Syllabus

Course : USCS404	TOPICS (Credits : 02, Lectures/Week : 03) Software Engineering

Unit	Details	Lectures
I	**Introduction :** The Nature of Software, Software Engineering, The Software Process, Generic Process Model, The Waterfall Model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Component-Based Development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming. **Requirement Analysis and System Modeling :** Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS , Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram. **(Refer Chapters 1, 2, 3 and 4)**	15
II	**System Design :** System/Software Design, Architectural Design, Low-Level Design, Coupling and Cohesion, Functional-Oriented Versus The Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design. **Software Measurement and Metrics :** Product Metrics - Measures, Metrics, and Indicators, Function-Based Metrics, Metrics for Object-Oriented Design, Operation-Oriented Metrics, User Interface Design Metrics, Metrics for Source Code, Halstead Metrics Applied to Testing, Metrics for Maintenance, Cyclomatic Complexity, Software Measurement - Size-Oriented, Function-Oriented Metrics, Metrics for Software Quality. **Software Project Management : Estimation** in Project Planning Process - Software Scope And Feasibility, Resource Estimation, Empirical Estimation Models - COCOMO II, Estimation for Agile Development, The Make/Buy Decision, **Project Scheduling** - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts. **(Refer Chapters 5, 6, 7 and 8)**	15
III	**Risk Management** - Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan. **Software Quality Assurance :** Elements of SQA, SQA Tasks, Goals, and Metrics, Formal Approaches to SQA, Six Sigma, Software Reliability, The ISO 9000 Quality Standards, Capability Maturity Model. **Software Testing :** Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design. **(Refer Chapters 9,10 and 11)**	15

❑❑❑

E-next
THE NEXT LEVEL OF EDUCATION
https://E-next.in
Scanned by CamScanner

UNIT I

Chapter 1 : Introduction to Software Engineering 1-1 to 1-10

1.1 Introduction 1-1
1.2 Software 1-1
1.2.1 Characteristics of Software 1-2
1.2.2 Classes of Software 1-2
✓ **Syllabus Topic :** The Nature of Software 1-3
1.3 The Nature of Software 1-3
1.3.1 The Evolving Role of Software 1-6
✓ **Syllabus Topic :** Software Engineering 1-7
1.4 Software Engineering 1-7
1.5 Software Engineering – A Layered Technology 1-8
1.5.1 Quality Focus 1-9
1.5.2 Software Process 1-9
1.5.3 Software Engineering Methods 1-9
1.5.4 CASE Tools 1-9

Chapter 2 : Software Process 2-1 to 2-41

✓ **Syllabus Topic :** The Software Process 2-1
2.1 Software Process 2-1
✓ **Syllabus Topic :** Generic Process Model 2-2
2.2 Generic Process Model 2-2
2.2.1 Framework Activities 2-2
2.2.2 Process Iteration Activities 2-3
2.2.3 Umbrella Activities 2-4
✓ **Syllabus Topic :** The Waterfall Model 2-5
2.3 Waterfall Model 2-5
✓ **Syllabus Topic :** Incremental Process Models 2-12
2.4 Incremental Process Models 2-12
2.5 RAD Model 2-14
✓ **Syllabus Topic :** Evolutionary Process Models 2-16
2.6 Evolutionary Process Models 2-16
2.6.1 Prototyping 2-17
2.6.2 The Spiral Model 2-19
✓ **Syllabus Topic :** Concurrent Models 2-21
2.7 Concurrent Model 2-21

✓ **Syllabus Topic :** Component-Based Development 2-22
2.8 Component-Based Development 2-22
✓ **Syllabus Topic :** The Unified Process Phases 2-24
2.9 The Unified Process Phases 2-24
✓ **Syllabus Topic :** Agile Development - Agility 2-31
2.10 Agile Development 2-31
2.10.1 Agility 2-31
✓ **Syllabus Topic :** Agile Process 2-32
2.10.2 Agile Process 2-32
✓ **Syllabus Topic :** Extreme Programming 2-32
2.10.3 Extreme Programming 2-32

