श्रद्धावान् लभते ज्ञानम्

# 15CSE202 Object Oriented Programming
# Lecture 4

## OO Analysis with Use case

Nalinadevi Kadiresan

CSE Dept.

Amrita School of Engg.

# Object Oriented Development

- Phases
  - **Object-Oriented Analysis: understand the domain**
    - Define an object-based model of it
  - **Object-Oriented Design: Define an implementation**
    - Design the solution
  - **Object-Oriented Programming: Build it**

# Object Oriented Analysis

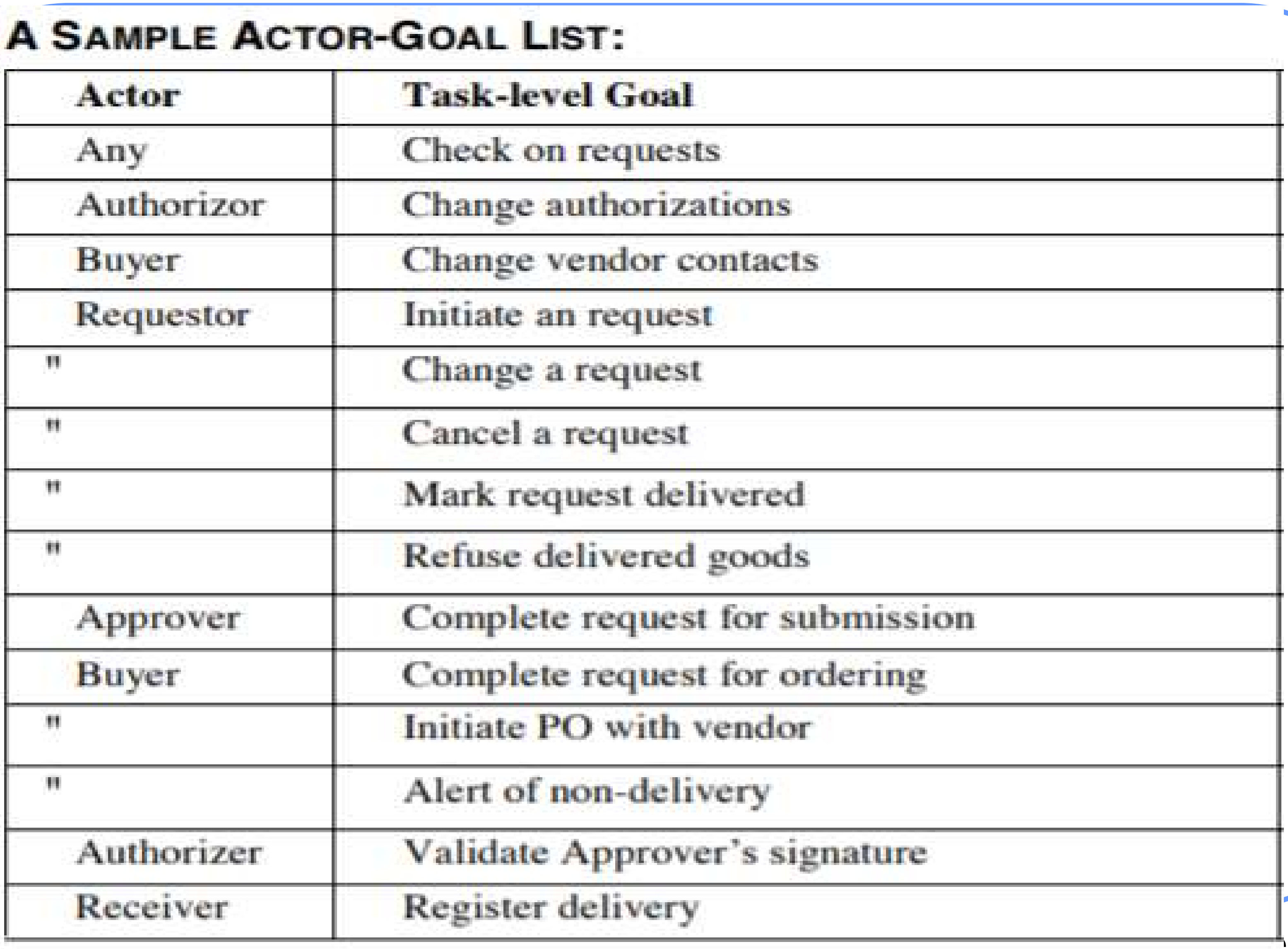
- Object-oriented analysis is a method of analysis that examines requirements from the perspective of the classes and objects found in the vocabulary of the problem domain
- Steps in examining requirements are:
  1. Use Case Document preparation
  2. Object Model preparation

# What is Use case?

- A use case is a narrative description of a goal-oriented interaction between the system under development and an external agent.
- A use case is sort of like a dialog script written for two actors. Use cases capture requirements in the form of interactions with an end user.
- When requirements are described in terms of interactions with end users missing details become obvious.

