Unit 1: Introduction to Software Project Management (SPM)

Software: Software is more than just a program code. A program is an executable code, which solve a problem. **Software** refers to a collection of programs, data, and instructions that tell a computer or other digital device what to do. It is a set of automatic instructions that enable a computer or device to perform specific tasks, solve problems, or carry out various functions.

Engineering: Engineering is the branch of science and technology concerned with the design, building, and use of engines, machines, and structures. On the other hand, engineering is all about developing products, using well defined, scientific principles and methods.

Software engineering: 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.

Project: A project is defined as a sequence of tasks that must be completed to attain a certain outcome. The term Project refers to "to any temporary work with a definite beginning and end".

Software Project: 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.

Management

A management involves the following activities:

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

Software Project Management (SPM) is a proper way of planning and leading software projects. It is a part of project management in which software projects are planned, implemented, monitored, and controlled.

Software project management is a discipline that involves planning, organizing, and managing resources to complete a software project. It consists of several different types of management, including conflict management, risk management, requirement management, change management, software configuration management, and release management.

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.

Software Project Vs. Other Projects

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

With Software, progress is not immediately visible since work is logical; however, for physical objects like bridges, work progress can be seen from time to time. In Software Development, there is a level of uncertainty.

2. Complexity

Software projects contain more complexity than other engineered artifacts. For example, in a bridge, there is a clear structural relationship between parts, whereas software component relationships are much more complicated. We can't measure the complexity of a software project until we work on it.

3. Conformity

Physical systems are governed by consistent physical law, where Software developers have to conform to the requirements of human clients.

4. Flexibility:

Software systems are particularly subject to change. A bridge has to be built in a specific order, whereas We can make Software much more flexible.

5. Customization:

Software products can be easily customized and updated according to user needs, while other products may require more effort to customize or update.

6. **Cost:**

Software products are generally cheaper to produce and distribute than other products, which require more resources and materials.

Software Product Attributes

Software products have a number of associated attributes that reflect the quality of that software. These attributes are not directly concerned with what the software does. They reflect its behavior while it is executing, the structure and organization of source program and associated documentation. The essential attributes of good software are:

1. **Acceptability** - Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use.
2. **Efficiency** - Software should not make uneconomical use of system resources such as memory and processor cycles. Efficiency therefore includes processing time, resource utilization, resource management etc.
3. **Maintainability** - Software should be written in such a way that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an expected requirement of a changing business environment.

   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.
4. **Performance-** Performance shows the response of the system to performing certain actions for a certain period of time. There are two ways 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.
5. **Usability-** 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 complete 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 confuses 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.

6. **Reliability-** 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.
7. **Availability**- 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.
8. **Security -** 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:
9. **Testability** - 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:
10. **Scalability -** 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.

Fundamentals of software project management:

1. **Planning:** The professional creates a complete project plan. The project plan will detail the procedures for executing the task, including its scope, resources, schedules, methodologies, strategies, communications, testing, and administration.
2. **Leading:** A software development manager collects and directs a group of system administrators, developers, software engineers, designers, and data experts. To lead a team, you must have excellent communication, professional skills, and leadership qualities.
3. **Execution:** SPM ensures that each step of a software project is completed effectively if the top management is familiar. These activities include checking

the project's development, monitoring the team's performance, and compiling status reports.

4. **Time management**: Continuing faithfulness to a schedule is necessary for completing tasks. This is challenging when it comes to the management of software systems because it is impossible to resist making alterations to the initial development plan as time passes. Software project managers should be experts at managing risks and being ready for emergencies to keep things moving forward even when things get change.
5. **Budget**: As with conventional project leaders, the software project manager should create a budget plan and follow it as accurately as possible, controlling expenditures and reallocating funds as necessary. We learn how to successfully handle the financial component of projects through SPM so that we can prevent coming up against financial trouble later on during the planning process.
6. **Maintenance**: Software project management emphasizes continuous product testing to detect and fix faults early, adjust the final product to the client's demands, and keep track of progress.