Chapter 3 : Requirement Analysis and System Modeling 3-1 to 3-45

3.1 Requirements 3-1
✓ **Syllabus Topic :** Requirements Engineering 3-2
3.2 Requirements Engineering 3-2
3.3 Requirements Engineering Tasks 3-3
3.3.1 Inception 3-3
✓ **Syllabus Topic :** Eliciting Requirements 3-4
3.3.2 Eliciting Requirements 3-4
3.3.3 Elaboration 3-4
3.3.4 Negotiation 3-4
3.3.5 Software Requirement Specification (SRS) 3-5
3.3.6 Requirements Validation 3-5
3.3.7 Requirements Management 3-6
3.4 Requirement Elicitation 3-7
3.4.1 Collaborative Requirements Gathering 3-7
3.4.2 Quality Function Deployment 3-9
3.4.3 User Scenarios 3-10
3.4.4 Elicitation Work Products 3-12
3.5 Software Requirement Specification 3-12
3.5.1 Benefits (Need) of SRS 3-12
✓ **Syllabus Topic :** Characteristics of SRS 3-13
3.5.2 Characteristics of SRS 3-13
✓ **Syllabus Topic :** Components of SRS 3-14
3.5.3 Components of SRS 3-14
3.6 Identifying the Stakeholders 3-16
3.7 Issues of Requirement Gathering 3-17

E-next THE NEXT LEVEL OF EDUCATION

Scanned by CamScanner

https://E-next.in

3.8 Techniques of Requirement Gathering 3-18
3.8.1 Questionnaires 3-18
3.8.2 Reviews 3-19
3.8.3 Interviews 3-19
3.8.4 Workflows 3-23
3.8.5 Building Prototypes 3-24
3.8.6 Structured Walkthroughs 3-25
3.9 Case Studies of SRS 3-25
3.9.1 Online Library System 3-25
3.9.2 Purchase Order System 3-30
3.9.3 Hospital Management System 3-33
3.9.4 Catering System 3-38

Chapter 4 : Object-oriented Design using UML 4-1 to 4-55

4.1 Introduction 4-1
4.2 UML 4-2
4.2.1 Features of UML 4-2
4.2.2 Need of UML 4-3
4.2.3 UML Advantages 4-4
4.2.4 UML Diagrams 4-4
✓ **Syllabus Topic :** Use-case Diagrams 4-5
4.3 Use-case Diagrams 4-5
4.3.1 Need of Use Case Diagrams 4-5
4.3.2 Elements of a Use Case Diagram 4-6
4.3.3 Use Cases Relationships 4-8
✓ **Syllabus Topic :** Class Diagram 4-11
4.4 Class Diagram 4-11
4.4.1 Identifying Classes and Objects 4-11
4.4.1(A) Noun Phrase Approach 4-12
4.4.1(B) Common Class Pattern Approach 4-16
4.4.1(C) Use Case Driven Approach 4-17
4.4.1(D) CRC Approach 4-17
4.4.2 Identifying Attributes 4-20
4.4.3 Defining Methods 4-20
4.4.4 Finalizing the Object (Class) Definition 4-21
✓ **Syllabus Topic :** Object Diagram 4-25
4.5 Object Diagram 4-25
4.6 Packages 4-27

4.7 Interaction Diagram 4-27
✓ **Syllabus Topic :** Sequence Diagram 4-28
4.7.1 Sequence Diagram 4-28
✓ **Syllabus Topic :** Collaboration Diagram 4-33
4.7.2 Collaboration Diagram 4-33
✓ **Syllabus Topic :** State-chart Diagram 4-36
4.8 State-chart Diagram 4-36
✓ **Syllabus Topic :** Activity Diagram 4-42
4.9 Activity Diagram 4-42
4.10 Implementation Diagram 4-48
✓ **Syllabus Topic :** Component Diagram 4-49
4.10.1 Component Diagram 4-49
✓ **Syllabus Topic :** Deployment Diagram 4-52
4.10.2 Deployment Diagram 4-52

