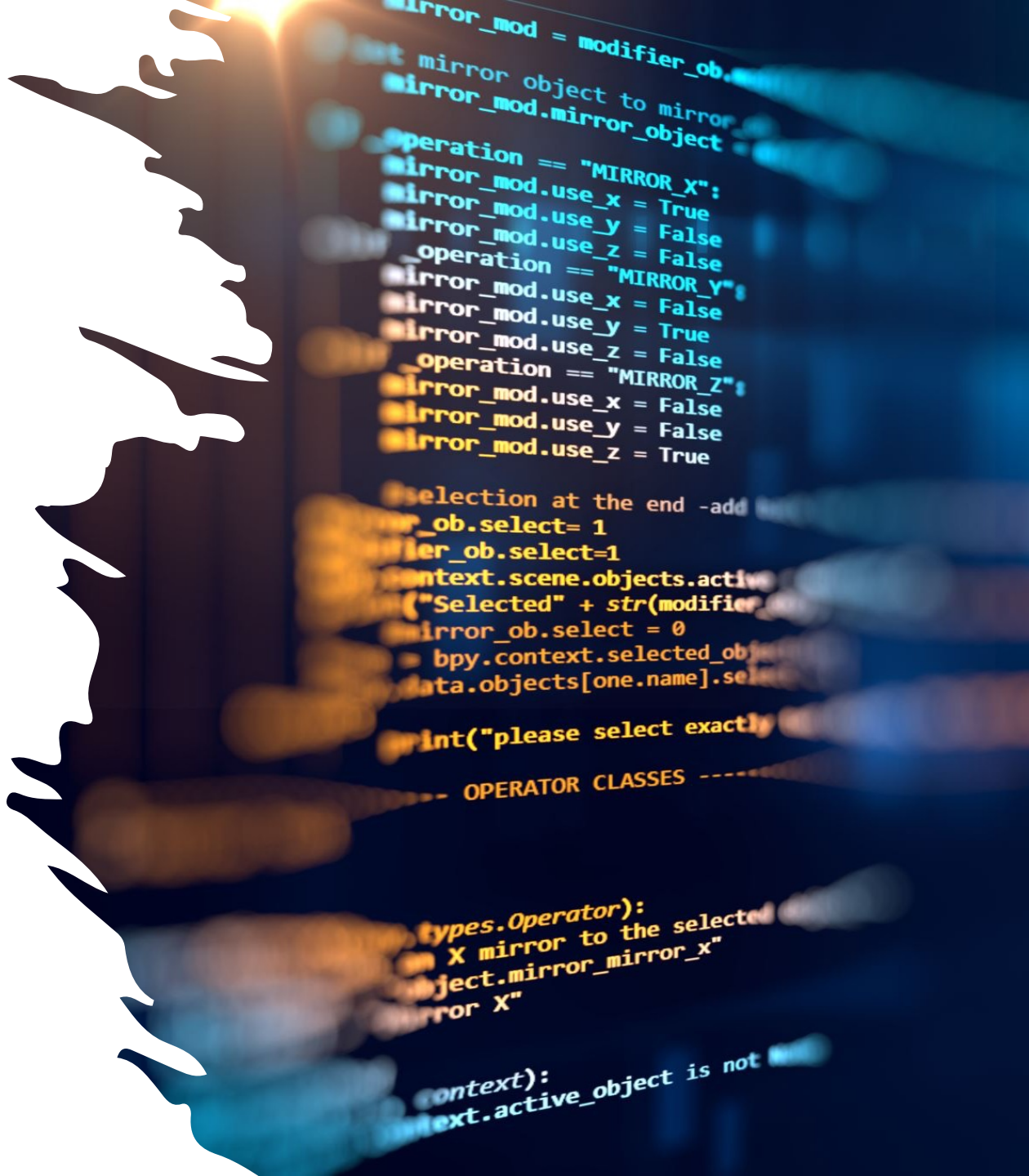
CS4471
Software Design and Architecture:

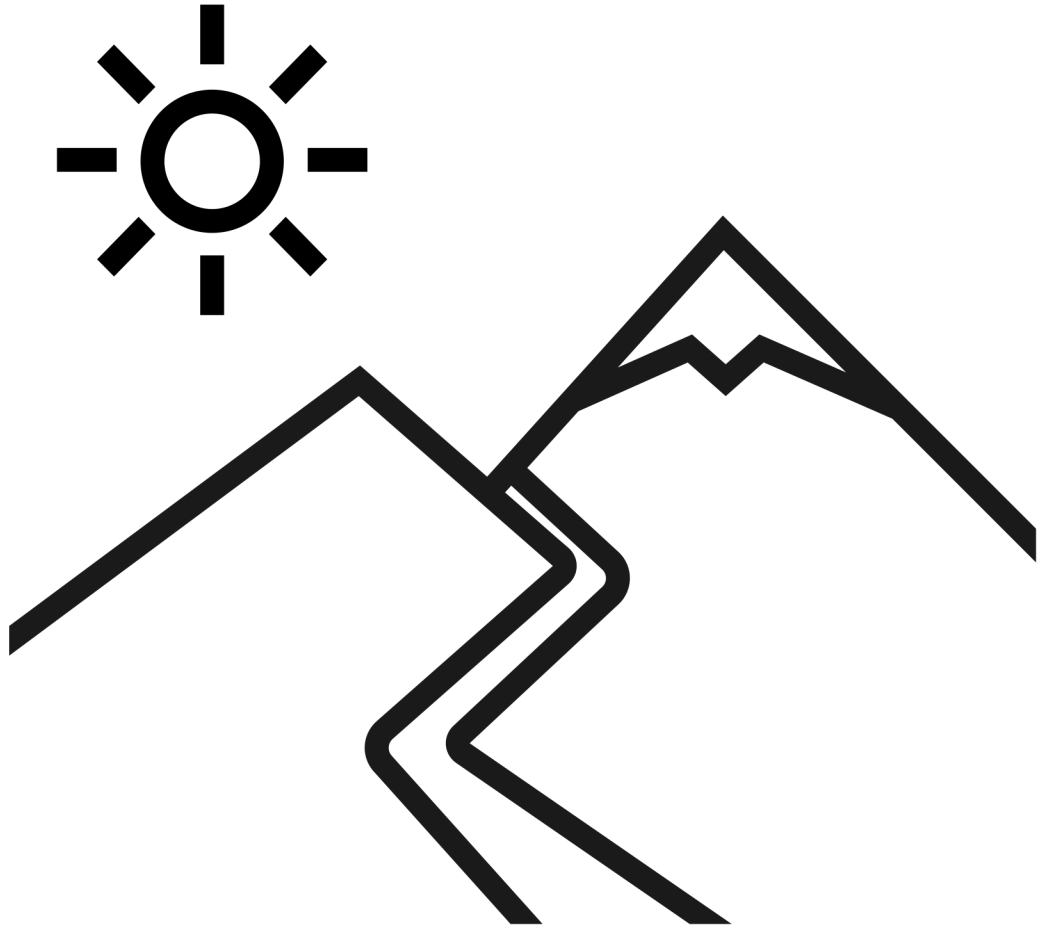
System Models –
Class Diagram (Noun Analysis)

Ahmed Ibrahim



Attendance

Project Roadmap



Problem Definition

- Determine the business requirements related to the problem's root cause.
- Examples:
  - We need an accurate monthly sales forecast which will help us increase our sales.
  - We need to generate a monthly report that indicates why lost customers closed their accounts.
  - We need a notification to account executives when a customer opens a problem ticket.

Project Objective

- Describe the scope of the project you propose to do. They communicate the project purpose in clear, and tangible pieces.
- Example:
  - C-level executives can generate an accurate monthly sales forecast report in three clicks or less.
  - Reduce the loss of customers who closed their accounts by 5%.

