# https://lh3.googleusercontent.com/58wn6-yfOr1YycMm79hHe35ARNqXLIxeOCE816abKgbHlvfwWL3D_S_viovStAJ4n-KSMAuUYMbSyK-dHZ67G5EcG1i9LgYdz5MjjSTwyK7sK1Cmbo_ylI2DsFtKfeZ0DkAqPFv0C:\KMIT EXAMINATION BRANCH-AUTONOMOUS\DESK TOP DOCUMENTS\kmit-bar.pngKESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

## B. Tech. III Year I Semester Course Syllabus (KR20) SOFTWARE ENGINEERING (CS502PC)

**Common for CSE, CSE (DATA SCIENCE), CSE (AIML), IT**

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **3** | **0** | **0** | **3** |

## Prerequisite/ Corequisite: NILL

**Course Objectives:**

1. Understand the basic Software engineering methods, practices and application and a general understanding of software process models like Agile and DevOps.
2. Analysis of software requirements, the SRS documents, data models, object models, context models, behavioural models and DevOps tools including Git and GitHub.
3. Understand the applying DevOps tools like Jenkins for continuous integration in Design Engineering concepts.
4. Understand the knowledge of testing strategies and how continuous deployment and monitoring happens using Dockers and Kubernetes.
5. Understand the interpretation of software risks and the idea of applying AWS cloud service for deployment on cloud.

**Course Outcomes:** After learning the concepts of the course, the student will be 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 evolving role of software, changing nature of software, software myths.

**Software Process Structure:** Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI), process assessment, personal and team process models.

**Process models:** The waterfall model, incremental process models, evolutionary process models, the unified process.

**Introduction to Agile and DevOps**: Agile versus traditional method comparison, various agile methodologies, 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, the software requirements document.

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

**System models:** Context models, behavioral models, data models, object models.

## UNIT - III

**Design Engineering:** The design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, a conceptual model of UML, class diagrams, sequence diagrams, use case diagrams, component diagrams**.**

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

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

## UNIT – IV

**Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

**Continuous Deployment:** Containerization using Docker - Understand Containerization, Docker Image and Docker CLI Commands, 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 **Continuous Monitoring** : Introduction to Continuous Monitoring using Nagios

**Product metrics:** Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

## UNIT-V

**Estimation of Software Projects:** Software project estimation, Decomposition techniques, Cost Estimation Models, and the make/buy decision

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

**Deployment on cloud** : Understand and create cloud infrastructure using AWS, and deploy the web application.

## Text Books:

1. Software Engineering, A practitioner’s Approach-Roger S. Pressman, 6thedition, McGraw Hill International Edition.
2. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations
4. Cloud Native DevOps With Kubernetes by John Arundel

## References:

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. The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Microservices, 1st Edition, Viktor Farcic, CreateSpace Independent Publishing Platform publications, 2016
4. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation, 1st Edition, Jez Humble and David Farley, 2010.
5. [http://khitguntur.ac.in/images/syllabus/btech/B. Tech. %20R19%204- 2%20sem.pdf](http://khitguntur.ac.in/images/syllabus/btech/B.Tech%20R19%204-2%20sem.pdf).

# https://lh3.googleusercontent.com/58wn6-yfOr1YycMm79hHe35ARNqXLIxeOCE816abKgbHlvfwWL3D_S_viovStAJ4n-KSMAuUYMbSyK-dHZ67G5EcG1i9LgYdz5MjjSTwyK7sK1Cmbo_ylI2DsFtKfeZ0DkAqPFv0C:\KMIT EXAMINATION BRANCH-AUTONOMOUS\DESK TOP DOCUMENTS\kmit-bar.pngKESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Hyderabad

## B. Tech. III Year I Semester Course Syllabus (KR20) SOFTWARE ENGINEERING LAB (CS505PC)

**Common for CSE, CSE (DATA SCIENCE), CSE (AIML), IT**

|  |  |  |  |
| --- | --- | --- | --- |
| **L** | **T** | **P** | **C** |
| **0** | **0** | **3** | **1.5** |

## Prerequisites/ Corequisites:

1. CS203ES Course
2. CS403PC Course
3. CS502PC Course

## Course Objectives

1. Understand the process of development of problem statement.
2. Understand the process of development of Software Requirement Specifications
3. Understand the process of development of Design documents and testing phase related documents.
4. Understand the usage of GitHub and Jenkins
5. Understand the deployment of project in AWS cloud using Docker and Kubernetes.

**Course Outcomes:** After learning the concepts of this course, the student will be able to

1. Outline to translate end-user requirements into system and software requirements
2. Illustrate a high-level design of the system from the software requirements
3. Use case tools in design phase of application development.
4. Use Jenkins to build project
5. Setup a project in AWS cloud using Dockers and Kubernetes.

## List of Experiments:

Do the following exercises for any one project given in the list of sample projects or any other projects?

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Study and usage of any Design phase CASE tool
4. Creating static pages of the project and committing using Git and GitHub
5. Building the project in Jenkins
6. Deploying the project in AWS cloud using Docker and Kubernetes
7. Develop test cases for unit testing and integration testing

## Sample Projects:

1. Book Bank
2. Online course reservation system
3. E-ticketing
4. Recruitment system

## Text Books:

1. Software Engineering, A practitioner’s Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition, 2015.
2. Software Engineering- Sommerville, 7th edition, Pearson Education, 2017.
3. The unified modeling language user guide Grady Brooch, James Rumbaugh, Ivar Jacobson, Pearson Education, 2016.
4. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, 2015.

## Reference Books:

1. Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale, 2018.
2. Cloud Native DevOps with Kubernetes by John Arundel, 2016.