UNIT II

Chapter 5 : System Design 5-1 to 5-22

5.1 Introduction 5-1
✓ **Syllabus Topic :** System/Software Design 5-2
5.2 System Design 5-2
5.2.1 Thumb Rule 5-2
5.2.2 Design Modeling Principles 5-3
✓ **Syllabus Topic :** Architectural Design 5-4
5.3 Architectural Design (High-level Design) 5-4
✓ **Syllabus Topic :** Low-Level Design 5-7
5.4 Low-Level Design 5-7
5.5 Top-down and Bottom-up Design 5-7
✓ **Syllabus Topic :** Functional-Oriented Versus The Object Oriented-Design Approach 5-8
5.6 Function Oriented vs. Object Oriented Design 5-8
✓ **Syllabus Topic :** Design Specification 5-12
5.7 Design Specification 5-12
5.8 Partitioning the Model 5-13
5.9 Abstraction and its Types 5-13
5.10 Modularity 5-14
✓ **Syllabus Topic :** Coupling and Cohesion 5-16
5.11 Coupling and Cohesion 5-16

E-next THE NEXT E VERSION OF EDUCATION
Scanned by CamScanner
https://E-next.in

5.12 Structure Charts 5-17
5.13 Flow Charts 5-18
✓ **Syllabus Topic :** Verification for Designs 5-19
5.14 Verification for Designs 5-19
✓ **Syllabus Topic :** Monitoring and Control for Design 5-21
5.15 Metrics, Monitoring and Control for Design 5-21

Chapter 6 : Software Measurement and Metrics **6-1 to 6-25**

✓ **Syllabus Topic :** Product Metrics – Measures, Metrics and Indicators 6-1
6.1 Measures, Metrics and Indicators 6-1
6.1.1 Need of Metrics 6-1
6.1.2 Role of Software Metrics 6-2
6.1.3 Types of Testing Metrics 6-6
6.2 Project Metrics 6-6
6.3 Process Metrics 6-7
6.4 Product Metrics 6-7
6.5 Closure Metrics 6-7
✓ **Syllabus Topic :** Function-based Metrics 6-8
6.6 Function-based Metrics 6-8
✓ **Syllabus Topic :** Metrics for Object-Oriented Design 6-11
6.7 Metrics for Object-Oriented Design 6-11
✓ **Syllabus Topic :** Operation-Oriented Metrics 6-13
6.7.1 Operation-oriented Metrics 6-13
6.7.2 Use Case Point (UCP) Estimation Method 6-13
✓ **Syllabus Topic :** Other Testing Metrics 6-15
6.8 Other Testing Metrics 6-15
✓ **Syllabus Topic :** Cyclomatic Complexity 6-15
6.9 Cyclomatic Complexity 6-15
✓ **Syllabus Topic :** Software Measurement 6-22
6.10 Software Measurement 6-22
✓ **Syllabus Topic :** Metrics for Software Quality 6-22
6.11 Metrics for Software Quality 6-22

Chapter 7 : Software Project Management Estimation **7-1 to 7-12**

✓ **Syllabus Topic :** Estimation in Project Planning Process 7-1
7.1 Estimation in Project Planning Process 7-1
✓ **Syllabus Topic :** Software Scope and Feasibility 7-2

7.1.1 Software Scope and Feasibility 7-2
✓ **Syllabus Topic :** Resource Estimation 7-2
7.2 Resource Estimation 7-2
7.2.1 Software Cost Estimation Process 7-3
7.3 Cost Estimation Techniques 7-4
7.4 Cost Estimation Parameters 7-5
7.5 Pragmatic Software Cost Estimation 7-6
✓ **Syllabus Topic :** Empirical Estimation Models 7-7
7.6 Empirical Estimation Models 7-7
7.6.1 COCOMO Model 7-8
✓ **Syllabus Topic :** COCOMO II Model 7-9
7.6.2 COCOMO II Model 7-9
✓ **Syllabus Topic :** Estimation for Agile Development 7-11
7.6.3 Estimation for Agile Development 7-11
✓ **Syllabus Topic :** The Make/Buy Decision 7-12
7.6.4 The Make/Buy Decision 7-12

