning, design, execution, monitoring, controlling and closure of a project. Construction, petrochemical, architecture, information technology and many different industries that produce products and services use this job title.

The project manager must have a combination of skills including an ability to ask penetrating questions, detect unstated assumptions and resolve conflicts, as well as more general management skills.

Key among a project manager's duties is the recognition that risk directly impacts the likelihood of success and that this risk must be both formally and informally measured throughout the lifetime of a project.

Risks arise from uncertainty, and the successful project manager is the one who focuses on this as their primary concern. Most of the issues that impact a project result in one way or another from risk. A good project manager can lessen risk significantly, often by adhering to a policy of open communication, ensuring every significant participant has an opportunity to express opinions and concerns.

A project manager is a person who is responsible for making decisions, both large and small. The project manager should make sure they control risk and minimize uncertainty. Every decision the project manager makes must directly benefit their project.

Project managers use project management software, such as Microsoft Project, to organize their tasks and workforce. These software packages allow project managers to produce reports and charts in a few minutes, compared with the several hours it can take if they do it by hand.

Roles and Responsibilities

The role of the project manager encompasses many activities including:

- Planning and Defining Scope
- Activity Planning and Sequencing
- Resource Planning
- Developing Schedules
- Time Estimating
- Cost Estimating
- Developing a Budget
- Documentation
- Creating Charts and Schedules
- Risk Analysis
- Managing Risks and Issues
- Monitoring and Reporting Progress
- Team Leadership
- Strategic Influencing
- Business Partnering
- Working with Vendors
- Scalability, Interoperability and Portability Analysis
- Controlling Quality
- Benefits Realization

Finally, senior management must give a project manager support and authority if he or she is going to be successful.

The nature of resources

In project management, resources are required to carry out the project tasks. They can be people, equipment, facilities, funding, or anything else capable of definition (usually other than labor) required for the

completion of a project activity. The lack of a resource will therefore be a constraint on the completion of the project activity. Resources may be storable or non-storable. Storable resources remain available unless depleted by usage, and may be replenished by project tasks which produce them. Non-storable resources must be renewed for each time period, even if not used in previous time periods.

Allocation of limited resources is based on the priority given to each of the project activities. Their priority is calculated using the Critical path method and heuristic analysis. For a case with a constraint on the number of resources, the objective is to create the most efficient schedule possible - minimizing project duration and maximizing the use of the resources available.

Resource will fall into one of following category

- Labor
- Equipment
- Material
- Space
- Services
- Time
- Money

Classification of Resources

Resources can be viewed from four standpoints:

1. Availability

- **Recurring** – such as human resources that are available in the same quantity day-after-day – spending these resources does not deplete them.
- **Depleting** – like time and money – spending these resources depletes them.

2. Place of Availability

- **Immovable** – available wherever they are situated – they cannot be moved from place to place, such as computer resources.
- **Movable** – these resources can be moved from place to place – such as human resources and money.

3. Elasticity

- **Elastic Resources** – their supply can be increased or decreased – such as human resources and money.
- **Plastic Resources** – their supply cannot be extended, such as time.

4. Shared and Dedicated

- **Dedicated resources** – these should be dedicated to the project for the required duration – such as computer systems and programmers.
- **Shared resources** – these are resources that are needed for short durations but would not be utilized for the entire duration of the project – such as DBAs (*Database Administrators*), programming language experts, functional specialists, etc.

Skill Sets Needed From the Resources

Use different skill sets for human resources required for software development projects, such as:

1. **Programmers** – to develop the software programs needed for the project – experts in the chosen programming language.
2. **Graphic Designers** – to design the graphics and the web pages / front-end required for the project.
3. **Database Administrator** – to design the database and assist the programmers in optimizing data retrieval queries so that the response time is shorter.
4. **System Architects** – to develop the software architecture for the project.
5. **System Integrators** – to integrate various components of the project and ensure that the end product is built conforming to the specifications.
6. **Functional Experts** – who are experts in the application domain of the project?

Next, perform the following resource management activities:

1. Planning Activities

- Estimate requirement of resources
- Request for requisite resources
- Schedule resource utilization
- Resource Leveling

2. Utilization Activities

- Allocation of resources to various activities
- Ensure the activities are performed

3. De-allocation and Release Activities

- Performance appraisals for human resources
- Reconciliation for monetary resources – planned vs. actual utilization
- Document lessons learned in planning and utilization of resources including best and worst practices
- Release resources

Resource Leveling

Resource leveling is a technique in project management that overlooks resource allocation and resolves possible conflict arising from over-allocation. This will benefit the organization without having to face conflicts and not being able to deliver on time. Resource leveling is considered one of the key elements to resource management in the organization. An organization starts to face problems if resources are not allocated properly i.e., some resource may be over-allocated whilst others will be under-allocated. Both will bring about a financial risk to the organization.

Two Key Elements of Resource leveling: As the main aim of resource leveling is to allocate resource efficiently, so that the project can be completed in the given time period. Resource leveling can be broken down into two main areas:

- Projects that can be completed by using up all resources
- Projects that can be completed with limited resources Projects, which use limited resources can be extended for over a period of time until the resources required are available.

Structure of Resource leveling: Many organizations have a structured hierarchy of resource leveling. A work-based structure is as follows:

- Stage/Phase
- Task/Deliverable

All of the above-mentioned layers will determine the scope of the project and find ways to organize tasks across the team. In addition, depending on the three parameters above, the level of the resources required (seniority, experience, skills, etc.) may be different.

Establishing Dependencies: The main reason for a project manager to establish dependencies is to ensure that tasks get executed properly. By identifying correct dependencies from that of incorrect dependencies allows the project to be completed within the set timeframe. Here are some of the constraints that a project manager will come across during the project execution cycle. The constraints a project manager will face can be categorized into three categories.

- **Mandatory** - These constraints arise due to physical limitations such as human strength.
- **Discretionary** - These are constraints based on preferences or decisions taken by teams.

- **External** - Often based on needs or desires involving a third party.

Resource leveling Techniques

- **Fast tracking** - This performs critical path tasks. This buys time. The prominent feature of this technique is that although the work is completed for the moment, possibility of rework is higher.
- **Crashing** - This refers to assigning resources in addition to existing resources to get work done faster, associated with additional cost such as labor, equipment, etc.

Identifying resource requirements

The first step in producing a resource allocation plan is to list the resources that will be required along with the expected level of demand. This will normally be done by considering each activity in turn and identifying the resources required. It is likely, however, that there will also be resources required that are not activity specific but are part of the project's infrastructure (such as the project manager) or required to support other resources (office space, for example, might be required to house contract software developers).

For effectively identifying resource requirement for an organization, following points need to be considered.

- A program organization chart is essential to allocate staff effectively,
- Develop the hierarchical program organization.
- Identify Roles and Responsibilities.
- Plan for number of staff in each role (at a high level).
- Establish Teams.

For an activity identified:

- Work amount required (in work units)
- Basic skill or experience level required (to even under take the task)
- Complexity of the task (this will help to determine the experience required)
- Task Category (Unskilled, skilled, leadership, expert, management)

Example:

- Activity—Install Network Hardware for 20 computers.
- Work units-20.
- Basic Skill—Bachelor's Degree in related field.
- Task Complexity: 5.
- Task Category: Skilled (other categories may be Management, Leadership, Expert)

Resource scheduling refers to the set of actions and methodology used by organizations to efficiently assign the resources they have to jobs, tasks or projects they need to complete, and schedule start and end dates for each task or project based on resource availability. Depending on industry, resources can be people (either employees or independent contractors), equipment and machines (this is frequent for construction, manufacturing or maintenance businesses) or rooms and facilities. There may also be a need for consumable resources (for instance, materials and parts for manufacturing).

Resource scheduling is a key step of project management. When resource availability and work capacity are the primary factors that determine a project's deadline, project managers sometimes

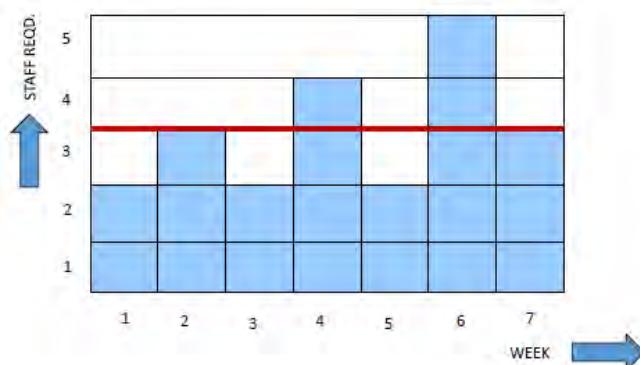
speak of **resource-constrained scheduling**. But resource scheduling is also often used for simple operation management, as it allows managers to outline completion dates for tasks assigned to their teams, which they can report to stakeholders such as customers or a board of directors.

- After all the required resources have been identified, they need to be scheduled effectively.
- The earliest start dates, last start dates will need to be taken into account to schedule resources efficiently.
- Resources should be balanced throughout the project.
- Human resource scheduling issues such as:
 - Planned Leave, Public Holidays.
 - Possible sick leave (random, subjective at best and hard to predict).
 - General motivation and enthusiasm for the task allocated (If they dislike the task, it will flow through into the output).
 - Work load and stress in project.
 - Stress outside work.

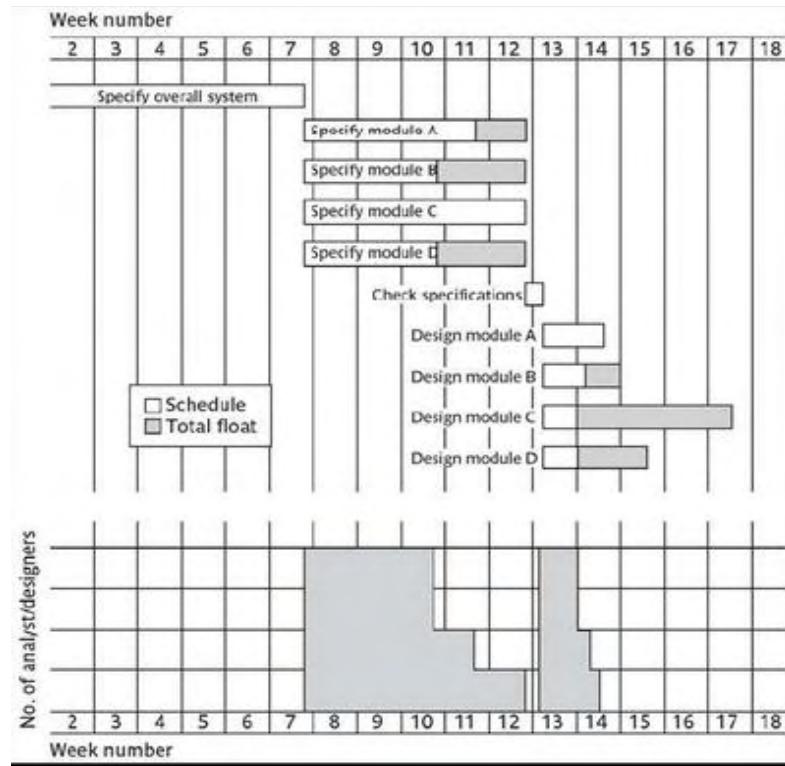
Resource histogram

The resource histogram is a tool that is often used by the project management team and or as a means of providing a visual representation to the team and to all of those interested parties. Specifically speaking, the resource histogram is specifically a bar chart that is used for the purposes of displaying the specific amounts of time that a particular resource is scheduled to be worked on over a predetermined and specific time period. Resource histograms may also contain the comparative feature of resource availability, used for comparison on for purposes of contrast. Resource histograms are indeed handy tools to utilize for the project management team and or the project management team leader because they allow a quick and easy single page view of exactly what resources are available, what resources are being utilized at the present time (or at whatever time the project management team and of project management team leader is seeking information on), and how long those resources are expected to be tied up. Resource histogram shows following details.

- People (by category) Vs. Week Number
- For each individual – estimated number of tasks (including complexity) over weeks
 - This helps in reducing work load sometimes to help the individual recover from any heavy load.
- Category Vs. Week



White rectangles indicate when an activity is scheduled and shaded rectangles the total float



Resource Clashes: During working process same resource may needed in more than one place at the same time, which needs to be managed by project manager. It can be resolved by:

- Delaying one of the activities
 - taking advantage of float to change start date
 - delaying start of one activity until finish of the other activity that resource is being used on - puts back project completion
- moving resource from a non-critical activity
- bringing in additional resource - increases costs

Prioritizing activities: Depending upon the condition we may need to prioritize activities or sometimes we need to know which activity should be completed first. Which is done in following ways

- *Total float priority* – those with the smallest float have the highest priority
 - Ordered according to their total float.
 - Smallest total float has highest priority.
 - Activities are allocated resources in ascending order of total float.
 - Changes to plan will require re-calculation.
- *Ordered list priority* – this takes account of the duration of the activity as well as the float–see next overhead
 - Activities that can proceed at the same time are ordered according to a set of simple criteria.
 - **Burman's priority list** takes into account activity duration as well as total float.
 - Gives priority to Shortest critical activities

- Gives priority to Other critical activities
- Gives priority to Shortest non-critical activities
- Gives priority to Non-critical activities with least float
- Gives priority to Non-critical activities

Note: Resource usage need to maximize to reduce idle periods between tasks. This balances costs against completion date and allows for contingency.

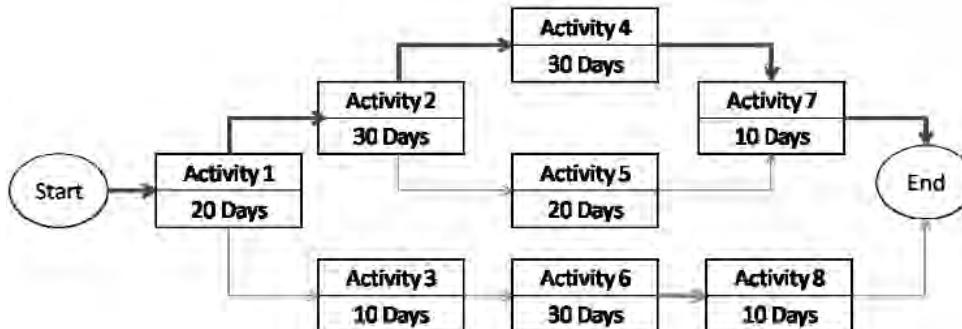
Creating Critical Paths

Scheduling resources can create new dependencies between activities. Therefore, Resource scheduling will almost always change the activity network. The changes often result in changes to the critical path. It is best not to add dependencies to the activity network to reflect resource constraints. If resources are scheduled, it makes network very messy.

A resource constraint may disappear during the project, but link remains on network. Thus you need to amend dates on **schedule** to reflect resource constraints. And delaying an activity due to lack of correct resources will cause that activity to become critical after it uses up all its slack time. These changes are often experienced after the project has started which will require adapting during the project (this is normally much harder in practice).

The Critical Path is the longest total duration. Activities on the critical path cannot be delayed or the whole project will be delayed, unless the loss of time can be offset somewhere else later on the critical path. To find the critical path add up the duration of the activities for each possible path through the network, to determine which has the longest total duration. The difference between the longest total duration and the shortest path(s) is the total amount of float or slack for the non-critical path activities.

The critical path is in essence the shortest time a project can be completed, even though the critical path is the longest path on the project. It is not the path with the most critical activities it is only concerned with the time dimension; it is not either the shortest path on a project network diagram. There are cases in which a network diagram may have two critical paths that have the same total duration.



To calculate the path using the diagram above use the duration of each activity and all the possible paths on the network.

Path A, Activities 1-2-4-7, total time = 90 days

Path B, Activities 1-2-5-7, total time = 80 days

Path C, Activities 1-3-6-8, total time = 70 days

The critical path is on Path A

Counting the Cost of Resources

The additional cost of employing extra staff or lengthening the overall duration of the project. The additional cost of employing extra staff would need to be compared to the cost of delayed delivery and the

increased the risk of not meeting the scheduled. All projects concentrate on completion in the shortest time span with minimum resources (in planning stage). However, once the project starts all un-planned issues and any risks will cause some strain on the cost.

Allocating individuals to activities (Being Specific)

Allocating resources and smoothing resources histogram is relatively straightforward where all resources of a given type can be considered more or less equivalent. In large building project among individual –there are likely to be many labors and they may be treated as equal so far as skills and productivity are concerned. The initial ‘resource types’ for a task have to be replaced by actual individuals. Factor need to be taken into account:

- **Availability:** We need to know if a particular individual will be available when required.
- **Criticality:** Allocation of more experienced personnel to activities on the critical path often helps in shortening project durations or at least reduces the risk of over run.
- **Risk:** Allocating the most experienced staff to the highest risk activities is likely to have the greatest effect in reducing overall project uncertainties
- **Training:** It will be benefit to the organization if positive steps are taken to allocate junior staff to appropriate non critical activities where there will be sufficient slack for them to train and develop skills.
- **Team building:** The selection of individuals must also take account of the final shape of the project team and the way they will work together.

Publishing the resource schedule: Work plans are commonly published either as lists or charts.

