FR. Conceicao Rodrigues College of Engineering

Department of Computer Engineering

T.E. (Computer) (semester V)

(2022-2023)

Subject: Software Engineering

Lab Code	Lab Name	Credit
CSL501	Software Engineering Lab	1

Prerequisite: Object Oriented Programming with Java , Python Programming

Lab Objectives:	
1	To solve real life problems by applying software engineering principles
2	To impart state-of-the-art knowledge on Software Engineering
Lab Outcomes: On successful completion of laboratory experiments, learners will be able to :	
1	Identify requirements and apply software process model to selected case study.
2	Develop architectural models for the selected case study.
3	Use computer-aided software engineering (CASE) tools.

Suggested List of Experiments - Assign the case study/project as detail statement of problem to a group of two/three students. Laboratory work will be based on course syllabus with minimum 10 experiments. Open-source computer-aided software engineering (CASE) tools can be used for performing the experiment.

Sr. No.	Title of Experiment
1	Application of at least two traditional process models.
2	Application of the Agile process models.
3	Preparation of software requirement specification (SRS) document in IEEE format.
4	Structured data flow analysis.
5	Use of metrics to estimate the cost.
6	Scheduling & tracking of the project.
7	Write test cases for black box testing.
8	Write test cases for white box testing.
9	Preparation of Risk Mitigation, Monitoring and Management Plan (RMMM).
10	Version controlling of the project.

Term Work:	
1	Term work should consist of 10 experiments.
2	Journal must include at least 2 assignments on content of theory and practical of “Software Engineering”
3	The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.
4	Total 25 Marks (Experiments: 15-marks, Attendance Theory & Practical: 05-marks, Assignments: 05-marks)

Oral & Practical exam	
	Based on the entire syllabus of CSC502 and CSL501 syllabus

Course Outcomes:

Upon completion of this course students will be able to:

CSL 501.1	Identify requirements and apply software process model to selected case study.
CSL 501.2	Develop architectural models for the selected case study.
CSL 501.3	Apply project estimation techniques for the selected case study
CSL 501.4	Apply software testing techniques to a given problem.
CSL 501.5	Use computer-aided software engineering (CASE) tools.

Relationship of course outcomes with program outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO 12	PSO1	PS0 O2
CSL501.1	1	1	3	3					3	1				
CSL501.2	1	1	3						3	1				
CSL501.3	1			3					3	1				
CSL501.4				2	2									
CSL501.5	1				3									

Justification of PO to CO mapping

Course Outcome	Competency	Performance Indicator

CSL501.1	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	2.2 Demonstrate an ability to formulate a solution plan and methodology for an engineering problem	2.2.4 Compare and contrast alternative solution/methods to select the best methods
	3.1 Demonstrate an ability to define a complex/ open-ended problem in engineering terms	3.1.1 Able to define a precise problem statement with objectives and scope. 3.1.2 Able to identify and document system requirements from stake- holders. 3.1.3 Able to review state-of-the-art literature to synthesize system requirements. 3.1.4 Able to choose appropriate quality attributes as defined by ISO/IEC/IEEE standard. 3.1.5 Explore and synthesize system requirements from larger social and professional concerns. 3.1.6 Able to develop software requirement specifications (SRS).
	4.3 Demonstrate an ability to analyze data and reach a valid conclusion	4.3.1 Use appropriate procedures, tools and techniques to and analyze collect data 4.3.2 Critically analyze data for trends and correlations, stating possible errors and limitations 4.3.4 Synthesize information and knowledge about the problem from the raw data to reach appropriate conclusions.
	10.2 Demonstrate competence in listening, speaking, and presentation	10.2.1 Listen to and comprehend information, instructions, and viewpoints of others
	9.2 Demonstrate effective individual and team operations– communication, problem-solving, conflict resolution and leadership skills 9.3 Demonstrate success in a team-based project	9.2.1 Demonstrate effective communication, problem-solving, conflict resolution and leadership skills 9.2.2 Treat other team members respectfully 9.2.3 Listen to other members 9.2.4 Maintain composure in difficult situations 9.3.1 Present results as a team, with smooth integration of contributions from all individual efforts

