Software Engineering

Course Code: BECSE2C025 /BECCS2C025
Course Title: Software Engineering
Semester: IV
Credits: 04(03 Theory and 01 Lab)

Rationale

This course provides an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.

Course Outlines:

Contents	**No. of Lectures**
UNIT–I Introduction to Software Engineering: Defining software, Software Applications, Legacy Software, Software myths. A Generic view of process: Software Engineering- a Layered Technology, A Process Framework, Process Assessment and Improvement. Process models: The Waterfall, Prototyping, Evolutionary, and Spiral Models. Agility and Process: Introduction to Agile Process, Scrum, Other Agile Frameworks: The XP Framework, Devops.	8
UNIT–II Software Requirements: Requirements Engineering, Functional and Non-Functional Requirements, User Requirements, System Requirements, Interface Specification, Software Requirement Specification Document, Feasibililty Studies, Requirements Elicitation. Scenario-Based Modeling: Actors and User Profiles, Creating Use Cases, Documenting Use Cases, Class Based Modeling: Identifying Analysis Classes, Defining Attributes and Operations, UML Class Models, Functional Modeling: A Procedural View, UML Sequence Diagrams, BehavioralModeling: UML State Diagrams, UML Activity Diagrams.	10
UNIT–III Design Engineering: Design process, Software Quality Guidelines, Basic issues in Software Design, Modularity, Cohesion, Coupling and Layering Software Metrics: Introduction to Software Metrics, Token Count Project Planning: Size Estimation, Cost Estimation Models, Models: COCOMO, PUTNAM Resource Allocation Model. Software Process Improvement: Software Security Engineering, Secure Development Life Cycle Models, Approaches to SPI, Maturity Models, The SPI Process, The CMMI, Coding Standards and Code Review Techniques, PSP and Six Sigma.	8

UNIT–IV Testing Strategies: Fundamentals of Software Testing, Test Coverage Analysis and Test Case Design Techniques, Back-box and White-box testing, Verification and Validation, System Testing, The Art of Debugging, Integration Testing: Top-Down Integration, Bottom-Up Integration, Continuous Integration, Artificial Intelligence and Regression Testing. Software Project Management: Project Planning and Control, PERT and GANTT Charts.	8
UNIT–V Risk management: Reactive Vs Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, RMMM plan. Quality Management: Quality Concepts, Software Quality Assurance, Software Reviews, Formal Technical Reviews, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards.	8

Course objectives & learning outcomes:

Upon successful completion of this course, candidates will be able to:

1. Translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Have experience and/or awareness of testing problems and will be able to develop a simple testing report.

Text Books/ Reference books

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.