Software Requirement Specifications

Unit: 2

Subject Name

Software Engineering

Course Details

B.Tech. 6th Sem

BISWARUP DUTTA

CSE

Content

- Course Objective
- Course Outcomes
- CO-PO Mapping
- CO-PSO Mapping
- Syllabus
- Prerequisite
- Requirement Engineering Process
- Elicitation, Analysis, Documentation,
- Review and Management of User Needs,
- Feasibility Study, Information Modelling,
- Data Flow Diagrams,
- Entity Relationship Diagrams

Content

- Decision Tables,
- IEEE Standards for SRS.
- Software Quality Assurance :(SQA):
- ISO 9000 Models,
- SEI-CMM Model
- Video Links
- Daily Quiz
- Weekly Assignment
- MCQ
- Old Question Papers
- Expected Questions for University Exam
- Summary
- References

Syllabus

Unit	TOPIC
I	Introduction: Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.
II	Software Requirement Specifications (SRS): Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modelling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.
III	Software Design: Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead’s Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

Syllabus

Unit	TOPIC
IV	Software Testing: Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, TopDown and BottomUp Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.
V	Software Maintenance and Software Project management: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

Course Objective(unit-2)

- An understanding of software requirements and SRS document.
- An understanding of implementation issues such as software Quality Frameworks, ISO 9000 Models, and SEI-CMM Model.

Objective of Topics

TOPIC	Objective
Requirement Engineering Process	To Understand the Requirement Engineering Process
Information Modelling	To draw the UFD, DFD diagrams
IEEE Standards for SRS	To develop the IEEE standard SRS document
Software Quality Assurance (SQA)	To Study the Software Quality Assurance
ISO 9000 Models SEI-CMM Model	To study the different standards for software development

Course Outcome

At the end of the Course, the student will be able

Course Outcomes (CO)		Bloom's Knowledge Level (KL)
NCS601.1	Explain various software characteristics and analyze different software Development Models.	K1, K2
NCS601.2	Demonstrate the contents of a SRS and apply basic software quality assurance practices to ensure that design, development meet or exceed applicable standards.	K1, K2
NCS601.3	Compare and contrast various methods for software design	K2, K3
NCS601.4	Formulate testing strategy for software systems, employ techniques such as unit testing, Test driven development and functional testing.	K3
NCS601.5	Manage software development process independently as well as in teams and make use of Various software management tools for development, maintenance and analysis.	K3

CO-PO Mapping

CO-PO Correlation Matrices

Correlation levels are taken 1, 2 and 3 as defined below:

1: Slight (Low) **2:** Moderate (Medium) **3:** Substantial (High)

Software Engineering **(Code: KCS-601)**									**Year of Study: 2020-21**			
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
C601.1	2	3	3	3	2	-	-	-	-	-	3	3
C601.2	3	3	3	3	3	-	-	-	-	-	2	3
C601.3	3	2	3	2	2	-	-	-	-	-	3	3
C601.4	2	2	2	2	3	3	-	3	3	-	3	3
C601.5	2	2	3	2	3	3	-	3	-	3	3	3

CO-PO and PSO Mapping

Program Specific Outcomes and Course Outcomes Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	-	3
CO2	3	3	2	3
CO3	3	3	-	3
CO4	3	3	-	3
CO5	3	3	-	3

*3= High *2= Medium *1=Low

Topic mapping with CO

TOPIC	CO
Requirement Engineering Process	CO2
Information Modelling	CO2
IEEE Standards for SRS	CO2
Software Quality Assurance (SQA)	CO2
ISO 9000 Models SEI-CMM Model	CO2

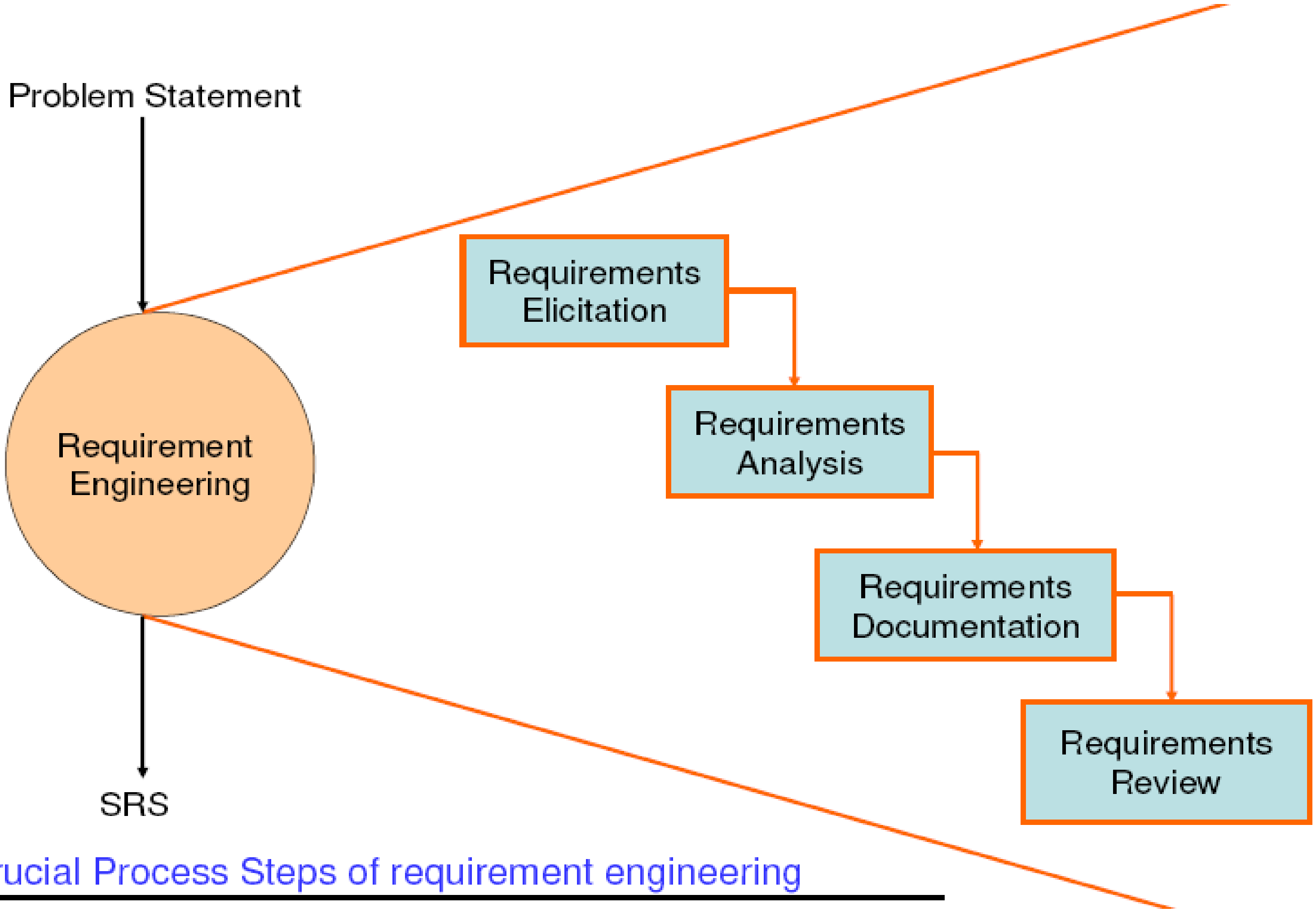
Prerequisite and Recap

- Basic Programming Skills
- Innovative Thinking.
- Enthusiasm to learn Management concepts.

Requirement Engineering (CO2)

- It is the disciplined application of proven principles, methods, tools, and notations to describe a proposed system's intended behavior and its associated constraints.
- It describe the "What" of a system, not the "How".
- It's input is problem statement prepared by the customer.
- It produces one large document(SRS),written in natural language, contain the description of what the system will do with out describing how it will do.
- It's process consists four steps:
  1. Requirement Elicitation.
  2. Requirement Analysis.
  3. Requirement Documentation.
  4. Requirement Review.

Requirement Engineering (CO2)



Crucial Process Steps of requirement engineering

Case Study

A University wish to develop a software system for the student result management of its M.Tech. Programe. A problem statement is to be prepared for the software development company. The problem statement may give an overview of the existing system and broad expectations from the new software system.

Problem statement(prepared by Exam. Division of Univ.)

- Univ. conduct 4-semeste M.Tech. program. student are offered four theory and two practical paper during I,II and IIIrd semester.
- In IV sem. Students have to give a seminar and submit a dissertation on topic area of their interest.
- Evaluation of each theory subject is done out of 150 marks . 100 for univ. conduct exam and 50 for sessional exam, attendance and student assignment.
- Evaluation of practical exam is done out of 50. 25 for univ. exam and 25 for internal in which student prepared lab record, viva, attendance.

Case Study

- Marks of IV sem. Dissertation is 450. 200 if for internal and 250 for external
- If total marks in each subject of a student is 50 % then Student is considered pass other wise Student is failed .
- At any time the latest information about subjects being offered in various semesters and their credit points can be obtained from univ. website.

- It is required to develop a system that manage information about subject offered in various sem. Students enrolled in various sem. Marks obtained by students in different sem.
- The system should also have the ability to generate printable mark-sheets for each student. Semester-wise detailed mark-lists and student performance reports also need to be generated.

Requirement Engineering Process (CO2)

1. Requirement Elicitation
   - It is known as requirement gathering.
   - Req. is identified with the help of customer and existing system process.
2. Requirement Analysis
   - it is start with req. elicitation.
   - It perform to identify inconsistencies, defects, etc.
3. Requirement Documentation
   - It is the end product of 1 and 2. known as SRS.
   - Doc. Provides the foundation of s/w design.
   - SRS may act as a contract between developer and customer.
4. Requirement Review
   - It is carried out to improve the quality of SRS.
   - It may also be called as requirement verification.

Type of Requirements

Stakeholder : Anyone who should have some direct or indirect influence on the system requirements. i.e. end user, developer, tester, coder etc.

- Known Requirements :
  - something a stakeholder believes to be implemented.
- Unknown Requirement :
  - forgotten by the stakeholder because they are not needed right now or needed only by another stakeholder.
- Undreamed Requirement :
  - stakeholder may not be able to think of new requirements due to limited domain knowledge.

Type of Requirement

Known, unknown, undreamed requirements may be functional or non-functional.

- Functional Requirements
  - it is related to the expectations from the intended s/w.
  - It describe what the s/w has to do.
  - Sometimes it may also specify what the s/w should not do.
- Non-Functional Requirements
  - It is mostly quality requirements that stipulate how well the s/w does what it has to do.
  - For **user** includes specification of performance, availability, usability and flexibility.
  - For **developers** are maintainability, portability and testability.

User and System Requirements

- User Requirements
  - User requirement are written for the users and include functional and non functional requirement.

- System Requirements
  - System requirement are derived from user requirement.
  - The user system requirements are the parts of software requirement and specification (SRS) document.