Project Deliverables

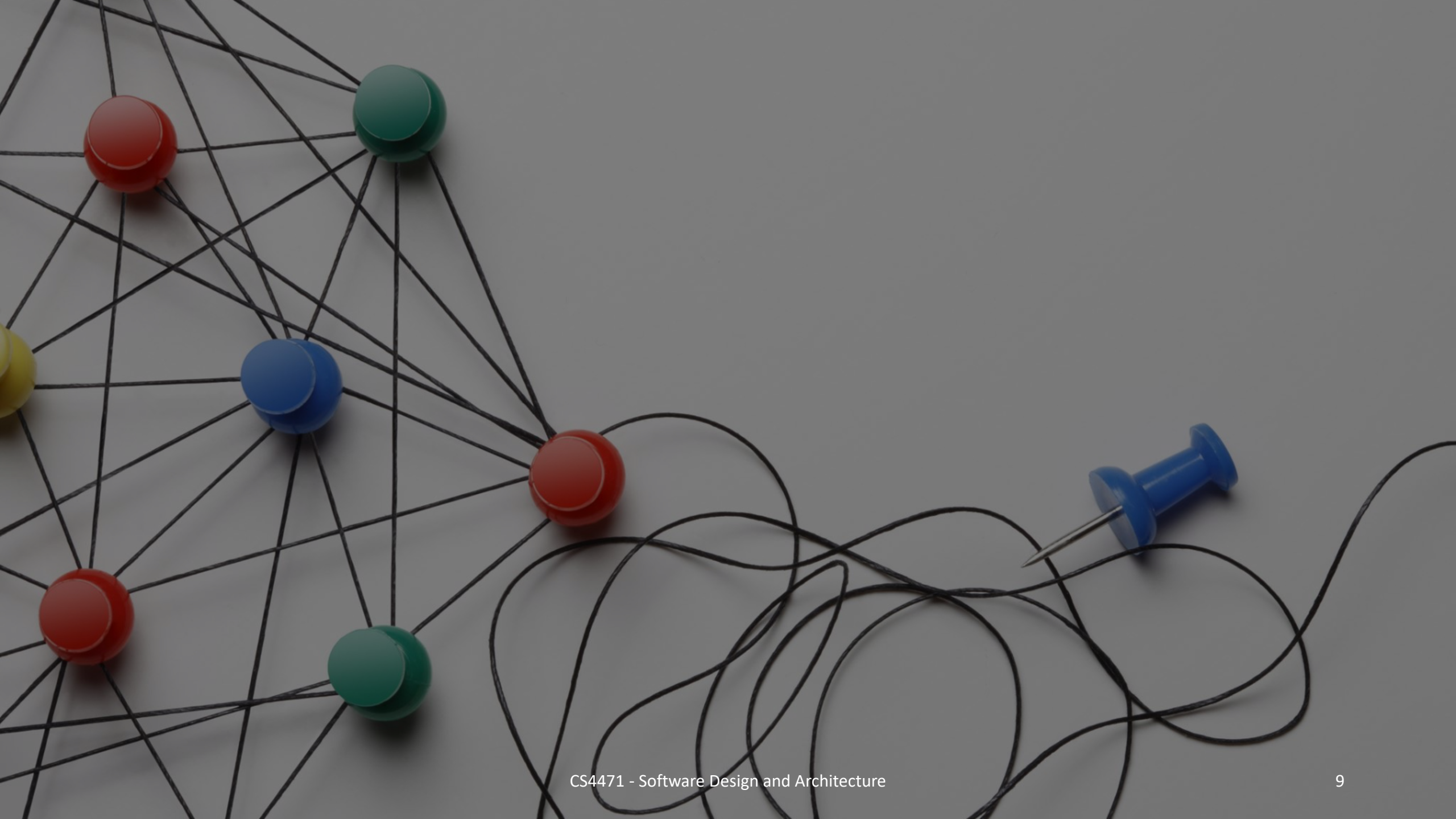
- Deliverables can vary based on the process model you are following.
- However, some expected deliverables could be listed:
  - A shippable product
  - Some documentation
  - Some lessons learned
  - Might be others…

Stakeholders

- To ensure success, the project team must identify stakeholders and understand their expectations.
- Anyone having any relation/interest in the project is known as a **stakeholder**.
  - **Internal stakeholder** – A person, group, or company **directly** involved in the project.
  - **External stakeholder** – A one linked **indirectly** to the project but has a significant contribution to the successful completion of the project.
- **A product is created for a customer**. You must understand precisely what the customer wants or wind up creating a product that could be shunned.

Success / Acceptance Criteria

- The project success criteria refer to measurable terms of the project's outcome acceptable to the end-user, customer, and stakeholders.
- Examples: Cost, Timeline, business requirements, scope, etc.
- Acceptance criteria (AC) are the conditions that a software product must meet to be accepted by a user, a customer, or other systems.



