

Learning Resource

On

Software Project Management

Unit-1: Part-1

Prepared By:

Kunal Anand

Assistant Professor, SCE

KIIT, DU, Bhubaneswar-24

This part includes the brief discussion on:

- Definition of Software Project Management (SPM)
- How are software products different from other kinds of projects?
- Components of SPM
- Activities covered under SPM
- Challenges and Opportunities in SPM
- Tools and Techniques in SPM
- Managing human and technical resources
- Setting Objectives
- Project success/failure

- **Software**

- A collection of programs
- Combined in a package
- To perform different applications

- **Project**

- A temporary activity but with a proper plan
- Performed by people in formally organized group
- To produce a unique product or service by adopting standard establishment and practices.

Characteristics of Projects

A task is more 'project-like' if it is:

- Non-routine
- Planned
- Aiming at a specific target
- Carried out for a customer
- Carried out by a temporary workgroup
- Involving several specialisms
- Made up of several different phases
- Constrained by time and resources
- Large and/or complex

- According to **Harold Koontz**, “Management is the art of getting the work done through people in formally organized groups.”
- Management is an individual or group of individuals that accepts the responsibilities to run an organization.
- The salient features of management:
 - It is one or group of individuals who plan, organize, direct and control all the essential activities of the organization.

Features of Management

- They don't work alone; rather, they motivate people to do the work and coordinate all the activities to achieve well-defined objectives.
- Management is a continuous and never-ending process.
- It is “Result Oriented”.
- It is dynamic in nature.
- Management may be aided but not replaced by computers.
- It follows established principles and rules.

Why is Project Management important?

- Large amounts of money are spent on ICT e.g., the UK government in 2003-4 spent £2.3 billion on contracts for ICT and only £1.4 billion on road building
- Projects often fail – Standish Group claim only a third of ICT projects are successful. 82% were late, and 43% exceeded their budget.
- Poor project management is one of the major factors in these failures ,

Are software projects different from other projects?

- Software projects differ from other type of projects in following aspects:
 - Tangible vs nontangible
 - Complexity
 - Flexibility
 - Invisibility

The above issues make software more problematic to build than other engineered artifacts.

Projects can be:

- **In-house:** clients and developers are employed by the same organization
- **Out-sourced:** clients and developers employed by different organizations
- **‘Project Manager’** could be:
 - a ‘contract manager’ in the client organization
 - a technical project manager in the supplier/services organization

Components of SPM

- SPM involves planning, executing, and controlling software projects to meet specific goals within constraints such as time, budget, and resources. The key components of software project management include:
 - **Project Planning**
 - **Scope Definition:** Identifying project goals, deliverables, and boundaries.
 - **Requirements Gathering:** Documenting functional and non-functional requirements.
 - **Work Breakdown Structure (WBS):** Dividing tasks into manageable sections.
 - **Scheduling:** Creating timelines with milestones and deadlines.
 - **Resource Planning:** Allocating human, technological, and financial resources.

- **Project Estimation**

- **Time Estimation:** Calculating the time required for each task or phase.
- **Cost Estimation:** Budgeting for development, tools, and contingencies.
- **Effort Estimation:** Determining the number of hours or days required for tasks.

- **Risk Management**

- **Risk Identification:** Spotting potential risks early in the project.
- **Risk Analysis:** Assessing the likelihood and impact of risks.
- **Risk Mitigation Planning:** Developing strategies to minimize or handle risks.

