BACHELOR OF COMPUTER APPLICATIONS

SEMESTER 6

DCA3245

SOFTWARE PROJECT MANAGEMENT

Unit 1

Software Development Organization and Roles

Table of Contents

1. INTRODUCTION

Dear student, in software engineering you learnt to develop software using Software Development Life Cycle. When you are working as a team for developing a huge project, you need to follow several guidelines. This self-learning material will help you to achieve efficiency in implementing a Software Project.

The software development organization specializes in software project implementation through its high-level process model. This process model is designed to regulate the functionality and interactions between different people participating in the project in various roles. The important roles we need to consider here are the developers, testers, managers, and customers and their interactions in a software engineering process system. For every software organization, the key element is people. People are tuned in to develop various products. Products require life cycles for their development.

The current unit discusses the management spectrum, organizational structures and job roles in organizational structures.

1.1 Objectives:

After studying this unit, you should be able to:

- *Explain the management spectrum.*
- *Describe the organizational structures.*
- *Describe job roles in software development.*

2. THE MANAGEMENT SPECTRUM

The management spectrum begins before any technical activity is initiated and continues throughout the rest of the phases of software development.

Software Project Management deals with:

Organizing, planning, and scheduling software projects.

Involved with activities needed to ensure that software is:

- It should be delivered on time.
- The software proposed should be within the budget constraints.
- The software should aim to comply with the given requirements.

The management activities involve the following process:

1. Proposal writing
2. Project planning and scheduling
3. Project costing
4. Project monitoring and reviews
5. Personnel selection and evaluation
6. Report writing and presentations

For effective software project management, we need to focus on the four P's - People, Process, Product, and Project.

The People:

The people factor is very important in software project management. Hence the Software Engineering Institute (SEI) has developed a People Management Capability Maturity Model (PM-CMM). The PMM Model defines the following key practice areas for software people.

- Recruiting
- Selection
- Performance Management
- Training
- Compensation
- Career Development
- Organization and Work Design

- Team Culture Development

People are an essential element for any organization. The growth of the organization directly depends on the people's capability. They are considered the Intellectual Capital of the organization.

The Product:

Before planning a project, we need to ensure that the objectives and scope of the product being developed are established properly. Along with the objective and the scope, we need to also ensure that alternate solutions for the product implementation are considered, and the technical and management constraints are considered. Without these details, it will be difficult to reasonably estimate the effort and analyze the risk involved with the Product. The software developer and the customer must meet to discuss and define the product objectives and scope. Objectives identify the overall goals of the project. At this point, we are not considering how these goals will be achieved, but only focusing on documenting the intended goals. The scope of the product identifies the features, primary data, and behaviors that characterize the product. Once we have the scope and the objectives finalized, we can consider alternate solutions for product implementation.

The Process:

The Process refers to the set of guidelines and tasks that can be used as a framework for carrying out a software project. As part of the process model, different software projects will have different types of activities to be carried out based on their needs and some activities will be common across all the projects. Also, there are some activities termed umbrella activities that overlay the process model. Quality assurance, Configuration management, and Metrics measurement are typical examples of umbrella activities, and these are independent of the framework activity and need to be carried out throughout the software process. The Software process does not address the technical solution needed for the implementation of the software product. The software development process defines a set of frameworks to address the difficulties and common problems faced during the implementation of software development. These processes help the managers to handle product development to address business needs.

The players

The software process involves following key players based on the role they play in the software project.

- **Senior Managers:** They define the business issues that often have a significant influence on the project.
- **Project Manager:** They must plan, motivate, organize, and control the practitioners who do software work.
- **Customers:** They specify the requirements for software to be engineered. They are also referred to as stakeholders.
- **End-Users**: They interact with the software once it is released for production.

Practitioners: Practitioners take care of the design, coding, testing, and installation of the software product, in other words, practitioners provide the skill to implement the software product.

The Project

A Project is a collection of tasks handled in a planned and systematic order. For smaller projects with a team of one or two working for a few weeks, the tasks of the project can be executed informally. This means a detailed procedural approach need not be followed; instead, for example, the requirement can be communicated verbally or through a note. However, for larger projects where many people work in the same team, these informal techniques will not help as communication and tracking of the project tasks become very difficult. The project tasks must be carefully planned and allocated to the right resources and tracked throughout the project execution.

The problems in software projects come from three different viewpoints. There are project managers, developers, and customers. Project managers will face problems if they are poor in defining roles, estimating, planning and decision-making. Project managers do need to face schedule, budget, and quality constraints. However, the problems faced by developers are if he/she has a lack of knowledge in the application area, lack of standards, lack of up-to-date documentation, deadline pressure, and changes in application requirements. Lastly, the problems faced by customers are the monetary constraints, receiving the products out of the actual time that should be delivered.

SELF-ASSESSMENT QUESTIONS – 1

1. Effective software project management focuses on four P's – People, Process, Product, and Project. (True / False)
2. People Management Capability Maturity Model (PM-CMM) has been developed by_____.
3. _____must plan, motivate, organize, and control the practitioners who do software work.
   a) Project Managers
   b) Senior Managers
   c) Customers
   d) End Users

Quality that a Team Leader should have:

The most important role in a software project is not of the system or the tools but it is of the people, including from manager to developer, from customer to end user. So, let us see the role of people in detail which contributes to the role of the management spectrum.

1. Motivation – the ability to encourage technical people to produce to their best ability, knowing who on your team is most capable of tackling a certain project or task and giving them the authority to do so is an essential part of being an effective leader. Team members who are given opportunities to demonstrate their abilities while also gaining valuable experience are more likely to take pride in their work because of delegation.
2. Organization – the ability to design new processes or mould existing processes that enable the initial concept to be translated into a final product. The best leaders teach by doing. If you act as an example of the kind of leader you want your team to follow, they will not only respect you more for it, but they will also be more likely to follow your lead. Someone competent shows confidence and positivism, accepts accountability and shows respect for others is likely to be looked up to.
3. Ideas or innovation: Capacity to motivate people to think creatively within the constraints of a certain software product or application.

