

*A*  
**COURSE FILE**  
*ON*  
**“DEVOPS”**

**III B-Tech I Semester**



**COMPUTER SCIENCE & ENGINEERING**

**CMR ENGINEERING COLLEGE**

**KANDLAKOYA (V), MEDCHAL (M), R.R.DIST.**

**A.Y 2023-24**

## **CONTENTS OF COURSE FILE:**

1. Department vision & mission
2. List of PEOs, POs, PSOs
3. List of Cos (action verbs as per blooms)
4. Syllabus copy and suggested or reference books
5. Individual Time Table
6. Session plan/ lesson plan
7. Session execution log
8. Lecture notes
9. Assignment Questions (samples)
10. Mid exam question papers (samples)
11. Scheme of evaluation
12. Mapping of Cos with Pos and PSOs
13. Cos,Pos,PSOs Justification
14. Attainment of Cos, Pos and PSOs (Excel sheet)
15. Previous Years Question papers
16. Power point presentations (PPTs)
17. Innovative Teaching Methods
18. Websites or URLs e- Resources

**Submitted By**

**Mohammed Azhar,**

**Asst.Professor,**

**CSE Dept**

## **1. DEPARTMENT VISION & MISSION**

### **VISION OF THE DEPARTMENT**

To produce globally competent and industry-ready graduates in Computer Science & Engineering by imparting quality education with the know-how of cutting-edge technology and holistic personality.

### **MISSION OF THE DEPARTMENT**

1. To offer high-quality education in Computer Science & Engineering in order to build core competence for the graduates by laying a solid foundation in Applied Mathematics and program framework with a focus on concept building.
2. The department promotes excellence in teaching, research, and collaborative activities to prepare graduates for a professional career or higher studies.
3. Creating an intellectual environment for developing logical skills and problem-solving strategies, thus developing, an able and proficient computer engineer to compete in the current global scenario.

### **2.1 Program Educational outcome (PEOs):**

**PEO 1:** Excel in professional career and higher education by acquiring knowledge of mathematical computing and engineering principles.

**PEO 2:** To provide an intellectual environment for analyzing and designing computing systems for technical needs.

**PEO 3:** Exhibit professionalism to adapt current trends using lifelong learning with legal and ethical responsibilities.

**PEO 4:** To produce responsible graduates with effective communication skills and multidisciplinary practices to serve society and preserve the environment.

**2.2 Program Outcomes (POs):**

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **2.3 Program Specific Outcomes (PSOs):**

**PSO1: Professional Skills and Foundations of Software development:** Ability to analyze, design and develop applications by adopting the dynamic nature of Software developments.

**PSO2: Applications of Computing and Research Ability:** Ability to use knowledge in cutting edge technologies in identifying research gaps and to render solutions with innovative ideas.

### **NBA Graduate Attributes**

PO1 Engineering knowledge

PO2 Problem analysis

PO3 Design/development of solutions

PO4 Conduct investigations of complex problems

PO5 Modern tool usage

PO6 The engineer and society

PO7 Environment and sustainability

PO8 Ethics

PO9 Individual and team work

PO10 Communication

PO11 Project management and finance

PO12 Life-long learning

### 3. Mapping of Course/ Program Outcomes to Program Educational Objectives

#### COURSE OUTCOMES:

CO's	Description
<b>CO1</b>	<b>Explain</b> traditional software development methodologies like a waterfall.
<b>CO2</b>	<b>Apply</b> the agile methodologies and comparing various other software development models with agile.
<b>CO3</b>	<b>Explain</b> implementing continuous integration and continuous delivery.
<b>CO4</b>	<b>Explain</b> CAMS for DevOps (Culture, Automation, Measurement, and Sharing)
<b>CO5</b>	<b>Create</b> quick MVP prototypes for modules and functionalities.

#### 4. Syllabus copy

R20 B.TECH

#### UNIT -I

##### UNIT I:

##### TRADITIONAL SOFTWARE DEVELOPMENT

The Advent of Software Engineering, Waterfall method, Developers vs IT Operations conflict

##### UNIT II:

##### RISE OF AGILE METHODOLOGIES

Agile movement in 2000, Agile Vs Waterfall Method, Iterative Agile Software Development, Individual and team interactions over processes and tools, Working software over comprehensive documentation, Customer collaboration over contract negotiation, Responding to change over following a plan

##### UNIT III:

##### DEFINITION OF DEVOPS

Introduction to DevOps, DevOps and Agile.