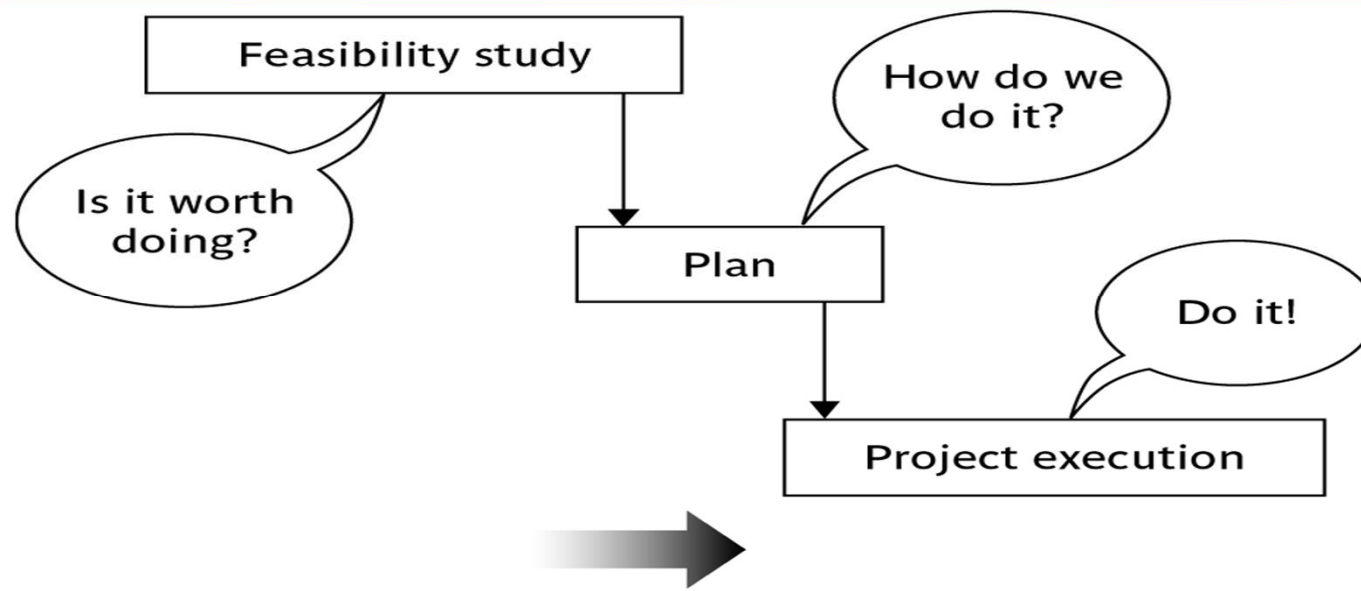
- **Team Management**
 - **Role Definition:** Assigning clear roles and responsibilities to team members.
 - **Communication Management:** Establishing channels and tools for effective collaboration.
 - **Motivation and Leadership:** Ensuring team morale and resolving conflicts.
- **Quality Management**
 - **Quality Assurance (QA):** Establishing processes to ensure high-quality deliverables.
 - **Quality Control (QC):** Testing and validating the software against requirements.
 - **Standards Compliance:** Adhering to industry and organizational standards.

- **Project Monitoring and Control**
 - **Progress Tracking:** Monitoring timelines and deliverables.
 - **Performance Metrics:** Measuring productivity and quality.
 - **Issue Management:** Identifying and resolving problems quickly.
- **Stakeholder Management**
 - **Engagement:** Keeping stakeholders informed and involved.
 - **Expectation Management:** Aligning project deliverables with stakeholder needs.
 - **Feedback Integration:** Incorporating stakeholder input into project adjustments.

- **Configuration Management**
 - **Version Control:** Managing changes to software artifacts.
 - **Baseline Management:** Maintaining approved versions of deliverables.
 - **Change Control:** Handling requests for modifications systematically.
- **Delivery and Deployment**
 - **Implementation Planning:** Defining deployment processes.
 - **User Training:** Providing necessary documentation and training to users.
 - **Post-Deployment Support:** Ensuring smooth transition and addressing issues after launch.

- **Project Closure**
 - **Final Review:** Ensuring all objectives are met and deliverables are accepted.
 - **Documentation:** Archiving lessons learned, project reports, and artifacts.
 - **Handover:** Transitioning the system to maintenance teams or end-users.
- Each of these components contributes to the success of software projects by ensuring a structured approach to meeting project objectives.

Activities covered by project management



- **Feasibility study:** The sole purpose is to decide whether the project is doable.
- **Planning:** If the feasibility study indicates the proposed work is viable, then the project planning can start. Here, the essential aspects like scheduling, budget and resource allocation, staffing, etc., are planned.
- **Execution:** Once the above two steps are done, one can start with the project execution. Usually, the execution contains two sub-phases named design and implementation.

Challenges and Opportunities in SPM

- SPM involves both challenges and opportunities. Understanding these can help managers anticipate potential issues and leverage available advantages to ensure project success.
- **Challenges in SPM**
 - **Unclear Requirements:** Incomplete or changing requirements can disrupt project plans. It can lead to scope creep, rework, and potential project failure.
 - **Time Constraints:** Unrealistic deadlines can compromise quality. It causes stress, rushed deliverables, and potential burnout.
 - **Budget Limitations:** Financial constraints may limit resources and tools. It affects quality, delivery timelines, and team satisfaction.

- **Team Collaboration:** Poor communication and lack of alignment can cause inefficiencies. It leads to delays and conflicts within the team.
- **Technical Challenges:** Complex technologies or unproven tools may introduce risks. It causes delays and increases debugging or redevelopment efforts.
- **Risk Management:** Failure to identify and mitigate risks early. It leads to unplanned issues and cost overruns.
- **Quality Assurance:** Balancing speed with maintaining high standards. Risk of delivering subpar software.
- **Adapting to Change:** Rapid changes in technology or market demands. It requires flexibility, often disrupting the original plan.