4. Problem-solving skills: Determine the problem, map out a remedy, make use of what you've learnt, and maintain adaptability. It also involves critical thinking, deliberation, and introspection are all required components. In addition, effective leaders get their subordinates involved in problem-solving by seeking and analyzing ideas from them to better understand the issue at hand and come up with appropriate solutions.
5. Managerial identity – Be in charge of the project, take control with confidence, and have belief in good people to let them do their jobs. Managers should have important Empathy, Managers must care for and consider the well-being of their employees, and they should follow and respect Work ethic, Management requires a strong work ethic and the willingness to go the additional mile for the team. Managers should care for and consider the well-being of their employees. The person in charge of the team needs to be truthful, fair, and inspiring, and on top of all that, he needs to be flexible so that we may fit into the category of more broad identities.
6. Achievement – He should be rewarded with initiative and demonstrate that controlled risk-taking will not be punished.
7. Influence and team building – Team leaders should be able to understand people, verbal, and nonverbal signals, and be able to react to signals.
8. Team Leaders should remain under control in high-stress situations.

The software team also should follow the following parameters:

Constructing a software development team is a very critical and tricky task.

Seven project factors to consider when structuring a software development team.

1. The difficulty of the problem to be solved.
2. The size of the resultant program(s) in source lines of code
3. The time that the team will stay together.
4. The degree to which the problem can be modularized.
5. The required quality and reliability of the system to be built.
6. The rigidity of the delivery date
7. The degree of sociability (communication) required for the project.

These factors make the team environment very volatile and unpredictable, resulting in below-standard software products.

Here are five things that can make a team less productive. Or the factors that cause a toxic team environment:

1. A chaotic work atmosphere: Having no way to make people responsible for their actions. Accountability is essential in any workplace, but it's especially challenging in an unorganized environment. Setting reasonable standards is the cornerstone of accountability. When this is complete, the next step is to ensure everyone sticks to their end of the bargain.
2. High frustration causing conflicts among team members: When two people can't seem to agree on anything and display open hostility towards one another, they are engaging in emotional conflict.
3. A fragmented or poorly coordinated software process leads to so much confusion.
4. An ambiguous definition of roles on the software team: leading the entire team in the wrong direction.
5. Continuous and repeated exposure to failure: Managers rarely take the rap for an employee's failure or subpar performance. Management can say this worker just doesn't get it. Or then the worker lacks ambition, is disorganized, and refuses to take direction. The employee is always held responsible for any issues that arise on the job, regardless of the cause.

The solution to these problems:

The Team should be encouraged to review its own procedures and working model periodically. Let us see How to avoid these problems.

- The team should have access to all information required for doing the job.
- Once major goals and objectives are defined, should not be modified, unless necessary.
- Delegate the responsibility for decision-making to the team as much as possible.
- The team should be able to recommend its process model.
- The team should be able to establish its procedures for accountability.
- It should be possible to establish team-based techniques for feedback and problem-solving.

The team should be encouraged to review its own procedures and working model periodically.

Key characteristics of modern software that make projects fail are scale, uncertainty, and interoperability like not having a clear understanding of the business needs, not involving end customers in the early stage of development, selecting the wrong team and technology, or not delivering on time, poor planning, and missing accountability all these factors might be the reasons for project fail.

To better ensure success in team building we must Establish effective methods for coordinating the people who do the work and Establish methods of formal and informal communication among team members.

Create reliable means of coordinating the individuals doing the work and set up channels of official and informal communication within the team.

The final stage looks at the team from the perspective of the well-being of the team instead of the perspective of handling a team through the original four stages of team growth.

- A team cannot be expected to perform well right from the time it is formed.
- Forming a team is just like maintaining a relationship.
- It takes time, patience, requires support, efforts and members often go through recognizable stages as they change from being a collection of strangers to a united group with common goals.
- There is a typical progression via which members evolve from being an assortment of strangers to a cohesive unit working towards a similar goal.

Bruce Tuckman presented a model of five stages:

These stages are commonly known as Forming, Storming, Norming, Performing, and Adjourning to develop as a group.

Bruce Tuckman proposed a five-step paradigm, which he called. The stages a group goes through as it matures are the "Forming," "Storming," "Norming," "Performing," and "Adjourning" phases. According to Tuckman's model, when a team grows in skill and experience, relationships form, and the leadership style shifts to one of shared responsibility, the team becomes more mature and effective.

Tuckman's early work only detailed the way he had seen groups develop, whether or not they were aware of it at the time. Identifying a team's current developmental stage and helping it progress into one that is more appropriate for the level of cooperation demonstrated is where CORAL shines. As teams form and reform in the real world, they may progress through the Tuckman Stages as they go. Even if a group is thriving in Norming or Performing, it can regress to Storming if a new member joins or if a team member repeatedly misses meetings. The team will be able to come back to Performing status as soon as possible thanks to the preparedness of the project guides.

The group Dynamics Based on studies was published by B. Tuckman in 1965 and later it got updated in 1977, Let us discuss in detail the different stages involved in the model and its importance:

1. **The first stage is called forming:** which means a High degree of guidance is needed from the manager, here the Individual roles are unclear and also the Process is usually not well established, The first step in building a team is to establish its foundation. Team members experience uncertainty and avoid disagreement at all costs to gain social acceptance. The members of a team depend on their leader, typically CORAL project guides, for advice and assistance.
2. **Storming:** Understanding how team decisions are made. The storming phase is the most trying and crucial one to go through. It's a time of strife and rivalry as distinct identities take shape. At this point, the team's performance may even suffer as members focus their efforts on meaningless pursuits. Team members may have conflicting ideas about their collective purpose, and smaller groups may form around influential members or common ground. To progress past this phase, the team must overcome challenges, accept differences, and resolve disagreements over how to accomplish their goals. This is a potentially problematic phase for teams. Long-term issues may develop if conflicts aren't resolved.
3. **Norming:** Relationships are well understood in the team, and Commitment to team goals and work to optimize team process, once a team moves through the "storming" phase, they can begin to work through their differences and come together. During the norming phase, group members settle on who will lead and what their specific responsibilities will be. A sense of oneness and togetherness grows when interpersonal

tensions subside. During this phase, team members learn to work together and start to prioritize the team's objectives, leading to improved performance. However, the peace is fragile, and if problems arise again, the team might easily revert to storming.

