SOFTWARE ENGINEERING (CS502PC)
B.Tech. III Year I Sem
COURSE PLANNER

I. COURSE PURPOSE:

At the end of the course the student should be in a position to

- Understanding of different software processes and how to choose between them
- How to elicit requirements from a client and specify them

Design in the large, including principled choice of software architecture, the use of modules and interfaces to enable separate development, and design patterns.

Understanding good coding practices, including documentation, contracts, regression tests and daily builds.

Various quality assurance techniques, including unit testing, functional testing, and automated analysis tools

II. PRE-REQUISITES:

It's expected to have basic Programming Skills, Innovative Thinking, and Enthusiasm to learn Management.

III. COURSE OBJECTIVIES:

Understand the software process models such as waterfall and evolutionary models.
Understand the software requirements and SRS document.
Understand the different software architectural styles.
Understand the software testing approaches such as unit testing and integration testing.
Understand the quality control and how to ensure good quality software

IV. COURSE COUTCOMES:

S. No.	Course Outcomes (CO)	Knowledge Level (Blooms Level)
CO1	Identify the minimum requirements for the development of application.	L1: REMEMBER
CO2	Develop and maintain efficient, reliable and cost effective software solutions	L6: CREATE
CO3	Critically Think and evaluate assumptions and arguments.	L5: EVALUATE

V. COURSE CONTENT:

UNIT- I

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software myths. A Generic view of process: Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns, process assessment, personal and team process models.

Process models: The waterfall model, Incremental process models, Evolutionary process models, Specialized process models, The Unified process.

UNIT- II

Software Requirements: Functional and non-functional requirements, User requirements, System requirements, Interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

System models: Context Models, Behavioral models, Data models, Object models, structured methods.

UNIT- III

Design Engineering: Design process and Design quality, Design concepts, the design model.

Creating an architectural design: software architecture, Data design, Architectural styles and patterns, Architectural Design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT- IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing, the art of Debugging.

Product metrics: Software Quality, Metrics for Analysis Model, Metrics for Design Model, Metrics for source code, Metrics for testing, Metrics for maintenance.

UNIT- V

Metrics for Process and Products: Software measurement, metrics for software quality.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

TEXT BOOKS:

1.Software engineering A practitioner's Approach, Roger S Pressman, sixth edition McGraw Hill International Edition.
2.Software Engineering, Ian Sommerville, seventh edition, Pearson education.

REFERENCE BOOKS:

1.Software Engineering, A Precise Approach, PankajJalote, Wiley India, 2010.
2.Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3.Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4.Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5.Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6.Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition 2006.
7.Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8.Software Engineering Principles and Practice, Hans Van Vliet, 3rd edition, John Wiley &Sons Ltd

