# 19CS2104 - 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:** | 4060100 |
| **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 basics of software engineering layers. |
| CO2 | Learn the engineering principles, planning and construction of analysis models. |
| 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 various risk strategies and improve software quality assurance. |
| **Course Content** | UNIT-I**Introduction to Software Engineering:** Software evolution, Software and its nature, Legacy software, Software myths.**A Generic View of Process:** Software engineering layers, Process frame work, Capability Maturity Model Integration (CMMI), Personal and team process models.**Process Models:** Prescriptive models, Waterfall model, Incremental process models, Evolutionary process models and Unified process.UNIT-II**Software Engineering Practice**: Engineering Principles, Communication, Planning, Modeling and Construction practices, Deployment.**Requirements Engineering:** Requirements engineering tasks, Initiation, Eliciting requirements, developing use-cases, Building the analysis model, Negotiating and validating requirements.UNIT-III**Building the Analysis Model:** Analysis model, Analysis modeling approaches, Data modeling concepts, Object-oriented analysis, Scenario based modeling, Flow-oriented modeling, Class-based modeling, Creating a behavioral model.**Design Engineering:** Design process, Design quality, Design concepts, and Design model.UNIT-IV**Creating an Architectural Design:** Software architecture, Data design, Architectural styles and patterns, Architectural design, Mapping data flow into software architecture.**Modeling Component-level Design:** Nature of component, Designing class-based components, Conducting component level design, Object constraint language, Designing conventional components.UNIT-V**Testing strategies:** A strategic approach to software testing, Test strategies for conventional software, Test strategies for object-oriented software, Validation testing, System testing, Art of debugging.**Product metrics:** Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.UNIT-VI**Risk management:** Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM plan.**Quality Management**: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability. |
| **Text Books and References:** | Text Books:1. Pressman R S, Software Engineering-A Practitioner‟s Approach, 6th edition, McGrawHill
 |
| Reference Books:1. Sommerville I, Software Engineering, 5th edition, Pearson Education, 1996.
2. Jawadekar W S, Software Engineering – Principles and Practice, Tata McGraw-Hill, 2004.Hill, 2005.
3. Carlo gezzi, Fundamentals of Software Engineering ,Second edition, Prentice Hall
 |
| **E-Resources** | 1. <https://nptel.ac.in/courses>
2. <https://freevideolectures.com/university/iitm>
 |