

1

Name : Abhishek Srivastava.

Reg. No. : 19BCE10071

Subject : Software Engineering.

Slot : B11 + DB1

Date : 01/09/21

Class No : 0034.

2) (a) Analysis model is a technical representation of the system. It acts as a link between system description and design model.

In Analysis modeling, information, behaviour and functions of the system are defined and translated into a the architecture, component, and interface level design in the design modeling.

It is a model of data domain which :

- define data objects.
- describe data attributes.
- establish data relationship.

(2)

Functions of Analysis Model is to :-

- Identify functions that transform object data.
- Identify how data flow through the system.
- represents producers and consumers of data.
- indicate different states of the system.
- specify events that cause the system to change data.

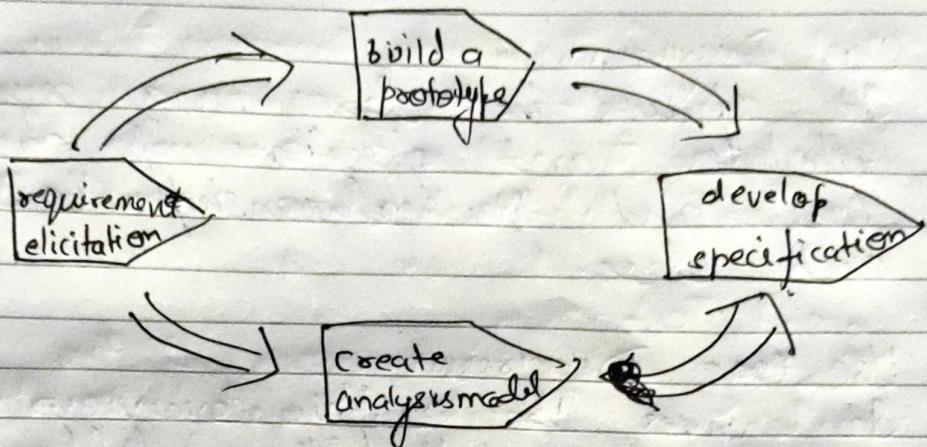
Partition of Model

- define each model to represent ~~lower~~ lower level of abstraction.
  - define data objects
  - create a functional hierarchy
  - represent behaviour at different level in detail.

Essence of Analysis Model.

Begin by focusing on the essence of the problem without regarding to implementation details.

3



### Analysis Process.

#### Objectives of Analysis Modelling,

- It must establish a way of creating software design.
- It must describe the requirement to the customer.
- It must define a set of requirements that can be validated, once the software is built.

#### Types of UML

Behavioural.

Eg: Class diagram

Structural.

Eg: Sequence diagram

(4)

### Class Diagram.

It shows the classes in a system, attributes, and operations of each class and the relationships between each class.

### Use Case Diagram

It gives the graphic overview of the actors and how those different functions interact.

### Sequence diagram

It shows how objects interact with each other and the order in which those interactions occur. Process is represented vertically and interactions are shown in arrows.

### Timing Diagram

It is similar to sequence diagram.  
It represents behaviour of the objects at a given time frame.

### Activity Diagram

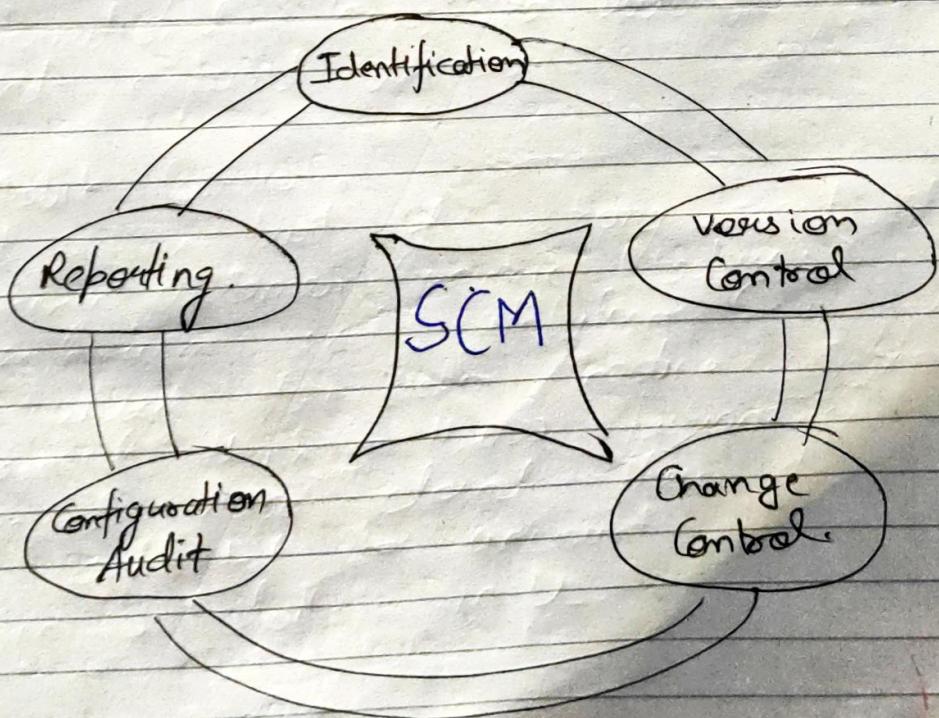
It represents workflow in a graphical way. It defines business / operational workflow of any component.

