Course Title: Software Engineering **Credit: 3**
Course No: CSIT.324 **Number of period per week: 3+3**
Nature of the Course: Theory + Lab **Total hours: 45+45**
Year: Third, Semester: Sixth
Level: B. Sc. CSIT

1. Course Introduction

This course is aimed to understanding of the software engineering discipline and its application to the development of software.It cover the software concept, different software process models, software requirements engineering process, systems analysis and design as a problem-solving activity, design architecture, configuration management and software quality assurance to software development process.

2. Objectives

After completion of Software Engineering course, Students will be able to:

- Understands the systematic, discipline and quantifiable approach of software development process and phases.
- Demonstrate problem solving, critical thinking and analytical skills in building and maintaining quality software systems in the most cost effective manner.
- Demonstrate leadership and creativity in software industries with proficient in oral and written communication, and effective in teamwork with the highest levels of ethical standards and social responsibilities.
- Engage in lifelong learning, advance their knowledge, and have skills and ability to pursue graduate studies and do research in software engineering and related interdisciplinary areas.

3. Specific Objectives and Contents

Specific Objectives	**Contents**
• Define software, characters and categories • Explore changing nature of software • Define software engineering, practices and myths	**Unit I: Software and Software Engineering (4 Hrs.)** 1.1. Definition, characteristics and application domain of software 1.2. Changing Nature of Software 1.3. Definition of software engineering and software process 1.4. Software engineering practices 1.5. Software Development Myths 1.6. Software Process Structure
• Analyze the modern software development process • Compare the classical and evolutionary software development model • Apply the Agile process in software development	**Unit II: Software Development Process Model(5 Hrs.)** 2.1. Waterfall Model 2.2. Prototype Model 2.3. Rapid Application Development Model 2.4. Spiral Model 2.5. Agile Process: Extreme Programming, Scrum 2.6. Aspect Oriented Software Development Model

• Create the function and non-functional requirement of software. • Understands the document structure of software requirement. • Identify the requirement engineering process in real development process.	**Unit III: Requirements Engineering (5)** 3.1. Functional and non-functional requirements 3.2. The software requirements document 3.3. Requirements specification 3.4. Requirements engineering processes 3.5. Requirements elicitation and analysis 3.6. Requirements validation 3.7. Requirements management
• Identify the software modeling concept • Describe the model driven software engineering • Explain the architecture design and pattern • Understands the mobile and web development architecture	**Unit IV:System Modeling and Architecture Design (7 Hrs.)** 4.1. Context models 4.2. Interaction models 4.3. Structural models 4.4. Behavioural models 4.5. Model-drivenengineering 4.6. Architectural design decisions 4.7. Architectural views 4.8. Architectural patterns 4.9. Application architectures 4.10. Web Application Design 4.11. Mobile Application Design
• Understand object oriented design principle • Describe UML • Design and Draw Use Case, Activity, Sequence, Class, Component and Deployment Diagram. • Compare the CASE and i-CASE Tools	**Unit V: Object Oriented Design (7 Hrs.)** 5.1. Object Oriented design principle and process 5.2. Unified Model Language 2.0 5.3. Use Case Diagram 5.4. Activity Diagram 5.5. Sequence Diagram 5.6. Class Diagram 5.7. Component Diagram 5.8. Deployment Diagram 5.9. CASE and I-CASE Tools
• Understand software configuration process • Describe the version management and maintenance process • Describe the software engineering process.	**Unit VI: Configuration Management(4 Hrs.)** 6.1. Software Configuration Management Activities 6.2. Change management 6.3. Version and Release management 6.4. Software Maintenance 6.5. Software Re-Engineering
• Understand elements of SQA • Define the SQA Process and Task • Understands the software reliability and Standards	**Unit VII: Software Quality Assurance (4 Hrs.)** 7.1. Elements of software Quality Assurance 7.2. SQA Process and product characterise 7.3. SQA Task, Goal and Metrics 7.4. Statistical Software Quality Assurance

	7.5. Software Reliability 7.6. ISO 9000 Quality standards
- Understand concepts of software Testing and Approach - Define the process of unit, integration and system Testing - Compare Validation and System Testing - Understands the Mobile and Web Application Testing Approach	**Unit VIII: Software Testing Strategies (5 Hrs.)** 8.1 Strategic Approach of Software Testing 8.2 Black Box and White Box Testing Approach 8.3 Unit and Integration Testing 8.4 Validation and System Testing 8.5 Testing Object Oriented software 8.6 Testing Web Application 8.7 Testing Mobile Application 8.8 Testing Tools
- Understand concept of project and its activities - List the planning activities - Use Risk management and Cost estimation tools	**Unit IX: Software Project Management (4 Hrs.)** 9.1 Project Activities 9.2 Project Planning 9.3 Risk Management 9.4 Cost Estimation