VI. LESSON PLAN:

Sl.No	Week	Unit No	Topics to be covered	Blow Up	PPT LINKS	PDF LINKS	Course Learning Outcomes	References
1	1	1	Introduction to Software Engineering	• Introduction, • Why is Software Engineering • Important of Software Engineering	https://docs.google.com/presentation/d/10vVub2ASIe1_m7v6Iwqc7KCou1ZP0vQT/edit?usp=sharing&ouid=116900341680369842887&rt	https://drive.google.com/file/d/1GOabbeJ86jyV4v061x7KiK6ijpIw4HQm/view?usp=sharing	Able to Understand what is Software Engineering, Roles and types of softwares and software myths	T1
2			The evolving role of software, Changing Nature of Software	• Defining Software • Software Application Domains • Legacy	https://docs.google.com/presentation/d/10vVub2ASIe1_m7v6Iwqc7KCou1ZP0vQT/edit?usp=sharing&ouid=11790034	https://drive.google.com/file/d/1GOabbeJ86jyV4v061x7KiK6ijpIw4HQm/view?usp=sharing		T1
3			legacy software, Software myths	• WebApps • Mobile Applications • Cloud Computing • Product Line	https://docs.google.com/presentation/d/10vVub2ASIe1_m7v6Iwqc7KCou1ZP0vQT/edit?usp=sharing&ouid=11790034	https://drive.google.com/file/d/1GOabbeJ86jyV4v061x7KiK6ijpIw4HQm/view?usp=sharing		T1
4			A Generic view of process: Software engineering- A layered technology	• What is software Development Myth • Management myths	https://docs.google.com/presentation/d/10vVub2ASIe1_m7v6Iwqc7KCou1ZP0vQT/edit?usp=sharing&ouid=11790034	https://drive.google.com/file/d/1GOabbeJ86jyV4v061x7KiK6ijpIw4HQm/view?usp=sharing	Able to Understand the layered technology of software engineering and how to develop a process framework, Process models and patterns	T1
5	2		A Process framework	• Layered Technology • Process Framework • Capability Maturity Model	https://docs.google.com/presentation/d/10vVub2ASIe1_m7v6Iwqc7KCou1ZP0vQT/edit?usp=sharing&ouid=11790034	https://drive.google.com/file/d/1GOabbeJ86jyV4v061x7KiK6ijpIw4HQm/view?usp=sharing		T1
6			The Capability Maturity Model Integration (CMMI)	• Tools • Methods • Process • A Quality Focus	https://docs.google.com/presentation/d/1xYcLQ_4bnIhRS5VVlJJe3QEt8TPNzv7U/edit?usp=sharing&ouid=1179003	https://drive.google.com/file/d/1cGvOkTcnuxJyWbbIMYy2tj1-VZD_LFqj/view?usp=sharing		T1
7			Process patterns, Process assessment	• What is Pattern Name • Forces • Pattern type	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=11790034168	https://drive.google.com/file/d/1gXBMJKYhNafSEciEdONRGDiS7KtZPe3k/view?usp=sharing		T1

8			Personal and team process models	• Level 0: Incomplete • Level 1: Performed • Level 2: Managed	https://docs.google.com/presentation/d/1cPpcJ5aON-a-71fD3rtleLIhQdmh0tvJ/edit?usp=sharing&ouid=1179003	https://drive.google.com/file/d/1J-Ft37ZO02JjL4N132_x-K9GXMVzeRMF/view?usp=sharing		T1
9	3		Process models: The waterfall model	•Communication • Planning • Modeling • Construction • Deployment	https://docs.google.com/presentation/d/1KzkdGoecXqmdI0DSw1mGVsfMWXHNMwaK/edit?usp=sharing&ouid=11	https://drive.google.com/file/d/18wLnW9WanTMKdsNsFPj_gcrNCjBP99yi/view?usp=sharing		T1
10			Incremental process models	• Standard CMMI Assessment Method for Process Improvement	https://docs.google.com/presentation/d/1KzkdGoecXqmdI0DSw1mGVsfMWXHNMwaK/edit?usp=sharing&ouid=11	https://drive.google.com/file/d/18wLnW9WanTMKdsNsFPj_gcrNCjBP99yi/view?usp=sharing	Able to understand the various software process models	T1
11			Evolutionary process models	• What is prototype •Communication • Quick plan • Modeling	https://docs.google.com/presentation/d/1KzkdGoecXqmdI0DSw1mGVsfMWXHNMwaK/edit?usp=sharing&ouid=11	https://drive.google.com/file/d/18wLnW9WanTMKdsNsFPj_gcrNCjBP99yi/view?usp=sharing		T1
12			Specialized process models, The Unified process	• Reasons for Unified Process • Life cycle of Unified processs • Inception	https://docs.google.com/presentation/d/1KzkdGoecXqmdI0DSw1mGVsfMWXHNMwaK/edit?usp=sharing&ouid=11	https://drive.google.com/file/d/18wLnW9WanTMKdsNsFPj_gcrNCjBP99yi/view?usp=sharing		T1
13			Mock test I					
14			Bridge class I					
15	4	2	Software Requirements: Functional and non-functional requirements	• What is a requirement •Requirements abstraction • Types of requirement	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing	Able to understand the types of software requirements	T1
16			User requirements, System requirements, Interface specification	• Describe functionality or system services •Requirements imprecision	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
17	5		The software requirements document	• What is User Requirements • Importance of User Requirements	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing	Able to prepare SRS for the development of application	T1