5

### 3) (b) Software Configuration Management

Software Configuration Management (SCM) is a branch of Software Engineering to provide a better process to handling, organizing and controlling the changes in requirements, code, team and other elements in the software project management & development life cycle.

The SCM primarily deals with version selection, tracking the changes and version control of software projects with high productivity and minimize the errors or risk factor.



(6)

SCM is a tool used in Software development lifecycle process to track, changing and managing the configuration items inside the product development. The product manager, developers, configuration manager, the product owner and testers are involved in the SCM process.

Processes involved in completing software configuration management are:-

- Planning and Identification Process.  
This is initial ~~process~~ level of the SCM process to planning properly for the development of the application and identifies the configurations item as per the scope of the project.
- Version Control Process or Baselines  
It indicates to store the different version of development/configuration by changing the scope of requirement, code or software environment.
- Changing Control Process.  
It is a new request created by the client to change some configuration on the software product.

7  
a

- Configuration Release Process

It is used to ensure that application will be developed as per the project plan and test/ verify that the application is per scope.

- Configuration Auditing Process

This is to verify if the software product is as per baseline or not.

- Review and Status reporting process.

In this process, we go through multiple reviews of an application to develop some application-related documents like user-manual, guide, etc.

### EXAMPLE

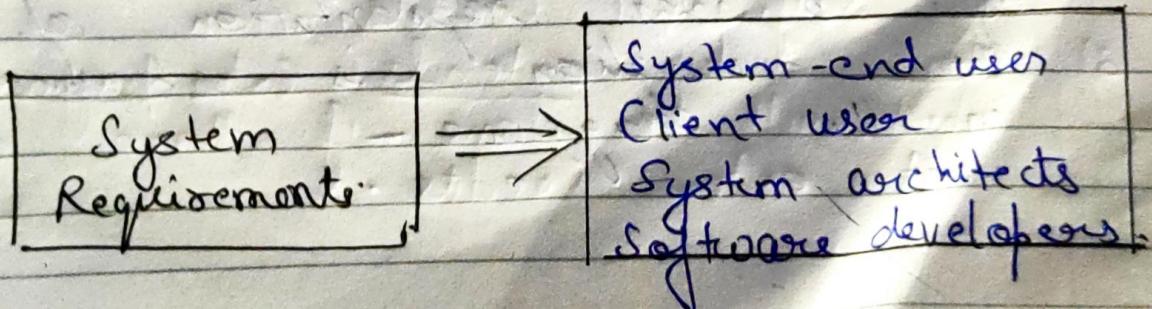
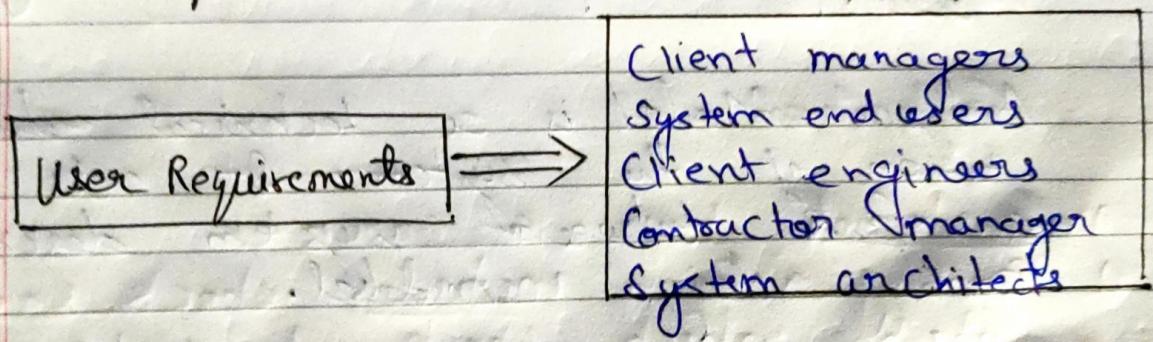
A telecom firm maintains configuration management database that includes relationships between components. This is used to automatically determine the impact of failures. For example, if router goes down ~~when~~, the firm will have list of impacted services.

