Software Engineering (23IT501)

UNIT–I

- Introduction to Software Engineering

Course Objectives

- • Understand software engineering concepts
- • Study process models and software evolution

UNIT-I Syllabus

- Evolving Role of Software
- Changing Nature of Software
- Software Myths
- Layered Technology
- Process Framework
- CMMI
- Waterfall, Spiral and Agile Models

Introduction to Software Engineering

- Software Engineering is a systematic, disciplined approach to software development, operation, and maintenance.

Objectives of Software Engineering

- Deliver quality software
- Meet user requirements
- Reduce development cost
- Ensure maintainability

The Evolving Role of Software

- Software acts as both a Product and a Vehicle for delivering information.

Software as a Product

- • Produces information
- • Manages information
- • Controls systems
- Examples: ERP, Banking Systems

Software as a Vehicle

- Delivers products and services
- Supports communication
- Enables decision making

Changing Nature of Software

- Software is evolving with AI, IoT, Cloud Computing, Mobile Apps, and Web Applications.

Categories of Software

- System Software
- Application Software
- Engineering Software
- Embedded Software
- Web Applications
- AI Software

Software Myths

- Misconceptions that affect software projects negatively.

Customer Myths

- Requirements can be easily changed
- General objectives are sufficient

Management Myths

- Existing standards are enough
- Adding manpower speeds projects

Practitioner's Myths

- • Working code means project completion
- • Quality can only be assessed after execution

Generic View of Process

- Framework activities applicable to all software projects.

Software Engineering as Layered Technology

- Layers:
- • Quality Focus
- • Process
- • Methods
- • Tools

Quality Focus

- Foundation for continuous process improvement.

Process Framework Activities

- Communication
- Planning
- Modeling
- Construction
- Deployment

Umbrella Activities

- Project Tracking
- Risk Management
- SQA
- Technical Reviews
- Configuration Management

CMMI Introduction

- Capability Maturity Model Integration improves software process maturity.

CMMI Maturity Levels

- 1. Initial
- 2. Managed
- 3. Defined
- 4. Quantitatively Managed
- 5. Optimizing

Waterfall Model

- Sequential model: Requirements → Design → Coding → Testing → Maintenance

Spiral Model

- Risk-driven iterative model combining prototyping and systematic development.

Agile Methodology

- Incremental development focusing on customer collaboration and adaptability.

Comparison of Process Models

- Waterfall: Sequential
- Spiral: Risk-Oriented
- Agile: Adaptive & Iterative

Real-Time Examples

- Amazon – Agile
- Defense Systems – Spiral
- Government Payroll – Waterfall

Summary

- Review of Unit-I concepts and exam highlights.