



Lesson Plan

Semester: V

Year: 2018-19

Course Title: Software Engineering	Course Code: 16CS54
Total Contact Hours: (L: T: P: S: 3:0:0:1)	Duration of SEE: 3hrs
SEE Marks: 100	CIE Marks: 100
Lesson Plan Author: Dr. H K Krishnappa, Dr. Vinay Hegde, Prof. Prafulla S B, Prof. Pavithra H	Date: 2-7-2018
Course Coordinator: Dr. H K Krishnappa	
Checked By:	Date:

Prerequisites:

After completing this course students should be able to:

Technical skills

1. Develop and write a software project proposal
2. Develop and write a Software Requirements Specification
3. Design a software system
4. Document the design of the software system
5. Document the implementation of a software system
6. Write a Test plan
7. Conduct a software inspection and review
8. Document software testing
9. Prepare the final report for the course

Soft skills

1. Work within a team and understand team dynamics
2. Be able to effectively communicate their work (Presentation skills)
3. Be able to write a project report (report writing skills)
4. Be able to reflect on their own work as well as the work of others (evaluate their experience)

i) **Relevance of the Course**

- To apply engineering principles to software development and to identify requirements for given software, and determine how the requirements will be elaborated into a working piece of software as well as software development.
- Software Engineering presents a broad perspective on software systems engineering, concentrating on widely used techniques for developing large scale software systems
- to help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain
- to foster an understanding of why these skills are important



Course Learning Objectives-CLO :

1. Understand software process, process models, activities involved in software engineering process, stages of software engineering process
2. Understand requirements engineering process and write the functional and non-function requirements using data flow diagrams, Use case diagrams as part of SRS document.
3. Express the Critical / Non Critical Software design using Structured, Object Oriented, Agile and Software Reuse Methodology based models and diagrams.
4. Learn the use of appropriate CASE tool for software development
5. Estimate the software development cost and prepare software project plan
6. Carry out software testing and formal verification and validation of software

Course Content/Syllabus:

Course Code: 16CS54

Course Title:

Software Engineering

Unit – I

1. Overview Introduction: FAQ's about software engineering, Professional and ethical responsibility. Socio-Technical systems: Emergent system properties. Systems engineering. Organizations, people and computer systems. Legacy systems. Software Evolution: Program evolution dynamics. Software maintenance. Evolution processes. Legacy system evolution. Software Processes: Models, process iteration, Process activities. The Rational Unified Process. Computer Aided Software Engineering	8hrs
---	-------------

Unit – II

2. Requirements: Software Requirements: Functional and Non-functional requirements. User requirements. System requirements. Interface specification. The software requirements document.	5hrs
Requirements Engineering Processes: Feasibility studies. Requirements elicitation and analysis. Requirements validation. Requirements management. Critical Systems: A simple safety-critical system. System dependability. Availability and reliability.	5hrs

Unit – III

3. Development: Rapid Software Development: Agile methods. Extreme programming. Rapid application development. System Models: Context models. Behavioral models. Data models. Object models. Structured methods. Software Design: Architectural Design: Architectural design decisions. System organization. Modular decomposition styles. Control styles. Object-Oriented design: Objects and Object Classes. An Object-Oriented design process. Design evolution.	8hrs
---	-------------

Unit – IV

Verification and Validation: Verification and Validation: Planning. Software inspections. Automated static analysis. Verification and formal methods.	8hrs
--	-------------



Software testing: System testing. Component testing. Test case design. Test automation.	
--	--

Unit – V

Project Management: Management activities. Project planning. Project scheduling. Risk management. Managing People: Selecting staff. Motivating people. Managing people. The People Capability Maturity Model. Software Cost Estimation: Productivity. Estimation techniques. Algorithmic cost modeling, Project duration and staffing.	10hrs
---	--------------

REFERENCE BOOKS:

1. Ian Sommerville, Software Engineering, Person Education Ltd., 7th Edition, First Indian reprint 2005.
2. Roger.S.Pressman, Software Engineering-A Practitioners approach, Tata-McGraw Hill 5th Edition.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, 2nd edition Narosa Publications.
4. Stephen R. Schacht, Object Oriented & Classical Software Engineering, Tata McGraw-Hill, 2002.