18			Requirements engineering process: Feasibility studies	• Introduction of Document • General Description • Specific Requirements	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
19			Requirements elicitation and analysis	• What is Interface Specification • Types of Interfaces	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
20			Requirements validation, Requirements management	• Evaluation Hypotheses: Motivation • Difficulties in Evaluating Hypotheses	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
21	6		System models: Context Models	• Feasibility studies • Requirement Elicitation • Requirement Analysis	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing	Able to understand the different software architectural styles	T1
22			Behavioral models	•Context Model • Behavioral Model • Data Model • Object Model • Structured	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
23			Data models, Object models	• Context Model • Behavioral Model • Data Model • Object Model	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
24	7		Structured methods	• Context Model • Behavioral Model • Data Model • Object Model	https://docs.google.com/presentation/d/1R8KnCqbh2BGHbmB1c2ub4YEe8fg75v_h/edit?usp=sharing&ouid=1179003416803	https://drive.google.com/file/d/1mLaK8kxtxmpOLnIl4DlRFwQVB_mDuUIp/view?usp=sharing		T1
26	8	3	Design Engineering: Design process and Design quality	• What is Design • Design within the Context of Software Engineering	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing	Able to understand design process, concepts, model and patterns	T1
27			Design concepts, Design model	• Software Quality Guidelines and Attributes • The Evolution of Software	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1

28			Pattern based software design	• Abstraction • Architecture • Patterns • Separation of Concerns • Modularity	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
			MID 1 EXAMINATIONS(8-11-202113-11-2021)					
29			Creating an architectural design: software architecture	• What is Architecture • Importance of Architecture	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing	Able to create the software architectural design	T1
30			Data design	Data Design elements Architectural Design Elements Interface	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
32	9		Architectural styles and patterns	• A Brief Taxonomy of Architectural Styles •Architectural Patterns	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
33			Architectural Design	• Representing the System in Context • Defining Archetypes • Refining the	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing	Able to create and assess the software architectural design	T1
34	1	3	Assessing alternative architectural designs	• Basic Structural Modeling • Class Diagrams • Sequence	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
35			Mapping data flow into a software architecture	Representing the System in Context Defining Archetypes	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
36	1		Modeling component-level design: Designing class-based components	Refining the Architecture into Components Describing Instantiations	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing	Able to design class-based, component-level and conventional Components	T1
37			Conducting component-level design	Architectural Design for Web Apps Architectural Design for Mobile Apps	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1

38			object constraint language	Basic Structural Modeling Class Diagrams	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
39	1		Designing conventional components	Sequence Diagrams Collaboration Diagram	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
40			Performing User interface design: Golden rules, User interface analysis and design	Sequence Diagrams Collaboration Diagram	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing	Able to understand Golden rules, user interface analysis and design	T1
41			Interface analysis	Use case Diagram Component Diagram	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
42			Interface design steps, Design evaluation	Use case Diagram Component Diagram	https://docs.google.com/presentation/d/1XL69hLz0rVSJ6J1KtQvpLjxXt56XPhoI/edit?usp=sharing&ouid=11790034168036	https://drive.google.com/file/d/1Ry1l0W8I7FQoIkCMskij6HSnx9n8GK_w/view?usp=sharing		T1
43			Bridge class III					
44	1	4	Testing Strategies: A strategic approach to software testing	• What is Testing • Role of Testing • Importance of Testing	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing	Able to understand various software testing strategies	T1
45			Test strategies for conventional software	• Unit Testing • Integration Testing	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
46			Black-Box and White-Box testing	• Graph-Based Testing Methods • Equivalence Partitioning • Boundary	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
47	1		Validation testing, System testing	• Validation-Test Criteria •Configuration Review • Alpha and Beta Testing	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1

