Introduction to Software Engineering

Day 2

Anjali Kulkarni

May 2024

Agenda

- Understand UML
- UseCase Design
- DFD
- ER Diagrams
- Class Diagrams
- WireFrames

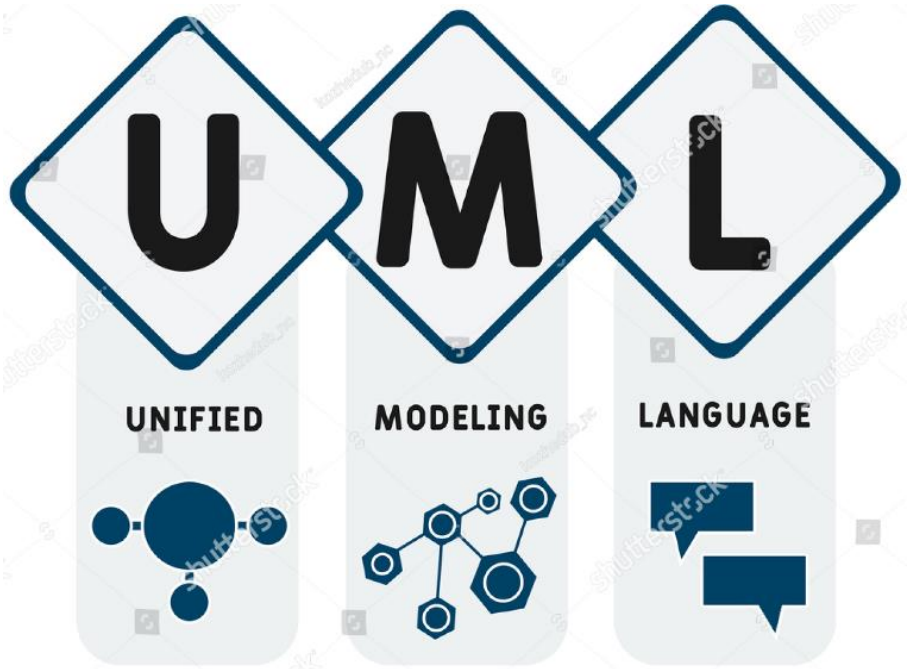
Requirements Document

- Describes user roles and their interactions with the system
- Identifies User Needs
- Mapping Needs to Functionalities
- User-Centric Focus

UML - Unified Modeling Language

Notation for modeling for OO Systems

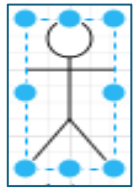
- Actors
- Entities
- Attributes
- Relationships



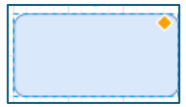
UML - Purpose

- Common notation keeps everyone on the same page
- Turns complex code into a visual diagram
- Easy to understand complex ideas
- Allows dev team to see big picture
- Makes requirements unambiguous

