Lesson Plan for Software Engineering

S. No.	Lecture No.	Topic	Subtopics
1	L1	Introduction to Program vs. Software	Difference between program & software, characteristics of software
2	L2	Software Engineering Overview	Definition, scope, importance of software engineering
3	L3	Programming Paradigms	Procedural, object-oriented, functional, logical paradigms
4	L4	Software Crisis	Problems, causes, consequences
5	L5	Phases in Software Development	Requirement analysis, design, coding, testing, maintenance
6	L6	Software Development Process Models	Introduction to process models
7	L7	Waterfall Model	Features, advantages, limitations
8	L8	Prototype Model	Prototyping, benefits, challenges
9	L9	Evolutionary Model	Iterative development, incremental approach
10	L10	Spiral Model	Risk-driven model, applications
11	L11	Role of Metrics in Software Development	Importance, types of metrics, examples
12	L12	Feasibility Study	Economic, technical, operational feasibility
13	L13	Software Requirement Specification (SRS)	Definition, importance of SRS
14	L14	Characteristics of a Good SRS	Correctness, completeness, consistency, verifiability
15	L15	Components of SRS	Functional & non-functional requirements
16	L16	Problem Analysis in SRS	Identifying problems, requirement issues
17	L17	Requirement Gathering Tools	Interviews, questionnaires, observations, brainstorming
18	L18	Requirement Specification & Validation	Validation process, review techniques
19	L19	Metrics in Requirement Analysis	Requirement quality metrics
20	L20	Structured Analysis Tools – Data Flow Diagram	Symbols, examples, levels of DFD
21	L21	Data Dictionary	Structure, purpose, examples
22	L22	Decision Table & Decision Trees	Representation of logic, advantages
23	L23	Structured English	Role, examples

S. No.	Lecture No.	Topic	Subtopics
24	L24	Entity-Relationship (E-R) Diagrams	Entities, relationships, attributes
25	L25	Software Project Planning	Overview, importance
26	L26	Cost Estimation – COCOMO Model	Basic COCOMO, intermediate, detailed models
27	L27	Project Scheduling & Staffing	Gantt charts, PERT/CPM, team organization
28	L28	Software Configuration Management	Version control, baseline, tools
29	L29	Software Quality Assurance & Project Monitoring	QA plans, project monitoring techniques
30	L30	Risk Management	Types of risks, risk analysis, mitigation
31	L31	Software Design Fundamentals	Principles, objectives, problem partitioning
32	L32	Design Methodologies	Structured design, object-oriented design
33	L33	Cohesion & Coupling	Types of cohesion, types of coupling, importance
34	L34	Software Testing Strategies	Unit, integration, validation, system, alpha & beta testing
35	L35	Software Maintenance	Types, management, process, maintenance characteristics

Unit I – Introduction & Software Development Process Models

1. Differentiate between a program and software with suitable examples.
2. Define Software Engineering. Why is it important in modern system development?
3. Explain the concept of Software Crisis. What were its major causes?
4. List and explain the phases of the Software Development Life Cycle (SDLC).
5. Compare the Waterfall Model and Prototype Model with merits and demerits.
6. Describe the Evolutionary model with its advantages and limitations.
7. What is the Spiral Model? How does it help in risk management?
8. Discuss the role of metrics in software development. Give examples.
9. Explain why programming paradigms are important in software development.
10. Write a short note on the differences between coding, testing, and maintenance phases.

Unit II – Feasibility Study, SRS, and Structured Analysis Tools

1. What is a feasibility study? Explain its types.
2. Define SRS. Why is it needed in software engineering?
3. Discuss the essential characteristics of a good SRS document.
4. List the major components of SRS with examples.
5. What are the common problems faced during requirement analysis?
6. Explain different requirement gathering techniques with their merits.
7. Describe requirement validation and the role of metrics in validation.
8. Draw and explain a Data Flow Diagram (DFD) for a library management system.
9. Explain the purpose and advantages of using a data dictionary.
10. Differentiate between decision tables, decision trees, and structured English with examples.

Unit III – Software Project Planning & Design

1. Define software project planning. Why is it necessary?
2. Explain the COCOMO model for cost estimation with an example.
3. What are the major steps involved in project scheduling?
4. Discuss different team structures in software project management.
5. Define software configuration management. Why is it important?
6. Explain different quality assurance activities in software development.
7. How is risk management performed in software projects? Give an example.
8. What are the fundamentals of software design?

9. Differentiate between cohesion and coupling with suitable examples.
10. Explain various design methodologies used in software engineering.

Unit IV – Software Testing and Maintenance

1. Define software testing. Why is it necessary?
2. Differentiate between unit testing and integration testing.
3. What is system testing? How does it differ from validation testing?
4. Explain the concepts of alpha testing and beta testing with examples.
5. Discuss different software maintenance types with examples.
6. What are the major challenges in software maintenance?
7. Describe the maintenance process in detail.
8. Define regression testing. Why is it important in maintenance?
9. Compare corrective, adaptive, and perfective maintenance.
10. Explain the characteristics of good software maintenance practices.