Detailed Note on Project Management, Agile, and SE

Detailed Note on Project Management, Software Engineering, and Agile Methodologies

Detailed Note on Project Management, Software Engineering, and Agile Methodologies

1. Introduction to Project Management

What is a Project

- A project is a temporary, unique endeavor undertaken to produce a product, service, or result.

- Characteristics include defined start and end, defined resources (time, cost, personnel), and

specific objectives.

What is Project Management

- The discipline of applying knowledge, tools, and techniques to meet project requirements.

- Involves planning, executing, monitoring, controlling, and closing tasks.

Key Aspects

- Defining scope, identifying deliverables, managing risks, and effective communication.

Importance in Software

- Ensures on-time and budget-compliant delivery while handling changing requirements.

Principles

1. Project structure

2. Goals & objectives

3. Sponsor identification

Roles

5. Accountability

6. Scope & change management

7. Risk management

8. Monitoring progress

9. Value delivery

10. Performance metrics

Detailed Note on Project Management, Agile, and SE

11. Project finalization

12. Outcome analysis

2. Software Engineering and Development

History

- Originated in the 1960s due to programming complexity.

- Influenced by Dijkstra, Brooks, Knuth.

- Emphasized modularity and structured programming.

Evolution

- Languages: Fortran, COBOL -> C, C++ -> Java, Python.

Software Crisis

- Projects were over budget and late -> pushed structured methods like OOP.

3. Software Development Life Cycle (SDLC)

Phases

1. Requirement Analysis - Gather/document needs.

2. Design - Use UML, ERDs for system blueprint.

3. Implementation - Coding and unit testing.

4. Testing - Ensure software meets all requirements.

5. Deployment - Release software and provide support.

Bug

4. Software Development Models

Waterfall

Prototyping

Incremental

Spiral

5. Introduction to Agile

Detailed Note on Project Management, Agile, and SE

Definition

- Agile is a flexible, iterative development methodology.

- Focus on customer collaboration, adaptability, and team interaction.

Agile Values

1. Individuals and interactions

2. Working software

3. Customer collaboration

4. Responding to change

Principles

- Frequent delivery, welcome change, motivated individuals, technical excellence.

6. Agile Frameworks

Scrum

Scrum

Artifacts

Scrum

Sprints

Kanban

- Visual workflow management, continuous delivery, WIP limits.

7. Agile Issue Types

Epic

User Story

Task

Bug

Real-World Agile Use

- Used by Google, Microsoft, Spotify.

- Tools: Jira, Trello, Azure DevOps.

9. Sprint Planning - Capacity, Velocity & Workload Distribution

9. Sprint Planning - Capacity, Velocity & Workload Distribution

Sprint Planning is the event where teams estimate how much work can be accomplished in the sprint.

Key Concepts

- Capacity: Total available work hours the team has for the sprint.

  Formula: Capacity = Team members × Hours per day × Sprint duration

- Velocity: Average number of story points completed over past sprints.

  Example: (24 + 26 + 28 + 30 + 32 + 34) / 6 = 29 story points

- Workload Distribution: Matches team hours to effort needed.

  Formula: Workload = (Members × Hours) × Sprint Duration

Example Calculations

- Team of 5 members (Developers, QA, DevOps)

- Sprint duration = 10 days

- Working hours/day = 6

=> Capacity = 5 × 6 × 10 = 300 hours

Assuming average velocity = 29 story points and each point = ~10 hours

=> Estimated workload = 29 × 10 = 290 hours (Well balanced workload)

This approach ensures achievable sprint goals and prevents overloading the team.