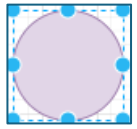
UML - Notations



Actor



Activity / Entity

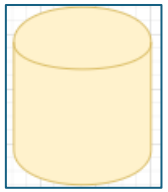


Process

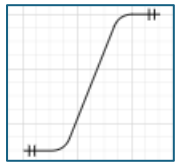
Decision Box

Bidirectional Arrow

Directional Arrow

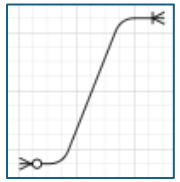


Database Object

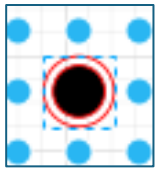


1 to 1 relationship

1 to Many relationship



Many to Many Relationship



End

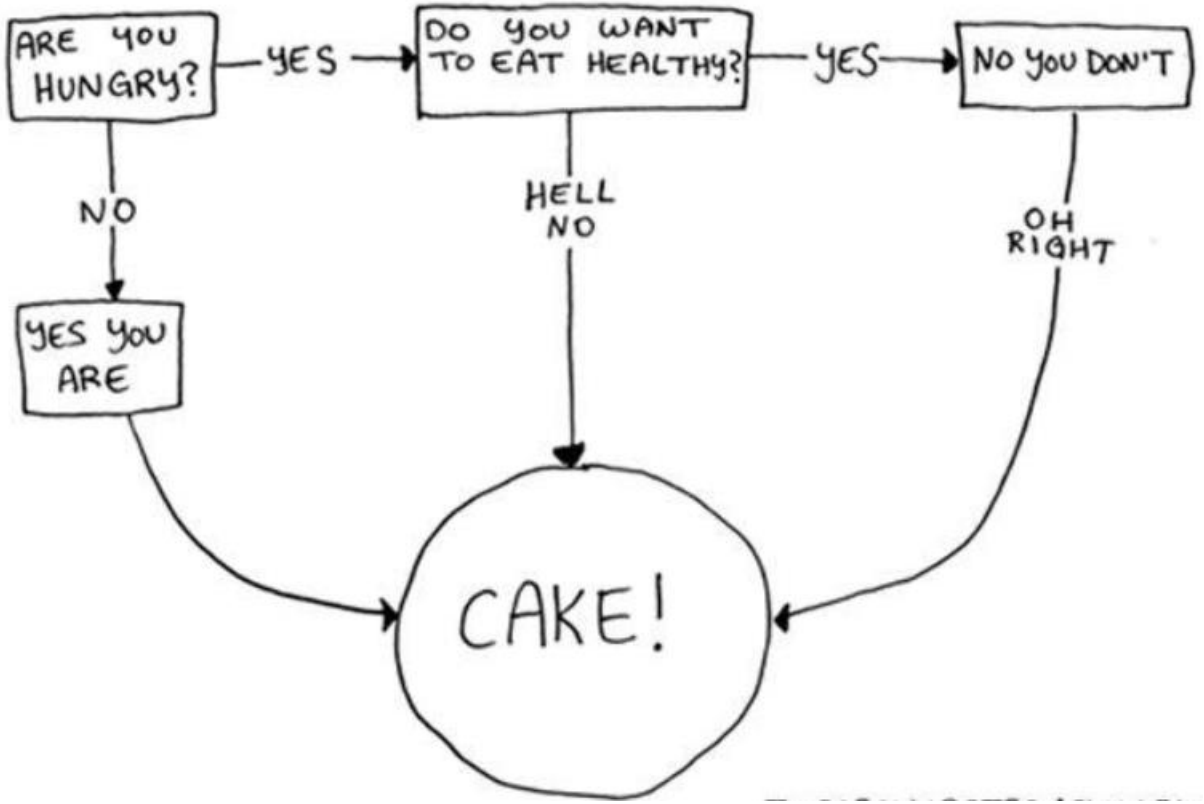
Use Case Design – Identify Use cases

Use Cases Format

“As A ….. I want To ……. So That I Can …..”

Use Case Diagram – Hands On

- Checkout Process Details
  - Add Shipping Address
  - Add Payment Method
  - Review Order Summary



