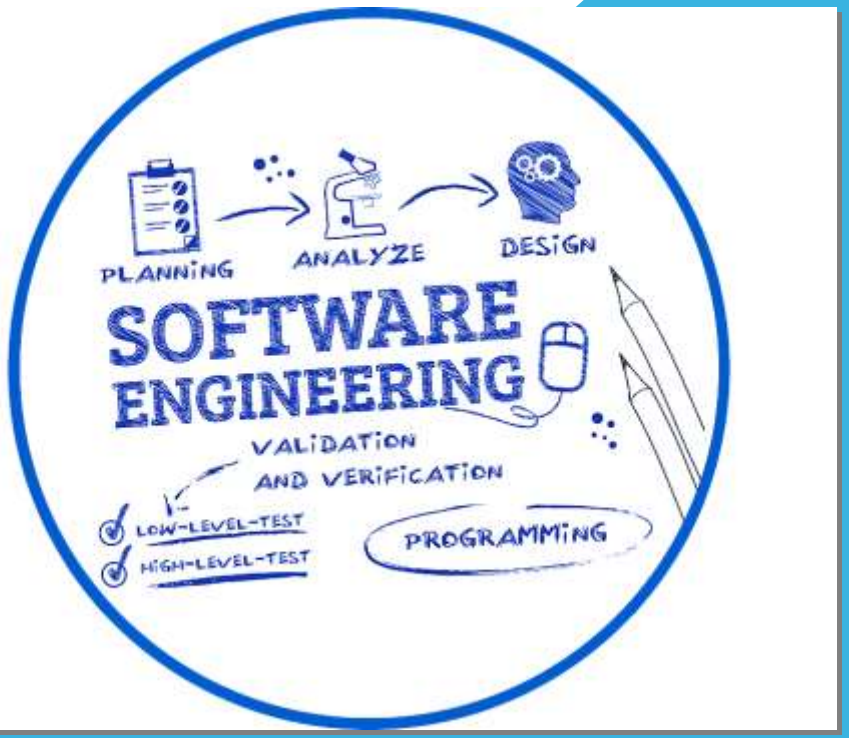
SOFTWARE F

CHAPTER-2



Contents-PART-I

Software Requirements:

- Functional Requirements
- Non-Functional Requirements
- User Requirements
- System Requirements
- Interface Specification
- Software Requirements Document

CONTENTS-PART-II

Requirements Engineering Process:

- Feasibility Studies
- Requirements elicitation and Analysis
- Requirements validation
- Requirements Management

CONTENTS-PART-III

System Models :

- Context Models
- Behavioral Models
- Data Models
- Object Models
- Structured Models

PART –I : SOFTWARE ENGINEERING : REQUIREMENT ENGINEERING.

PART I

The Software requirements are a description of features and functionalities of the target system.

Characteristics:

- Clear
- Concise
- Correct
- Coherent
- Modifiable
- Verifiable
- Prioritized.

Functional Requirements

Functional Requirements usually define if/then behaviors and include calculations, data input and business processes.

The Features Which allow the system to function as it was intended.

Examples:

- Business Rules
- Transaction Corrections
- Authentication
- Audit Tracking
- External Interfaces.

Non-Functional Requirements

Non-Functional Requirements are the constraints or the requirements imposed on the system.

They Specify the quality attribute of the software.

Non-Functional Issues like Security, Reliability, maintenance, performance etc. are monitored

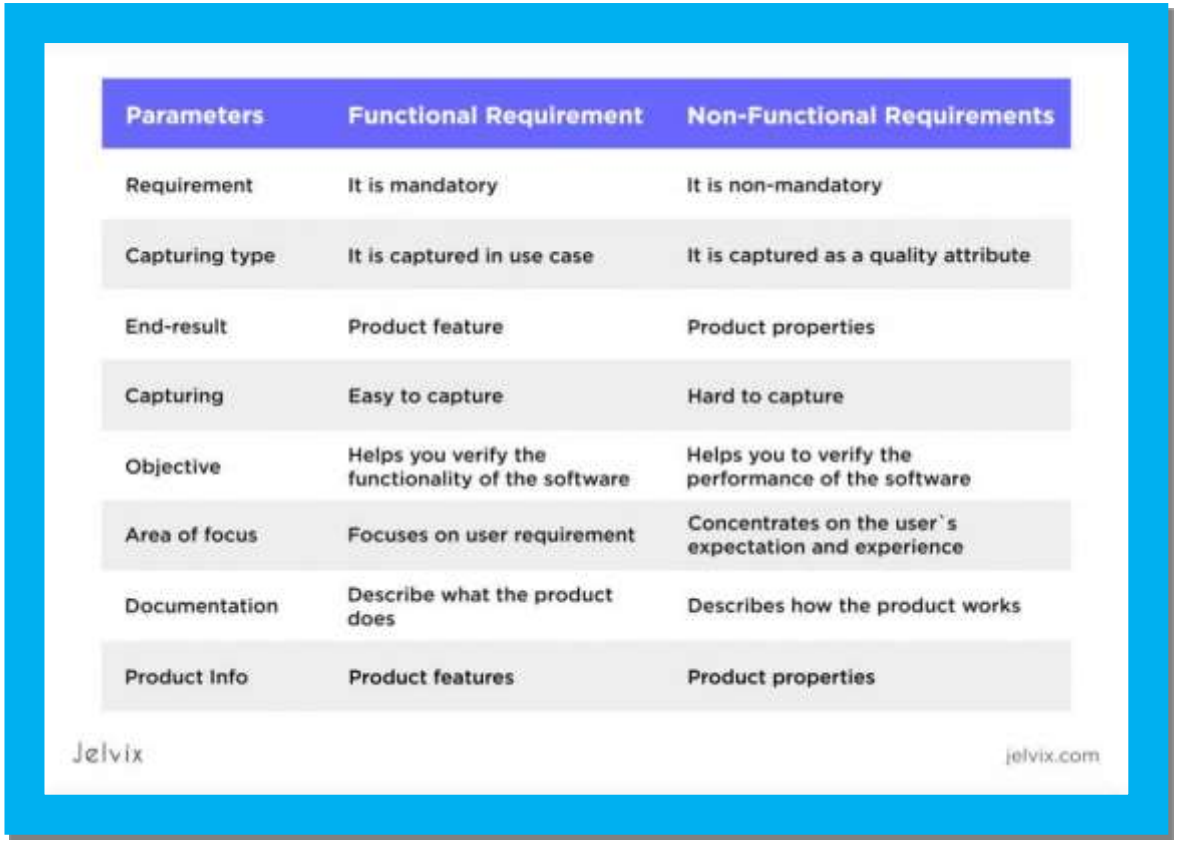
Types:

- Performance Constraints :
  - Reliability, Security, Response Time etc.
  - Operating Constraints : Physical Constraints
  - Interface Constraints : How the system is to interface with its environment.
  - Economic Constraints: Long term and Immediate Costs
  - Lifecycle Requirements : Quality of Design.

Advantages and Disadvantages of Non-Functional Requirements.

Advantages	Disadvantages
They ensure the software system follows legal and adherence rules	The nonfunctional requirement may affect the various high-level software subsystem.
They specify the quality attribute of the software.	They generally increase the cost as they require special consideration during the software architecture/high-level design phase.
They help in constructing the security policy of the software system.	It is difficult to change or alter non-functional requirements once you pass them to the architecture phase.

Functional vs Non-Functional.



Parameters	Functional Requirement	Non-Functional Requirements
Requirement	It is mandatory	It is non-mandatory
Capturing type	It is captured in use case	It is captured as a quality attribute
End-result	Product feature	Product properties
Capturing	Easy to capture	Hard to capture
Objective	Helps you verify the functionality of the software	Helps you to verify the performance of the software
Area of focus	Focuses on user requirement	Concentrates on the user`s expectation and experience
Documentation	Describe what the product does	Describes how the product works
Product Info	Product features	Product properties

User requirements

The User Requirement Specification describes the business needs for what user is expecting from the system.

It is written early in the validation process, before system is created.

Many user requirements deal with how a user will interact with a system and what that user expects

When user requirements such as these are written down, they can often break into multiple system requirements later due to switching of screens, the maximum delays in starting the process, and finally what the next screen should look like

System Requirements

A System Requirements Specification (SRS) (also known as a Software Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

In a System Requirements Document we have :

- Business Drivers
- Business Model
- Functional And System Requirement
- Business and System Use Cases
- System Qualities
- Constraints and Assumptions
- Acceptance Criteria

Interface Specification

Interface Specification refers to the document that captures the details of Software user interface into a written document.

The Specification covers all possible actions that an end user may perform all visual auditory and other interaction elements.

There are two types of UI :

- GUI (Graphical User Interface)
- CLI(Command Line Interface)

Software Requirements Document

A software requirements is a document that describes the intended use-case features and challenges of a software application.

These documents are created before the project has started development in order to get every stakeholder on the same page regarding the software's functionality.

The Document contains many parts : Introduction, General Description, Functional Requirements, Interface, Performance requirements, Design Constraints, Non-Functional Attributes, Appendices etc..

SAMPLE SRS DOCUMENT

Table of Contents for a SRS Document

1. Introduction

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. System Features

3.1 Functional Requirements

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

CHA

PART – II

Requirements Engineering Process

The Process of Defining, Documenting and Maintaining the requirements. Process of gathering and defining service provided by the system,

Consists of the following main activities :

- Requirements Elicitation
- Requirements Specification
- Requirements verification and validation
- Requirements Management.

FEASIBILITY STUDIES

Feasibility Studies in Software Engineering is a study to evaluate the feasibility of proposed project or system.

It is one of the stage among important 4 stages of Software Project Management Process.

Carried out based on many purposes to analyze whether software product will be right in terms of development , implantation , contribution etc..

Types: Technical , Operational , Economic, Legal and Schedule

Technical Feasibility

In Technical Feasibility current resources both hardware software along with required technology are analyzed/assessed to develop project.

This technical feasibility study gives report whether there exists correct required resources and technologies which will be used for project development.

Along with this, feasibility study also analyzes technical skills and capabilities of technical team, existing technology can be used or not, maintenance and up-gradation is easy or not for chosen technology

Operational Feasibility

Degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment.