4. **Performing:** The team is committed to performing well, it should focus on being strategic. At this point, the team has reached a level of maturity, organization, and effectiveness where they can work together effectively and achieve common goals. The team has a well-defined hierarchy, and everyone is invested in its goals. There are still challenges and disagreements, but they are managed well. Conflict and its resolution will be addressed in the next section. As a unit, they are committed to finding solutions and achieving their objectives.
5. **Closure**: At this stage, the project is coming to an end and the team members are moving off in different directions. Most of the team's objectives have been met at this point, and they can adjourn. The focus is on completing any remaining chores and recording any relevant data. The team may disintegrate as its members are transferred to other groups in response to a decrease in workload. A formal ceremony celebrating the team's accomplishments will help allay any feelings of sadness that may be associated with its dissolution. Teams that have continuous responsibilities and in which members are subject to replacement can restart the development process by returning to the forming or storming phase.

Factors influencing the meaningful indication of the progress of a Product:

Proper identification of the following factors provides a meaningful indication of progress.

1. an effective assessment of risk,
2. a realistic breakdown of project tasks,
3. a manageable project schedule

Before planning a project, the scope and objectives of the product should be properly established, identification of technical and management constraints should be done, and finally, alternative solutions should be considered.

It is impossible to define accurate and reasonable estimates of the cost without the knowledge of the product objectives and constraints.

The software developer and customer should sit together to define product objectives and scope.

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

The product is the goal of the project. This is any type of project that has to be developed.

- Objectives identify the overall goals for the product from the viewpoint of the customer without considering mechanisms for achieving the goals.
- Scope defines the primary data, functions and behaviors that characterize the product, and outlines attempts to bind these characteristics quantitatively.
- Once the product objectives and scope are understood, alternative solutions are considered.
- The proposed alternative solutions help managers and practitioners to select a "best" approach, under given constraints imposed by delivery deadlines, budgetary restrictions, personnel availability, technical interfaces, and various other factors.

Product Importance and its Scope:

Software project scope must be unambiguous and understandable at both the managerial and technical levels. The scope of software development must be established and bounded.

Context – How does the software to be built fit into a larger system, product, or business context, and what constraints are imposed because of the context? How does the software be developed interact with other parts of the system, product, or business, and what limitations does this interaction impose?

Information objectives – Define customer-visible data objects that will be produced as output from the software. What will be the input data objects required for producing the output?

Function and performance – Determine the functions performed by the software to convert input data into output. Client and vendor can share the same knowledge of the project's requirements thanks to the Functional Performance Specification document, which in turn opens the door to more creative ways of meeting those requirements. Function

performance is a method for outlining the features and specifications of a project, product, or service about its essential Functions.

Special Performance: it deals with checking if is there any need to address any special performance characteristics, it also Thinks about each role and determines what is required to fulfil that role well. establish the standards that will be applied to determine if the function has been fulfilled. (Many can serve the same purpose)

Management spectrum involved in selecting a Process:

Project manager must select the most appropriate process model based on

1. Customers who have ordered the product and the people who will produce the product
2. Product characteristics
3. Environment of the project in which the software team works

Steps involved in overlapping the process:

1. After selecting the process model, a preliminary project plan is defined depending on the process framework activities
2. Process decomposition starts
3. Activity results in a complete plan that specifies the tasks needed to populate the framework activities
4. Software process defines the framework from which an exhaustive plan for software development can be specified
5. Number of different tasks sets tasks, milestones, work products, and quality assurance points
6. Number of different tasks sets tasks, milestones, work products, and quality assurance points
7. Finally, the process model gets overlapped by umbrella activities

Management Spectrum-People:

After understanding about the project now Let us put few things together about the people management in order to avoid failures and rectify the causes of project failure from the side of people management

- Start on the right foot: We have to take care on selection process, carefully choose the right project
- Understand the problem: project requirements should be analysed well so as to make proper understanding of the problem
- set realistic objectives and expectations;
- form a good team who can real work on it in an efficient manner rather than taking irrelevant resources

Maintain momentum: set a target for every work to be carried out and perform as per the deadline and try to meet the target works in the proposed flow. Turnover in employees may be quite costly to businesses. As a result, businesses may learn the true cost of staff turnover by doing the maths. In addition, it can assist businesses in identifying weaknesses and openings for development, which in turn can boost output and expansion.

Provide incentives to reduce turnover of people; emphasize quality in every task; have senior management stay out of the team's way. Nowadays, workers want more from their employers than just a paycheck. They would rather work in a stimulating setting where they may advance professionally.

Management Spectrum – Project:

Project encompasses the entire software development process from requirements gathering through final delivery and ongoing maintenance and updates. Project manager of a project is accountable for managing the people, product and process.

- To avoid failure of the project the responsibilities or activities of software project manager should be properly followed
- Software project may be extremely complex and according to the industry data the failure rate is high
- Main reason should be the development process but mostly failure happens due to the activities performed before development and sometimes due to less maintenance

Project:

The term "project" refers to the entire software development process, from initial requirements gathering through final modifications. A project manager's duties span the

management of the project's human, physical, and technological resources. The software project manager's duties and procedures must be strictly adhered to in order to guarantee success.

A project is a short-term endeavour with a defined purpose that aims to produce something new. Projects are different from regular or continuous operations since they have a beginning and a finish, a certain scope, and a specific set of activities to carry out in order to reach a specific goal.

- Software project may be extremely complex and according to the industry data the failure rate is high
- Main reason should be the development process but mostly failure happens due to the activities performed before development and sometimes due to less maintenance

Factors to be taken into consideration while making the project to be an ideal one :

- Before writing new code, one should use commercial off-the-shelf or existing software
- Follow standard approaches
- Identify and avoid risks
- Always allocate more time than you think you need to do complex or risky tasks
- Lessons learned for each project must be tracked and planned schedules must be compared
- Collect and analyse software project metrics
- Take feedback from teams members and customers
- Record findings in written form

Reasons for Project Failures:

1. Business needs changes frequently (or the requirements are poorly defined)
2. Deadlines are unrealistic
3. Users are resistant
4. Sponsorship is lost (or was never properly obtained)
5. Software people don’t understand their customer's needs
6. Product scope is poorly defined
7. Changes are managed poorly

8. Chosen technology changes

Certainly, here are some key characteristics of a project that has to be answered before taking up an projects:

Questionnaire used to define the scope and objectives of a project and create a project plan.