Chapter 8 : Project Scheduling **8-1 to 8-12**

8.1 Project Scheduling 8-1
✓ **Syllabus Topic :** Basic Principles 8-2
8.2 Basic Principles 8-2
✓ **Syllabus Topic :** Relationship between People and Effort 8-2
8.3 Relationship between People and Effort 8-2
✓ **Syllabus Topic :** Effort Distribution 8-3
8.4 Effort Distribution 8-3
8.4.1 Defining a Task Set 8-3
✓ **Syllabus Topic :** Time-Line Charts 8-4
8.5 Time-line Charts 8-4
8.5.1 Gantt Chart 8-5
8.5.2 PERT/CPM Chart 8-7
8.5.2(A) Difference between PERT and CPM 8-8
8.5.2(B) Use of PERT & CPM Chart 8-8
8.5.2(C) Solved PERT/CPM Network Diagrams 8-9

UNIT III

Chapter 9 : Risk Management **9-1 to 9-12**

✓ **Syllabus Topic :** Software Risks 9-1
9.1 Software Risks 9-1

Scanned by CamScanner
E-next
THE NEXT LEVEL OF EDUCATION
https://E-next.in

9.1.1 Types of Risks 9-2
9.2 Seven Principles of Risk Management 9-3
9.3 Risk Management Process 9-3
✓ **Syllabus Topic :** Risk Identification 9-4
9.3.1 Risk Identification 9-4
9.3.2 Risk Analysis 9-5
9.3.3 Risk Planning 9-6
9.3.4 Risk Monitoring and Control 9-7
✓ **Syllabus Topic :** Risk Projection 9-8
9.4 Risk Projection (Estimation) 9-8
9.4.1 Risk Table 9-8
✓ **Syllabus Topic :** Risk Refinement 9-9
9.5 Risk Refinement 9-9
✓ **Syllabus Topic :** RMMM Plan 9-9
9.6 RMMM Plan 9-9

Chapter 10 : Software Quality Assurance **10-1 to 10-27**

10.1 Quality 10-1
10.1.1 Quality Perceptions 10-1
10.1.2 Importance of Software Quality 10-1
10.1.3 Quality Challenges 10-2
10.1.4 Causes of Errors in a Software Product 10-2
10.2 Quality Control and Quality Assurance 10-3
10.2.1 Changing View of Quality 10-4
10.2.2 PDCA Cycle 10-4
10.3 SQA : Software Quality Assurance 10-5
10.3.1 Need of SQA 10-5
✓ **Syllabus Topic :** SQA Tasks 10-5
10.3.2 SQA Tasks 10-5
✓ **Syllabus Topic :** Goals 10-6
10.3.3 SQA Goals 10-6
✓ **Syllabus Topic :** Elements of SQA 10-7
10.3.4 Elements of SQA 10-7
✓ **Syllabus Topic :** Metrics 10-8
10.4 Metrics for Software Quality 10-8
10.5 Quality Factors 10-10
✓ **Syllabus Topic :** Formal Approaches to SQA 10-13
10.6 Formal Approaches to SQA 10-13
10.6.1 Proof of Correctness 10-13

10.6.2 Statistical Quality Assurance 10-14
10.6.3 Clean Room Software Development 10-14
✓ **Syllabus Topic :** Six Sigma 10-17
10.7 Six Sigma 10-17
✓ **Syllabus Topic :** Software Reliability 10-18
10.8 Software Reliability 10-18
10.8.1 Reliability Measures 10-18
10.8.2 Reliability Models 10-19
10.9 SQA Planning and Standards 10-21
10.9.1 SQA Standards 10-22
10.9.2 ISO Quality Standards 10-23
✓ **Syllabus Topic :** The ISO 9000 Quality Standards 10-23
10.9.2(A) ISO 9000 Quality Standard 10-23
10.9.2(B) ISO 9001 10-24
10.9.2(C) ISO 9126 Quality Standard 10-25
✓ **Syllabus Topic :** Capability Maturity Model 10-26
10.10 CMMI Process Improvement Framework 10-26