##### UNIT IV:

##### PURPOSE OF DEVOPS

Minimum Viable Product, Application Deployment, Continuous Integration, Continuous Delivery

##### UNIT V:

##### CAMS (CULTURE, AUTOMATION, MEASUREMENT AND SHARING)

CAMS – Culture, CAMS – Automation, CAMS – Measurement, CAMS – Sharing, Test-Driven Development, Configuration Management, Infrastructure Automation, Root Cause Analysis, Blamelessness, Organizational Learning.

### TEXT BOOKS:

1. Dev Ops – Volume 1 , Pearson and Xebia Press

### REFERENCE BOOKS:

1. The DevOps Handbook - Book by Gene Kim, Jez Humble, Patrick Debois, and Willis
2. What is DevOps? - by Mike Loukides

## 5.Individual Time Table

Mr.MD.Azhar

	I(9:10-10:10)	II(10:10-11:00)	III(11:00-11:50)	IV(11:50-12:40)		V(1:20-2:20)	VI(2:20-3:10)	VII(3:10-4:00)	
MON									DEVOPS-III-D
TUE									
WED									
THU									
FRI		DEVOPS-III-C					DEVOPS-III-B	DEVOPS-III-A	
SAT									

## 6.SESSION PLAN / LESSON PLAN

S.No	Topic (JNTU syllabus)	Sub-Topic	No.of Lectures Required	Suggested Books	Method of Teaching	Remarks
1	UNIT-I	TRADITIONAL SOFTWARE DEVELOPMENT The Advent of Software Engineering	L1	T1	M1	
2		Waterfall method	L2	T1	M1	
3		Developers vs IT Operations conflict	L3	T1	M1	

4	<b>UNIT-II</b>	<b>RISE OF AGILE METHODOLOGIES</b> Agile movement in 2000, Agile Vs Waterfall Method	L4	T1	M4	
		Iterative Agile Software Development	L5	T1	M1	
6		Individual and team interactions over processes and tools	L6	T1	M1	
7		Working software over comprehensive documentation	L7	T1	M1	
8		Customer collaboration over contract negotiation, Responding to change over following a plan	L8	T1	M1	
9	<b>UNIT-III</b>	<b>DEFINITION OF DEVOPS</b> Introduction to DevOps	L9	T1,R2	M4	
10		DevOps and Agile	L10	T1,R2	M1	
11	<b>UNIT-IV</b>	<b>PURPOSE OF DEVOPS</b> Minimum Viable Product, Application Deployment	L11	T1	M1	
12		Continuous Integration, Continuous Delivery	L12	R1	M4	
13	<b>UNIT-V</b>	<b>CAMS (CULTURE, AUTOMATION, MEASUREMENT AND SHARING)</b> CAMS – Culture, CAMS – Automation, CAMS – Measurement, CAMS – Sharing	L13	T1	M1	
14		Test-Driven Development, Configuration Management,	L14	T1	M1	
15		Infrastructure Automation, Root Cause Analysis,	L15	T1	M1	

<b>16</b>		Blamelessness, Organizational Learning.	L16	T1	M1	
<b>TOTAL</b>			<b>16</b>			

### ***METHODS OF TEACHING***

<b>M1 : Lecture Method</b>	<b>M6 : Tutorial</b>
<b>M2 : Demo Method</b>	<b>M7 : Assignment</b>
<b>M3 : Guest Lecture</b>	<b>M8 : Industry Visit</b>
<b>M4 : Presentation /PPT</b>	<b>M9 : Project Based</b>
<b>M5 : Lab/Practical</b>	<b>M10 : Charts / OHP</b>

### **7.Session execution log**

<b>S.NO</b>	<b>UNIT No.</b>	<b>Starting Date</b>	<b>Ending Date</b>
1	UNIT-I	21/8/2023	15/9/2023
2	UNIT-II	16/9/2023	15/10/2023
3	UNIT-III	16/10/2023	15/11/2023
4	UNIT-IV	16/11/2023	30/11/2023
5	UNIT-V	01/12/2023	23/12/2023

**8.Lecture Notes**

(Attached)

**9.Assignment Questions**

N/A

**10.Mid exam question papers (samples)**

N/A

**11.Scheme of evaluation**

N/A

**12.Mapping of Cos with Pos and PSOs**

Course Outcomes	Relationship of Course outcomes to Program Outcomes (PO AVG)													
	CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1
CO1	1	-	-	-	-	-	-	-	-	-	-	2		1
CO2	2		2	-	-	-	-	-	-	-	-	2		2
CO3	1	-	-	-	1	-	-	-	-	-	-	1		1
CO4	2	-	1	-	2	-	-	-	-	-	-	2		1
CO5	1	-	-	-	-	-	-	-	-	-	-	1		1
AVERAGE	1		2		2							2		1

