Course code	Course Name	L-T-P - Credits	Year of Introduction
RLMCA202	**Application Development and Maintenance**	**3-1-0-4**	**2016**

Course Objectives

- To impart the practical aspects of Application Development and Maintenance
- To emphasizes the pragmatic and practical aspects of building industry ready applications
- To understand and adhere to best practices while developing applications
- To understand the basics of continuous development and focus on industry practices around continuous integration and continuous development

Syllabus

Principles of Software Delivery, Configuration Management, Continuous Integration, Implementing a Testing Strategy, Build and Deployment Scripting, The Commit Stage, Automated Acceptance Testing, Testing Nonfunctional Requirements, Deploying and Releasing Applications, Application Development Guidelines.

Expected Outcome

The students will be

i. Able to work in a continuous integration environment
ii. Understand to follow coding best practices, and to follow the same in academic projects

References

1. Andrew Hunt, David Thomas, "The Pragmatic Programmer: From Journeyman to Master", Addison-Wesley Professional, 1999
2. Jez Humble, David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010
3. Travis Swicegood, "Pragmatic Guide to Git", Pragmatic Bookshelf, 2010

Suggested MOOC

1. https://www.udemy.com/short-and-sweet-get-started-with-git-and-github-right-now/
2. https://www.coursera.org/learn/software-processes-and-agile-practices
3. https://www.coursera.org/specializations/agile-development

Course Plan

Module	Contents	Hours	Sem. Exam Marks
I	Principles of Software Delivery – Configuration Management – Introduction to Continuous Integration - Implementing a Testing Strategy **Reference:** Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Part I, Chapters 1, 2, 3,4)	9	15%
II	Using Git for version Control – Leveraging Github.com repositories for projects/Assignments – Getting Started with Git – Working with Git- Organizing Your Repository with Branches and Tags – Working in a team – Branches and Merging – Git History - Fixing Commits **Reference:** Pragmatic Guide to Git: (Part I, 2, 3,4,5,6,7)	11	20%
FIRST INTERNAL EXAMINATION			
III	Introduction to the Deployment Pipeline – Different Stages of Deployment Pipeline – Scripting for Deployment stages –	9	15%

	Details of Commit Stage **Reference:** Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Part II, Chapters 5, 6,7)		
IV	Automated Testing – Testing for Non Functional Requirements – Deploying and releasing applications **Reference:** Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Part II, Chapters 8,9,10)	9	20%
V	Best practices for Software Development –Practical Approach in Software development- The Basic Tools **Reference:** i) The Pragmatic Programmer: From Journeyman to Master (Chapter I, 2, 3, 4)	9	15%
SECOND INTERNAL EXAMINATION			
VI	Best practices and principles in Application Development – Dealing with requirements – Pragmatic Projects **Reference:** The Pragmatic Programmer: From Journeyman to Master (Chapter 5, 6, 7, 8)	9	15%
END SEMESTER EXAM			

QUESTION PAPER PATTERN

There will be two parts in the Question paper - **Part A and Part B.**

Part A will have 8 short answer questions of 3 marks each (8 X 3 M = 24 M). There will be no choice questions.

Part B will have 6 essay questions one from each module of 6 marks each, with an alternative choice question from the same module (6 x 6M=36M). The maximum number of sub part questions in **Part B** to be limited to 2.

The total marks assigned to questions in Part A (Short answer) and Part B (Essay) together from a single module will not exceed the marks assigned to that module specified in the course plan.