Determining suggested solution by software development team is acceptable or not

Operational Feasibility helps in taking advantage of the opportunities and fulfills the requirements as identified during the development of the project.

Economic feasibility

Study Cost and Benefit of the Project is analyzed.

A detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on.

After that it is analyzed whether project will be beneficial in terms of finance for organization or not.

Legal Feasibility

Project is analyzed in legality point of view.

Analyzing barriers of legal implementation of project, data protection acts or social media laws, project certificate, license, copyright etc. are done in this study.

To confirm if proposed project conform legal and ethical requirements.

Schedule Feasibility

Timelines/ deadlines are analyzed for proposed project which includes how many teams will take to complete the Project.

Includes how many times teams will take to complete final project which has a great impact on the organization as purpose of project may fail if it can’t be completed on time.

Feasibility Study Process

Information Assessment

Information Collection

Report Writing

General Information

Requirements elicitation and Analysis

Requirements Elicitation is the practice of researching and discovering the requirements of a system from users, customers and other stakeholders.

This process is also known as requirement gathering.

It is needed to know what the users really need.

This step defines what the users the need is and how the developers can develop this project.

The various methods are : Interviews, Brainstorming Sessions, Facilitated Application Specification Technique, Quality Function Deployment, Use Case Approach.

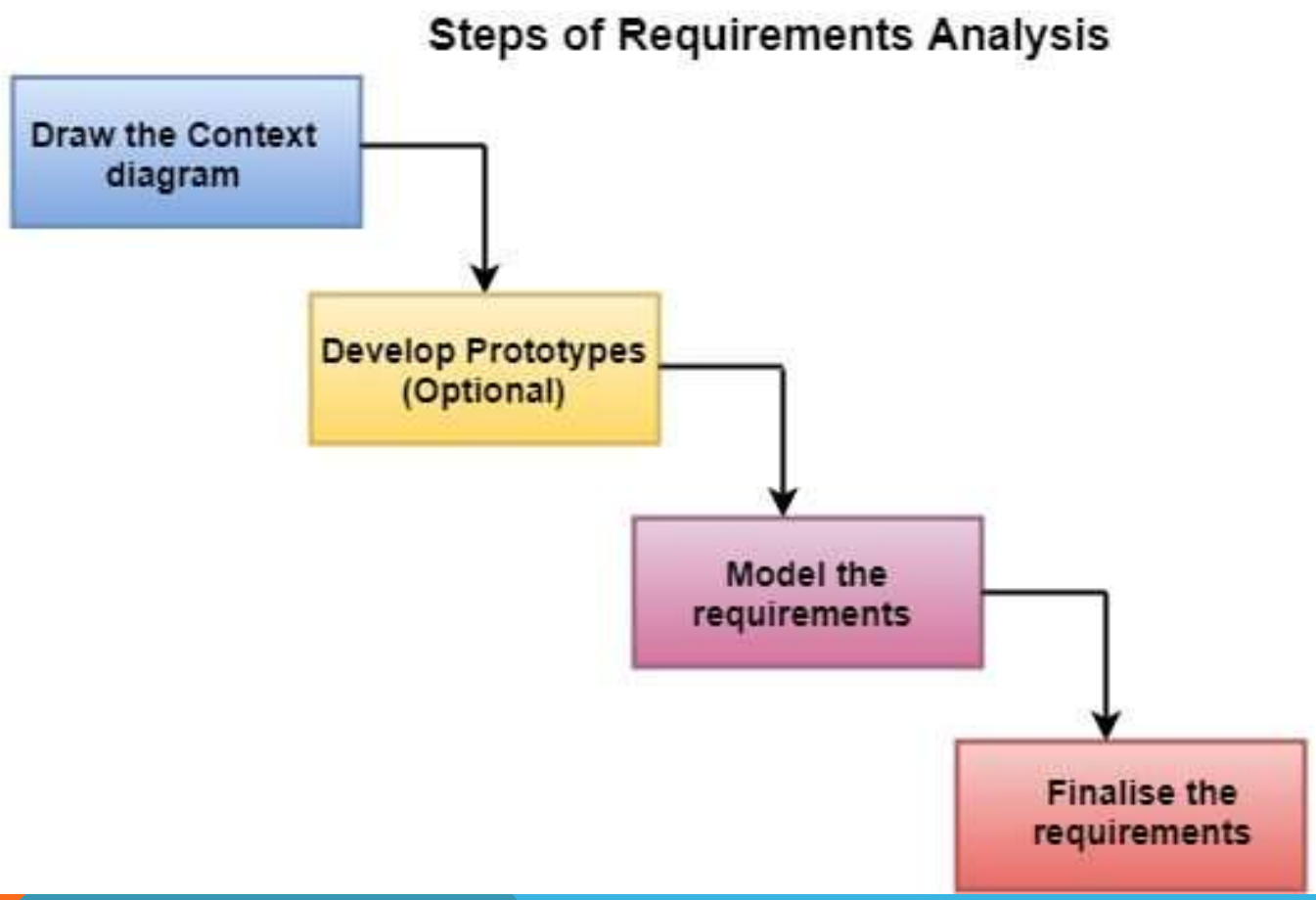
Requirement Analysis

It is the essential activity after elicitation . We analyze, refine and scrutinize the gathered requirements to make consistent and un-ambiguous requirements.