Chapter 11 : Software Testing **11-1 to 11-66**

✓ **Syllabus Topic :** Verification and Validation 11-1
11.1 Verification and Validation 11-1
11.1.1 Static and Dynamic Verification 11-2
11.1.2 V and V Model 11-3
✓ **Syllabus Topic :** Introduction to Testing 11-4
11.2 Introduction to Testing 11-4
11.2.1 Bugs 11-6
✓ **Syllabus Topic :** Testing Principles 11-8
11.3 Testing Principles 11-8
✓ **Syllabus Topic :** Testing Objectives 11-12
11.4 Testing Objectives 11-12
✓ **Syllabus Topic :** Test Oracles 11-13
11.5 Test Oracles 11-13
✓ **Syllabus Topic :** Levels of Testing 11-14
11.6 Levels of Testing 11-14
11.6.1 Level 1 : Unit Testing 11-14
11.6.2 Level 2 : Component Testing 11-15
11.6.3 Level 3 : Integration Testing 11-15
11.6.3(A) Top-down Testing 11-16

Scanned by CamScanner
E-next THE NEXT LEVEL OF EDUCATION
https://E-next.in

11.6.3(B) Bottom-up Testing 11-17
11.6.3(C) Big-Bang Approach 11-18
11.6.3(D) Sandwich Testing 11-19
11.6.4 Level 4 : System Testing 11-19
11.6.5 Level 5 : Acceptance Testing 11-22
11.6.6 Integration Testing vs. System Testing 11-23
11.7 Static Verification 11-24
11.7.1 Code Review 11-24
11.7.2 Desk Checking 11-25
11.7.3 Code Walk-Through 11-25
11.7.4 Code Inspection 11-26
✓ **Syllabus Topic :** White-box Testing / Structural Testing 11-28
11.8 White-box Testing (Structural Testing) 11-28
11.8.1 Code Coverage Testing 11-30
11.8.2 Code Complexity Testing 11-32
✓ **Syllabus Topic :** Functional / Black Box Testing 11-40
11.9 Functional (Black Box) Testing 11-40
11.9.1 Requirements based Testing 11-41
11.9.2 Positive and Negative Testing 11-41
11.9.3 Boundary Value Analysis 11-42
11.9.4 Equivalence Class Partition 11-43
11.9.5 Domain Testing 11-46
11.9.6 Cause Effect Graphing 11-46
11.10 Black Box vs. White Box 11-49
✓ **Syllabus Topic :** Test Plan 11-49
11.11 Test Plan 11-49
11.11.1 Test Plan Template 11-50
11.11.2 Test Plan Examples 11-54
✓ **Syllabus Topic :** Test-Case Design 11-60
11.12 Test Cases 11-60
11.12.1 Login Test Case 11-61
11.12.2 Email Address Test Case 11-64
11.12.3 Test Case for Calculator 11-65
11.12.4 Test Case for ATM 11-65
- **Model Question Papers** ..... M-1 to M-5
- **Appendix A : Solved University Question Paper of April 2018** ..... A-1 to A-4

□□□

UNIT I

CHAPTER 1

Introduction to Software Engineering

Syllabus :
The Nature of Software, Software Engineering.

1.1 Introduction

– Computers are becoming an unavoidable necessity or say the part and parcel of our life. We do most of our day's work using computers and digital devices – to book railway tickets or movie tickets, search for some books on amazon, bank transactions, and any task you name it is done using computers. Therefore it is very important to build these computerized systems in an effective way.

– Building such systems requires certain technical as well as communication skills and capabilities to understand and follow a planned and systematic procedure.

1.2 Software

Q. Define software.

– Software is a set of instructions to acquire inputs and to process them to produce the desired output in terms of functions and performance as determined by the user of the software. It is developed to handle an *Input-Process-Output* system to achieve predetermined goals.

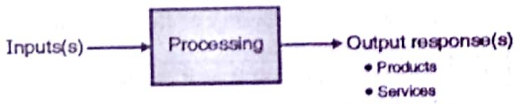

Fig. 1.2.1 : Basic diagram of a software

– **Software encompasses of**
1. Instructions (Computer programs) 2. Documents (Describes programs)
3. Architecture including Data Structures (Enables programs)

– Software includes things such as websites, programs or video games that are coded by programming languages like C or C++.

☞ **Software - a Product and a Vehicle :**

"Software plays a dual role - both as a *product* and a *vehicle* that delivers a product".

☞ **Software is a product :**

1. Delivers computing potential

Scanned by CamScanner