GUJARAT TECHNOLOGICAL UNIVERSITY

INFORMATION AND COMMUNICATION TECHNOLOGY

SOFTWARE ENGINEERING

SUBJECT CODE:2153202

B.E. 5th SEMESTER

Type of course: Core

Prerequisite: none

Rationale:

- acquaint with systematic and organized approach for developing the software
- Provide a scope to students where they can solve small, real life problems
- learn an individual as well as teamwork approach for project development
- understand various software quality assurance as well as testing approaches

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks			Practical Marks			
				ESE (E)	PA (M)		ESE (V)		PA (I)	
					PA	ALA	ESE	OEP		
4	0	2	6	70	20	10	20	10	20	150

Content:

Sr. No.	Content	Total Hrs	% Weighta
1	**Software process Models and lifecycle:** Software Product, Product, Software Processes, Evolving Role of Software, Software: A Crisis on the Horizon and Software Myths, Software Engineering: A Layered Technology, Study of different Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Component-Based Development, Process, Product and Process, Object Oriented Software Engineering	7	12
2	**Project Management Concepts & Project Metrics:** The Management Spectrum, People, Product, Process, Project, The W5HH Principle, Metrics in the Process and Project Domains (FP & LOC), Software Measurement, Metrics for Project and Software Quality	5	10
3	**Software Project Planning, Scheduling and Tracking:**Project Planning Objectives, Software Project Estimation using COCOMO Model, Software Scope and Resources, Empirical Estimation Models, Automated Estimation Tools, Basic Concepts and Relationship Between People and Effort, Defining a Task Set for the Software Project, Selecting Software Engineering Tasks, Defining a Task Network and Scheduling, Earned Value Analysis and Error Tracking	8	10

4	**Software Requirements Specification:** Requirement Gathering and Analysis, Software Requirement Specification(SRS), Formal requirements specification and verification - axiomatic and algebraic specifications	**4**	12
5	**Analysis Modeling, SoftwareDesign Concepts and Principles:** The Elements of the Analysis Model, Data Modeling, Functional Modeling and Information Flow, Behavioral Modeling and Structured Analysis, Software Design and Software Engineering, The Design Process, Design Principles, Design Concepts, Modular Design, Design Heuristics for Effective Modularity, The Design Model ,Design Documentation, Function oriented v/s object-oriented design, Object Modeling using UML, Software Architecture and Data Design, Architectural Styles, Analyzing Alternative Architectural Designs, Mapping Requirements into a Software Architecture	**11**	15
6	**User Interface Design, Component Level Design:**User Interface Design, Task Analysis and Modeling, Interface Design Activities and Implementation Tools, Design Evaluation, Structured Programming and Comparison of Design Notation	**3**	7
7	**Risk Analysis & Management:** Reactive versus Proactive Risk Strategies, Software Risks (Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation), Risks Monitoring and Management	**3**	7
8	**Coding, Software Testing Techniques & Software Testing Strategies:** Software Testing Fundamentals and Test Case Design, White-Box Testing and Black-Box Testing, ISO/IEC/IEEE Software Testing standards, Testing for Specialized Environments, A Strategic Approach to Software Testing and Issues, Unit Testing, Integration and Validation Testing, System Testing,Software Documentation and Debugging Techniques	**10**	15
9	**Software Quality Assurance and Configuration Management** - Quality Concepts and Software Quality Assurance, Quality Planning and Control, Software Reviews (Formal Technical Reviews), Software Reliability and Fault Tolerance, The ISO 9000 Quality Standards, The SCM Process, Identification of Objects in the Software Configuration, Six Sigma, Version Control and Change Control	**6**	7
10	**Emerging and advanced topics in Software Engineering**: Security Engineering, Agile Methods, Client Server Software Engineering, Aspect Oriented Software Development, Software Engineering Aspects of Programming Languages, Reverse Engineering, Re-engineering, Web Engineering, CASE.	**7**	5

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
20	**30**	**20**	**-**	**-**	**-**

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create and above Levels (Revised Bloom's Taxonomy)

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

Text Book:

1. Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill.
2. Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India.
3. Pankaj Jalote, An integrated approach to Software Engineering by Springer.
4. Ian Sommerville, Software Engineering, Addision and Wesley.

Course Outcome:

After completion of the course students will be able to

1. Prepare SRS (Software Requirement Specification) document and SPMP (Software Project Management Plan) document.
2. Apply the concept of Functional Oriented and Object Oriented Approach for Software Design.
3. Recognize how to ensure the quality of software product, different quality standards and software review techniques.
4. Apply various testing techniques and also upgrade it using advanced Software Engineering Methods.

List of Experiments:

Prepare case study on Following:

1) Library Information System
2) Villager Telephone System
3) Waste Management Inspection Tracking System (WMITS)
4) Flight Control System
5) Ambulance Dispatching System
6) Development of requirements specification
7) Function oriented design using SA/SD
8) Object-oriented design using UML
9) Test case design
10) Implementation using Java and testing
11) Use of appropriate CASE tools and other tools such as configuration management tools
12) Program analysis tools in the software life cycle

Design based Problems (DP)/Open Ended Problem:

1. For Natural Language Processing (NLP) applications, estimate project failure rate.
2. Design and develop an open source method of detecting the DIFFERENCESS between two engineering designs for the same problem.

Major Equipment: Computer

List of Open Source Software/learning website:

- Software:-Rational Rose, Microsoft Visio, Enterprise resource planning
- Project Management Tools
- SCM Tools
- SQA Tools
- Analysis and Design Tools
- User Interface Development Tools
- Object-Oriented Software Engineering Tools
- Testing Tools
- Formal Methods Tools
- Client/Server Tools
- Web Engineering Tools
- Reengineering Tools

ACTIVE LEARNING ASSIGNMENTS: Preparation of power-point slides, which include videos, animations, pictures, graphics for better understanding theory and practical work – The faculty will allocate chapters/ parts of chapters to groups of students so that the entire syllabus to be covered. The power-point slides should be put up on the web-site of the College/ Institute, along with the names of the students of the group, the name of the faculty, Department and College on the first slide. The best three works should submit to GTU.