

A horizontal bar with a teal segment on the left and an orange segment on the right, located above the main title.

Pragmatic Product Leader Course

Become a Product Manager with Superpowers



LESSON 6A Introduction

A horizontal decorative bar with a teal segment on the left and an orange segment on the right.

6.0. Why model?



**Communicate effectively,
unambiguously is the most
important thing you will be doing**





2 types of models

Behavior Model or Dynamic model

Tries to answer - “What does the system do?”. It is verb-oriented. Tries to judge or analyse the system by its actions.

Eg. Use case diagrams, Decisions trees, activity diagrams etc.

Structural Model or Static model

Tries to answer - “What is the system?”. It is noun oriented. You want to know all the terms, jargons etc. that are used in the system

Eg. Class diagram, package diagram etc.



Is your modeling decision good?

You'll know you've gone in a good direction if your model has these qualities:

- **Elegance:** A simple solution to a complex problem
- **Adaptability:** Can easily be changed to reflect a change in the requirements
- **Non-redundancy:** Does not repeat itself - each fact is “in one place”

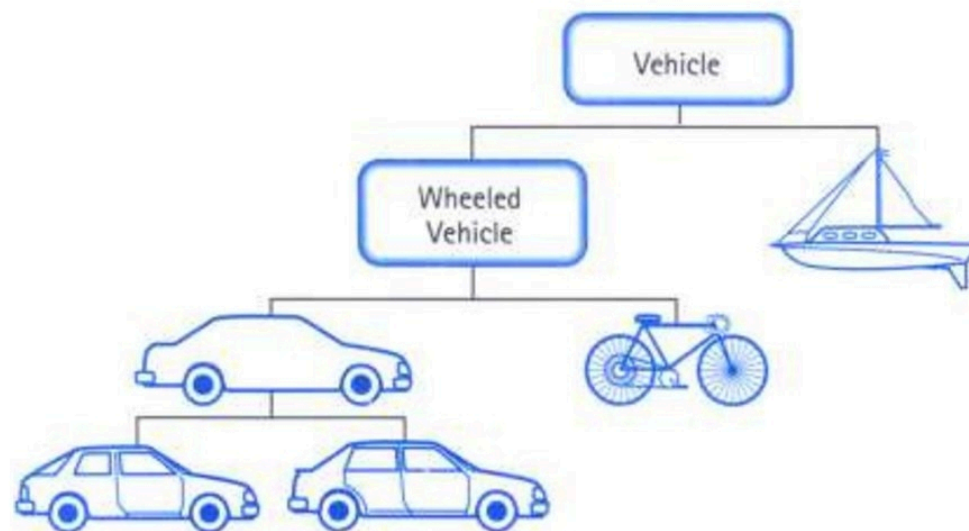
A horizontal decorative bar with a teal segment on the left and an orange segment on the right, positioned above the main title.

6.1. Object Orientation



What is OO?

- Acronym for “object-oriented”
- OO system consists of a set of objects that collaborate by sending requests to each other
- Conceptual framework that affects:
 - The way PM analyzes and models the requirements
 - The software engineer designs the system specifications
 - The way the code itself is structured



A horizontal decorative bar with a teal segment on the left and an orange segment on the right.