1. Why is the system been developed : Assesses the validity of business reasons and justifications
2. What will be done?: Establishes the task set required for the project
3. When will it be done?: defines the deadline of the project and then establish a project schedule
4. Find out Who is responsible for a function then define the role and responsibility of each team member also track where are they organizationally located and note the organizational location of team members, customers, and other stakeholders
5. Where are they organizationally located? Tells clearly about the organizational location of team members, customers, and other stakeholdersUsers, customers, and stakeholders, not only software developers, also play an important part in this from a business perspective. Knowing the project's scope allows for the definition of all technical strategies and management regulations.
6. How will the job be done technically and managerially ? This establishes the management and technical strategy for the project, After estimating each resource based on the requirements of customers and users, software developers are able to determine this. This is where you determine which team member takes on which responsibilities. You may also identify external stakeholders with a claim in the project. a strategy for developing the software and managing the project is concluded upon.
7. How much of each resource is needed? The goal of this step is to figure out the number of resources necessary to complete the project. This Establishes estimates based on the answers to the previous questions,

The goals of the project are honed during the planning phase once initial input has been obtained. It entails figuring out how you're going to achieve those goals, down to the exact tasks and materials you'll need. Now that these goals have been identified, they need to be spelt out in full, with careful consideration given to every one. Our perception of the goal may

shift as a result of this analysis. Attempting a precise description of an object often leads to a deeper appreciation of what we're looking

3. ORGANIZATIONAL STRUCTURE

Organizational structure depends on the product to be developed. Two extremes of organizations are project organizations and functional organizations. In functional organizations, the departments are organized based on technical disciplines whereas the project organizations are based on the product and project. Wheelwright and Clark define a continuum of organizational structures between two extremes, functional organizations and project organizations. Let's look into these different types of organizational structures.

Each organizational structure has its pros and cons. The trick is to find a form of organizational structure in which a business gets the most benefit from the pros and suffers least from the cons. A company's structure describes the channels via which it channels its efforts to realise its objectives.The roles of all employees in a well-structured organization are clearly defined and their responsibilities are well articulated.

- Every software development organization manages multiple projects at any time.
- Software organizations allocate separate teams of engineers to handle different software projects.
- The organizational structure depicts the responsibilities and define the reporting and communications lines that bind everything together.

The reporting structure in an organization will be an important factor in deciding how much influence the Project Manager will have within the organization.

- The role of Project Managers will be affected by the type of company's organizational structure and how resources are allocated to the project.

Reporting structure:

Reporting structure in an organisation will be an important factor in deciding how much influence the project manager will have within the organisation. The business's aims, the industry in which it operates, and the company's culture are all important considerations for top management when determining the optimal organizational structure.

Need to formalise the organisational structure even if it Is a relatively small business, Having a framework in place can boost productivity and make things clearer for everyone involved. Every division can increase output because of a heightened awareness of the value of conserving resources.

While in a centralised system, employees know their place in the hierarchy, those in a decentralised structure have more autonomy. Organizational structures can be categorised as either "functional," "divisional," "flatarchy," or "matrix."

Organisational structure depicts the responsibilities and define the reporting and communications lines that bind everything together for reporting structure

Fuzzy Organisational Structure:

If organizational structure is not clear, then w call such type of structure to be fuzzy , then there are several consequences for a business.

- Some of the things that can go wrong if the organizational structure is fuzzy:
- Everyone thinks someone else is responsible, as a result nothing happens .
- It'll get duplicated because everyone thinks it's part of their job.
- People are going to get mad and frustrated because they don't know what to do or are treading on each other's toes.
- Information fails to flow via the most efficient channel or reach the right people.
- The business can't do a single thing without referring to top management who then slowly go crazy because of the workload.
- Reduced output is one effect of neglecting your company's organisational structure. Employees are more inclined to delay completing a task when they receive conflicting instructions from multiple managers.

Types of Organizational Structure:

dentifying, defining and communicating roles and responsibilities is fundamental to a team's ability to deliver.

Depending on the working environment of the organization, their set goals and the nature of their work, the organizations are structured in three ways:

1. Functional Organizational Structure

2. Project Based Organization Structure
3. Matrix Organizational Structure

1. **Functional Organization Structure:** As mentioned earlier **functional organizations** are set up according to technological streams. Resource allocation will be handled by senior functional managers. Coordination among the resources occurs through rules and procedures, detailed specifications, shared traditions among engineers and meetings (ad-hoc and structured). Thus the responsibility for the total product is not allocated to a single person. Products that need a high level of specialized knowledge require a functionally organized structure.

2. **Project Based Organisational Structure:**

Project Based Organizational:

- In the project based organizational structure, the project development staff are divided depending on the project for which they work.
- In this format, a group of engineers are allocated to the project at the beginning of the project and that they remain associated with the project until the completion of the project.
- Thus, the identical team carries out all the life cycle activities.

Project manager has complete project authority. He has jurisdiction over the project's budget, schedule, and the project team

- There is a clear, properly defined line of authority that results in faster decision-making and approval.
- Communication becomes more effective and simple and project team members gain more experience worki authority would fall to the hands of a functional manager as it is in a functional organisation, ng on different types of projects as the need for them arises.

3. **Matrix Organisational Structure**

The matrix organizational structure can be found lying somewhere between the functional organizational structure and the projectized organizational structure depending on what type of matrix structure is being run.Matrix organizations merge features of project-based and functional organizational structures.

A major challenge with a matrix organization is that every employee has two (or more) managers they report to, their Functional Manager and the Project Manager. If they are assigned multiple projects, they may even have to report to more managers.

The matrix organizational structure can be found lying somewhere between the functional organizational structure and the projectized organizational structure depending on what type of matrix structure is being run.