Evaluation Scheme

CIE Scheme

Assessment	Weight age in Marks
CIE Tests (Nos. Max.Marks. Duration)	150 marks(3 tests) Reduced to 50marks
Chapter End Quizzes	30 marks
Assignment	10marks (15 +5) (5 marks for learning methodology evaluation)
Lab	50
Total	150

Course Outcomes:

1. Exploring fundamentals of s/w engineering, process models, critical systems, RE, design, development and testing
2. Analyse, design and efficient implementation of s/w using rapid methods
3. Define a project management plan, tabulate testing plans, and reproduce effective procedures
4. Implement working code, schedule tasks to achieve goals, design computer – human interfaces, and manage time and physical resources

Course Articulation Matrix
Mapping of Dept. PO's with CO's

Course Outcomes – CO Software Engineering -16CS54	Program Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
Comprehend various software life cycle models and steps of software development process.	-	L	L	-	-	-	-	-	-	L	-	M
Apply concepts of Software Project Planning and software Design techniques	M	M	M	M	M	-	L	L	M	M	M	M
Analyze capabilities of various tools to assist in the software development activities.	M	L	M	L	L	-	-	-	M	M	L	M
Develop correct and robust software design and software project plan from requirement gathering to implementation.	M	L	M	M	L	L	-	L	M	M	M	M

Program Outcome Matrix

Program Outcome	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8	Outcome 9	Outcome 10	Outcome 11	Outcome 12
Software Engineering (16CS54)	M	M	M	M	M	L	L	L	M	M	M	M

Program Specific Outcome (PSO)	PSO1	PSO2
Software Engineering (16CS54)	H	M



Course Unitization for Internals and Semester End Examination

Part	Teaching Hours	No. of Questions in			No. of Questions in SEE
		Internals I	Internals II	Compensatory Internals	
Unit 1	8	3	-	-	2
Unit 2	8	2	1	-	2
Unit 3	7	-	2	-	2
Unit 4	7	-	4	2	2
Unit 5	7	-	-	3	2

Faculty In-charge

Head of Department

LESSON PLAN /WEEK

Weeks	Days	Units	Main Topics	Sub Topics	Activity & Materials to be used
1	1	Unit1	Introduction,	FAQ's about software engineering,	PPT + Black board lecture
	2			Professional and ethical responsibility	PPT + Black board lecture
	3			Software Processes	PPT + Black board lecture
2	1		Software Models, systems	Process iteration, Software specification, ,	PPT + Black board lecture
	2			Software design and implementation	PPT + Black board lecture
	3			Software validation,	PPT + Black board lecture



				Software evolution, Automated Process support	
3	1	Unit 2	Critical systems, Requirements Engineering	Critical system, Availability and reliability,	PPT + Black board lecture
	2			Safety and Security	PPT + Black board lecture
	3			Functional and Non-functional requirements, User requirements	PPT + Black board lecture
4	1		Software Requirements	System requirements, ,	TPS-1
	2			The software requirements document.	TPS-1
	3			Interface specification	TPS-1
5	1		Requirements engineering processes, System models: Critical system Specification	Feasibility studies, Requirements elicitation and analysis	TPS-1
	2			, Requirements validation, Requirements management. Context models,	TPS-1
	3			Behavioral models, Data models, Object models, CASE workbenches.	TPS-1
6	1	Unit 2,3	Critical system Specification, Software Design, Architectural design:	Software reliability specification, safety specification	PPT + Black board lecture
	2			Architectural design decisions, System organization	Flipped Classroom http://slideplayer.com/slide/6225213/