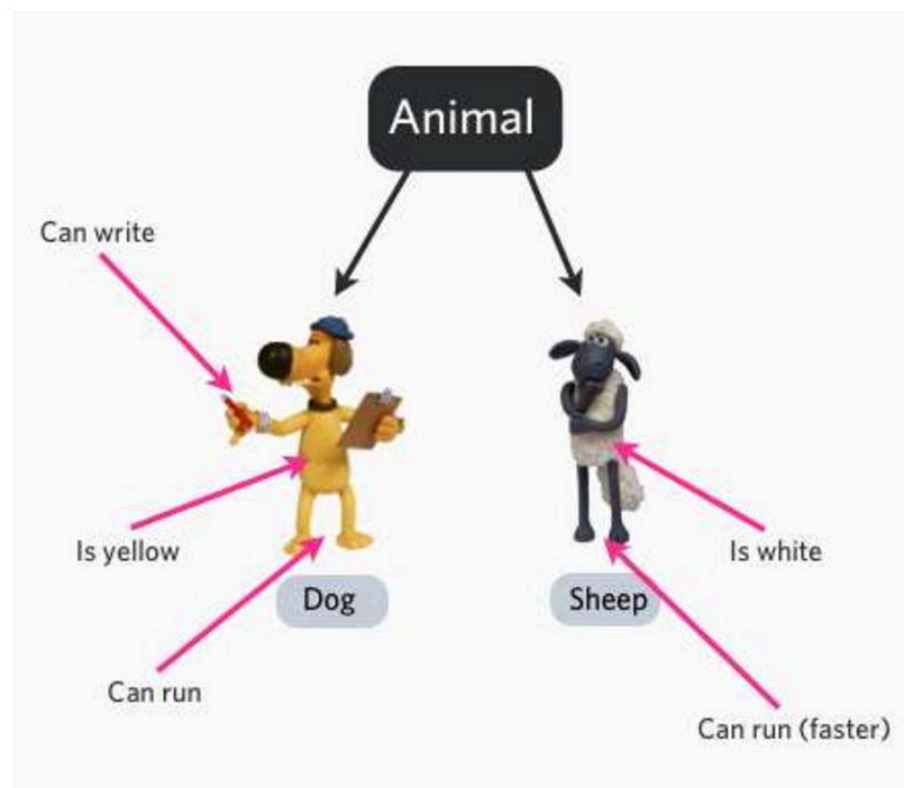
OO concepts: Objects

- Business objects
- Must be tracked or
- Participates in business process
- For eg. Invoice, call, credit note, user



OO concepts: Attributes and Operations

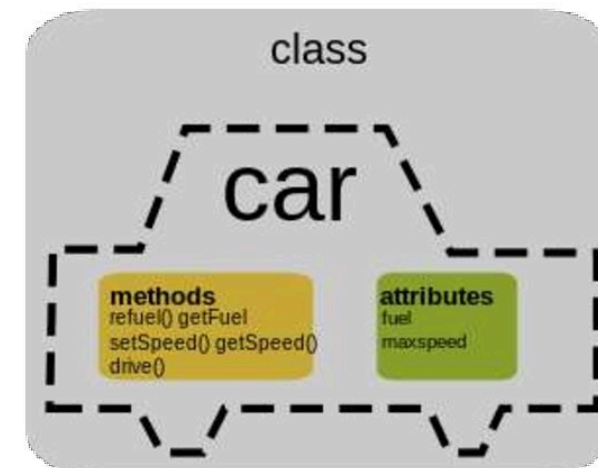
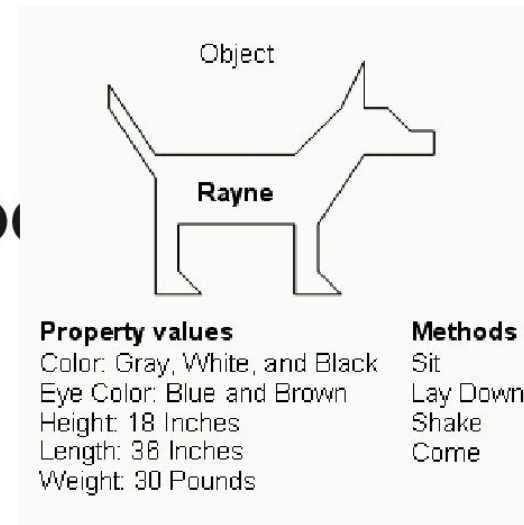
- Attributes represent the properties of objects and store the values
- Operations represent what we can do with the objects





OO concepts: Operations and Method

- How to work with Objects
- The procedure to follow
- The input and output of the procedure





OO concepts: Encapsulation

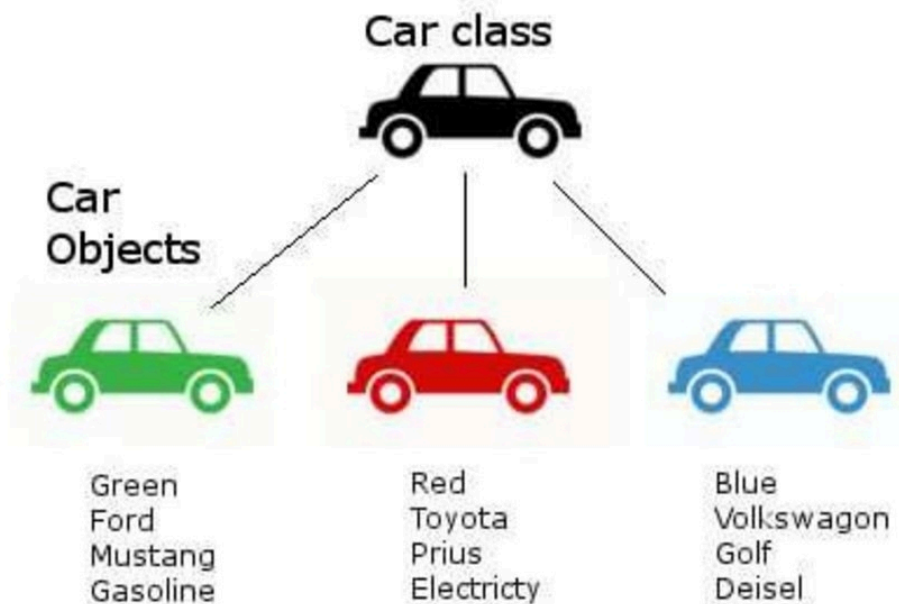
- Only object's operations are visible to other objects
- Attributes and methods remain hidden from view





