**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**



**WORK INTEGRATED LEARNING PROGRAMMES**

**COURSE HANDOUT**

**Part A: Content Design**

|  |  |
| --- | --- |
| **Course Title** | Introduction to DevOps |
| **Course No(s)** | CSI ZG514/SE ZG514 |
| **Credit Units** | 4 |
| **Course Author** | YOGESH BHATIA, SONIKA RATHI |
| **Version No** | v1.0 |
| **Date** | June 2018 |

**Course Description**

|  |
| --- |
| This course introduces the need for Devops, the evolution of Devops. It focuses on how Devops is influencing the software development lifecycle from the perspective of process, people and technology. It also focuses on version control, configuration management and automating them. This course also helps us to gain understanding between agile and Devops, how the cloud and DevOps work together to help businesses achieve their transformation. |

**Course Objectives**

|  |  |
| --- | --- |
| **No** | Objective |
| **CO1** | To learn the key ideas and techniques to bring development and operations together to produce higher-quality software and deliver it more quickly. |
| **CO2** | To learn the core principles, business and technical terms used in DevOps from perspective of business and IT teams |
| **CO3** | To gain knowledge of the Principles and practices of the DevOps Lifecycle including Continuous Integration, Continuous Inspection, Continuous delivery, Continuous deployment and Continuous monitoring. |
| **CO4** | To understand the usage of tools and technologies used for implementing DevOps. |

**Text Book(s)**

|  |  |
| --- | --- |
| No | Author(s), Title, Edition, Publishing House |
| T1 | DevOps: A Software Architect's Perspective (SEI Series in Software Engineering)  by Len Bass, Ingo Weber, Liming Zhu , Publisher: Addison Wesley (18 May 2015). |
| T2 | Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation by Jez Humble, David Farley. Publisher: Addison Wesley, 2011 |

**Reference Book(s) & other resources**

|  |  |
| --- | --- |
| No | Author(s), Title, Edition, Publishing House |
| R1 | Effective DevOps: Building A Culture of Collaboration, Affinity, and Tooling at Scale by Jennifer Davis , Ryn Daniels. Publisher: O'Reilly Media, June 2016 |
| R2 | The DevOPS Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations by Gene Kim, Patrick Debois, John Willis, Jez Humble, John Allspaw. Publisher: IT Revolution Press (October 6, 2016) |
| R3 | **Web Resources**:   1. <https://jenkins.io/> 2. <https://xebialabs.com/solutions/devops/> 3. <https://www.ibm.com/ibm/devops/us/en/casestudies/#all> 4. <https://git-scm.com/> 5. <https://hub.docker.com/> 6. https://www.atlassian.com/git/tutorials/comparing-workflows |