Requirement Elicitation (CO2)

- It is most important goal of RE to find out what users really need.
- It is ability that helps to understand the problem to be solved.
- It is conduct by asking que., writing down answers, asking other que., etc.
- Developers and users have different mind set, expertise and vocabularies so there are chances of conflicts that may led to inconsistencies, misunderstanding and omission of requirements .
- It requires the collaboration of several groups of participants who have different background.

Requirement Elicitation Methods

1. Interviews.
2. Brainstorming Sessions.
3. Facilitated Application Specification Technique(FAST)
4. Quality Function Deployment.
5. The use Case Approach.

Interviews

After receiving problem statement from customer, the first step to arrange a meeting between customer and specialized developer(requirement engineer)

Interviews are of two types

I. Open ended: no preset agenda.

II. Structured: a proper questionnaire is designed for interview

Brainstorming Sessions

- It is a group discussion technique that may lead to a lot of new ideas quickly and help to promote creative thinking.
- This technique may carried out with specialized groups like actual users, middle level managers etc, or with total stakeholders.
- To handle any undesirable situations(conflicts) a highly trained facilitator may be required.
- White boards, overhead transparencies or a computer projection system can be used to make it visible to every participant.
- After the session a detailed report will be prepared as facilitator will reviews the report.
- Finally, a document will be prepared which will have list requirements and their priority, if possible.

Facilitated Application Specification Technique(FAST)

- Similar to brainstorming sessions.
- Team oriented approach
- Creation of joint team of customers and developers.

Guidelines

- Arrange a meeting at a neutral site.
- Establish rules for preparation and participation.
- prepare Informal agenda to encourage free flow of ideas.
- Appoint a facilitator to control the meeting.
- Prepare definition mechanism board, worksheets, wall stickier.
- Participants should not criticize or debate.

FAST session Preparations

Each attendee is asked to make a list of objects that are:

- 1. Part of environment that surrounds the system.
- 2. Produced by the system.
- 3. Used by the system.
- A. List of constraints
- B. Functions
- C. Performance criteria

FAST session Preparations

Activities of FAST session

- Every participant presents his/her list of objects, services, constraints and performance for discussion.
- Combine list for each topic by elimination redundant entries and adding new ideas.
- By Discussion, consensus lists are finalized by the facilitator.
- Sub teams develop mini specifications for one or more entries.
- Presentations of mini-specifications, new objects, services, constraints, performance requirements are add to original list.
- Each attendee prepare a list of Validation criteria, a consensus list of validation criteria is then created.
- A sub team write the complete draft specifications using all inputs from FAST meeting.

Quality Function Deployment

- Focus on what is valuable to the customer and deploy these values through the SE process.
- Three types of requirements are identified:
  - Normal requirements: objective and goals of proposed s/w which discussed with the customer.
  - Expected requirements: implicitly req. in s/w that customer not state them explicitly. Eg. Authentication, warning features
  - Exciting requirements: feature go beyond the customer, if they present prove very satisfying. Eg. Unauthorized access notification
- QFD steps:
  - 1. Identify stakeholders
  - 2. List out requirements
  - 3. Degree of importance to each requirement

QFD...

  - 5 Points : V. Important
  - 4 Points : Important
  - 3 Points : Not Important but nice to have
  - 2 Points : Not important
  - 1 Points : Unrealistic, required further, exploration
- Final list of requirements categorize like:
  - (i) It is possible to achieve
  - (ii) It should be deferred & Why
  - (iii) It is impossible and should be dropped from consideration
- First Category requirements will be implemented as per priority assigned with every requirement.

Use Case Approach

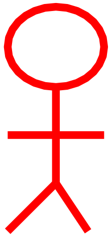
Components of Use Case approach

- Use Cases : it is initiated by a user with a particular goal in mind, and completes successfully when that goal is satisfied.
  - Use Cases are focus on “what” the system is , not “how” the system will be designed.
  - It describes the sequence of interactions between actors and the system necessary to deliver the services that satisfies the goal.
- Actor: An actor or external agent, lies outside the system model, but interacts with it in some way. Actor are Person, machine, information System, etc
- Association : it represent the relationship between actor and useca.se

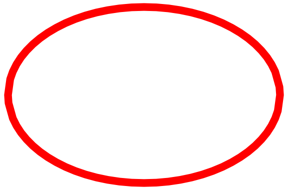
Use case Diagram

-- represents what happens when actor interacts with a system.

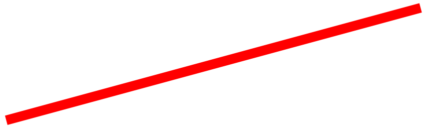
-- captures functional aspect of the system.



Actor



Use Case



Relationship between actors and use case and/or between the use cases.

-- Actors appear outside the rectangle.

--Use cases within rectangle providing functionality.

--Relationship association is a solid line between actor & use cases.

Use Case Template

Introduction: Describe a quick background of the use case.

Actors : List the actors that interact and participate in the use cases.

Pre Conditions : Pre conditions that need to be satisfied for the use case to perform.

Post Conditions : Define the different states in which we expect the system to be in, after the use case executes.

Use Case Template

Flow of events

Basic Flow : List the primary events that will occur when this use case is executed.

Alternative Flows: Any Subsidiary events that can occur in the use case should be separately listed. List each such event as an alternative flow. A use case can have many alternative flows as required.

Special Requirements : Business rules will be used for writing test cases. Both success and failures scenarios should be described.

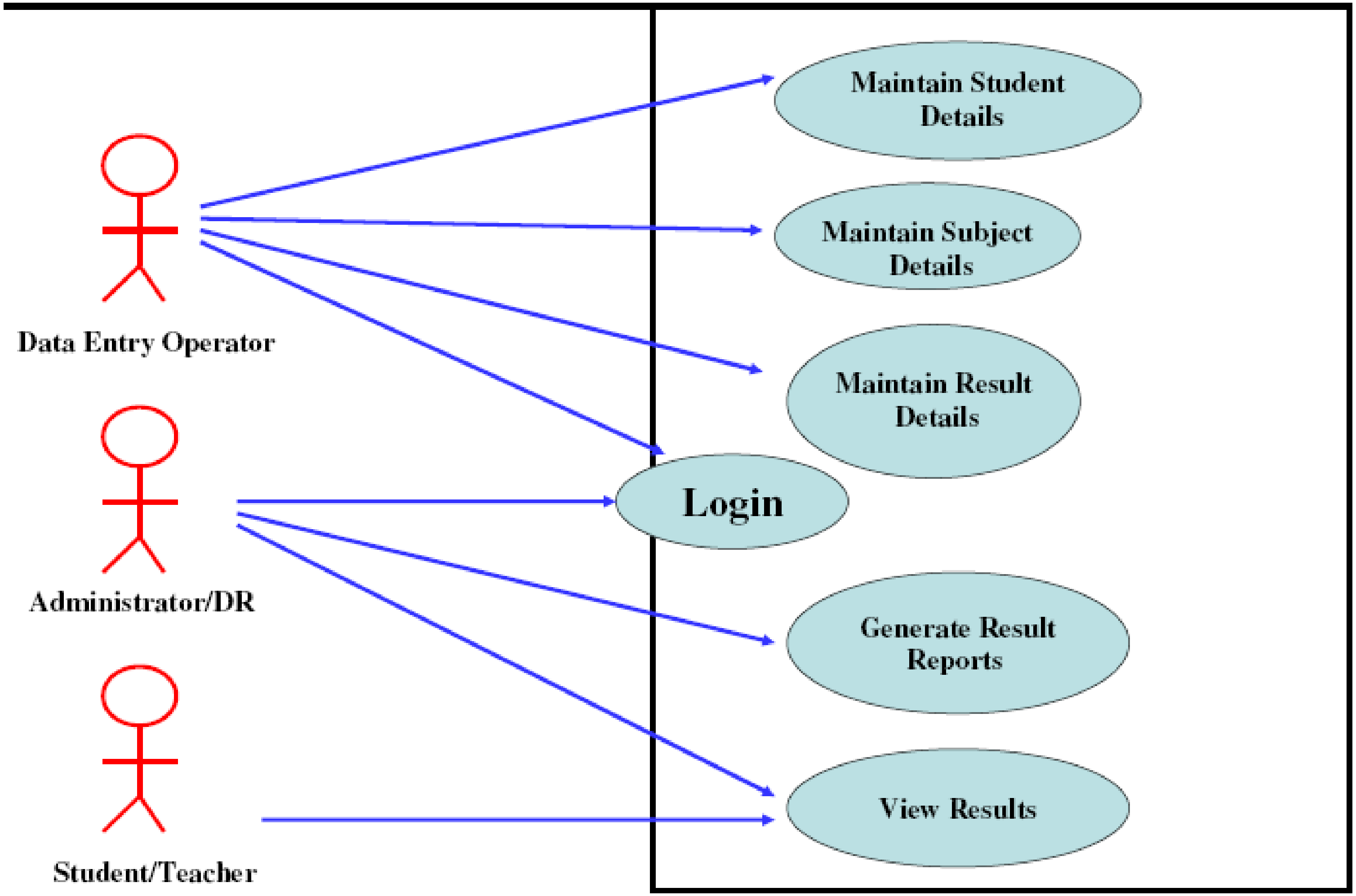
Use Case relationships : For Complex systems Listing the relationships between use cases also provides a mechanism for traceability

Use Case Template

1. Identify all users.
2. Create a user profile for each category of users including all roles of the users play that are relevant to the system.
3. Create a use case for each goal, following the use case template maintain the abstraction throughout the use case.
4. Structure the use case.
5. Review and validate with users.

Cont..

Use case diagram for Result Management System



Login(Template)

Login

- 1.1 Introduction : This use case describes how a user logs into the Result Management System.
- 1.2 Actors : (i) Data Entry Operator

  (ii) Administrator/Deputy Registrar
- 1.3 Pre Conditions : None
- 1.4 Post Conditions :

  If the use case is successful, the actor is logged into the system. If not, the system state is unchanged.

Login(Template)

1.5 Basic Flow : This use case starts when the actor wishes to login to the Result Management system.

- (i) System requests that the actor enter his/her name and password.
- (ii) The actor enters his/her name & password.
- (iii) System validates name & password, and if finds correct allow the actor to logs into the system.

Login(Template)

- 1.6 Alternate Flows
  - 1.6.1 Invalid name & password

  If in the basic flow, the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the basic flow or cancel the login, at that point, the use case ends.

