

Program	Master of Computer Applications (MCA)	Semester - 2
Type of Course	-	
Prerequisite		
Course Objective	To understand fundamental concepts of software engineering and apply project management techniques based on software engineering principles.	

Teaching Scheme (Contact Hours)				Examination Scheme				
Lecture	Tutorial	Lab	Credit	Theory Marks		Practical Marks		Total Marks
				SEE (T)	CIA (T)	SEE (P)	CIA (P)	
3	0	2	4	40	30	20	10	100

SEE - Semester End Examination, CIA - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content		T - Teaching Hours W - Weightage	
Sr.	Topics	T	W
1	Software Development Process Explain Software and Software Engineering: Defining Software, Software Application Domains, Characteristics of Software, Introduction of Software Engineering, Software Engineering layers, Software Processes/Software Development Life Cycle, Software Engineering practices, Software Myths, Generic Framework activities, Umbrella activities. Project Process Models: The Waterfall Model, Incremental Process Model, Evolutionary Process Model, Prototype Model, Spiral Model, RAD Model, Component Based Process Model.	9	20
2	Agile Developments Agile Models: Agile Software Development: Defining Agility, Agile Manifesto and Principles, Agile Process, Extreme Programming, Adaptive Software Development, Scrum, Dynamic System Development Method Agile Scrum Framework: Introduction to Scrum, User Story Definition, Characteristics and Content of User Stories, Acceptance Tests and Verifying Stories, Agile Estimation, Scrum Roles and Responsibilities, The scrum Phases, Sprint Phases, Burn Down Chart, Tools for Agile Project Management.	9	20
3	Software Analysis and Design Software Analysis: Analysis and Identify Software Requirement: Requirement Engineering, Software Requirements Specification, Eliciting Requirements, Developing Usecase, Requirement Modeling(Scenario based, Class based, Behavioral, Flow-orientated & Data-orientated) Software Design: Design Concepts, Architectural Design, Component level Design, User Interface Design, Web Application Design	11	20
4	Software Coding, Testing & Quality Management Software Coding & Testing: Code Review, Software Documentation, Testing Strategies, Testing Techniques and Test Case, Test Suites Design, Testing Conventional Application, Testing Object Oriented Application, Testing Web and Mobile Application Quality Assurance and Management: Quality Concepts and Software Quality Assurance, Software Review (Formal Technical Review), Software Reliability, The Quality Standard.	9	20
5	Project Management, Configuration Management & Advances Topics Project Management: Software Metrics (Process, Product, Project), Software Project Estimation, Software Project Planning, Software Project Scheduling & Tracking, Risk Analysis & Management, Risk Strategies (Reactive vs. Proactive), Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring and Management. Configuration Management: Type of Software maintenance, Re-Engineering, SCM process, Version Control Advanced Topics: DevOps, Component-Based Software Engineering, Computer-Aided Software Engineering	10	20
Total		48	100

Suggested Distribution Of Theory Marks Using Bloom's Taxonomy

Level	Remembrance	Understanding	Application	Analyze	Evaluate	Create
Weightage	30	40	15	15	0	0

NOTE : This specification table shall be treated as a general guideline for the students and the teachers. The actual distribution of marks in the question paper may vary slightly from above table.

Course Outcomes

At the end of this course, students will be able to:

C01	describe software engineering process and model.
C02	describe agile development and scrum framework.
C03	analyse system requirement and prepare software design.
C04	apply various testing techniques to achieve software quality.
C05	discuss various software management strategies.

