

UNIT-2

- 1a) Software engineering systematically develops, tests, and maintains programs. categories include system, application, embedded, web, and AI-driven software for various functions.
- 1b) Software models: Waterfall (linear), Iterative (gradual refinement), V-model (testing included), Agile (adaptive), Spiral (risk-focused), RAD (fast-paced development).

UNIT-3

- 3a) SRS documents software requirements, functions, and constraints. It includes functional, non-functional needs and design limitations, ensuring clear communication and reduced errors.
- 3b) cohesion represents internal module strength, stronger cohesion enhances maintainability. Coupling refers to inter-module reliance, lower coupling improves flexibility and debugging.

UNIT-4

- 4a) function-oriented design breaks software into modules. It emphasizes modular structure, hierarchy, functional relationships, and process-driven organization often visualized with DFDs.
- 4b) UIs facilitate user-software interaction. Examples: GUI (Windows), CLI (Linux), VUI (Alexa), Touch (Smartphones). A well-designed UI improves experience and accessibility.

UNIT - IV

- 7a) Coding develops software as per specifications, code reviews examine quality, identify bugs, enhance readability, security, and reduce future debugging.
- 7b) Testing verifies software functionality. Types: Unit (modules), Integration (connections), System (full validation), Acceptance (user-based), plus Black-box, white-box, and performance testing.

UNIT - V

- 9a) Reliability ensures continuous function without failure, availability measures operational uptime. Reliable software reduces breakdowns, availability ensures accessibility.
- 9b) SQM maintains software standards via Quality Assurance (prevention), Quality control (detection), and Testing (verification). Uses ISO 19001, CMMI frameworks.