CSL501.2	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	3.3 Demonstrate an ability to select optimal design scheme for further development	3.3.2 Consult with domain experts and stakeholders to select candidate engineering design solution for further development
	3.4 Demonstrate an ability to advance an engineering design to defined end state	3.4.1 Able to refine architecture design into a detailed design within the existing constraints. 3.4.2 Able to implement and integrate the modules. 3.4.3 Able to verify the functionalities and validate the design.
	9.2 Demonstrate effective individual and team operations– communication, problem-solving, conflict resolution and leadership skills	9.2.1Demonstrate effective communication, problem-solving, conflict resolution and leadership skills 9.2.2 Treat other team members respectfully 9.2.3 Listen to other members 9.2.4 Maintain composure in difficult situations
	9.3 Demonstrate success in a team-based project.	9.3.1 Present results as a team, with smooth integration of contributions from all individual efforts
	10.2 Demonstrate competence in listening, speaking, and presentation	10.2.1 Listen to and comprehend information, instructions, and viewpoints of others
CSL501.3	1.4 Demonstrate competence in specialized engineering knowledge to the program	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem
	4.3 Demonstrate an ability to analyze data and reach a valid conclusion	4.3.1 Use appropriate procedures, tools and techniques to and analyze collect data 4.3.2 Critically analyze data for trends and correlations, stating possible errors and limitations 4.3.3 Represent data (in tabular and/or graphical forms) so as to facilitate analysis and explanation of the data, and drawing of conclusions 4.3.4Synthesize information and knowledge about the problem from the raw data to reach appropriate conclusions

	9.2 Demonstrate effective individual and team operations– communication, problem-solving, conflict resolution and leadership skills 9.3 Demonstrate success in a team-based project. 10.2 Demonstrate competence in listening, speaking, and presentation	9.2.1 Demonstrate effective communication, problem-solving, conflict resolution and leadership skills 9.2.2 Treat other team members respectfully 9.2.3 Listen to other members 9.2.4Maintain composure in difficult situations 9.3.1 Present results as a team, with smooth integration of contributions from all individual efforts 10.2.1 Listen to and comprehend information, instructions, and viewpoints of others
CSL501.4	4.1 Demonstrate an ability to conduct investigations of technical issues consistent with their level of knowledge and understanding 5.1 Demonstrate an ability to identify/create modern engineering tools, techniques and resources 5.2 Demonstrate an ability to select and apply discipline-specific tools, techniques and resources	4.1.2 Able to choose appropriate procedure/algorithm, dataset and **test cases.** 4.1.3 Able to choose appropriate hardware/software tools to conduct the experiment. **(testing)** 5.1.1 Identify modern engineering tools, techniques and resources for engineering activities 5.2.2 Demonstrate proficiency in using discipline-specific tools
CSL501.5	1.4 Demonstrate competence in specialized engineering knowledge to the program 5.1 Demonstrate an ability to identify/create modern engineering tools, techniques and resources	1.4.1 Apply theory and principles of Computer Science and engineering to solve an engineering problem 5.1.1 Identify modern engineering tools, techniques and resources for engineering activities 5.1.2 Create/adapt/modify/extend tools and techniques to solve engineering problems

	5.2 Demonstrate an ability to select and apply discipline-specific tools, techniques and resources	5.2.1 Identify the strengths and limitations of tools for (i) acquiring information, (ii) modelling and simulating, (iii) monitoring system performance, and (iv) creating engineering designs. 5.2.2 Demonstrate proficiency in using discipline-specific tools

CO Assessment Tools:

Course Outcomes	***Indirect Method (20%)***				
	Attendance	Lab Performance	Journal Assessment	End Sem Exam	Course exit survey
CSL501.1	10%	20%	20%	50%	100%
CSL501.2	10%	20%	20%	50%	100%
CSL501.3	10%	20%	20%	50%	100%
CSL501.4	10%	20%	20%	50%	100%
CSL501.5	10%	20%	20%	50%	100%

Rubrics for assessing Course Outcome with each assessment tool:

Rubrics for Lab Experiments:

EXPERIMENT 1: SRS VERSION 1

Performance Indicator	**Excellent**	**Good**	**Below average**	**Poor**
Organization and readability	Well organized, strictly followed Standard IEEE template, Information presented in a logical sequence which flows naturally and is engaging to the audience. (4)	Organized to some extent, Standard IEEE template followed sufficiently, Information presented in a logical sequence which is followed by the reader with little or no difficulty. (3)	Poorly organized, less readable, standard IEEE template not followed at many places, Information presented in a sequence which is difficult to follow by the reader (2)	Poorly organized, standard IEEE template not followed, Information not presented in a sequence which reader can follow (1)
Level of content (4)	All points are covered in depth and breadth and answered accurately (4)	All points are covered in sufficient depth and breadth and answered correctly. (3)	Some important points are omitted/Addressed minimally (2)	Many important points are missing and the answers are not accurate. (1)

Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

EXPERIMENT 2: SRS VERSION 2 with UML diagrams

Performance Indicator	**Excellent**	**Good**	**Below average**	**Poor**
Use case diagram and use case analysis template (3)	Covers all the desired features mentioned in the requirement document, use of correct notations, identified include and extend relationship between use cases accurately(3)	Cover most of the desired features mentioned in the requirements document, use of correct notations, identified include and extend relationship between use cases at most of the places (2)	Cover some of the desired features mentioned in the requirements document, Use of notations not followed adequately , have failed to identify include and extend relationship between use cases at most of the places (1)	Use cases failed to cover most of the desired features, Use case notations not followed, have failed to identify include and extend relationship between use cases. (0)
Data flow Diagram (3)	Provided well designed diagram highlighting all major processes and the data flow among them. (3)	Provided reasonable diagram highlighting most of the major processes and the data flow among them. (2)	Provided diagram having some flaws or omission of details few major processes and the data flow among them are missing (1)	Provided Diagram which is vague and needs major improvement (0)
Sequence and activity Diagram (3)	Provided well designed diagram highlighting all major processes and the data flow among them. (3)	Provided reasonable diagram highlighting most of the major processes and the data flow among them. (2)	Provided diagram having some flaws or omission of details few major processes and the data flow among them are missing (1)	Provided Diagram which is vague and needs major improvement (0)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 3: Function Point

Performance Indicator	Excellent	Good	Below average	Poor
Identification of EI, EO, EQ, ILF, ELF(4)	Clear understanding and accurate computation of EI, EO, EQ, ILF, ELF for a given case study (3)	Sufficient understanding and correct computation of EI, EO, EQ, ILF, ELF for a given case study(2)	Failed to clearly understand and compute EI, EO, EQ, ILF, ELF for a given case study(1)	Vague understanding and wrong computation of EI, EO, EQ, ILF, ELF for a given case study(0)
Computation of Complexity adjustment factor(4)	Clear understanding and accurate weightage assignment to 14 questions necessary to calculate Σfi for a given case study. (4)	Sufficient understanding and correct weightage assignment to 14 questions necessary to calculate Σfi for a given case study. (3)	Average understanding and inappropriate weightage assignment accurate to 14 questions necessary to calculate Σfi for a given case study. (2)	poor understanding and vague weightage assignment to 14 questions necessary to calculate Σfi for a given case study. (1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 4: COCOMO Model

Performance Indicator	Excellent	Good	Below average	Poor
Identifying project type and Computation of KLOC using FP (4)	Clear understanding and accurate identification of project type and programming language needed for calculation of KLOC for a given case study (4)	sufficient understanding and correct identification of project type and programming language needed for calculation of KLOC for a given case study (3)	Average understanding and inappropriate identification of project type and programming language needed for calculation of KLOC for a given case study (2)	Poor understanding and vague identification of project type and programming language needed for calculation of KLOC for a given case study (1)
Ratings Assigned to cost diver attributes for intermediate and detailed Cocomo (4)	Clear understanding and accurate weightage assignment to cost driver attributes. (4)	Sufficient understanding and correct weightage assignment to cost driver attributes. (3)	Average understanding and inappropriate weightage assignment to cost driver attributes at some places. (2)	Poor understanding and inappropriate weightage assignment to cost driver attributes at many places. (1)

Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 5: Project Management Tool: Jira

Performance Indicator	Excellent	Good	Below average	Poor
Identification of epics and Sprints needed for a case study and their creation using JIRA tool (4)	Clear understanding and accurate identification of user stories to be included in an ongoing epic/sprint (4)	Sufficient understanding and correct identification of user stories to be included in an ongoing epic/sprint at most of the places (3)	Average understanding and inappropriate identification of user stories to be included in an ongoing epic/sprint at few places (2)	Poor understanding and inappropriate identification of user stories to be included in an ongoing epic/sprint at most of the places (1)
Dashboard design and creation of custom filters using JIRA tool (4)	Designed an attractive dashboard with many useful custom filters (4)	Designed a good dashboard with most of the useful custom filters (3)	Designed an average dashboard with few of the useful custom filters (2)	Designed a poor dashboard with very few useful custom filters (1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 6: White Box testing