48			The art of Debugging.	• The Debugging Process •Psychological Considerations • Debugging	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
49			Mock test II					
50			Product metrics: Software Quality, Frame work for Product metrics	• Measures, Metrics, and Indicators • The Challenge of Product	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing	Able to understand Product and process metrics	T1
51	1		Metrics for Analysis Model and Design Model	• Function-Based Metrics • Metrics for Specification Quality	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
52			Metrics for source code, testing and maintenance	• Important of Source Code • Measure of Source code Techniques	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
53			Metrics for Process and Products: Software Measurement, Metrics for software quality.	Process Metrics and Software Process Improvement Project Metrics	https://docs.google.com/presentation/d/1LDJWbIWjAzcIkYlllA2pP4KqVDzEO3eY/edit?usp=sharing&ouid=1179003416	https://drive.google.com/file/d/14kDj09VEbcgt8SXs-zik4zRJj-5E-Oli/view?usp=sharing		T1
55	1	5	Risk management: Reactive vs Proactive Risk strategies	• What is Risk • How to Identify the Risk • Reactive versus	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing	Able to understand various Risk strategies, Risk identification, projection and refinement	T1
56			Software risks, Risk identification	• Assessing Overall Project Risk • Risk Components and Drivers	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing		T1
57			Risk projection, Risk refinement	• Developing a Risk Table • Assessing Risk Impact	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing		T1
58			RMMM, RMMM Plan	• Importance of Plan • RMMM document • Risk Information	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing		T1

59	1	Quality Management: Quality concepts, Software quality assurance	• Quality Concepts • Quality Assurance	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing	Able to understand Software quality assurance and how to perform Software Reviews & Formal technical reviews	T1
60		Software Reviews, Formal technical reviews	• Reviews- A Formality Spectrum • Review Metrics and Their Use	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing		T1
61		Statistical Software quality Assurance, Software reliability, ISO Quality Standards	• A Generic Example • Six Sigma for Software Engineering	https://docs.google.com/presentation/d/1Jodusd77JivaFwyktTIc4x2PgnIQeVYP/edit?usp=sharing&ouid=117900341680369841	https://drive.google.com/file/d/1ONgm9G3g-XCW2crpJsvn5PnuD9f6HbVc/view?usp=sharing	Able to understand quality control and how to ensure good	T1
62		Revision					
63		Revision					

TEXT BOOKS:

1.Software engineering A practitioner's Approach, Roger S Pressman, sixth edition McGraw Hill International Edition.
2.Software Engineering, Ian Sommerville, seventh edition, Pearson education.

REFERENCE BOOKS:

1.Software Engineering, A Precise Approach, PankajJalote, Wiley India, 2010.
2.Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3.Fundamentals of Software Engineering, Rajib Mall, PHI, 2005
4.Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5.Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6.Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition 2006.
7.Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8.Software Engineering Principles and Practice, Hans Van Vliet, 3rd edition, John Wiley &Sons Ltd.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes (POs)		**Level**	**Proficiency assessed by**
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	3	Assignments

PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and	2	Assignments
PO3	Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	2	Open ended experiments /
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2	Open ended experiments /
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	2	Mini Project
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	1	--
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	2	--
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.	1	--
PO9	Individual and team work: Function effectively as an individual, and as member or leader in diverse teams, and in multidisciplinary settings.	1	--
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	1	Seminars / Term Paper
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	-	--
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life- long learning in the broadest context of technological change.	1	Competitive\ Examinations

VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes (PSO)		**Level**	**Proficiency assessed by**
PSO1	Foundation of mathematical concepts: To use mathematical methodologies to crack problem using suitable mathematical	2.5	Lectures, Assignments,

	analysis, data structure and suitable algorithm.		Exams
PSO2	Foundation of Computer System: The ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.	3.0	Lectures, Assignments, Exams
PSO3	Foundations of Software development: The ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.	2.0	Lectures, Assignments, Exams

MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFICOUTCOMES:

Course	Program Outcomes												Program Specific Outcomes		
Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	3	1	2	2	-	-	-	-	-	-	2	3	2	3	2
2	2	2	3	2	-	-	-	-	-	-	3	3	3	2	2
3	3	1	2	2	-	-	-	-	-	-	3	2	2	3	2
AVG	**2.6**	**1.4**	**2.3**	**2**	-	-	-	-	-	-	**2.6**	**2.6**	**2.3**	**2.6**	**2**

