

UNIT I

Chapter 1 : Software Engineering Fundamentals 1-1 to 1-11

Nature of Software, Software Engineering Practice, Software Process, Software Myths.

1.1 Introduction to Software Engineering 1-1

1.2 Nature of Software 1-1

 1.2.1 Absence of Fundamental Theory1-2

 1.2.2 Ease of Change1-2

 1.2.3 Rapid Evolution of Technologies1-2

 1.2.4 Low Manufacturing Cost1-2

1.3 Software Engineering Practice 1-2

 1.3.1 The Essence of Practice1-3

 1.3.2 Core Principles.....1-4

1.4 Software Process 1-5

 1.4.1 Umbrella Activities1-6

1.5 Software Myths..... 1-7

 1.5.1 Management Level Myths (or Manager Level Myths).....1-8

 1.5.2 Customer Level Myths.....1-9

 1.5.3 Practitioner Level Myths (or Developer Level Myths)..... 1-10

1.6 The Characteristics of Software 1-11

Chapter 2 : Process Models 2-1 to 2-7

A Generic Process Model, Linear Sequential Development Model, Iterative Development Model, The incremental Development Model.

2.1 Generic Process Model 2-1

2.2 Linear Sequential Development Model (Waterfall Model) 2-2

2.3 Iterative Development Model : Component Based Development 2-4

2.4 The Incremental Development Model 2-5

2.5 Comparison between Evolutionary and Incremental Models..... 2-6

Chapter 3 : Agile Software Development and Practices 3-1 to 3-24

Agile manifesto, agility principles, Agile methods, myth of planned development, Introduction to Extreme programming and Scrum. Test driven development, pair programming, continuous integration in DevOps, Refactoring

3.1 Introduction to Agile Software Development Process 3-1

3.2	Agility Principles	3-2
3.2.1	Relation of Agility and Cost of Change	3-3
3.3	Agile Methods	3-4
3.3.1	Agile Manifesto.....	3-4
3.3.2	Agile/XP methodology	3-4
3.3.3	Benefits of Agile and XP Methodology project management.....	3-5
3.4	Myth of Planned Development	3-5
3.5	Introduction to Extreme Programming	3-6
3.5.1	XP Values.....	3-6
3.5.2	The XP Process	3-7
3.5.3	Industrial XP (IXP)	3-7
3.5.4	The XP Debate.....	3-8
3.6	SCRUM	3-8
3.6.1	Process Flow.....	3-9
3.6.2	Scrum Roles	3-10
3.6.3	Scrum Cycle Description	3-11
3.6.4	Product Backlog.....	3-12
3.6.5	Sprint Planning Meeting.....	3-13
3.6.6	Sprint Backlog.....	3-14
3.6.7	Sprint Execution.....	3-15
3.6.8	Daily Scrum Meeting.....	3-15
3.6.9	Maintaining Sprint Backlog and Burn-Down Chart.....	3-16
3.6.10	Sprint Review and Retrospective	3-17
3.7	Agile Practices	3-17
3.7.1	Pair Programming.....	3-17
3.7.2	Refactoring.....	3-19
3.7.3	Test Driven Development (TDD).....	3-20
3.7.4	Continuous Integration in DevOps.....	3-22
3.7.5	Exploratory Testing Versus Scripted Testing	3-23

UNIT II

Chapter 4 : Requirements Engineering **4-1 to 4-13**

User and system requirements, Functional and non-functional requirements, requirements engineering (elicitation, specification, validation, negotiation) prioritizing requirements (Kano diagram), requirement traceability matrix (RTM).