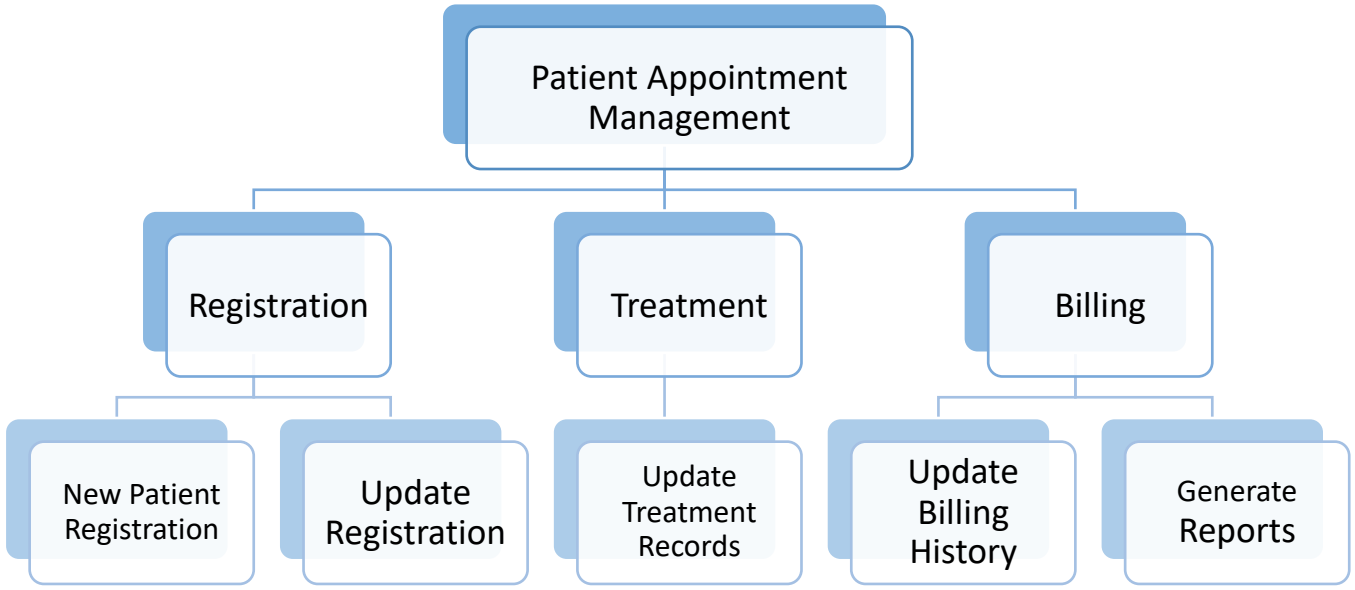
Top-down Approach

- Given the system **specifications**
- Develop a working system
  - Divide the problem into **abstract** modules (Functional Decomposition)
  - Reiterate until constituent parts are reached

- Pros
  - Highly predictable design cycle
  - Efficient division of team members
- Cons
  - More time spent in planning

Functional Decomposition

- Functional decomposition is used to facilitate the understanding of large and/or complex business functions.
- A top-down representation of a **business function**(s).
- Business functions are broken down into lower-level processes.



