CSE320:SOFTWARE ENGINEERING

L:3 T:0 P:0 Credits:3

Course Outcomes: Through this course students should be able to

- Plan and deliver an effective software engineering process, based on knowledge of widely used development life cycle models.
- Construct implementable design from requirement specification, following a structured and organised process.
- Translate a requirements specification into an implementable design, following a structured and organised process.
- Formulate a testing strategy for a software system, employing test case design techniques such as functional and structural testing.
- Analyze project including planning, scheduling, estimation and configuration management.

Unit I

Introduction to software engineering : Evolution and impact of software engineering, Software life cycle models, Waterfall model, Prototyping model, Evolution and spiral models, Feasibility study, Functional and non-functional requirements, Requirement gathering, Requirement analysis and specification

Unit II

Issues in software design : Basic issues in software design, Modularity, Cohesion, Coupling and layering, Function oriented software design, Data flow diagram and structure chart

Unit III

Object modelling : User interface design, unified process, Object modelling using UML, use case model development, Coding standards and code review techniques

Unit IV

Testing : Fundamentals of testing, Black box testing techniques, White box testing techniques, Levels of testing, Test cases

Introduction to selenium : Feature of selenium, Versions of selenium, Record and play back

Unit V

Software project management : Project managment, Project planning and control, Cost estimation, Project scheduling using PERT and GANTT charts, Software configuration management

Unit VI

Quality management : Quality management, ISO and SEI CMMI, PSP and six sigma, Computer aided software engineering, Software maintenance, Software reuse, Component based software development

Advance techniques of software engineering : Agile development methodology, Scrum, Aspect oriented programming, Extreme Programming, Adaptive software development, Rapid application development (RAD), Software coloning

Text Books:

1. FUNDAMENTALS OF SOFTWARE ENGINEERING by RAJIB MALL, PRENTICE HALL

References:

1. SOFTWARE ENGINEERING by IAN SOMMERVILLE, PEARSON

2. SOFTWARE ENGINEERING:A PRACTITIONER APPROACH by ROGER S.PRESSMAN, MCGRAW HILL EDUCATION

3. SOFTWARE ENGINEERING FUNDAMENTALS by ALI BEHFOROOZ AND FREDERICKS J. HUDSON, OXFORD UNIVERSITY PRESS