

CSE 0326: Software Engineering

Programme: B.Tech. (CSE)
Course: Program Core (CSE, CCE)

Year: 3
Credits: 3

Semester: 5th
Hours: 40

Context and Overview (100 words):

This course imparts the fundamental concepts, techniques, methodologies and best practices related to engineering software, and meeting the specified functional and quality requirements. Students will learn skills such as requirements elicitation and analysis, software architecture and design, testing and maintaining software, and basics of project management. Through a practical project work, students will have opportunities to practice key software engineering and project management techniques in small teams using selected tools and development environments.

Prerequisites Courses:

Data Structures and Algorithms

Course outcomes (COs):

CO1 familiarity; CO2, 3 and 5: usage; CO4: assessment

On completion of this course, the students will have the ability to:

CO1: describe the need for delivering quality software on time and on budget by engineering software, and different process models of engineering software with associated phases of software development life cycle;

CO2: gather software requirements and analyze those requirements so as to eliminate ambiguities, inconsistencies, incompleteness and other bad characteristics, and architect and design the software modules, and meet the functional and non-functional requirements;

CO3: perform software project management activities such as effort estimation, and project scheduling in small teams;

CO4: explain software validation and verification techniques including development of test plans and conducting different types of tests (unit, integration, system & regression levels);

CO5: apply the knowledge and techniques of engineering software in a team project covering all the phases of software development life cycle;

Keyword Syllabus:

Software development processes, Software project management, Tools and environments, Requirements engineering, Software design, User interface design, Software construction, Software verification and validation, Software evolution and reliability;

Detailed Syllabus:

| Topics | Unit (Hrs) |
|---|--|
| Waterfall, incremental, iterative and agile process models, SDLC, types of software systems, programming in the large vs. individual systems, comparison of process models | 3 (Software development processes) |
| Team processes and responsibilities, participation and conflict resolution, introduction to effort estimation, risk categories and managing the risk (including risk associated with virtual teams) | 2 (Software project Management) |
| Requirements analysis and design modeling tools, programming environments, software configuration management and version control, release management, continuous integration, tool integration mechanisms | 2 (Tools and environments) |
| Describing functional and non-functional requirements, software requirements elicitation methods; properties of requirements, evaluation and use of requirements specification; UML techniques such as use case diagrams, use case descriptions, activity diagrams and class diagrams | 8 (Requirements engineering) |
| Design principles such as abstraction, separation of concerns, information hiding, coupling and cohesion, and reuse; design paradigms (structured, OOAD, component design, service-oriented design, event-driven design); relationship between requirements and design; design patterns; component design | 8 (Software design) |
| Contexts for user interface design; processes for user-centered development; evaluation measures; usability heuristics and principles of usability testing; principles of good design and tradeoffs; principles of GUIs; elements of visual design; task analysis; low-fidelity prototyping; user interface standards | 8 (User interface design) |
| Coding practices, coding standards, Integration strategies, Development context – green field vs. existing code base | 2 (SE/Software construction) |
| Verification and validation concepts; inspections, reviews and audits; Testing fundamentals (types, test plan creation and test case generation, black-box and whitebox testing, regression testing and test automation); defect tracking, limitations | 4 (SE/Software verification and validation) |
| Software development in the context of large, pre-existing code bases; software evolution; characteristics of maintainable software; Reengineering systems; Software reuse | 3 (SE/Software evolution and reliability) |

Note:

- a) The number of hours allotted from each unit is specified along with the unit name.
- b) The order of the above topics may vary with the systems development methodology selected for teaching and project work.

Assessment:

| Item | Weightage |
|------------|-----------|
| Quiz 1 | 10 |
| Quiz 2 | 10 |
| Quiz 3 | 10 |
| Assignment | 10 |
| Mid Term | 25 |
| End Term | 35 |

Textbook references (IEEE format):**Text Books:**

1. Essential Scrum – A practical guide to the most popular agile process by Kenneth Rubin, Addison-Wesley, 2013
2. Software Engineering, Ian Sommerville, Pearson, 2017, 8th edition.
3. Software Engineering: A Practitioner's Approach by Roger Pressman, TMH, 6th / 7th Editions

Reference books:

4. Software Engineering by S.L. Pfleeger, MacMillan Publishing
5. Software Engineering by Ian Sommerville, Pearson Education LPE, 8th or 9th Edition
6. An Integrated Approach to Software Engineering by Pankaj Jalote, Narosa Publishing 2nd Edition

Additional Resources - online references related to software engineering and project management will be provided during the course.

Prepared By: Course Instructor name: Ravi Prakash Gorthi

Last Update: August 3, 2020