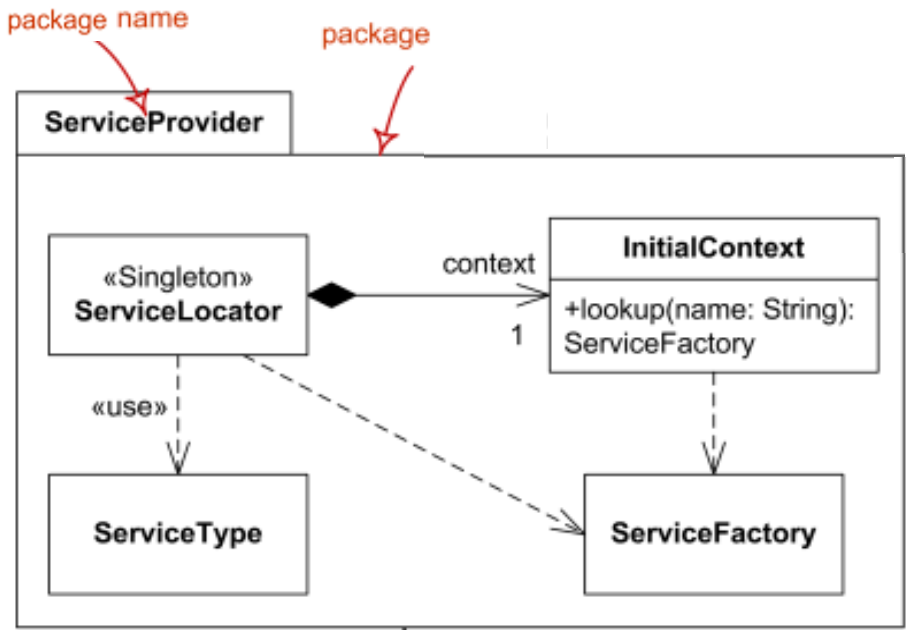
Bottom-up Approach

- Given simple parts/components/classes
- Develop a working system
- Build **packages** to accomplish **specific tasks**
- Integrate packages into a working system
- For example
  - Given a supply AND, OR and NOT gates.
  - Build a computer

- Pros
  - Leads to efficient subsystems
- Cons
  - Complexity is difficult to manage
  - There was little effort given to building reusable packages.

