

Approval: 4th Senate Meeting

Course Name	: Software Engineering
Course Number	: CS-303
Credits	: 2-0-2-3
Prerequisites	: None.
Intended for	: 3 rd , 4 th year B. Tech.
Distribution	: Discipline-elective for CSE B.Tech; Free elective for other B. Tech. students
Semester	: Odd, Even

Preamble: This course builds upon topics covered in CS core courses like Mathematical Foundations of Computer Science, Information Systems, and Advanced Data Structures and Algorithms. It provides an in-depth coverage of topics that are crucial for development of industrial strength software systems. Such systems consist of thousands of lines of code and where execution is done in distributed-transnational teams. The coverage of topics in lectures is accompanied by a hands-on experience on real-world case studies. These case studies touch upon the analysis and design components of industrial strength systems.

Course Outline:

This course will cover the techniques for development of software systems, commonly referred to as “Software Engineering.” Software Engineering is essential for executing any software project and requires skills in two dimensions – engineering and project management. Engineering deals with issues of architecture, design, coding, and testing; whereas, project management deals with planning, monitoring, and risk management. This course is intended to give students both knowledge about, and practical experience in, engineering and project management areas. Briefly course topics will be cover: qualities; software development models; requirement specifications; software risk management; software design; software testing; and, software project management. Although a CS elective, this course is a “must have” for those planning to work in the IT industry after graduation.

Modules:

Module 1: Software Qualities and Cooperation in Global Software Development (6hours)

How different qualities impact the development approach and process: usability, performance, and safety and reliability. The basis for the project establishment, the choice of communication channels, tools for sharing documents, and the collaboration protocol. Lab: Project begins; Project groups are formed and case studies are assigned.

Module 2: Software Development Models (6hours)

Organize software engineering work in a systematic manner. A number of development models are discussed: waterfall, V, RAD, Spiral, Incremental, and Agile (Scrum and XP). Lab: The project teams are expected to produce an overview of the development models and select one for their case study.

Module 3: Project Management (6hours)

Estimation, reporting and re-planning, stakeholders, organisation of the software development team, and different roles. Lab: The project teams are expected to produce a

project plan involving different software engineering activities corresponding to their development model. Also, they need to assign roles to different activities in their plan.

Module 4: Requirements Engineering and Risk Management (6hours)

Understand a complex use situation through rich pictures and UML class diagrams as means to analyze the problem domain. Discuss the documentation of requirements, their handling throughout the development, and their analysis with an emphasis on legal aspects of software. Managing software project risks, software risk management, elements of risk management, risk program strategy. Lab: The project teams are expected to make rich pictures, UML class diagrams, and gather and write their requirement specifications. Also, they need to identify the product and process risks and discuss mitigation strategy.

Module 5: Software Quality-in-Use and Configuration Management: (6hours)

Participatory design, future workshops, scenarios and mock-up based design and show how they feed into and expand object-oriented analysis and design methods. Also, discuss document conventions and draft a configuration management plan. Lab: Project teams make scenarios and mock-up based design. Also, they figure out how these methods help in object-oriented analysis and design methods. As part of this week's lab, the teams also create a configuration management plan and a document convention.

Module 6: Static and Dynamic Test (6hours)

Introduces quality assurance and static test techniques, especially peer reviews. Also, introduces into planning, processes, and techniques for dynamic testing. Lab: Project teams write test cases, send their documents for peer-review, and also create a quality assurance plan.

Module 7: Software Architecture, Design, and Process Qualities (6hours)

Use architectural design as a start for the software design and implementation. Software process quality and why it is important. Discuss the Capability - Maturity - Model and software process improvement. Lab: Project teams write software architecture on their projects and prepare portfolio for final submission.

Textbooks:

Pankaj Jalote. An Integrated Approach to Software Engineering, 2nd edition or higher, Narosa Publishing House. [JAL]

Ian Sommerville. Software Engineering, 9th edition. Pearson. [SOM]

Articles:

Articles from IEEE. I will either hand out photocopies or send them via email during the semester.