- **Opportunities in SPM**
 - **Innovation in Tools and Processes:**
 - Opportunity: Leverage advanced project management tools (e.g., Agile, DevOps).
 - Benefit: Improves efficiency, collaboration, and tracking.
 - **Global Talent Pool:**
 - Opportunity: Access skilled professionals worldwide.
 - Benefit: Enhances expertise and innovation.
 - **Agile Methodologies:**
 - Opportunity: Implement flexible frameworks for iterative development.
 - Benefit: Allows faster adaptation to changing requirements.

- **Improved Communication Technology:**
 - Opportunity: Use tools like Slack, Zoom, or MS Teams for collaboration.
 - Benefit: Bridges communication gaps in distributed teams.
- **Data-Driven Decision-Making:**
 - Opportunity: Use analytics and performance metrics to guide decisions.
 - Benefit: Identifies bottlenecks and optimizes resource allocation.
- **Automation in Testing and Deployment:**
 - Opportunity: Automate repetitive tasks such as code testing.
 - Benefit: Reduces manual errors and accelerates delivery.

- **Focus on Sustainability:**
 - Opportunity: Adopt eco-friendly and cost-efficient practices.
 - Benefit: Aligns with organizational goals and public sentiment.
- **Scalability and Cloud Technology:**
 - Opportunity: Use cloud platforms for flexible infrastructure.
 - Benefit: Facilitates seamless scaling and efficient resource usage.
- **Emphasis on Soft Skills:**
 - Opportunity: Invest in leadership and interpersonal skills.
 - Benefit: Builds cohesive teams and effective communication.

- **Enhanced Stakeholder Engagement:**
 - Opportunity: Involve stakeholders through collaborative planning.
 - Benefit: Ensures alignment and increases project acceptance.
- By addressing these challenges proactively and seizing opportunities strategically, software project managers can enhance the chances of project success while driving innovation and value creation.

Tools and Techniques in SPM

- **Tools in SPM**

- **Project Planning Tools:**

- Examples: Microsoft Project, Smartsheet, Monday.com, Asana.
 - Purpose: Create project plans, timelines, and schedules.

- **Collaboration and Communication Tools:**

- Examples: Slack, Microsoft Teams, Zoom, Google Workspace.
 - Purpose: Facilitate team communication and file sharing.

- **Version Control Systems:**

- Examples: Git, GitHub, Bitbucket, GitLab.
 - Purpose: Manage code versions and track changes collaboratively.

– **Task and Workflow Management Tools:**

- Examples: Jira, Trello, ClickUp, Wrike.
- Purpose: Track tasks, assign responsibilities, and visualize workflows.

– **Risk Management Tools:**

- Examples: RiskWatch, Active Risk Manager (ARM).
- Purpose: Identify, analyze, and mitigate risks.

– **Resource Management Tools:**

- Examples: Resource Guru, Hub Planner, TeamGantt.
- Purpose: Manage resource allocation and availability.

– **Quality Assurance (QA) Tools:**

- Examples: Selenium, TestRail, JMeter, Postman.
- Purpose: Automate and manage testing processes.

- **Documentation Tools:**
 - Examples: Confluence, Notion, Microsoft OneNote.
 - Purpose: Create and manage project documentation.
- **Time Tracking and Reporting Tools:**
 - Examples: Toggl, Harvest, Clockify.
 - Purpose: Track time spent on tasks and generate reports.
- **Agile Tools:**
 - Examples: Rally, Azure DevOps, Scrumwise.
 - Purpose: Support Agile methodologies like Scrum and Kanban.
- **Budgeting and Financial Tools:**
 - Examples: QuickBooks, Planview, Scoro.
 - Purpose: Manage project budgets and financial planning.

- **Deployment and Integration Tools:**
 - Examples: Jenkins, Kubernetes, Docker, Ansible.
 - Purpose: Automate CI/CD pipelines and deployment processes.
- **Techniques in SPM**
 - **Work Breakdown Structure (WBS):** Breaking down the project into smaller, manageable tasks.
 - **Critical Path Method (CPM):** Identifying the sequence of tasks that determine the project duration.
 - **PERT (Program Evaluation Review Technique):** Estimating project duration using optimistic, pessimistic, and most likely time estimates.