IX. QUESTION BANK (JNTUH)

UNIT I

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Define software and explain the various characteristics of software?	Knowledge	2
2	Explain software Engineering? Explain the software engineering layers?	Understand	2
3	Explain in detail the capability Maturity Model Integration (CMMI)?	Understand	2
4	Describe with the help of the diagram discuss in detail waterfall model. Give certain reasons for its failure?	Knowledge	2
5	Explain briefly on (a) the incremental model (b) The RAD Model?	Understand	2
6	Explain product and process are related?	Understand	3
7	Explain personal and team process models?	Understand	3
8	Explain process frame work activities?	Knowledge	3
9	Explain and contrast perspective process models and iterative process models?	Understand	3
10	Explain about the evolutionary process models?	Understand	3

Short Answer Questions

S.No	Question	Blooms Taxonomy Level	Course Outcomes
1	Define Software Engineering	Understand	1
2	Discuss Management Myths	Understand	1
3	Discuss Practitioners myths	Understand	2
4	Discuss Customer myths	Understand	2
5	Analyze the changing nature of software	Knowledge	3

UNIT II

Long Answer Questions-

S.N0	Question	Blooms Taxonomy Level	Course Outcome
1	Compare functional requirements with nonfunctional requirements?	Knowledge	3
2	Discuss briefly how requirement validation is done?	Knowledge	3
3	Discuss your knowledge of how an ATM is used; develop a set of use-cases that could serve as a basis for understanding the requirements for an ATM system?	Understand	3
4	Describe four types of non-functional requirements that may be placed on a system. Give examples of each of these types of requirement?	Understand	3
5	Explain how requirements are managed in software project management?	Understand	4
6	Explain Behavioral models and Data model?	Knowledge	4
7	Explain Object models and Context model?	Understand	4
8	Explain SRS document and explain along with its contents?	Understand	4
9	Demonstrate class hierarchy for library by using interface specification?	Understand	4
10	Explain state machine model with a suitable example?	Understand	4

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Discuss Functional requirements	Understand	2
2	Discuss Non Functional requirements	Understand	2
3	Explain about Requirements Gathering.	Understand	2
4	Explain about Requirements Analysis	Understand	1
5	Explain about Requirements Management.	Understand	2

UNIT III

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Discuss briefly the following fundamental concepts of software design: Abstraction Modularity Information hiding	Understand	4
2	Explain briefly the following: Coupling between the modules, The internal Cohesion of a module	Understand	5
3	Discuss the fundamental principles of structured design. Write notes on transform analysis?	Knowledge	5
4	Explain software design? Explain data flow oriented design?	Understand	5
5	Explain the goals of the user interface design?	Understand	5
6	Discuss briefly about the golden rules for the user interface design?	Knowledge	5
7	Discuss interface design steps in a brief manner?	Knowledge	6
8	Explain pattern based software design in a detail manner?	Understand	6
9	Discuss architectural styles and patterns?	Knowledge	6
10	Describe the way of conducting a component level design?	Understand	6

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcomes
1	Discuss about Design Process	Understand	3
2	Discuss about Design Quality	Understand	3
3	Define a Design Model	Understand	2
4	Discuss Design Steps	Understand	2
5	Discuss about Component Level Design	Understand	2

UNIT IV

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcomes
1	Explain about the importance of test strategies for conventional software?	Knowledge	6
2	Discuss and Compare black box testing with white box testing?	Apply	6

3	Compare validation testing and system testing?	Knowledge	6
4	Discuss software quality factors and quality metrics ? Discuss their relative importance?	Understand	7
5	Explain validation test and verification test? Who will involve in this testing? Explain with suitable example?	Apply	7
6	Explain about Metrics for maintenance?	Knowledge	7
7	Explain strategic approach to software testing	Understand	7
8	Demonstrate art of debugging	Knowledge	7
9	Discuss a framework for product metrics	Knowledge	7
10	List the metrics for the design model	Understand	7

Short Answer Questions-

S.N0	Question	Blooms Taxonomy Level	Course Outcome
1	Define Validation Testing and when it will be performed.	Analyze	3
2	Define System Testing	Apply	3
3	Define Debugging	Understand	4
4	Define Software Quality	Understand	3
5	Define White-box Testing	Understand	2