OO concepts: Classes

- Attributes and operations are defined at the class level and apply to all objects within that class
- All objects in a class have the same attributes. This means that the same properties are significant for all objects in a class, though the value may change from object to object
- All objects in a class have the same operations and methods. In other words, all objects in a class can do the same things - and they do them the same way



A horizontal decorative bar with a teal segment on the left and an orange segment on the right.

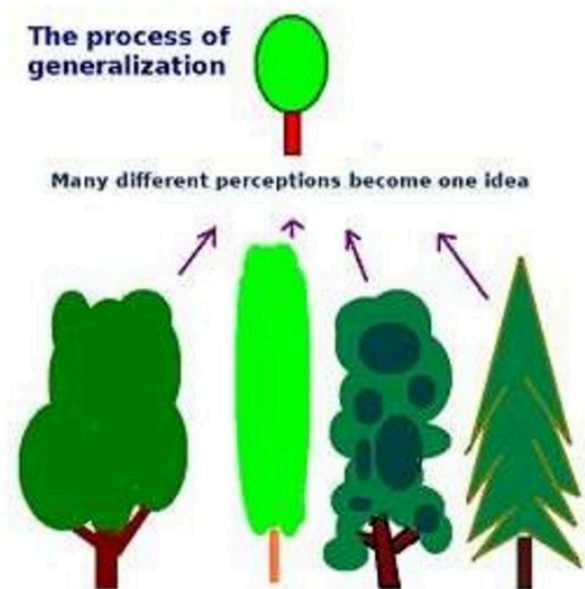
OO concepts: Relationships

- Generalization
- Association
- Aggregation
- Composite aggregation (composition)
- Polymorphism



OO concepts: Relationships

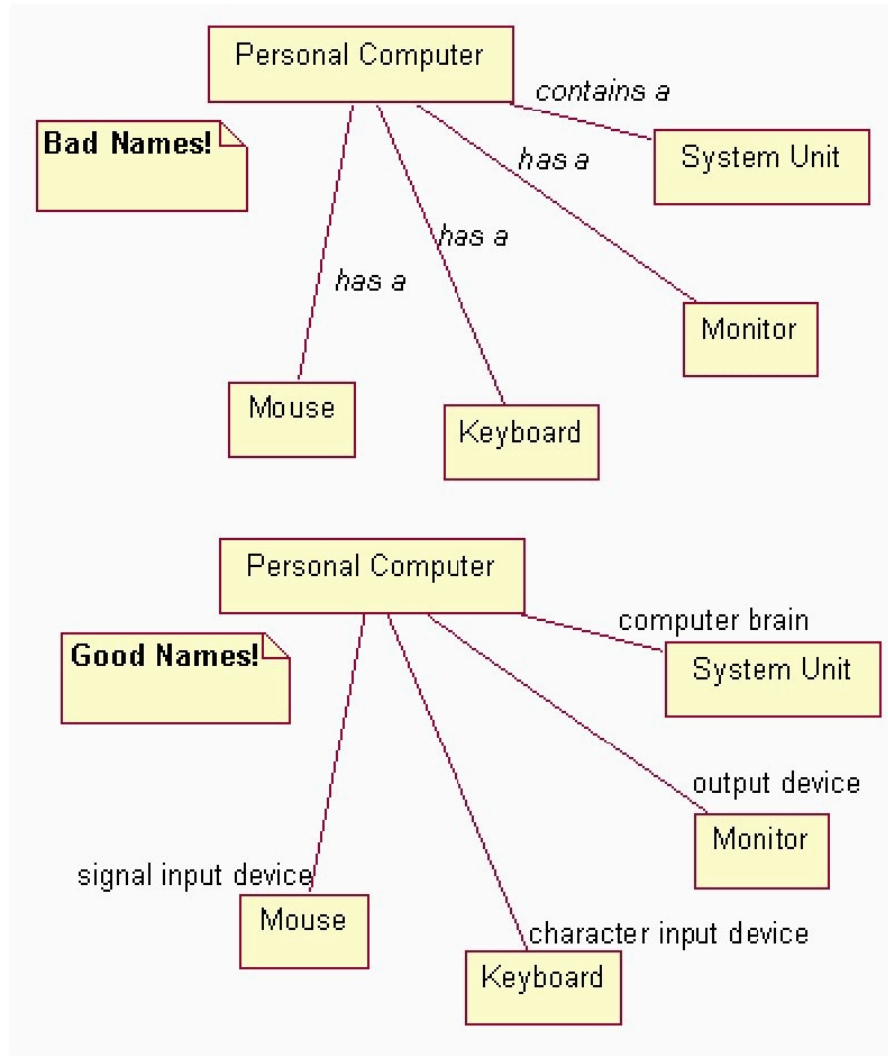
- Generalization
- Association
- Aggregation
- Composite aggregation (composition)
- Polymorphism