- **Agile Methodologies:** Frameworks like Scrum, Kanban, and SAFe for iterative development.
- **Gantt Charts:** Visualizing project schedules and timelines.
- **Earned Value Management (EVM):** Measuring project performance against scope, schedule, and budget.
- **Kanban Boards:** Visualizing task progress in columns (e.g., To Do, In Progress, Done).
- **Risk Assessment Techniques:** Tools like SWOT Analysis, Risk Matrices, and Monte Carlo simulations.
- **MoSCoW Prioritization:** Categorizing tasks as Must-have, Should-have, Could-have, and Won't-have.
- **Rapid Prototyping:** Building quick prototypes for early feedback.

- **Continuous Integration and Continuous Deployment (CI/CD):** Automating code integration and deployment to enhance reliability.
- **Stand-Up Meetings:** Short, daily team meetings to review progress and address blockers.
- **Change Control Processes:** Systematically evaluating and approving project changes.
- **Root Cause Analysis (RCA):** Investigating the cause of problems and preventing recurrence.
- By using these tools and techniques effectively, software project managers can enhance team collaboration, minimize risks, and ensure successful project delivery.

Managing Human Resources

- Managing human resources in SPM is crucial for ensuring that the right people are in the right roles, effectively collaborating to achieve project goals.
- It involves planning, organizing, and leading the team to optimize productivity and maintain motivation.
- **Stakeholders:** These are people who have a stake or interest in the project. They could be users/clients or developers/implementers.
- They could be:
 - Within the project team
 - Outside the project team, but within the same organization
 - Outside both the project team and the organization.
- Need to define common project objectives.

- Key Aspects of Human Resource Management in Software Projects
- **Resource Planning:**
 - **Role Identification:** Define roles and responsibilities for developers, testers, designers, and other team members.
 - **Skill Mapping:** Match team members' skills to project requirements.
 - **Resource Allocation:** Ensure efficient use of available personnel for each phase of the project.
- **Team Building**
 - **Hiring and Onboarding:** Recruit individuals with the required technical and soft skills.
 - **Training:** Provide necessary training on tools, technologies, or methodologies (e.g., Agile, DevOps).
 - **Team Cohesion:** Foster collaboration and trust among team members.

- **Defining Roles and Responsibilities**
 - Clearly outline roles such as project manager, business analyst, developer, tester, and UI/UX designer.
 - Set expectations for deliverables and performance metrics.
- **Communication Management**
 - **Channels:** Use tools like Slack, Teams, or email to facilitate clear communication.
 - **Meetings:** Schedule regular team meetings, stand-ups, or one-on-ones.
 - **Feedback:** Establish a system for ongoing performance feedback.

- **Conflict Resolution**

- Address conflicts promptly and fairly to maintain a harmonious team environment.
- Encourage open communication to resolve misunderstandings or misalignments.

- **Motivation and Leadership**

- Incentives: Offer bonuses, flexible work options, or growth opportunities.
- Support: Provide emotional support and guidance to help team members manage stress.

- **Time Management**

- Scheduling: Ensure team members have realistic workloads and deadlines.

- **Retention and Succession Planning**
 - **Retention:** Offer career growth, competitive compensation, and a positive work environment.
 - **Succession Planning:** Prepare backups for key roles to ensure continuity in case of turnover.
- **Challenges in Managing Human Resources**
 - **Skill Gaps:** Addressing shortages in required technical expertise.
 - **Team Dynamics:** Managing diverse personalities and work styles.
 - **Remote Collaboration:** Ensuring productivity in geographically dispersed teams.
 - **Burnout:** Balancing workloads to prevent team fatigue.
 - **Attrition:** Managing the impact of key team members leaving mid-project.

Managing Technical Resources

- Managing technical resources in SPM involves effectively planning, allocating, and utilizing the tools, infrastructure, and technologies required for successful project delivery.
- This ensures that technical assets are available, optimized, and aligned with project goals.
- **Key Aspects of Technical Resource Management**
 - **Resource planning:** Identify existing technical resources. Determine additional resources needed based on project requirements. Plan costs for procuring and maintaining technical resources.
 - **Technology selection:** Choose appropriate tools for development, testing, deployment, and project management. Select technologies and frameworks that align with the project's needs.

– **Infrastructure management**

- Development Environment: Set up workstations, servers, and network configurations.
- Test Environment: Provision staging and testing environments that mimic production.
- Continuous Integration/Continuous Deployment (CI/CD): Implement automated pipelines for efficiency.

– **Version Control:** Use tools like Git, GitHub, or Bitbucket for managing code versions.

– **Resource allocation:**

- Ensure timely procurement and renewal of software licenses.
- Allocate sufficient computational resources to avoid bottlenecks.