Some other fundamentals of software project management are:

- 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 passable tools and training to get the job done
- Measuring risks and justifying them

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 accepted “off the shelf”, there may be no any coadding, but this is still fundamentally a software project because so many of the other activities associated with software will still be present here. Basic activities covered by software project management are: -

1. Feasibility Study
2. Planning
3. Project Execution

1. **Feasibility Study**: -
   Feasibility study measures whether a project is worth starting-that it has a valid business case or not. Information is gathered about the requirements of the proposed application. The stakeholders may know the aim they wish to follow, but not be sure about achievement. The developmental, operational costs, and benefits of the new system are to be projected.
2. **Planning: -**
   If a feasibility study indicates the project as worthy, planning starts. Normally a complete detailed plan is created for smaller projects. For larger projects, an outline plan for the whole project is prepared and a detailed for the first stage of plan will be created. After completing one stage then detailed of next stage will create.
3. **Project Execution: -**
   Execution often contains design and implementation sub phases. Design is making decisions about the form of the products to be created. External appearance of the software i.e., user interface (UI) or the internal architecture. Plan details the activities to be carried out to create the products. Planning and design can be confused because at the most detailed level.

Importance of Software Project Management: -

Project management software is designed to make the workflow of a team working on the same project easier. This software is suitable for small start-up teams as well as large organizational teams working at various levels. Software project management is a crucial aspect of the software development process, as it involves planning, organizing, and executing tasks to ensure that a software project is completed on time, within budget, and with the desired quality.

1. **Project Planning:** Effective project management begins with thorough planning. Project managers define **project goals, scope, deliverables, and timelines**. This planning phase helps in setting clear expectations and objectives for the project, which is essential for its success.
2. **Resource Allocation**: Project managers allocate resources such as developers, designers, and testers to various tasks based on their skills and availability. This ensures that the right people are working on the right tasks at the right time.
3. **Risk Management:** Every software project comes with inherent risks, whether they are related to **technical challenges, changing requirements, or resource limitations**. Project managers identify potential risks and develop strategies to mitigate them, reducing the likelihood of project failure.

4. **Cost Control:** Software projects often have budget restrictions. Project managers are responsible for tracking and managing project costs, ensuring that the project stays within budget without compromising its quality.
5. **Time Management:** Meeting project deadlines is critical for customer satisfaction and market competitiveness. Project managers create schedules, set milestones, and monitor progress to ensure that the project is completed on time.
6. **Quality Assurance:** Software project managers are responsible for maintaining and improving the quality of the project's deliverables. They implement processes and standards, conduct reviews and testing, and ensure that the final product meets the required quality standards.
7. **Communication:** Effective communication is essential in software project management. Project managers facilitate communication among team members, stakeholders, and clients. They keep everyone informed about project status, changes, and issues, ensuring transparency and arrangement.
8. **Stakeholder Management**: Projects involve multiple stakeholders, including clients, end-users, sponsors, and team members. Project managers must involve with these stakeholders to gather requirements, manage expectations, and ensure that their needs are met with this project.
9. **Documentation:** Proper documentation is essential for project transparency and knowledge transfer. Project managers ensure that project documentation, including requirements, design, and testing documents, is created and maintained throughout the project.
10. **Continuous Improvement:** After project completion, project managers conduct post-project reviews to identify lessons learned and areas for improvement. This knowledge can be used to enhance future projects and organizational processes.
11. **Effective Task Delegation**: Managers may find it difficult to distribute tasks to team members in a fair manner. Inefficient team progression & even internal frustration can result from inefficient task distribution.

    Managers can easily delegate tasks to members using project management software because it provides clear overviews of individual ongoing team tasks & allows them to see who is available to take on any extra work.

Problems of Software Project Management:

Software project management is a complex effort, and various challenges and problems can arise during the development of a project. These problems can lead to delays, increase cost, and even project failures if not effectively addressed. Here are some common software project management problems:

1. **Scope Creep:** This occurs when additional features, requirements, or changes are introduced after the project has started. It can lead to delays and increased costs as the project team tries to accommodate these changes.
2. **Unclear Requirements:** Incomplete or unclear requirements can lead to misunderstandings and misinterpretations, producing the project to turn from its planned course.
3. **Poor Planning**: Insufficient project planning, including inaccurate time and resource estimates, can lead to schedule missing and budget issues.
4. **Lack of Stakeholder Involvement:** Projects suffer when key stakeholders are not actively involved or when their needs and expectations are not properly addressed.
5. **Ineffective Communication:** Poor communication among team members, stakeholders, and project managers can lead to misunderstandings, misalignment, and project delays.
6. **Technical Challenges:** Complex technical issues or unexpected problems can arise during development, causing delays and requiring additional resources.
7. **Change Management:** Poorly managed changes, whether they are related to requirements, technology, or personnel, can disrupt project plans and cause resistance among team members.
8. **Estimation Errors:** Inaccurate time and cost estimates can result in impractical project plans and expectations.
9. **Lack of Documentation:** Insufficient or outdated documentation can delay knowledge transfer and future maintenance efforts.
10. **Cultural and Organizational Issues:** Organizational culture and processes can either support or delay project management efforts, depending on their alignment with project goals and methodologies.

