Question Paper for IAT-1

CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

First Internal Test

Sub:	Software Engineering					Sub Code:	15CS42	Branch:	ISE
Date:	12/03/18	Duration:	90 min's	Max Marks:	50	Sem/Sec:	IV/ A & B		OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain the term software engineering. What are the key challenges that software engineering is facing?	[05]	CO1	L2
1 (b)	Describe the professional responsibilities of a software engineer.	[05]	CO2	L1
2	What is software process model? Explain incremental model with neat diagram with its merits and demerits.	[10]	CO4	L3
3	Explain functional and non-functional requirements of any system.	[10]	CO1	L3
4	Briefly discuss the desirable characteristics and structure of a requirement document.	[10]	CO1	L2
5(a)	Describe the requirement elicitation and analysis process with diagram.	[5]	CO3	L5
5(b)	What is software? What are the attributes of good software?	[5]	CO4	L3
6 (a)	What are the fundamental activities of software engineering?	[04]	CO1	L1
6 (b)	With neat diagram, explain the spiral model of software development process.	[06]	CO4	L3
7 (a)	Explain the different checks to be carried out during requirement validation process.	[05]	CO1	L2
7 (b)	Explain various ways of writing requirement specification.	[05]	CO5	L4

Solution and Scheme

CMR INSTITUTE OF TECHNLOGY

USN

First Internal Test

Sub:	**Software Engineering**				Sub Code:	15CS42	Branch:	**ISE**	
Date:	12/03/18	Duration:	90 min's	Max Marks:	50	Sem/Sec:	IV/ A & B		OBE

Answer any FIVE FULL Questions

	MARKS	CO	RBT
1 (a) Explain the term software engineering. What are the key challenges that software engineering is facing?	[05]	CO1	L2

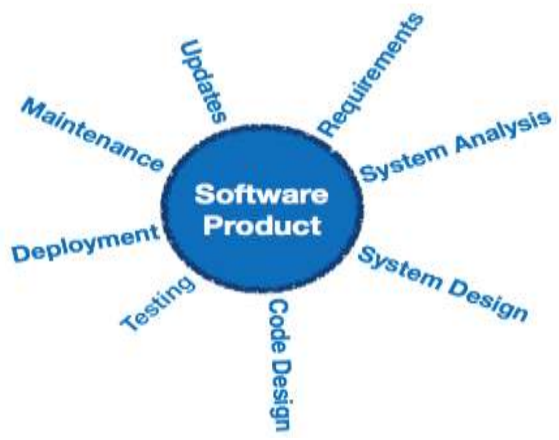
Scheme: Explanation for Software Engineering (02 Marks)

Key Challenges (03 Marks)

Solution:

Explanation for Software Engineering:

Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product. The process of developing a software product using software engineering principles and methods is referred to as **software Engineering.** This includes the initial development of software and its maintenance and updates, till desired software product is developed, which satisfies the expected requirements.



Key Challenges:

The need of software engineering arises because of higher rate of change in user requirements and environment on which the software is working.

- **Large software -** It is easier to build a wall than to a house or building, likewise, as the size of software become large engineering has to step to give it a scientific process.
- **Scalability-** If the software process were not based on scientific and engineering concepts, it would be easier to re-create new software than to scale an existing one.
- **Cost-** As hardware industry has shown its skills and huge manufacturing has lower down he price of computer and electronic hardware. But the cost of software remains high if proper process is not adapted.
- **Dynamic Nature-** The always growing and adapting nature of software hugely depends

upon the environment in which user works. If the nature of software is always changing, new enhancements need to be done in the existing one. This is where software engineering plays a good role.

- **Quality Management-** Better process of software development provides better and quality software product.

1 (b) Describe the professional responsibilities of a software engineer. [05] CO2 L1

Scheme: Each responsibility and its explanation 1.25 Marks each

1. **Confidentiality** -Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
2. **Competence** - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is out with their competence.
3. **Intellectual Property Rights** - Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
4. **Computer Misuse-** Software engineers should not use their technical skills to misuse other people's computers.