- **Get Schedule Approval:** This step also includes negotiations with project stakeholders, for example the time a certain activity is needed by the administrative function of the organization must be negotiated so that the resources and people are made available. Another negotiation include meeting with community leaders or beneficiary representatives to agree on specific milestones or dates when the project requires the participation of the beneficiaries. Once all people have agreed, at least in principle to the schedule, it is ready to be published. These negotiations may include in changes to the schedule, the skills of the project manager in making negotiations will determine the amount of change on the schedule. Stakeholders and management will often pressure the project team to either shorten or change the start dates of certain activities so that they do not interfere with their own schedules. The project team must learn to defend their estimates and learn to negotiate with these demands.
- **Schedule Baseline:** Once the team has completed the development of the project schedule they need to lock or set a baseline that will be used to monitor the schedule as the project makes progress. This baseline will be reviewed on a regular basis and all approved changes to the schedule will be updated against this baseline. This is the schedule that will be published to the team and stakeholders. Communicate the Schedule Once the project has “locked” the schedule with a baseline version it is ready to be shared with all the people that will be affected by it. This obviously includes the project team, and the best way to communicate the schedule is to draw a large version of the schedule and place it on a wall in a central location of the project office or room, this way the project schedule is visible to all the team and visitors to the project room. A copy of the schedule also needs to reach the organization and the administrative support functions, especially management. Stakeholders also receive a copy of the schedule, although it may include short versions that only highlight the major activities, copies can be distributed to the donor, key representatives of the community or beneficiaries, consultants, and partner organizations. The project schedule is the principal communication tool that graphically shows the progress of the project and it is used to identify activities that are not on track, by publishing a copy of the schedule

the project begins to educate the stakeholders about the project's complexity and critical dependencies that will impact the project and ensure that all people who are responsible for delivering outputs for the project are aware of their responsibilities along the project timeline.

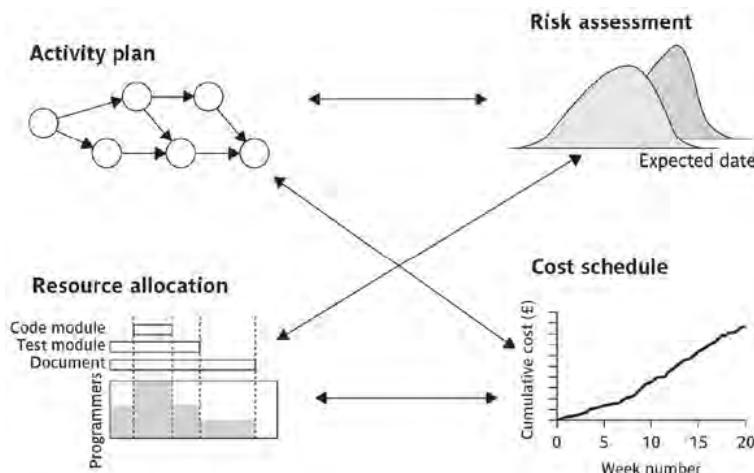
- **Schedule Updates:** As the project starts to make progress, the project manager will use the activity status reports from the project team to update the schedule and update the information of progress. This information comes from the Work Assignment Sheet or the Scope of Work report. Schedule progress reporting includes information such as the actual start and actual finish dates and any remaining durations for any unfinished scheduled activities. To facilitate the reporting of schedule progress a form can be used that maintains a consistent use during the life of the project. The project manager updates the schedule using two methods: by placing the percentage by which each activity has been completed during a reporting period, or by placing the number of days that have been worked on an activity. These reporting periods could be weekly, monthly or quarterly, all depending on the length and detail of the project schedule. The results of this updates with actual data, gives the project manager information on the general status of the project, and result in changes to the planned project end

Cost schedules: Cost schedules are used to produce a detailed cost showing weekly or monthly cost over the life of the project schedule. Calculation of cost is straightforward where organization has standard figure for staff & other resources. Cost are categorized as follows:

- **Staff cost:** includes not just staff salaries, but also social security contributions by the employer, pension scheme contribution, holiday pay etc. Timesheets are often used to record actual hours spent on each project by an individual. One issue can be how much time a staff member is allocated and available to the project, but is not actually working on the project, is dealt with.
- **Overheads:** Office Space, Interest charges, Travel Costs, Insurance and so on. Some overheads might be directly attributed to the project; in other cases a percentage of departmental overheads may be allocated to project costs.
- **Usage charges:** some charges can be on a pay as you go basis e.g. telephone charges, car mileage. At the planning stage an estimate of these may have to be made.

The scheduling sequence: successful project scheduling is not a simple sequence because of the inter-linking of different concerns project planning will need to be iterative. The consequences of decisions will need to carefully access and plans adjusted accordingly. It can be represented as sequence of steps.

- Successful resource allocation
- cost schedule
- Activity plans
- The interplay between the plans and schedule is complex



Topics Covered:

- Introduction
- Creating the frame work
- Collecting the data
- Visualizing process
- Cost monitoring
- Earned value
- Prioritizing monitoring
- Getting the project back to target
- Change control

The Monitoring and Controlling process oversees all the tasks and metrics necessary to ensure that the approved and authorized project is within scope, on time, and on budget so that the project proceeds with minimal risk. This process involves comparing actual performance with planned performance and taking corrective action to yield the desired outcome when significant differences exist. Monitoring and Controlling process is continuously performed throughout the life of the project. A project is monitored and controlled using the monitor and control project work process, which is a high- level process that is performed by executing more specific processes, such as cost control, schedule control and scope control. **Monitoring** includes: measuring the project performance, collecting and distributing information about the project performance and evaluating the performance information to see the trends.

Controlling: Taking action to either stay the course or change the wrong course. Continuous monitoring helps the project management team identify the areas that need to be controlled closely by, for example, taking corrective or preventive actions.

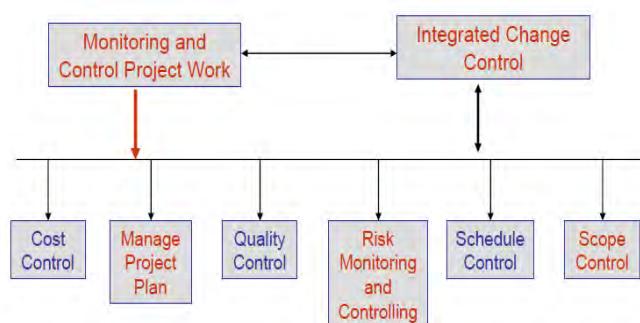
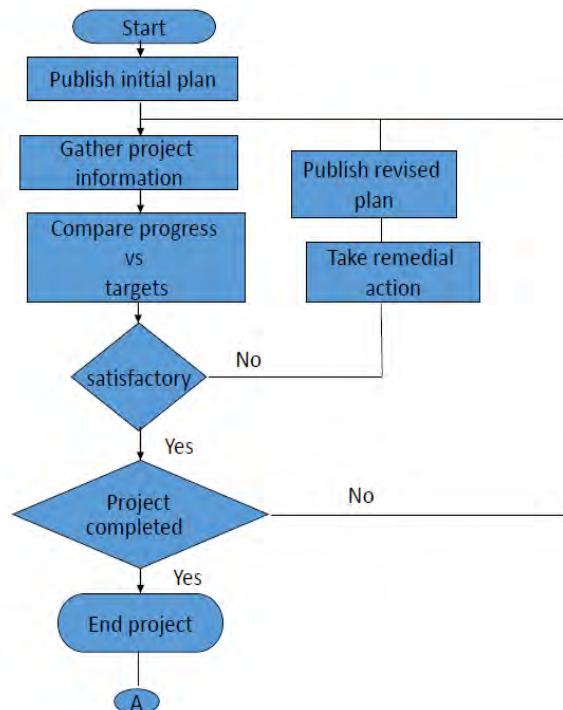


Figure : Some Processes in the Monitoring and Controlling Process Group



It also identifies any areas where changes to the project management plan are required and initiates the required changes. The Monitoring & Controlling process group contains eleven processes, which are:

1. **Monitor and control project work.** The generic process under which all other monitoring and controlling activities fall under.

2. **Perform integrated change control.** The tasks involved with making changes to the project plan. When changes to the schedule, cost, or any other area of the project management plan are necessary, the plan is changed and re-approved by the project sponsor.
3. **Validate scope.** The activities involved with gaining approval of the project's deliverables.
4. **Control scope.** Ensuring that the scope of the project does not change, and that unauthorized activities are not performed as part of the project (scope creep).
5. **Control schedule.** The tasks involved with ensuring the project work is performed according to the schedule, and that project deadlines are met.
6. **Control costs.** The tasks involved with ensuring the project costs stay within the approved budget.
7. **Control quality.** Ensuring that the quality of the project's deliverables is to the standard defined in the project management plan.
8. **Control communications.** Providing for the communication needs of each project stakeholder.
9. **Control risks.** Safeguarding the project from unexpected events that negatively impact the project's budget, schedule, stakeholder needs, or any other project success criteria.
10. **Control procurements.** Ensuring the project's subcontractors and vendors meet the project goals.
11. **Control stakeholder engagement.** The tasks involved with ensuring that all of the project's stakeholders are left satisfied with the project work.

Creating the frame work: It deals with exercising control over a project and ensuring that targets are met is a matter of regular monitoring, Finding out what is happening and comparing it with current targets. Again when the projects starts its execution, the project must be carefully monitored to ensure the project's progress. And also the expected outcomes are compared with the actual ones.

Project control is a continuous process of monitoring the progress of the project plan and is also includes re-planning of activities.

Revising the planning strategy is due to, Delay in completion of the project within the target time, Quality factors, inadequate functionality in adopting newer techniques and Actual estimation

- **Responsibility:** The overall responsibility for ensuring satisfactory progress on a project is often the role of the project steering committee or project board. Another aspect is reporting about the project, the reporting are classified as, **formal** and **informal**.
- Formal regular types can be oral or written. Standard oral communication of minutes are kept where as written type gets the reporting issues in a separate written format. Formal ad hoc are mostly received information of different levels towards the end of the project and generate written reports. Informal, oral and ad hoc provides early warning to the system and must be backed up by formal reporting procedures.
- **Assessing Progress:** The basis of information collected and collated at regular intervals or when specific events occur. Information will be objective and tangible. The information can however, measure the project's objectives in determining whether the project can produce deliverables or not. Single activity will not yield a deliverable work product but a group of activities can achieve the specified tangible product. The development of the project measures the progress assessment. It is carried out by the team members who are associated with the project activities.
- **Setting Check points:** They are regular, tied to specific events such as the production of a report and predetermined times when progress is checked. Check points may be
 - **Event driven:** check takes place when a particular event has been achieved
 - **Time driven:** date of the check is pre-determined
- **Taking snapshots:** Manager needs to receive information about progress will depend upon the size and degree of risk of the project. Progress reviews will generally take place at particular points during the life of a project it is known as review points or control points.

Collecting the data: It is the process of gathering information about partially completed activities. And it is difficult to make the forecasts accurately.

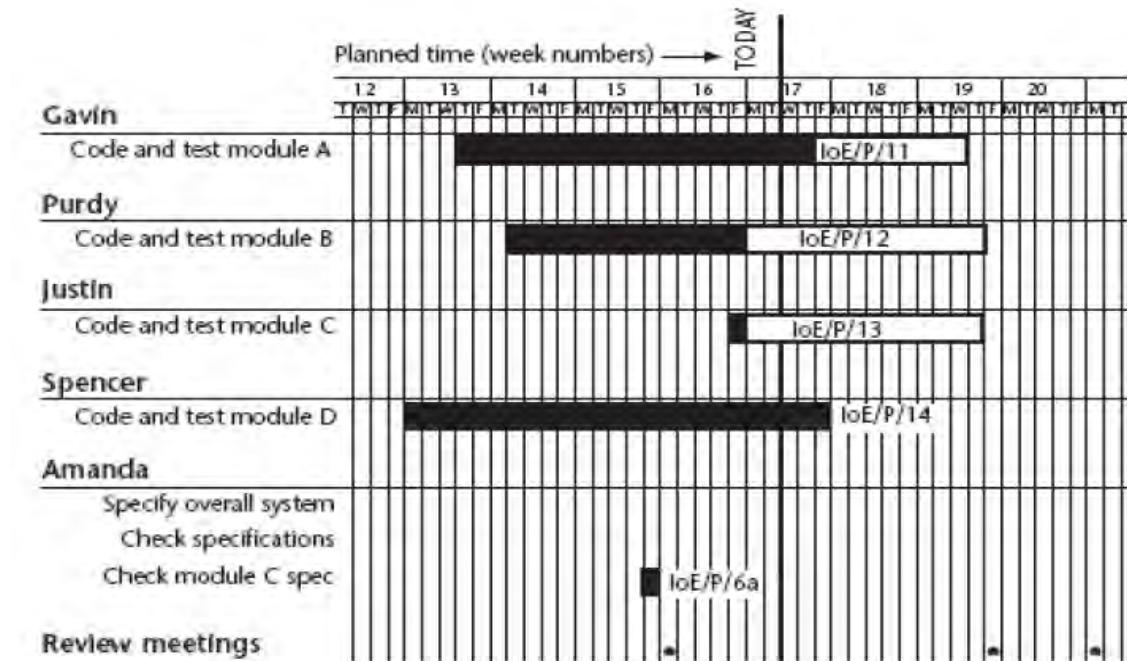
- **Partial completion reporting:** Organizations use standard accounting systems with weekly timesheets to charge staff time to individual jobs. Most project tends to about completion. There is no any proper measurement tool for measuring the progress of software projects and when one says how much work is completed, the usual answer is 99% this is called 99% completion syndrome. The possible solutions for this is
 - Control of products, not activities
 - Subdivide into lots of sub-activities
- **Risk reporting:** Reporting is to avoid asking for estimated completion dates. Thus use Traffic-light methods. Steps used in traffic light method are,
 - i. Identify the first level elements for assessment
 - ii. Break the first level elements in to second level elements
 - iii. Asses the second level elements and mark the colors.

Traffic-light methods,

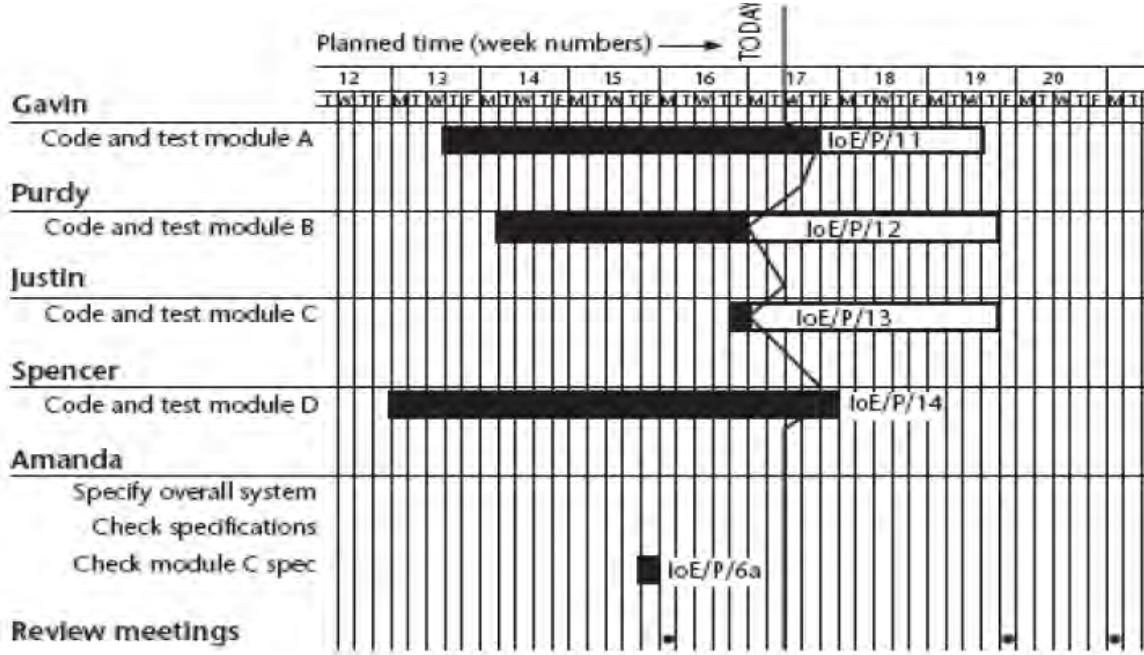
- Green – on target
- Amber – not on target but recoverable
- Red – not on target and difficult to recover
- Review all second level elements to reach the first level assessments.
- Review both first and the second level assessments to produce an overall assessments.
- Focus on non-achievement factors.
- Assessment forms can be used to evaluate the overall status of the project.
- Critical activities denoted by red color.

Visualizing progress: A manager needs some way of presenting that data to greatest effect. Some methods of presenting picture are,

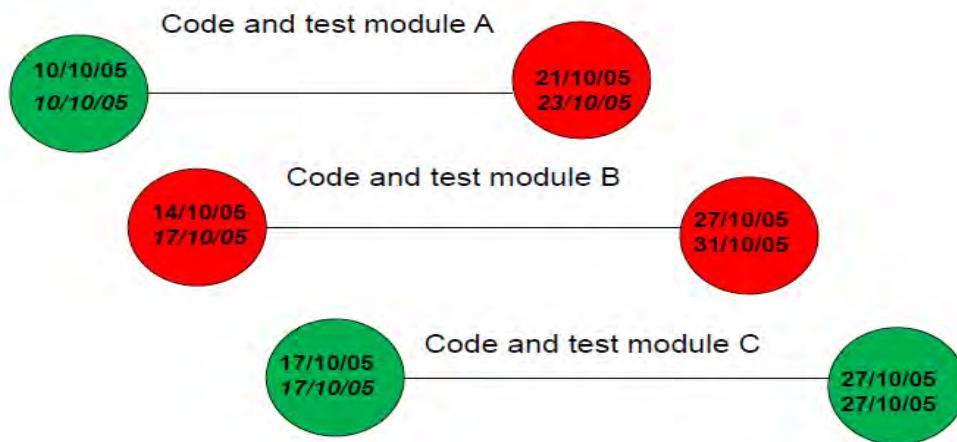
The Gantt chart: It is the simple and the oldest form of representing the progress of the project. It consists of activity bar that indicates the scheduled activity dates and the duration along with the activity floats.