OO concepts: Relationships

- Generalization
- **Association**
- Aggregation
- Composite aggregation (composition)
- Polymorphism





OO concepts: Relationships

- Generalization
- Association
- Aggregation
- Composite aggregation (composition)
- Polymorphism

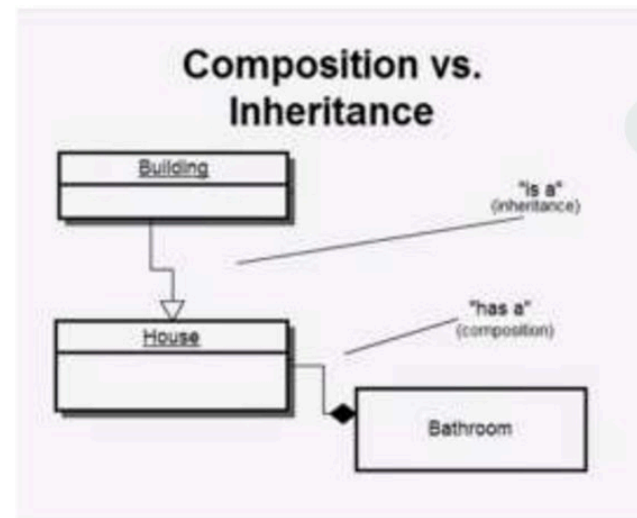


Composition: every car has an engine.

1



Aggregation: cars may have passengers, they come and go



2



OO concepts: Relationships

- Generalization
- Association
- Aggregation
- Composite aggregation (composition)
- Polymorphism

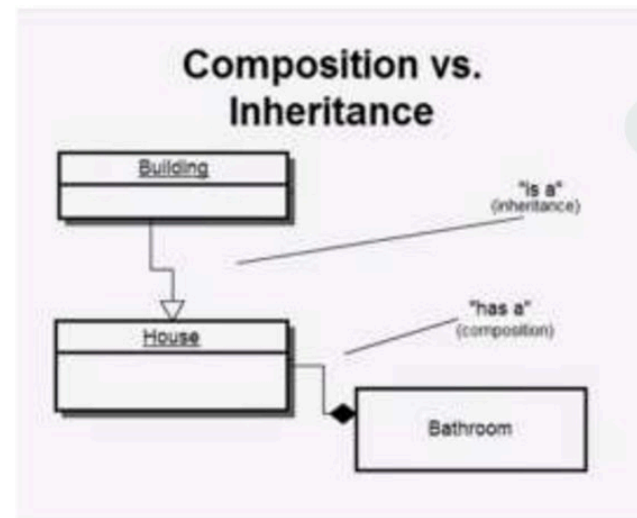


Composition: every car has an engine.