Performance Indicator	Excellent	Good	Below average	Poor

Test coverage using basis path testing (4)	Clear understanding and accurate design of test cases covering all independent paths (4)	Sufficient understanding and satisfactory design of test cases covering most of the independent paths (3)	Average understanding and design of test cases missing few independent paths (2)	Poor understanding and design of test cases missing most of the independent paths (1)
Test coverage using control structure testing (4)	Designed test cases covering all conditions and data flows of program (4)	Designed test cases covering most of the conditions and data flows of program (3)	Designed test cases covering few of the conditions and data flows of program (2)	Designed test cases covering very few conditions and data flows of program (1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 7: Black Box Testing

Performance Indicator	**Excellent**	**Good**	**Below average**	**Poor**
Test coverage using equivalence partitioning (4)	Clear understanding and accurate design of test cases covering all valid and invalid partitions (4)	Sufficient understanding and satisfactory design of test cases covering most of the valid and invalid partitions (3)	Average understanding and design of test cases covering few of the valid and invalid partitions (2)	Poor understanding and design of test cases covering very few valid and invalid partitions (1)
Test coverage using boundary value analysis (4)	Clear understanding and accurate design of test cases covering all boundary cases for valid partitions (4)	Sufficient understanding and satisfactory design of test cases covering boundary cases for most of the valid partitions (3)	Average understanding and design of test cases covering boundary cases for few of the valid partitions (2)	Poor understanding and design of test cases covering boundary cases for very few of the valid partitions (1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 8: Junit Testing

Performance Indicator	Excellent	Good	Below average	Poor
Test coverage using equivalence partitioning (4)	Clear understanding and accurate design of test cases covering all valid and invalid partitions (4)	Sufficient understanding and satisfactory design of test cases covering most of the valid and invalid partitions (3)	Average understanding and design of test cases covering few of the valid and invalid partitions (2)	Poor understanding and design of test cases covering very few valid and invalid partitions (1)
Test coverage using boundary value analysis (4)	Clear understanding and accurate design of test cases covering all boundary cases for valid partitions (4)	Sufficient understanding and satisfactory design of test cases covering boundary cases for most of the valid partitions (3)	Average understanding and design of test cases covering boundary cases for few of the valid partitions (2)	Poor understanding and design of test cases covering boundary cases for very few of the valid partitions (1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 9: RISK INFORMATION SHEET

Performance Indicator	Excellent	Good	Below average	Poor
Risk identification (4)	Clear understanding and Accurate identification of risk with its probability and impact (4)	Sufficient understanding and correct identification of risk with its probability and impact (3)	Average understanding and identification of risk with its probability and impact is somewhat correct (2)	Poor understanding and identification of risk with its probability and impact is not up to the mark (1)
Risk Mitigation and Monitoring Steps (4)	Mitigation and monitoring steps are written after detailed study of the scenarios and are perfect (4)	Mitigation and monitoring steps are written after sufficient study of the scenarios and are correct(3)	Mitigation and monitoring steps are written after study of the scenarios in brief and are somewhat correct(2)	Mitigation and monitoring steps are written after study of the scenarios in brief and are not up to the mark(1)
Timeline (2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)	Submitted after 1 week (0)

Experiment 10: change and version control

Performance Indicator	Excellent	Good	Below average

Timeline(2)	Submitted on time or early (2)	Submitted next day (1)	Submitted in same week (0.5)
Organization (2)	Well organized, neat and clear handwriting, neat diagrams with all labels (2)	Organized to some extent, diagrams and handwriting is neat with some missing labels (1)	Poorly organized, diagrams incomplete(0.5)
Level of content (3)	All points are covered and answered accurately (3)	Some important points are omitted /Addressed minimally (1-2)	Many important points are missing and the answers arenot accurate. (1-0)
Knowledge about the topic (3)	All Concepts of a topic are clear and knows the application to real world problems (3)	All Concepts of a topic are mostly clear lacks understanding about the application to real World problems (2-1)	Poor understanding of concepts and application to real world problems (1-0)

CLASS	**TE Computer Engineering, Semester V**
Academic Term	**July – October 2022**
Subject	**SOFTWARE ENGINEERING(CSL501)**