2 What is software process model? Explain incremental model with neat diagram with its merits and demerits. [10] CO4 L3

Scheme: Software process model (03 Marks)
Incremental Model with diagram (03Marks)
Merits and demerits (04 Marks)
Solution:

Software process model:

A set of activities whose goal is the development or evolution of software.
Generic activities in all software processes are:

- Specification - what the system should do and its development constraints
- Development - production of the software system
- Validation - checking that the software is what the customer wants
- Evolution - changing the software in response to changing demands

A simplified representation of a software process, presented from a specific perspective. Examples of process perspectives are

- Workflow perspective - sequence of activities
- Data-flow perspective - information flow
- Role/action perspective - who does what

Generic process models

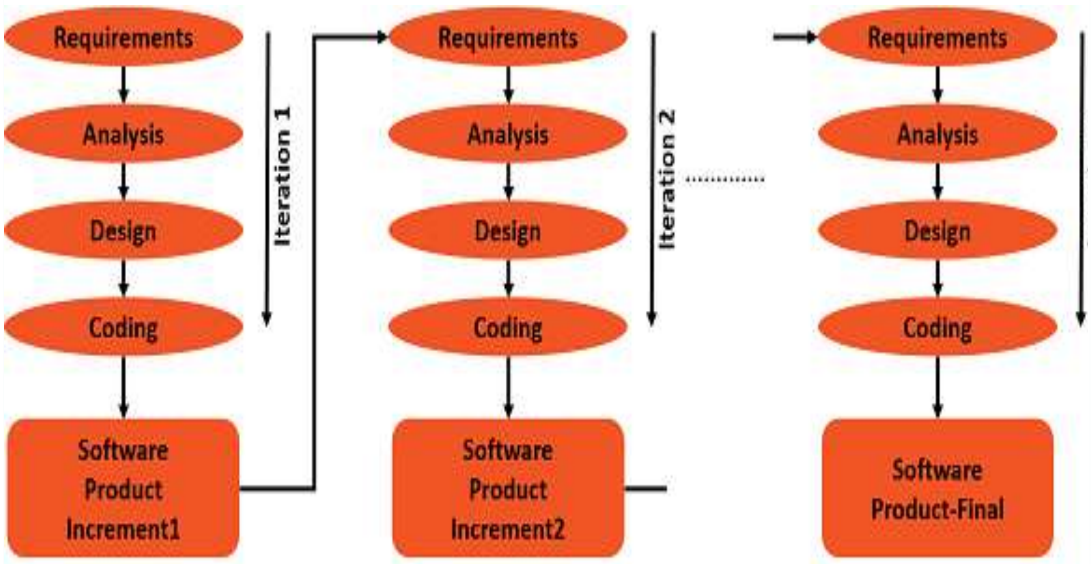
- Waterfall
- Incremental Model
- Spiral Model

Incremental Model with diagram:

- In an Iterative Incremental model, initially, a partial implementation of a total system is constructed so that it will be in a deliverable state.
- Increased functionality is added. Defects, if any, from the prior delivery are fixed and the working product is delivered.
- The process is repeated until the entire product development is completed.
- The repetitions of these processes are called iterations. At the end of every iteration,

a product increment is delivered.

For example, the word-processing software is developed using the incremental model.



Merits:

- You can develop prioritized requirements first.
- Initial product delivery is faster.
- Customers gets important functionality early.
- Lowers initial delivery cost.
- Each release is a product increment, so that the customer will have a working product at hand all the time.
- Customer can provide feedback to each product increment, thus avoiding surprises at the end of development.
- Requirements changes can be easily accommodated.

Demerits:

- Requires effective planning of iterations.
- Requires efficient design to ensure inclusion of the required functionality and provision for changes later.
- Requires early definition of a complete and fully functional system to allow the definition of increments.
- Well-defined module interfaces are required, as some are developed long before others are developed.
- Total cost of the complete system is not lower.

3 Explain functional and non-functional requirements of any system. [10] CO1 L3

Scheme: Functional (4 Marks)
Non-functional (06 Marks)
Solution:

Functional requirements

- Describe functionality or system services
- Depend on the type of software, expected users and the type of system where the software is used
- Functional user requirementsmay be high-level statements of what the system should do; functional system requirements should describe the system services in detail