1



Aggregation: cars may have passengers, they come and go

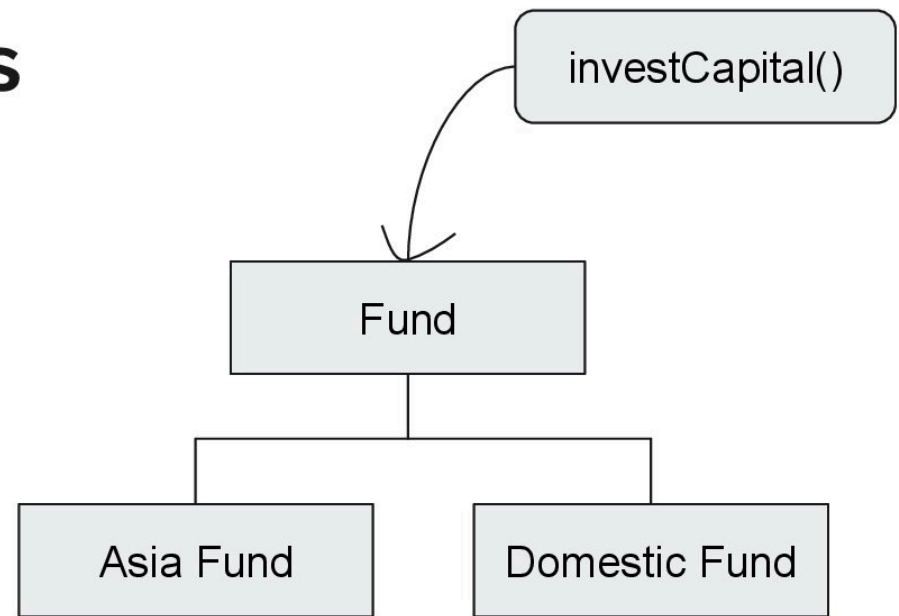


2



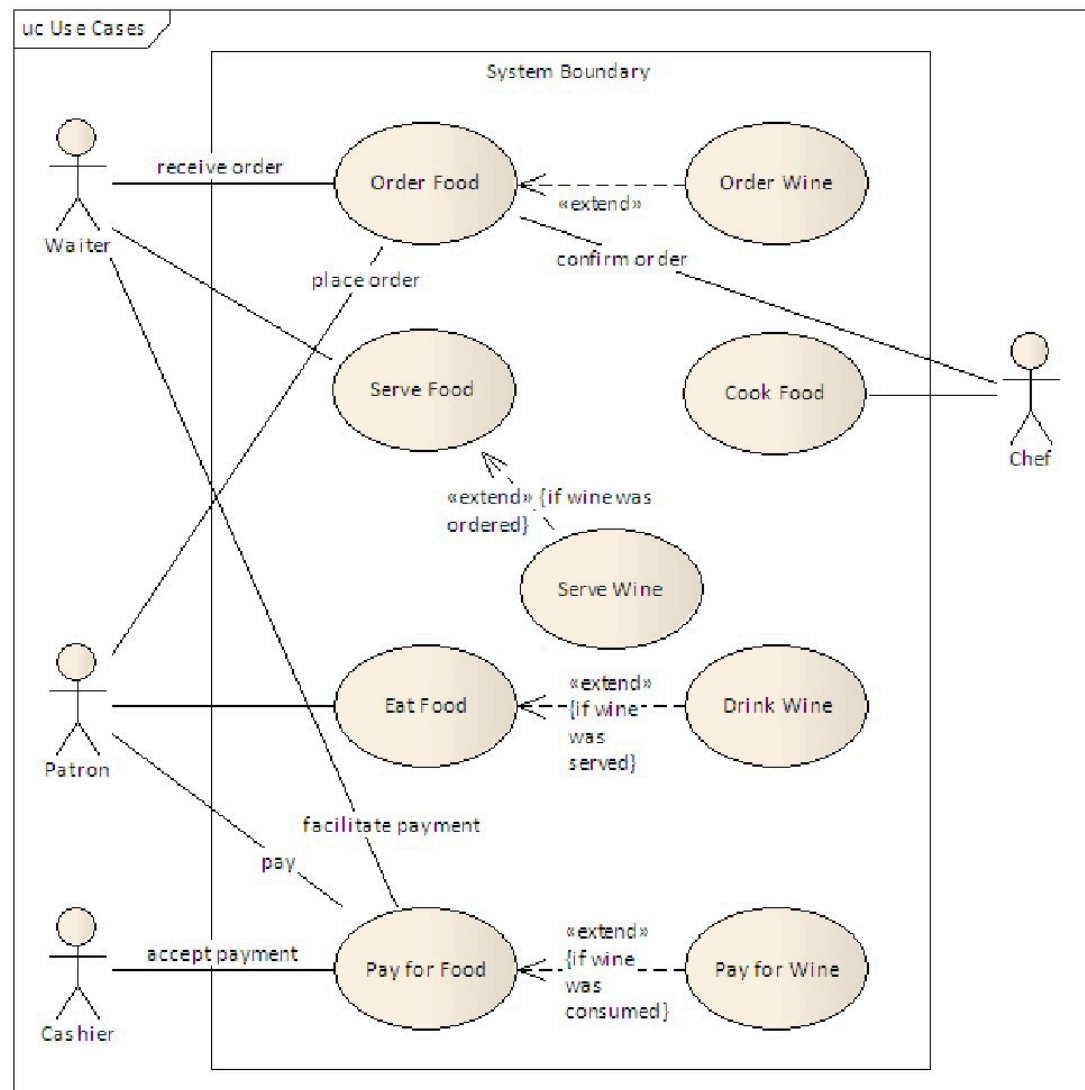
OO concepts: Relationships

- Generalization
- Association
- Aggregation
- Composite aggregation (composition)
- **Polymorphism**





OO concepts: Use cases & scenarios





OO concepts: Use cases & scenarios

Use case: refers to an interaction with any type of system. The question is, what type of system is being referring to?

Business use case: interaction with a business system. For example, Process Claim is a business use case describing an interaction with an insurance company.