Reference Books

1.	Software Engineering- A practitioner's Approach By Roger S. Pressman McGraw-Hill International Editions 7th
2.	Software Engineering By Ian Sommerville Pearson Education Asia 10th
3.	Fundamentals of Software Engineering By Rajib Mall Prentice Hall India Learning Private Limited 4th
4.	Pankaj Jalote's Software Engineering: A Precise Approach By Pankaj Jalote Wiley
5.	Software Project Management By Sanjay Mohapatra Cengage Learning 1st

List of Practical

1.	Prepare description, functional and non-functional requirement for allocated project. 1 Allocation of project title. 2 Write a project abstract (Short description of project minimum 150 word). 3 Write a project detailed description (minimum 600 word) 4 List out project requirement. 4.1 Functional requirementNon-Functional requirement 4.2 Non-Functional requirement 5 Introduction of Version control using GITLAB
2.	To prepare a usecase diagram for the given project. 1. Prepare Use-case diagram for given project
3.	To prepare an activity diagram and swimlane diagram for the given project. 1. Prepare Activity diagram for given project 2. Prepare Swimlane diagram for given project
4.	To prepare a sequence diagram for the given project. 1. Prepare Sequence diagram for given project

5.	<p>To prepare a state diagram for the given project.</p> <p>1. Prepare State diagram for given project</p>
6.	<p>To prepare a class diagram for given project.</p> <p>1. Prepare Class diagram for given project</p>
7.	<p>To prepare data flow diagram of your project.</p> <p>Prepare:</p> <p>1. DFD level 0 (context) diagram for given project</p> <p>2. DFD level 1 diagram for given project</p> <p>3. DFD level 2 diagram for given project</p>
8.	<p>Prepare screens of the given project & provide detailed description of elements of each screen.</p> <p>1. Prepare screen list.</p> <p>2. Design each screen. (for mobile App use Marvel App and for website use Responsive HTML Template)</p> <p>3. Describe each screen along with the fields used in screen.</p>
9.	<p>To prepare database schema and ER diagram of the given project.</p> <p>1. Prepare database schema.</p> <p>2. Prepare ER Diagram.</p> <p>3. Manage following information along with schema: Design Date, Verified By, Verification Date, List of Changes, Approved By, Approved Date</p>
10.	<p>Draw a control flow diagram and apply cyclomatic complexity and write a test case.</p> <p>Draw a control flow diagram and apply cyclomatic complexity for the given codes. Be sure about following points.</p> <ul style="list-style-type: none"> • Guarantees that all independent execution path is exercised at least once. • Guarantees that both the true and false side of all logical decisions are exercised. • Executes the loop at the boundary values and within the boundaries. • Identify numbers of independence path require for testing. <p>Exercise 1</p> <pre>void main(){ int i,j,k; readln (i,j,k); if((i < j) (i > k)){ writeln("then part"); if (j < k) writeln ("j less then k"); else writeln ("j not less than k"); } else writeln("else Part"); }</pre> <p>Exercise 2</p> <pre>i = 0; n=4; //N-Number of nodes present in the graph while (i j = i + 1; while (j if A[i] swap(A[i], A[j]); end do; i=i+1; end do;</pre> <p>Exercise 3</p> <pre>public Hashtable countAlphabet(String aString) { Hashtable table = new Hashtable(); If (aString.length > 4000) return table;</pre>

```
StringBuffer buffer = new StringBuffer(aString);
While (buffer.length() > 0){
    String firstChar = buffer.substring(0, 1);
    Integer count = (Integer)table.get(firstChar);
    if (count == null){
        count = new Integer(1);
    }
    else{
        count = new Integer(count.intValue() + 1);
    }
    table.put(firstChar, count);
    buffer.delete(0, 1);
}
Return table;
}
```

Exercise 4

```
public class CyclomaticComplexityDemo {
public static void main(String[] args) {
// TODO Auto-generated method stub
int var1 = 10, var2 = 9, var3 = 8, var4 = 7;
if (var1 == 10){
    if(var2 > var3){
        var2 = var3;
    }
    else{
        if (var3 > var4){
            var3 = var4;
        }
        else{
            var4 = var1;
        }
    }
}
else{
    var1=var4;
}

System.out.println("Printing value for var 1,
var2, var3, and var4 "+
var1+" "+var2+" "+var3+" "+var4);
}
```

Write a test case

1. Write a test case for the screen.
2. Demonstration of web testing tool Selenium (Automation Testing)
3. Demonstration of Mobile App Testing (Automation Testing)

11. Apply empirical estimation model FP (function point value) for the project estimation.

Exercise 1:

A system has 10 external inputs, 20 external outputs, 25 different external queries, manages 4 internal logical files, and interface with 4 different legacy system. All of these data are average complexity, and overall system is relatively simple. Compute FP for the system.