Examples

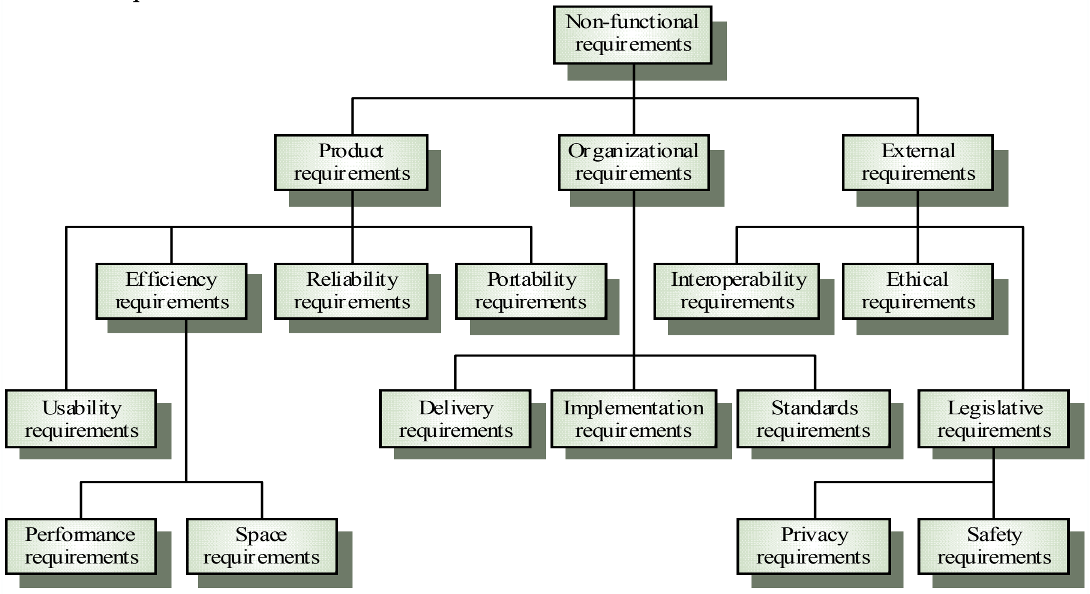
- The user shall be able to search either all of the initial set of databases or select a subset from it

- The system shallprovideappropriateviewersfor the user to readdocuments in the document store.
- Every order shall be allocated a unique identifier (ORDER ID) which the user shall be able to copy to the account's permanent storage area.

Non-functional requirements

- Product requirements
  - Product behavior
  - Ex: Timing, performance, memory, reliability, portability, usability
- Organizational requirements
  - Policies and procedures in the customer's and developer's organizations
  - Example: Process requirements, implementation requirements, delivery requirements
- External requirements
  - Factors externals to the system and the development process
  - Example: Interoperability, legislation, ethics

- Non-functional requirements may be more critical than functional requirements.



4 Briefly discuss the desirable characteristics and structure of a requirement document. [10] CO1 L2

Scheme: Characteristics of SRS (02 Marks)
Structure of requirement of document (08 Marks)

Characteristics of SRS:

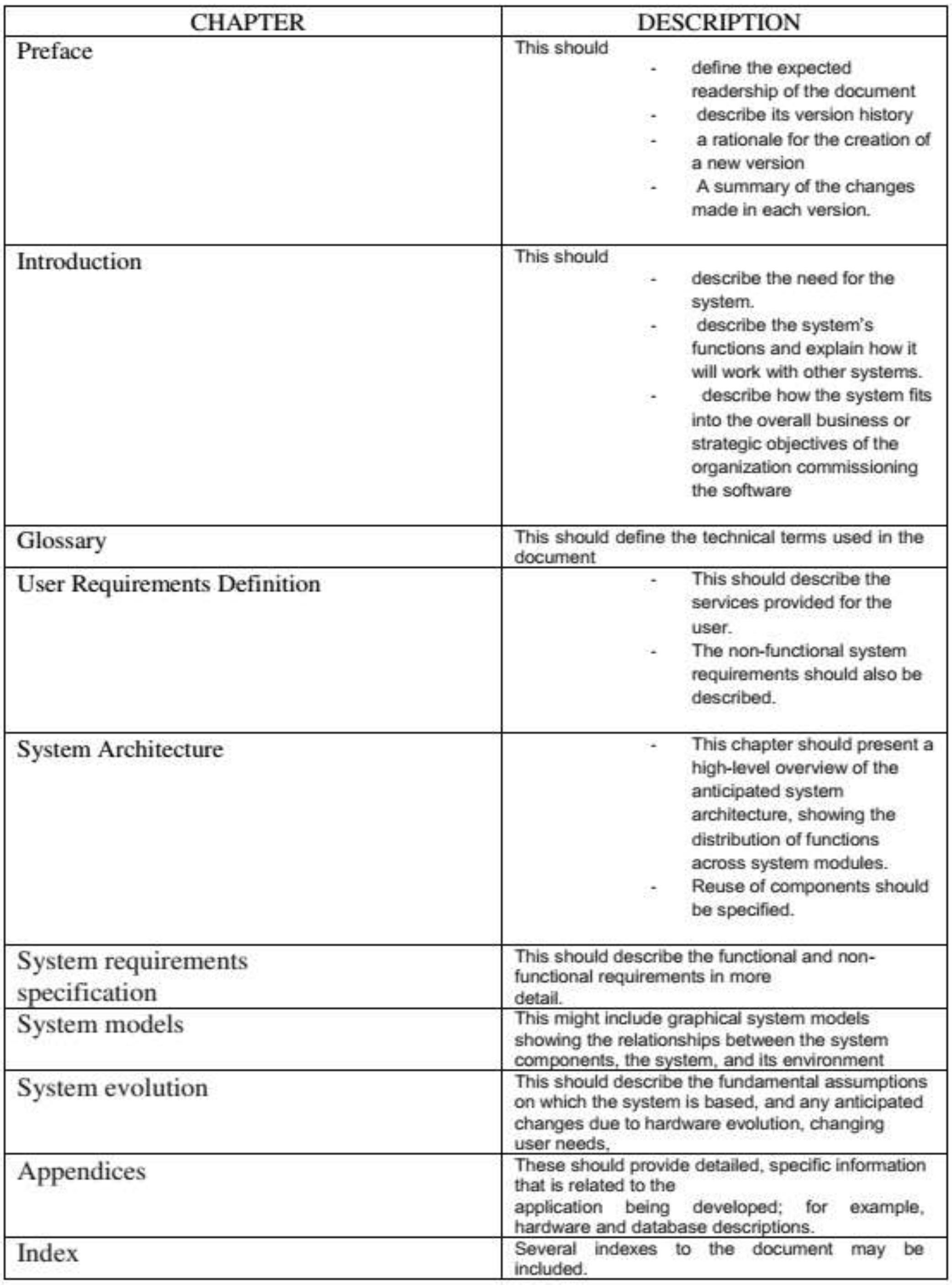
To properly satisfy the basic goals, an SRS should have certain properties and should contain different types of requirements. Some of the desirable characteristics of an SRS are

1. Correct
2. Complete
3. Unambiguous
4. Verifiable
5. Consistent

6. Ranked for importance and/or stability

Structure of requirement of document:

SOLUTION:



CHAPTER	DESCRIPTION
Preface	This should - define the expected readership of the document - describe its version history - a rationale for the creation of a new version - A summary of the changes made in each version.
Introduction	This should - describe the need for the system. - describe the system's functions and explain how it will work with other systems. - describe how the system fits into the overall business or strategic objectives of the organization commissioning the software
Glossary	This should define the technical terms used in the document
User Requirements Definition	- This should describe the services provided for the user. - The non-functional system requirements should also be described.
System Architecture	- This chapter should present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. - Reuse of components should be specified.
System requirements specification	This should describe the functional and non-functional requirements in more detail.
System models	This might include graphical system models showing the relationships between the system components, the system, and its environment
System evolution	This should describe the fundamental assumptions on which the system is based, and any anticipated changes due to hardware evolution, changing user needs.
Appendices	These should provide detailed, specific information that is related to the application being developed; for example, hardware and database descriptions.
Index	Several indexes to the document may be included.

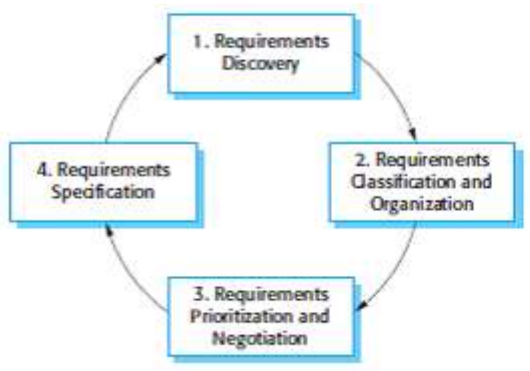
5(a) Describe the requirement elicitation and analysis process with diagram. [5] CO3 L5

Scheme: Description (03 Marks) +diagram (02 Marks)

Description:

- Sometimes called requirements elicitation or requirements discovery

- Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints
- May involve end-users, managers, engineers involved in maintenance, domain experts, trade unions, etc. These are called *stakeholders*


1. Requirements Discovery
2. Requirements Classification and Organization
3. Requirements Prioritization and Negotiation
4. Requirements Specification

- **Requirements gathering** - The developers discuss with the client and end users and know their expectations from the software.
- **Organizing Requirements** - The developers prioritize and arrange the requirements in order of importance, urgency and convenience.
- **Negotiation & discussion** - If requirements are ambiguous or there are some conflicts in requirements of various stakeholders, if they are, it is then negotiated and discussed with stakeholders. Requirements may then be prioritized and reasonably compromised.