Slip chart: Visual indication of activities that are not progressing to schedule. It is an alternative view of Gantt chart by providing a visual indication of those activities which are not on schedule. The more bend in the greater the variation in the project plan. If the slip line deviates more towards the non-achievement of project objectives then it has to be reconsidered. Additional slip lines can be included at regular intervals.



Ball charts: way of showing or not targets have been met or not. It is represented in the form of circles that indicate the start and the end point completion of activities. Circles of the ball chart mostly contain only two dates the original and the revised one. An activity is denoted by a red circle and green color denotes that the activity is ahead of its schedule. Slippage in the project completion date but it is overcome by the timeline charts.



Green: On time

Red: Missed the target

Cost Monitoring: A project could be late because the staff originally committed, have not been deployed. In this case the project will be *behind time* but *under budget*. A project could be *on time* but only because additional resources have been added and so by *over budget*. Thus we need to monitor both *achievements* and *costs* together.

Common Methods

- The 0/100 technique: where a task is assigned a value of zero until such time that is completed when it is given a value of 100% of the budgeted value
- The 50/50 technique: where a task is assigned a value of 50% of its value as soon as it is started and then given a value of 100% once its complete
- The milestone technique: where a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan.

Earned Value Analysis: Earned Value Analysis (EVA) is an industry standard method of measuring a project's progress at any given point in time, forecasting its completion date and final cost, and analyzing variances in the schedule and budget as the project proceeds. It compares the planned amount of work with what has actually been completed, to determine if the cost, schedule, and work accomplished are progressing in accordance with the plan. As work is completed, it is considered "earned".

EVA is a snapshot in time, which can be used as a management tool as an early warning system to detect deficient or endangered progress. It ensures a clear definition of work prior to beginning that work. It provides an objective measure of accomplishments, and an early and accurate picture of the contract status. It can be as simple as tracking an elemental cost estimate breakdown as a design progresses from concept through to 100% construction documents, or it can be calculated and tracked using a series of mathematical formulae (see below). In either case, it provides a basis for course correction. It answers two key questions:

1. At the end of the project, is it likely that the cost will be less than, equal to or greater than the original estimate?
2. Will the project likely be completed on time?

Work Breakdown Structure (WBS): EVA works most effectively when it is compartmentalized, i.e. when the project is broken down into an organized Work Breakdown Structure, or WBS. The WBS is used as the basic building block for the planning of the project. It is a product-oriented division of project tasks that ensures the entire Scope of Work is captured and allows for the integration of technical, schedule, and cost information. It breaks down all the work scope into appropriate elements for planning, budgeting, scheduling, cost accounting, work authorization, progress measuring, and management control. The two most common WBS systems are the Construction Specifications Institute (CSI) format, and the Uniform at. Often at the preliminary stages of design the Uniform at lends a better understanding of the cost centers, and at final bid level of documents often the CSI format is used. The indirect costs of design, oversight, and management must be included in the WBS to reflect the full budget.

Calculating Earned Value: Software packages such as Microsoft Project can perform earned value calculations automatically, and they're simple calculations that can quickly be performed manually as needed. Earned value calculations require the following:

- **Planned Value (PV)** = the budgeted amount through the current reporting period
- **Actual Cost (AC)** = actual costs to date
- **Earned Value (EV)** = total project budget multiplied by the % of project completion

With these readily available numbers, we're ready to do some calculations.

- **Schedule Performance Index (SPI)** calculation: SPI = EV/PV

SPI measures progress achieved against progress planned. An SPI value <1.0 indicates less work was completed than was planned. SPI >1.0 indicates more work was completed than was planned.

- **Cost Performance Index (CPI)** calculation: $CPI = EV/AC$

CPI measures the value of work completed against the actual cost. A CPI value <1.0 indicates costs were higher than budgeted. CPI >1.0 indicates costs were less than budgeted.

For both SPI and CPI, >1 is good, and <1 is bad. Note that if you're in a hurry, for both cost and schedule, you can subtract instead of dividing to get the variance. Schedule variance = $EV - PV$, and cost variance = $EV - AC$. Subtracting can quickly be done in your head, and for these cases, >0 is good, and <0 is bad. But unlike SPI and CPI, variance cannot be effectively compared across projects or over time, where the budget for a project may have changed, because they're relative to the size of the project.

- **Estimated at Completion (EAC)** calculation: $EAC = (\text{Total Project Budget})/CPI$

EAC is a forecast of how much the total project will cost.

An example: Assume we're halfway through a year-long project that has a total budget of \$100,000. The amount budgeted through this six-month mark is \$55,000. The actual cost through this six-month mark is \$45,000.

So, in summary:

Planned Value (PV) = \$55,000

- Actual Cost (AC) = \$45,000
- Earned Value (EV) = $(\$100,000 * 0.5) = \$50,000$
- Schedule Variance (SV) = $EV - PV = \$50,000 - \$55,000 = -\$5,000$ (bad because <0)
- Schedule Performance Index (SPI) = $EV/PV = \$50,000/\$55,000 = 0.91$ (bad because <1)
- Cost Variance (CV) = $EV - AC = \$50,000 - \$45,000 = \$5,000$ (good because >0)
- Cost Performance Index (CPI) = $EV/AC = \$50,000/\$45,000 = 1.11$ (good because >1)
- Estimated at Completion (EAC) = $(\text{Total Project Budget})/CPI = \$100,000/1.11 = \$90,000$

Because SV is negative and SPI is <1 , the project is considered behind schedule. We're 50% of the way through the project but have planned for 55% of the costs to be used. There will have to be some catch-up in the second half of the project.

Because CV is positive and CPI is >1 , the project is considered to be under budget. We're 50% of the way through the project, but our costs so far are only 45% of our budget. If the project continues at this pace, then the total cost of the project (EAC) will be only \$90,000, as opposed to our original budget of \$100,000.

Prioritizing Monitoring: So far we have assumed that all aspects of a project will receive equal treatment in terms of the degree of monitoring applied. We must not forget, however, that monitoring takes time and uses resources that might sometimes be put to better use. We might focus more on monitoring certain types of activity:

- **Critical path activities:** Any delay in an activity on the critical path will cause a delay in the completion date for the project. Critical path activities are therefore likely to have a very high priority for close monitoring.
- **Activities with no free float:** A delay in any activity with no free float will delay at least some subsequent activities even though, if the delay is less than the total float, it might not delay the project completion date. These subsequent delays can have serious effects on our resource schedule as a delay.
- **Activities with less than a specified float:** If any activity has very little float it might use up this float before the regular activity monitoring brings the problem to the project manager's attention. It is common practice to monitor closely those activities with less than, say, one week free float.
- **High risk activities:** A set of high risk activities should have been identified as part of the initial risk profiling exercise. If we are using the PERT three-estimate approach we will designate as high risk those activities that have a high estimated duration variance. These activities will be given close attention because they are most likely to overrun or overspend.

- **Activities using critical resources:** Activities can be critical because they are very expensive (as in the case of specialized contract programmers). Staff or other resources might be available only for a limited period, especially if they are controlled outside the project team. In any event, an activity that demands a critical resource requires a high level of monitoring.

Getting back on track

- Renegotiate the deadline—if not possible then
- Try to shorten activities on critical path e.g.
 - Work overtime
 - Re-allocate staff from less pressing work
 - Buy in more staff
- Reconsider activity dependencies
 - Over-lap the activities so that the start of one activity does not have to wait for completion of another
 - Split activities

Change Control: Change Control is the process that a company uses to **document, identify and authorize changes** to an IT environment. It reduces the chances of unauthorized alterations, disruption and errors in the system.

Whenever any new or different changes are requested for the system, especially by stakeholders, it is neither optional nor ignorable. It has to be implemented without affecting other components of the system. This is when the change control comes handy. It helps project teams to modify the scope of the project using specified controls and policies. Change Control is practiced whenever a project is not progressing as planned.

It is mandatory that a formal document for change request is completed and reviewed in order to keep control of change requests.

Number of question one might encounter while analyzing Change Control like

- Who will approve the change?
- Does it require to run through a change control board?
- How much time will be required to research and implement the change?
- What are the impacts of changes to other components of the system (schedules, cost, resources, etc.)?
- Is there any threshold under which the project management can approve it?

Different factors of Change Control process: There are various factors that a Change Control process should consider

Steps in Change Control Process

- Change request initiation and Control
- Impact Assessment
- Control and Documentation of Changes

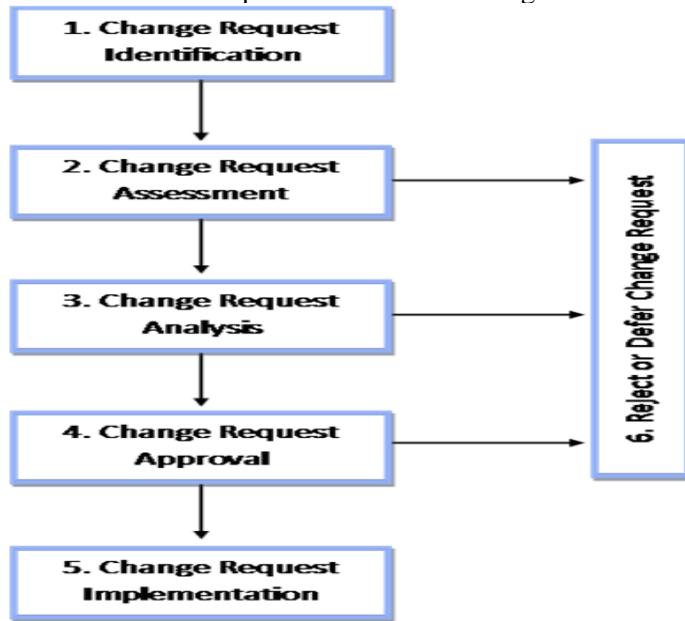
Action taken in Change Control

- Request for changes should be standardized and subject to management review
- Change requestor should be kept informed
- Make sure that all requests for change are assessed in a structured way for analyzing possible impacts
- A change log should be maintained that tells the date, person details who made changes and changes implemented
- Only authorized individual should be able to make changes
- A process for rolling back to the previous version should be identified

- Documentation and Procedures
- Authorized Maintenance
- Testing and User signoff
- Version Control
- Emergency Changes
- Whenever system changes are implemented the procedures and associated document should update accordingly
- System access right should be controlled to avert unauthorized access
- Software should be thoroughly tested
- Control should be placed on production source code to make sure that only the latest version is updated
- A verbal authorization should be obtained, and the change should be documented as soon as possible

Change Process Flow-Diagram

Change Process follows a specific pattern to implement the changes in the product or system. Here through flow-diagram we explained what are the steps involved in the Change Process.



Steps for Change Control

Steps for Change Control

- Change request identification
- Change request assessment
- Change request analysis

Action

- Identify the need for a change and describe it on the project change request form
- If the change is not valid, it has to be deferred or rejected
- Determine appropriate resources required to analyze the change request
- Perform quick assessment of the potential impact and update the change request form
- At this stage, rejected change request should stopped
- For analysis assign the change request to an authorized member
- Deferred change re-enter this analysis step
- At this stage, rejected change request should stopped

- Change request approval
 - Identify change risk and complexity level before approval
 - Identify the impact level of the change before approval
 - Review impact of Change Request to authorized person for approval
 - At this stage, rejected change request should stopped
 - Update project procedure and management plans
 - Inform about the changes to the team
 - Monitor progress of change request
 - Record the completion of change request
 - Close change request
- Change request implementation

NOTE: The approval for Change Control may be done by **Project Manager, Lead IT or Lead Developer, Stakeholder.**

Topics Covered:

- Introduction
- Type of contract
- Stages in contract
- Placement
- Typical terms of contract
- Contract management
- Acceptance
- Managing people and organizational terms
 - Introduction
 - Understanding behavior
- Organizational behavior
- A back ground
- Selecting the right person for the job
- Instruction in the best methods
- Motivation
- Working in groups
- Becoming a team
- Decision making
- Leadership
- Organizational structure
- Conclusion
- Further exercises

In the world of business, contracts are used for establishing business deals and partnerships. The parties involved in the business engagement decide the type of the contract. Usually, the type of the contract used for the business engagement varies depending on the type of the work and the nature of the industry. The contract is simply an elaborated agreement between two or more parties. One or more parties may provide products or services in return to something provided by other parties (client). The contract type is the key relationship between the parties engaged in the business and the contract type determines the project risk. Let's have a look at most widely used contract types.

1. **Fixed Price (Lump Sum):** This is the simplest type of all contracts. The terms are quite straightforward and easy to understand. To put it simple, the service provider agrees to provide a defined service for a specific period of time and the client agrees to pay a fixed amount of money for the service.

This contract type may define various milestones for the deliveries as well as KPIs (Key Performance Indicators). In addition, the contractor may have an acceptance criteria defined for the milestones and the final delivery. The main advantages of this type of contract is that the contractor knows the total project cost before the project commences.

2. **Unit Price:** In this model, the project is divided into units and the charge for each unit is defined. This contract type can be introduced as one of the more flexible methods compared to fixed price contract. Usually, the owner (contractor/client) of the project decides on the estimates and asks the bidders to bid of each element of the project. After bidding, depending on the bid amounts and the qualifications of bidders, the entire project may be given to the same service provider or different units may be allocated to different service providers. This is a good approach when different project units require different expertise to complete.
3. **Cost Plus:** In this contract model, the services provider is reimbursed for their machinery, labor and other costs, in addition to contractor paying an agreed fee to the service provider. In this method, the service provider should offer a detailed schedule and the resource allocation for the project. Apart from that, all the costs should be properly listed and should be reported to the contractor periodically. The payments may be paid by the contractor at a certain frequency (such as monthly, quarterly) or by the end of milestones.
4. **Incentive:** Incentive contracts are usually used when there is some level of uncertainty in the project cost. Although there are nearly-accurate estimations, the technological challenges may impact on the overall resources as well as the effort. This type of contract is common for the projects involving pilot programs or the project that harness new technologies. There are three cost factors in an Incentive contract; target price, target profit and the maximum cost. The main mechanism of Incentive contract is to divide any target price overrun between the client and the service provider in order to minimize the business risks for both parties.
5. **Retainer (Time and Material - T&M):** This is one of the most beautiful engagements that can get into by two or more parties. This engagement type is the most risk-free type where the time and material used for the project are priced. The contractor only requires knowing the time and material for the project in order to make the payments. This type of contract has short delivery cycles, and for each cycle, separate estimates are sent of the contractor. Once the contractor signs off the estimate and Statement of Work (SOW), the service provider can start work. Unlike most of the other contract types, retainer contracts are mostly used for long-term business engagements.
6. **Percentage of Construction Fee:** This type of contracts is used for engineering projects. Based on the resources and material required, the cost for the construction is estimated. Then, the client contracts a service provider and pays a percentage of the cost of the project as the fee for the service provider. As an example, take the scenario of constructing a house. Assume that the estimate comes up to \$230,000. When this project is contracted to a service provider, the client may agree to pay 30% of the total cost as the construction fee which comes up to \$69,000.

Tender: To tender is to invite bids for a project or accept a formal offer such as a takeover bid. Tendering usually refers to the process whereby governments and financial institutions invite bids for large projects that must be submitted within a finite deadline. The term also refers to the process whereby shareholders submit their shares or securities in response to a takeover offer.

The Tendering Process

Open tendering: Open tendering is the preferred competitive public procurement method used for acquiring goods, services and infrastructure works. It is executed in accordance with established procedures set out in the procurement guidelines and detailed in the standard bidding documents. Government bodies may have to do this by local / international law

Open tendering is also known as open competitive bidding, open competition or open solicitation, and the procurement notices used to call for bids for these requirements are identified as: Invitation for Bids or Invitation to Tender.

The fundamental requirements of open tendering are that they should:

- Be open to all qualified and interested bidders,
- Be advertised locally (and internationally, when required),
- Have objective qualifications criteria,
- Have neutral and clear technical specifications,
- Have clear and objective evaluation criteria, and
- Be awarded to the least-cost provider, without contract negotiations.

Restricted Tendering: Restricted tendering is a procurement method that limits the request for tenders to a select number of suppliers, contractors or service providers. This method of procurement is also called: Limited Bidding and Selective Tendering. Although considered a competitive procurement method, competition is limited to only firms shortlisted or invited by the procuring entity.

A process should be in place for arriving at the number and specific firms that will be invited; that number however is dependent on the stipulations of the public procurement legal framework. Any decision to use the Restricted Tendering procurement method must conform to the policies and procedures governing the procurement system. A basic characteristic of this method is that competition is confined to a certain number of firms either because only a few firms are qualified to fulfill the specific type of requirement, or certain conditions warrant the use of a limited number of firms in order to reduce the time and cost of the selection process.

Negotiated tendering: Negotiated tendering occurs when the client approaches a single supplier based on their track-record or a previous relationship and the terms of the contract are then negotiated. Negotiating with a single supplier may be appropriate for highly specialist contracts (where there may be a limited number of potential suppliers), or for extending the scope of an existing contract. It can give the client the confidence of working with a supplier they already know, can reduce the duration and costs of tendering and can allow early supplier involvement.

Stages in contract placement

Requirements analysis: Before potential supplier can be approached, you need to have a clear set of requirements. Two points need to be emphasized here. The first is that it is easy for this step to be skimped where the user has many day-to-day pressures and not much time to think about future developments. In this situation, it can be useful to bring in an external consultant to draw up a requirements document. Even here, users and their managers need to look carefully at the resulting requirements document to ensure that it accurately reflect their needs. As David Bainbridge has pointed out: 'the lock of, or defects in, the specification are probably the heart of most disputes resulting from the acquisition of computer equipment and software' the requirements document might typically have sections with the headings. This requirements document is sometimes called an operational requirement or OR.