4.1	User and System Requirements	4-1
4.1.1	Importance of Requirement Engineering	4-2
4.2	Functional and Non-functional Requirements	4-2
4.2.1	Functional Requirements	4-3
4.2.2	Non-functional Requirements	4-3
4.3	Introduction to Requirements Engineering	4-4
4.4	Requirements Engineering	4-4
4.4.1	Elicitation.....	4-4
4.4.1(A)	Collaborative Requirements Gathering	4-5
4.4.1(B)	Quality Function Deployment.....	4-5
4.4.1(C)	Usage Scenarios.....	4-6
4.4.1(D)	Elicitation Work Product	4-6
4.4.1(E)	Elicitation Techniques.....	4-7
4.4.1(F)	Developing Use Cases	4-7
4.4.2	Specification	4-8
4.4.2(A)	Requirement Monitoring.....	4-10
4.4.3	Validation.....	4-10
4.4.4	Negotiation.....	4-10
4.5	Prioritizing Requirements (Kano diagram)	4-11
4.6	Requirement Traceability Matrix - RTM (Requirement Management)	4-12
4.7	Requirement Characteristic	4-12

Chapter 5 : Software Requirements Specification (SRS)	5-1 to 5-6
--	-------------------

Software requirements Specification document, structure of SRS, writing a SRS, structured SRS for online		
5.1	Software Requirements Specification	5-1
5.2	Writing Software Requirements Specifications	5-2
5.2.1	What is a Software Requirements Specification?	5-2
5.2.2	What Kind of Information Should an SRS Include?	5-3
5.3	SRS Template	5-3
5.3.1	Characteristics of an SRS.....	5-4
5.4	Structured Specifications for an Insulin Pump Case Study	5-4
5.5	Tabular Specifications for an Insulin Pump Case Study	5-5

Chapter 6 : Requirements Analysis **6-1 to 6-25**

Analysis Model, data modeling, scenario based modeling, class based modeling, Flow oriented modeling, behavioral modeling-Introduction to UML diagrams.

6.1	Requirement Analysis	6-1
6.1.1	Analysis Model.....	6-1
6.1.2	Analysis Rules of Thumb.....	6-2
6.1.3	Domain Analysis	6-2
6.1.4	Requirements Modeling Approaches	6-3
6.2	Data Modelling	6-3
6.2.1	Data Objects.....	6-3
6.2.2	Data Attributes	6-4
6.2.3	Relationship.....	6-4
6.2.4	Cardinality and Modality.....	6-4
6.3	Introduction to UML Diagram (Scenario Based Modeling)	6-6
6.3.1	Diagramming in UML.....	6-7
6.3.2	Developing Use Cases Diagram	6-9
6.3.3	Developing Activity Diagram	6-10
6.3.4	Swim Lane Diagram	6-11
6.3.5	Class Diagram	6-12
6.3.5(A)	Aggregation.....	6-12
6.3.5(B)	Generalization.....	6-13
6.3.5(C)	Associations and Dependency	6-13
6.4	Class Based Modeling.....	6-15
6.4.1	Basic Design Principles.....	6-16
6.4.2	Conducting Component-Level Design	6-17
6.5	Flow Oriented Modeling	6-17
6.5.1	Data Flow Model.....	6-17
6.5.2	Control Flow Model.....	6-19
6.5.3	Control Specifications.....	6-20
6.5.4	Process Specifications (PSPEC).....	6-21
6.6	Behavioural Modeling	6-21
6.6.1	Identifying the Events with Use-Cases.....	6-22
6.6.2	Create the Sequence for Use-Case	6-22

6.6.3	State Machine Diagram with Orthogonal States	6-23
6.6.3(A)	Orthogonal States	6-24
➤	Model Question Paper (In Sem.)	M-1 to M-1

UNIT III

Chapter 7 : Design Engineering **7-1 to 7-16**

Design Process & quality, Design Concepts, design Model, Pattern-based Software Design.