(8)

1)(a) User requirements are statements, in natural language but plus diagrams, of what services the system is expected to provide to system users and the customers under which it must operate.

System requirements are more detailed description of the software system's function, services and operational components constraints. The system requirement document should define exactly what is to be required and implemented.

### Example



(9)

The readers of the user requirement are not usually concerned with ~~the~~ how the system will be implemented and ~~they~~ may be managers who are not interested in the details facilities of the system.

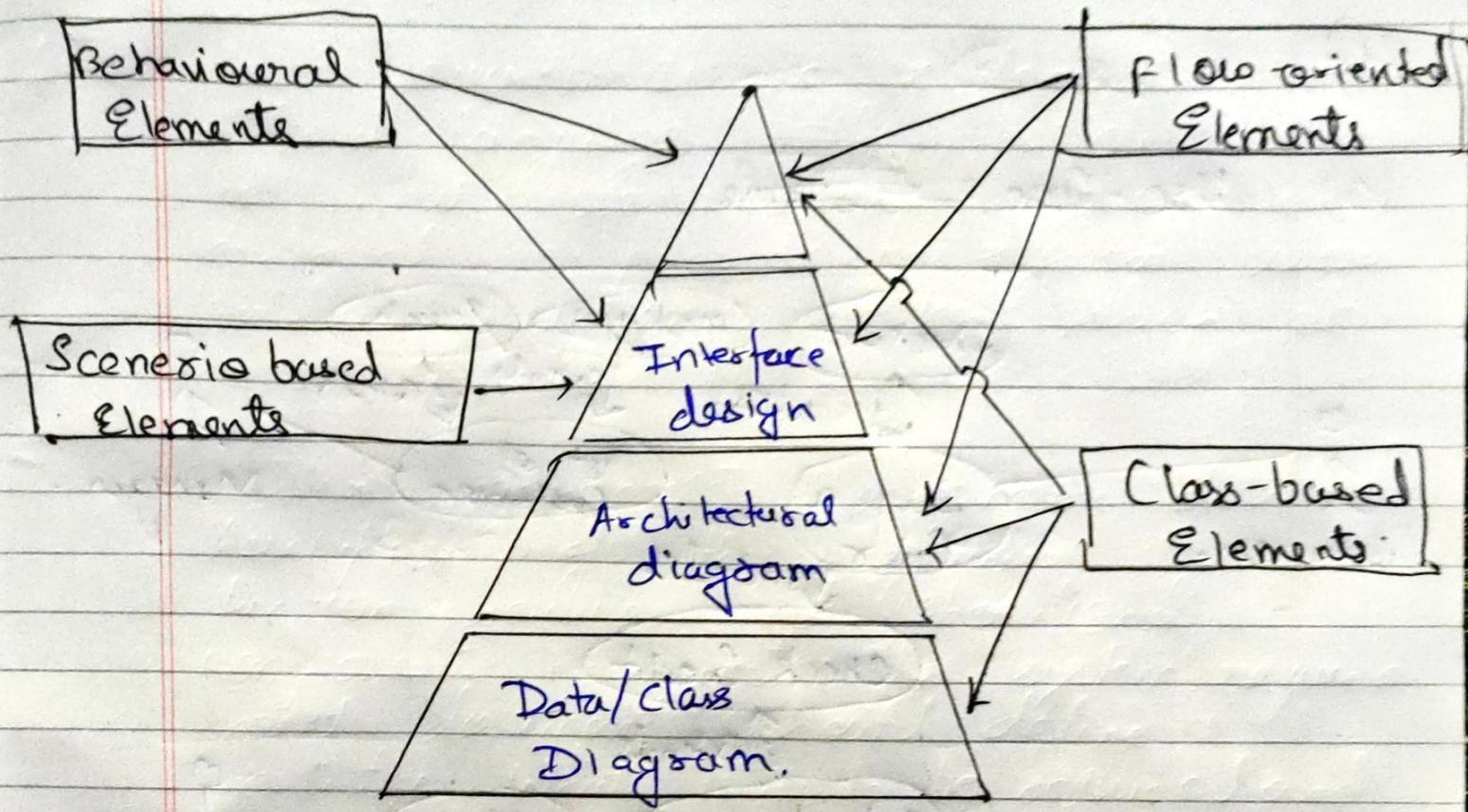
The readers of ~~the~~ system requirement need to know more precisely what the system will do because they are concerned with how it will support the business processes or because they are involved in the system implementation.