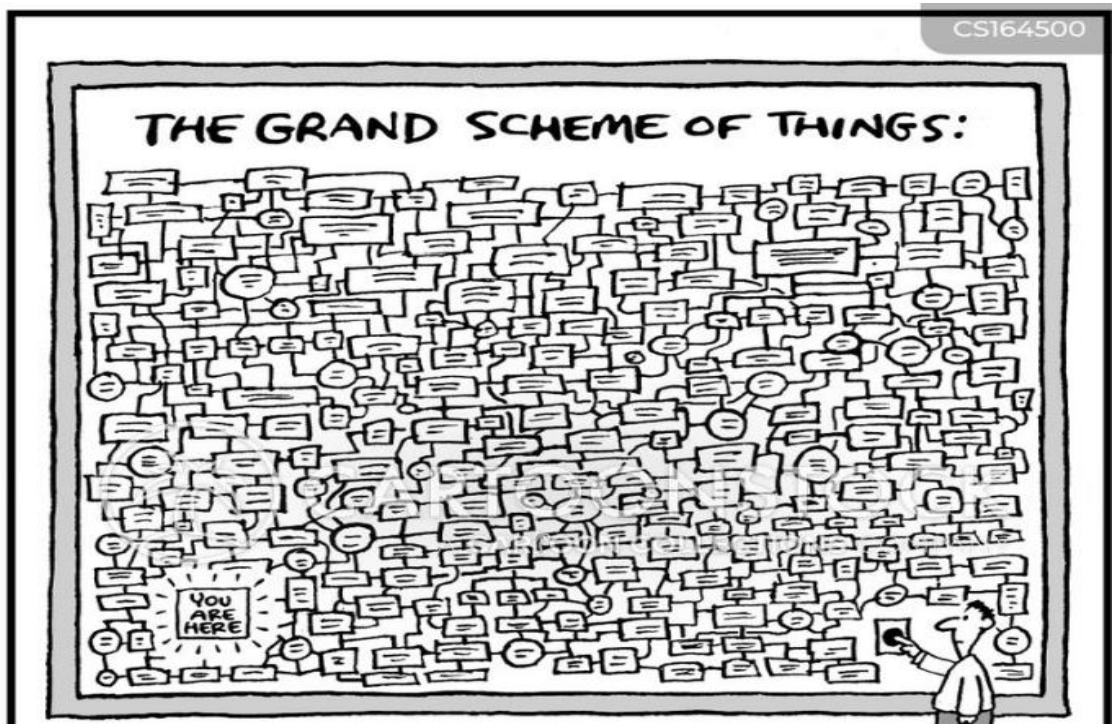
Data Flow Diagrams

Graphical representation of flow of data through systems

- Processes
- Data Stores
- Data Flows
- External Entities

DFD - Levels

- Level 0 (Context Diagram) – High level
- Level 1 - Detailed
- Level 2 and beyond



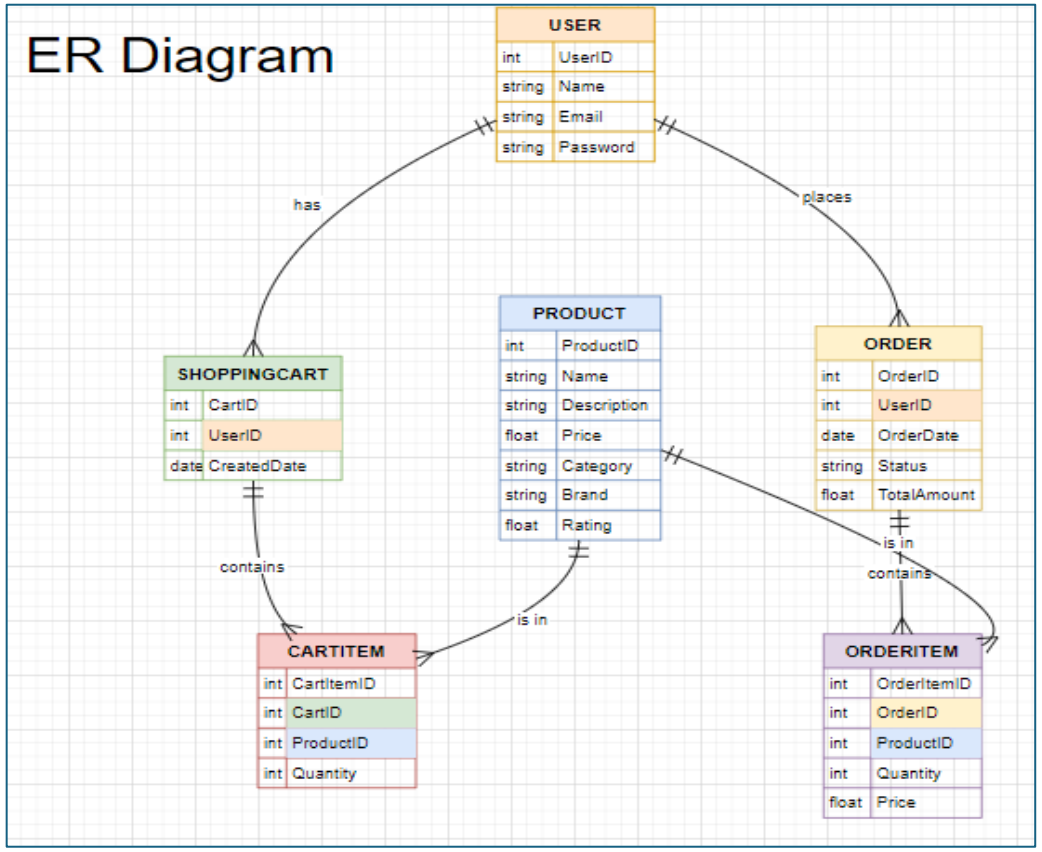
DFD – Hands On

- Add User Database
  - Request from database
    - User details
    - User Authentication details
- Send to System
  - User Details
  - Valid user confirmation

Entity Relationship Diagrams

Graphical representation of entities and their relationships

- Database Design
- Data Modeling



ER Diagrams – Key Components

- **Entities**: Represent real-world objects or concepts (rectangles).
- **Attributes**: Characteristics or properties of entities (ovals).
- **Relationships**: Define how entities interact (diamonds).
- **Primary Keys**: Unique identifiers for entities (underlined text).
- **Foreign Keys**: Attributes that create a link between entities.