1.7 Special Requirements: None

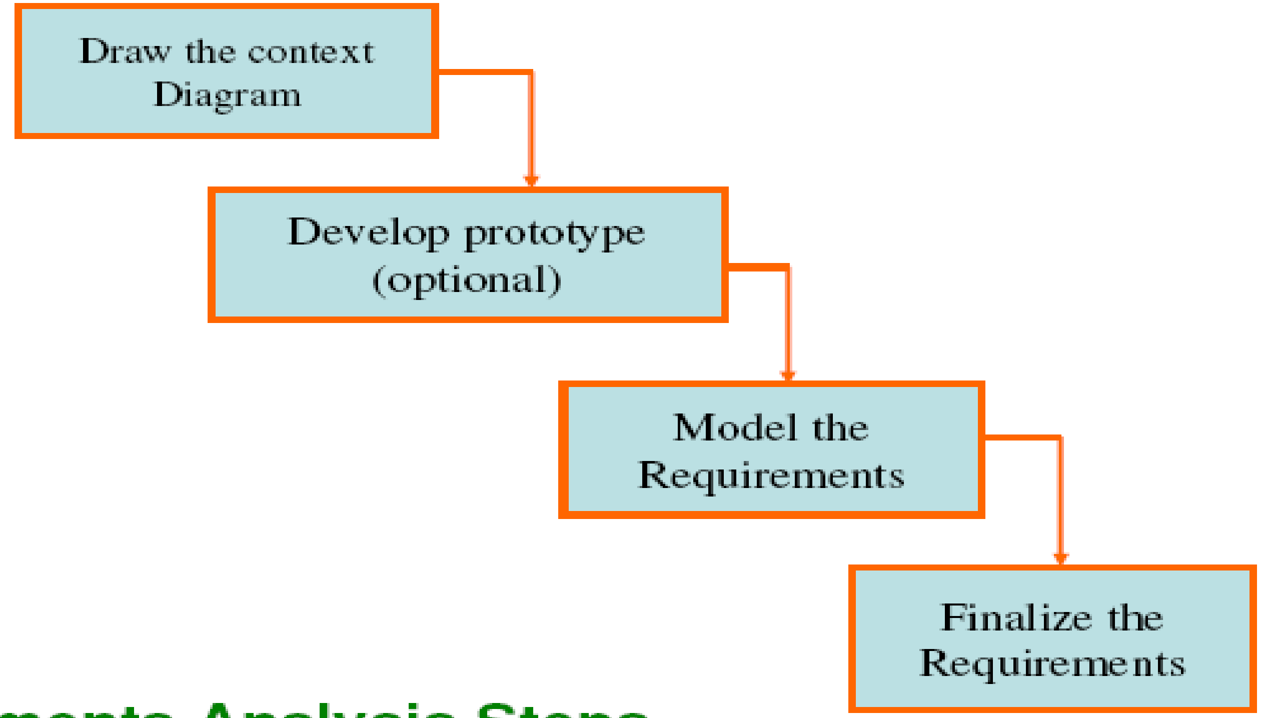
1.8 Use case Relationships: None

Requirement Analysis (CO2)

- We analyze, refine and scrutinize gathered requirements to make consistent & unambiguous requirements.
- Activity reviews all the requirement ands may Provide graphical view of the entire system.
- May also interact with customer to resolve their confusion and find priority of requirements.

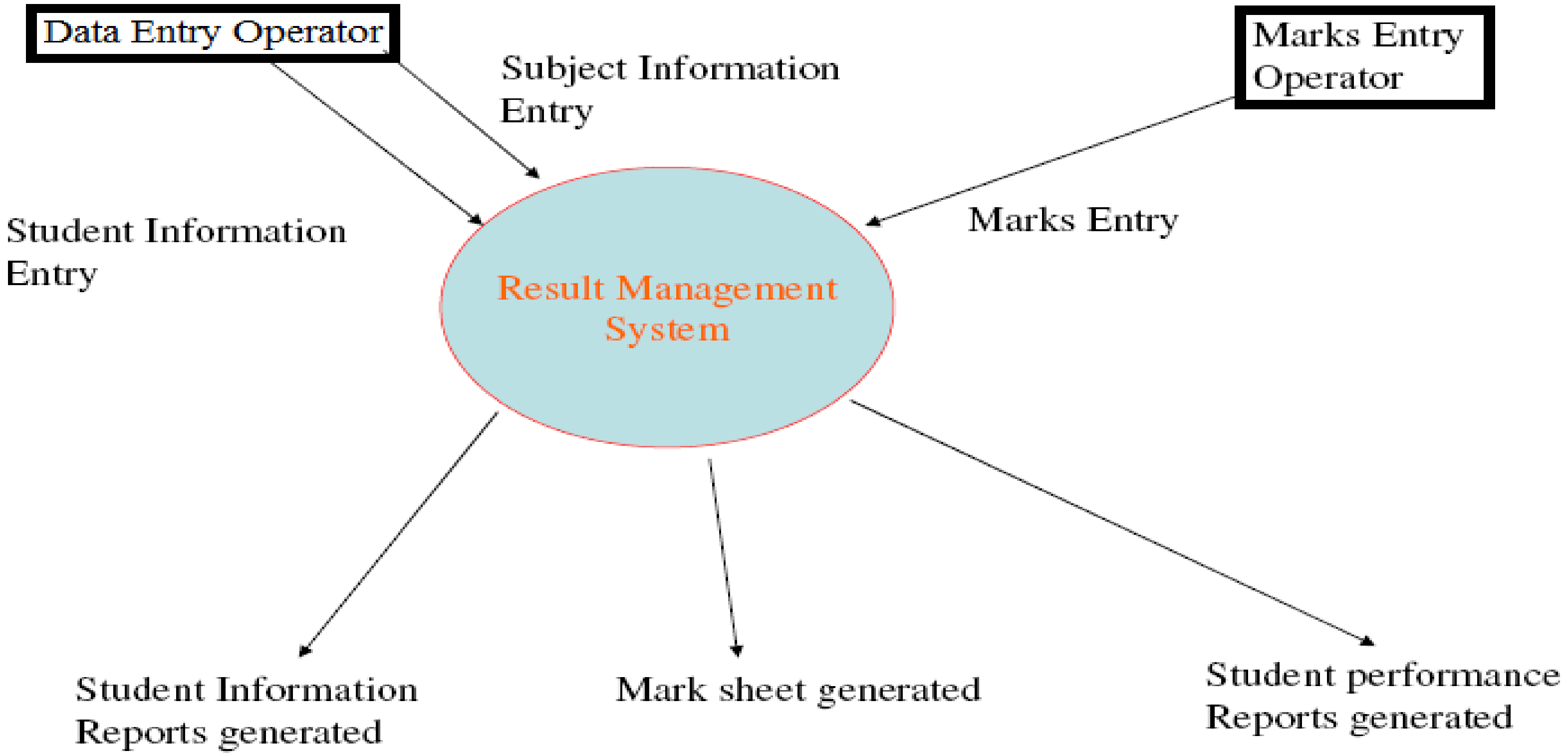
Steps of Requirement Analysis

Steps



Requirements Analysis Steps

Context Diagram (CO2)


Data Entry Operator
Subject Information Entry
Marks Entry Operator
Student Information Entry
Marks Entry
Result Management System
Student Information Reports generated
Mark sheet generated
Student performance Reports generated

Development of Prototype(Optional)

- It is effective way to find out what the customer really want.
- We take customer feedback to continuously modify the prototype until he/she is satisfied.
- It should be built quickly and at a relatively low cost.
- Due to its limitations and would not be acceptable in final system it is an optional activity.
- Many organization are developing prototypes for better understanding before the finalization of SRS.

Model the Requirement

- This process consists of various graphical representation of functions, data entities, external entities and relationship between them.
- Graphical view help us to find incorrect, missing, and inconsistent requirements.
- Such models includes
  - Data flow diagram
  - Entity Relationship Diagram(ERD)
  - Data Dictionary
  - Decision Table, etc.

Finalize the Requirements

- After modeling the requirements, we will have better understanding of the system behavior.

- Inconsistencies and ambiguities have been identified.

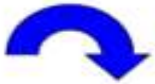
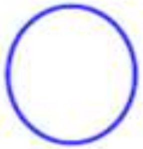
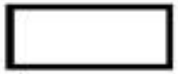
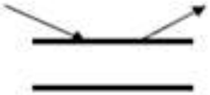
- Now we finalize the analyzed requirements and next step is to document these requirements in a prescribed format

Data Flow Diagrams(CO2)

- Also known as data flow graph or bubble chart.
- It is show the flow the data through a system.
- It represent a systems in terms of input data to the system, various processing carried out on those data, and the output data generated by the system.
- System may be an organization, a set of procedures, a computer H/W system, a s/w system, etc.

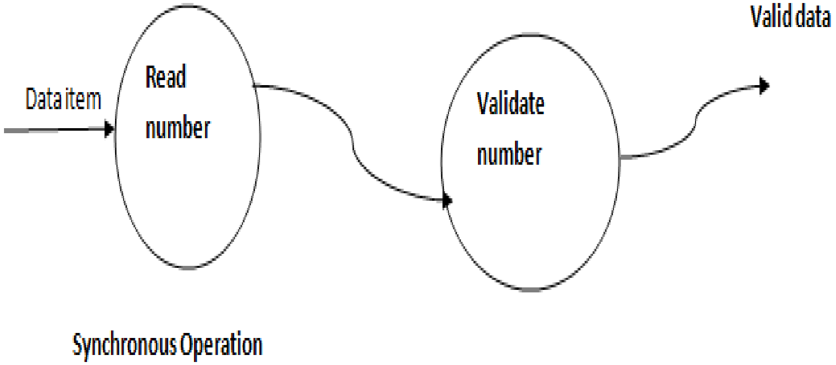
- In DFD
  - All names should be unique.
  - It is not a flow chart. Arrow represent flowing of data without any order.
  - Suppress logical decisions.
  - Defer error conditions & handling until the end of the analysis.

DFD

Symbol	Name	Function
	Data Flow	Connect process
	Process	Perform some transformation of its input data to yield output data.
	Source or sink	A source of system inputs or sink of system outputs
	Data Store	A repository of data, the arrowhead indicate net input and net outputs to store

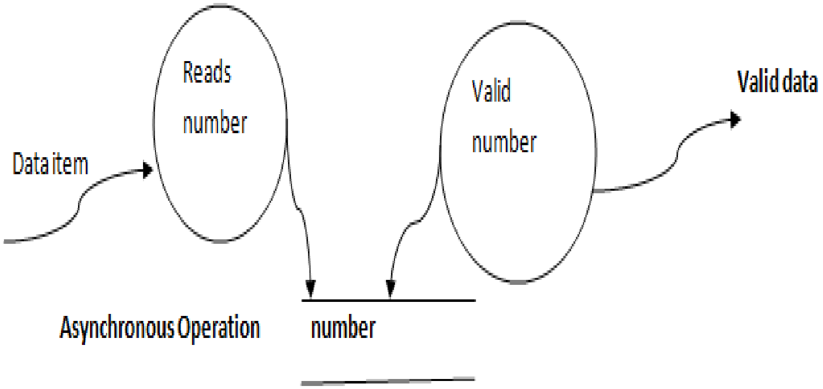
Operation in DFD

- Synchronous : if two bubble are directly connected by a data flow arrow i.e. they are operate at the same speed.



Synchronous Operation

- Asynchronous: if two bubble are connected through a data store then speed of operation of the bubble are independent.

