

**B. Tech. in COMPUTER SCIENCE AND ENGINEERING****III Year I Semester Syllabus (KR23)****SOFTWARE ENGINEERING (23CC501PC)****Common to CSE, IT, CSE (AI&ML) and CSE (DS)**

<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-requisites/ Co-requisites:**

1. 23CC305PC- OOPS through Java

**Course Objectives: The course will help to**

1. Analyze end-user needs to formulate detailed system requirements with UML, constructing a Software Requirements Document (SRD) that aligns with Agile, DevOps.
2. Analyze user requirements to determine an optimal process model, integrating Git and GitHub for proficient version control
3. Apply relevant software architectures and patterns for the high-level design of a system, integrating CI/CD processes through tools like Jenkins.
4. Apply various testing strategies to showcase understanding, while skillfully grasping Docker and Kubernetes concepts for project deployment.
5. Evaluate the essential role of risk management in project execution, integrating AWS cloud principles for strategic project deployment

**Course Outcomes: After learning the concepts of the course, the student is able to**

1. Identify and translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Illustrate appropriate process model depending on the user requirements understanding the concept of DevOps.
3. Apply appropriate software architectures and patterns to carry out the high-level design of a system, CI/CD process using Jenkins,
4. Demonstrate different testing strategies and develop a simple testing report
5. Apply new software models, techniques and technologies to bring out innovative and novelistic solutions for the growth of the society

**UNIT – I:**

**Introduction to Software Engineering:** The Nature of Software, Software Application Domains, Legacy Software, Software myths.

**Software Process Structure:** Software engineering- a layered technology, a process framework, Generic Processmodel.

**Process models:** The waterfall model, Incremental process models, evolutionary process models.

**Agile and DevOps:** Agile Process, Scrum agile methodology, Introduction and Benefits of working in a DevOps environment, DevOps Lifecycle: DevOps Stages, DevOps Delivery Pipeline.

## **UNIT – II:**

**Understanding Requirements:** Functional and non-functional requirements, user requirements, system requirements, software requirements document.

**Requirements engineering process:** Feasibility studies, Requirements elicitation, and analysis, requirements monitoring, and validation.

**Git & GitHub:** Version control, Branches, Merge, Revert, Fork and working with remote repository.

## **UNIT – III:**

**Design Engineering:** The design process and design quality, design concepts, the design model.

**Architectural design:** software architecture, architectural styles and patterns, context model.

**Conceptual model of UML:** class diagrams, sequence diagrams, use case diagrams, component diagrams.

**Build Tools and Continuous Integration using Jenkins:** Introduction to Maven - Installation and configuration, Jenkins Architecture, Build Pipeline project using Jenkin script.

## **UNIT – IV:**

**Testing Strategies:** A strategic approach to software testing, Verification and validation, Software Test strategy, TestStrategies for conventional software, black- box and white-box testing, validation testing, system testing, the art of debugging.

**Continuous Deployment:** Understand Containerization, Handling Docker Containers- Docker Image, Building and pushing images to Docker Hub, Orchestrating Containers- Setup Docker Compose, Deploy a multi- container application using Docker Compose.

**Orchestration using Kubernetes** - Understand Container Orchestration, Kubernetes Core Concept, Deploy Pods, Create Deployments to manage Pods. Introduction to Continuous Monitoring using Nagios.

## **UNIT-V:**

**Risk management:** Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

**Introduction to cloud and its services:** Cloud, cloud services, Understand and create cloud infrastructure using AWS. Deployment of the web application on to the cloud.

## **TEXTBOOKS:**

1. Software Engineering, A practitioner's Approach-Roger S. Pressman, 7<sup>th</sup> edition, McGraw Hill International Edition, 2010.
2. The unified modeling language user guide Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education, 2005.
3. Learning DevOps, 2<sup>nd</sup> Edition, by Mikael Krief, PackT Publications.
4. Learning Docker, 2<sup>nd</sup> Edition by Jeeva S. chelladurai, Vinod Singh, Pethuru Raj by PackT publications.

**REFERENCE BOOKS:**

1. Software Engineering, an Engineering approach-James F.Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice-Waman S Jawadekar,The McGraw-Hill Companies.
3. <https://kubernetes.io/docs/tutorials/hello-minikube/>
4. <https://minikube.sigs.k8s.io/docs/start/>
5. <https://www.jenkins.io/doc/>
6. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>
7. Introducing Maven by, Balaji Varanasi and Sudha Belida, APRESS publications.
8. Practical Devops, Second Edition. By Joakim Verona. Ingram short title; 2nd edition (2018).