Main sections in a requirements document

Section name

- 1 Introduction
- 2 A description of any existing systems and the current environment
- 3 The customer's future strategy or plans
- 4 System requirements
- 5 Deadlines
- 6 Additional information required from potential suppliers

These will include

- functions in software, with necessary inputs and outputs
- standards to be adhered to
- other applications with which software is to be compatible

- quality requirements e.g. response times

Evaluation Plan: Evaluation plan tries to answer, how are proposals to be evaluated? The methods for evaluation plan could include

- reading proposals
- interviews
- demonstrations
- site visits
- practical tests

Evaluation plan needs to assess value for money (VFM) for each desirable feature. VFM approach an improvement on previous emphasis on accepting lowest bid

- Example:
 - feeder file saves data input
 - 4 hours' work a month saved at £20 an hour
 - system to be used for 4 years
 - if cost of feature £1000, would it be worth it?

Invitation to tender: Bidder makes an *offer* in response to Invitation to tender. The *acceptance* of offer creates a *contract*. The customer may need further information about tender like Problem of different technical solutions to the same problem etc.

Memoranda of agreement (MoA)

- Customer asks for technical proposals
- Technical proposals are examined and discussed
- Agreed technical solution in MoA
- Tenders are then requested from suppliers based in MoA
- Tenders judged on price
- Fee could be paid for technical proposals by customer

Contract management

Contract management is the process of managing contract creation, execution and analysis to maximize operational and financial performance at an organization, all while reducing financial risk. Organizations encounter an ever-increasing amount of pressure to reduce costs and improve company performance. Contract management proves to be a very time-consuming element of business, which facilitates the need for effective and automated contract management system. The contract management should address following

- usability of an existing package
- usability of an application yet to be built
- maintenance costs of hardware
- time taken to respond to requests for software support
- training

Contracts should include agreement about how customer/supplier relationship is to be managed e.g.

- *decision points* - could be linked to payment
- *quality reviews*
- *changes to requirements*

The communication between the supplier and the customer while the work contracted for is being carried out. Certain decision points, the customer needs to examine work already done and make decisions about the future direction of the project. The project will require representative of the supplier and customer to interact at many points in the development cycle.

When the contract is being negotiated, certain key points in the project can be identified where customer approval is needed before the project can proceed. For each decision point the deliverable to be presented by the suppliers the decisions to be made by the customer and the outputs from the decision point all need to be defined. Meet one or more legal obligations

Contract Acceptance: When work is completed, customer needs to carry out acceptance testing. Contract may put a time limit to acceptance testing—customer must perform testing before time expired. Part or all payment to the supplier should depend on acceptance testing

Managing people and Organizing Teams

It is the most difficult areas in managing software development projects “Most managers are willing to concede the idea that they’ve got more people worries than technical worries. But they seldom manage that way. One reason: technical experts become managers.

Important areas:

- Selecting right people for the job
- Working as a team
- Select and Induct new staff into a project
- Increase staff motivation
- Improve group working
- Evaluate the characteristics of the various team structures
- Use the most appropriate leadership styles.

Some objectives can be the health and safety during the project life cycle. The project leaders might have little control over organizational structure. The scope and nature of objectives can be set in a way that will enhance staff motivation and many risks to project success relate to staffing.

Understanding Behavior

Behaviors associated with complex and challenging mental health, dementia or other neurological conditions include aggression, wandering, agitation. These apparent changes in the personality of the person with the disease are a major source of distress both to the person who is presenting the behaviors and to those who experience them – the caregiver, the family members, and the service providers in all sectors of the health-care system.

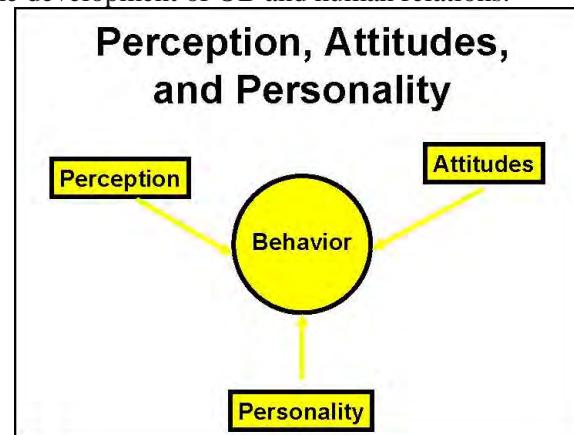
People differ from each other in their needs and values. Group effort eases their task of achieving organizational goals effectively. Human relations can be defined as motivating people in organizations to work as a team. Although human relationships have existed from quite some time in the past, the study of human relations has developed only recently. Social sciences like sociology, psychology, anthropology, economics and political science have contributed to the development of OB and human relations.

Organizational behavior

A field of study that investigates the impact that individuals, groups and structure have on behavior within organizations, for the purpose of applying such knowledge toward improving an organization’s effectiveness.

Taylor attempted to analyze the most productive way of doing manual tasks. Taylor had 3 basic objectives:

- to select the best people for the job
- to instruct them in the best methods



- to give incentives in the form of higher wages to the best workers

Taylor's view emphasize on the financial basis of staff motivation, however, the other issues of motivation should be encouraged staff not just on such rewards. Theory X and Theory Y by Donald McGregor draws attention to the way that expectations influence behavior

Theory X holds that:

- The average human has an innate dislike of work
- There is a need therefore for coercion, direction and control
- People tend to avoid responsibility

Theory Y holds that:

- Work is as natural as rest or play
- External control and coercion are not the only ways of bringing about effort directed towards and organization's ends
- Commitment to objectives is a function of the rewards associated with their achievement
- The average human can learn to accept and further seek responsibility
- The capacity to exercise imagination and other creative qualities is widely distributed
- One way of judging whether a manager espouses Theory X or Theory Y is to observe how staff react when the boss is absent:
- If there is no discernible change then this is a Theory Y environment;
- If everyone visibly relaxes, it is a Theory X environment

Therefore,

A "reward" does not have to be a financial reward- it could be something like a sense of achievement. Theory X and Theory Y illustrated how the state of mind of workers influenced their productivity

Select the right person for the job

Taylor stressed "the need for the right person for the job". Also called as **recruitment** process.

Examples of question;

- What sort of characteristics should they be looking for?
- Is an experienced programmer better than a new graduate with a first class mathematics degree?
- Recruitment is often an organizational responsibility

There are 2 types of candidates that are distinguished by Meredith Belbin:

- **Eligible candidates** have curriculum vitae (CV) which shows, for example, the 'right' number of years in some previous post and the 'right' paper qualifications.
- **Suitable candidates** can actually do the job well.

A mistake is to select an eligible candidate who is not in fact suitable. Thus, Belbin suggests we should try to assess actual skills rather than past experience and provide training to make good minor gaps in expertise. And it also has the general approach for recruitment process

Sources of Recruitment

- Internal Sources
 - Personnel already in the payroll of an organization
 - Present work force
 - Promotion or post upgrade

- External Sources
 - Recruitment advertising
 - Recruitment from colleges
 - Casual caller
 - Use of special consultant
 - Employment exchanges and agencies

Motivation: It is an inner state of our mind that activates and directs our behavior. It makes us move to act. One's willingness to exert efforts towards the accomplishment of his/her goal. Besides this there are different **motivating factors** include achievement, advancement, autonomy, personal growth, recognition, responsibility, and the work itself. There are different models proposed for motivation

The Taylorist model: Taylor's viewpoint is reflected in the use of piece-rates in manufacturing industries and sales bonuses amongst sales forces.

- Piece-rates can cause difficulties if a new system will change work practices.
- If new technology improves productivity, adjusting piece-rates to reflect this will be a sensitive issue.
- "Piece-rates" are where workers are paid a fixed sum for each item they produce.
- "Day-rates" refer to payment for time worked
- Rewards based on piece-rates need to relate directly to work produced

So, this model emphasizes on the reward system

Maslow's hierarchy of needs: The motivation of individual varies. Money is a strong motivator when you are broke. However, as the basic need for cash is satisfied, other motivations are likely emerge. In practice, people are likely to be motivated by different things at different stages of their life.

Herzberg's two-factor theory: Some things about a job can make you dissatisfied. If the causes of this dissatisfaction are removed, this does not necessarily make the job more exciting. There are two sets of factors about a job:

- **Hygiene or maintenance factors**, which can make you dissatisfied if they are not right, for example the level of pay or the working conditions;
- **Motivators**, which make you feel that the job is worthwhile, like a sense of achievement or the challenge of the work itself

A model of motivation developed by Vroom and his colleagues. It identifies three influences on motivation:

- expectancy: the belief that working harder will lead to a better performance
- instrumentality: the belief that better performance will be rewarded
- perceived value: of the resulting reward

Motivation will be high when all three factors are high. A zero level for any one of the factors can remove motivation.

Working in Groups

A working group or working party is a group of experts working together to achieve specified goals. The groups are domain-specific and focus on discussion or activity around a specific subject area. The term can sometimes refer to an interdisciplinary collaboration of researchers working on new activities that would be difficult to sustain under traditional funding mechanisms. The development schedule for most non-trivial

software projects is such that they cannot be completed by one person working alone, it is a group activity. Group interaction is a key determinant of group performance. Flexibility in group composition is limited and Managers must do the best they can with available people. Examples of common goals for working groups include:

- creation of an informational document
- creation of a standard
- resolution of problems related to a system or network
- continuous improvement
- research

Group composition: One factor that determines the efficiency of collaborative learning is the composition of the group. This factor is defined by several variables: the age and levels of participants, the size of the group, the difference between group members, etc.

Regarding the number of members, small groups seem to function better than large groups in which some members tend to be 'asleep' or excluded from interesting interactions. Most of the mechanisms described in the previous section, e.g. mutual regulation, social grounding, shared cognitive load, ..., can only occur between a few participants. This does not argue in disfavor of large group sessions. It simply means that distance learning activities should also include 'closed' sessions, in which a restricted number of subjects collaborate and/or 'monitored' session in which the teacher takes care that no learner is left out the interaction.

Regarding the participants, some developmental level is necessary to be able to collaborate, but this is only an issue for children and does hence not directly concern current distance education activities which mainly concern adult learners. Group composed of members who share the same motivation can be problematic.

- Task-oriented-everyone wants to do their own thing;
- Self-oriented-everyone wants to be the boss;
- Interaction-oriented-too much chatting, not enough work.

An effective group has a balance of all types. This can be difficult to achieve software engineers are often task-oriented. And interaction-oriented people are very important as they can detect and defuse tensions that arise.

Group Leadership: Man, by nature is a member of one group or the other. The group can be a family, club, society, school, college..... The instinct to move around in groups is there in animals also. Elephants, lions, leopards, sheep – all in general move around in groups.

A leader as the word stands is someone who leads others. He should have a vision, commitment and drive to achieve the goal of the group. In other words, team leadership is the management of a group of people to achieve the desired result. The leader must motivate and inspire confidence in the members of the group. The leader must be flexible, adaptable and should be capable to face all types of situations including setbacks, challenges and failures in a calm and cool manner. He must be able to extract work from each member of the group after assigning a particular task according to the individual's capacity as it may differ from person to person.

Characteristics of a Group Leader

Some are born leaders; others are groomed as leaders. The leadership qualities can be developed and improved through proper training and practice. The important qualities of a leader are:

- **Effective communication:** The leader should have the capacity to communicate effectively to other team members. The skill should be both in speaking and writing modes.
- **Motivation:** The leader should be able to inculcate a sense of motivation in the members. Only then can the full capacity of each member of the group be extracted. To create personal rapport, he should address each member by their names.

- **Proper Planning:** At the start of the work, a detailed study should be undertaken to assess the various implications and systematic planning should be done. For this, a suitable software can be used and assistance of experts in different domains sought.
- **Energetic:** The leader must be energetic and physically and mentally fit for the task. His energy level should always be high.
- **Experience and Knowledge:** The leader should have theoretical knowledge in the particular field/domain and also enough practical experience.
- **Self Confidence:** This is a very important characteristic of a good leader. Only a confident leader will be able to generate confidence in others. For completing a task successfully, each member of the group should work confidently.
- **Assertive:** The leader should assert himself and give necessary instructions to his team members.
- **Domination:** This is another face of assertiveness.
- **Honesty:** The old proverb ‘Honesty is the best policy’ is perfectly apt in this case also. There is no need to overemphasize the importance of honesty for a leader.
- **Charismatic:** This is the ability to influence the thinking and attitude of others.
- **Integrity:** Should be beyond doubt.

What do Leaders Do?

- Enable people and groups to achieve their objectives
- Set and communicate objectives
- Monitor performance and give feedback
- Establish basic values
- Clarify and solve problems for others
- Organise resources
- Longer term?
- Administer rewards and punishments
- Provide information, advice and expertise
- Provide social and emotional support
- Make decisions on behalf of others
- Represent the group to others
- Arbitrate in disputes
- Act as a father figure
- Become a scapegoat

Group cohesion: Group cohesion is a social process that characterizes groups whose members interact with each other and refers to the forces that push group members closer together. A lot of work these days is accomplished in groups. Most people have had both good and bad experiences from participating in such group work. One important element that influences one's group work experience is cohesion. Cohesion has two dimensions: emotional (or personal) and task-related. The advantages of a cohesive group are:

- Group quality standards can be developed;
- Group members work closely together so inhibitions caused by ignorance are reduced;
- Team members learn from each other and get to know each other's work;
- Egoless programming where members strive to improve each other's programs can be practiced.

Group loyalty: Group loyalty is wholly natural since human beings are group made as well as self made. They are inherently social, and behind the most anti-social actions there may be group influences. It is chiefly in the play-day of childhood that group loyalties are stimulated in human beings. In associating with parents, and particularly with other children, the child experiences the growth of his social nature, or of

social personality. Through such association, the spirit of appreciation is developed. The individual learns that others have feelings, desires, problems, sufferings which are similar to his own. In consequence, social attitudes are formulated, group loyalties arise, and socially harmonious actions follow.

- Group members tend to be loyal to cohesive groups.
- 'Groupthink' is preservation of group irrespective of technical or organizational considerations.
- Management should act positively to avoid group think by forcing external involvement with each group.

Group Communication: **Group communication** is a mode of communication in an organization, between employers and employees, and employees in teams/groups. Group communication can further be looked from marketing perspective as communicating to a group of people or target customers in order to market a product.

Small group communications can be looked at as groups of size 3 to 20. And larger groups looked can be of sizes 100 to 200 in size. Group communication can be effective, when there is a specific purpose to the communication, a proper means of communication, and content of the communication suitable to the target audience, and a proper communicator who can drive the initiative and process.

Group communication can be of various means, like social media, digital media, print media, speeches and focus group communication etc. The choice of the means of communication depends on factors like the target audience, the means and availability of the communication mode, cost implication for the communication initiative. Group communication can have effective results in case of marketing, where the communication is vital for selling and marketing products and product launches etc. following factors influence group communication:

- **Group size:** The larger the group, the harder it is for people to communicate with other group members.
- **Group structure:** Communication is better in informally structured groups than in hierarchically structured groups.
- **Group composition:** Communication is better when there are different personality types in a group and when groups are mixed rather than single sex.
- **The physical work environment:** Good work place organization can help encourage communications.

Group organization: Small software project management groups are usually organized informally without a rigid structure. For large projects, there may be a hierarchical structure where different groups are responsible for different sub-projects.

In an organization, the formation of groups is very natural, whether it is created by the management for the purpose of accomplishing the goals of the organization or by the members of the organizations themselves to fulfill their social needs. There are two types of group, namely, formal groups and informal groups.

Formal groups are the ones that are created as per official authority, so as to fulfill the desired objective. Unlike, **informal groups** are formed by the employees as per their likes, interests, and attitudes.

The most common reason behind the creation of a group is the urge of people to talk and make their own circle, where they can interact freely, know each other, work unitedly and accomplish the tasks which are being assigned to them. In the given article, the difference between formal and informal groups are presented.

BASIS FOR COMPARISON	FORMAL GROUP	INFORMAL GROUP
Meaning	Groups created by the organization, for the purpose of accomplishing a specific task are known as Formal Groups.	Groups created by the employees themselves, for their own sake are known as Informal Groups.
Formation	Deliberately	Voluntarily
Size	Large	Comparatively small.
Life	Depends on the type of group.	Depends on the members.
Structure	Well Defined	Ill Defined
Importance is given to	Position	Person
Relationship	Professional	Personal
Communication	Moves in a defined direction.	Stretches in all the directions.

Extreme programming groups: Extreme programming groups are variants of an informal, democratic organization. In extreme programming groups, some ‘management’ decisions are devolved to group members. Programmers work in pairs and take a collective responsibility for code that is developed.

Chief programmer teams: Consist of a kernel of specialists helped by others added to the project as required. The motivation behind their development is the wide difference in ability in different programmers. Chief programmer teams provide a supporting environment for very able programmers to be responsible for most of the system development

Becoming a Team:

- **Group:** A set of people who are assigned to a common task and who work individually to accomplish their assignment.
- **Team:** A small group of people working on the same problem or sub-problem in a project. The team members depend on one another to do their tasks.
 - **Project Team:** Based on the premise that every member can and must make a valuable contribution to the project.
- **Committee:** Comprised of people who come together to review and critique issues, propose recommendations for action

Team Building: Team building is the process of taking a collection of individuals with different needs, backgrounds and expertise and transforming them by various methods into an integrated, effective work unit. In this transformation process, the goals and energies of individual contributors’ merge and support the objectives of the team. The concept of team building becomes critically important as bureaucratic hierarchies decline and horizontally-oriented teams and work units become increasingly important. In most cases, team building involves relationships among peers with a wide diversity of expertise.