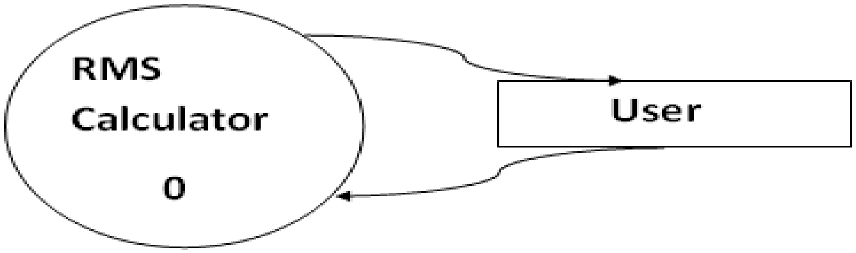


Asynchronous Operation

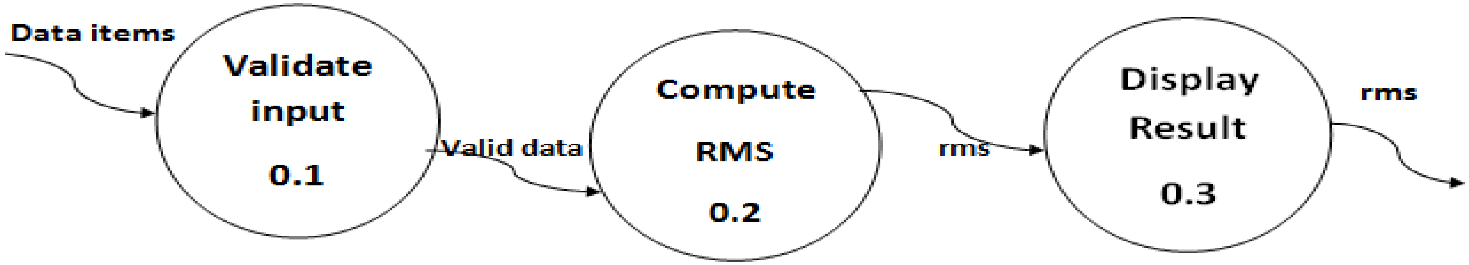
Developing DFD model of a system

- Step1: Construction of context diagram. It a top level DFD , usually called 0 level DFD.(A level 0 DFD is called fundamental system model or context model represents entire software element as a single bubble with input and output data indicating by incoming & outgoing arrows.)
- Step2: Construction of level1 DFD. Expend the process of context diagram.
- Step3: Construction of lower level DFD. Decompose the process of level 1 DFD recursively.

A case study of Root Mean Square Calculating System

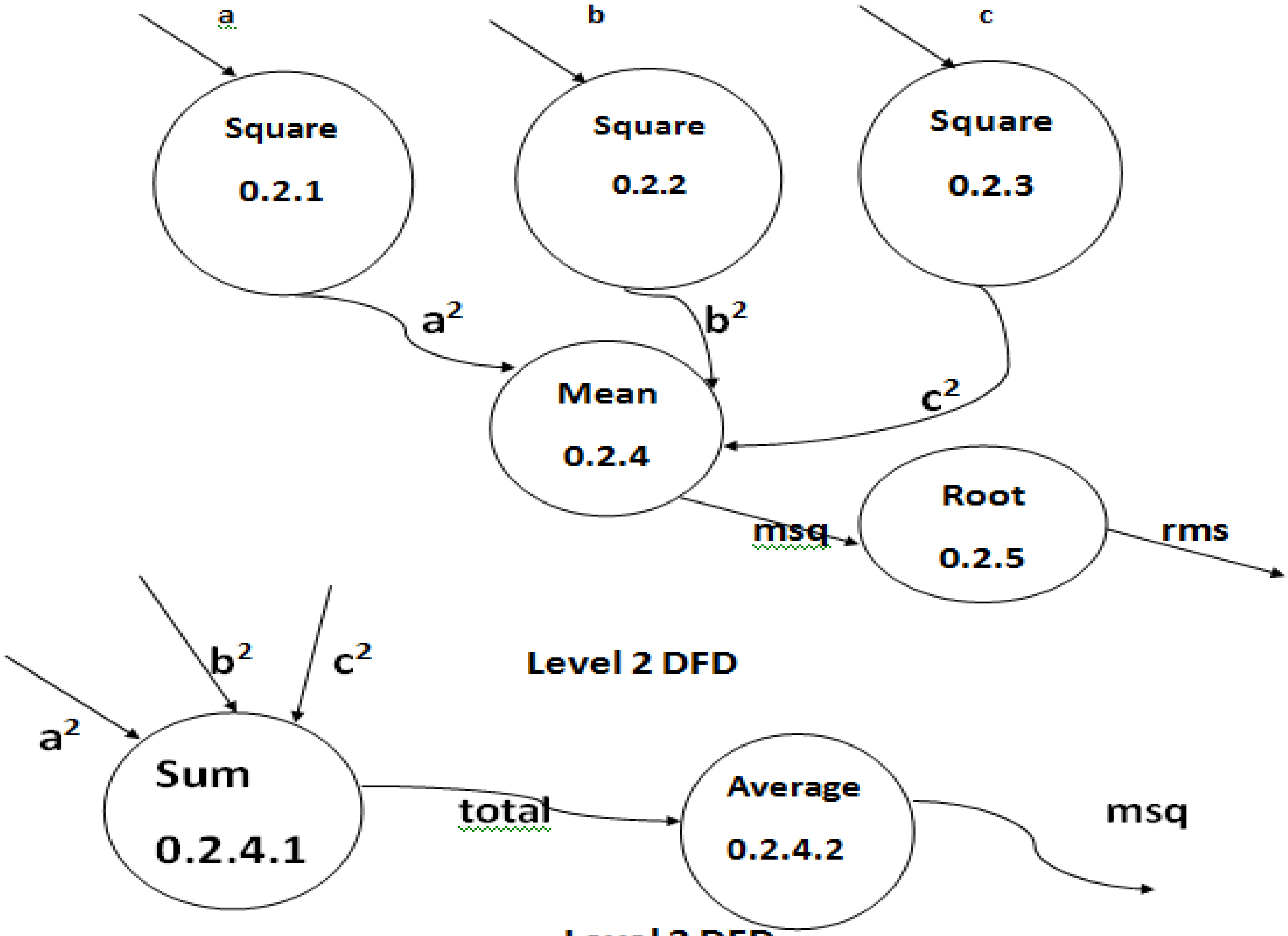


Context Diagram (0 Level DFD)



Level 1 DFD

2 and 3 Level DFD



Level 2 DFD

Level 3 DFD

Data Dictionary

- It is the repositories to store information about all data items defined in DFD.
- It is an organized collection of all the data elements of the system and their brief description
    - Includes :
        - Name of data item
        - Aliases (other names for items)
        - Description/Purpose
        - Related data items
        - Range of values
        - Data flows
        - Data structure definition

Advantage

Advantage:

- Define data items unambiguously.
- Valuable reference in any organization because it provides documentation.
- It is a good tool for different stakeholder to understand the requirements and designs.
- It store the information of all data items that can link to all phases of SDLC.
- It is an important step in building a database. Most of DBMS have a data dictionary as a standard format.

ER Diagram (CO2)

It is a detailed logical representation of data for an organization and uses three main constructs.

- Entities
- Fundamental thing about which data may be maintained. Each entity has its own identity.
- Entity Type is the description of all entities to which a common definition and common relationships and attributes apply.

ER Diagram

It is a detailed logical representation of data for an organization and uses three main constructs.

- Entities
- Fundamental thing about which data may be maintained. Each entity has its own identity.
- Entity Type is the description of all entities to which a common definition and common relationships and attributes apply.

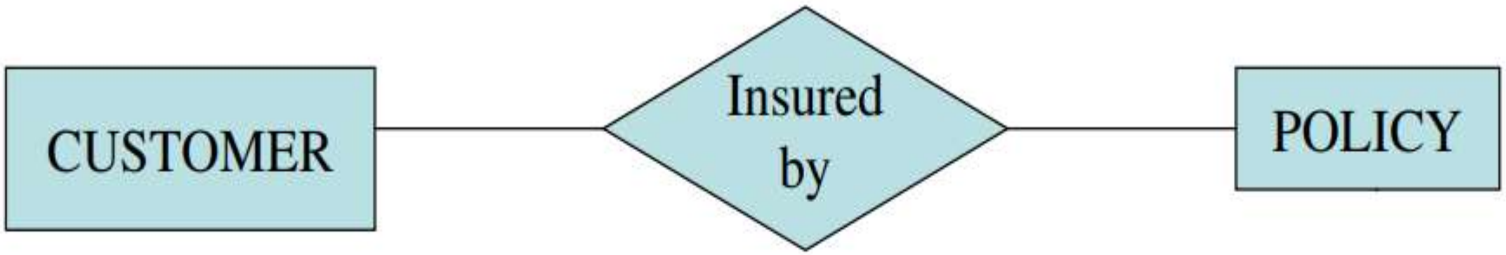
POLICY

CUSTOMER

ER Diagram

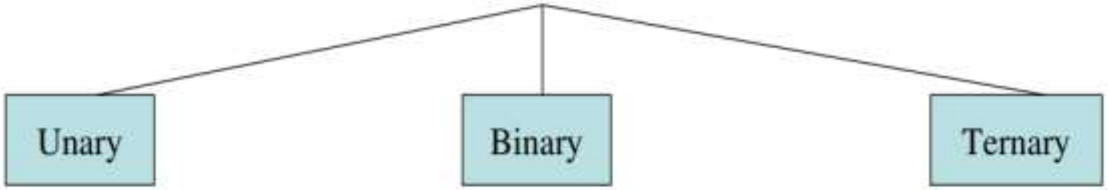
Relationships

- A relationship is a reason for associating two entity types. Binary relationships involve two entity types
- A CUSTOMER is insured by a POLICY.
- Relationships are represented by diamond notation in a ER diagram



ER Diagram

Degree of relationship

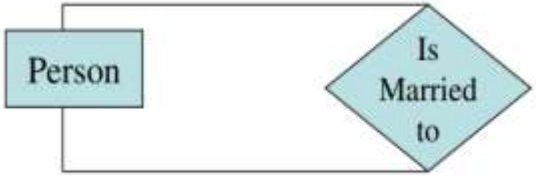


Unary

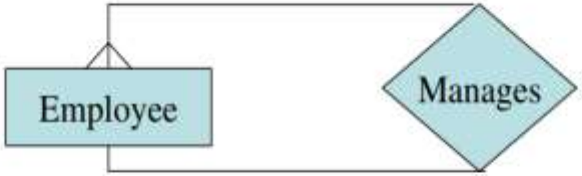
Binary

Ternary

Unary relationship



One to One



One to many

Decision Table(CO2)