# Identifying use case

- Functional requirements are used to identify use cases.
- The tools to identify use cases are:
  - Goals of the Actor



## A Sample Actor-Goal List:

| Actor | Task-level Goal |
|---|---|
| Any | Check on requests |
| Authorizor | Change authorizations |
| Buyer | Change vendor contacts |
| Requestor | Initiate an request |
| " | Change a request |
| " | Cancel a request |
| " | Mark request delivered |
| " | Refuse delivered goods |
| Approver | Complete request for submission |
| Buyer | Complete request for ordering |
| " | Initiate PO with vendor |
| " | Alert of non-delivery |
| Authorizer | Validate Approver's signature |
| Receiver | Register delivery |

# History of Object Oriented Software Engineering

- use cases was pioneered by Ivar Jacobson in 1995
- The use cases are an important component of Jacobson's Objectory methodology and its simplified OOSE

# Use Cases Do Not Require Technical Knowledge

# An example – Home security system

| Use Case Title: | Deactivate Alarm |
|---|---|
| Actors: | Home owner |
| Description: | This use case begins when the home owner enters the home and the alarm sounds. The home owner enters a preassigned code on the keypad to deactivate the alarm. If the correct code is entered within 30 seconds the alarm is deactivated, otherwise a security alert is sent to the local police station. |

# Main sections of Use case document

- Use Case Name
- Description
- Actors
- Trigger
- Preconditions
- Flow – Basic Flow, Alternates and Exceptions
- Level
- Post condition
- Stakeholders

# Create Unique and Explanatory Names

- Use Case naming is usually done based on an organization’s data standards.
- If your organization has already been using Use Cases, ensure that you name your Use Case using the same terminology as the other use cases.
- Use cases need to be searchable and they need to be easily available when needed.
- Some companies simply assign numerical IDs to ensure that use cases are easily indexed and maintained.

# Actors

- The actors in the use case are the people or elements who are involved in the process
- Primary Actor
  - person who is responsible for the event for which the Use Case exists
- Secondary Actor
  - person or group of people that is needed to complete the process successfully

# Trigger

- A trigger simply defines the exact action which results in the Use Case
- Types of Trigger:
  - External events are those started by an actor, either a person or another system requesting information, such as an airline reservation system requesting flight information from an airline system.
  - Temporal events are those that are triggered or started by time. Events occur at a specific time, such as sending an email about special offers once a week on a Sunday evening

# Precondition

- The preconditions are the conditions that need to be met to ensure that the use case can be fulfilled. If these conditions are not met then the case cannot run its course.

# Flow

- You need to define the flow of the process that starts when a use case is started.
- The flow needs to detail how the communication will flow, who the information will be displayed to, what they need to do, and where the primary actor will end up.

# Flow - contd

- 3 things which you need to mention when writing the flow.
  - Basic Flow: best case scenario of what should happen in the use case if all the conditions are met
  - Alternates: Are there any alternate routes that the action can be done?
  - Exceptions: This dictates what happens when a failure occurs in the flow.
    - Exceptions are just as important to define as basic flow.