7.1	Introduction to Design Engineering	7-1
7.2	Design Process.....	7-1
7.3	Design Quality	7-1
7.3.1	Quality of Design Guidelines.....	7-2
7.3.2	The Quality Attributes.....	7-2
7.4	Design Concepts.....	7-3
7.4.1	Abstraction.....	7-3
7.4.2	Architecture.....	7-4
7.4.3	Patterns.....	7-4
7.4.4	Modularity.....	7-4
7.4.5	Information Hiding.....	7-6
7.4.6	Functional Independence	7-6
7.4.7	Refinement.....	7-9
7.4.8	Refactoring.....	7-9
7.4.8(A)	Importance of refactoring.....	7-9
7.4.9	Design Classes.....	7-9
7.4.10	Differentiation between Abstraction and Refinement	7-10
7.5	The Design Model.....	7-10
7.5.1	Data Design Elements.....	7-11
7.5.2	Architectural Design Elements.....	7-11
7.5.3	Interface Design Elements.....	7-12
7.5.4	Component-Level Design Elements.....	7-12
7.5.5	Deployment-Level Design Elements	7-13
7.5.6	Translating Requirements Model to Design Model	7-13
7.5.7	Guidelines for the Data Design	7-14
7.6	Pattern-Based Software Design.....	7-15

7.6.1	Describing a Design Pattern.....	7-15
7.6.2	Using Patterns in Design	7-15
7.6.3	Frameworks.....	7-16

Chapter 8 : Architectural Design and Component Level Design

8-1 to 8-15

Design Decisions, Views, Patterns, Application Architectures.

8.1	Introduction to Architectural Design.....	8-1
8.2	Architectural Design Decisions	8-3
8.3	Architectural Views.....	8-4
8.4	Architectural Patterns.....	8-6
8.4.1	Software Architecture	8-6
8.5	Application Architectures	8-7
8.5.1	Transaction Processing Systems	8-8
8.5.2	Language Processing Systems	8-9
8.6	Conducting Component level Design.....	8-11
8.7	Designing Class-based Components	8-13
8.7.1	Basic Design Principles.....	8-14
8.7.2	Component-Level Design Steps.....	8-14

Chapter 9 : User Interface Design

9-1 to 9-15

The golden rules, Interface Design steps & Analysis, Design Evaluation.

9.1	User Interface Design.....	9-1
9.1.1	Type of User Interface	9-2
9.1.2	Characteristics of Good User Interface	9-4
9.1.3	Benefits of Good Interface Design	9-4
9.2	The Golden Rules.....	9-4
9.2.1	Place the user in Control	9-4
9.2.2	Reduce the User’s Memory Load	9-5
9.2.3	Make the Interface Consistent.....	9-6
9.2.4	Necessity of a Good User Interface	9-7
9.3	Shneiderman’s 8 Golden Rules for UI Analysis.....	9-7
9.4	Interface Analysis and Design Models	9-8
9.4.1	Interface Analysis and Design Models.....	9-9

9.4.2	User Interface Design Process	9-9
9.5	Interface Design Steps and Analysis.....	9-10
9.5.1	Applying Interface Design Steps	9-11
9.5.2	User Interface Design Patterns.....	9-11
9.5.3	Interface Design Issues	9-11
9.5.4	Interface Design Evaluation.....	9-12
9.6	Design Evaluation.....	9-13
9.7	WebApp Interface Design.....	9-14
9.7.1	WebApp Design Principles.....	9-14

UNIT IV

Chapter 10 : Project Planning	10-1 to 10-14
--------------------------------------	----------------------

Project initiation, Planning Scope Management, Creating the Work Breakdown Structure, scheduling: Importance of Project Schedules, Developing the Schedule using Gantt Charts, PERT/ CPM

10.1	Introduction to Project Planning.....	10-1
10.2	Project Initiation	10-1
10.2.1	Business Case.....	10-2
10.2.2	Feasibility Study	10-2
10.2.3	Project Charter	10-2
10.2.4	Project Team.....	10-2
10.2.5	Project Office	10-2
10.2.6	Phase Review	10-2
10.3	Planning Scope Management	10-3
10.3.1	Obtaining Information Necessary for Scope.....	10-3
10.3.2	Feasibility	10-4
10.3.3	A Scoping Example.....	10-4
10.4	Creating the Work Breakdown Structure.....	10-4
10.5	Project Scheduling	10-5
10.5.1	The Structure of Estimation Models	10-5
10.5.2	The COCOMO II Model.....	10-6
10.5.3	The Software Equation.....	10-7
10.6	Importance of Project Schedules.....	10-8
10.7	Developing the Schedule using Gantt Charts.....	10-9

