**Name of Department:- Computer Science and Engineering**

1. Subject Code: TCS 510 Course Title: **Software Engineering**
2. Contact Hours: L: 3 T: - P: 0
3. Semester: V

4. Pre-requisite: Basics of Programming

5. Course Outcomes: After completion of the course students will be able to

1. Understand Software Development Life Cycle and importance of engineering the software.
2. Development of efficient software requirement specification for desired product.
3. Compare various software development methodologies ad conclude on their applicability in developing specific type of product.
4. Construct an efficient design specification document for attainment of user desired product.
5. Develop applications using the concepts of various phases of software development life cycle.
6. Study various software testing techniques and identify their relevance to developing a quality software.

7. Detailed Syllabus

| UNIT | CONTENTS | Contact Hrs |
|---|---|---|
| **Unit – I** | Introduction:What is Software Engineering and its history, Software Crisis, Evolution of a Programming System Product, Characteristics of Software, Brooks' No Silver Bullet, Software Myths<br>Software Development Life Cycles: Software Development Process, The Code-and-Fix model, The Waterfall model, The Evolutionary Model, The Incremental Implementation, Prototyping, The Spiral Model, Software Reuse, Critical Comparisons of SDLC models, An Introduction to Non-Traditional Software Development Process: Rational Unified Process, Rapid Application Development, Agile Development Process | **10** |
| **Unit - II** | **Requirements:** Importance of Requirement Analysis, User Needs, Software Features and Software Requirements, Classes of User Requirements: Enduring and Volatile; Sub phases of Requirement Analysis, Functional and Non-functional requirements; Barriers to Eliciting User Requirements, The software requirements document and SRS standards, Requirements Engineering, Case Study of SRS for a Real Time System<br>Tools for Requirements Gathering: Document Flow Chart, Decision Table, Decision Tree; Structured Analysis: DFD, Data Dictionary, Introduction to non-traditional Requirements Analysis Tools: FSM, Statecharts and Petrinets; | **9** |
| **Unit – III** | **Software Design:** Goals of Good Software Design, Design Strategies and Methodologies, Data Oriented Software Design, Structured Design: Structure Chart, Coupling, Cohesion,, Modular Structure, Packaging; Object | **8** |

| | | |
|---|---|---|
| | Oriented Design, Top-Down and Bottom-Up Approach, Design Patterns<br>Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.<br>**Development:** Selecting a Language, Coding Guidelines, Writing Code, Code Documentation | |
| **Unit – IV** | **Testing:** Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards, Automated Testing | **10** |
| **Unit – V** | **Software Maintenance and Software Project Management**: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance:Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.<br>Software Quality Assurance: SQA Plans, ISO 9000 models, SEI-CMM Model | **8** |
| | **Total** | **45** |

**Text Books:**


1. R. S. Pressman," Software Engineering: A Practitioners Approach", McGraw Hill.
2. P.K.J. Mohapatra," Software Engineering (A Lifecycle Approach)", New Age International Publishers


**Reference Books:**


1. Ian Sommerville," Software Engineering", Addison Wesley.
2. Pankaj Jalote:" An Integrated Approach to Software Engineering", Narosa Publishing House.
3. Carlo Ghezzi, M. Jarayeri, D. Manodrioli," Fundamentals of Software Engineering", PHI Publication.
4. Rajib Mall," Fundamentals of Software Engineering", PHI Publication.
5. Pfleeger, "Software Engineering", Macmillan Publication.