SAD

Lecture 2

Eng. Joud Khattab

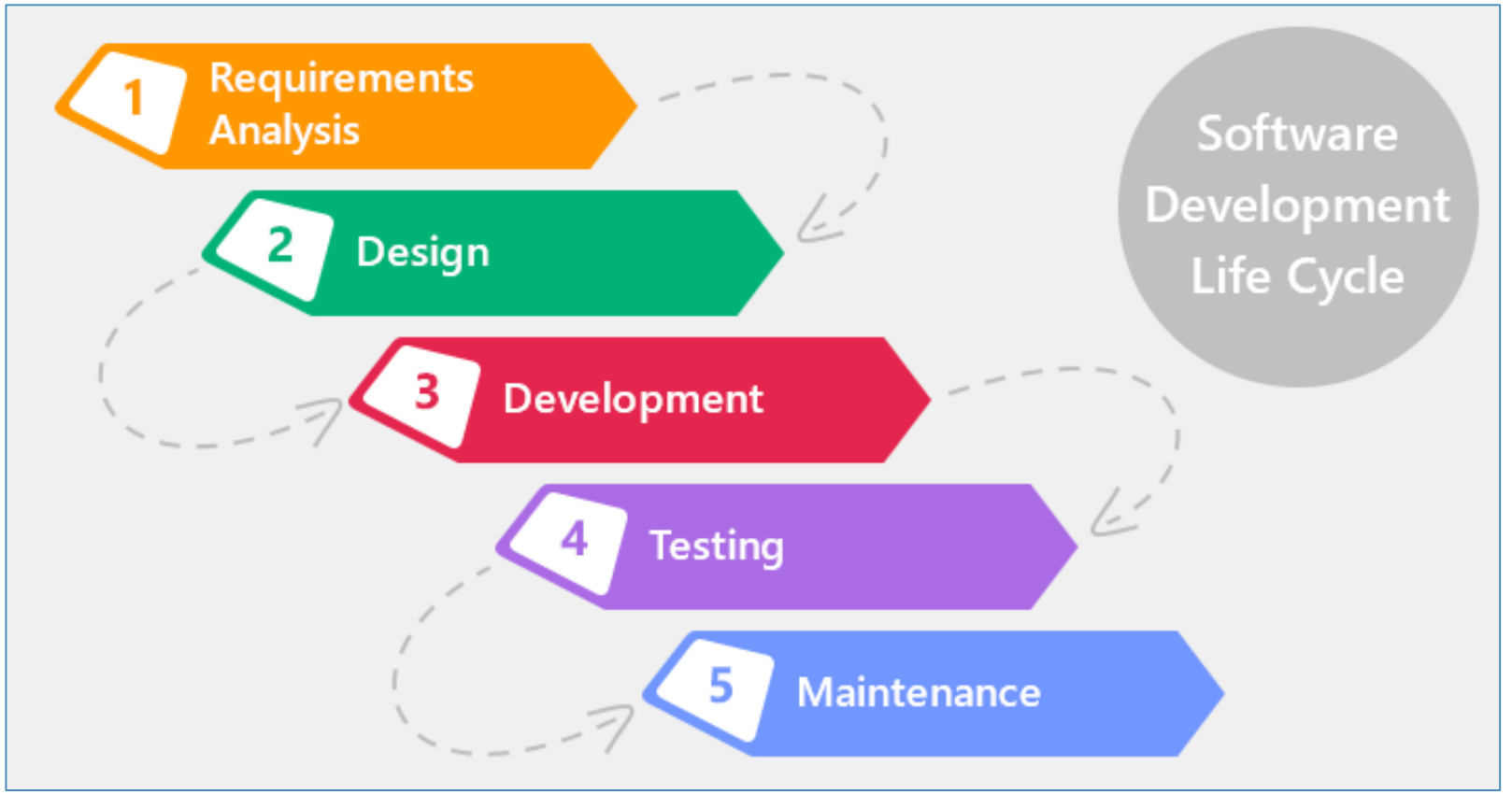
Overview

- Software Processes
- Software Development Life Cycle (SDLC)
- Requirement Engineering
- Stakeholders
- Unified Modeling Language (UML)
- Use Case Diagram
- Use Case Specification

Software Definition

- A software is the programs with data and documentation, and we do not mean here to document the comments between the lines of the program.
- Documenting all the steps we are taking to develop any software system at all stages of analysis, design, implementation and testing.

Software Development Life Cycle (SDLC)



Requirement Engineering

Functional Requirements	• Describing the behavior of the system
Non Functional Requirements	• Elaborates a performance characteristic

Requirement Engineering
Non Functional Requirements

- Security:
  - We mean that users or anyone should not access information within the system that they do not have access to.
- Availability:
  - If we develop a web system and every few minutes a message tells us that the server is unavailable, then our system is bad and is not permanently available to users.
- Reliability:
  - There are a lot of applications that interest us in reliability such as an application to transfer money between two accounts.

Requirement Engineering
Non Functional Requirements

- Safety:
  - Such as the system of automatic control of the aircraft or so-called autopilot and this system must be present in an aircraft when any emergency happens on the pilot and must have the necessary protection to preserve human lives.
  - These systems are called real time software or real-time systems.
- Efficiency:
  - We say software is effective if you use fewer resources than CPU and memory.
- Acceptance:
  - Acceptance criteria for a software system differ from one company to another and from one user to another. The basic criterion for accepting a software system is interfaces.

Stakeholders

- They are persons or institutions interested in developing software systems because they affect or are affected directly or indirectly by the proposed system.
- Example:
  - Education system for the Ministry of Higher Education or online management system.
  - Both the ministry and the students are the stakeholders because they are affected by this system and have a key role in developing this system.
- stakeholders kinds:
  - Developers who develop the system
    - Where developers are interested because they are affected in one way or another by the system.
  - Users:
    - They are stakeholders because they are using the system.
  - Sponsors:
    - Officials who finance the system. They are the ones who pay the money.

Stakeholders VS Actors

- The Stakeholders are the ones who are interested in the project and are the ones who use the system and are they are always just people.
- The actor is the one who use/activate the system or use the service of the system.
  - Not necessarily a person where the actor may be a person, hardware, software, and can be anything but the system itself.

Practical Exercise

- Actors Identification
- Requirement Gathering

Practical Exercise Actors

1. Primary Actors:
    1. Admin (Many Admins)
    2. Customer
    3. Restaurant Manager
    4. Restaurant Employee
    5. Driver
2. Secondary Actors:
    1. Call Center
    2. Billing System
    3. Notification System

Practical Exercise Requirements

- Update menu (add/remove/edit)
- Approve order
- Add location
- Payment
- Login/logout
- Register
- Offers/vouchers
- Make order (cart)
- View restaurant list
- Order delivery
- Track order
- Search
- Filter
- Ban customer
- Calculate cost
- Assign best restaurant
- Recommended cart
- View order history
- Rate restaurant
- Rate app
- Feedback
- Dashboard
- Current orders
- Ticket
- View menu
- View restaurant description
- View food specification
- Profile/edit
- Select service
- Add/remove/edit restaurant
- Add/remove/edit customer
- Add/remove/edit order

HOME WORK

Link the Actors with the requirements that we discussed in the class, with brief description of each service