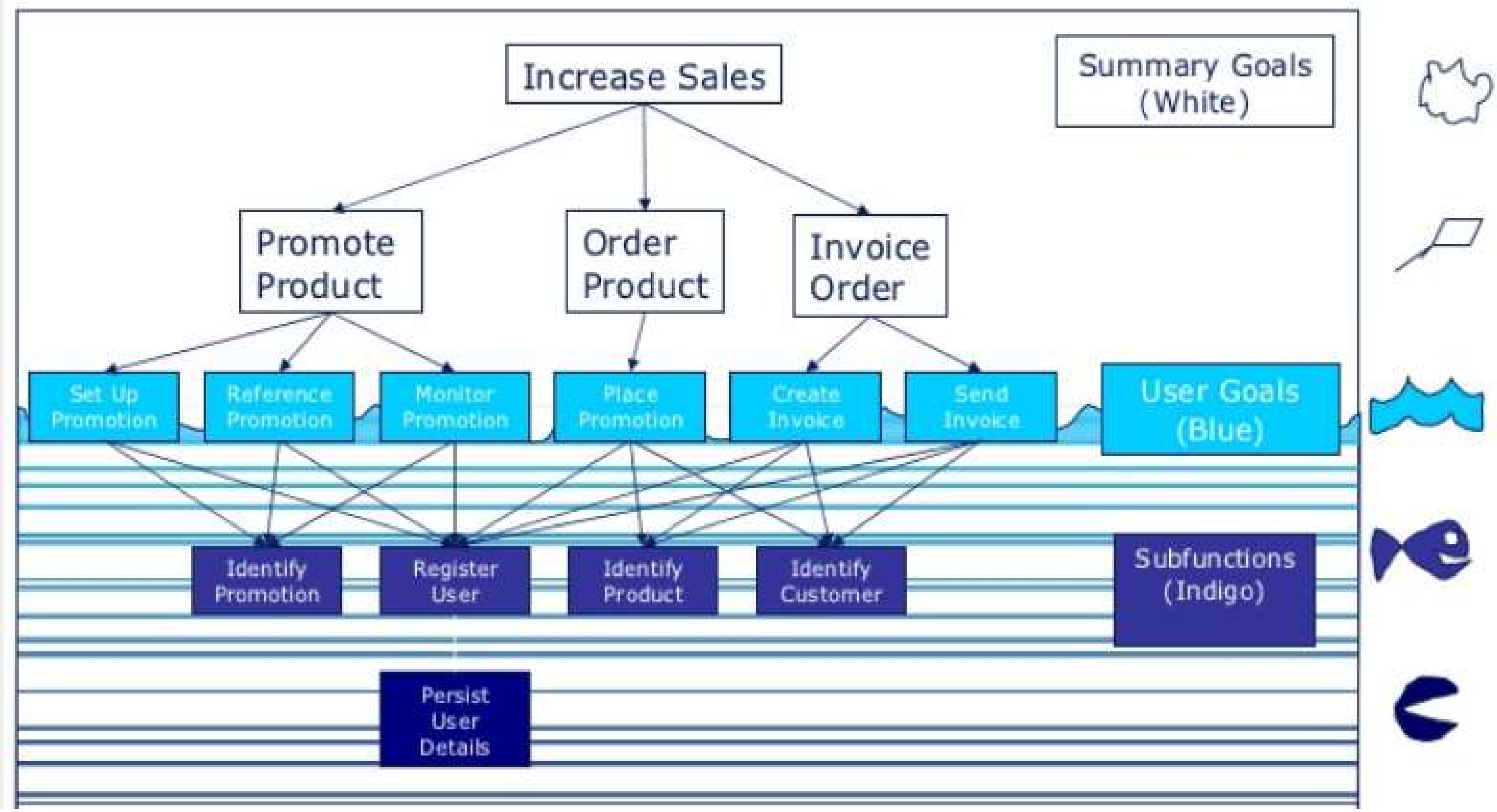
# Level

- It is important to classify the use case with a level in order to explain the urgency which it needs to be dealt with.
- You need to ensure that whoever reads the use case realizes its importance to ensure that they give it the right amount of time.

# Level

- Different levels of use-case:
  1. High Level Summary(white/cloud) -Enterprise level
  2. Summary of goals (Kite) - business unit or department level
  3. User Goal (Blue/ Sea level) - usually written for a business activity and each person should be able to do one blue level activity in anywhere from 2 to 20 minutes.
  4. Sub-Function (Indigo/ fish) - shows lots of detail
  5. Low Level (Black/calm) - most detailed use cases

# Level - Example



Increase Sales

Summary Goals (White)

Promote Product

Order Product

Invoice Order

Set Up Promotion

Reference Promotion

Monitor Promotion

Place Promotion

Create Invoice

Send Invoice

User Goals (Blue)

Identify Promotion

Register User

Identify Product

Identify Customer

Subfunctions (Indigo)

Persist User Details

# Postcondition

- It is the state of the system after the use case has finished, including output people have received, transmissions to other systems, and data that have been created or updated.