	3			Modular decomposition Control styles	Flipped Classroom http://slideplayer.com/slide/6225213/
7	1	Unit 3	Architectural design: Object-Oriented design,	reference architectures	Flipped Classroom http://slideplayer.com/slide/6225213/
	2			Objects and Object Classes	Flipped Classroom http://slideplayer.com/slide/6225213/
	3			An Object Oriented design process	Flipped Classroom http://slideplayer.com/slide/6225213/
8	1	Unit 3,4	Object-Oriented design, Rapid Software Development	Design evolution.	PPT + Black board lecture
	2			Agile methods	PPT + Black board lecture
	3			Extreme Programming	PPT + Black board lecture
9	1	Unit 4	Rapid Software Development, Software Reuse	Rapid Application Development	PPT + Black board lecture
	2			Software Prototyping	PPT + Black board lecture
	3			The reuse landscape	PPT + Black board lecture
10	1	Unit 4	Software Reuse	Design patterns	PPT + Black board lecture
	2			Generator-based reuse	PPT + Black board lecture
	3			Application System reuse	PPT + Black board lecture
11	1	Unit 5	Verification And Validation	Planning Verification and Validation	PPT + Black board lecture
	2			software inspection	PPT + Black board lecture
	3			automated static analysis	PPT + Black board lecture
12	1	Unit 5	Verification And Validation, Software Testing	verification and formal methods	PPT + Black board lecture
	2			clean room software development	PPT + Black board lecture
	3			System testing	



13	1	Unit 5	Software Testing	component testing	PPT + Black board lecture
	2			test case design	PPT + Black board lecture
	3			test automation	PPT + Black board lecture
14	1	Unit 5	Project management	Management activities	PPT + Black board lecture
	2			Project planning	PPT + Black board lecture
	3			Project Scheduling	PPT + Black board lecture
15	1	Unit 5	Project management, Software cost estimation	Risk management	PPT + Black board lecture
	2			Productivity	PPT + Black board lecture
	3			Estimation techniques	TPS-2
16	1	Unit 5	Software cost estimation	Algorithmic cost modeling,	TPS-2
	2			Project duration and staffing	TPS-2

Flipped Class Room Activity

Contains three out of class activity and two in-class activity –

Expected Outcome of out of class activity –

- Comprehend the scenario and identify design solutions to the given problem
- Represent the solution in diagrammatic fashion for the given problem
- Articulate the proposed solution on the given problem.

Activity -1

Key Concepts covered – Basics of Systems Models

Link - <http://slideplayer.com/slide/6225213/>

Duration – 14:18

Key Concepts covered - Object Model

Link - <https://www.youtube.com/watch?v=-MxGg1Rk4Zo>



Duration: 8:26

Activity -2

Key Concepts covered – Data Models

Link - <https://www.youtube.com/watch?v=MssCuo2Crcw>

Duration: 11:25

Activity -3

Key Concepts covered – Context Diagram

Link - <https://www.youtube.com/watch?v=zQvmrmvqyS4>

Duration: 3:46

In -classroom Activity

TPS-1

Topic	Requirement Collection and Validation for the given case study scenario
Target Students	5 th Sem CSE
Overall Duration	40 mins
Question-	<p>Identify the requirement for the following systems and validate the same.</p> <p>“The system is intended to support the day-to-day operations of a restaurant by improving the processes of making reservations and allocating tables to customers. The Restaurant system provides the facilities like</p> <ul style="list-style-type: none">Record BookingCancel BookingRecord ArrivalTable Transfer <p>The new system can offer diners eat at the restaurant without making an advance booking, if a free table is available. This is known as Walk-in. The new system should display the same information as the existing booking sheet and in same format, to make it easy for restaurant staff to transfer, to the new system. When new bookings are recorded or changes made to existing bookings, the display should be immediately updated, so that restaurant staff is working with the latest information available.”</p>

TPS 2



Topic	Cost Estimation Techniques
Target Students	5th Sem CSE
Overall Duration	40 mins
Question	Identify the cost metrics using the COCOMO -2 Model for Online Restaurant System.



UNIT 1

Course Code and Title: 16CS54	
Chapter Number and Title: 1/Overview, 4/Software process Model, 3/Critical systems	Planned Hours: 8

Teaching Learning Process: Active Learning will be used in Unit I, where real world problem will be given and students are required to solve the same. Students will engage in activities, such as reading, writing, discussion, or problem solving that promote evaluation of class content.