System use case: interaction with an IT system. For example, system use cases that support the aforementioned business use case are Record Claim, Validate Coverage, Assign Adjuster, and so on. Each of these describes an interaction between a user and the computer system. A system use case typically involves one active (primary) user and takes place over a single session on the computer. At the end of the system use case, the user should feel that he or she has achieved a useful goal.

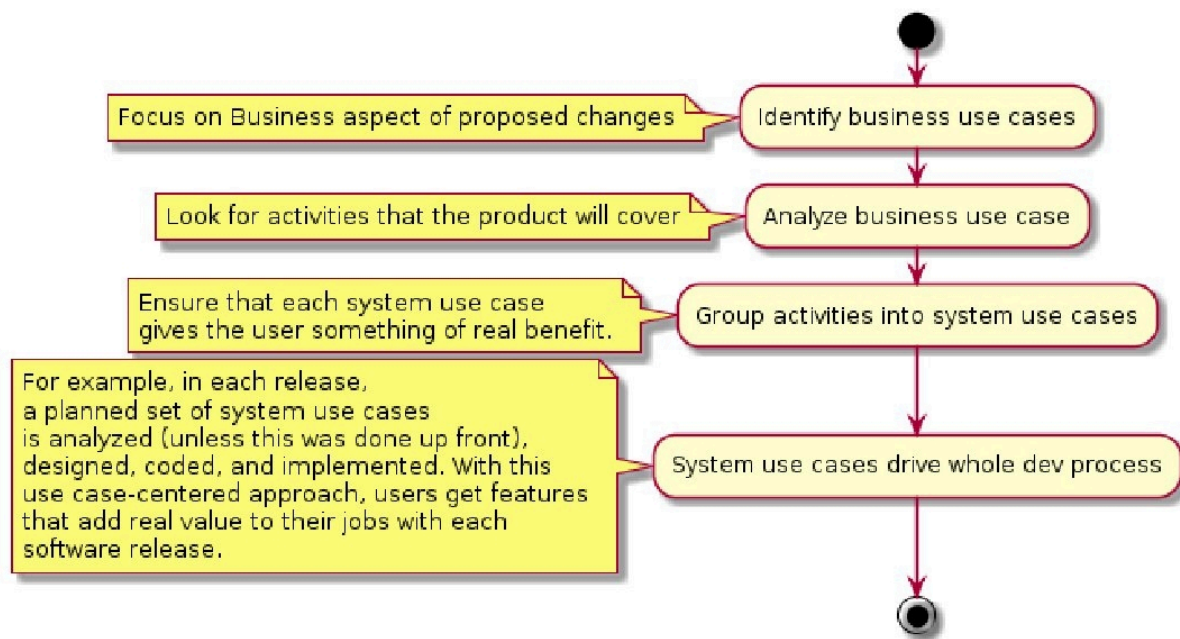
System use cases are often called just use cases.



OO concepts: Use cases & scenarios

How to apply?

Activity Diagram





To conclude

A short horizontal bar with a teal-to-orange gradient, positioned above the main text.

Thank you.