UNIT V

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcomes
1	What is software risks?Elaborate the concepts of Risk management Reactive vs Proactive Risk strategies?	Understand	8
2	Explain about RMMM Plan?	Understand	8
3	Explain about Quality concepts and Quality assurance?	Knowledge	9
4	Explain about formal technical reviews?	Understand	9
5	Explain in detail ISO 9000 quality standards?	Understand	9
6	Briefly explain about formal approaches to SQA and statistical SQA?	Knowledge	9
7	Explain risk projection in detail?	Understand	9
8	Explain seven principals of risk management by developing a risk table?	Knowledge	9
9	Explain six sigma for software engineering?	Knowledge	9
10	Explain quality management with their terms?	Understand	9

Short Answer Questions-

SNo	Question	Blooms Taxonomy Level	Course Outcomes
1	Discuss about Software Reviews	Analyze	4

2	Discuss about ISO 9000 Quality Standards	Understand	2
3	Define about Software Quality Assurance	Understand	2
4	Discuss about RMMM	Understand	2
5	Define the Risk Projection Process	Analyze	3

X. OBJECTIVE QUESTIONS:

FILL IN THE BLANKS:

UNIT-1

1._____is a collection of programs written to service other programs.

Answer :.System software

2.Water fall model is sometimes called as _____

Answer :.classical life cycle model

3._____model is suitable for project requiring shorter development times .

Answer :.RAD model

4.Spiral Model was developed by _____

Answer :.Berry Bohem

5.CMMI stands for _____

Answe r:.Capability Maturity Model Integration.

6.Phases in Unified Process are _____

Answer :.Inception Elaboration Conception Transition

7._____ is a process model that removes defects before they can precipitate serious hazards.

Answer :.Cleanroom software engineering

8.The quick design of a software that is visible to end users leads to _____process model

Answer :prototype

9._____models is not suitable for accommodating any change?

Answer :.Waterfall Model

10.The _____-is a result of combination of elements of Linear Model & Prototyping Model

Answer :.Incremental Model

UNIT-II

1. What are the types of requirements ____,____,____

Answer: Availability,Reliability,Usability

2. Select the developer-specific requirement ___,___

Answer: Portability and Maintainability

3. Which one of the following is not a step of requirement engineering_____

Answer: design

4. FAST stands for_____

Answer: Facilitated Application Specification Technique

5. QFD stands for_____

Answer: quality function deployment.

6. A Use-case actor is always a person having a role that different people may play____

Answer: FALSE

7. The user system requirements are the parts of which document _____

Answer: SRS

8. A stakeholder is anyone who will purchase the completed software system under development____

Answer: False

9. Conflicting requirements are common in Requirement Engineering, with each client proposing his or her version is the right one_____
Answer: True
10. Which is one of the most important stakeholder from the following _____
Answer: Users of the software

UNIT-III

1. _____ is assessed by evaluating the feature set and capabilities of the program and the security of the overall system.
Ans: Functionality
2. _____is a named collection of data that describes a data object.
Ans: Data Abstraction
3. Which are the behavioral aspects of the program architecture_____
Ans :Dynamic models
4_____is a reorganization technique that simplifies the design of a component without changing its
function or behavior.
Ans : Refactoring
5. _____the translation of a data model into a database is pivotal to achieving the business objectives of a system
Ans : application level
6.data mining techniques, also called_____
Ans: knowledge discovery in databases (KDD),
7.A pipe and filter pattern has a set of components is called _____
Ans:filters
8.System response time has _____ important characteristics
Ans: length and variability.
9. _____is measured by processing speed, response time, resource consumption, throughput,and efficiency
Ans: Performance
10. _____ can be used to represent the functional hierarchy of a system.
Ans:Functional model

UNIT-IV

1._____ focuses on the functional requirements of the software.
Answer:Black Box Testing
2._____ Divides all possible inputs into classes such that there are a finite equivalence classes.
Answer:Equivalence partitioning
3.Brute force focus on _____.
Answer: Most common and least efficient, Applied when all else fails and Memory dumps are taken
4.Back tracking mainly used for _____.
Answer: Common debugging approach
5._____are ISO 9126 Quality Factors.
Answer : Functionality and Reliability
6._____is refers to a different set of activities that ensures that the software is traceable to the customer requirements.
Answer:Validation

7._____is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester

Answer: White Box Testing

8. Equivalence Partition mainly focus on_____.

Answer: An input or output range of values such as have only one value in the range becomes a test case.