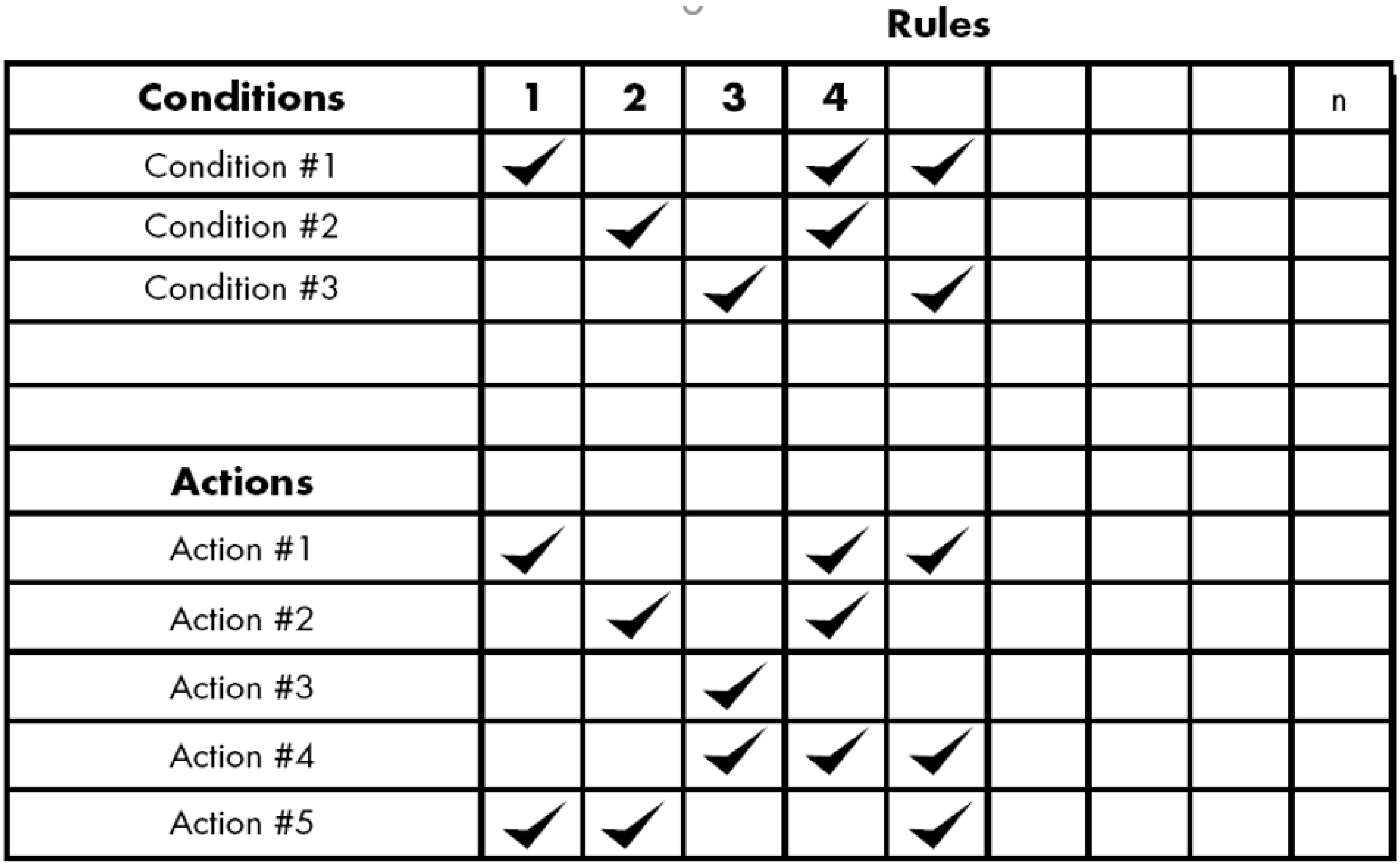
- It is a tabular form that presents a set of conditions and their corresponding actions.
- There four portion of decision table

Condition	**Condition Alternatives**
Actions	Actions Entries

Decision Table

- 2-Dim. Matrix:
    - Row1 : for each possible actions.
    - Row2: for each relevant conditions.
    - One column for each combination of condition states.
  - Upper rows specify the variables or conditions to be evaluated and lower rows specify the corresponding actions to be taken when an evaluation test is satisfied
  - A column in a decision table is called a rule. It implies if a condition is true then execute the corresponding action.

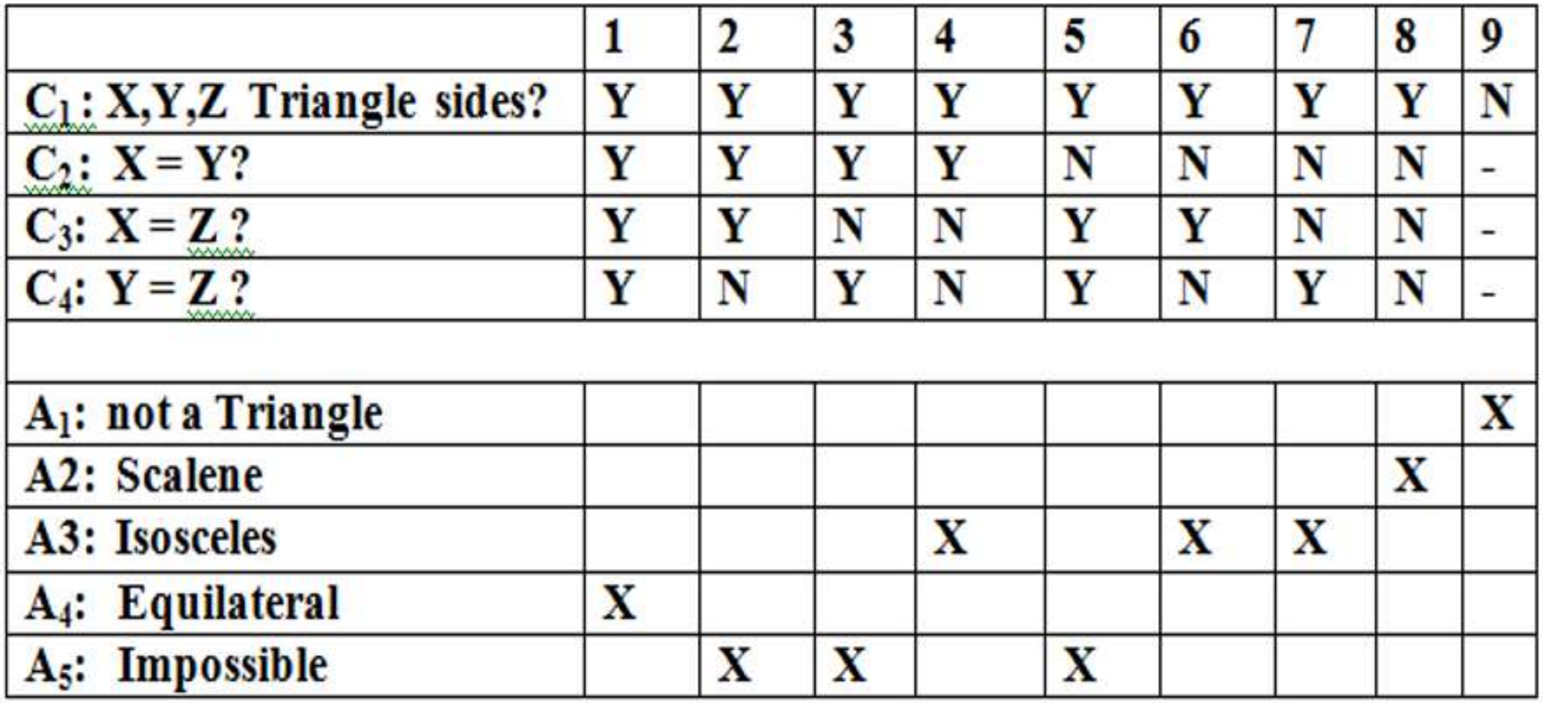
Decision Table



Rules

Conditions	1	2	3	4					n
Condition #1	✓			✓	✓				
Condition #2		✓		✓					
Condition #3			✓		✓				
Actions									
Action #1	✓			✓	✓				
Action #2		✓		✓					
Action #3			✓						
Action #4			✓	✓	✓				
Action #5	✓	✓			✓				

Decision Table for Triangle Problem



	1	2	3	4	5	6	7	8	9
C_1: X,Y,Z Triangle sides?	Y	Y	Y	Y	Y	Y	Y	Y	N
C_2: X = Y?	Y	Y	Y	Y	N	N	N	N	-
C_3: X = Z ?	Y	Y	N	N	Y	Y	N	N	-
C_4: Y = Z ?	Y	N	Y	N	Y	N	Y	N	-
A_1: not a Triangle									X
A2: Scalene								X	
A3: Isosceles				X		X	X		
A_4: Equilateral	X								
A_5: Impossible		X	X		X				

Requirement Documentation (CO2)

- The two scenario create entirely different situation and establish entirely different purposes for the document.

- First case SRS is used to define the needs and expectations of the user. Second case ,SRS is written for different purpose and serve as a contract document between customer and developer.

SRS Document(CO2)

- Requirement document is called software requirement specification(SRS). The SRS is a specification for a particular software product program or set of program that perform certain functions in a specific environment. It serves a number of purposes depending on who is writing it. First the SRS could be written by the customer of a system. Second the SRS could be written developer of a system.

Characteristics of SRS

- **Correct:** The SRS should be made up the date when appropriate requirements are identified.
- **Unambiguous:** When the requirements are correctly understood then only it is possible to write unambiguous software.
- **Complete**: To make SRS complete, its hold be specified what a software designer wants to create software.
- **Consistent:** It should be consistent with reference to the functionalities identified.
- **Specific:** The requirements should be mentioned specifically.
- **Traceable:** What is the need for mentioned requirement? This should be correctly identified.

Organization of the SRS (CO2)

- IEEE has published guidelines and standards to organize an SRS.
- First two sections are same. The specific tailoring occurs in section-3.

