UCS507: SOFTWARE ENGINEERING

Introduction: Introduction to Software Engineering, importance of Software, The Software Evolution, Software Characteristics, Software Applications, Software Crisis: Problem and Causes.

Software Processes: Software process models (Waterfall, Incremental, and Evolutionary process models and Agile), software quality concepts, process improvement, software process capability maturity models, Personal Software process and Team Software Process, Overview of Agile Process.
Tools and Environments: CASE and its Scope, CASE support in Software Life Cycle, Documentation Support, Architecture of CASE Environment.

Requirements Engineering: Problem Analysis, Requirement elicitation and Validation, Requirements modeling: Scenarios, Information and analysis classes, flow and behavioral modeling, documenting Software Requirement Specification (SRS).

Software Design: System design principles: levels of abstraction (architectural and detailed design), separation of concerns, information hiding, coupling and cohesion, Structured design (top-down functional decomposition), object-oriented design, event driven design, component-level design, data-structured centered, aspect oriented design , function oriented, service oriented, Design patterns.

Software Construction: Coding Practices: Techniques, mechanisms for building quality programs including Secure Coding Practices, Integration Strategies, Internal Documentation, Verification.
Software Verification and Validation: Levels of Testing, Functional Testing, Structural Testing, Test Plan, Test Case Specification, Software Testing Strategies, Verification & Validation, Unit, Integration Testing, Top Down and Bottom Up Integration Testing, Alpha & Beta Testing, White box and black box testing techniques, System Testing and Debugging.

Software Quality Assurance: Software Quality Control and Quality Assurance, ISO 9000 Certification for Software Industry, SEI CMM and Comparison of ISO & SEICMM.

Software Evolution: Software development in the context of large, pre-existing code bases, Software evolution, Characteristics of maintainable software, Software Reengineering, Software reuse.

Formal Methods: Role of formal specification and analysis techniques in the software development cycle. Formal approaches to Software Modeling and Analysis.

Technical Metrics for Software: A Framework for Technical Software Metrics, Metrics for the Analysis Model, Metrics for Design Model, Metrics for Source Code, Testing and Maintenance.

Laboratory work: Implementation of Software Engineering concepts using tools like Rational Suite etc. Exposure to CASE tools like Rational Software suit, Turbo Analyst, Silk Suite.

Text Books:
1. S. Pressman R., Software Engineering, A Practitioner's Approach, McGraw Hill International (2014), 8th ed.
2. Sommerville I., Software Engineering, Addison-Wesley Publishing Company (2010,) 9th ed.

Reference Books:
1. Jalote P., An integrated Approach to Software Engineering, Narosa (2005).

2. James F. Peter, Software Engineering - An Engineering Approach, John Wiley (2004)