- There are three types of matrix organizations:
  1. Weak Matrix
  2. Balanced Matrix
  3. Strong Matrix

1. **Weak Matrix**: In weak matrix structure, the project If the organization is running a weak matrix structure, then the project authority would fall to the hands of a functional manager – as it is in a functional organization. Interestingly enough, in a balanced matrix organization, both the project manager and the functional manager shares equal authority for the project.It promotes greater efficiency, helping the organization respond to customer demands or changes in the marketplace, faster.
2. **Balanced Matrix:** In a balanced matrix organisation, both the project manager and the functional manager shares equal authority for the project. Balanced version is best if an organisation finds itself working in a dynamic environment romotes greater efficiency, helping the organisation respond to customer demands or changes in the marketplace, faster
3. **Strong Matrix:** In project management, a robust matrix is a frequent organisational framework. One defining feature is the emphasis placed on internal projects. Project managers in a robust matrix structure have extensive autonomy over their initiatives, and project work is prioritised over departmental or functional priorities. A strong matrix is a type of organizational structure that is commonly used in project management. It is characterized by a significant emphasis on projects within the organization. In a strong matrix structure, project managers have considerable authority and control over the projects they are responsible for, and project work takes precedence over functional or departmental activities.

SELF-ASSESSMENT QUESTIONS – 2

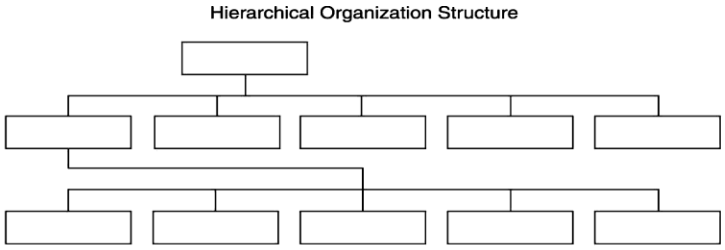
4. Organization structure depends on the people working in it. (True / False)
5. Wheelwright and Clark define a continuum of organizational structures between two extremes,_____ organizations, and organizations.
6. The realistic percentage for striving to work in Software Project Development is.
   a) 40%
   b) 80%
   c) 50%
   d) 100%

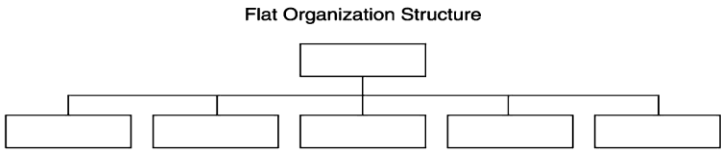
4. TYPES OF ORGANIZATIONAL STRUCTURES

The organizational structure corresponds to the flow of information and reporting structure within an organization. In the traditional approach, the design of the organization structure can be classified as hierarchical, flat, or matrix.

- The hierarchical organization follows a top-down approach wherein the middle manager communicates with the subordinates about the work to be completed and reports to superiors the outcome of the work.
- The flat structured organization is a flexible approach towards work allocation, employees do whatever is needed. This structure easily supports offloading of extra work and is better suited for intra-company communications.
- In the matrix organization, work is broken into small workgroups that can be managed and integrated at the regional/national level. It is aimed at reducing operational complexities and expenses by effective sharing of information among different managerial groups.

Other than these three structures, networked organizational structure and T-form organizational structure are generally considered.





Matrix Organization Structure

Industry \ Regions	Region 1	Region 2	Region 3
Industry 1	Position for industry 1 in region 1	Position for industry 1 in region 2	Position for industry 1 in region 3
Industry 2	Position for industry 2 in region 1	Position for industry 2 in region 2	Position for industry 2 in region 3
Industry 3	Position for industry 3 in region 1	Position for industry 3 in region 2	Position for industry 3 in region 3

Fig. 1.1: Types of Organizational Structures

4.1 Hierarchical Organizational Structure

The key features of Hierarchical Organization Structure are given below:

- The work allocation is based on the concepts of division of labor, specialization, and unity of command.
- The decisions made by top management will flow through the organization.
- Middle-level managers are primarily responsible for work delegation in the form of information processing and communication. This organizational structure typically is used to store and communicate information along the top-down flow of hierarchy and to support the information management function of the managers.

4.2 Flat Organizational Structure

The following are the key features of Flat Organizational Structure:

- Decision-making is centralized.
- Since the employee does whatever needs to be done, they can respond quickly to dynamic, uncertain environments.

- However, a flat organizational structure often becomes less flexible as the organization grows. Routine work is often off-loaded but, as a hierarchy develops, this routine work becomes the 'glue' that acts as the means of tying other parts organization for effective communication. Missing routine work creates trouble in the entire structure.

4.3 Matrix Organizational Structure

The features of the Matrix Organization Structure are listed below:

- In Matrix structure for each worker typically two or more supervisors are assigned. This is done to ensure that multiple dimensions of the business are integrated into the product being worked on as each supervisor looks for the respective aspect of the employee's work based on business needs.
- Due to the multi-dimensional structure the information processing becomes more hence often using the matrix organizations managers fail to achieve business strategies.

4.4 Networked Organizational Structure

The networked organization structure is shown in Fig 1.2. The following are the advantages of a Networked Organizational Structure:

- As depicted in the diagram, we can see that the rigid hierarchies are replaced by formal and informal communication networks to connect all parts of the company.
- Networked structure promotes creativity and flexibility in its structure while maintaining operational process control. This control is achieved by substituting hierarchical control over the controls based on IS.

- Coordination across functional boundaries becomes easier as networked architecture makes extensive use of communication technologies and networks.