**13.COs,POs,PSOs Justification**

N/A

**14. Attainment of COs,POs and PSOs (Excel Sheet)**

N/A

## 15.Previous year Question Papers



### III - B.TECH- I-SEM END EXAMINATION

**Date: 28-11-2022**

**Time: 3 HOURS**

**Subject: DEVOPS**

**Branch: CSE**

**Marks: 100M**

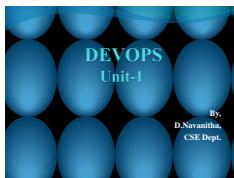
---

*Answer any five questions. Each question carries 20 marks.*

**5x20=100 M**

1. Explain about Waterfall model in detail.
2. Distinguish between Waterfall model and Agile model.
3. Discuss about iterative agile software development.
4. a) Define Devops. Discuss about Devops tools in detail.  
b) Compare Devops and Agile.
5. Explain about Continuous Integration and Continuous Delivery.
6. Explain about CAMS in detail.
7. Discuss about Test Driven Development.
8. What is Configuration Management? Explain in detail.

## 16..Power Point Presentations (PPTs)



### COURSE OBJECTIVES

To enable learners realize various aspects of DevOps Ecosystem.

- To enable students appreciate the agile led development environment.
- To give the students a perspective to grasp the need for maintainable product led development using Sprints.
- To enable students acquire fundamental knowledge of CI/CD and CAMS.

### COURSE OUTCOMES

CO1: Explain traditional software development methodologies like waterfall.

CO2: Apply the Agile Methodology and comparing various other software development models with agile.

CO3: Explain implementing Continuous Integration and Continuous Delivery.

CO4: Explain CAMS for DevOps (Culture, Automation, Measurement and Sharing).

CO5: Create quick MVP prototypes for modules and functionalities.

### UNIT - I

- TRADITIONAL SOFTWARE DEVELOPMENT**  
The Advent of Software Engineering, Waterfall method, Developers vs IT operations conflict!

### INTRODUCTION TO SOFTWARE

- Definition
- Software: Software is:
- Instructions (computer programs) that provide desired features, function and performance, when executed.
- Data structures that enable the programs to adequately manipulate information.
- Documents that describe the operation and use of the programs.

### SOFTWARE ENGINEERING

- The systematic, disciplined, quantifiable approach to the development, operation and maintenance of software; that is, the application of engineering to software.

### Software Development Life Cycle (SDLC)

- Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high-quality softwares. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.
- SDLC is the acronym of Software Development Life Cycle.
- It is also called as Software Development Process.
- SDLC is a framework defining tasks performed at each step in the software development process.

### Software Development Life Cycle

### Stage 1: Planning and Requirement Analysis

- Requirement analysis is the most important and fundamental stage in SDLC.
- It is performed by the senior members of the team with input from the project manager, domain experts, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

### Stage 2: Defining Requirements

- Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

### Stage 3: Designing the Product Architecture

- SRS is the reference for product architects to come out with the best architecture for the product to be developed.
- Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

- This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

### Stage 4: Building or Developing the Product

- In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner code generation can be accomplished without much hiccups.
- Developers follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code.

### Stage 5: Testing the Product

- This stage is usually a subset of all the stages as in the model of SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

### Stage 6: Deployment in the Market and Maintenance

- Once the product is tested and ready to be deployed it is released formally in the appropriate market. Since it is product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and then in the real business environment (UAT: User acceptance testing).
- Then based on the feedback, the product may be released as it is in a limited segment and then in the remaining market segment. After the product is released in the market, its maintenance is done for the existing customer base.

### PROCESS MODELS

- Prescriptive process models** define a set of activities, actions, tasks, milestones, and work products that are required to engineer high-quality software.

### THE WATERFALL MODEL

### Developers vs IT Operations conflict

- Developers are in charge of building an application, where there are more 'Dev' and 'Dev' conflicts among themselves, more about in a single module.
- Through maintaining Development and Operations being part technical and business based in an organization, yet the job position and operation are absolutely different in nature.
- Developers are more on producing new systems and applications and concern for the user get to know about the system and its performance. On the other hand, IT operations and support aspect as their focus is to ensure the user get to use a fast and bug free stable system.
- Both Development and Operations strive towards the goal of making a customer happy and satisfied with their different systems and services.

Thank you

## 17. Innovative Teaching Methodologies

N/A

**18Reference[Text Books/Websites/Journals]**

<https://www.coursera.org/learn/intro-to-devops>

<https://www.coursera.org/specializations/devops-cloud-and-agile-foundations>

<https://www.udemy.com/course/decodingdevops/>

<https://www.udemy.com/course/learn-devops-ci-cd-with-jenkins-using-pipelines-and-docker/>

<https://www.edureka.co/devops-certification-training>