Topic Learning Objectives:

Sl. No.	Topic Learning Objectives	Blooms Taxonomy level[L1-L6]
1	The Objective of this chapter is to introduce the subject of software engineering. When we read this chapter we will understand:	1
2	What software engineering is and why it is important	1
3	Know the answers to key questions which provide an introduction to software engineering	1
4	Understand ethical and professional issues which are important for software engineers	1

Lesson Schedule

Class No. Portion covered per hour

1. Introduction: FAQ's about software engineering,
2. Professional and ethical responsibility.
3. Software Processes: Software Process Models, Process iteration,
4. Software specification, Software design and implementation,
5. Software validation, Software evolution,
6. Automated Process support.
7. Critical Systems

Model Questions [minimum 4]

1. What is software engineering? Is it an art, craft or science? Discuss.
2. Explain the major differences between software engineering and other traditional engineering disciplines.
3. Distinguish between generic and customized software products. Which one has larger share of market and why?
4. Distinguish between software product and software process.



5. Explain the salient features of IEEE code of ethics.
6. What are critical systems? Explain the significant of dependability in critical systems.
7. Write short notes on:
 - a) System reliability
 - b) System availability
 - c) Classification of safety - Critical system Security

Unit-2

Teaching Learning Process: Blended Learning will be used in unit II, in which the requirements

Course Code and Title: 16CS54 Software engineering	
Chapter Number and Title: 2. Requirement Engineering, 6/Software Requirements, 7/Requirement Engineering Process, 8/System Models, 9/Critical System Specification	Planned Hours: 5

will be defined in course materials and they are required to give the requirement specifications by solving the same.

Topic Learning Objectives

Sl. No.	Topic Learning Objectives	Blooms Taxonomy level [L1-L6]
1	Understand the concepts of user requirements and system requirements and why these may be expressed using different notations	3
2	Differences between functional and non functional requirements	2

Lesson Schedule

Class No. Portion covered per hour

1. Software Requirements: Functional and Non-functional requirements,
2. User requirements, System requirements,
3. Interface specification, the software requirements document.
4. Requirements engineering processes: Feasibility studies,
5. Requirements elicitation and analysis, Requirements validation,
6. Requirements management.
7. System models: Context models,
8. Behavioral models, Data models,
9. Object models,
10. CASE workbenches

Model Questions

1. What are the different levels of software requirements? Who are the users of these requirements?



2. Explain briefly different notations for requirement specification.
3. Explain the characteristic features of a software requirements document
4. Describe the major activities of requirement engineering process.
5. Explain why it is very difficult to produce a complete and consistent set of requirements?
6. What are critical systems? Explain the significant of dependability in critical systems.
7. What are reliability metrics? Explain software and hardware reliability metrics in detail.
8. How reliability can be predicted? Explain.

UNIT-3

Teaching Learning Process: Active Learning will be used in Unit III, where real world problem

<i>Course Code and Title: 16CS54</i>	
<i>Chapter Number and Title: 3. Software Design, 11/ Architecture Design, 14/Object Oriented Design.</i>	<i>Planned Hours: 5</i>

will be given and students are required to solve the same. Students will engage in activities, such as reading, writing, discussion, or problem solving that promote evaluation of class content.

Learning Objectives

Sl. No.	Topic Learning Objectives	Blooms Taxonomy level[L1-L6]
1	Why architectural Design of software is important?	2
2	Introduction to the three complementary architectural styles and reference architecture.	3
3	Important activities in general object-oriented design process.	2
4	Documentation on Object-oriented design.	3
5	Principal activities in the user interface design process.	4
6	Graphical and textual presentation of information	2



Lesson Schedule

Class No. Portion covered per hour

1. Architectural Design decisions
2. System Organization and Modular decomposition styles
3. Control Styles and Reference architectures
4. Objects and object class
5. Object Oriented Design Process
6. Design evolution

Model Questions

1. Explain architectural design process in detail.
2. Explain repository model. Discuss its advantages & disadvantages.
3. How Object Oriented model is useful in modular decomposition? Explain.
4. Explain all the stages of Object Oriented Design with an example.
5. Discuss advantages and disadvantages of object oriented design
6. List and discuss various design principles involved in user interface design.
7. Explain important consideration in information display design. Discuss the advantages of graphical information display.