10.7.1	Tracking the Schedule	10-9
10.7.2	Schedule and Cost Slippage.....	10-10
10.8	Project Scheduling Tools and Techniques : PERT/ CPM	10-10
10.8.1	CPM (Critical Path Method)	10-11
10.8.2	PERT (Program Evaluation and Review Technique).....	10-11
10.8.2(A)	Advantages using PERT	10-14

Chapter 11 : Project Management	11-1 to 11-13
--	----------------------

The Management Spectrum, People, Product, Process, Project, The W5HH Principle, Metrics in the Process and Project Domains, Software Measurement : size &function-oriented metrics (FP & LOC), Metrics for Project

11.1	The Management Spectrum.....	11-1
11.1.1	The People	11-1
11.1.1(A)	Stake Holders	11-2
11.1.1(B)	Team Leaders.....	11-2
11.1.1(C)	Software Team.....	11-3
11.1.1(D)	Agile Teams.....	11-4
11.1.1(E)	Co-ordination and Communication Issues	11-4
11.1.2	The Product.....	11-4
11.1.3	The Process	11-5
11.1.4	The Project	11-5
11.2	The W5HH Principle.....	11-6
11.3	Metrics in the Process and Project Domains	11-6
11.3.1	Process Metrics	11-7
11.3.2	Project Metrics (Metrics For Project).....	11-7
11.4	Software Measurement	11-8
11.4.1	Size-Oriented Metrics	11-8
11.4.2	Function-Oriented Metrics (FP and LOC)	11-9
11.4.3	Reconciling LOC and FP Metrics	11-9
11.4.4	Comparison between FP and LOC.....	11-10
11.4.5	Object-Oriented Metrics	11-10
11.4.6	Integrating Metrics within the Software Process.....	11-11

Chapter 12 : Project Estimation **12-1 to 12-5**

Software Project Estimation, Decomposition Techniques, Cost Estimation Tools and Techniques, Typical Problems with IT Cost Estimates.

12.1 Software Project Estimation..... 12-1

12.2 Decomposition Techniques..... 12-1

 12.2.1 Problem Decomposition..... 12-1

 12.2.2 Process Decomposition..... 12-2

12.3 Cost Estimation Tools and Techniques..... 12-3

12.4 Typical Problems with IT Cost Estimates 12-4

UNIT V

Chapter 13 : Quality Concepts **13-1 to 13-8**

Quality, software quality, Quality Metrics, software quality dilemma, achieving software quality.

13.1 Quality 13-1

13.2 Software Quality 13-1

 13.2.1 McCall’s Quality Factors 13-2

 13.2.2 ISO 9126 Quality Factors 13-3

13.3 Quality Metrics 13-4

 13.3.1 Product Metrics 13-4

 13.3.1(A) The Challenge of Product Metrics 13-5

 13.3.1(B) Measurement Principles..... 13-5

 13.3.2 Process Metrics 13-6

 13.3.2(A) Process Metrics and Software Process Improvement..... 13-6

 13.3.3 Project Metrics..... 13-6

13.4 Software Quality Dilemma 13-7

13.5 Achieving Software Quality..... 13-7

 13.5.1 Software engineering methods..... 13-7

 13.5.2 Project management techniques 13-7

 13.5.3 Quality control 13-8

 13.5.4 Quality assurance 13-8

Chapter 14 : Software Testing **14-1 to 14-35**

Introduction to Software Testing, Principles of Testing, Test plan, Test case, Types of Testing, Verification & Validation, Testing strategies, Defect Management, Defect Life Cycle, Bug Reporting, debugging.

