# 20CS2203 - SOFTWARE ENGINEERING

|  |  |  |  |
| --- | --- | --- | --- |
| Course Category: | Program Core | Credits: | 3 |
| Course Type: | Theory | Lecture-Tutorial-Practical: | 3-0-0 |
| Prerequisite: | Require the fundamental concepts of computers and basic analytical capabilities | Sessional Evaluation:  Univ. Exam Evaluation:  Total Marks: | 40  60  100 |
| Objectives: | * To define various software engineering phases. * Explore the concepts of software products and processes. * To facilitate the environment of software development in the outside world. * To expose the importance of risk management and strive for quality assurance. | | |

|  |  |  |
| --- | --- | --- |
| Course Outcomes | Upon successful completion of the course, the students will be able to: | |
| CO1 | Understand the software Process and various Process Models. |
| CO2 | Learn the Agile Development and understanding Requirements. |
| CO3 | Study the data modeling concepts to create a behavioral model and exposure on design concepts. |
| CO4 | Identify various architectural styles to get the support for designing conventional components. |
| CO5 | Examine different Testing Strategies for conventional software and metrics to evaluate the product. |
| CO6 | Study Software Configuration Management and improve software quality assurance. |
| Course Content | UNIT-I  **The Software Process**: A generic process model, Process Assessment and Improvement.  **Process Models:** The Waterfall model, Incremental process models, Evolutionary process models, Concurrent Models, The Unified process, Personal and Team Process models.  UNIT-II  **Agile Development**: Agility and the cost of change, What is an agile process?, Extreme Programming, Other Agile process models, A tool set for the Agile process.  **Understanding Requirements**: Requirements engineering, Eliciting requirements, Developing Use cases, Building the Requirements model, Negotiating Requirements, Validating requirements.  UNIT-III  **Requirements Modeling**: Requirements modeling approaches, Scenario based modeling, Data Modeling Concepts, Class-based modeling, Flow-oriented modeling, Creating a behavioral model.    **Design Concepts**: The Design process, Design concepts, The Design model.  UNIT-IV  **Architectural Design**: Software architecture, Architectural styles, Architectural design, Assessing Alternative Architectural Designs, Architectural Mapping using Data flow.  **Component-level Design**: What is a component?, Designing class-based components, Conducting component-level design, Designing traditional components, Component-Based Development.  UNIT-V  **Software Testing strategies**: A strategic approach to software testing, Test strategies for conventional software, Validation testing, System testing, The Art of debugging.  **Product metrics:** A Framework for Product metrics, Metrics for the Requirements Model, Metrics for the Design Model, Metrics for Source code, Metrics for Testing, Metrics for Maintenance.  UNIT-VI  **Software Configuration Management:** Software Configuration Management, The SCM Repository, The SCM Process.  **Software Quality Assurance:** Elements of Software quality assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan | |
| Text Books &  References  Books | **TEXT BOOKS:**   1. Software Engineering - A Practitioner’s Approach, Pressman R S, 7th edition, McGraw-Hill.   **REFERENCE BOOKS:**   1. Sommerville I, Software Engineering, 9th edition, Pearson Education. 2. Waman S Jawadekar, Software Engineering – Principles and Practice, McGraw-Hill, 2008. | |
| E-Resources | 1. <https://nptel.ac.in/courses> 2. <https://freevideolectures.com/university/iitm> | |