1. Introduction

    1.1 Purpose

    1.2 Scope

    1.3 Definition, Acronyms and abbreviations

    1.4 References

    1.5 Overview

Organization of the SRS

2. The Overall Description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware Interfaces

2.1.4 Software Interfaces

2.1.5 Communication Interfaces

2.1.6 Memory Constraints

2.1.7 Operations

2.1.8 Site Adaptation Requirements

Organization of the SRS

2.2 Product Functions

2.3 User Characteristics

2.4 Constraints

2.5 Assumptions for dependencies

2.6 Apportioning of requirements

Organization of the SRS

3. Specific Requirements

3.1 External Interfaces

3.2 Functions

3.3 Performance requirements

3.4 Logical database requirements

3.5 Design Constraints

3.6 Software System attributes

3.7 Organization of specific requirements

3.8 Additional Comments.

Organization of the SRS

3. Specific Requirements

3.1 External Interfaces

3.2 Functions

3.3 Performance requirements

3.4 Logical database requirements

3.5 Design Constraints

3.6 Software System attributes

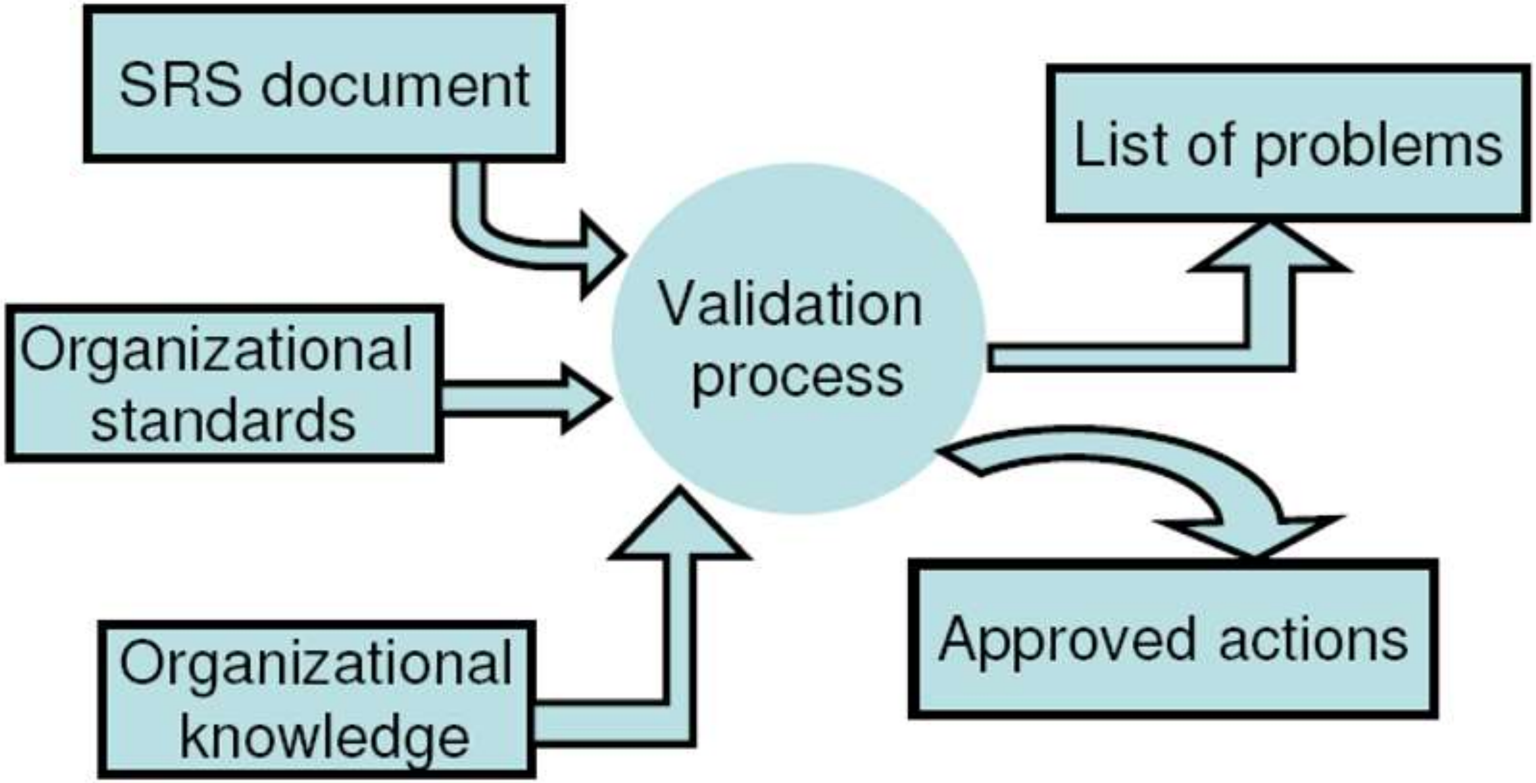
3.7 Organization of specific requirements

3.8 Additional Comments.

Requirement validation (CO2)

- After completion of SRS Check the document for
  - Completeness & consistency
  - Conformance to standards
  - Requirements conflicts
  - Technical errors
  - Ambiguous requirements
- Objective of req. validation is to certify the SRS doc., Is an acceptable doc. Of the system to be implemented.
- It find the error in doc. And improves the quality of s/w development process.
  - **Analysis**: work with row requirement as collected from various stakeholder
  - **Validation:** work with a final draft of the SRS doc. With negotiated and agreed requirements.

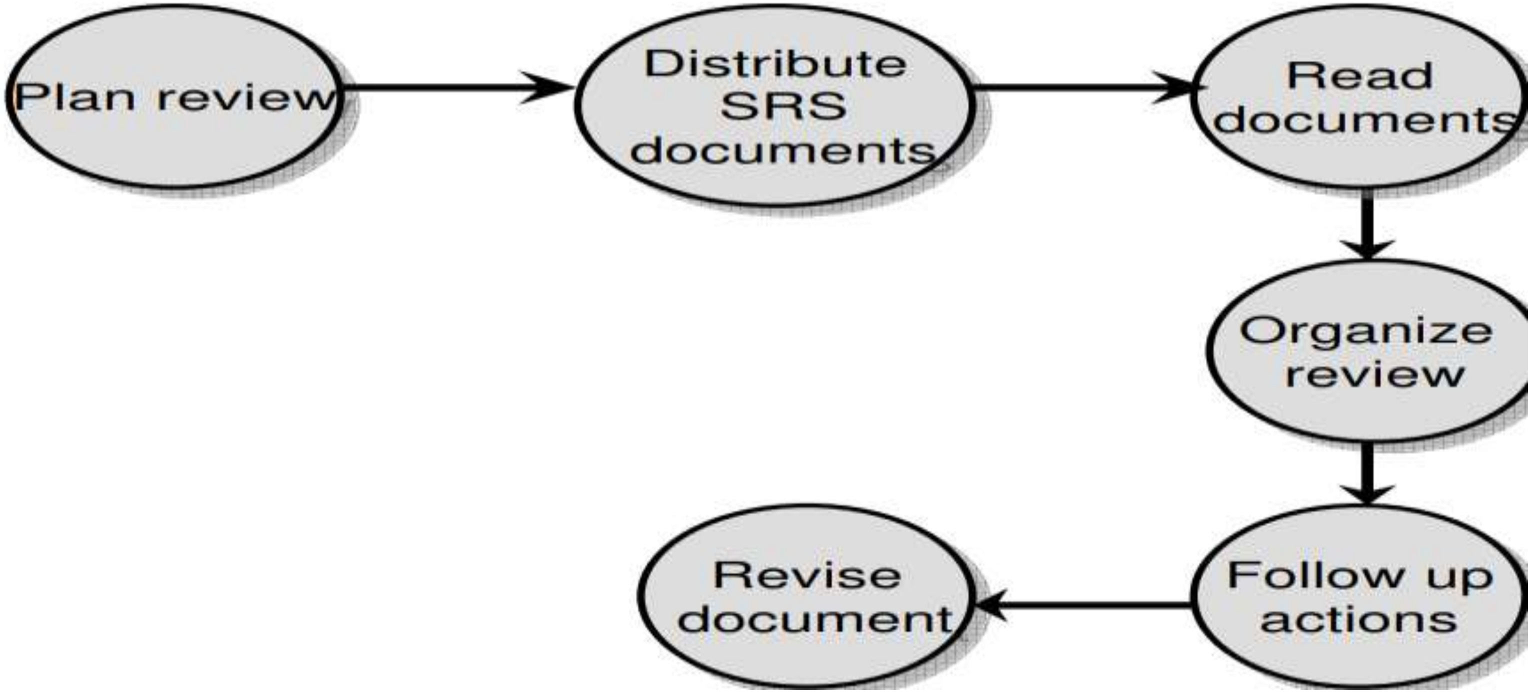
Requirement Validation



Requirement Review (CO2)

- Plan review: review team is selected, time and place is fixed for review meeting.
- Distributed SRS doc: each member should read doc. To find conflict, inconsistencies, deviations and other problems.
- Organize review meeting: each member present his/her view, problem are discussed and action on them are approved.
- Follow-up actions: chairperson of team checks that approved action have been carried out
- Revise SRS doc: SRS doc. Is revised to reflect the approved actions

Requirement Review Process



Management of User Need

- User Requirement Document(URD) :
- it is used in SE, That specify the req. the use expect from sw.
- After URD customer can not demand extra feature similarly developer cannot claim the product ready until it does not meet URD.
- Types of requirements:
  - Enduring requirements: They are core requirements & are related to main activity of the organization. Example: in library management. System issue/return of a book, cataloging etc.
  - Volatile requirements: likely to change during software development lifer cycle or after delivery of the product .

.

Management of User Need

- **Reason for change are:**
  - Change in environment.
  - Change in Technology.
  - Change in policies.
  - Change in customer’s expectations.

Requirements Management

- Requirements Management : Process of understanding and controlling changes to system requirements.
- Requirement change Mgmt. process can be applied in three Stages:
    - Problem Analysis and Change Specification : change request made for partial problem then change specification are analyzed in order to validate the req. change
    - Change Analysis and Costing: estimate cost of change then take decision to implement it or not.
    - Change Implementation: when decide to change in req. Req doc. Is re-written or re-organized.

Verification and Validation(CO2)

- The purpose of V & V is to confirm system specification and to meet the requirement of system customers.
- Verification: represent the set of activities that are carried out to confirm that the s/w correctly implemented the specific functionality.(are we building the product right?)
- It is the process of determining whether the output of one phase of software development conforms to that of its previous phase. Thus verification is concerned with phase containment of errors
- Validation: represent set of activities that ensure that the s/w that has been build is satisfying the customer requirements.(are we building the right product?)
- It is the process of determining whether a fully developed system conforms to its requirements specification. the aim of validation is that the final product be error free.

Software Quality Assurance(CO2)

- It is a set of activities that are designed to evaluate the process by which s/w developed and/or maintained.
- The aim of this process is to develop high quality s/w product.
- SQA objectives:
  - Quality mgmt. approach.
  - Measurement and reporting mechanism
  - To make Effective s/w engg. Technique.
  - Formal
  - technical reviews applied on through s/w process.
  - A multi tired testing strategy.
  - Control of s/w documentation and change make to it.

SQA Plan (CO2)

- It consists various tasks which carried out by two groups:
  - Group1: consist of s/w Engineers who do technical work.
  - Group2: it is SQA group. Responsible to prepare SQA plan, record keeping, analysis and reporting. It assist s/w team in achieving a high quality end product.
- SQA plan provides a road map for SQA.
- SQA plan identifies:
  - evaluations to be performed
  - audits and reviews to be performed
  - standards that are applicable to the project
  - procedures for error reporting and tracking
  - documents to be produced by the SQA group
  - amount of feedback provided to the software project team

SQA plan is a three stage process

- Stage 1: SQA team should write in detail activities related for s/w requirements.

- Stage2: team should analyze in detail the preparation of the development team for detailed build-up.

