

SOFTWARE PROCESS ACTIVITIES

Romenia Silva
romaniasilva@sjp.ac.lk

Today's Agenda

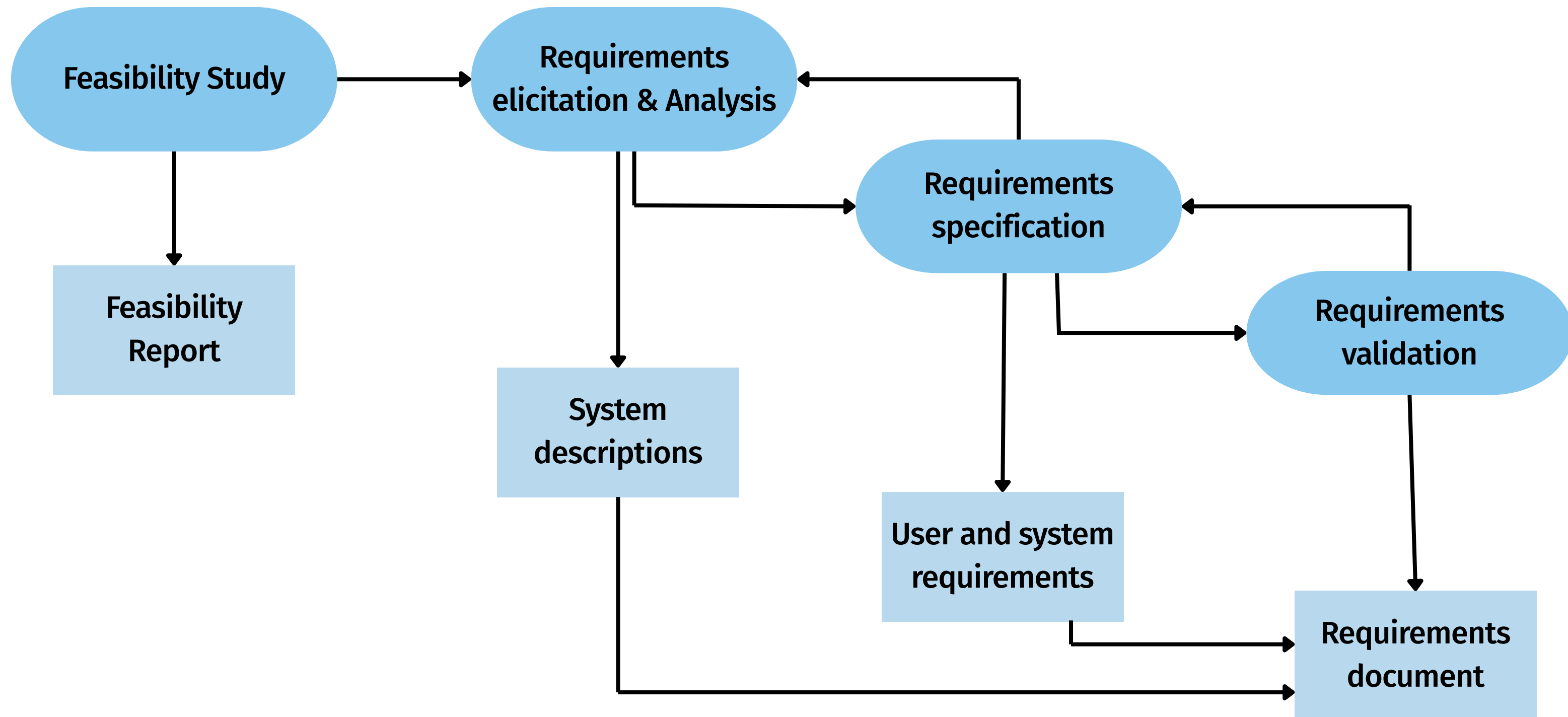
- ◆ Software process activities
 - ◆ The requirements engineering process
 - ◆ Software design and implementation
 - ◆ Software validation
 - ◆ Software evolution
- 

Process activities

- ✧ Real software processes are inter-leaved sequences of technical, collaborative and managerial activities with the overall goal of specifying, designing, implementing and testing a software system.
- ✧ The four basic process activities of specification, development, validation and evolution are organized differently in different development processes.