After the completion of the analysis, it is expected that the understandability of the project may improve significantly.

We may also use the interaction with the customer to clarify points of confusion and to understand which requirements are more important than others.

Requirement Analysis Steps



Requirements Validation

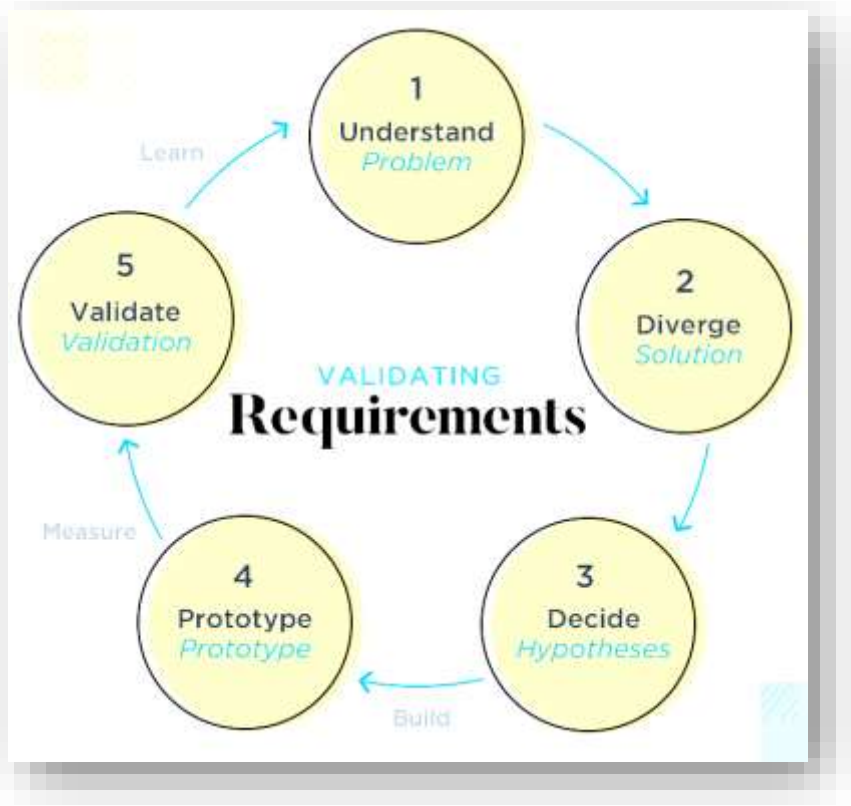
Process of Checking that requirements defined for development, define the system that the customer really wants.

To Check issues related to requirements , we perform requirements validation.

Use this validation step to check error at the initial phase of development as the error may increase excessive rework when detected later.

These checks include: Completeness, Consistency, Validity, Realism, Ambiguity, Verifiability.