9.White Box testing also called_____.

Answer:Glass box testing

10._____a structured testing or white box testing technique used for designing test cases intended to examine all possible paths of execution at least once.

Answer :Basis path testing

UNIT-V

1. Risk management is one of the most important jobs for a_____

Answer: Project manager

Explanation: Risk management involves anticipating risks that might affect the project schedule or the quality of the software being developed, and then taking action to avoid these risks.

2. Which of the following risk is the failure of a purchased component to perform as expected?

Answer: Product risk

Explanation: Risks that affect the quality or performance of the software being developed.

3. Which of the following term is best defined by the statement: "There will be a change of organizational management with different priorities."_____

Answer: Management change

4. Which of the following term is best defined by the statement: "The underlying technology on which the system is built is superseded by new technology."?

Answer: Technology change

5. What assess the risk and your plans for risk mitigation and revise these when you learn more about the risk_____

Answer:) Risk monitoring

6. Which of the following risks are derived from the organizational environment where the software is being developed _____

Answer: Organizational risks

7. Which of the following risks are derived from the software or hardware technologies that are used to develop the system _____

Answer: Technology risks

8. Which of the following term is best defined by the statement: "Derive traceability information to maximize information hiding in the design"_____

9. Which of the following strategies means that the impact of the risk will be reduced_____

Answer: Minimization strategies

10. Risk management is now recognized as one of the most important project management tasks____ Answer:- True

MCQ:-

UNIT I

1. In analysis phase the customer will analyze about

a) organization

b) skills of the employee in the organization
c) whether the organization can handle the project or not
d) all of the above
Answer : d

2. Waterfall model is also called as
a) Traditional Model
b) Classic Life Cycle Model
c) Linear Sequential Model
d) all of the above
Answer : d

3. In Waterfall model testing comes at
a) Starting of the project
b) middle of the project
c) final stage of the project
d) in the middle of the project
Answer : c

4.Prototype is the process of implementing
a)the copy of the original product
b) original product
c) can be both a and b
d) none of the above
Answer: a

5.The fastest model among the following is
a) Waterfall Model
b) Spiral Model
c) RAD Model
d) Prototyping Model
Answer: c

6) The framework activities can also be treated as
a) processes
b) Tasks
c) models
d) things
Answer: b

7) By using which model the customer can be happy
a) Waterfall model
b) Unified model
c) prototyping model
d) all of the above
Answer: c

8) The customer requirements continuously changes. This statement is

a) True
b) False
c) Cannot be determined
d) None of the above
Answer: a

9) How many phases are there in a software project?
a) 2
b) 5
c) 3
d) 4
Answer: b

10) Which of the following are iterative models?
a) Spiral model
b) RAD Model
c) Prototype model
d) all of the above
Answer:d

UNIT II

1.Adding comfort to the end user in a project is an example of
a) functional requirement
b) non functional requirement
c) customer requirement
d) business requirement
Answer : b

2.Requirements elicitation is nothing but
a) Requirements gathering
b) requirements analysis
c) requirements specification
d) requirements validation
Answer: a

3.The requirements will be gathered from
a) customer
b) end user
c) can be both a and b
d) none of the above
Answer: c

4.Which of the following is an example of fact finding technique
a) Interviewing
b) questionnaires
c) view point
d) all of the above
Answer: d
5. A View point may differ from person to person. This statement is

a) True
b) False
c) Cannot be determined
d) None of the above
Answer: a

6.A contract between customer an organization I s an example for
a) User requirements
b) business requirements
c) Functional requirements
d) Non functional requirements
Answer: b

7.Requirements validation is a
a) defect detection process
b) defect removal process
c) both a and b
d) none of the above
Answer: c

8. Requirements elicitation is a
a) discovery process
b) defect detection process
c) defect removal process
d) none of the above
Answer: a

9.Structured analysis can also be called as
a) process modeling
b) data flow modeling
c) both a and b
d) none of the above
Answer: c

10.Structure Analysis uses
a) Data Flow diagrams
b) process flow diagrams
c) E-R Diagrams
d) all of the above
answer: a