  The requirements come from various stakeholders. To remove the ambiguity and conflicts, they are discussed for clarity and correctness. Unrealistic requirements are compromised reasonably.

- **Documentation** - All formal & informal, functional and non-functional requirements are documented and made available for next phase processing.

5(b) What is software? What are the attributes of good software? [5] CO4 L3

Scheme: Software Definition (01 mark)

Characteristics of good software (04 marks)

Software is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called **software product.** A software product can be judged by what it offers and how well it can be used.

This software must satisfy on the following attributes:

- Operational
- Transitional
- Maintenance

Well-engineered and crafted software is expected to have the following characteristics:

Operational

This tells us how well software works in operations. It can be measured on:

- Budget
- Usability
- Efficiency
- Correctness
- Functionality
- Dependability
- Security
- Safety

Transitional

This aspect is important when the software is moved from one platform to another:

- Portability
- Interoperability
- Reusability
- Adaptability

Maintenance

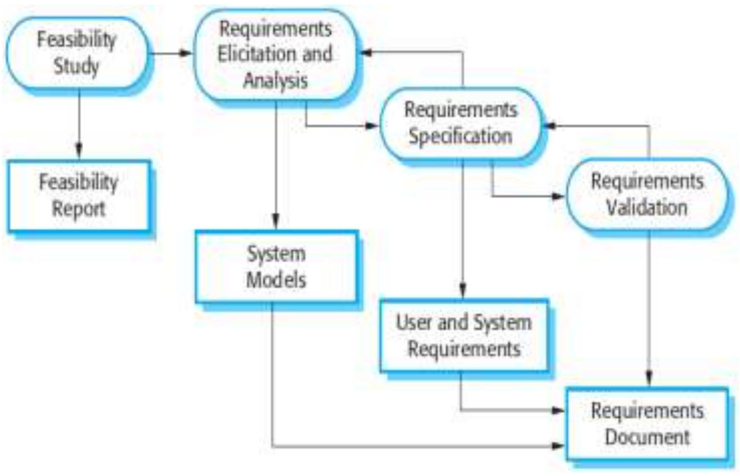
This aspect briefs about how well a software has the capabilities to maintain itself in the ever-changing environment:

- Modularity
- Maintainability
- Flexibility
- Scalability

6(a) What are the fundamental activities of software engineering? [04] CO1 L1

Scheme: List all activities with description + diagram (1+2+1)

There are four main activities (or sub-activities) of requirements engineering:



1. **Feasibility study:** An estimate is made of whether the identified can be achieved using the current software and hardware technologies, under the current budget, etc. The feasibility study should be cheap and quick; it should inform the decision of whether or not to go ahead with the project.

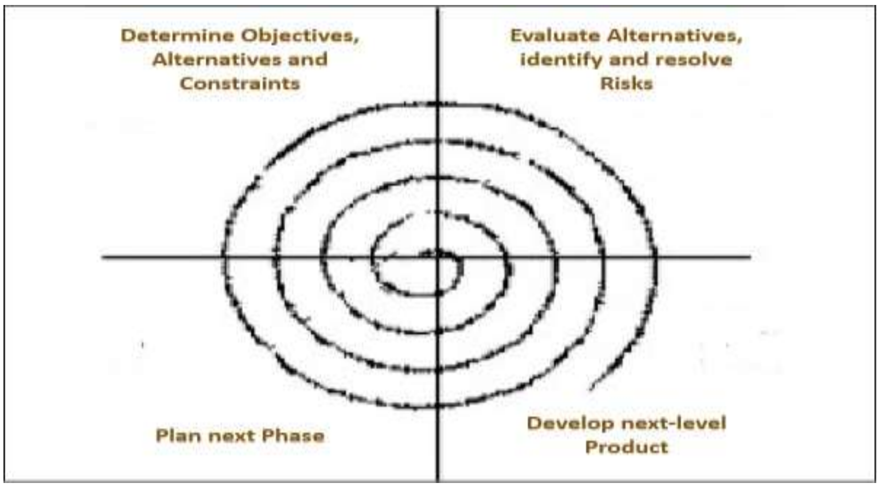
2. **Requirements elicitation and analysis:** This is the process of deriving the system requirements through observation of existing systems, discussions with stakeholders, etc. This may involve the development of one or more system models and prototypes that can help us understanding the system to be specified.