Evaluation System

Undergraduate Programs							
External Evaluation	**Marks**	**Internal Evaluation**	**Weight age**	**Marks**	**Practical**	**Weight age**	**Mark**
End semester examination	60	Assignments	20%	20	Practical Report copy	25%	20
(Details are given in the separate table at the end)		Quizzes	10%		Viva	25%	
		Attendance	20%		Practical Exam	50%	
		Internal Exams	50%				
Total External	60	Total Internal	100%	20		100%	20
Full Marks 60+20+20 = 100							

External evaluation

1. **End semester examination:**
   It is a written examination at the end of the semester. The questions will beasked covering all the units of the course. The question model, full marks, time and others will be as per the following grid.

2. **External Practical Evaluation:**
   After completing the end semester theoretical examination, practical examination will be held. External examiner will conduct the practical examination according to the above

mentioned evaluation. There will be an internal examiner to assist the external examiner. Three hours time will be given for the practical examination. In this examination Students must demonstrate the knowledge of the subject matter.

Full Marks: 100, Pass Marks: 45, Time: 3 Hrs

Nature of question	Total questions to be asked	Total questions to be answered	Total marks	Weightage
Group A: multiple choice*	20	20	20×1 = 20	60%
Group B: Short answer type questions	7	6	6×8 = 48	60%
Group C: Long answer type questions	3	2	2×16 =32	60%
			100	100%

Each student must secure at least 50% marks in internal evaluation in order to appear in the end semester examination. Failed student will not be eligible to appear in the end semester examinations.

Internal evaluation

Assignment: Each student must submit the assignment individually. The stipulated time for submission of the assignment will be seriously taken.

Quizzes: Unannounced and announced quizzes/tests will be taken by the respective subject teachers. Such quizzes/tests will be conducted twice per semester. The students will be evaluated accordingly.

Attendance in class: Students should regularly attend and participate in class discussion. Eighty percent class attendance is mandatory for the students to enable them to appear in the end semester examination. Below 80% attendance in the class will signify NOT QUALIFIED (NQ) to attend the end semester examination.

Presentation: Students will be divided into groups and each group will be provided with a topic for presentation. It will be evaluated individually as well as group-wise. Individual students have to make presentations on the given topics.

Mid-term examination: It is a written examination and the questions will be asked covering all the topics in the session of the course.

Discussion and participation: Students will be evaluated on the basis of their active participation in the classroom discussions.

Instructional Techniques: All topics are discussed with emphasis on real-world application. List of instructional techniques is as follows:

- Lecture and Discussion
- Group work and Individual work
- Assignments

- Presentation by Students
- Quizzes
- Guest Lecture

Students are advised to attend all the classes and complete all the assignments within the specified time period. If a student does not attend the class(es), it is his/her sole responsibility to cover the topic(s) taught during that period. If a student fails to attend a formal exam/quiz/test, there won’t be any provision for re-exam. Unless and until the student clears one semester he/she will not be allowed to study in the following semesters.

Laboratory Work

Student should practice software engineering principle for real world applications. Students are recommended to use UML tools as a part of lab work. The choice of CASE Tools can range from MS-Visio, MS-Project manager, Rational Rose so as to provide practical exposure for realizing system design issues. Students should select the development model and apply requirement engineering. Students should use software quality assurance activities and testing techniques for quality product. The lab work and case study should be practiced for minimum of 3 lab hours or case study per week.

It is highly recommended that project proposal, system requirement specification document, design specification, test case are prepared to real world application should be practiced. A group of four or five students can work together.

Prescribed Texts

1. Sommerville, I. (2010). Software engineering. 9th Edition, Wokingham, England: Addison-Wesley Pub. Co.
2. Pressman, R.S (2014)., “Software Engineering – A Practitioner's Approach”, 8th Edition, New Delhi, McGraw Hills

References

1. LethbridgeTimothy and LaganiereRobert (2010). Object-oriented Software Engineering: Practical Software Development using UML and Java. New Delhi, McGraw Hills
2. Pankaj Jalote,(2005) “An Integrated Approach to Software Engineering”, 3rd Edition,New Delhi, Narosa Publishing House.
3. Pfleeger, S. L., & Atlee, J. M. (2010). Software engineering: theory and practice (4th ed). N.J. Prentice Hall.
4. Schwaber, K., &Beedle, M. (2002). Agile software development with Scrum. Upper Saddle River, NJ: Prentice Hall.