- **Performance monitoring:** Track the utilization of technical resources to prevent overuse or underuse. Use tools like New Relic or AWS CloudWatch for real-time monitoring of systems.
- **Security management:** Access control, Data protection, Threat detection
- **Scalability and Flexibility:** Utilize services like AWS, Azure, or Google Cloud to handle growing demands.
- **Training and Documentation:** Train the team on how to use specific tools and technologies effectively. Maintain documentation for tools, configurations, and troubleshooting.

- **Lifecycle Management**

- Procurement to Retirement: Plan the acquisition, maintenance, and eventual decommissioning of technical assets.
- Upgrades: Regularly update tools and technologies to remain compatible with industry standards.

- **Challenges in Managing Technical Resources**

- **Budget Constraints:** High costs of tools, infrastructure, and licenses.
- **Rapid Technological Change:** Keeping up with emerging technologies and trends.
- **Overuse or Underutilization:** Inefficient resource allocation leading to wastage or shortages.
- **Security Risks:** Protecting resources from cyber threats and unauthorized access.

Setting Objectives

- Objectives focus on the desired outcomes of the project rather than tasks within it.
- Need for a project authority who sets the project scope and allocates/approves costs.
- Could be one person - or a group like a Project Board, Project Management Board, Steering committee
- Informally, the objective of a project can be defined by completing the statement:

“The project will be regarded as a success if.....
.....”
- Objectives are like post-conditions for the project, focus on what will be put in place, rather than how activities will be carried out.

Goals/sub-objectives

- These are steps along the way to achieving the objective. Informally, these can be defined by completing the sentence
- To reach objective X, the following must be in place

A.....

B.....

C..... etc

Often, a goal can be allocated to an individual.

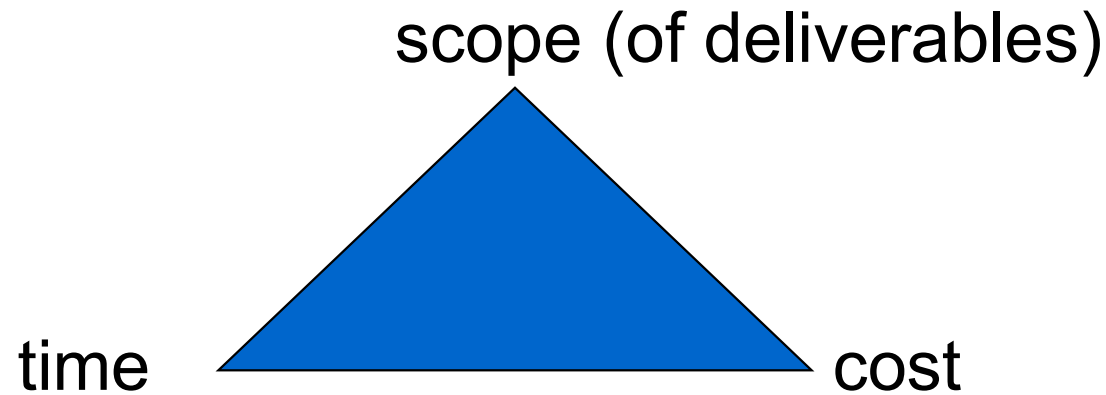
- Individuals might have the capability of achieving goals on their own, but not the overall objective e.g.
 - Overall objective – user satisfaction with software product
 - Analyst goal – accurate requirements
 - Developer goal – reliable software

Measures of Effectiveness

- Measure of effectiveness is a practical method of checking that an objective has been met.
 - How do we know that the goal or objective has been achieved?
 - By a practical test, that can be objectively assessed.
 - e.g., for user satisfaction with software products:
 - **Repeat business** – they buy further products from us
 - **Number of complaints** – if low etc.

Project Success/Failure

- Degree to which objectives are met



- In general, if, for example, the project is running out of time, this can be recovered by reducing scope or increasing costs.
- Similarly, costs and scope can be protected by adjusting other corners of the 'project triangle'.

Reasons for Project Failure

- Reasons that make a project “Failure”:
 - Insufficient resources
 - Unrealistic timeline
 - Unclear specifications
 - Change in scope
 - Disagreement among the stakeholders
 - Bad planning
 - Absence of suitable project management mechanism.

How a Project gets to success???

- Factors contributing to the success of a project include:
 - A comprehensive, realistic plan.
 - The plan must be up-to-date.
 - Resolving conflicts among stakeholders to gain consensus over the outcome.
 - Reasonable resource requirements and their optimal utilization.
 - Build an efficient team and take good care of them.
 - Stakeholders must be informed on a regular basis.
 - Willingness to change or try new ideas.
 - Be not only a manager but also a leader.