Identifying Entities

Entity represents a real-world object or concept that can have data stored about it

- Uniqueness
- Attributes
- Relevance
- Persistence

Identifying relationships

- “can” – 1 to many
- “contains” – 1 to many
- “have” – 1 to many
- “references” – many to many

ER Diagram – Hands On

- Identify Entities and their relationships

Class Diagrams

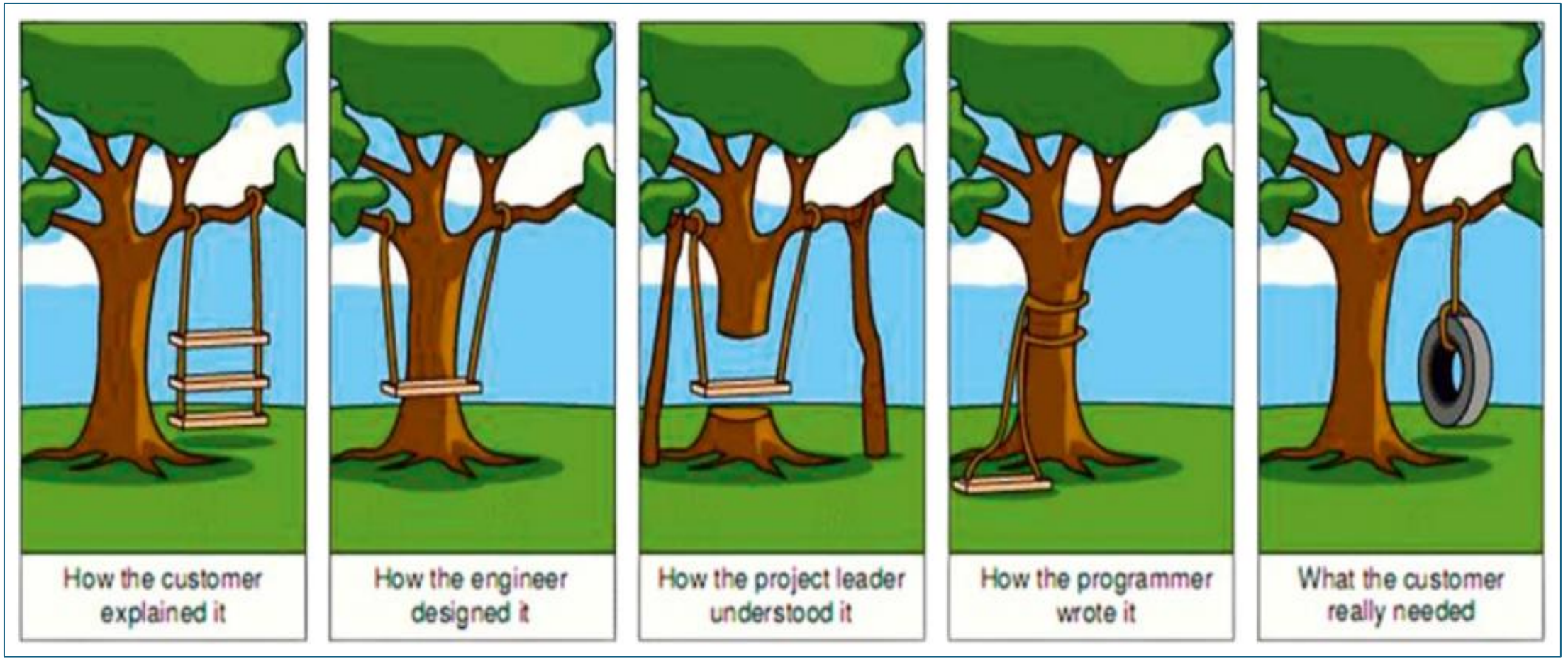
<u>Describes structure of system by showing classes, attributes, operations, relationships</u>

- Classes
- Attributes
- Methods
- Relationships
  - Associations
  - Inheritance
  - Aggregation/Composition
  - Multiplicity