**Content Structure**

|  |  |  |
| --- | --- | --- |
| **No** | **Title of the Module** | **References** |
| M0 | **Module 0: Foundational Terminology and Concepts**   * + - Software development lifecycle     - The Waterfall approach     - Agile Methodology     - Operational Methodologies: ITIL     - Development, Testing, Release, and Deployment Concepts     - Provisioning, Version Control     - Test Driven Development, Feature Driven Development     - Behavior-driven development   *(This module is to set the stage uniformly for all participants and will be covered based on the set of individuals enrolling for the course)* | T2-Chapter 1  R1-chapter 4 |
| M1 | **Module 1: Why and What is DevOps?**   * + - Problems of Delivering Software     - Principles of Software Delivery     - Need for DevOps     - Evolution of DevOps     - DevOps Practices     - The Continuous DevOps LifeCycle Process (Continuous Integration, Continuous Inspection, Continuous Deployment, Continuous Delivery, Continuous Monitoring)     - DevOps Culture     - Case Study- (IBM/Facebook/NetFlix) | T1- Chapter 1  T2- Chapter 1  R1- Chapter 2,3 |
| M2 | **Module 2: DevOps Dimensions**   * + - Three dimensions of DevOps – People, Process, Technology/Tools     - DevOps- Process       * DevOps and Agile       * Agile methodology for DevOps Effectiveness       * Flow Vs Non-Flow based Agile processes       * Choosing the appropriate team structure: Feature Vs Component teams       * Enterprise Agile frameworks and their relevance to DevOps       * Behavior driven development, Feature driven Development       * Cloud as a catalyst for DevOps     - DevOps – People       * Team structure in a DevOps       * Transformation to Enterprise DevOps culture       * Building competencies, Full Stack Developers       * Self-organized teams, Intrinsic Motivation     - Technology in DevOps(Infrastructure as code, Delivery Pipeline, Release Management)     - Tools/technology as enablers for DevOps | T1- Chapter1, 2,3  T2 – Chapter 6  R1- Chapter 11  R2 – Chapter 1, 3 |
| M3 | **Module 3: Source Code Management (Using GIT as an example tool)**   * + - * Version control system and its types       * Introduction to GIT       * GIT Basics commands (Creating Repositories, clone, push, commit, review)       * Git workflows- Feature workflow, Master workflow, Centralized workflow       * Feature branching       * Managing Conflicts       * Tagging and Merging       * Best Practices- clean code | T2-Chapter 2,14  R3- 4, 6 |
| M4 | **Module 4: Continuous build and code quality**   * Manage Dependencies * Automate the process of assembling software components with build tools * Use of Build Tools- Maven, Gradle * Unit testing   + - * Enable Fast Reliable Automated Testing       * Setting up Automated Test Suite – Selenium       * Continuous code inspection - Code quality       * Code quality analysis tools- sonarqube | T1- Chapter 5  T2- Chapter 4, 6, 13  R2-Chapter 3 |
| M5 | **Module 5: Continuous Integration and Continuous Delivery**   * + - * Implementing Continuous Integration-Version control, automated build, Test       * Prerequisites for Continuous Integration       * Continuous Integration Practices       * Team responsibilities       * Using Continuous Integration Software (Jenkins as an example tool)       * Jenkins Architecture       * Integrating Source code management, build, testing tools etc., with Jenkins - plugins       * Artefacts management       * Setting up the Continuous Integration pipeline       * Continuous delivery to staging environment or the pre-production environment       * Self-healing systems | T2- Chapter 3, 15  R2- Chapter 3  R3-1 |
| M6 | **Module 6: Continuous Deployment**   * Deployment pipeline * Human-free deployments   + - Implementing and Automating the deployment process     - Deploying it to testing environments     - Releasing software into production     - Environment-based release patterns     - Rolling Back Deployments and Zero-Downtime Releases     - Blue/Green Deployment     - Rolling Upgrade     - The canary release pattern- Dark Launches | T1- Chapter 6, 12  T2- Chapter 10  R2- Chapter 3, 4 |
| M7 | **Module 7: Continuous Monitoring**   * + - Need for continuous monitoring     - Goals of monitoring     - Challenges of monitoring under continuous change     - Alert management     - Analytics     - Continuous customer feedback and optimization     - Use of ELK (Elasticsearch, Logstash, and Kibana) Stack | T1- Chapter 7  R1- Chapter 11 |
| M8 | **Module 8: Configuration Management**   * + - Infrastructure as code     - Managing Infrastructure and Environments(Production, pre-production, Test, Developer Environment)     - Environment provisioning     - Automating and Managing Server Provisioning     - Configuration management tools- Chef, Puppet     - Managing on-demand infrastructure, Auto scaling | T2- Chapter 2, 11  R1- Chapter 14 |
| M9 | **Module 9: Virtualization and Containerization**   * + - Virtualization     - Virtualization vs Containerization     - Containerization using Dockers     - Docker Images     - Micro-services and Containerization     - Current Trends- Kubernetes, DevOps on Cloud, Function-As-A-Service (AWS Lambda) | T1- chapter 13  R3- 5 |

**Learning Outcomes:**

|  |  |
| --- | --- |
| **No** | **Learning Outcomes** |
| LO1 | Explain the need for DevOps and list down the primary benefits of DevOps from perspective of business and IT teams |
| L02 | List the ways in which DevOps uses new tools/technologies to deliver quality software more rapidly. |
| LO3 | Illustrate the practices of version control and configuration management. |
| LO4 | Summarize the essentials of continuous integration (CI) and outline the principles and practices of continuous delivery (CD) |
| LO5 | Implement an automated deployment pipeline and create a DevOps toolchain |