Characteristics of good project manager:

Regardless of the industry that you work in, if you want to be a successful project manager there are several qualities you should possess. Some of the qualities manager need to have been as follow: -

1. **Leadership skills:** Successful project management requires strong leadership skills on behalf of the manager overseeing the project. As a project manager, must be able to effectively lead his/her team from start to finish to ensure the efficient completion of a project. Being a good leader means that you can motivate your team to perform at their best throughout the project and ensure all team members have a clear understanding of what is expected from them.

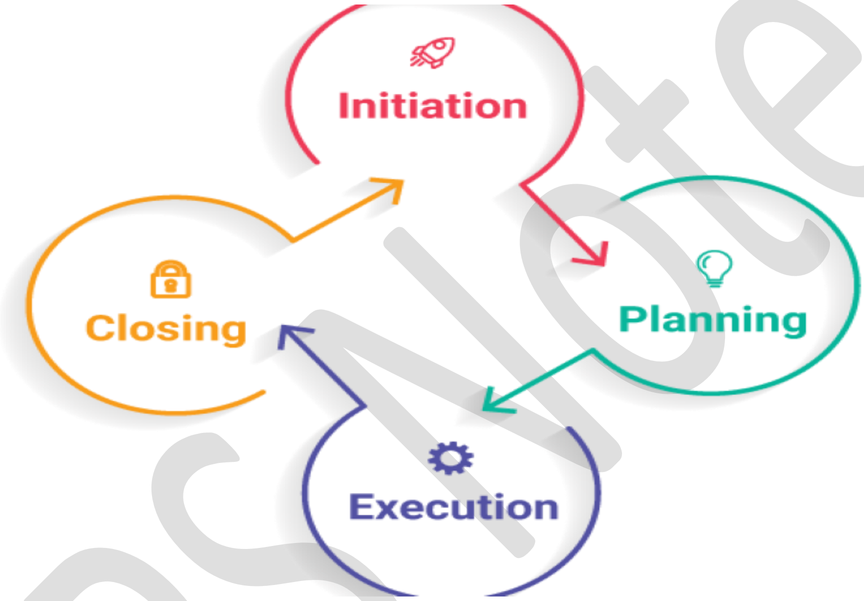
2. **Communication skills:** Effective project management requires clear and competent communication about the expectations, goals and responsibilities of the team who will be completing the project. Being able to efficiently communicate with your team as well as clients and management can ensure that everyone is of the same understanding regarding project expectations. Good communication skills also allow you to provide constructive feedback to your team to better guide them. Both written and oral communication skills are important for project managers to have.
3. **Problem-solving skills:** Successful project managers should be able to solve a variety of problems throughout all stages of a project. Issues that may need to be solved could involve team members, clients or stakeholders related to the project. Being able to think on your feet and address disputes and problems is key to ensuring the project is completed in an efficient and timely manner.
4. **Delegation skills:** Being able to assign and oversee tasks is a fundamental component of successful project management. As the project manager, you should have the ability to access the skills of your team and sign tasks based on these skills. Effective delegation also requires you to trust your team members to fulfill their duties and allows you to avoid micromanaging them.
5. **Team-building skills:** Being a good team builder can help you bring a successful team together that works as a unit to complete a project. Leading your team towards a common purpose and keeping the enthusiasm and motivation alive throughout the entire project will make you a better project manager and leader.
6. **Integrity:** Integrity, sometimes also referred to as loyalty or honesty, is an important quality for a project manager to have. Integrity can help you set a good example for your team and make them more likely to follow that example. It also shows your team that you are committed to the project and are willing to see it through to the end at all costs.

Project managers in software organize software projects and can assign tasks to software engineering teams according to the specifications of a task. Software project managers use their leadership skills, technical knowledge and experience to direct their teams and ensure software meets client requirements. Understanding the role of a project manager can help you determine if it’s a suitable career for you.

