

CP309: SOFTWARE ENGINEERING
CREDITS = 5 (L=3, T=0, P=2)

Course Objective:

To impart knowledge of Software Engineering for development of software product from inception to deployment.

Teaching and Assessment Scheme:

| Teaching Scheme | | | Credits | Assessment Scheme | | | | |
|-----------------|---|---|---------|-------------------|----|-----------|----|-------------|
| L | T | P | C | Theory | | Practical | | Total Marks |
| | | | | ESE | CE | ESE | CE | 150 |
| 3 | 0 | 2 | 5 | 70 | 30 | 30 | 20 | |

Course Contents:

| Unit No. | Topics | Teaching Hours |
|----------|---|----------------|
| 1 | <u>Software process Models and lifecycle:</u> Software Product, Software Processes, Evolving Role of Software, A Crisis on the Horizon and Software Myths, Software Engineering: A Layered Technology, Study of different Software Process Models, Linear Sequential Model, Prototyping Model, RAD Model, Evolutionary Process Models, Component-Based Development, Process, Product and Process, Object Oriented Software Engineering. | 05 |
| 2 | <u>Project Management Concepts & Project Metrics:</u> The Management Spectrum, People, Product, Process, Project, The W5HH Principle, Metrics in the Process and Project Domains (FP & LOC), Software Measurement, Metrics for Project and Software Quality. | 06 |
| 3 | <u>Software Project Planning, Scheduling and Tracking:</u> Project Planning Objectives, Software Project Estimation using COCOMO Model, Software Scope and Resources, Empirical Estimation Models, Automated Estimation Tools, Basic Concepts and Relationship Between People and Effort, Defining a Task Set for the Software Project, Selecting Software Engineering Tasks, Defining a Task Network and Scheduling, Earned Value Analysis and Error Tracking. | 08 |

4 **Software Requirements Specification, Analysis and Design:**

Requirement Gathering and Analysis, Software Requirement Specification(SRS), 10
Formal requirements specification and verification - axiomatic and algebraic specifications, The Elements of the Analysis Model, Data Modeling, Functional Modeling and Information Flow, Behavioral Modeling and Structured Analysis, Software Design and Software Engineering, The Design Process, Design Principles, Design Concepts, Modular Design, Design Heuristics for Effective Modularity, The Design Model ,Design Documentation, Function oriented v/s object-oriented design, Object Modeling using UML, Software Architecture and Data Design, Architectural Styles, Analyzing Alternative Architectural Designs, Mapping Requirements into a Software Architecture.

5 **User Interface Design and Risk Management:**

User Interface Design, Task Analysis and Modeling, Interface Design Activities 08
and Implementation Tools, Design Evaluation, Structured Programming and Comparison of Design Notation, Reactive versus Proactive Risk Strategies, Software Risks (Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation), Risks Monitoring and Management, Introduction of CASE tools.

6 **Coding, Software Testing Techniques & Software Testing Strategies:**

Software Testing Fundamentals and Test Case Design, White-Box Testing and 08
Black-Box Testing, ISO/IEC/IEEE Software Testing standards, Testing for Specialized Environments, A Strategic Approach to Software Testing and Issues, Unit Testing, Integration and Validation Testing, System Testing, Software Documentation and Debugging Techniques, Testing tools like Win Runner.

TOTAL 45

List of References:

1. Roger S. Pressman, “*Software Engineering: A practitioner's approach*”, McGraw Hill.
2. Rajib Mall, “*Fundamentals of Software Engineering*”, Prentice Hall India.
3. Pankaj Jalote, “*An integrated approach to Software Engineering*”, Springer.
4. Ian Sommerville, “*Software Engineering*”, Addison and Wesley.

Course Outcomes (COs):

1. Implement software life cycle model for Software development
2. Develop Software Requirement Specification
3. Create Software Design elements (class diagram, activity diagram, Use case, etc)
4. Apply software testing principles and recognize software product quality parameters.
5. Manage software projects from inception to deployment.
6. Understand QA and Risk parameters of Software engineering.