**Part B: Contact Session Plan**

|  |  |
| --- | --- |
| **Academic Term** | First Semester 2024-2025 |
| **Course Title** | Introduction to DevOps |
| **Course No** | CSI ZG514/SE ZG514 |
| **Lead Instructor** | YOGESH BHATIA, SONIKA RATHI |

***Course Contents***

|  |  |  |
| --- | --- | --- |
| **Contact Sessions(**Each contact session = 2 contact hours | **List of Topic Title**  **(from content structure in Course Handout)** | **Text/Ref Book/external resource** |
| 1 | **Foundational Terminology and Concepts**   * Software development lifecycle * The Waterfall approach * Agile Methodology * Operational Methodologies: ITIL * Development, Testing, Release, and Deployment Concepts * Provisioning, Version Control * Test Driven Development, Feature Driven Development * Behavior-driven development | T2-Chapter 1  R1-Chapter 4 |
| 2 | **Why and What is DevOps?**   * Problems of Delivering Software * Principles of Software Delivery * Need for DevOps * Evolution of DevOps * DevOps Practices * The Continuous DevOps LifeCycle Process (Continuous Integration, Continuous Inspection, Continuous Deployment, Continuous Delivery, Continuous Monitoring) * DevOps Culture and Team Structure * Case Study- (IBM/Facebook/NetFlix) | T1- Chapter 1  T2- Chapter 1  R1- Chapter 2,3 |
| 3 | **DevOps Dimensions**   * Three dimensions of DevOps – People, Process, Technology/Tools * DevOps- Process * DevOps and Agile * Agile methodology for DevOps Effectiveness * Flow Vs Non-Flow based Agile processes * Choosing the appropriate team structure: Feature Vs Component teams * Enterprise Agile frameworks and their relevance to DevOps * Behavior driven development, Feature driven Development * Cloud as a catalyst for DevOps | T1- Chapter1, 2,3  T2 – Chapter 6  R1- Chapter 11  R2 – Chapter 1, 3 |
| 4 | * DevOps – People * Team structure in a DevOps * Transformation to Enterprise DevOps culture * Building competencies, Full Stack Developers * Self-organized teams, Intrinsic Motivation * Technology in DevOps(Infrastructure as code, Delivery Pipeline, Release Management) * Tools/technology as enablers for DevOps | T1- Chapter1, 2,3  T2 – Chapter 6  R1- Chapter 11  R2 – Chapter 1, 3 |
| 5 | **Source Code Management (Using GIT as an example tool)**   * Version control system and its types * Introduction to GIT * GIT Basics commands (Creating Repositories, clone, push, commit, review) * Git workflows- Feature workflow, Master workflow, Centralized workflow * Feature branching * Managing Conflicts * Tagging and Merging * Best Practices- clean code | T2-Chapter 2,14  R3- 4, 6 |
| 6 | **Continuous build and code quality**   * Manage Dependencies * Automate the process of assembling software components with build tools * Use of Build Tools- Maven, Gradle * Unit testing * Enable Fast Reliable Automated Testing * Setting up Automated Test Suite – Selenium * Continuous code inspection - Code quality * Code quality analysis tools- sonarqube | T1- Chapter 5  T2- Chapter 4, 6, 13  R2-Chapter 3 |
| 7 | **Continuous Integration and Continuous Delivery**   * Implementing Continuous Integration-Version control, automated build, Test * Prerequisites for Continuous Integration * Continuous Integration Practices * Team responsibilities * Using Continuous Integration Software (Jenkins as an example tool) | T2- Chapter 3, 15  R2- Chapter 3 |
| 8 | * Jenkins Architecture * Integrating Source code management, build, testing tools etc., with Jenkins - plugins * Artefacts management * Setting up the Continuous Integration pipeline * Continuous delivery to staging environment or the pre-production environment * Self-healing systems | T2- Chapter 3, 15  R2- Chapter 3  R3-1 |
| 9 | **Continuous Deployment**   * Deployment pipeline * Human-free deployments   + - Implementing and Automating the deployment process     - Deploying it to testing environments | T1- Chapter 6, 12  T2- Chapter 10  R2- Chapter 3, 4 |
| 10 | * Releasing software into production * Environment-based release patterns * Rolling Back Deployments and Zero-Downtime Releases * Blue/Green Deployment * Rolling Upgrade * The canary release pattern- Dark Launches | T1- Chapter 6, 12  T2- Chapter 10  R2- Chapter 3, 4 |
| 11 | **Continuous Monitoring**   * Need for continuous monitoring * Goals of monitoring * Challenges of monitoring under continuous change * Alert management * Analytics * Continuous customer feedback and optimization * Use of ELK (Elasticsearch, Logstash, and Kibana) Stack | T1- Chapter 7  R1- Chapter 11 |
| 12 | **Configuration Management**   * Infrastructure as code * Managing Infrastructure and Environments(Production, pre-production, Test, Developer Environment) * Environment provisioning | T2- Chapter 2, 11  R1- Chapter 14 |
| 13 | * Automating and Managing Server Provisioning * Configuration management tools- Chef, Puppet * Managing on-demand infrastructure, Auto scaling | T2- Chapter 2, 11  R1- Chapter 14 |
| 14 | **Virtualization and Containerization**   * Virtualization * Virtualization vs Containerization * Containerization using Dockers * Docker Images | T1- chapter 13  R3- 5 |
| 15 | * Micro-services and Containerization * Current Trends- Kubernetes, DevOps on Cloud, Function-As-A-Service (AWS Lambda) | T1- chapter 13 |
| 16 | Recap Session |  |