Team work involves working together to achieve something beyond the capabilities of individuals working alone. Much of the work in organizations is completed through *teamwork*. Effective team members behave differently than individual members. Understanding *group process* can improve your team behavior and performance. The foundation of team performance is *interpersonal skills*.

Organizations expect teams to achieve higher levels of performance in less time with fewer resources and Firms need to select team leaders and members who have the balance of *technical* and *interpersonal* skills.

Decision making: Decision making is the process of making choices by identifying a decision, gathering information, and assessing alternative resolutions. Using a step-by-step decision-making process can help you make more deliberate, thoughtful decisions by organizing relevant information and defining alternatives. This approach increases the chances that you will choose the most satisfying alternative possible.

- **Programmed decision** involves simple, common problems with predetermined solutions.
- **Non-programmed decision** involves a complex, unique problem or opportunity with important consequences for the organization

Steps in decision making

1 Identify the decision: You realize that you need to make a decision. Try to clearly define the nature of the decision you must make. This first step is very important.

2 Gather relevant information: Collect some pertinent information before you make your decision: what information is needed, the best sources of information, and how to get it. This step involves both internal and external “work.” Some information is internal: you’ll seek it through a process of self-assessment. Other information is external: you’ll find it online, in books, from other people, and from other sources.

3 Identify the alternatives: As you collect information, you will probably identify several possible paths of action, or alternatives. You can also use your imagination and additional information to construct new alternatives. In this step, you will list all possible and desirable alternatives.

4 Weigh the evidence: Draw on your information and emotions to imagine what it would be like if you carried out each of the alternatives to the end. Evaluate whether the need identified in Step 1 would be met or resolved through use of each alternative. As you go through this difficult internal process, you’ll begin to favor certain alternatives: those that seem to have a higher potential for reaching your goal. Finally, place the alternatives in a priority order, based upon your own value system.

5 Choose among alternatives: Once you have weighed all the evidence, you are ready to select the alternative that seems to be best one for you. You may even choose a combination of alternatives. Your choice in Step 5 may very likely be the same or similar to the alternative you placed at the top of your list at the end of Step 4.

6 Take action: You’re now ready to take some positive action by beginning to implement the alternative you chose in Step 5.

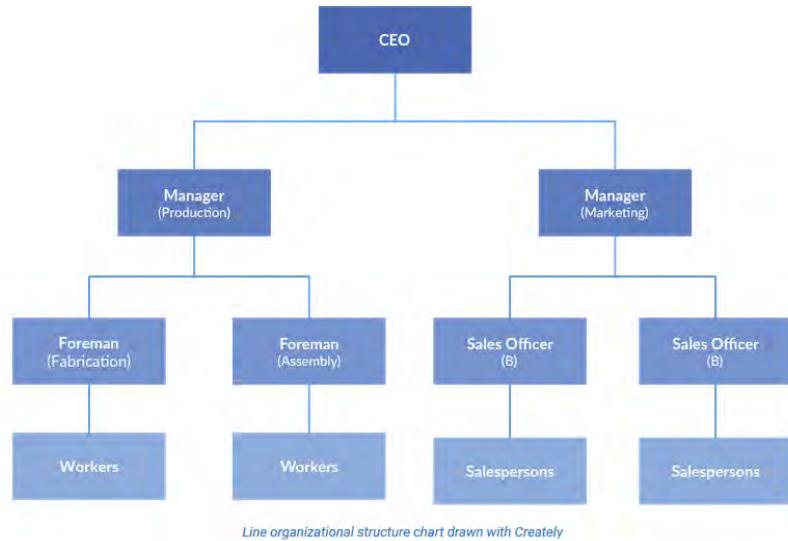
7 Review your decision & its consequences: In this final step, consider the results of your decision and evaluate or not it has resolved the need you identified in Step 1. If the decision has *not* met the identified need, you may want to repeat certain steps of the process to make a new decision. For example, you might want to gather more detailed or somewhat different information or explore additional alternatives.

Organizational Structure: An organizational structure is a system that outlines how certain activities are directed in order to achieve the goals of an organization. These activities can include rules, roles, and responsibilities. The organizational structure also determines how information flows between levels within the company. For example, in a centralized structure, decisions flow from the top down, while in a decentralized structure, decision-making power is distributed among various levels of the organization.

Businesses of all shapes and sizes use organizational structures heavily. They define a specific hierarchy within an organization. A successful organizational structure defines each employee's job and how it fits

within the overall system. Put simply, the organizational structure lays out who does what so the company can meet its objectives. This structuring provides a company with a visual representation of how it is shaped and how it can best move forward in achieving its goals. Organizational structures are normally illustrated in some sort of chart or diagram like a pyramid, where the most powerful members of the organization sit at the top, while those with the least amount are at the bottom. The broadest kind of organizational structure are as mentioned below:

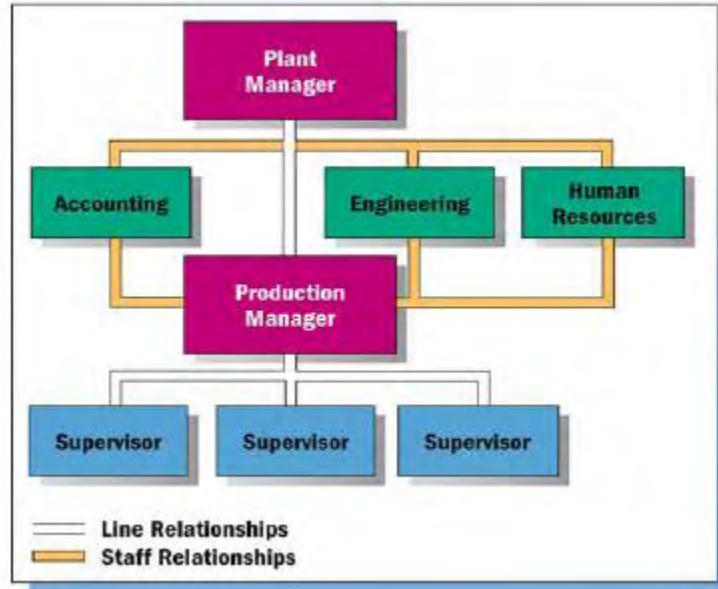
Line Organizational Structure: Line organizational structure is one of the simplest types of organizational structures. Its authority flows from top to bottom. Unlike other structures, specialized and supportive services do not take place in these organizations. The chain of command and each department head has control over their departments. The self-contained department structure can be seen as its main characteristic. Independent decisions can be taken by line officers because of its unified structure. The main advantage of a line organizational structure can be identified as the effective communication that brings stability to the organization.



Line organizational structure chart drawn with Creately

Line and staff organization: The line and staff structure most often is defined by the managers and staff classifications. The structure can, however, climb higher with upper management roles and executive roles built into a tree-style structure.

For example, the **top level** will have a CEO, followed by a president and vice president on the next level. Then, comes the **upper management** for each department, followed by **management for each department**. Under each manager, individual **staff members** will fall into the organizational structure.



This makes it exceptionally easy to assign management responsibilities. The finance manager on the line is responsible for every employee that falls under his or her department. The sales manager is responsible for every sales team member located within their department, same for the HR manager, etc. The idea here is to cleanly assign roles according to department. If more than one manager falls on the line, an upper manager can be assigned the higher level departmental duties on another line.

Committee organizational structure: A **committee organization** is an association of people set up to arrive at solutions to common problems. The line people are given opportunities to discuss their problems in the committee. The **committee organizational structure** is not like line or functional organization, but is similar to staff organization. Its decisions are implemented, whereas staff decisions are not necessarily implemented. It is a formal part of the organizational structure wherein the members are specifically mentioned. For example, the Finance Committee will include all the functional managers, viz. Marketing Manager, Production Manager, Personnel Managers, etc. as members, and the Managing Director as the Chairman. It will decide the financial requirements of each and every department. The decisions taken by the committee are followed by the line people, as the committees are representatives of various functional departments.



Committee organizational structure provides integrated ideas of various related people of the company. Participative management in true form is visible under committee organization. It is an incentive to volunteer to from integrated ideas and to willingly follow them. New ideas and solutions of various problems are feasible with the committee organization. It is a very good example of democratic management wherein every member has an equal opportunity to raise his voice and come to a common solution. Flexibility and technical excellence are possible under this organization. The top management is relieved from certain problems. The company can encounter the changing and uncertain environment in a better way. It facilitates high quality and innovative solutions to technical problems. Coordination and control become easy because open discussion is invited in the committee. Ideas and specialized functions are feasible under committee organization.

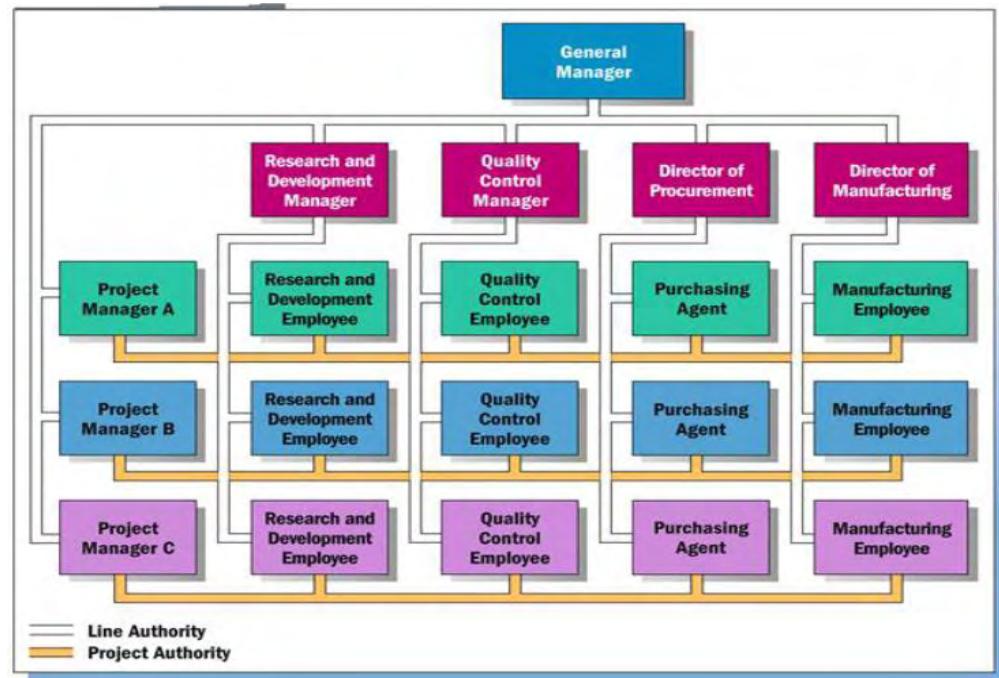
However, committee organizational structure may prove ineffective in some cases because of time consuming and inefficient devices, aggressive attitudes of some persons and inactive role of a particular group. The committee organization should not be used to supplement or support inefficient managers. An able and competent top manager with the capacity to handle the proceedings of the committee and manage disgruntled employees during meetings can get the maximum benefits out of committee organization. On the contrary, a weak and submissive manager or chairman may cause a number of problems in committee organization.

Functional Structure: Functional Structure organizational is a structure which includes undertakings like supervision, direction, management, and allocation of responsibilities. The organizational structure selects how the processes and presentations of the organization can carry. The communication organization structure narrates to how the associate in a company are gathered and to whom can they report. One unoriginal means of establishing individual is done function. Few collective activities in a company contain marketing, HR, manufacture, and bookkeeping. The benefits and importance of functional structure include quick decision making as the members of the group are able to interconnect effortlessly with each other. Also, since the members already own same sets of skill and interests' individuals in this type of structures can easily learn from each other.



Matrix Structure: A matrix organizational structure is a company structure in which the reporting relationships are set up as a grid, or matrix, rather than in the traditional hierarchy. In other words, employees have dual reporting relationships - generally to both a functional manager and a product manager.

The roles and duties incline to be considerably more complex defined in the matrix structure. The matrix structure bonds employees



by both function as well as the product. A matrix company over and over again exploits and develops groups of staffs to accomplish the task, so as to take advantage of the power and in order to hide the weaknesses, of reorganized and practical forms.

Topics Covered:

- Testing principles and objectives
- Test plan
- Types and levels of testing
- Test strategies
- Program verification and validation
- Software quality
- SEI-CMM
- SQA activities
- QA organizations structure
- SQA plan

Software Testing

Software testing is defined as the execution of a program to find its faults. While more time typically is spent on testing than in any other phase of software development, there is considerable confusion about its purpose. Many software professionals, for example, believe that tests are run to show that the program works rather than to learn about its faults.

Myers has provided some useful testing definitions:

- Testing: The execution of a program to find its faults
- Debugging: Diagnosing the error and correct it

Objectives of Testing: Software Testing has different goals and objectives. The major objectives of Software testing are as follows:

- Finding errors which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.
- To check if the system meets the requirements and be executed successfully in the Intended environment.
- To check if the system is “Fit for purpose”.
- To check if the system does what it is expected to do.
- A good test case is one that has a probability of finding a non-discovered error.
- A successful test is one that uncovers a yet undiscovered error.
- A good test is not redundant.
- A good test should be “best of breed”.
- A good test should neither be too simple nor too complex.

Software testing helps in finalizing the software application or product against business and user requirements. It is very important to have good test coverage in order to test the software application completely and make it sure that it's performing well and as per the specifications.

What is a Test Plan?

A test plan is a document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task,

degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

- **Master test plan:** A test plan that typically addresses multiple test levels.
- **Phase test plan:** A test plan that typically addresses one test phase

A test plan contains:

1. **Test plan ID:** This is a unique ID which defines the test plan. It can be a number or name or mix of both, as per the convenience.
2. **Test environment:** This section defines what kind of environment is needed for the testing to carry out. For e.g. in device testing, usually a virtual set up is made to test emergency calling.
3. **Features to be tested/Not tested:** This will have all the details about the features which tester needs to test and what are the feature which are not tested (may be because it is not yet implemented or not tested for that particular release).
4. **Entry/Exit criteria:** These are the terms which define when to start or stop the testing. Standards will be defined under test strategy and followed by testers in test plan.
5. **Status:** Whether a test case is passed or failed or not tested, all these test results are included in test plan with a proper reason.
6. **Types of testing:** The types of testing required such as regression, functional, non-functional, stress etc. are defined and then executed by the respective tester.
7. **Brief Intro:** Brief introduction is also included sometimes so that if any new member joins the team, he should get an idea how things work.

Test Plan Types: One can have the following types of test plans:

- **Master Test Plan:** A single high-level test plan for a project/product that unifies all other test plans.
- **Testing Level Specific Test Plans:** Plans for each level of testing.
 - Unit Test Plan
 - Integration Test Plan
 - System Test Plan
 - Acceptance Test Plan
- **Testing Type Specific Test Plans:** Plans for major types of testing like Performance Test Plan and Security Test Plan.

Unit testing: Individual components of program or project are tested. It is a path test to focus on a relatively small segment of code and aim to exercise a high percentage of the internal path. It is the most ‘micro’ scale of testing. Tests are done on particular functions or code modules. Unit testing requires knowledge of the internal program design and code. It is done by Programmers (not by testers).

Integration Testing: Testing of combined parts of an application to determine their functional correctness. These ‘Parts’ can be code modules, individual applications, client/server applications on a network. There are following types of integration testing:

- **Big bang** is an approach to Integration testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle. So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.
- **Top Down** is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.
- **Bottom Up** is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is

followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

- **Sandwich/Hybrid** is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

System testing: It is a level of software testing where a complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements. It is the process of testing an integrated system to verify that it meets specified requirements.

Acceptance testing: It is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. It is formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Testing Methodologies

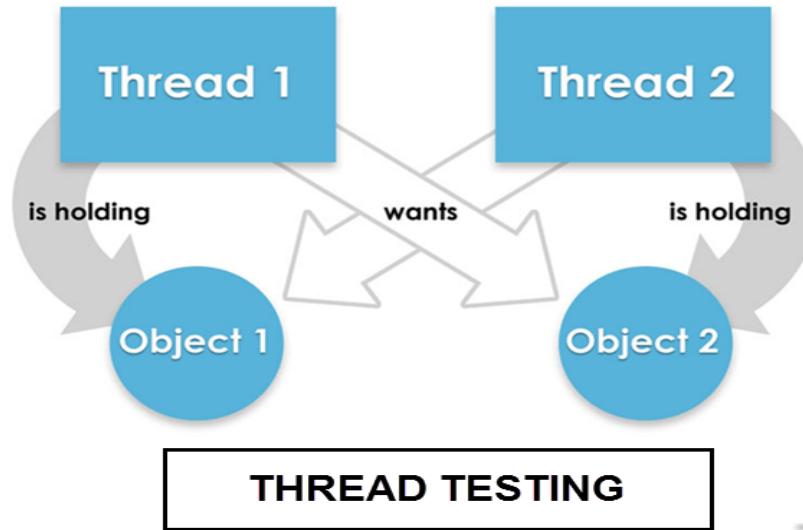
- **Alpha testing:** Alpha testing is a type of acceptance testing; performed to identify all possible issues/bugs before releasing the product to everyday users or the public. The focus of this testing is to simulate real users by using a black box and white box techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually, the testers are internal employees of the organization. To put it as simple as possible, this kind of testing is called alpha only because it is done early on, near the end of the development of the software, and before beta testing. Testing done when development is nearing completion; minor design changes may still be made as a result of such testing.
- **Beta-testing:** Beta Testing of a product is performed by "real users" of the software application in a "real environment" and can be considered as a form of external User Acceptance Testing. Beta version of the software is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation. It is the final test before shipping a product to the customers. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in customer's environment. Testing when development and testing are essentially completed and final bugs and problems need to be found before release.
- **Black box testing:** In black box testing, no knowledge of internal design or code required. Tests are based on requirements and functionality
- **White box testing:** Knowledge of the internal program design and code required. Tests are based on coverage of code statements, branches, paths, conditions.