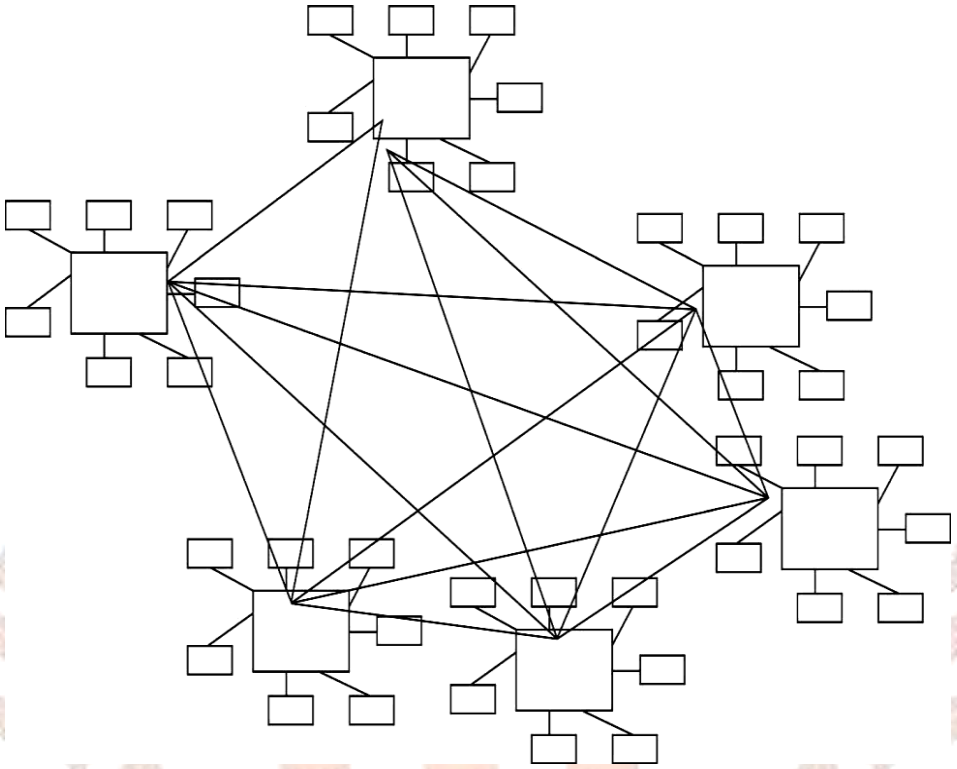


Fig. 1.2: Networked Organizational Structure

4.5 T-form Organization

Features of T-form Organization are as follows:

- The T-form (Technology-based) organizations are an improved version of network structured organization. They combine the traditional components of network structure with IT to make a new structure of technology-based components.
- The electronic medium is used for communication in the form of electronic linking, production automation, electronic workflows, electronic customer/supplier relationships and self-service Internet portals.
- Work is often coordinated electronically, while systems enable information to move around the organization more easily, decentralizing decision-making.

SELF-ASSESSMENT QUESTIONS – 3

7. In flat-structured organizations, work is more flexible, and employees do whatever is needed. (True / False)
8. In_____ organizations, work is organized into small work groups and integrated regionally and nationally/globally.
9. _____Organizations take the networked structure one step further by combining IT with traditional components to form new types of components.
   a) Hierarchical
   b) Flat
   c) T-form (Technology-based)
   d) Matrix

5. JOB ROLES IN SOFTWARE DEVELOPMENT

The following are the roles based on the Microsoft Solutions Framework (MSF).

- **Middle-Management Leadership**

Manage people, resources, and budgets. Oversee and provide a vision for several major projects simultaneously. Review employees in all other positions. Work with other MM leaders for interaction between projects.

- **Team Leadership**

Manage people, resources, and timelines for one major project or several minor projects. Act as the central point of contact on those projects. Involve or be aware of virtually every issue or decision in the project. Team Leader is responsible for all aspects of the project. Work in all other positions.

- **Product Management**

Work with clients to define requirements and resolve issues. Design and maintain functional specifications and other documentation. Often provide prototypes for user interfaces or design interfaces of services. Work with Team Leadership and Software Development.

- **Logistics**

Manage hardware/software requirements for development, testing, validation, and production environments. Perform or oversee installations. Own the installation process and any installation utilities. Work with resource teams to obtain servers/software and address issues within the environments. Work with Team Leader.

- **Software Development (Programming)**

Design and code the software to match the specifications, prototypes, and other documentation. Define timelines. Work with Product Management to refine expectations and clarify requirements. Often interact with Team Leader, Tester, User Documentation, and User Education.

- **Software Testing**

Define testing procedures and certification process. Define timelines. Create and execute tests on software. Manage a bug-tracking procedure. Work with Team Leadership. Collaborate with Product Management to define areas and specifics of testing. Often interact with Software Developer. Work with Team Leader.

- **User Documentation**

Create and maintain user-centric documentation. Work with Product Management and Software Development to define and document functionality. Often provide training materials for User Education. Work with Team Leader.

- **User Education**

Create training procedures and policies. Design training materials. Execute training sessions. Work with Product Management and User Documentation. Work with Team Leader.

- **Software Support**

Define support procedures. Handle user issues. Provide resolutions or formulate a workaround for software issues. Forward hardware/ infrastructure issues to Logistics. Notify Software Testing and Development of software bugs. Work with Team Leader.

Team Structure:

There are several other variations to these Team structures. Generally the Team structure addresses the issue of organization of the individual project teams.

There are three formal ways in which the individual project teams can be organized, Problems of different complexities and sizes often require different team structures for chief solution:

1. Chief programmer,
2. Democratic, and
3. The mixed team organizations

- Chief Programmer Team consists of project manager , reporting and software Engineer. The role of chief programmer is to monitor the work and coordinate between the various teams and handle the smooth flow to work . A chief programmer team is a star-shaped programming group whose "chief" is the software engineer with the most comprehensive understanding of the system's goals and objectives. The rest of the team is used in a supporting capacity. The chief programmer team is subject to single point failure since too much responsibility and authority is assigned to the chief programmer

In this organization, a senior engineer delivers the technical leadership and is designated as the chief programmer. The chief programmer partitions the task into small activities and assigns these tasks or activities to the team members. The lead programmer checks the work of his or her colleagues and integrates it into the final output. The head programmer provides authority, and in well-defined problems, this structure is more effective than the democratic team. Team morale is poorer when the chief programmer is in charge, because everyone on the team has to answer to him or her all the time.

However, the chief programmer team results in lower team morale, since team-members work under the constant supervision of the chief programmer. A chief programmer team is a star-shaped programming group whose "chief" is the software engineer with the most comprehensive understanding of the system's goals and objectives. The rest of the team is used in a supporting capacity. The morale of a team suffers when its main programmer micromanages every aspect of the team's work.

- Democratic Team: The democratic team goal is determined by a vote of all its members, and everyone's opinion is considered when making major decisions. Leadership of the group passes from person to person. Ego-free teams are inherently democratic teams because of how they function. The design encourages participation from all

stakeholders, which can improve decision-making across a range of issues. This shows that this approach works best for time-bound research initiatives with a lengthy timeline. The above diagram indicates the process of democratic team were the circle indicates software engineer and the arrow maks indicates the communication path bethween the engineers.

For large sized projects, a pure democratic organization tends to become chaotic.

