GUJARAT TECHNOLOGICAL UNIVERSITY

Bachelor of Engineering
Subject Code: 3173213
SOFTWARE ENGINEERING
B.E. 7th Semester

Type of course: Core

Prerequisite: Object Oriented Programming fundamental, UML

Rationale:

- To study Software Development Life Cycle, Development models and Agile Software development.
- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
- To learn the process of improving the quality of software work products.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects.
- To expose Software Process Improvement and Reengineering

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks				Total Marks
				Theory Marks		Practical Marks		
L	T	P	C	ESE (E)	PA	ESE (V)	PA (I)	
3	0	2	4	70	30	30	20	150

Contents:

Sr. No.	Content	Total Hrs
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.	06
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.	03
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	05
4	**Software Requirements Specification:** Requirement Gathering and Analysis, Software Requirement Specification (SRS), Formal requirements specification and verification - axiomatic and algebraic specifications	04
5	**Analysis Modeling, Software Design Concepts and Principles:** The Elements of the	06

	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	
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.	04
7	**Risk Analysis & Management:** Reactive versus Proactive Risk Strategies, Software Risks (Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation), Risks Monitoring and Management	03
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.	06
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.	04
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.	04
11	**DevOps Viewpoints:** Overview, Problem Case Definition, Benefits of Fixing Application Development Challenges, DevOps Adoption Approach through Assessment, Solution Dimensions, What is DevOps?, DevOps Importance and Benefits, DevOps Principles and Practices, 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps - Assessment, Definition, Implementation, Measure and Feedback	04

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks					
R Level	U Level	A Level	N Level	E Level	C Level
15	**25**	**20**	**10**	**--**	**--**

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 from above table.

Books:

1. Roger S.Pressman, Software Engineering- A practitioner's Approach, McGraw-Hill International Editions
2. Ian Sommerville, Software engineering, Pearson education Asia
3. Pankaj Jalote, Software Engineering – A Precise Approach Wiley
4. Behhforoz & Frederick Hudson, Software Engineering Fundamentals, OXFORD
5. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.
6. Deepak Gaikwad, Viral Thakkar, DevOps Tools from Practitioner's ViewPoint, Wiley
7. Merlin Dorfman (Editor), Richard H. Thayer (Editor), Software Engineering
8. Robert C. "Uncle Bob" Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design

Course Outcomes: Students will be able to

Sr. No.	CO Statement	Marks % Weightage
1	Prepare SRS (Software Requirement Specification) document and SPMP (Software Project Management Plan) document.	25
2	Apply the concept of Functional Oriented and Object Oriented Approach for Software Design.	30
3	Recognize how to ensure the quality of software product, different quality standards and software review techniques.	25
4	Apply various testing techniques and also upgrade it using advanced Software Engineering Methods.	20

List of Experiments:

(Pl. Note: List of Experiments and Tutorials should be as per theory covered in the class, below mentioned practical are just for the reference purpose)

Tutorial-1

Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases. For each SDLC phase, **identify** the objectives and **summaries** outcomes.

Tutorial-2

Consider any project to be developed in any technology as a Software Architect or Project Manager. **Construct** Software Requirement Specification (SRS) document for the project.

Tutorial-3

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 follow:

- Data entry (0.9 KDSI)
- Data update (0.7 KDSI)
- Query (0.9 KDSI)
- Report generation and display (2 KDSI)

Also take into consideration the following cost drivers with their ratings:

- Storage constraints (Low)
- Experience in developing similar software (High)
- Programming capabilities of the developers (High)
- Application of software engineering methods (High)
- Use of software tools (High)

(All other cost drivers have nominal rating).

Now answer the following:

- **Solve** the problem by **Applying** Basic and intermediate COCOMO
    - Find Project Type?
    - Find Project Size?
    - Find Initial Effort Estimation?
    - Find Adjusted Effort Estimation?
    - Find schedule?
    - Find minimum size of the team you would require to develop this system?
- If your client would pay Rs. 50,000 per month of development, how much would be the likely billing?

Tutorial-4

Function Point: http://conferences.embarcadero.com/article/32094#Bonus .

Analyze the case study and **identify** the error and **solve** it. At the end, need to **assess** calculation part of effort using FP oriented estimation model.

Tutorial-5

Consider the following Java code segment:

```
public Hashtable countAlphabet(String aString){
   Hashtable table = new Hashtable();
   If (aString.length > 4000) return table;
   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;
}
```

1. Guarantees that all independent execution path is exercised at least once;
2. Guarantees that both the true and false side of all logical decisions are exercised;
3. Executes the loop at the boundary values and within the boundaries.

Sketch out Design control flow diagram and **Apply** Cyclomatic complexity for given Code. **Identify** numbers of Independence path require for testing.

Tutorial 6

Subject Project: For below mentioned Systems and other systems assign a mini-project two a group of students to prepare Software documents mentioned as A to E

1. Library Information System
2. Villager Telephone System
3. Waste Management Inspection Tracking System (WMITS)
4. Flight Control System

5. Ambulance Dispatching System

A. Development of Software Requirements Specification (SRS)
B. Function oriented design using SA/SD
C. Object-oriented design using UML
D. Test case design
E. Implementation using Java and testing

Design based Problems (DP)/Open Ended Problem:

1. Assume that you are Software Architect or Project Manager in organization. You have been assigned the task of constructing a website for a specific company with your team. Design and priorities the test cases using test case templates for this project.
2. For Natural Language Processing (NLP) applications, estimate project failure rate.

3. Design and develop an open source method of detecting the DIFFERENCESS between two Engineering designs for the same problem.

List of Open Source Software/learning website:

- www.en.wikipedia.org/wiki/Software_engineering
- www.win.tue.nl
- www.rspa.com/spi
- www.onesmartclick.com/engsineering/software-engineering.html
- www.sei.cmu.edus
- https://www.edx.org/school/uc-berkeleyx
- https://devops.com/most-popular-open-source-devops-tools/
- https://www.guru99.com/devops-tutorial.html

Various Web Based SE Tools

- 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