3. **Requirements specification:** It's the activity of writing down the information gathered during the elicitation and analysis activity into a document that defines a set of requirements. Two types of requirements may be included in this document; user and system requirements.

4. **Requirements validation:** It's the process of checking the requirements for realism, consistency and completeness. During this process, our goal is to discover errors in the requirements document. When errors are found, it must be modified to correct these problems.

6(b) With neat diagram, explain the spiral model of software development process.
[06] CO4 L3

- Spiral model adds Risk Analysis and RAD prototyping to the Waterfall model.
- Each cycle involves the same sequence of steps as the Waterfall model.
- Spiral model has four quadrants.


Determine Objectives, Alternatives and Constraints
Evaluate Alternatives, identify and resolve Risks
Plan next Phase
Develop next-level Product

❑ **Quadrant 1 -** Determine objectives, alternatives and constraints
- **Objectives** – Functionality, performance, hardware/software interface, critical success factors, etc.
- **Alternatives** – Build, reuse, buy, sub-contract, etc.
- **Constraints** – Cost, schedule, interface, etc.

❑ **Quadrant 2 -** Evaluate alternatives, identify and resolve risks
- Study alternatives relative to the objectives and constraints that are determined.
- Identify risks such as lack of experience, new technology, tight schedules, etc.
- Resolve the identified risks evaluating their impact on the project, identifying the needed mitigation and contingency plans and implementing them. Risks always need to be monitored.

❑ **Quadrant 3 -** Develop next-level product
- Create a design
- Review design
- Develop code
- Inspect code
- Test product

❑ **Quadrant 4 -** Plan next phase
- Develop project plan
- Develop configuration management plan
- Develop a test plan
- Develop an installation plan

7 (a)	Explain the different checks to be carried out during requirement validation process. [5]	CO1 L2

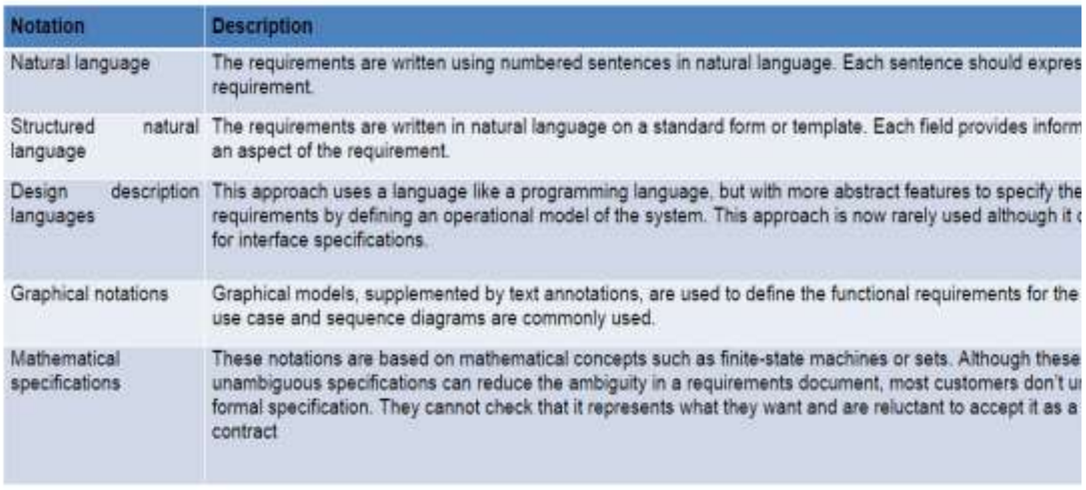
Scheme: Naming and Explanation (1+04 Marks)

Concerned with demonstrating that the requirements define the system that the customer really wants. Requirements error costs are high so validation is very important. Fixing a requirements error after delivery may cost up to 100 times the cost of fixing an implementation error

- Requirements reviews: Systematic manual analysis of the requirements
- Prototyping: Using an executable model of the system to check requirements.
- Test-case generation: Developing tests for requirements to check testability
- Automated consistency analysis : Checking the consistency of a structured requirements description

7 (b)	Explain various ways of writing requirement specification.	[05]	CO5 L4

Scheme: naming and description(1+04 marks)



Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should expres requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides inform an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't u formal specification. They cannot check that it represents what they want and are reluctant to accept it as a contract