For example, in the waterfall model, they are organized in sequence, whereas in incremental development, they are interleaved.

The requirements engineering process



Feasibility study

- ✧ A study of whether the customer requirements will be satisfied using current software and hardware technologies is a feasibility study.
- ✧ The study considers the cost-effectiveness and practicability of the required system.
- ✧ The results should inform the decision of whether or not to go ahead with a more detailed analysis.

Requirements Engineering / Specification Goal:

Requirements engineering process

✧ The process of establishing what services are required and the constraints on the system's operation and development.

Feasibility study

- Whether the customer requirements will be satisfied using current software and hardware technologies

Requirements elicitation and analysis

- What do the system stakeholders require or expect from the system?

Requirements specification

- Defining the requirements in detail

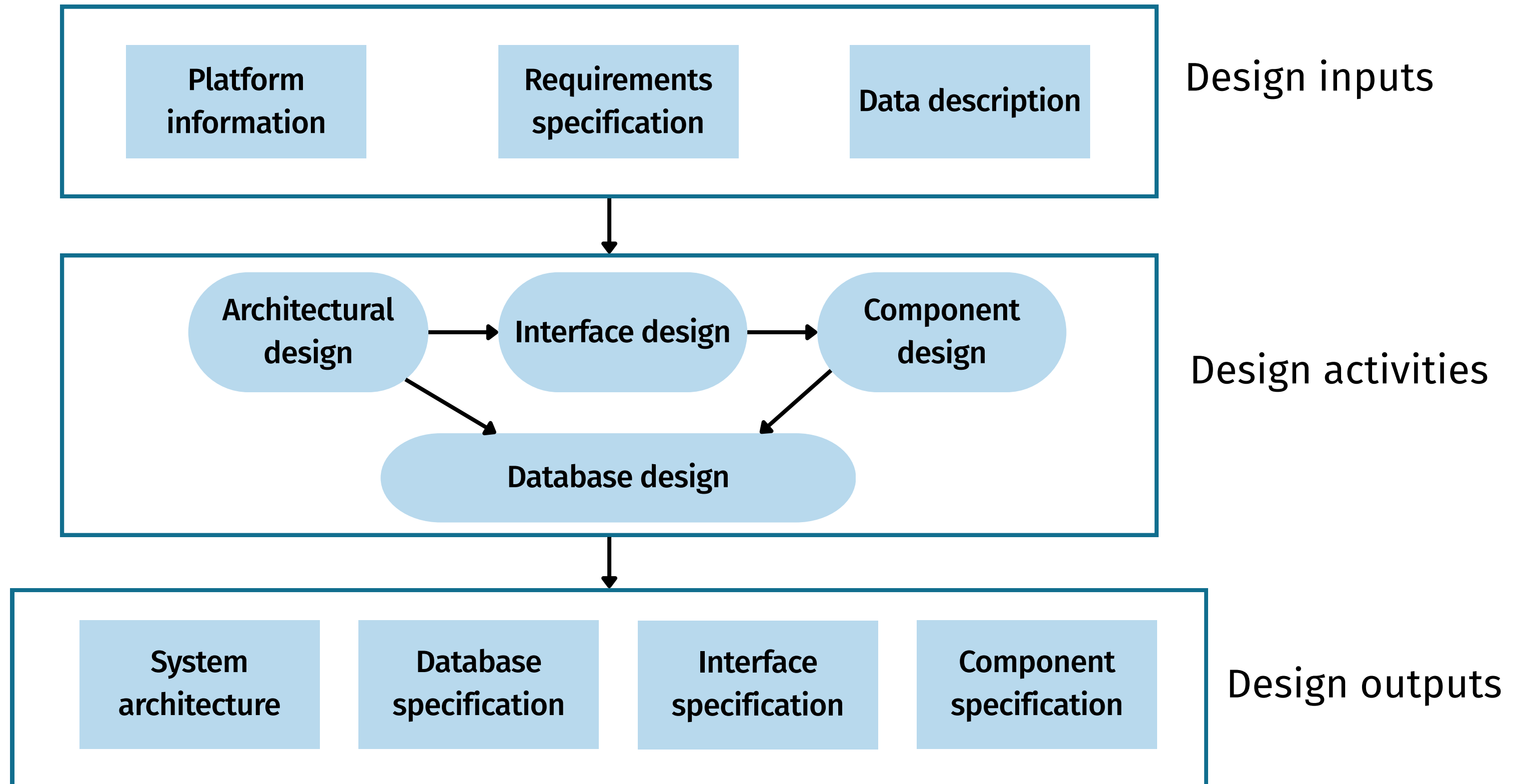
Requirements validation

- Checking the validity of the requirements

Software design and implementation

- ✧ The process of converting the system specification into an executable system.
- ✧ Software design
 - Design a software structure that realises the specification;