Requirements Validation



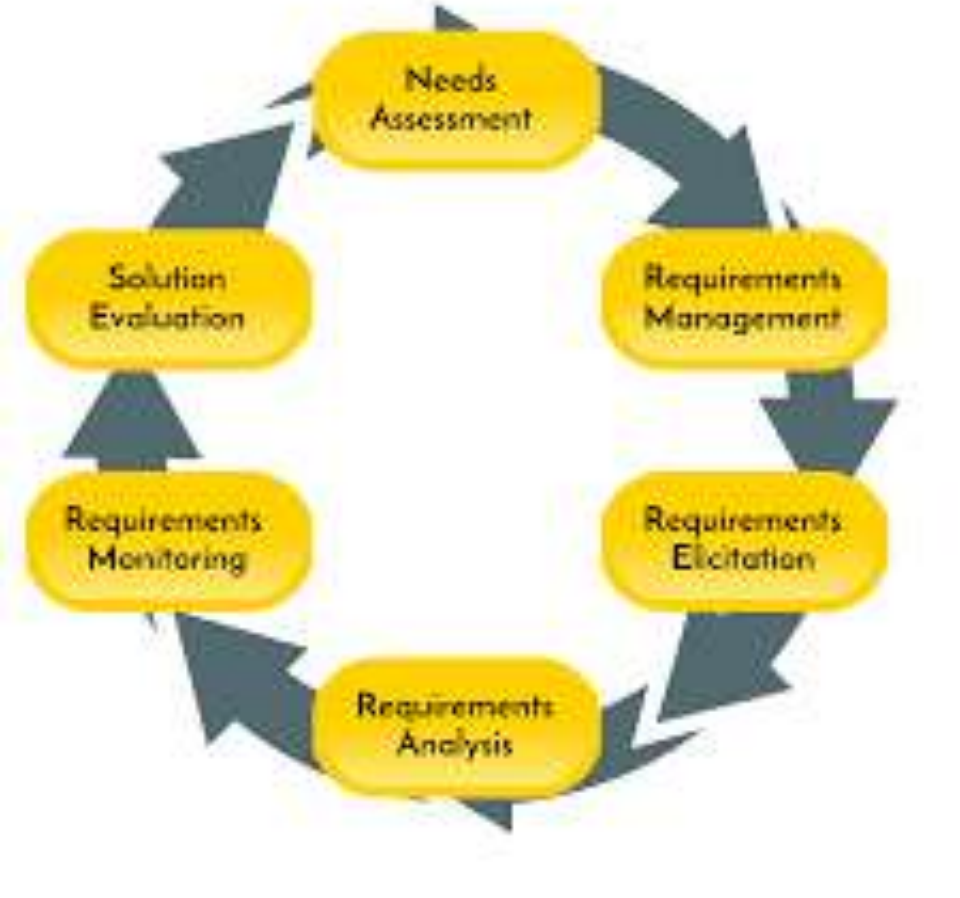
Requirements management

Requirement management is the process of analyzing , documenting tracking prioritizing and agreeing on the requirement and controlling the communication to relevant stakeholders.

This stake takes care of the changing nature of requirements.

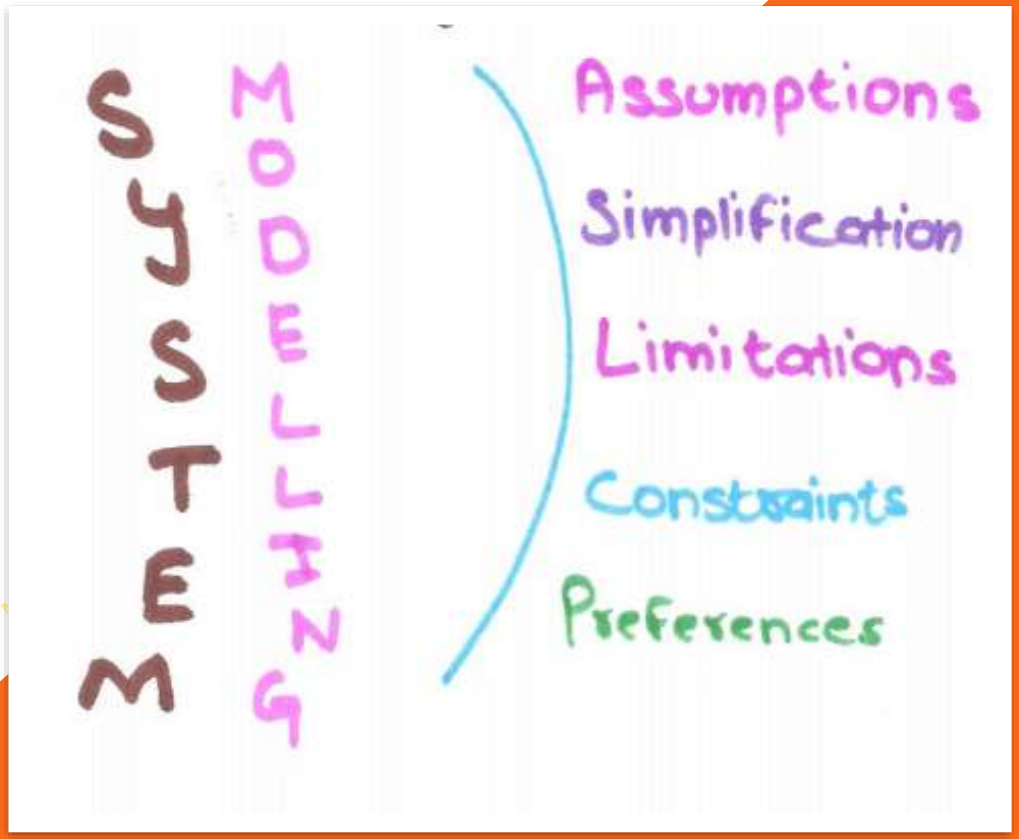
Should be ensured that SRS is as modifiable as possible so as to incorporate changes in requirements specified in a systematic and controlled manner is an important part of the requirements process.

Requirements Management



Chapter 2

PART-III: SYSTEM



SYSTEM MODEL & SYSTEM MODELLING

The interdisciplinary study of the use of models to conceptualize and construct systems in businesses and IT development.

System Modelling helps the analyst to understand the functionality of the system and models are used to communicate with customers.

For System Modelling we use the concept of data models to understand.

We have the following models : Context, Behavioral, Data, Object and Structured.