14.1 Introduction to Software Testing 14-1

14.2 Software Testing Fundamentals 14-2

 14.2.1 Test Characteristics (Attributes of good test)..... 14-3

14.3 Principles of Testing..... 14-3

14.4 Testing Life Cycles..... 14-4

 14.4.1 Requirement Analysis 14-5

 14.4.2 Test Planning..... 14-5

 14.4.3 Test Case Development..... 14-5

 14.4.4 Test Execution 14-5

 14.4.5 Test Cycle Closure 14-5

14.5 Test Plan..... 14-5

 14.5.1 Test strategy vs. Test plan 14-6

 14.5.2 The importance of a test plan..... 14-6

 14.5.3 How to write a test plan ? 14-6

14.6 Test Case..... 14-7

 14.6.1 How to write test cases for software 14-7

 14.6.2 Benefits of Writing Test Cases 14-8

14.7 Types of Testing..... 14-8

 14.7.1 White-Box Testing..... 14-9

 14.7.1(A) Basis Path Testing..... 14-9

 14.7.1(B) Control Structure Testing 14-13

 14.7.2 Black-Box Testing..... 14-15

 14.7.2(A) Graph-Based Testing Method..... 14-16

 14.7.2(B) Equivalence Partitioning..... 14-16

 14.7.2(C) Boundary Value Analysis 14-17

 14.7.2(D) Orthogonal Array Testing..... 14-17

 14.7.3 Differentiation between White-box and Black-box Testing..... 14-18

14.8 Verification and Validation..... 14-19

 14.8.1 Difference between Verification and Validation 14-20

14.9 Testing Strategies..... 14-20

14.9.1	Unit Testing.....	14-21
14.9.2	Integration Testing.....	14-22
14.9.3	Validation Testing.....	14-24
14.9.3(A)	Validation Test Criteria.....	14-24
14.9.3(B)	Configuration Review.....	14-25
14.9.3(C)	Acceptance Testing.....	14-25
14.9.3(D)	Alpha and Beta Testing.....	14-25
14.9.4	System Testing.....	14-26
14.9.4(A)	Recovery Testing.....	14-27
14.9.4(B)	Security Testing.....	14-27
14.9.4(C)	Stress Testing.....	14-28
14.10	Defect Management.....	14-28
14.10.1	Defect Management Process.....	14-28
14.10.2	Defect Removal Efficiency.....	14-29
14.11	Defect Life Cycle.....	14-29
14.12	Bug Reporting.....	14-31
14.12.1	Debugging.....	14-31
14.12.2	Psychological Considerations.....	14-32
14.12.3	Debugging Approaches.....	14-32

UNIT VI

Chapter 15 : Recent Trends in Software Engineering **15-1 to 15-27**

SCM, Risk Management, Technology evolution, process trends, collaborative development, software reuse, test-driven development, global software development challenges, CASE - taxonomy, tool-kits, workbenches, environments, components of CASE, categories (upper, lower and integrated CASE tools), Introduction to agile tools Jira, Kanban.

15.1	Software Configuration Management.....	15-1
15.1.1	SCM Basics.....	15-2
15.1.2	SCM Repository.....	15-3
15.1.3	SCM Features.....	15-5
12.1.4	SCM Process.....	15-6
15.1.5	Importance of SCM.....	15-10
15.2	Risk Management.....	15-10
15.2.1	Reasons for Project Delay.....	15-11
15.3	Technology Evolution.....	15-12

15.4	Process Trends	15-12
15.4.1	Model-driven development.....	15-13
15.4.2	Test-driven development.....	15-14
15.4.3	Challenges of global software development.....	15-15
15.4.4	Business drivers and global delivery challenges.....	15-15
15.5	Collaborative Development	15-15
15.6	Software Reuse	15-16
15.6.1	Advantages of software reuse.....	15-16
15.6.2	Problem in software reuse	15-17
15.7	CASE (Computer-Aided Software Engineering)	15-17
15.7.1	CASE tools.....	15-17
15.7.2	CASE - taxonomy.....	15-18
15.7.2(A)	Workbenches	15-18
15.7.2(B)	Tool-kits	15-19
15.7.2(C)	Environments.....	15-19
15.7.2(D)	Components of CASE.....	15-19
15.7.2(E)	Categories	15-20
15.8	Introduction to Agile Tools	15-20
15.8.1	JIRA.....	15-20
15.8.2	Kanban	15-23
➤	Model Question Paper (End Sem.)	M-1 to M-2
➤	Multiple Choice Questions	M-1 to M-20