➤ Criteria	Black Box Testing	White Box Testing
<i>Definition</i>	Black Box Testing is a software testing method in which the internal structure/design/ implementation of the item being tested is NOT known to the tester	White Box Testing is a software testing method in which the internal structure/design/ implementation of the item being tested is known to the tester.
<i>Levels Applicable To</i>	Mainly applicable to higher levels of testing: Acceptance Testing System Testing	Mainly applicable to lower levels of testing: Unit Testing Integration Testing
<i>Responsibility</i>	Generally, independent Software Testers	Generally, Software Developers
<i>Programming Knowledge</i>	Not Required	Required

<i>Implementation Knowledge</i>	Not Required	Required
<i>Basis for Test Cases</i>	Requirement Specifications	Detail Design

For a combination of the two testing methods, see Gray Box Testing.

- **Incremental testing:** After unit testing is completed, developer performs integration testing. It is the process of verifying the interfaces and interaction between modules. While integrating, there are lots of techniques used by developers and one of them is the incremental approach. In Incremental integration testing, the developers integrate the modules one by one using stubs or drivers to uncover the defects. This approach is known as incremental integration testing. To the contrary, big bang is one other integration testing technique, where all the modules are integrated in one shot. It includes:
 - **Top down Integration** - This type of integration testing takes place from top to bottom. Unavailable Components or systems are substituted by stubs
 - **Bottom Up Integration** - This type of integration testing takes place from bottom to top. Unavailable Components or systems are substituted by Drivers
 - **Functional incremental** - The Integration and testing takes place on the basis of the functions or functionalities as per the functional specification document.
- **Thread testing:** Thread testing is defined as a software testing type, which verify the key functional capabilities of a specific task (thread). It is usually conducted at the early stage of Integration Testing phase. Thread based testing is one of the incremental strategies adopted during System Integration Testing. That's why, thread test should probably more properly be called a "thread interaction test."



Types of Thread Testing

Thread based testing are classified into two categories

- **Single thread testing:** A single thread testing involves one application transaction at a time
- **Multi-thread testing:** A multi-thread testing involves several concurrently active transaction at a time

How to do Thread Testing: The thread process focuses on the integration activities rather than the full development lifecycle. For Example,

- Thread-based testing is a generalized form of session-based testing, in that sessions are a form of thread, but a thread is not necessarily a session.
- For thread testing, the thread or program (small functionality) are integrated and tested incrementally as a subsystem, and then executed for a whole system.
- At the lowest level, it provided integrators with better knowledge of the scope of what to test
- Rather than testing software components directly, it required integrators to concentrate on testing logical execution paths in the context of the entire system.

Test strategy: It is a high level document which defines the approach for software testing. It is basically derived from the Business Requirement document. Test strategy is developed by project manager or business analyst. It is kind of static document which sets the standards for testing so not updated often. It contains:

1. **Scope and objective:** The objective of the business and how much testing scope is there is defined under test strategy.
2. **Business Issues:** How much is the budget of the project, how much time is required for testing, how much resources are needed etc. are the part of business issues which needs to be considered before the actual testing starts.
3. **Testing approach:** What type of testing is needed (performance, load, stress, functional etc.) and whether the testing is only manual or automation or both are some of the crucial points which defines the testing approach.
4. **Test deliverables:** What are the documents required from the testing team, how they would keep the record of the testing cycles etc. will be included here.
5. **Defect tracking approach:** Which tool will be used for tracking the defects and how will the testing team communicate with the development team and how the flow would go for defects are decided at this point in test strategy.
6. **Training:** If there is some complex or new tool is introduced in the business then it is helpful if the team members are given proper training. What type of training and the responsible person to conduct such training is defined here?
7. **Automation:** If the project or business needs automation testing then the script language, tool used, reporting and code maintained is planned in test strategy.
8. **Risks:** Nobody can anticipate all the risks beforehand but obvious risks can be avoided and also solution (if risk occur) can be included in the document for future help.

Verification & validation

Before getting into the various forms and strategies of testing we must understand the process of **verifying** and **validating** the software code. Verification and validation is the generic name given to checking processes which ensure that the software **conforms to its specification** and meets **the needs of the customer**. The system should be verified and validated at each stage of the software development process using documents produced in earlier stages. Verification and validation thus starts with **requirements reviews** and continues through **design and code reviews** to **product testing**.

Verification and validation are sometimes confused, but they are different activities. The difference between the two can be summarized as follows:

- *Validation: Are we building the right product?*
- *Verification: Are we building the product right?*

Verification is a static practice of verifying documents, design, code and program. It includes all the activities associated with producing high quality software: inspection, design analysis and specification analysis. It is a relatively objective process. Verification will help to determine whether the software is of

high quality, but it will not ensure that the system is useful. Verification is concerned with whether the system is well-engineered and error-free.

Methods of Verification: Static Testing

- *Walkthrough*
- *Inspection*
- *Review*

Validation is the process of evaluating the final product to check whether the software meets the customer expectations and requirements. It is a dynamic mechanism of validating and testing the actual product.

Methods of Validation: Dynamic Testing

- *Testing*
- *End Users*

Differences between verification and validation

Verification

1. **Verification** is a static practice of verifying documents, design, code and program.
2. It does not involve executing the code.
3. It is human based checking of documents and files.
4. **Verification** uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.
5. **Verification** is to check whether the software conforms to specifications.
6. It can catch errors that validation cannot catch. It is low level exercise.
7. Target is requirements specification, application and software architecture, high level, complete design, and database design etc.
8. **Verification** is done by QA team to ensure that the software is as per the specifications in the SRS document.
9. It generally comes first-done before validation.

Validation

1. **Validation** is a dynamic mechanism of validating and testing the actual product.
2. It always involves executing the code.
3. It is computer based execution of program.
4. **Validation** uses methods like black box (functional) testing, gray box testing, and white box (structural) testing etc.
5. **Validation** is to check whether software meets the customer expectations and requirements.
6. *It can catch errors that verification cannot catch. It is High Level Exercise.*
7. Target is actual product-a unit, a module, a bent of integrated modules, and effective final product.
8. *Validation is carried out with the involvement of testing team.*
9. It generally follows after **verification**.

Software Quality Assurance

SQA is a set of activities for ensuring quality in software engineering processes. It ensures that developed software meets and complies with the defined or standardized quality specifications. SQA is an ongoing process within the Software Development Life Cycle (SDLC) that routinely checks the developed software to ensure it meets the desired quality measures.

SQA practices are implemented in most types of software development, regardless of the underlying software development model being used. SQA incorporates and implements software testing methodologies to test the software. Rather than checking for quality after completion, SQA processes test for quality in each phase of development, until the software is complete. With SQA, the software development process moves into the next phase only once the current/previous phase complies with the required quality standards. SQA generally works on one or more industry standards that help in building software quality guidelines and implementation strategies.

It includes the following activities –

- Process definition and implementation
- Auditing
- Training

Processes could be –

- Software Development Methodology
- Project Management
- Configuration Management
- Requirements Development/Management
- Estimation
- Software Design
- Testing, etc.

Once the processes have been defined and implemented, Quality Assurance has the following responsibilities –

- Identify the weaknesses in the processes
- Correct those weaknesses to continually improve the process

Components of SQA System: An SQA system always combines a wide range of SQA components. These components can be classified into the following six classes –

- **Pre-project components:** This assures that the project commitments have been clearly defined considering the resources required, the schedule and budget; and the development and quality plans have been correctly determined.
- **Components of project life cycle activities assessment:** The project life cycle is composed of two stages: the development life cycle stage and the operation–maintenance stage.
- The development life cycle stage components detect design and programming errors. Its components are divided into the following sub-classes: Reviews, Expert opinions, and Software testing.
- The SQA components used during the operation–maintenance phase include specialized maintenance components as well as development life cycle components, which are applied mainly for functionality to improve the maintenance tasks.
- **Components of infrastructure error prevention and improvement:** The main objective of these components, which is applied throughout the entire organization, is to eliminate or at least reduce the rate of errors, based on the organization's accumulated SQA experience.
- **Components of software quality management:** This class of components deal with several goals, such as the control of development and maintenance activities, and the introduction of early managerial support actions that mainly prevent or minimize schedule and budget failures and their outcomes.
- **Components of standardization, certification, and SQA system assessment:** These components implement international professional and managerial standards within the organization. The main objectives of this class are utilization of international professional knowledge, improvement of coordination of the organizational quality systems with other organizations, and assessment of the achievements of quality systems according to a common scale. The various standards may be classified into two main groups: quality management standards and project process standards.

- **Organizing for SQA – the human components:** The SQA organizational base includes managers, testing personnel, the SQA unit and the persons interested in software quality such as SQA trustees, SQA committee members, and SQA forum members. Their main objectives are to initiate and support the implementation of SQA components, detect deviations from SQA procedures and methodology, and suggest improvements.
- **Pre-project Software Quality Components:** These components help to improve the preliminary steps taken before starting a project. It includes –
 - Contract Review
 - Development and Quality Plans
- **Contract Review:** Normally, a software is developed for a contract negotiated with a customer or for an internal order to develop a firmware to be embedded within a hardware product. In all these cases, the development unit is committed to an agreed-upon functional specification, budget and schedule. Hence, contract review activities must include a detailed examination of the project proposal draft and the contract drafts.
 - Specifically, contract review activities include –
 - Clarification of the customer's requirements
 - Review of the project's schedule and resource requirement estimates
 - Evaluation of the professional staff's capacity to carry out the proposed project
 - Evaluation of the customer's capacity to fulfil his obligations
 - Evaluation of development risks
- **Development and Quality Plans:** After signing the software development contract with an organization or an internal department of the same organization, a development plan of the project and its integrated quality assurance activities are prepared. These plans include additional details and needed revisions based on prior plans that provided the basis for the current proposal and contract.

Most of the time, it takes several months between the tender submission and the signing of the contract. During these period, resources such as staff availability, professional capabilities may get changed. The plans are then revised to reflect the changes that occurred in the interim.

The main issues treated in the project development plan are –

- Schedules
- Required manpower and hardware resources
- Risk evaluations
- Organizational issues: team members, subcontractors and partnerships
- Project methodology, development tools, etc.
- Software reuse plans

The main issues treated in the project's quality plan are –

- Quality goals, expressed in the appropriate measurable terms
- Criteria for starting and ending each project stage
- Lists of reviews, tests, and other scheduled verification and validation activities

Software Quality

Software quality is defined as a field of study and practice that describes the desirable attributes of software products. There are two main approaches to software quality: defect management and quality attributes. The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs.

As with any definition, the definition of 'software quality' is also varied and debatable. Some even say that 'quality' cannot be defined and some say that it can be defined but only in a particular context. Some even state foolishly, yet confidently, that 'quality is simply a lack of bugs'. Whatever the definition, it is true that high quality is something we all aspire to.

The various factors, which influence the software, are termed as software factors. They can be broadly divided into two categories. The first category of the factors is of those that can be measured directly such as the number of logical errors, and the second category clubs those factors which can be measured only indirectly. For example, maintainability but each of the factors is to be measured to check for the content and the quality control.

Several models of software quality factors and their categorization have been suggested over the years. The classic model of software quality factors, suggested by McCall, consists of 11 factors (McCall et al., 1977). Similarly, models consisting of 12 to 15 factors, were suggested by Deutsch and Willis (1988) and by Evans and Marciniak (1987).

All these models do not differ substantially from McCall's model. The McCall factor model provides a practical, up-to-date method for classifying software requirements (Pressman, 2000).

McCall's Factor Model: This model classifies all software requirements into 11 software quality factors. The 11 factors are grouped into three categories – product operation, product revision, and product transition factors.

- **Product operation factors** – Correctness, Reliability, Efficiency, Integrity, Usability.
- **Product revision factors** – Maintainability, Flexibility, Testability.
- **Product transition factors** – Portability, Reusability, Interoperability.

Product Operation Software Quality Attributes: According to McCall's model, product operation category includes five software quality factors, which deal with the requirements that directly affect the daily operation of the software. They are as follows –

- **Correctness:** These requirements deal with the correctness of the output of the software system. They include –
 - Output mission
 - The required accuracy of output that can be negatively affected by inaccurate data or inaccurate calculations.
 - The completeness of the output information, which can be affected by incomplete data.
 - The up-to-date ness of the information defined as the time between the event and the response by the software system.
 - The availability of the information.
 - The standards for coding and documenting the software system.
- **Reliability:** Reliability requirements deal with service failure. They determine the maximum allowed failure rate of the software system, and can refer to the entire system or to one or more of its separate functions.
- **Efficiency:** It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS). It also deals with the time between recharging of the system's portable units, such as, information system units located in portable computers, or meteorological units placed outdoors.
- **Integrity:** This factor deals with the software system security, that is, to prevent access to unauthorized persons, also to distinguish between the group of people to be given read as well as write permit.
- **Usability:** Usability requirements deal with the staff resources needed to train a new employee and to operate the software system.

Product Revision Quality Factors: According to McCall's model, three software quality factors are included in the product revision category. These factors are as follows –

- **Maintainability:** This factor considers the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.
- **Flexibility:** This factor deals with the capabilities and efforts required to support adaptive maintenance activities of the software. These include adapting the current software to additional circumstances and customers without changing the software. This factor's requirements also support perfective maintenance activities, such as changes and additions to the software in order to improve its service and to adapt it to changes in the firm's technical or commercial environment.
- **Testability:** Testability requirements deal with the testing of the software system as well as with its operation. It includes predefined intermediate results, log files, and also the automatic diagnostics performed by the software system prior to starting the system, to find out whether all components of the system are in working order and to obtain a report about the detected faults. Another type of these requirements deals with automatic diagnostic checks applied by the maintenance technicians to detect the causes of software failures.

Product Transition Software Quality Factor: According to McCall's model, three software quality factors are included in the product transition category that deals with the adaptation of software to other environments and its interaction with other software systems. These factors are as follows –

- **Portability:** Portability requirements tend to the adaptation of a software system to other environments consisting of different hardware, different operating systems, and so forth. The software should be possible to continue using the same basic software in diverse situations.
- **Reusability:** This factor deals with the use of software modules originally designed for one project in a new software project currently being developed. They may also enable future projects to make use of a given module or a group of modules of the currently developed software. The reuse of software is expected to save development resources, shorten the development period, and provide higher quality modules.
- **Interoperability:** Interoperability requirements focus on creating interfaces with other software systems or with other equipment firmware. For example, the firmware of the production machinery and testing equipment interfaces with the production control software.

Software Quality Metrics

Software metrics can be classified into three categories –

- **Product metrics** – Describes the characteristics of the product such as size, complexity, design features, performance, and quality level.
- **Process metrics** – these characteristics can be used to improve the development and maintenance activities of the software.
- **Project metrics** – these metrics describe the project characteristics and execution. Examples include the number of software developers, the staffing pattern over the life cycle of the software, cost, schedule, and productivity.

Some metrics belong to multiple categories. For example, the in-process quality metrics of a project are both process metrics and project metrics.

Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. These are more closely associated with process and product metrics than with project metrics.

Software quality metrics can be further divided into three categories –

- Product quality metrics

- In-process quality metrics
- Maintenance quality metrics

Product Quality Metrics: This metrics include the following –

- Mean Time to Failure
- Defect Density
- Customer Problems
- Customer Satisfaction

Mean Time to Failure: It is the time between failures. This metric is mostly used with safety critical systems such as the airline traffic control systems, avionics, and weapons.

Defect Density: It measures the defects relative to the software size expressed as lines of code or function point, etc. i.e., it measures code quality per unit. This metric is used in many commercial software systems.

Customer Problems: It measures the problems that customers encounter when using the product. It contains the customer's perspective towards the problem space of the software, which includes the non-defect oriented problems together with the defect problems. The problems metric is usually expressed in terms of **Problems per User-Month (PUM)**.

PUM = Total Problems that customers reported (true defect and non-defect oriented problems) for a time period + Total number of license months of the software during the period

Where,

$$\text{Number of license-month of the software} = \text{Number of install license of the software} \times \text{Number of months in the calculation period}$$

PUM is usually calculated for each month after the software is released to the market, and also for monthly averages by year.

Customer Satisfaction: Customer satisfaction is often measured by customer survey data through the five-point scale –

- Very satisfied
- Satisfied
- Neutral
- Dissatisfied
- Very dissatisfied

Satisfaction with the overall quality of the product and its specific dimensions is usually obtained through various methods of customer surveys. Based on the five-point-scale data, several metrics with slight variations can be constructed and used, depending on the purpose of analysis. For example –

- Percent of completely satisfied customers
- Percent of satisfied customers
- Percent of dis-satisfied customers
- Percent of non-satisfied customers

Usually, this percent satisfaction is used.

In-process Quality Metrics: In-process quality metrics deals with the tracking of defect arrival during formal machine testing for some organizations. This metric includes –

- Defect density during machine testing
- Defect arrival pattern during machine testing
- Phase-based defect removal pattern
- Defect removal effectiveness

Defect density during machine testing: Defect rate during formal machine testing (testing after code is integrated into the system library) is correlated with the defect rate in the field. Higher defect rates found during testing is an indicator that the software has experienced higher error injection during its development process, unless the higher testing defect rate is due to an extraordinary testing effort.