Unit-4

Course Code and Title: 16CS54

Chapter Number and Title: 17/ Rapid S/W Development, 18/Software Reuse.

Planned Hours: 8

Teaching Learning Process: Active Learning will be used in Unit IV, where real world problem will be given and students are required to solve the same. Students will engage in activities, such as reading, writing, discussion, or problem solving that promote evaluation of class content.

Learning Objectives

Sl. No.	Topic Learning Objectives	Blooms Taxonomy level
1	Understand how an iterative, incremental s/w development approach leads to faster delivery of more s/w	1
2	Know the benefits and problems of reusing software when developing new systems.	2
3	Contribution of fault avoidance and fault tolerance in developing dependable systems.	3



Lesson Schedule

Class No. Portion covered per hour

1. Iterative development
2. fault avoidance
3. dependable processes
4. Extreme programming

Model Questions

1. Explain the various difficulties with iterative development and incremental delivery?
2. What are the principles of agile method?
3. Explain the extreme programming practices?
4. What are the benefits of S/W reuse and landscape of reuse?
5. Characteristics of dependable process?
6. What is fault Tolerance? Explain fault detection and damage assessment?

Unit-5

Course Code and Title: 16CS54	
Chapter Number and Title: 22/Verification And Validation, 23/Software Testing, 5/Project management, 26/Software cost estimation	Planned Hours: 8

Teaching Learning Process: Problem-Based learning approach will be used in the unit V, where the students will have sufficient domain knowledge and are required to solve open ended problem. The instructor will provided guidance and monitoring of the problem solving.

Topic Learning Objectives

Sl. No.	Topic Learning Objectives	Blooms Taxonomy level
1	Discuss the verification and validation techniques used in the development of critical systems and the reliability of software system.	2
2	Introduction to programming techniques for fault avoidance.	3
3	Distinguish between Software Verification and Validation.	3
4	Principles of System testing and Component testing.	2
5	Software tools that support test automation.	4
6	Know the principle task of project managers and the need for project planning in software projects	4
7	Introduction to the notion of risk management.	2



8	Discuss the fundamentals of software costing and the metrics that are used for software productivity assessment	2
---	---	---

Lesson Schedule

Class No. Portion covered per hour

1. Reliability validation
2. Safety assurance and security assessment
3. Planning verification and validation
4. Verification and formal methods
5. Fault tolerance and its architectures
6. Reuse landscape and generator based reuse
7. Application system reuse
8. System Testing and Component Testing
9. Test case design
10. Test automation
11. Management Activities, Project planning and scheduling
12. Risk Management
13. Software Productivity
14. Estimation Techniques and Algorithmic cost modeling

Model Questions

1. Explain the differences between verification and validation and discuss the importance of verification and validation planning.
2. What are the main objectives of software testing? Explain.
3. What is Black-box testing? Explain with an example.
4. Write note on Testing Workbenches.
5. Why should the parameters for cost estimation be determined from a company's data?
6. Calculate COCOMO effort, TDEV, average staffing, and productivity for an organic project that is estimated to be 39,800 lines of code.
7. List and explain various factors affecting the software pricing.



SNo	Questions	CO
1.	Is the course allowed you to understand the fundamental knowledge of Software Engineering	1
2.	Are the topics in this course are appropriate to Computer science	1
3.	Does this course help you to understand and design the software development	2
4.	Was this course useful in making yourself capable of analyse and design of the applications	2
5.	Did this course helped in designing and developing the real time application for social and environmental cause?	2,4
6.	Do you feel topics included in this course will give good background for higher education?	1
7.	Are you able to do any research work in the field of computer science by process of leadership, project management and financial skills with professional ethics?	1
8.	To what extent you grade the quality of contents in this subject?	1,3
9.	Are the topics in this course are more application based or theoretical?	1
10.	Whether you were provided with adequate orientation and guidance for proceeding with new techniques and design methodologies.	1