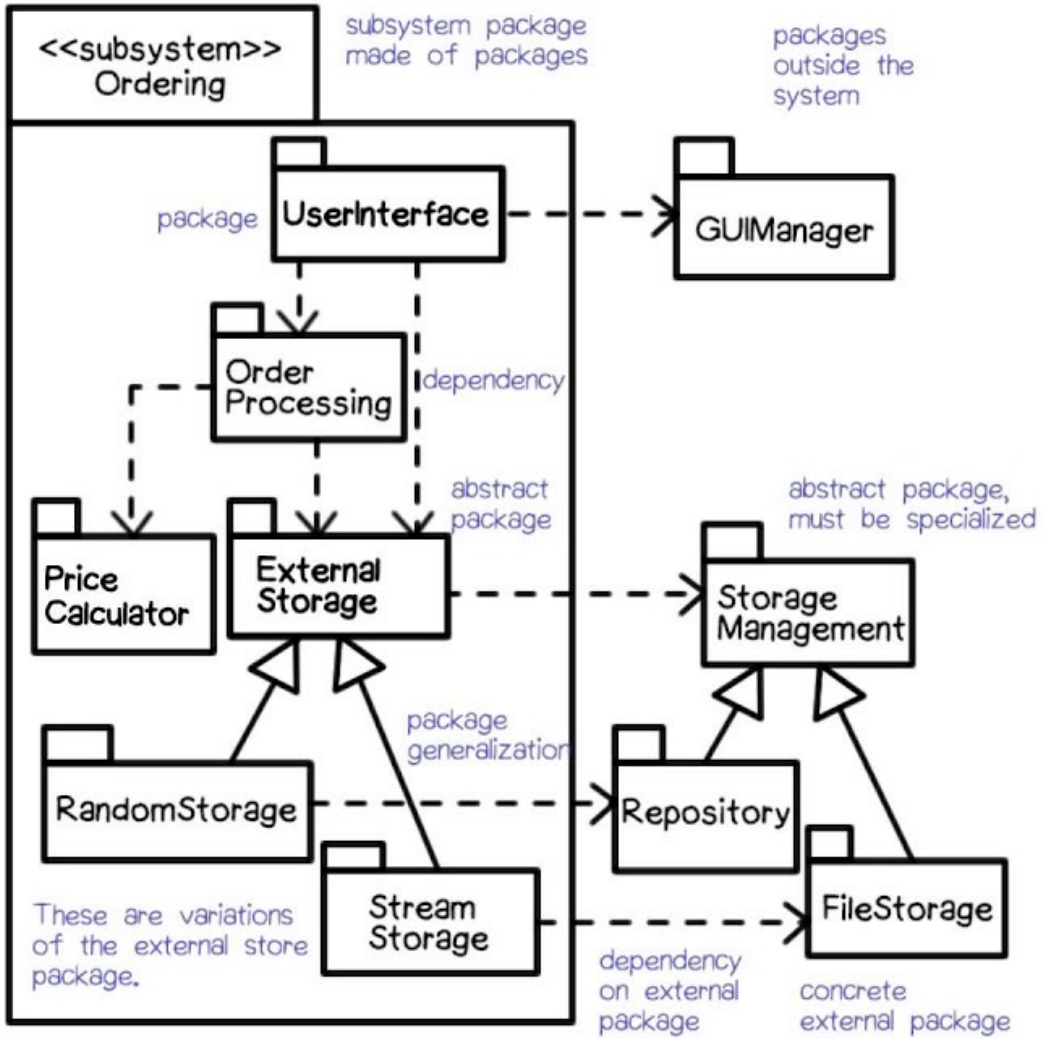
Object-Oriented Design

- Collection of classes, associations, operations, events and constraints that are closely interrelated with each other.
- The objects and classes from the object model are the “seeds” for the **subsystems**.
- In UML subsystems are modelled as **packages**.