*# The above contact session and topics can be adapted for non-specific and specific WILP programs depending on the requirements and class interests.*

## *Lab Details*

|  |  |
| --- | --- |
| **Title** | **Access URL** |
| Lab Setup Instructions | To be developed |
| Lab Capsules | To be developed |
| Additional References |  |

***Select Topics and Case Studies from business for experiential learning***

|  |  |  |
| --- | --- | --- |
| **Topic No.** | **Select Topics in Syllabus for experiential learning** | **Access URL** |
|  | Version Control using GIT  Creating repositories in GIT,  Exercises to demonstrate the use of GIT operations and commands(Push, pull, clone etc.,)  Creating branches and merging branches using GIT | R3 |
|  | Installation of Jenkins and Configuration of Jenkins to work with different version control, build and testing tools | R3 |
|  | Create jobs and projects in Jenkins | R3 |
|  | Demonstration of continuous integration with Jenkins through source code polling and build triggers | R3 |
|  | Demonstrate continuous inspection with Jenkins using sonarqube to ensure code quality | R3 |
|  | Demonstration of continuous deployment/delivery to staging/production environment with Jenkins. | R3 |

***Evaluation Scheme***

Legend: EC = Evaluation Component

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Name | Type | Duration | Weight | Day, Date, Session, Time |
| EC-1 | Quiz-I | Online | - | 5% | September 1-10, 2024 |
| Quiz-II | Online |  | 5% | October 10-20, 2024 |
| Assignment-I | Online |  | 20% | November 1-10, 2024 |
| EC-2 | Mid-Semester Test | Closed Book | 2 hours | 30% | Sunday, 22/09/2024 (EN) |
| EC-3 | Comprehensive Exam | Open Book | 2 ½ hours | 40% | Sunday, 01/12/2024 (EN) |

***Important Information***

Syllabus for Mid-Semester Test (Closed Book): Topics in Weeks 1-8

Syllabus for Comprehensive Exam (Open Book): All topics given in plan of study

Evaluation Guidelines:

1. EC-1 consists of either two Assignments or three Quizzes. Announcements regarding the same will be made in a timely manner.
2. For Closed Book tests: No books or reference material of any kind will be permitted. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
3. For Open Book exams: Use of prescribed and reference text books, in original (not photocopies) is permitted. Class notes/slides as reference material in filed or bound form is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
4. If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam. The genuineness of the reason for absence in the Regular Exam shall be assessed prior to giving permission to appear for the Make-up Exam. Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course handout, attend the lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the handout.