Every project manager wants to be successful, yet there are many difficulties for “success.” To be successful as a software project manager, you should have a good working knowledge of project estimation techniques and the ability to motivate team members. Ultimately, a top-notch software project manager should have good communication and decision-making skills.

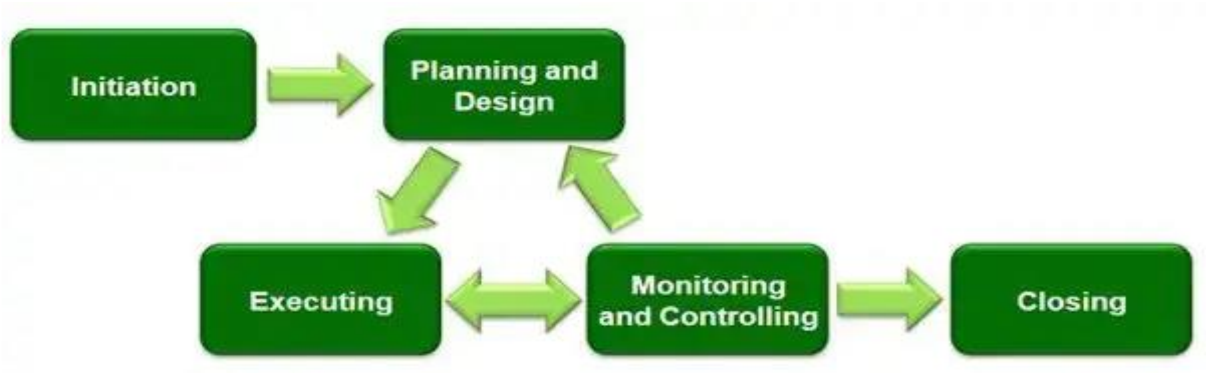
Project management cycles:

A project never exists in a vacuum. Rather, a project is executed by a team within an organization that has some sort of a project management framework established to enable the process. After the project has been defined and the project team has been appointed, you are ready to enter the second phase in the project management life cycle: the detailed project planning phase.



1. **Project Initiation** – this is the start of the project. It may involve many sub-activities including: a feasibility study, identifying the scope, identifying deliverables, identifying project stakeholders, developing a business case, creating a statement of work, and possibly initial costs, price, and timeline for work to be done.
2. **Project Planning** – Once the project is approved from the initiation phase, it moves into planning. This phase involves creating a project plan, including the tasks, schedule, resources, and constraints on the project. The budget for the project is also created in this phase. In addition, risk should be anticipated and identified at this stage, as well as mitigation plans.
3. **Project Execution** – This phase is where the work gets done. Task owners begin work and the project manager oversees that tasks are done in a timely manner and workflow continues smoothly. Monitoring and Controlling (managing the work and financials) are a big part of this phase, as issues will always arise and require quick adjustments as the project progresses.

4. **Project Closing** – Once the team has completed all the tasks, and the project owner signs off that all deliverables are complete, the project is closed. Any documentation is handed over to the project owner and if required to an ongoing maintenance organization. The project is then analyzed for performance to determine whether the project's goals were met (tasks completed, on time and on budget).



Types of Project Plans:

There are three types of project plans:

1. **Manual:** Manual projects are for persons who quickly need a Gantt chart of a nice visual representation of schedules and other project components for use in a presentation (as required by almost 50% of users of PM tools). In this case, there are no values for efforts and there are no durations for work packages.
2. **Effort – Driven**: The effort-driven project plan calculates the duration based on the assigned staff. There is no fixed duration. The user enters a cost and indirectly determines the duration through the resource allocation (usually, the planning of IT projects follows this). An effort-driven project plan could be team-enabled or role-enabled or both teamenabled and role-enabled.
3. **Fixed Duration:** Fixed duration project plans are those for which the duration of the project is of prime importance; the duration of the project is known and not the effort/cost. Moreover, since the duration has to be kept fixed, the resource loading will have to be done in a way that the tasks are completed in the given time. So, for a fixed duration project when we schedule work packages (fixed duration type), the resource loading needs to be changed in order to complete the work packages in a given duration.

We have covered:

Unit -1 Software Project Management Concepts - 8 Hrs.

Introduction, Project and Software project, Software project vs other project, Importance and Problems in software project management, Process of SPM. Characteristics of good project manager, Successful Software Project Manager, Overview of Software Project Planning.