CONTEXT MODEL

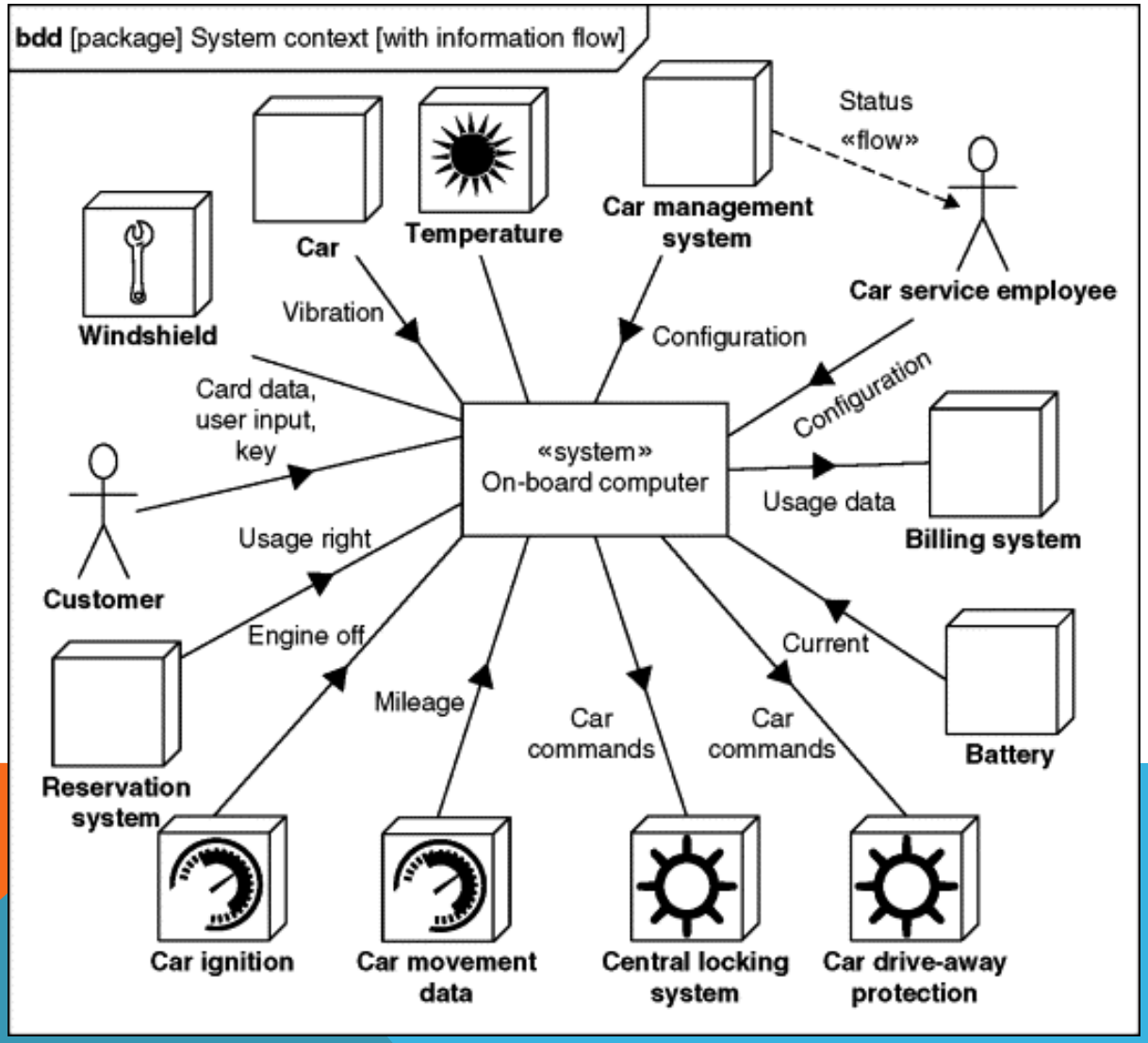
Defines how context data are structured and maintained.

A key role of context model is to simplify and introduce greater structure into the task of developing context aware applications.

A key role of developing a context model is to simplify and introduce a greater structure into the task of developing a context aware applications.

Best Example: Unified Modelling Language as used in systems engineering defines a context model as the physical scope of the system being designed.