Exercise 2:

Compute function point value for a project with the following domain characteristics:

- Number of user input: 5

- Number of user output: 5
- Number of user enquires: 6
- Number of files: 5
- Number of external interfaces: 5

Assume that all the complexity adjustment values are simple. Where $\Sigma (Fi) = 25$

Exercise 3:

Given the following values, compute function point when all complexity adjustment factor (CAF) and weighting factors are average.

- User Input = 50 | User Output = 40 | User Inquiries = 35 | User Files = 6 | External Interface = 4

Exercise 4:

Compute the function point for the following data

- Number of user inputs = 24
- Number of user outputs = 46
- Number of inquiries = 8
- Number of files = 4
- Number of external interfaces = 2

Weighting factor are simple and Where $\Sigma (Fi) = 35$

Exercise 5:

Calculate the function point for software application with multiple Processing Factors 5, 1, 0, 4, 3, 5, 4, 3, 4, 5, 2, 3, 4, 2 by using following given Data:

- The number of EI(Avg): 22
- The number of EO(Low): 45
- The number of EI(High): 06
- The number of ILF(Avg): 05
- The number of ELF(Low): 02

12. Using COCOMO model, estimate the effort and development time based on given project details.

Exercise 1:

Considering your immense expertise in software development, the absolute beginners Inc. has recently allotted you a mega project.

- The goal of the project is to create a database of all Hindi films released since 2000.
- The software would allow one to generate a list of top ten hit films, top ten flop films, best comedy films, and so on.
- Using your prior experience you have decided the approximate sizes of each module of the software as follows:
 - Data entry (0.9 KDSI)
 - Data update (0.7 KDSI)
 - Query (0.9 KDSI)
 - Report generation and display (2 KDSI)

Solve the problem by applying the basic COCOMO model

- Find project type.
- Find project size.
- Find initial effort estimation.
- Find time for development.
- Find the minimum size of the team you would require to develop this system.

Assuming that your client would pay Rs. 50,000 per month of development, how much would be the likely billing?

Exercise 2:

Consider a project to develop a full screen editor.

- The major components identified are
 - Screen edit,
 - Command Language Interpreter,
 - File input and output,
 - Cursor movement and
 - Screen movement.
- The sizes for these are estimated to be 4K, 2K, 1K, 2K and 3K delivered source code lines.
- Use COCOMO model to determine:
 - Schedule estimates for different phases.

Exercise 3:

Consider a project having 30,000 lines of code

- Consider a semi-detached mode software
- Consider an embedded software.

Find out

Estimation for this project.

Exercise 4:

Consider a software project using semi-detached mode with 300 KLOC.

Find out

- Effort estimation
- Development time, and
- Person estimation

Exercise 5:

- Assume that the size of a semi-detached type software product has been estimated to be 52 KLOC. Assume that the average salary of software developers is Rs. 10,000 per month. Determine the effort required to develop the software product, the nominal development time, and the cost to develop the product.

Exercise 6:

- Consider that an On-the-shelf software product for business application costs Rs. 50,000 and that its size is 40 KLOC. Assuming that in-house cost Rs. 6000 per programmer month, would it be more cost effective to buy the product or build it?

13. Identify possible APIs of given project and prepare API Document using POSTMAN tool.

Prepare API Document (Use POSTMAN tool to save API details). Maintain following information for each API.

- Name of API
- Description of API
- Input parameter
- Output parameter
- Use Swagger API for creating dummy API
- Refer <https://www.getpostman.com> for POSTMAN tools details