- The democratic team structure does not enforce any formal team hierarchy.
- Specifically, a manager provides the administrative leadership. At different times, different members of the group provide technical leadership.
- The democratic organization leads to higher morale and job satisfaction. The design encourages participation from all stakeholders, which can improve decision-making across a range of issues. This shows that this approach works best for time-bound research initiatives with a lengthy timeline.
- Democratic team structure is best suited for less understood problems, since a group of engineers can invent better solutions than a single individual as in a chief programmer team.
- A democratic team structure is suitable for projects requiring less than five or six engineers and for research-oriented projects. The democratic team organization encourages egoless programming as programmers can share and review one another's work.
- The mixed control team: A mixed control team organisation is a type of hybrid organisational structure that includes functional and project-based features. It is also known as a mixed authority or balanced matrix structure. In this structure, authority and responsibility are split between a functional manager and a project manager. The goal is to combine the strengths of both the project-based method and functional expertise. In a mixed control team organization, employees have dual reporting relationships. They report to a functional manager who oversees their day-to-day work and to a project manager who manages their work on specific projects. Functional managers are responsible for assigning resources, setting performance standards, and managing the technical and specialized aspects of employees' work. This ensures that the organization can leverage the expertise within functional departments.

The team organization is eminently suited to handle large and complex programs.The mixed team organization, derives ideas from both the democratic organization and the chief-programmer organization. The mixed control team organization is ideally suitable for large team sizes. The democratic arrangement at the senior engineers level is used to decompose the problem into small components. Project managers have authority over the project-related aspects of an employee's work, such as defining project tasks, schedules, priorities, and making project-specific decisions. Decision-making authority is shared between functional and project managers. Functional managers may influence decisions related to resource allocation and career development, while project managers have authority over project-specific decisions.

Management Spectrum:

Apart from the above discussed cases for the failure we also have few parameters which is directly or indirectly hinders the effect of the project development, The project is the complete software project that includes requirement analysis, development, delivery, maintenance and updates. It involves the following process as well

- ✓ Tracking the progress of the flow of the project
- ✓ Track the completion of work products;
- ✓ collect software process and project measures;
- ✓ Assess progress against expected averages
- ✓ Make smart decisions
- ✓ Keep it simple;

- Functional Organisational Structure:

Development staff are divided based on the functional group to which they belong.

Different projects borrow engineers from the required functional groups for specific phases to be undertaken in the project and return them to the functional group upon the completion of the phase

Functional Organization: The functional organization are set up according to technological streams. Resource allocation will be handled by senior functional managers. Development staff are divided based on the functional group to which they belong. Different projects

borrow engineers from the required functional groups for specific phases to be undertaken in the project and return them to the functional group upon the completion of the phase

- Functional Organizational Structure:

Completely separate groups of programmers perform different phases of a project.

Partially completed product is passed from one team to a different as the project evolves

For example, one team may do the necessities specification, another do the planning, and so on.

- ✓ In a functional organizational structure, the development staff are divided based on the functional group to which they belong.
- ✓ The different projects borrow engineers from the required functional groups for specific phases to be undertaken in the project and return them to the functional group upon the completion of the phase.
- ✓ In the functional format, completely separate groups of programmers perform different phases of a project.
- ✓ The partially completed product is passed from one team to a different as the project evolves.

- Functional Organisational Structure:

To be useful the scheme needs significant communication among the various groups.

Work of one team should be clearly understood by the next teams engaged on the project.

This necessarily requires good quality documentation to be made after each activity.

- ✓ To be useful the scheme needs significant communication among the various groups.
- ✓ The work of one team should be clearly understood by the next teams engaged on the project.
- ✓ This necessarily requires good quality documentation to be made after each activity.

The system can only be effective if all involved parties are able to effectively communicate with one another. It's important for subsequent teams working on a project to fully grasp the work completed by the previous team. Inevitably, this calls for thorough record-keeping following every action.

- Functional Organisational Structure:

Everyone knows how to hold accountable if something were to go wrong as responsibilities are predetermined. The top managers comes into picture in maintaining and managing , the requirements , design phase , coding section, testing sections and finally project management looks into the whole phases of development and maintenance . So we will look into the different phases of project life cycle in detail and how the top management deals in handling all these phases , the involvement of the different project teams in different phases can be seen and their coordination between the teams is very crucial to have a successful delivery of the product.

- Advantages of Functional Organisational Structure:

Even though greater communication among the team members may appear as an avoidable overhead, the functional format has many advantages

- ✓ Functional structure has several advantages. The main advantages of a functional organisation are:
- ✓ Ease of staffing
- ✓ Production of good quality documents
- ✓ Job specialization
- ✓ Efficient handles the problems concerning manpower turnover.
- Engineers become specialists in specific roles, like requirements analysis, design, coding, testing, maintenance, etc.
- Engineers perform these roles repeatedly for multiple projects and develop deep insights to their work

The staffing pattern should approximately follow the Rayleigh distribution for efficient utilization of the personnel by minimizing their wait times

- ✓ The engineers become specialists in specific roles, like requirements analysis, design, coding, testing, maintenance, etc.
- ✓ They perform these roles repeatedly for multiple projects and develop deep insights to their work.
- ✓ More attention is paid to proper documentation at the end of each phase as different teams perform different phases.
- ✓ The functional organization also provides an efficient solution to the staffing problem.

- ✓ More attention is paid to proper documentation at the end of each phase as different teams perform different phases.
- ✓ Functional organisation also provides an efficient solution to the staffing problem
- ✓ More attention is paid to proper documentation at the end of each phase as different teams perform different phases.
- ✓ The functional organization also provides an efficient solution to the staffing problem.

Because of the layout, processes are more streamlined and productive. The organization's ability to multitask effectively is a direct result of the division of labour into several departments. There is no need to invest time in training because roles and responsibilities rarely shift, and everyone knows what they're responsible for. Since there is only one level of management, everyone is clear on who they report to. Employees benefit from this since it simplifies communication and cuts down on missteps. Since everything is standard, workers can rest assured that they are performing a good job. Their devotion to the team and the company as a whole will increase.

Project staffing approach is possibly the most important advantage of the functional organization.