CONTEXT MODEL

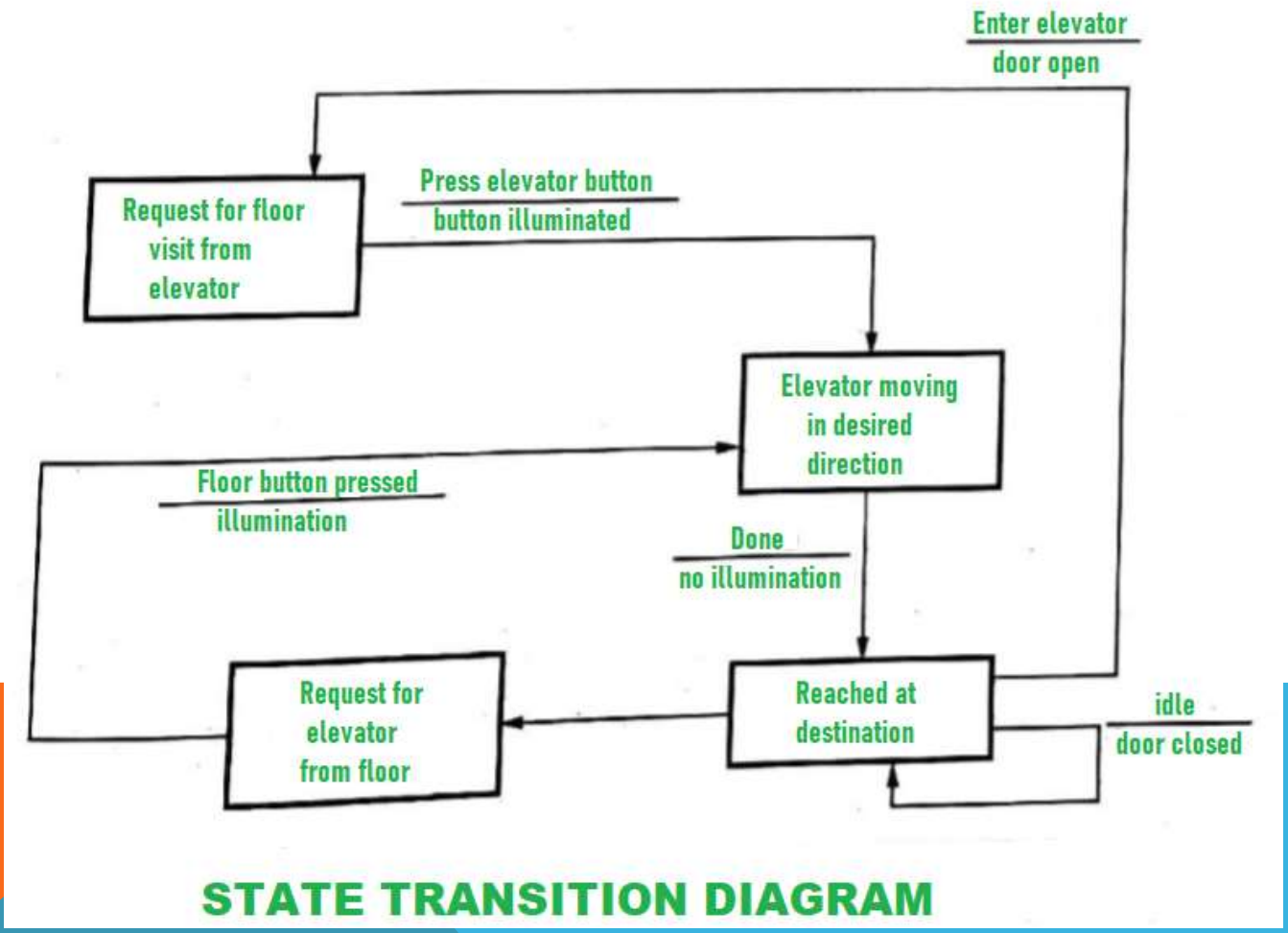


Behavioral Model

Behavioral Model is specially designed to make us understand behavior and factors that influence behavior of a System. Behavior of a system is explained and represented with the help of a diagram. This diagram is known as State Transition Diagram. It is a collection of states and events. It usually describes overall states that a system can have and events which are responsible for a change in state of a system.

So, on some occurrence of a particular event, an action is taken and what action needs to be taken is represented by State Transition Diagram.

