

Lesson 3: Software Project Management

Project Management Tools

- Help handle complexity and scale of projects.

1. Gantt Chart

- It is a horizontal bar chart with bars representing activities and time scheduled for the project activities.

2. PERT Chart

- It is capable of graphically representing main events of project in both parallel and consecutive ways.

3. Resource Histogram

- This is a graphical tool that contains bar or chart representing number of resources (usually skilled staff) required over time for a project event (or phase).

4. Critical Path Analysis

- This tool is useful in recognizing interdependent tasks in the project.
-

Lesson 4: Software Requirements and Requirement Engineering

Feasibility Study

- When the client approaches the organization for getting the desired product developed, it comes up with a rough idea

about what all functions the software must perform and which all features are expected from the software.

Requirement Gathering

- Analysts and engineers communicate with the client and end-users to know their ideas on what the software should provide and which features they want the software to include.

Software Requirement Specification (SRS)

- SRS defines how the intended software will interact with hardware, external interfaces, speed of operation, response time of system, portability of software across various platforms, maintainability, speed of recovery after crashing, Security, Quality, Limitations etc.

Software Requirement Validation

- The requirements mentioned in this document are validated. User might ask for illegal, impractical solution or experts may interpret the requirements inaccurately.

Requirement Elicitation Process

Steps:

- Requirements Gathering
- Organizing Requirements
- Negotiation & Discussion

- Documentation

Requirement Elicitation Techniques

- Interviews (structured/unstructured, one-to-one, group)
 - Organization may conduct surveys among various stakeholders by querying about their expectation and requirements from the upcoming system.
- Surveys & Questionnaires
 - A document with pre-defined set of objective questions and respective options is handed over to all stakeholders to answer, which are collected and compiled.
- Task Analysis
 - Team of engineers and developers may analyze the operation for which the new system is required.
- Domain Analysis
 - Every software falls into some domain category. The expert people in the domain can be a great help to analyze general and specific requirements.
- Brainstorming
 - An informal debate is held among various stakeholders and all their inputs are recorded for further requirements analysis.
- Observation
 - Team of experts visit the client's organization or workplace. They observe the actual working of the existing installed systems.
- Prototyping

- Prototyping is building user interface without adding detail functionality for user to interpret the features of intended software product.

Characteristics of Good Requirements

- Clear, Correct, Consistent
- Coherent, Comprehensible
- Modifiable, Verifiable
- Prioritized, Traceable
- From a credible source

Non-Functional Requirements

- Security
- Logging
- Storage
- Configuration
- Performance
- Cost
- Interoperability
- Flexibility
- Disaster recovery

- Accessibility

System Analyst

- System analyst in an IT organization is a person, who analyzes the requirement of proposed system and ensures that requirements are conceived and documented properly and accurately.

Software Metrics and Measures

- **Size:** Lines of Code (LOC), Function Points
- **Complexity:** McCabe's Cyclomatic Complexity
- **Quality:** Defects (number, severity)
- **Process:** Tools/methods used in SDLC phases
- **Resource:** Time, effort, materials uses

Lesson 5: Software Design Basics

Software Design

- Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

Modularization

- Modularization is a technique to divide a software system into multiple discrete and independent modules.

Concurrency in Design

- In software design, concurrency is implemented by splitting the software into multiple independent units of execution, like modules and executing them in parallel.

Coupling and Cohesion

- **Cohesion – Intra-dependability within a module**
 - Cohesion is a measure that defines the degree of intra-dependability within elements of a module.
- **Coupling – Inter-dependability between modules**
 - Coupling is a measure that defines the level of inter-dependability among modules of a program.

Design Verification

- The output of software design process is design documentation, pseudo codes, detailed logic diagrams, process diagrams, and detailed description of all functional or non-functional requirements.