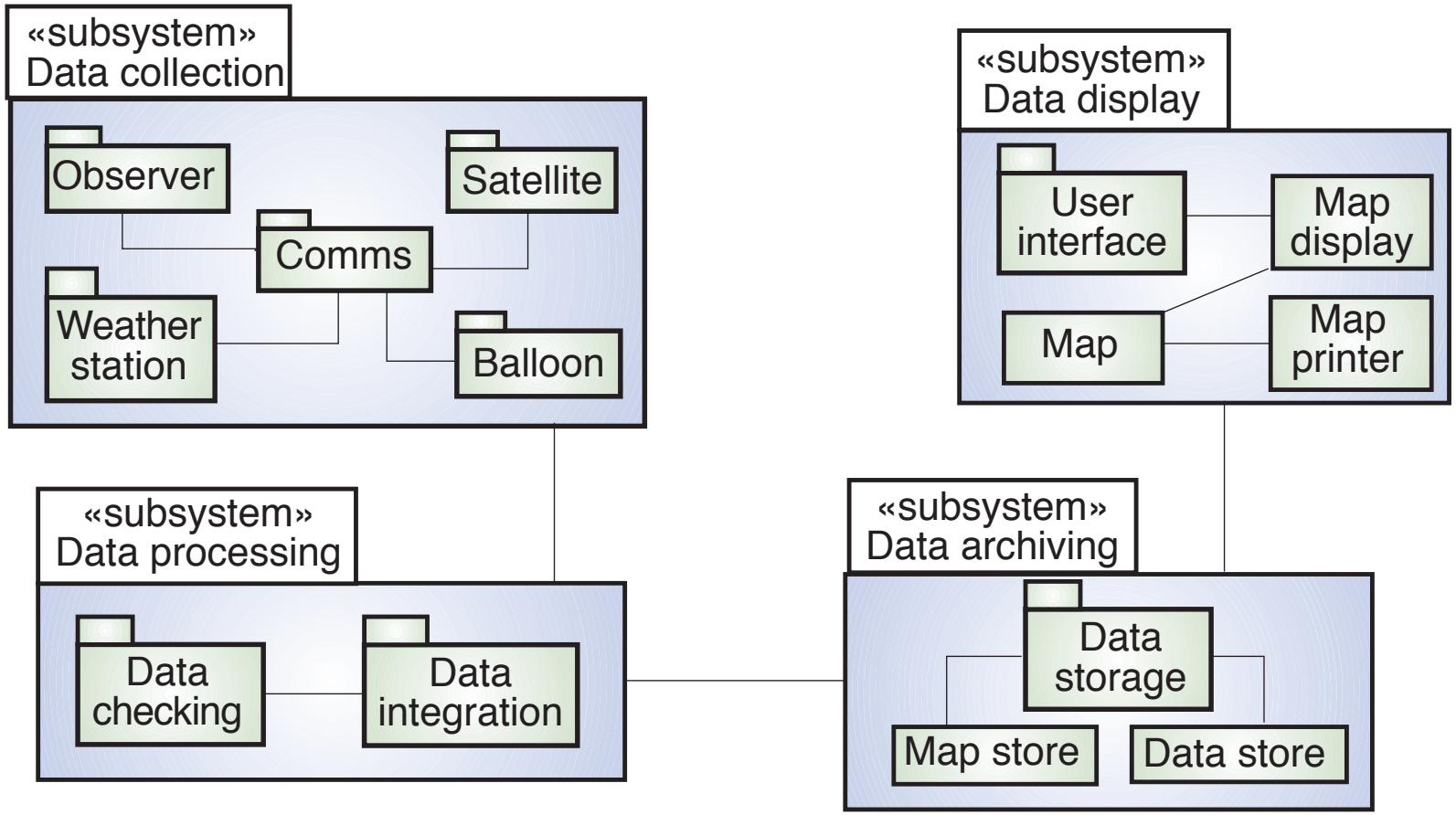
Source: https://rb.gy/5vgtvh

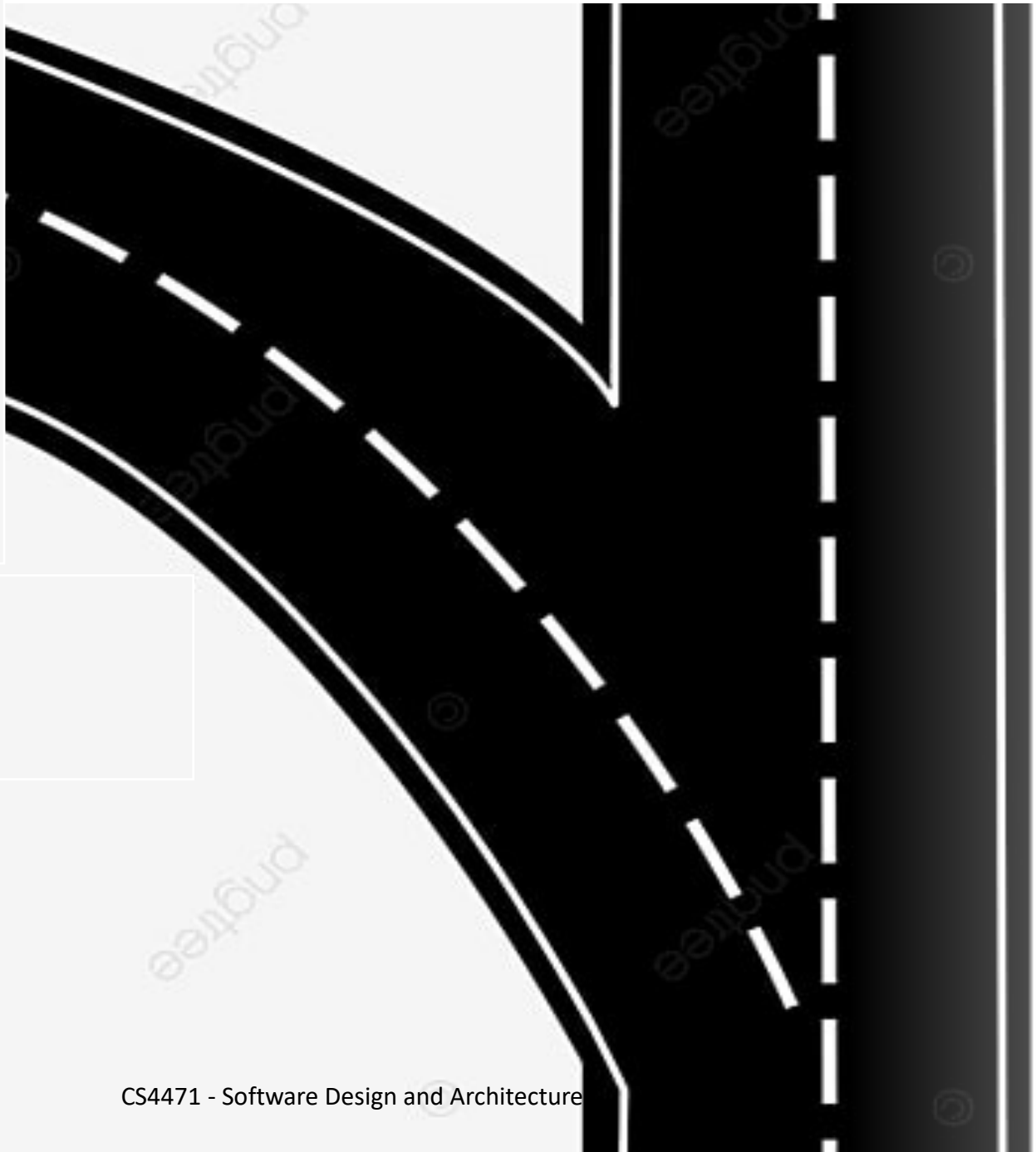
Example of a subsystem with packages (FYI)