Behavioral Model

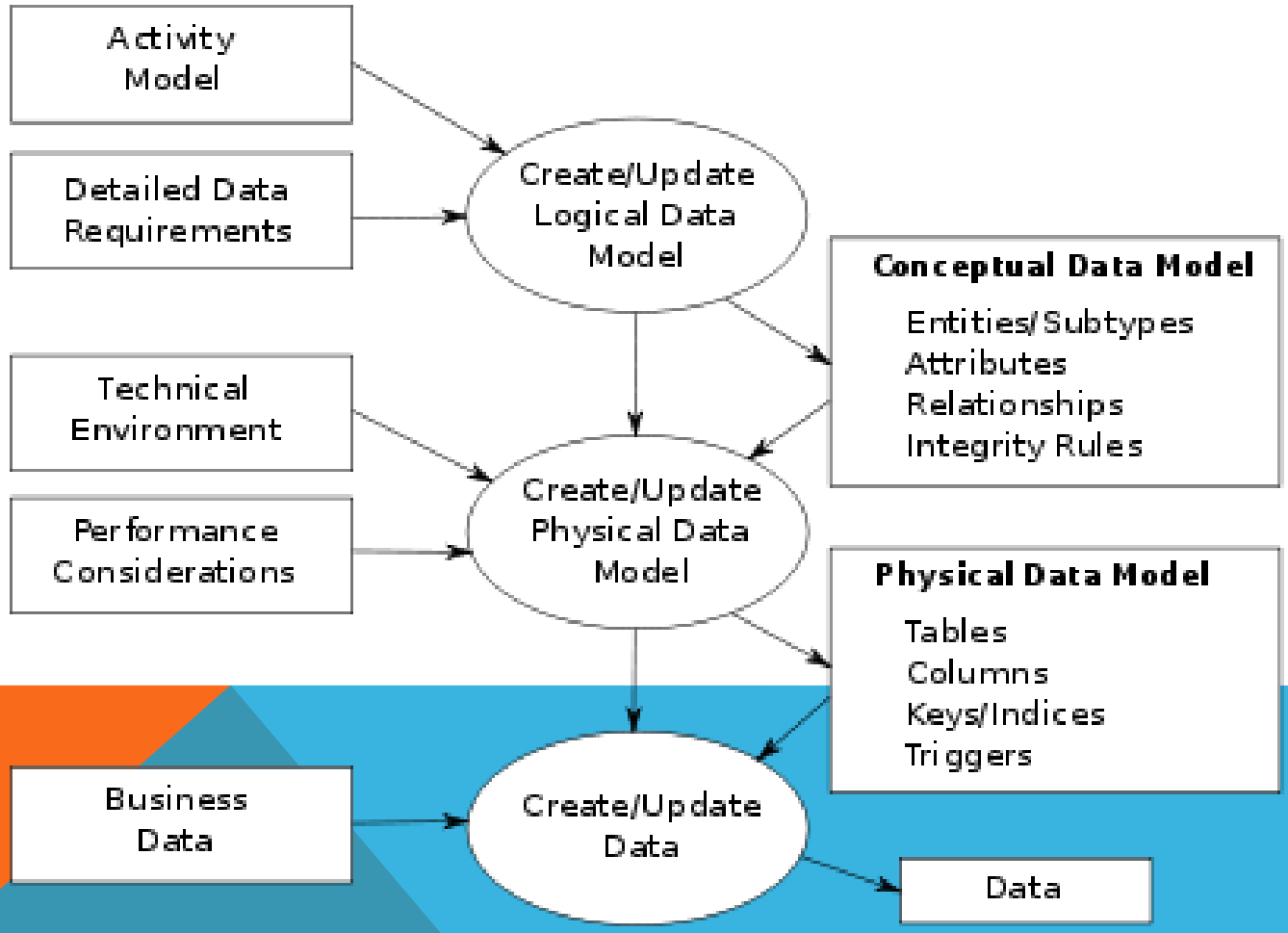


DATA MODEL

Data modeling in software engineering is the process of creating a data model by applying formal data model descriptions using data modeling techniques. Data modeling is a technique for defining business requirements for a database.

The goal is to create a visual data map that accurately describes the data structure, how data will flow through the system whilst highlighting important data relationships. This can involve the data input itself, the data infrastructure and output, whether that's predictive models, ML algorithms, AI or other products/services.

DATA MODEL


Activity Model
Detailed Data Requirements
Technical Environment
Performance Considerations
Business Data
Create/Update Logical Data Model
Create/Update Physical Data Model
Create/Update Data
Conceptual Data Model
Entities/Subtypes
Attributes
Relationships
Integrity Rules
Physical Data Model
Tables
Columns
Keys/Indices
Triggers
Data

OBJECT MODEL

Object Modeling Technique (OMT) is real world based modeling approach for software modeling and designing.

It was developed basically as a method to develop object-oriented systems and to support object-oriented programming.

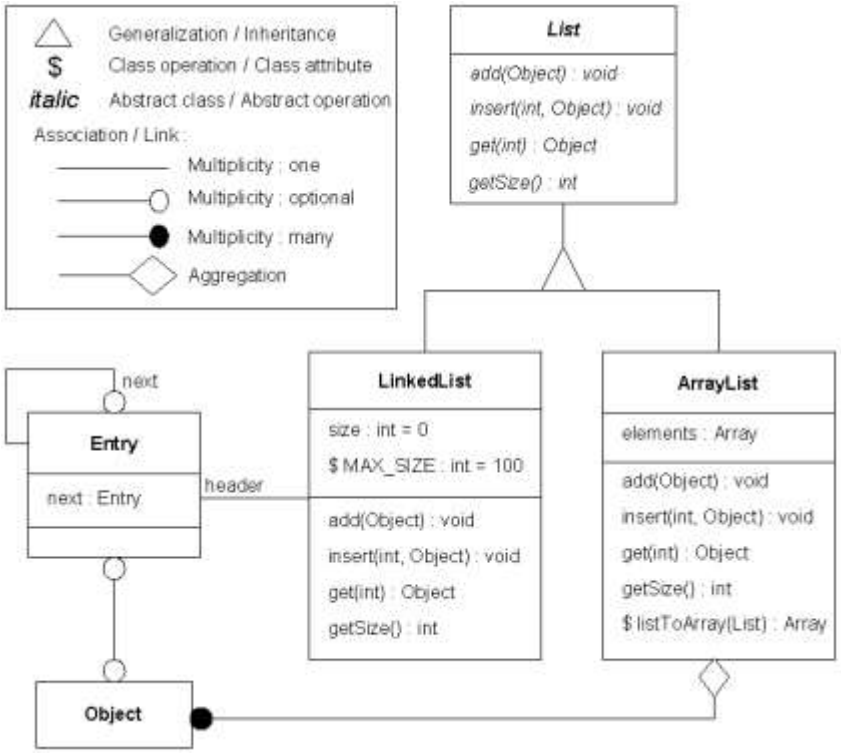
It describes the static structure of the system.

Object Modeling Technique is easy to draw and use

OMT is one of the most popular object oriented development techniques used now-a-days.

OMT was developed by *James Rambaugh*.

OBJECT MODEL



STRUCTURED MODEL

Structural models of software display the organization of a system in terms of the components that make up that system and their relationships.

Structural models may be static models, which show the structure of the system design, or dynamic models, which show the organization of the system when it is executing.

Structural models show the organization and architecture of a system.

Class diagrams are used to define the static structure of classes in a system and their associations.

STRUCTURED MODEL