## Translating Analysis Model to Design Model.

Analysis Model is a technical representation of the system. It acts as a link between system description and design model.

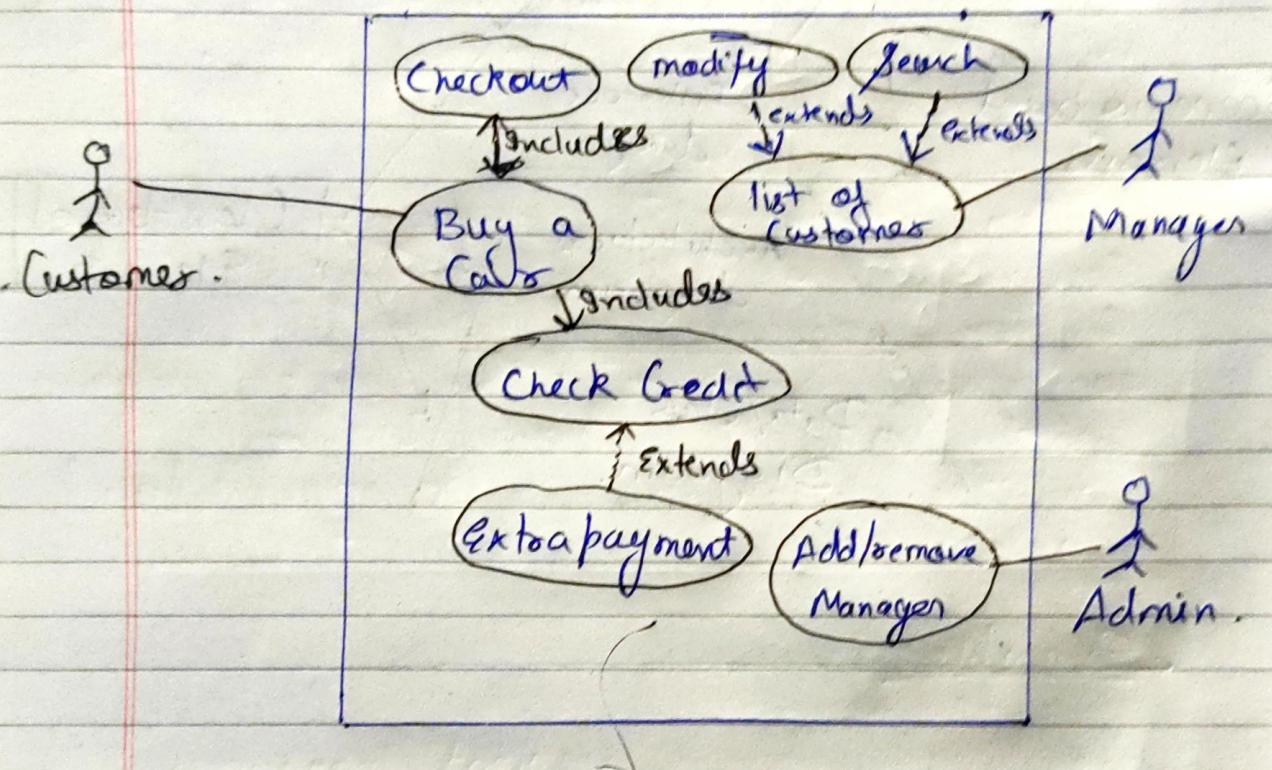
The ideal representation of this translation is shown in the diagram:-

