# Unit-06

# Software Management

## 6.1. Project Management

Project Management in software development involves planning, organizing, and managing resources to bring about the successful completion of specific project goals and objectives. It encompasses various stages, including initiation, planning, execution, monitoring, and closure, and involves coordinating people, processes, and technologies to meet predefined requirements.

### 6.1.1 Risk Management

Risk management is the process of identifying, analyzing, and responding to potential risks that could impact the project. It aims to minimize the impact of adverse events and capitalize on opportunities.

- Risk Identification
- Risk Analysis
- Risk Response Planning
- Risk Monitoring and Control

### 6.1.2 Managing People

**Managing people** involves leading and coordinating the project team to achieve the project's objectives. It includes motivating, directing, and supporting team members to maximize their performance.

- Team Building
- Motivation
- Conflict Resolution
- Performance Management

### 6.1.3 Teamwork

Teamwork is the collaborative effort of a group to achieve a common goal. In software projects, effective teamwork is essential for successful project completion.

- Communication
- Collaboration

- Trust and Respect
- Shared Goals

## 6.2. Project Planning

Project planning is essential for defining the roadmap of a software project, setting expectations, and ensuring that the project is completed on time, within budget, and to the required quality standards.

### *6.2.1 Software Pricing*

Software pricing involves determining the cost of developing a software project. Accurate pricing is important for budgeting and financial planning.

- **Cost Estimation**: Estimate the total cost of the project, including development, testing, deployment, and maintenance.
- **Pricing Models**: Choose an appropriate pricing model based on the project's nature and client requirements.

### 6.2.2 Project Scheduling

Project scheduling involves planning the timeline for the project, including task assignments, milestones, and deadlines.

- **Task Breakdown**: Break down the project into smaller, manageable tasks and activities.
- **Timeline Development**: Develop a timeline for task completion, considering dependencies and resource availability.
- **Milestones and Deadlines**: Define key milestones and deadlines to ensure the project stays on track.

### *6.2.3 Agile Planning*

Agile planning is a flexible and iterative approach to planning software projects, emphasizing adaptability and collaboration.

- **Iteration Planning**: Plan short development cycles (iterations or sprints) to deliver incremental value.
- **Backlog Management**: Maintain a prioritized list of features and tasks (product backlog) to guide development.

- **Daily Stand-ups**: Hold daily stand-up meetings to discuss progress, identify obstacles, and coordinate efforts.

### 6.2.4 Estimation Techniques

Estimation techniques are used to predict the effort, time, and resources required to complete a software project.

- **Expert Judgment**: Rely on the experience and knowledge of experts to make estimates.
- **Analogy-Based Estimation**: Use historical data from similar projects to make estimates.
- **Parametric Models**: Use mathematical models to estimate project parameters based on historical data and project characteristics.
- **Delphi Technique**: Use a structured communication technique where experts provide estimates independently and iteratively refine them through feedback.

## 6.3. Software Quality and Standards

### Software Standards

Software quality refers to the degree to which a software product meets the specified requirements, user expectations, and maintains robustness under operational conditions. Software standards provide guidelines and best practices to ensure the development of high-quality software.

**Software Quality Focuses on:**

- **Functionality**: The software should perform its intended functions correctly and efficiently.
- **Reliability**: The software should operate without failure under defined conditions for a specified period.
- **Usability**: The software should be user-friendly and easy to navigate.
- **Efficiency**: The software should use system resources (like memory, processing power) efficiently.
- **Maintainability**: The software should be easy to modify to correct defects, improve performance, or adapt to a changed environment.
- **Portability**: The software should be able to operate in different environments (e.g., different hardware, operating systems).

**Software Standards**

Software standards are established guidelines, best practices, and criteria that provide a framework for developing, managing, and maintaining software. These standards ensure consistency, quality, and interoperability across software projects and teams. They are essential for achieving high levels of software quality and compliance with industry and regulatory requirements.

## 6.4. Version Management

Version management, also known as version control, is the practice of managing changes to documents, code, and other collections of information in a systematic and organized manner. It is essential for tracking modifications, ensuring consistency, and facilitating collaboration in software development projects. Version management is typically implemented using Version Control Systems (VCS), which are tools designed to help developers track changes, manage different versions, and collaborate effectively. There are two type of version control system i.e. Centralized Version Control Systems and Distributed Version Control Systems. The various activities in version management:

- Version Control
- Branching and Merging
- Tagging
- Conflict Resolution
- Auditing and Logging
- Dependency Management
- Release Management
- Documentation

## 6.5. Change Management

Change management in the context of software engineering refers to the process of managing changes to software products and systems throughout their lifecycle. This includes identifying, documenting, approving, implementing, and reviewing changes to ensure they are made systematically and efficiently, minimizing disruptions and maintaining quality. The various activities in change management:

- Identification of Need
- Impact Assessment

- Planning
- Communication
- Training and Education
- Testing and Validation
- Implementation
- Monitoring and Evaluation
- Documentation

## 6.6. Release Management

Release management in software development refers to the process of planning, scheduling, coordinating, and deploying software releases, including both new features and bug fixes. The various activities in release management:

- Planning and Scheduling
- Build and Testing
- Deployment
- Monitoring and Feedback
- Documentation and Communication
- Version Control