Source: https://rb.gy/ycg1pg

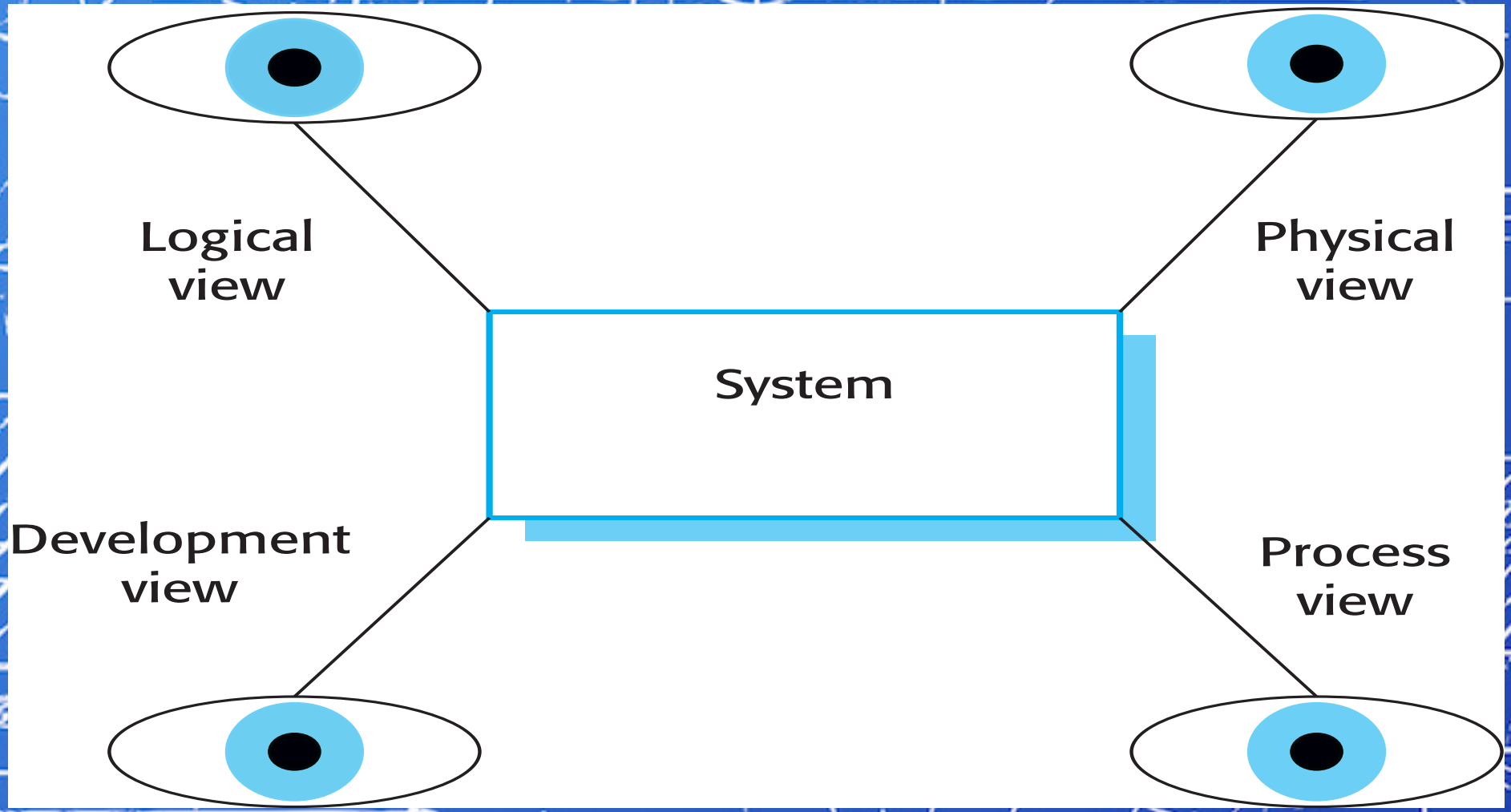
A Weather Monitoring System





Top-down Approach

Bottom-up Approach


Logical view
Physical view
System
Development view
Process view


thank you