10



4)

(a) Use-Case & ~~Customer~~

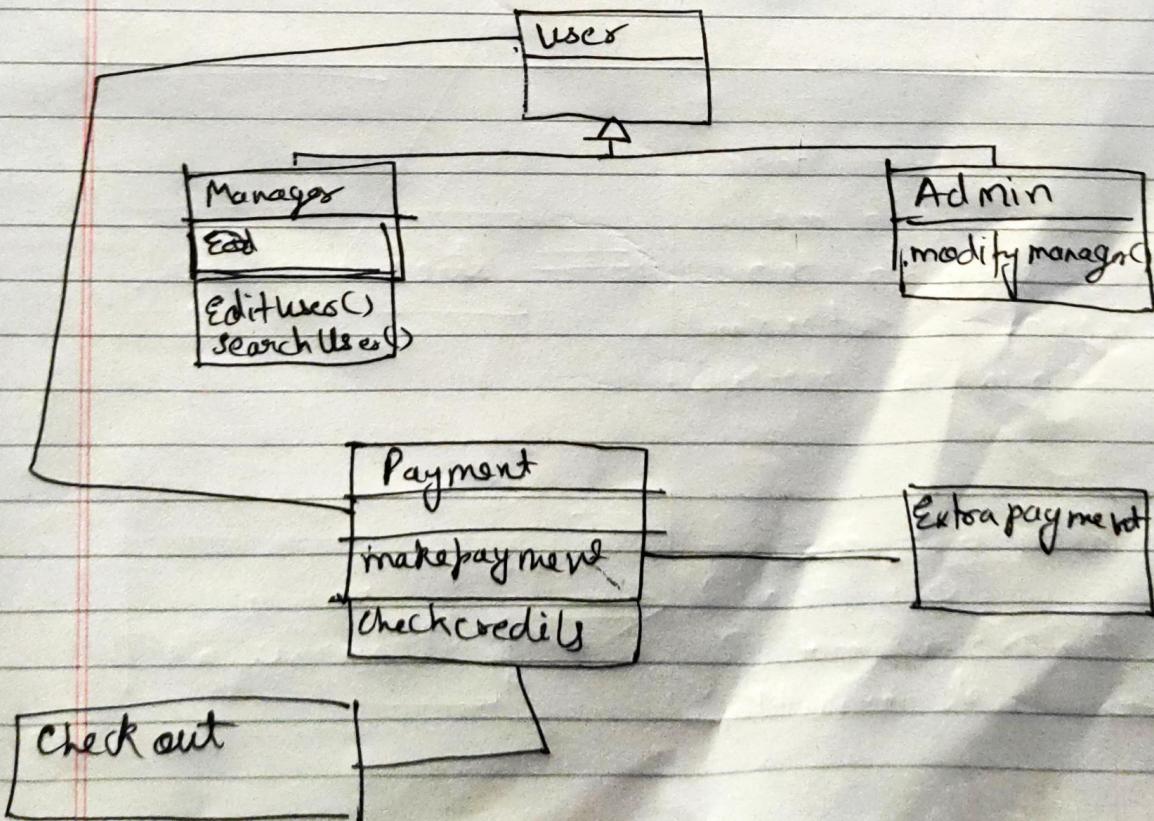


Explanation.

- 1) Customer buys a car.
- 2) Buy a Car includes check credit which which Check if customer has sufficient credits.
- 3) Check credit extends Extra payment which will be done if there are not extra credits.