Evaluation System		***Hours***	***Marks***
	Practical Examination	--	--
	Oral Examination	--	25
	Term work	--	25
	Total	--	50

Time Table	***Day***	***Batch***	***Time***
	Tuesday	***B***	***1:30-3:30pm***
	Wednesday	***C***	***1:30-3:30pm***
	Thursday	***A***	***1:30-3:30pm***
	Friday	***D***	***1:30-3:30pm***

Title of Experiments

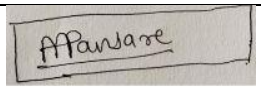
Sr.	***Title***	***Attained Cos***
1	SRS VERSION 1-To prepare software requirements specification document for a selectedcase study in IEEE format	CSL 501.1
2	SRS VERSION 2-To prepare software requirements specification document for a selected case study in IEEE format including UML diagrams	CSL 501.1, CSL 501.5
3	To perform project estimation using function point method for a selected case study	CSL 501.3
4	To perform cost estimation for a selected case study using COCOMO Model	CSL 501.3
5	Use project management tool to schedule project plan for a selected case study	CSL 501.3
6	Prepare risk table using a standard template for any risk identified for a selected case study	
7	Generate test cases to perform white box testing for a selected case study	CSL 501.4
8	To design test cases for performing black box testing (equivalence partitioning and boundary value analysis) for a selected case study	CSL 501.4
9	Perform Junit Testing for a given program	CSL 501.4, CSL 501.5
10	Use Github to learn concept of change and version control	CSL 501.5

Newly added experiments

10	Use GitHub to learn concept of change and version control

PRACTICAL SESSION PLAN			
Batch	***Dates***		***Remarks***
	Planned	***Actual***	
Experiment No. 1 SRS VERSION 1-To prepare software requirements specification document for a selectedcase study in			
B	02/08/2022	02/08/2022	
C	03/08/2022	03/08/2022	
A	04/08/2022	04/08/2022	
D	05/08/2022	05/08/2022	
Experiment No. 2 SRS VERSION 2-To prepare software requirements specification document for a selected case study in			
B	23/08/2022	23/08/2022	
C	10/08/2022	10/08/2022	
A	11/08/2022	11/08/2022	
D	12/08/2022	12/08/2022	
Experiment No 3: To perform project estimation using function point method for a selected case study			
B	30/08/2022	30/08/2022	
C	17/08/2022	17/08/2022	
A	18/08/2022	18/08/2022	
D	26/08/2022	26/08/2022	
Experiment No 4: To perform cost estimation for a selected case study using COCOMO Model			
B	31/08/2022	31/08/2022	
C	24/08/2022	24/08/2022	
A	25/08/2022	25/08/2022	
D	8/09/2022	8/09/2022	
Experiment No.5 Use project management tool to schedule project plan for a selected case study			
B	6/09/2022	6/09/2022	
C	7/09/2022	7/09/2022	
A	8/09/2022	8/09/2022	
D	9/09/2022	9/09/2022	
Experiment No. 6 Prepare risk table using a standard template for any risk identified for a selected casestudy			
B	13/09/2022	13/09/2022	
C	14/09/2022	14/09/2022	
A	15/09/2022	15/09/2022	
D	16/09/2022	16/09/2022	
Experiment No. 7 Generate test cases to perform white box testing for aselected case study			
B	20/09/2022	20/09/2022	
C	21/09/2022	21/09/2022	
A	22/09/2022	22/09/2022	
D	23/09/2022	23/09/2022	
Experiment No. 8 To design test cases for performing black box testing (equivalence partitioning and boundary value analysis) for a selected case study			
B	27/09/2023	27/09/2023	
C	28/09/2022	28/09/2022	

A	29/09/2022	29/09/2022	
D	30/09/2022	30/09/2023	
Experiment No. 9 Perform Junit Testing for a given program			
B	04/10/2023	04/10/2023	
C	12/10/2023	12/10/2023	
A	06/10/2023	06/10/2023	
D	07/10/2023	07/10/2023	
Experiment No. 10 Use GitHub to learn concept of change and version control			
B	11/10/2023	11/10/2023	
C	12/10/2023	12/10/2023	
A	13/10/2023	13/10/2023	
D	14/10/2023	14/10/2023	

Submitted By	**Approved By**
Prof. Ashwini Pansare	Prof. Roshni Padate
Sign: 	Sign:
Date of Submission:	**Date of Approval:**
Remarks by DQAC (if any)	