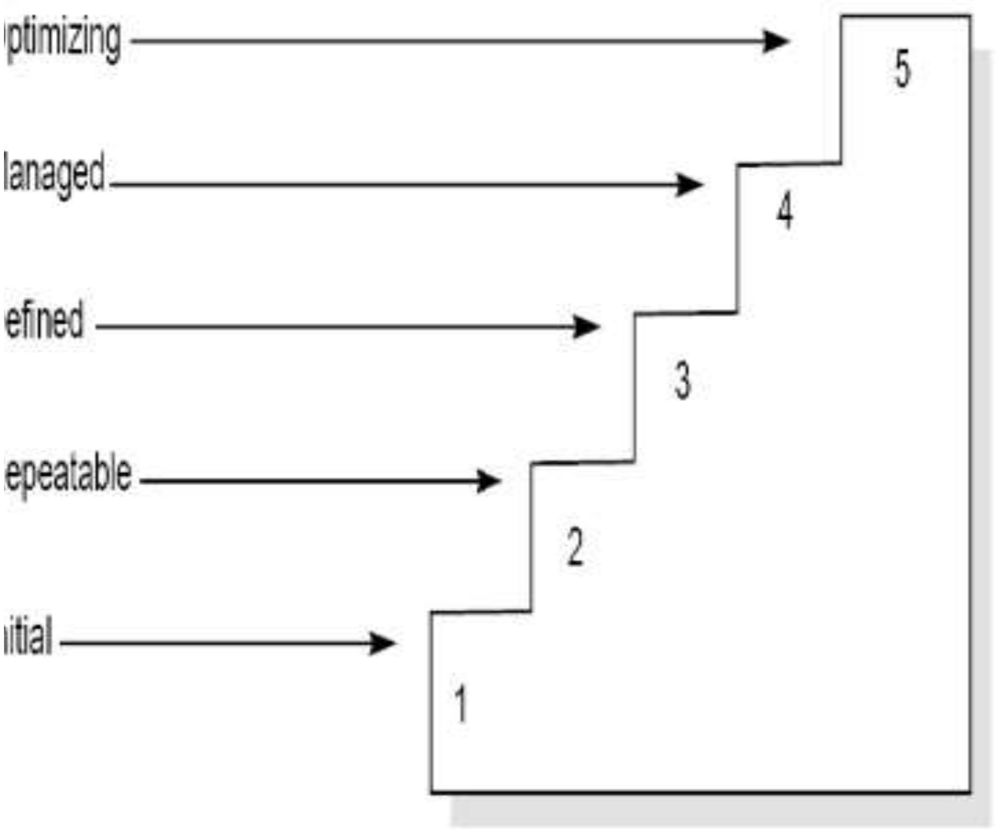
- Stage3: tackles the QA plan for detailed design and actual product.(it is longest among three phases)

Capability Maturity Model(CO2)

- It is not a SDLC model.
- It is a strategy for improving the software process, irrespective of the actual life cycle model used.
- It was developed by S/W Engg. Institute(SEI) of Carnegie-Mellon Univ. in 1996.
- It is an assessment that results in a five point grading scheme

Capability Maturity Model(CO2)

Maturity Level	Characterization
Initial	Adhoc Process
Repeatable	Basic Project Management
Defined	Process Definition
Managed	Process Measurement
Optimizing	Process Control



Maturity Levels

- Initial (Maturity Level 1)
  - The ad hoc software process development.
  - Very Few or no processes are defined and followed.
  - Different engineers follows their own process.
  - Success of project depends on individual effort.
  - Totally depends on current staff. When developer leave org., successor face grate difficult to understand the process.
  - No formal project management practices are followed.
  - It not possible to predict accurately time and cost of s/w product.
- Repeatable (Maturity Level 2)
  - Basic project management process are established to track cost, schedule, and functionality.
  - The necessary process discipline is in place to repeat earlier success on project with similar application but process is not documented.
  - In similar project success story of development can be repeated for another.

CMM…

Defined (Maturity Level 3)

- s/w process for both mgmt. and engg. Activities is documented, standardized into a standard s/w process of the organization.
- All project use an documented and approved version of the organization standard s/w process for developing and maintaining s/w.
- But the process and practices are not analyzed quantitatively.

Managed (Maturity Level 4)

- Both the s/w process and product are quantitatively understood and controlled.
- It focus on s/w metrics:
  - Product metrics: measure the char. Of the product being developed such as size, reliability, time complexity, understandability, etc.
  - Process metrics: it reflect the effectiveness of the process being used such as productivity, average no of defect found, average no of failure detected during testing per LOC etc.

CMM…

Optimizing (Maturity Level 5)

- Continuous process improvement Is enabled by quantitative feedback from the process and from introducing innovative ideas and technologies.
- Organization Is committed to continuous process improvement.

So best s/w Engg. & mgmt . Practices are used throughout the organization.

ISO 9000 (CO2)

- It is a consortium of 120 countries to formulate and bring up standardization and published its 9000 series of standard in 1987.
- It specify the guidelines for maintaining of quality system.
- It focused both operational aspect(process) and organizational aspects(responsibilities, reporting etc.).
- ISO-9000 series of standards is a set of document dealing with quality systems that can be used for quality assurance purposes.

ISO 9000 (CO2)

- **ISO-9000 Series**: it comprises following standard
  - **ISO 9001** : Quality System-Model for Quality Assurance in design, development, production, installation and servicing.
  - **ISO 9002** : Quality System-Model for Quality Assurance only in production and installation.
  - **ISO 9003 :** Quality System-Model for Quality Assurance in final inspection and testing.

Reasons that s/w industry get ISO certification

- It is a symbol of customer confidence.
- It highlights weakness and suggests correction measure for improvement.
- It makes process more focused, efficient and cost effective.
- It is a motivating factor for business organization.
- It helps to designing high quality repeatable s/w product.
- It facilitates the development of optimal processes and total quality measurement.
- It emphasizes the need of proper documents.

How to get ISO 9000 Certification

1. Application Stage: once org. decide ISO 9000 certification, it applies to registrar for registration.
2. Pre-Assessment: registrar make a rough assessment of org.
3. Doc. Review and audit: registrar reviews the doc. Submitted by the organization and makes suggestion for possible improvement.
4. Compliance Audit: registrar checks whether suggestion complied by organization or not.
5. Registration: after satisfaction registrar award ISO 9000.

- In india ISO 9000 certification is offered by STQC(Standardization Testing and Quality Control), IRQS(Indian Register Quality System) and BIS(Bureau Indian Standard)

Faculty Video Links, Youtube & NPTEL Video Links and Online Courses Details

- https://nptel.ac.in/courses/106/105/106105182/
- https://www.youtube.com/watch?v=crz9WmoUoKc
- https://www.youtube.com/watch?v=rKG7mgVFCTM
- https://www.youtube.com/watch?v=WjwEh15M5Rw

Daily Quiz

1) Which one is not a step of requirement engineering?

(a) Requirements elicitation (b) Requirements analysis

(c) Requirements design (d) Requirements documentation

2) Requirements elicitation means

(a) Gathering of requirements (b) Capturing of requirements

(c) Understanding of requirements **(d) All of the above**

3) SRS stands for

(a) Software requirements specification (b) System requirements specification

(c) Systematic requirements specifications (d) None of the above

4) SRS document is for

(a) "What" of a system? (b) How to design the system?

(c) Costing and scheduling of a system (d) System's requirement.

5) Requirements review process is carried out to

(a) Spend time in requirements gathering **(b) Improve the quality of SRS**

(c) Document the requirements (d) None of the above

Daily Quiz

6) Which one is not a type of requirements?

(a) Known requirements (b) Unknown requirements

(c) Undreamt requirements **(d) Complex requirements**

7) Which one is not a requirements elicitation technique?

(a) Interviews (b) The use case approach (c) FAST

(d) Data flow diagram.

8) Context diagram explains

(a) The overview of the system (b) The internal view of the system

(c) The entities of the system (d) None of the above

9) Outcome of requirements specification phase is

(a) Design Document **(b) SRS Document**

(c) Test Document (d) None of the above

10) IEEE standard for SRS is:

(a) IEEE Standard 837-1998 **(b) IEEE Standard 830-1998**

(c) IEEE Standard 832-1998 (d) IEEE Standard 839-1998

Weekly Assignment

1. What do you mean by formal and informal review?
2. What is SRS?
3. What is the purpose of requirement document?
4. What do you mean by use case diagram? Prepare a use case diagram for student result management system and explain it all use case using different parameter in its template.
5. What is requirement engineering process? Explain all the methods of requirement elicitation.
6. What do you mean by data flow diagram? Discuss higher level to low level DFD with suitable diagram.
7. What is the requirement review process? Explain it with suitable diagram.
8. How the modularity effect on software cost? Explain with suitable diagram.
9. What do you mean by SRS? Give the structure of SRS in IEEE format
10. What do you mean by requirement review?

MCQ

– Which one is not a step of requirement engineering?

a) Requirements elicitation
b) Requirements analysis
c) Requirements design
d) Requirements documentation

– SRS stands for

a) Software requirements specification
b) System requirements specification
c) Systematic requirements specifications
d) None of the above

– SRS document is for

a) “What” of a system?
b) How to design the system?
c) Costing and scheduling of a system
d) System’s requirement.

MCQ

- Context diagram explains
  a. The overview of the system
  b. The internal view of the system
  c. The entities of the system
  d. None of the above

- DFD stands for
  - Data Flow design
  - Descriptive functional design
  - Data flow diagram
  - None of the above