This simple metric of defects per KLOC or function point is a good indicator of quality, while the software is still being tested. It is especially useful to monitor subsequent releases of a product in the same development organization.

Defect arrival pattern during machine testing: The overall defect density during testing will provide only the summary of the defects. The pattern of defect arrivals gives more information about different quality levels in the field. It includes the following –

- The defect arrivals or defects reported during the testing phase by time interval (e.g., week). Here all of which will not be valid defects.
- The pattern of valid defect arrivals when problem determination is done on the reported problems. This is the true defect pattern.
- The pattern of defect backlog overtime. This metric is needed because development organizations cannot investigate and fix all the reported problems immediately. This is a workload statement as well as a quality statement. If the defect backlog is large at the end of the development cycle and a lot of fixes have yet to be integrated into the system, the stability of the system (hence its quality) will be affected. Retesting (regression test) is needed to ensure that targeted product quality levels are reached.

Phase-based defect removal pattern: This is an extension of the defect density metric during testing. In addition to testing, it tracks the defects at all phases of the development cycle, including the design reviews, code inspections, and formal verifications before testing.

Because a large percentage of programming defects is related to design problems, conducting formal reviews, or functional verifications to enhance the defect removal capability of the process at the front-end reduces error in the software. The pattern of phase-based defect removal reflects the overall defect removal ability of the development process.

With regard to the metrics for the design and coding phases, in addition to defect rates, many development organizations use metrics such as inspection coverage and inspection effort for in-process quality management.

Defect removal effectiveness: It can be defined as follows –

$$DRE = \frac{\text{Defect removed during a development phase}}{\text{Defects latent in the product}} \times 100\%$$

This metric can be calculated for the entire development process, for the front-end before code integration and for each phase. It is called **early defect removal** when used for the front-end and **phase effectiveness** for specific phases. The higher the value of the metric, the more effective the development process and the fewer the defects passed to the next phase or to the field. This metric is a key concept of the defect removal model for software development.

Maintenance Quality Metrics: Although much cannot be done to alter the quality of the product during this phase, following are the fixes that can be carried out to eliminate the defects as soon as possible with excellent fix quality.

- Fix backlog and backlog management index
- Fix response time and fix responsiveness
- Percent delinquent fixes
- Fix quality

Fix backlog and backlog management index: Fix backlog is related to the rate of defect arrivals and the rate at which fixes for reported problems become available. It is a simple count of reported problems that remain at the end of each month or each week. Using it in the format of a trend chart, this metric can provide meaningful information for managing the maintenance process.

Backlog Management Index (BMI) is used to manage the backlog of open and unresolved problems.

$$BMI = \frac{\text{Number of problems closed during the month}}{\text{Number of problems arrived during the month}} \times 100\%$$

If BMI is larger than 100, it means the backlog is reduced. If BMI is less than 100, then the backlog increased.

Fix response time and fix responsiveness: The fix response time metric is usually calculated as the mean time of all problems from open to close. Short fix response time leads to customer satisfaction.

The important elements of fix responsiveness are customer expectations, the agreed-to fix time, and the ability to meet one's commitment to the customer.

Percent delinquent fixes: It is calculated as follows –

Percent Delinquent Fixes=

$$\frac{\text{Number of fixes that exceeded the response time criteria by severity level}}{\text{Number of fixes delivered in a specified time}} \times 100\%$$

Fix Quality: Fix quality or the number of defective fixes is another important quality metric for the maintenance phase. A fix is defective if it did not fix the reported problem, or if it fixed the original problem but injected a new defect. For mission-critical software, defective fixes are detrimental to customer satisfaction. The metric of percent defective fixes is the percentage of all fixes in a time interval that is defective.

A defective fix can be recorded in two ways: Record it in the month it was discovered or record it in the month the fix was delivered. The first is a customer measure; the second is a process measure. The difference between the two dates is the latent period of the defective fix.

Usually the longer the latency, the more will be the customers that get affected. If the number of defects is large, then the small value of the percentage metric will show an optimistic picture. The quality goal for the maintenance process, of course, is zero defective fixes without delinquency.

SEI –CMM

SEI stands for 'Software Engineering Institute' at Carnegie-Mellon University, initiated by the U.S. Defense Department to help improve software development processes.

CMM stands for 'Capability Maturity Model', developed by the SEI. It's a model of 5 levels of organizational 'maturity' that determine effectiveness in delivering quality software. It is geared to large organizations such as large U.S. Defense Department contractors. However, many of the QA processes involved are appropriate to any organization, and if reasonably applied can be helpful. Organizations can receive CMM ratings by undergoing assessments by qualified auditors.

- **Level 1** – Characterized by chaos, periodic panics, and heroic efforts required by individuals to successfully complete projects. Few if any processes in place; successes may not be repeatable.
- **Level 2** – Software project tracking, requirements management, realistic planning, and configuration management processes are in place, successful practices can be repeated.
- **Level 3** – Standard software development and maintenance processes are integrated throughout an organization, a Software Engineering Process Group is in place to oversee software processes, and training programs are used to ensure understanding and compliance.
- **Level 4** – Metrics are used to track productivity, processes, and products. Project performance is predictable, and quality is consistently high.
- **Level 5** – The focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required.

Software Engineering Institute (**SEI**) developed Capability Maturity Model (**CMM**). **CMM** describes the prime elements -planning, engineering, managing software development and maintenance **CMM can be used for**

- Software process improvement
- Software process assessment
- Software capability evaluations

What is Maturity?

Definitions vary but mature processes are generally thought to be:

- Well defined

- Repeatable
- Measured
- Analyzed
- Improved

And most importantly ... effective. Poor but mature processes are just as bad as no maturity at all!

The CMM helps to solve the maturity problem by defining a set of practices and providing a general framework for improving them. The CMM focus is on identifying key process areas and the exemplary practices that may comprise a disciplined software process.

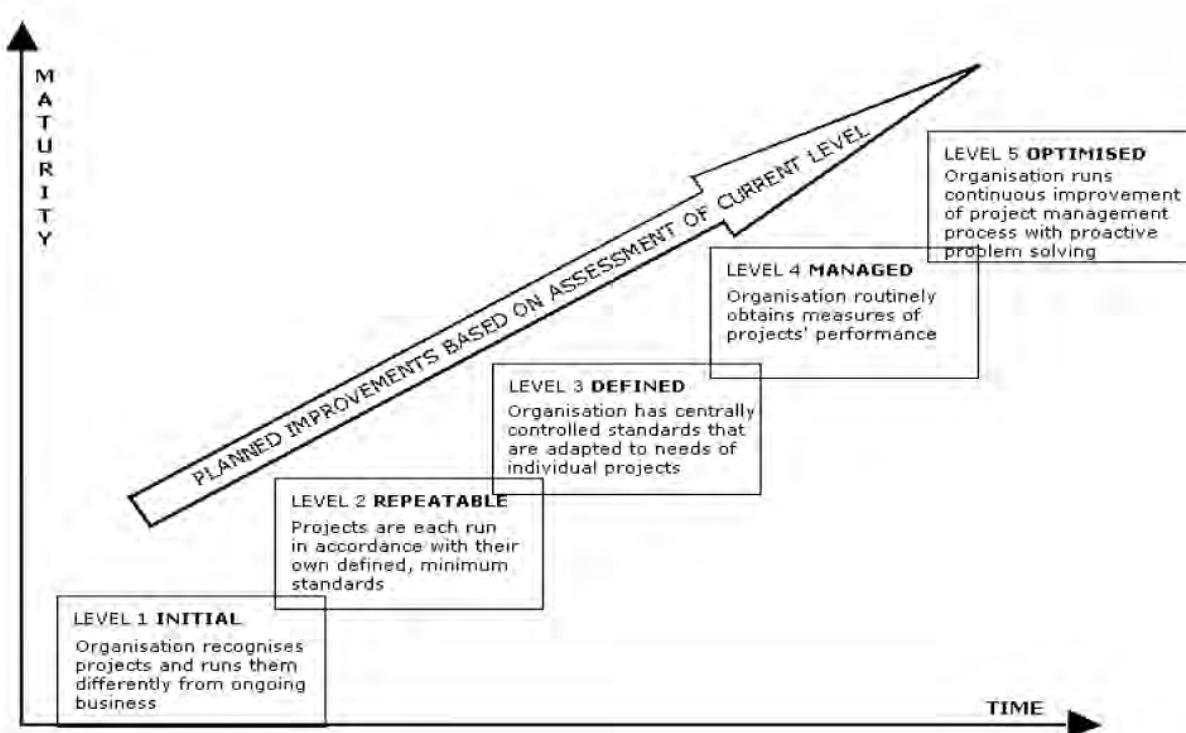
Immature vs Mature Organization:

There are following characteristics of an immature organization:

- Process improvised during project
- Approved processes being ignored
- Reactive, not proactive
- Unrealistic budget and schedule
- Quality sacrificed for schedule
- No objective measure of quality

There are following characteristics of a mature organization:

- Inter-group communication and coordination
- Work accomplished according to plan
- Practices consistent with processes
- Processes updated as necessary
- Well defined roles/responsibilities
- Management formally commits



CMM Level	Focus	Key Process Areas
1. Initial	Competent people	
2. Repeatable	Project management	Software project planning Software configuration management
3. Defined	Definition of processes	Process definition Training program Peer reviews
4. Managed	Product and process quality	Quantitative process metrics Software quality management
5. Optimizing	Continuous process improvement	Defect prevention Process change management Technology change management

QA organizations structure

When establishing a new QA organizational structure, one of the first activities is to determine what the QA organizational structure will look like. It is important to gain executive approval and support with the QA organizational structure that you will build. There are a few critical components of a QA organization. Those components are:

- **Functional Testing**
- **Functional Testing**
- **Automated Testing**
- **Performance Testing**
- **Environment Management**

Depending on the size of the QA organizational structure, there will be flexibility that you will be able to setup. Usually the first focus area is going to be on the functional testing component. Once that group is established, you will then be able to add the supporting cast. For example, you don't want to build the automation or performance team first because they will not necessarily have much work to do. If you wanted to concentrate on two areas in parallel, you could setup the functional team with the environment management team.

If I were setting up a small to medium size organization, I would appoint three leads or managers. I would have functional testing lead, environments lead, and automation/performance lead. This will allow sufficient focus with those disciplines. If you have the luxury to appoint a fourth lead, I would separate automation and performance. That will really allow your organization to have the right amount of focus with all those disciplines.

That is just a general guide. If you find that your organization is not working as well as you think it should, it is probably a good idea to restructure in order to have the best QA organization possible.

Topics Covered:

- Software configuration management
- SCM tasks and roles
- Risk management
- Risk management process
- SPM tools

Configuration Management (CM) is the process of identifying and defining the configuration items in a system, controlling the release and change of these items throughout the system lifecycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items. Configuration Management is practiced in one form or another as part of any software engineering project where several individuals or organizations have to coordinate their activities. While the basic disciplines of Configuration Management are common to both hardware and software engineering projects, there are some differences in emphasis due to the nature of software products.

Software Configuration Management (SCM) is a system for managing the evolution of software products, both during the initial stages of development and during all stages of maintenance. A software product encompasses the complete set of computer programs, procedures, and associated documentation and data designated for delivery to a user. All supporting software used in development, even though not part of the software product, should also be controlled by SCM.

The SCM system is the collection of activities performed during a software engineering project to:

- determine and identify those entities of the software product that need to be controlled;
- ensure those entities have necessary and accurate definitions and documentation;
- ensure changes are made to the entities in a controlled manner;
- ensure that the correct version of the entities/software product are being used; and
- ascertain, at any point in time, the status of an entity (e.g., whether a specific entity is completed, being changed, waiting to be tested, or released to the customer).

SCM is performed within the context of several basic configuration management disciplines, including:

- Configuration Identification
- Configuration Control (change control)
- Configuration Status Accounting
- Audits and Reviews
- Release Processing

Why Is Software Configuration Management Important?

The primary reason for implementing an SCM system is to keep the changing and iterative entities/software product(s) in a non-degrading state throughout the software lifecycle. This is a challenge that must be met in order to develop and maintain quality software products. The quality of the software products is fundamental to the level of quality of the complete system.

SCM provides a common point of integration for all planning, oversight, and implementation activities for a software project or product line. It provides the framework (labeling and identification) for interfacing different activities and defining the mechanisms (change controls) necessary for coordinating parallel activities of different groups. SCM also provides a framework for controlling computer program interfaces with their underlying support hardware, and coordinating software changes when both hardware and

software may be evolving during development or maintenance activities. SCM provides management with the visibility (through status accounting and audits) of the evolving software products that make technical and managerial activities more effective.

What Are the Benefits of Software Configuration Management?

SCM provides significant benefits to all projects regardless of size, scope, and complexity. Some of the most common benefits experienced by project teams applying the SCM disciplines described in this guide are possible because the SCM system:

- Provides a snapshot of dynamically changing software to help:
 - Make decisions based on an instantaneous view
 - Determine status at module and system levels
 - Make better decisions about the future of a software project.
- Tracks concurrent development of modules or components of the overall system to:
 - Prevent different developers from making changes to the same module at the same time
 - Allow for the overall system progress to be faster
 - Provide visibility of the entire software project to all developers.
- Organizes all concurrently developing code and associated documentation to:
 - Save overall project time
 - Focus each phase of the software product development to be organized and executed in a documented, prescribed manner.

The effective use of an SCM system also:

- Permits the orderly development of software configuration items.
- Ensures the orderly release and implementation of new or revised software products.
- Ensures that only approved changes to both new and existing software products are implemented.
- Ensures that the software changes that are implemented are in accordance with approved specifications.
- Ensures that the documentation accurately reflects updates.
- Evaluates and communicates the impact of changes.
- Prevents unauthorized changes from being made.

Tasks in SCM process

Configuration Identification: Configuration identification is a method of determining the scope of the software system. With the help of this step, you can manage or control something even if you don't know what it is. It is a description that contains the CSCI type (Computer Software Configuration Item), a project identifier and version information.

Activities during this process:

- Identification of configuration Items like source code modules, test case, and requirements specification.
- Identification of each CSCI in the SCM repository, by using an object-oriented approach
- The process starts with basic objects which are grouped into aggregate objects. Details of what, why, when and by whom changes in the test are made
- Every object has its own features that identify its name that is explicit to all other objects
- List of resources required such as the document, the file, tools, etc.

Example:

- Instead of naming a File login.php its should be named login_v1.2.php where v1.2 stands for the version number of the file
- Instead of naming folder "Code" it should be named "Code_D" where D represents code should be backed up daily.

Baseline: A baseline is a formally accepted version of a software configuration item. It is designated and fixed at a specific time while conducting the SCM process. It can only be changed through formal change control procedures.

Activities during this process:

- Facilitate construction of various versions of an application
- Defining and determining mechanisms for managing various versions of these work products
- The functional baseline corresponds to the reviewed system requirements
- Widely used baselines include functional, developmental, and product baselines

In simple words, baseline means ready for release.

Change Control: Change control is a procedural method which ensures quality and consistency when changes are made in the configuration object. In this step, the change request is submitted to software configuration manager.

Activities during this process:

- Control ad-hoc change to build stable software development environment. Changes are committed to the repository
- The request will be checked based on the technical merit, possible side effects and overall impact on other configuration objects.
- It manages changes and making configuration items available during the software lifecycle

Configuration Status Accounting: Configuration status accounting tracks each release during the SCM process. This stage involves tracking what each version has and the changes that lead to this version.

Activities during this process:

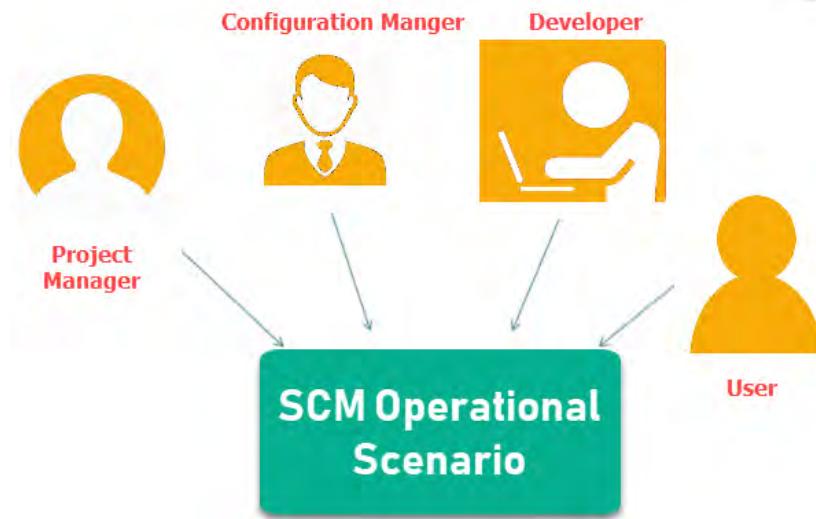
- Keeps a record of all the changes made to the previous baseline to reach a new baseline
- Identify all items to define the software configuration
- Monitor status of change requests
- Complete listing of all changes since the last baseline
- Allows tracking of progress to next baseline
- Allows to check previous releases/versions to be extracted for testing

Configuration Audits and Reviews: Software Configuration audits verify that all the software product satisfies the baseline needs. It ensures that what is built is what is delivered.

Activities during this process:

- Configuration auditing is conducted by auditors by checking that defined processes are being followed and ensuring that the SCM goals are satisfied.
- To verify compliance with configuration control standards. auditing and reporting the changes made
- SCM audits also ensure that traceability is maintained during the process.
- Ensures that changes made to a baseline comply with the configuration status reports
- Validation of completeness and consistency

Participant of SCM process: Following are the key participants in SCM



1. Configuration Manager

- Configuration Manager is the head who is Responsible for identifying configuration items.
- CM ensures team follows the SCM process
- He / She needs to approve or reject change requests

2. Developer

- The developer needs to change the code as per standard development activities or change requests.
- He is responsible for maintaining configuration of code.
- The developer should check the changes and resolves conflicts

3. Auditor

- The auditor is responsible for SCM audits and reviews.
- Need to ensure the consistency and completeness of release.