- ✓ The project staffing problem is simplified significantly because personnel can be assigned to a a project as needed, and returned to the functional group when they are no longer required..
- ✓ A further advantage of the functional organization is that it is more effective in handling the problem of manpower turnover.
- ✓ This happens because engineers can be brought in from the functional pool when needed. Also, this organization necessitates production of good quality documents, so new engineers can quickly understand the work already done. This occurs because engineers can be quickly accessed from the available functional pool. In addition, this structure calls for the creation of high-quality documentation to help new engineers pick up on the work with minimal training time.
- ✓ In addition, a well-defined chain of command instructs workers on how to perform their duties effectively. In a hierarchical organisation, for instance, workers will have to exert more effort to woo or pay off those in positions of authority. Workers in a

decentralised setting are expected to take more initiative and bring fresh perspectives to the table when addressing issues. This is useful for both the organization and its prospective employees, as it lays out the standards by which workers will be evaluated and the talents that will be highlighted in the workplace.

- Till now we had discussed about the advantages of organizational structure , we will also see the disadvantage part of it :
  - ✓ Even though functional organization has several advantages, it is not very popular in the software industry.
  - ✓ Taking into consideration the present skill shortage, it is very difficult for the functional organizations to fill in slots for some roles such as maintenance, testing, and coding groups.
  - ✓ Another problem with the functional structure is that if an organization handles projects requiring knowledge of specialized domain areas, then these domain experts cannot be brought in and out of the project for the different phases, unless the company handles a large number of such projects.
  - ✓ Therefore for the stated reasons the functional format is not suitable for small organizations handling just one or two projects.
- Project Based Organisational Structure:
  - ✓ There is a clear, properly defined line of authority that results in faster decision-making and approval.
  - ✓ Communication becomes more effective and simple and project team members gain more experience working on different types of projects as the need for them arises.
  - ✓ Project Based Organizational:
  - ✓ In the project based organizational structure, the project development staff are divided depending on the project for which they work.
  - ✓ In this format, a group of engineers are allocated to the project at the beginning of the project and that they remain associated with the project until the completion of the project.
  - ✓ Thus, the identical team carries out all the life cycle activities.
  - ✓ Project manager has complete project authority. He has jurisdiction over the project's budget, schedule, and the project team

- Drawbacks of Project Based Organisational Structure:

  Every team member takes on the role of the designer, coder, tester, etc during the course of the project. Let us see the Drawbacks which we will encounter in organisational structure.

  - ✓ A project organization structure compels the manager to take in nearly a fixed number of engineers for the entire duration of the project.
  - ✓ This results in engineers idle in the initial phase of the software development and under tremendous pressure in the final stages of the development.
  - ✓ Employees may generally find themselves under a lot of pressure, especially if they happen to work on multiple projects at the same time.
  - ✓ This often leads to poor communication amongst the team members as everyone is left more or less playing "catch-up".
- Mixed Control Team Organisation:

Each democratic setup at the programmer level attempts solution to a single component.

This structure is extremely popular and is being used in many software development companies Each programmer-level democratic structure provides an approach to solving a specific part. This framework is widely adopted, and it is used by many different software development organizations.

- Characteristics of a Good Software Engineer:

An experiment conducted by Sackman [1968] shows that the ratio of coding hour for the worst to the best programmers is 25:1, and the ratio of debugging hours is 28:1.

A good software engineers should posses the following attributes:

Exposure to systematic techniques, i.e. familiarity with software engineering principles. Good technical knowledge of the project areas (Domain knowledge). Good programming abilities and Good communication skills. These skills comprise of oral, written, and interpersonal skills. The following characteristics characterise talented software engineers: Learning to use methodical methods,

- Characteristics of a Good Software Engineer:

1. Studies show that these attributes vary as much as 1:30 for poor and bright candidates, also we will see the characteristics that a software engineer should posses
   - ✓ He should be highly motivational person , even the brightest can be poor programmers, we can increase it with better salaries and good working environment.
   - ✓ Sound knowledge of fundamentals of computer science. So problem solving abilities will increase since fundamental knowledge is very clear he will be able to solve the problem by understanding its critical analytics
   - ✓ Intelligence, Ability to work in a team and Discipline all these factors are also very much important characteristics that a software engineer should possess.

SELF-ASSESSMENT QUESTIONS – 4

10. Team Leader is responsible for all aspects of the project. (True / False)
11. _____manage hardware/software requirements for development, testing, validation, and production environments.
12. _____define testing procedures and certification process.
    a) Software Support
    b) Software Development
    c) Software Management
    d) Software Testing

6. SUMMARY

In this unit, you have learnt about Software Development Organization and Roles in Software Development Industry. Let's summarize:

- Information Systems (IS) must be a key component of organizational design. Organizational designers must understand what IS can do.
- The flow of information determines organization structures.
- Virtual and networked organizations are rising in use and are replacing older legacy structures.
- IT affects managerial control mechanisms and managers must ensure that these controls are in place.
- Virtual organizations make it possible for employees to live anywhere. Virtual Teams are increasing in frequency and the challenges that they pose must be addressed.
- We have discussed the four important Ps, namely People, Process, Project and Product.
- We have discussed IT organizational structures - flat organizational structures, matrix organizational structures and network organizational structures etc. These organizational structures facilitate communication among the teams in the organization.

7. TERMINAL QUESTIONS

1. Explain the organizational structures in detail.
2. Differentiate between different organizational structures.
3. Explain the different roles of software development.

8. ANSWERS

Self-Assessment Questions

1. True
2. Software Engineering Institute (SEI)
3. a)
4. False
5. Functional, Project
6. b)
7. True
8. Matrix
9. c)
10. True
11. Logistics
12. d)

Terminal Questions

1. Traditional organizations are hierarchical, flat or matrix in design. In hierarchical organizations, middle managers tell subordinates what to do and tell superiors the outcomes. IS supports this hierarchy. In flat-structured organizations, work is more flexible, and employees do whatever is needed. It allows offloading extra work and supports intra- firm communications. (Refer to Section 4)
2. Hierarchical organizations are based on the concepts of division of labor, specialization, and unity of command. In flat organizations, decision-making is centralized. (Refer Sections 4.1 to 4.4)
3. Middle Management, Team Leader, Logistics, and Product Development are some of the roles in Software Development. (Refer to Section 5)