# Stakeholders

- Who are the people that are going to be affected by the use case?
- There’s no way to write a good process if you do now know who will be affected because only when you know all the people involved will you be able to ensure that your design does everything that it needs to do.

# Example-1 Coin Flip

**Use Case Name**: Coin Flip

**Description**: A player at random offers a prediction of coin flip. The other play gets the other option. The coin is flipped. The correct guess wins

**Actors**:

1.A player who makes the prediction
2.A player who gets the other option
3.Coin
4.Coin game

# Example-1 Coin Flip

**Trigger:** A player at random offers a prediction.

**Preconditions**:

1. 2 players are available
2. A coin available

**Basic Flow ("Sunny Day Scenario")**:

1. A player at random is picked to predict the coin flip.
2. A player picked offers the prediction for the coin flip.
3. The coin is flipped and the result is provided.
4. A winner and loser is picked.
5. Offer to try it again.

# Example-1 Coin Flip

**Alternates**: no

**Exceptions:** no failure condition.

**Post conditions**: one player wins and other loses

**Stakeholders: Nil**

# Example-2: Car Rental Application

**Use Case Name**: Release a Vehicle (to a Customer)

**Description**:
A customer arrives to acquire the vehicle and depart for desired destination. The vehicle reservation contract is signed and the vehicle is released to the customer.

**Actors**: Front-Desk Clerk, Customer
**Preconditions**: Vehicle has been assigned to the customer

**Basic Flow ("Sunny Day Scenario")**:

1. A customer comes to the office to acquire a vehicle.
2. The clerk locates the vehicle reservation contract by means of the reservation number and/or customer name. [Exception: Required vehicle type is not available due to late arrivals.]
3. The customer signs the contract and the clerk gives the keys to the vehicle.
4. The clerk then marks the contract active by entering the vehicle release date (today's date) onto the vehicle reservation contract. The use case terminates at this point.

- **Exceptions ("Rainy Day Scenario")**:

  - –Required vehicle type is not available due to late arrivals:
  - –Raised when the reserved vehicle is not available due to late returns. The customer is informed of the situation and told about the other vehicle types that are available. The customer is offered an incentive to accept another vehicle type. If the customer is not satisfied, the reservation is cancelled without penalty charges. The customer either accepts another vehicle type or cancels the reservation.

- **Postconditions**: The customer departs with the vehicle and the reservation contract is marked active, or the reservation is cancelled.

# Example-3: Online Support Request

- Use Case Name : Support Request
- Description: Suppose a person generates a support request on a website for electronic products.
- Actors
  - Person who generates support request by clicking button
  - Customer Support Representative

- Preconditions:
  - The person who generates the request needs to have an active internet connection
  - The website needs to be accessible by the customer
  - The support agent needs to have an active internet connection
- Basic Flow
  1. The customer visits the support website
  2. The customer clicks the “generate support ticket button”
  3. The customer is taken to a page where they are told that support will be present shortly.

4. The ticket is sent to the support department
5. A customer support representative takes hold of the ticket
6. The customer support representative is sent to the same page as the user where there is a chat box
7. The customer support representative provides the help and support to the customer
8. The customer closes the chat window
9. The customer support representative closes the support ticket and enters information about the session.
10. The use case ends

- Alternates:
  - The customer asks to be contacted over the phone
  - The customer might request someone come to their home to help them with the product
- Exception:
  - The customer loses their internet connection between the chat. They are then shown an error message.
  - The customer service representative accidentally closes the chat window, ending the support session. The customer is informed that another representative will be with them shortly.

- Level:
  - Customer complaints
  - General Feedback
- Trigger : customer clicks the button on the website to generate a support ticket
- Stakeholders:
  - The Customer Support Department – The whole department's purpose is to ensure customer satisfaction.
  - The Sales Department – If the customer is unable to get their problem fixed they might want to return the product, which will be a loss for the sales department.

# Things to remember

Reminder 1. A use case is a prose essay

Reminder 2. Make the use case easy to read (active voice).

Reminder 3. Just one sentence form

Reminder 4. Include sub use cases

Reminder 5. Keep the GUI out

Reminder 6. Mandatory two endings

# Reference book

- Alistair Cockburn, Writing Effective Use Cases, Addison-Wesley, 2001.

Next Session will be

Object Oriented Analysis with Domain Model