4. Project Manager:

- Ensure that the product is developed within a certain time frame
- Monitors the progress of development and recognizes issues in the SCM process
- Generate reports about the status of the software system
- Make sure that processes and policies are followed for creating, changing, and testing

5. User

The end user should understand the key SCM terms to ensure he has the latest version of the software

Software Configuration Management Plan

The SCMP (Software Configuration management planning) process planning begins at the early phases of a project. The outcome of the planning phase is the SCM plan which might be stretched or revised during the project.

- The SCMP can follow a public standard like the IEEE 828 or organization specific standard
- It defines the types of documents to be management and a document naming. Example Test_v1
- SCMP defines the person who will be responsible for the entire SCM process and creation of baselines.
- Fix policies for version management & change control
- Define tools which can be used during the SCM process
- Configuration management database for recording configuration information.

Software Configuration Management Tools: Any Change management software should have the following 3 Key features:

- **Concurrency Management:** When two or more tasks are happening at the same time, it is known as concurrent operation. Concurrency in context to SCM means that the same file being edited by multiple persons at the same time. If concurrency is not managed correctly with SCM tools, then it may create many pressing issues.
- **Version Control:** SCM uses archiving method or saves every change made to file. With the help of archiving or save feature, it is possible to roll back to the previous version in case of issues.
- **Synchronization:** Users can check out more than one files or an entire copy of the repository. The user then works on the needed file and checks in the changes back to the repository. They can synchronize their local copy to stay updated with the changes made by other team members.

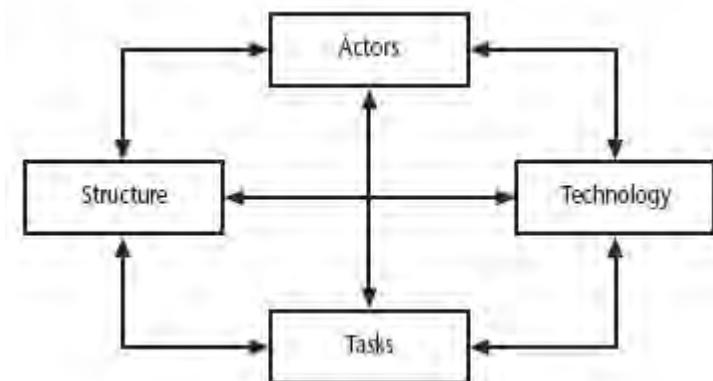
Risk Management

Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. A possibility of suffering from loss in software development process is called a software risk. Loss can be anything, increase in production cost, development of poor quality software, not being able to complete the project on time. Software risk exists because the future is uncertain and there are many known and unknown things that cannot be incorporated in the project plan.

Boehm's top 10 development risks

<i>Risk</i>	<i>Risk reduction techniques</i>
Personnel shortfalls	Staffing with top talent; job matching; team building; training and career development; early scheduling of key personnel
Unrealistic time and cost estimates	Multiple estimation techniques; design to cost; incremental development; recording and analysis of past projects; standardization of methods
Developing the wrong software functions	Improved software evaluation; formal specification methods; user surveys; prototyping; early user manuals
Developing the wrong user interface	Prototyping; task analysis; user involvement
Gold plating	Requirements scrubbing, prototyping, design to cost
Late changes to requirements	Change control, incremental development
Shortfalls in externally supplied components	Benchmarking, inspections, formal specifications, contractual agreements, quality controls
Shortfalls in externally performed tasks	Quality assurance procedures, competitive design etc
Real time performance problems	Simulation, prototyping, tuning
Development technically too difficult	Technical analysis, cost-benefit analysis, prototyping, training

Categories of Risk



Actors: refers to all the people involved in the development of the application in question. These include the various development specialists, the different user group, and managers with differing responsibilities. A typical risk in this area is that high staff turnover leads to information of value to the project being lost. For example, if a software developer builds a software component and then leaves before it has been fully tested, the team member taking over the component might find that their lack of familiarity with the software makes diagnosis and correction of faults difficult.

Technology: Encompasses both the technology used to implement the application and that embedded in the delivered products.

Risk here could relate to the appropriateness of the technologies and to possible fault within them, especially if they are novel

Structure: describes the management structures and systems including those affecting planning and control. For example, the implementation will need the users to carry out some tasks, but the responsibility for managing the users' contribution to the project might not have been clearly allocated.

Tasks means the work to be carried out. For example the complexity of the work might lead to delays because of the additional time required to co-ordinate and integrate the large number of different elements.

Types of Risk

The various categories of risks associated with software project management are enumerated below.

1. Schedule / Time-Related / Delivery Related Planning Risks
2. Budget / Financial Risks
3. Operational / Procedural Risks
4. Technical / Functional / Performance Risks
5. Other Unavoidable Risks

Schedule / Time-Related / Delivery Related Planning Risks

These risks are related to running behind schedule and are essential time-related risks, which directly impact the delivery of the project.

Some of the reasons for such risks are

- Incorrect Time Estimation, and consequently an incorrect project schedule
- Improper Resource Allocation
- Underutilization of Resources
- Superficial Understanding of Project Complexities
- Unexpected Expansion of Project Scope
- Incorrect time estimation may occur because activities may have external dependencies such as client approvals, subcontractors etc. and a delay in a critical path activity has a cascading effect on the entire project.

- Resource Allocation may be improper / Underutilization of resources may take place, especially if resources are shared between projects.
- A silo approach of members in various teams in a project may lead to an isolated superficial understanding of project complexities which may result in delays in subsequent stages e.g. when different development teams work on various aspects of a software project and run into issues during system/integration testing.

Budget / Financial Risks

These are the monetary risks which are associated with budget overruns.

Some of the reasons for such risks are

- Improper Budget Estimation
- Cost Overruns due to underutilization of resources
- Expansion of Project Scope
- Improper Tracking of Finances
- Underutilization of resources especially happens when resources are shared between projects because it becomes difficult to effectively manage such resources and a certain amount of productivity may go waste.
- Further, unexpected expansion of project scope (due to addition of features by clients, etc) may lead to budget overruns as such expansions may not have been factored in to the original estimates.
- Delay of projects may also have certain penalty costs associated with it e.g. construction projects

Operational / Procedural Risks

These are risks which are associated with the day-to-day operational activities of the project.

These could be due to any of the below reasons

- Improper Process Implementation
- Silo approach followed by software development teams leading to conflicts
- Conflicting Priorities
- Lack of conflict resolution / team spirit
- Lack of clarity in responsibilities
- Breakdown in communications
- Lack of sufficient training

Effective team communication is an essential part of project management and in people-intensive projects such as software projects, there is a strong need for an established communication structure, a setup for escalation, a conflict resolution process, established project priorities and above all, the employees need to be trained in making use of these processes within the organization.

Technical / Functional / Performance Risks

These are technical risks associated with the functionality of the software or with respect to the software performance.

- In order to compensate for excessive budget overruns and schedule overruns, companies sometimes reduce the functionality of the software.
- Software testing is a downstream stage in the software development lifecycle and as the project falls behind schedule, downstream activity times are shrunk in order to meet delivery dates which results in insufficient software testing.
- Further, developers face a constant trade-off between achieving maximum functionality of the software (in terms of software features) and peak performance (maximum speed and quick response time by minimizing and eliminating unnecessary frills from the software)

In order to maintain the sanctity of the software development process, while simultaneously catering to the customer's needs, a mutually agreed-upon cut-off date should be determined, beyond which "expected software functionality" would be frozen and any further requirements would be handled in subsequent software's releases.

Other Unavoidable Risks

All the risks described above are those which can be anticipated to a certain extent and planned for in advance. However there are certain risks which are unavoidable in nature.

The reasons for such unavoidable risks are described below.

- Changes in government policy
- Obsolescence of software due to new technology from a rival company
- Loss of contracts due to changes at customers end

Although these risks are broadly unavoidable, an organization may anticipate and thereby reduce the impact of such risks by

- keeping abreast with changes in government policy
- monitoring the competition
- catering to the needs of the customer and ensuring customer satisfaction

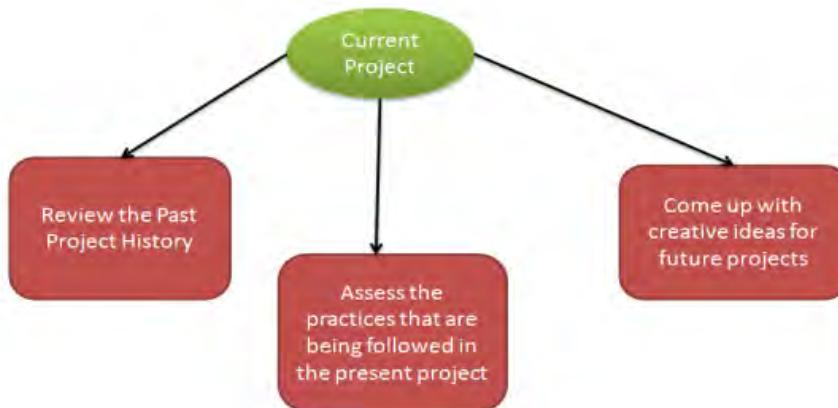
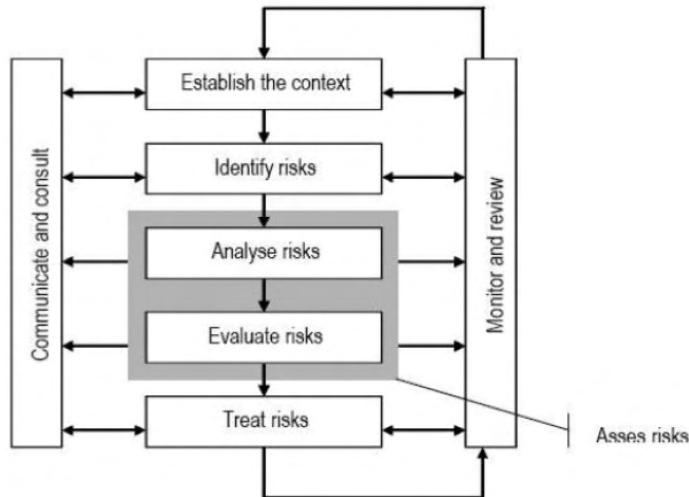
Risk Management comprises of following processes:

1. Software Risk Identification
2. Software Risk Analysis
3. Software Risk Planning
4. Software Risk Monitoring

These Processes are defined below.

Software Risk Identification

In order to identify the risks that your project may be subjected to, it is important to first study the problems faced by previous projects. Study the project plan properly and check for all the possible areas that are vulnerable to some or the other type of risks. The best ways of analyzing a project plan is by converting it to a flowchart and examine all essential areas. It is important to conduct few brainstorming sessions to identify the known unknowns that can affect the project. Any decision taken related to technical, operational, political, legal, social, internal or external factors should be evaluated properly.



Software Risk Identification

In this phase of Risk management you have to define processes that are important for risk identification. All the details of the risk such as unique Id, date on which it was identified, description and so on should be clearly mentioned.

Software Risk Analysis

Software Risk analysis is a very important aspect of risk management. In this phase the risk is identified and then categorized. After the categorization of risk, the level, likelihood (percentage) and impact of the risk is analyzed. Likelihood is defined in percentage after examining what are the chances of risk to occur due to various technical conditions. These technical conditions can be:

1. Complexity of the technology
2. Technical knowledge possessed by the testing team
3. Conflicts within the team
4. Teams being distributed over a large geographical area
5. Usage of poor quality testing tools

With impact we mean the consequence of a risk in case it happens. It is important to know about the impact because it is necessary to know how a business can get affected:

1. What will be the loss to the customer
2. How would the business suffer
3. Loss of reputation or harm to society
4. Monetary losses
5. Legal actions against the company
6. Cancellation of business license

Level of risk is identified with the help of:

Qualitative Risk Analysis: Here you define risk as:

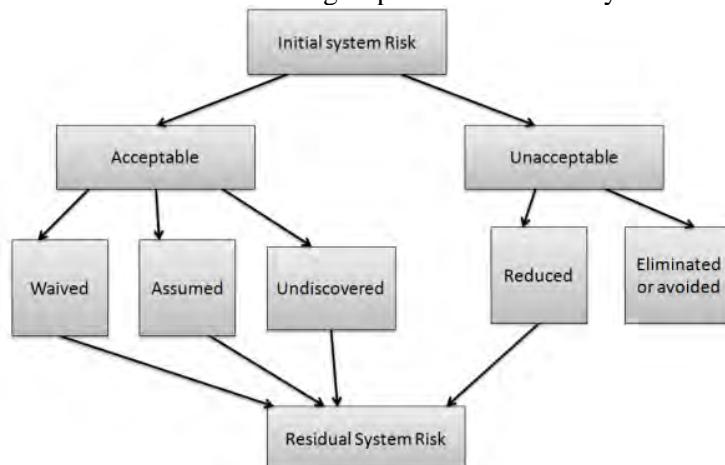
- High
- Low
- Medium

Quantitative Risk Analysis: can be used for software risk analysis but is considered inappropriate because risk level is defined in % which does not give a very clear picture.

Software Risk Planning

Software risk planning is all about:

1. Defining preventive measure that would lower down the likelihood or probability of various risks.
2. Define measures that would reduce the impact in case a risk happens.
3. Constant monitoring of processes to identify risks as early as possible.



Software Risk Planning

Software Risk Monitoring

Software risk monitoring is integrated into project activities and regular checks are conducted on top risks. Software risk monitoring comprises of:

- Tracking of risk plans for any major changes in actual plan, attribute, etc.
- Preparation of status reports for project management.
- Review risks and risks whose impact or likelihood has reached the lowest possible level should be closed.
- Regularly search for new

Planning the Risk Response

Risk Response planning can start after risk identification, qualitative risk analysis, or quantitative risk analysis. Central task in risk response planning is to develop actions and options to meet the following two goals

- Minimize threats to meeting project objectives
- Maximize opportunities

Three kinds of strategies available to handle three kinds of scenarios:

- Strategies to respond to negative risks when actions is required
- Strategies to respond to positives risks when actions is required
- Strategies that can be used to respond to both negative and positive risks when no action or a conditional action is taken

Planning the Risk Response: Response Strategies for Threats

Three strategies: Avoid, Transfer, Mitigate

- **Avoid:** This can be accomplished in various ways, including
 - Obtaining information and clarifying requirements for risks based on misunderstanding or miscommunication. This answers two question: Do we really have this risk and if yes, how can we avoid it?
 - Acquiring expertise for risks that exist due to a lack of expertise
 - Isolating the project objectives from the risk whenever possible
 - Relaxing the objective that is under threat, such as extending the project schedule
- **Transfer:** Risk transfer means you shift to the risk (the ownership of the risk), the negative impact of the risk or both. Note that transferring the risk transfers the responsibility for risk management and does not necessarily eliminate the risk.
 - Risk transfer almost always involves making payment of a risk premium to the party to which the risk has been transferred.
 - Planning the Risk Response: Response Strategies for Threats
- **Mitigate:** Mitigation in general means taking action to reduce or prevent the impact of a disaster that is expected to occur.
 - Risk Mitigation means reducing the probability of risk occurrence, reducing the impact of the risk if it does occur, or both
 - A good mitigation strategy is to take actions early onto first reduce the probability of the risk happening, and then to plan for reducing its impact if it does occur, rather than letting it occur and they trying to reduce the impact or repair the damage

Following are some examples of mitigation:

- Adopting less complex process
- Conducting more tests on the product or service of the project
- Choosing a more stable supplier for the project supplies

- Designing redundancy into a system so that if one part fails, the redundant part takes over and the system keeps working

Software Project Management Tools

- **PLANNING TOOLS:** There are several very popular planning tools available for project management planning. Microsoft offers a popular product called Microsoft Project. There is a similar program available that is offered for free. It is Niku's Open Workbench. Primavera offers Primavera Project Planner and Inter Plan systems offers ATC Professional for use in project planning.
- **ESTIMATIONTOOLS:** Estimation is an important aspect of project management. Like planning, there are numerous tools available to help with estimation. The most important tool for estimation is experience, which cannot be bought or installed. With that being said, some of the tools available to help with estimation are eTask Maker by Inter Plan Systems, SCE Plus by Scheme Quest, Tassc: Estimator by Tassc Limited, and SLIM-Estimate by QSM, Inc.
- **RISKMANAGEMENTTOOLS:** Risk management is extremely important in software project management. Dr. Dannelly stated in class that it is easier and cheaper to detect and fix the errors at the beginning of the project, than to find them at the end of the project. Detecting risks and planning for them is extremely important in planning for costs. Some of the tools to help with this include Risk Radar by ICE, Inc., Risk Trak, and CORA (Cost-Of-Risk-Analysis) by International Security Technology, Inc.
- **RESOURCEMANAGEMENTTOOLS:** Resource management is important to project management in many areas and tasks. Managing personal between several projects and tasks is vital in having a productive and profitable project plan. Some of the tools available that help out with resource management are Frog Point, Tracker Office and Time Scope.
- **CHANGEMANAGEMENTTOOLS:** Change management tools are important in keeping track of versions and changes in projects relating to software development. Managers must keep track of components and milestone changes. Two change management tools that help keep track of these changes are Track-IT Pro Suite by DOVICO Inc., and Work2gether by KM Technologies, Inc.