MCQ

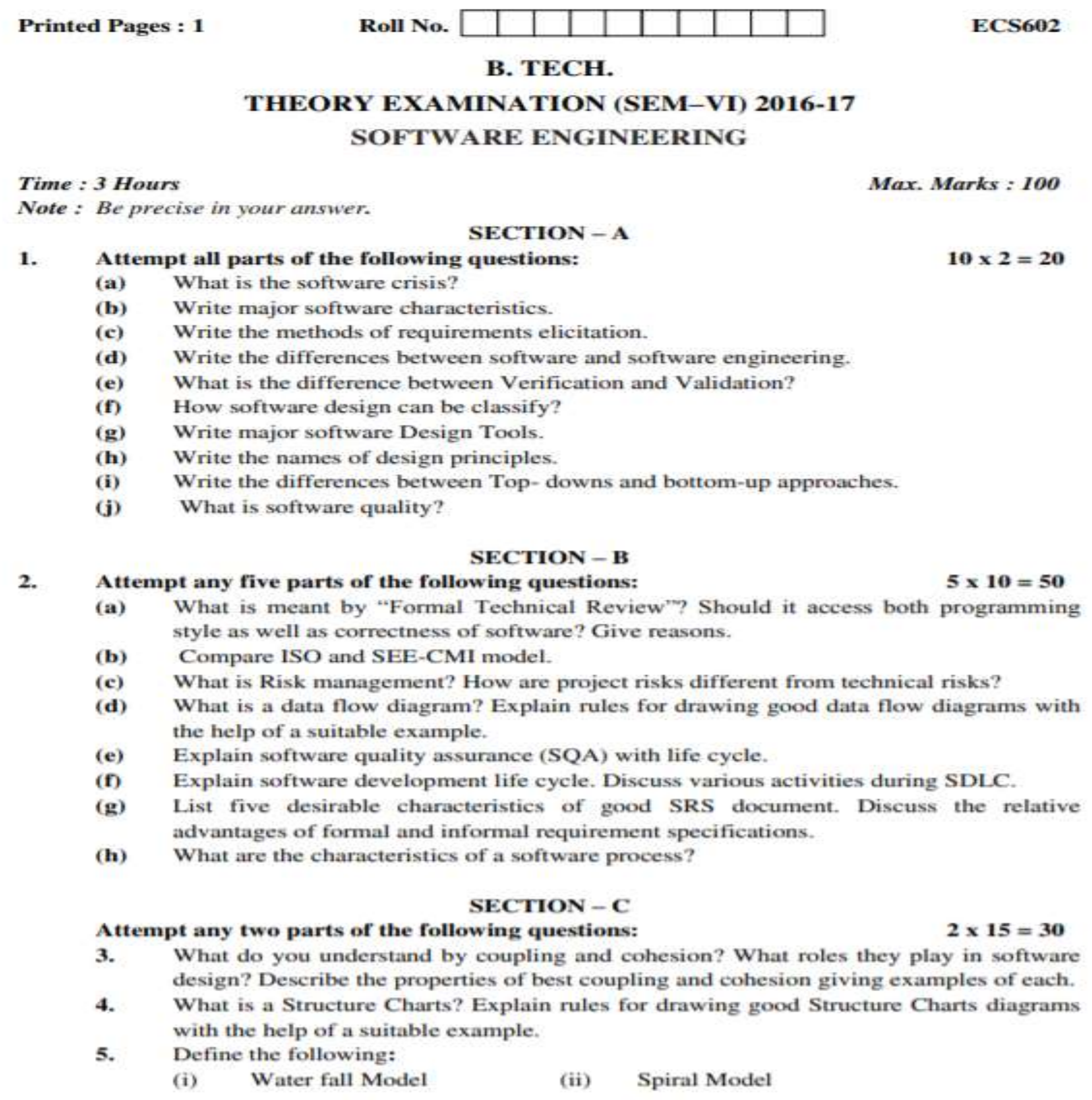
- Which is one of the most important stakeholder from the following ?

  a) Entry level personnel
  b) Middle level stakeholder
  c) Managers
  d) Users of the software

- The SRS document is also known as _____________ specification.

  a) black-box
  b) white-box
  c) grey-box
  d) none of the mentioned

Old Question Papers



Printed Pages : 1 Roll No. □□□□□□□□□□ ECS602

B. TECH.

THEORY EXAMINATION (SEM–VI) 2016-17

SOFTWARE ENGINEERING

Time : 3 Hours *Max. Marks : 100*

Note : Be precise in your answer.

SECTION – A

1. Attempt all parts of the following questions: 10 x 2 = 20

(a) What is the software crisis?
(b) Write major software characteristics.
(c) Write the methods of requirements elicitation.
(d) Write the differences between software and software engineering.
(e) What is the difference between Verification and Validation?
(f) How software design can be classify?
(g) Write major software Design Tools.
(h) Write the names of design principles.
(i) Write the differences between Top- downs and bottom-up approaches.
(j) What is software quality?

SECTION – B

2. Attempt any five parts of the following questions: 5 x 10 = 50

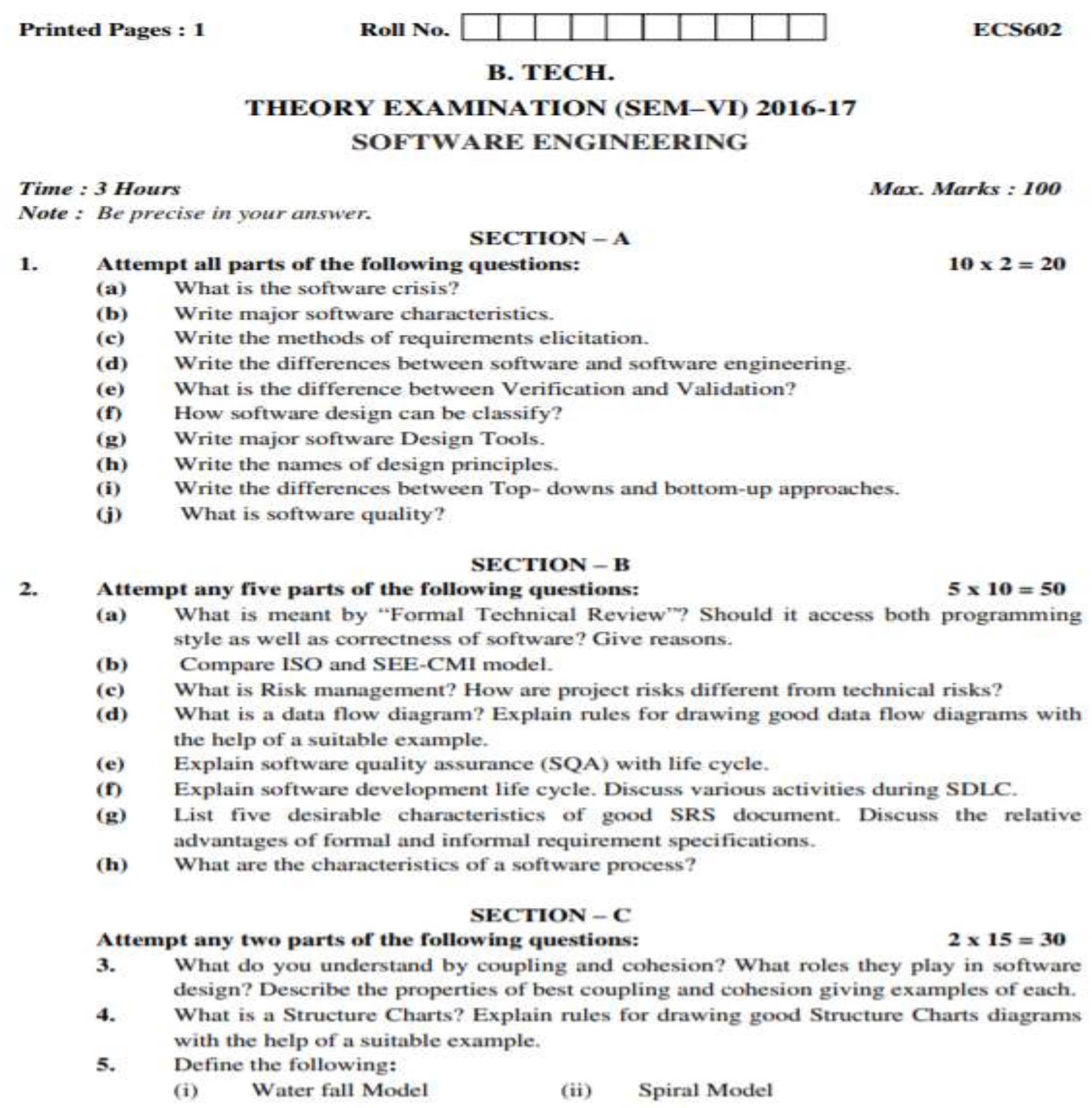
(a) What is meant by "Formal Technical Review"? Should it access both programming style as well as correctness of software? Give reasons.
(b) Compare ISO and SEE-CMI model.
(c) What is Risk management? How are project risks different from technical risks?
(d) What is a data flow diagram? Explain rules for drawing good data flow diagrams with the help of a suitable example.
(e) Explain software quality assurance (SQA) with life cycle.
(f) Explain software development life cycle. Discuss various activities during SDLC.
(g) List five desirable characteristics of good SRS document. Discuss the relative advantages of formal and informal requirement specifications.
(h) What are the characteristics of a software process?

SECTION – C

Attempt any two parts of the following questions: 2 x 15 = 30

3. What do you understand by coupling and cohesion? What roles they play in software design? Describe the properties of best coupling and cohesion giving examples of each.
4. What is a Structure Charts? Explain rules for drawing good Structure Charts diagrams with the help of a suitable example.
5. Define the following:
   (i) Water fall Model (ii) Spiral Model

Old Question Papers



Printed Pages : 1 Roll No. □□□□□□□□□□ ECS602

B. TECH.

THEORY EXAMINATION (SEM–VI) 2016-17

SOFTWARE ENGINEERING

Time : 3 Hours *Max. Marks : 100*

Note : Be precise in your answer.

SECTION – A

1. Attempt all parts of the following questions: **10 x 2 = 20**

(a) What is the software crisis?
(b) Write major software characteristics.
(c) Write the methods of requirements elicitation.
(d) Write the differences between software and software engineering.
(e) What is the difference between Verification and Validation?
(f) How software design can be classify?
(g) Write major software Design Tools.
(h) Write the names of design principles.
(i) Write the differences between Top- downs and bottom-up approaches.
(j) What is software quality?

SECTION – B

2. Attempt any five parts of the following questions: **5 x 10 = 50**

(a) What is meant by "Formal Technical Review'"? Should it access both programming style as well as correctness of software? Give reasons.
(b) Compare ISO and SEE-CMI model.
(c) What is Risk management? How are project risks different from technical risks?
(d) What is a data flow diagram? Explain rules for drawing good data flow diagrams with the help of a suitable example.
(e) Explain software quality assurance (SQA) with life cycle.
(f) Explain software development life cycle. Discuss various activities during SDLC.
(g) List five desirable characteristics of good SRS document. Discuss the relative advantages of formal and informal requirement specifications.
(h) What are the characteristics of a software process?

SECTION – C

Attempt any two parts of the following questions: **2 x 15 = 30**

3. What do you understand by coupling and cohesion? What roles they play in software design? Describe the properties of best coupling and cohesion giving examples of each.
4. What is a Structure Charts? Explain rules for drawing good Structure Charts diagrams with the help of a suitable example.
5. Define the following:
(i) Water fall Model (ii) Spiral Model

Expected Questions for University Exam

1. State any two problems that may associated during Requirement Analysis.
2. What do you mean by umbrella of activities in Software Quality assurance?
3. Prepare a ER diagram for hotel management system and specify all attribute of entities.
4. What are the difference between validation and verification?
5. What do you mean by data dictionary and decision table? Explain with an example.
6. What do you mean Software Quality Assurance?
7. What do you mean by ISO 9000 series?
8. What do you mean by SEI-CMM model? Explain its all level with suitable diagram in details.

Summary

- Software Requirement Specifications (SRS)
- Requirement Engineering Process
- Elicitation, Analysis, Documentation,
- Review and Management of User Needs,
- Feasibility Study, Information Modeling,
- Data Flow Diagrams,
- Entity Relationship Diagrams,
- Decision Tables,
- SRS Document,
- IEEE Standards for SRS.
- Software Quality Assurance
- Verification and Validation,
- SQA Plans,
- Software Quality Frameworks,
- ISO 9000 Models,
- SEI-CMM Model

References

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
3. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
4. Pankaj Jalote, Software Engineering, Wiley
5. Deepak Jain,” Software Engineering: Principles and Practices”,Oxford University Press.
6. Munesh C. Trivedi, Software Engineering, Khanna Publishing House
7. N.S. Gill, Software Engineering, Khanna Publishing House