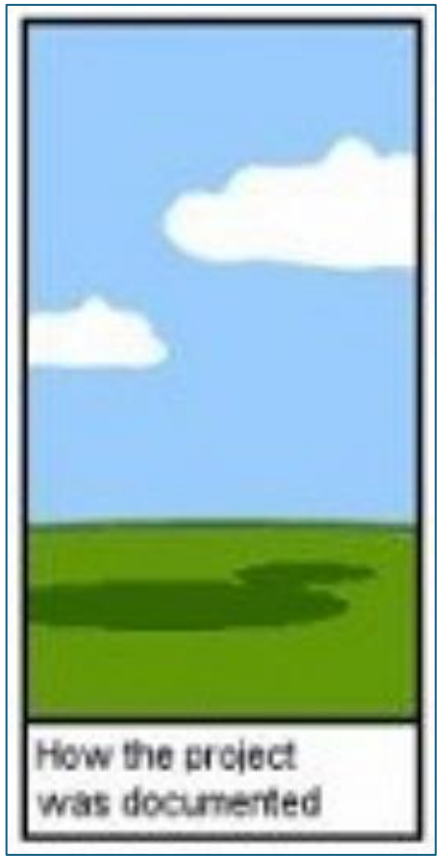
UML Modelling - Benefits

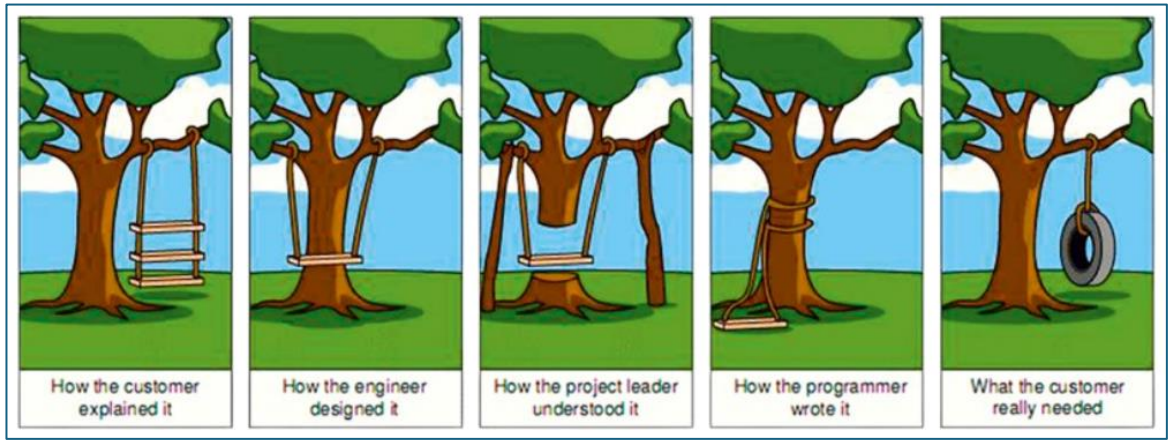
- **Clarity**: Provides a clear and concise view of the system structure.
- **Design**: Helps in the logical design of the system.
- **Documentation**: Serves as documentation for workflow, entities and relationships.

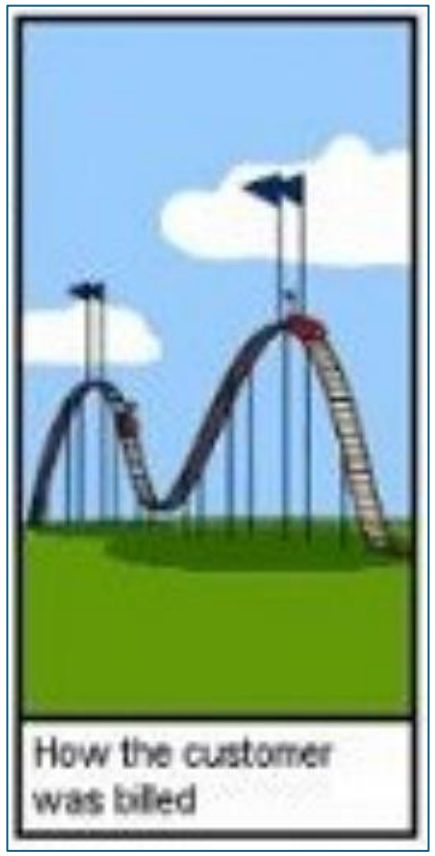
Tree Swing analogy!



What about customers?







Thank You !!