UNIT III

1.Coupling and cohesion are examples of
a) modular design
b) functional design
c) information hiding
d) all of the above
Answer: a

2. The strength of interconnection between the components in a module is called
a) cohesion
b) coupling
c) both a and b
d) none of the above
Answer: b

3. The strength of interconnection between other modules is called
a) cohesion
b) coupling
c) both a and b
d) none of the above
Answer: a

4) Communication protocols and device interfaces is an example of
a) message coupling
b) data coupling
c) control coupling
d) external coupling
Answer: d

5) What is input of the design phase?
a) Analysis
b) implementation
c) maintenance
d) coding
Answer: a

6) The output of the design phase is used as input to
a) testing
b) coding
c) maintenance
d) all of the above
Answer: b

UNIT IV

1.Process of finding the error is called
a) testing
b) debugging
c) decomposition
d) none of the above
Answer: a

2.Process of removing error is called
a) testing
b) debugging
c) decomposition
d) none of the above
Answer :b

3.Testing an entity in a class is an example for
a) boundary value analysis
b) equivalence class partitioning
c) error based on previous experience
d) all of the above
Answer: b

4. In loops we use which process for testing?
a) boundary value analysis
b) equivalence class partitioning
c) error based on previous experience
d) all of the above
Answer: a
5. A node at which a decision can be made is called?
a) root node
b) predicate node
c) child node
d) all of the above
Answer: b

6.Unit testing is performed by
a) end user
b) developer
c) customer
d) all of the above
Answer: a

7.Acceptance Testing is performed by
a) end user
b) developer
c) customer
d) all of the above
Answer: end user

8.Tesing performed while combing different modules is called
a) acceptance testing
b) integration testing
c) unit testing
d) security testing
Answer: a

9.loop testing is performed as a part of
a) black box testing
b) integration testing
c) control structure testing
d) validation testing
Answer: c

10.White box testing can also be called as
a) glass box testing
b) functional testing
c) both a and b
d) none of the above
Answer: c

UNIT-V

1.Risk projection says about
a) probability of occurrence of risk
b) Consequences due to occurrence of risk
c) both a and b
d) none of the above
Answer: c

2. Which of the following are risk components?
a) performance risk
b) cost risk
c) support risk
d) all of the above
Answer: d
3. Which one is used to point out needed improvements?
a) review
b) inspection
c) walkthrough
d) all of the above
Answer: a
4. What are the things to be tested in SQA activity?
a) quality
b) reliability
c) scalability
d) all of the above
Answer: a
5. Which of the following is used to improve the quality of a product?
a) review
b) inspection
c) walkthrough
d) all of the above
Answer: d
6.V-Model consists of
a) Verification
b) validation
c) both a and b
d) none of the above
Answer: c
7.In V-Model ,testing starts
a) at the end of the project
b) in the middle of the project

c) before starting the project
d) from the starting of the project
Answer: c

8.Initial failure costs are
a) high
b) low
c) moderate
d) can't be determined
Answer: b

9. Quality costs include costs for Training. This statement is
a) true
b) false
c) cannot be determined
Answer: a

10.Quality Assurance and Quality control and
a) verification process
b) validation process
c) both verification and validation processes
d) none of the above
Answer: c

XI. WEBSITES:

https://www.edx.org/course/software-engineering-introduction-ubcx-softeng1x
https://www.edx.org/micromasters/software-development
https://www.edx.org/course/software-engineering-essentials

XII. EXPERT DETAILS: NA

XIII. JOURNALS:

INTERNATIONAL
International Journal of Software Engineering(IJSE)
International Journal of Software Engineering Technologies(IJSET)
NATIONAL
Journal of Software Engineering and Applications
Journal of Software Engineering Research and Development

XIV. LIST OF TOPICS FOR STUDENT SEMINARS:

Various Process Models
Types of Requirements
Coupling and Cohesion in Design Process
Black-Box and White-Box Testing
Software Quality Activities

XV. CASE STUDIES / SMALL PROJECTS:

Preparing SRS Document for E-Billing System, College Management System
Developing prototype by using any language for Employee Management System