CS401 Software Engineering

Prerequisites:
CS304

Course Contents:

Introduction: **(5%)**

What is Software Engineering? History and motivation and challenges. Software Quality. Software process. Characteristics of software process. Software development processes and methodologies: Waterfall, prototyping, iterative, spiral, unified process, agile methodologies.

Requirements: **(15%)**

Elicitation, specification, verification, management. Developing vision documents, identifying features, relating requirements to features. Functional and nonfunctional requirements.(Techniques such as Interviewing, Requirement workshop, brainstorming, prototyping). SRS, modern SRS, Characteristics of SRS. User stories-specification, prioritization, estimation and management.

Human Computer Interaction: **(10%)**

Goal-directed design, user goals, goal directed design process. Task analysis. Implementation models and mental models. User categories, contextual inquiry, personas, interaction styles, orchestration and flow, navigation, principles of visual interface design and hiding, metaphors, idioms and affordances, direct manipulation, controls, error handling, help. Usability testing.

Design & Coding: **(20%)**

TDD, Refactoring, BDD.

Design patterns: **(20%)**

Reusability at design level. Principles of good design. Creational, structural and behavioral patterns.(some of the patterns to be covered)

Verification and validation: **(10%)**

Reviews-walkthroughs, peer reviews, inspection, Testing. Types of test: unit testing, integration testing, system testing, acceptance testing. Test planning, test design, test case execution and reporting. Debugging: principles, approaches, use of debuggers.

Maintenance: **(5%)**

The Context of Maintenance – Definitions -- Economics of Maintenance – Evolution of Software Products -- Maintaining Systems Effectively -- Categorizing Software products Deployment Models – Types of maintenance. Reverse engineering, re-engineering, BPR. Legacy systems.

Project Management using Scrum/ Lean approach: **(10%)**

Project planning and monitoring. Team management. Retrospectives.

Software configuration management: **(5%)**

Baseline, identification, accounting, control, audit, source and version control, change control procedure. Tools used in SCM

'Main Reading

1. Dean Leffingwell, Managing Software Requirements, Pearson Education.
2. Martin Fowler, Refactoring, Pearson Education.
3. Erich Gamma, Richard Helm, Ralph Johnson,, John Vlissides, Design Patterns: Elements of Reusable Object-oriented Software, Pearson Education.
4. Joshua Kareivesky, Refactoring to Patterns, Pearson Education
5. Steve McConnell, Code Complete, 2nd Edition. Redmond, Wa.: Microsoft Press, 2004

6. Pankaj Jalote, An Integrated Approach to Software Engineering, Third Edition, Narosa Publishing House
7. Alan Cooper & Robert Reimann, About Face 2.0: The Essentials of Interaction Design, Wiley