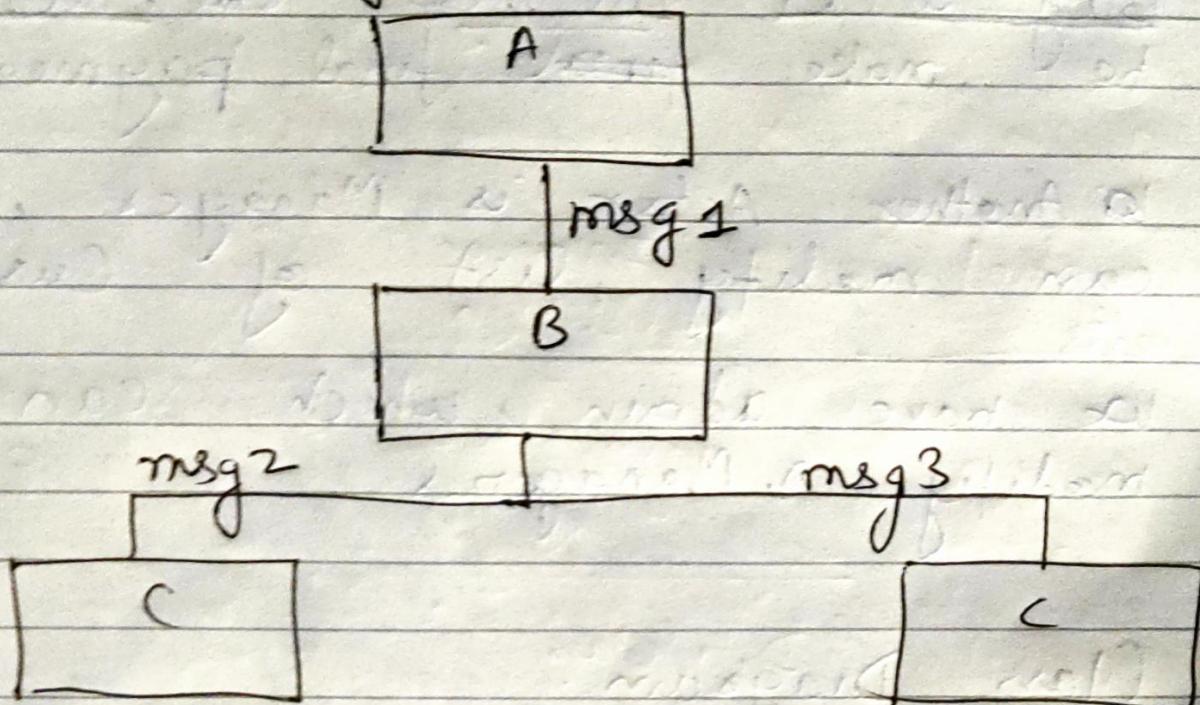
- 4) Once credits are checked, then buy a car includes ~~the checkout~~ to make final payment
- 5) Another Actor is Manager, which can modify list of customers
- 6) We have admin, which can modify Managers

(b) Class Diagram.



(B)

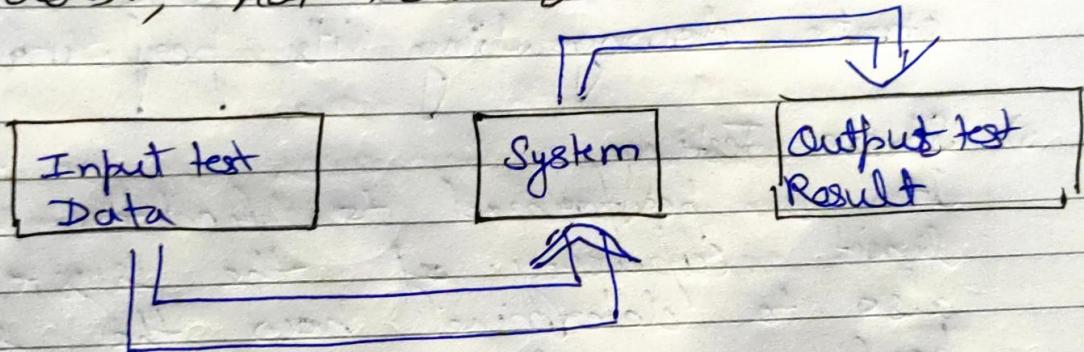
(c) Class Diagram.



14

5) Testing is intended to show that a program does what it is intended to do and to discover program defects before it is put into use. When we test software, we execute the program using artificial data.

"Testing can only show the presence of errors, not their absence".



To provide good user experience, we go ahead with NON-FUNCTIONAL Testing.

Non-functional testing is defined to figure out if your product will provide a good user experience.

Types of non-functional testing:

### 1) Performance Testing

To evaluate the performance of components of a particular system under a particular workload.

### 2) Load Testing

Testing the behaviour of the system under a specific load or to get the break even points where system starts downgrading its performance.

### 3) Stress Testing

It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

### 4) Security Testing

This intends to uncover vulnerabilities of the system and determine that its data and resources are protected from possible intruders.

5) Portability testing.

Software reliability is the probability that software will work properly in a specific environment and for a given amount of time.