- ✧ Implementation
 - Translate this structure into an executable program;
- ✧ The activities of design and implementation are closely related and may be inter-leaved

A general model of the design process



Design activities

- ✧ **Architectural design**, where you identify the overall structure of the system, the principal components (subsystems or modules), their relationships and how they are distributed.
- ✧ **Database design**, where you design the system data structures and how these are to be represented in a database.
- ✧ **Interface design**, where you define the interfaces between system components.
- ✧ **Component selection and design**, where you search for reusable components. If unavailable, you design how it will operate.

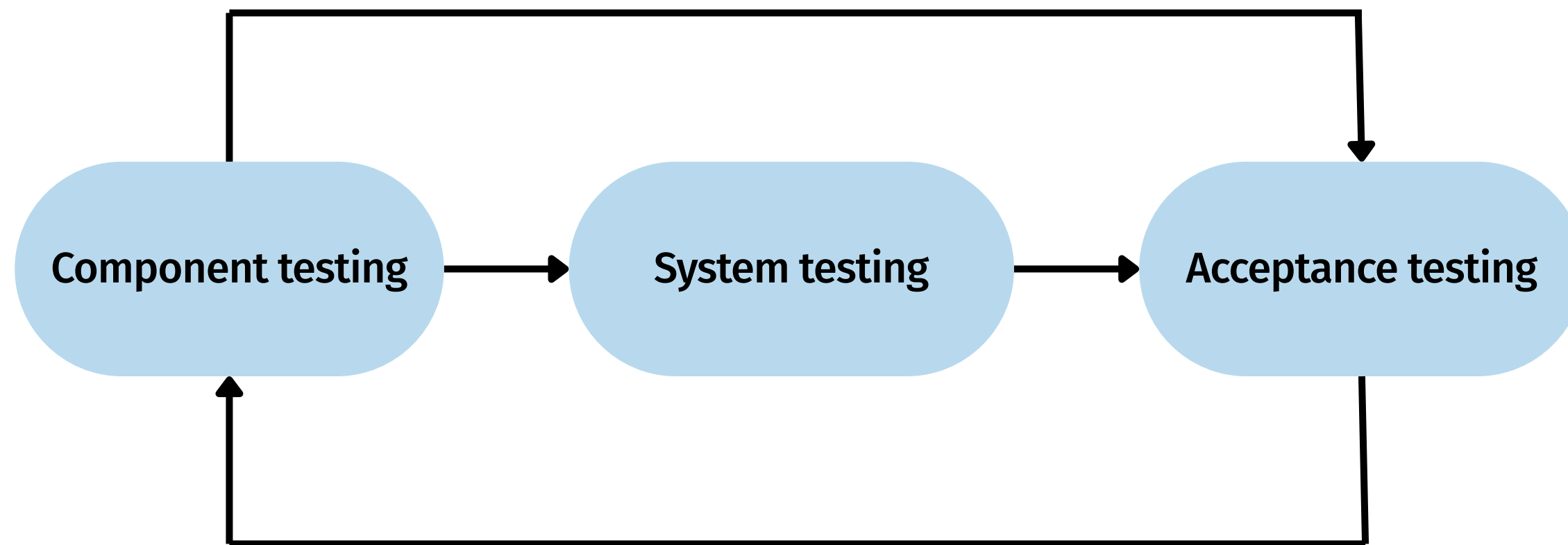
System implementation

- ✧ The software is implemented either by developing a program or programs or by configuring an application system.
- ✧ Design and implementation are interleaved activities for most types of software systems.
- ✧ While programming styles can be deeply individual, modern development relies on team standards, code reviews, and shared practices to ensure consistency.
- ✧ Debugging is the activity of finding program faults and correcting these faults.

Software validation

- ✧ Verification and validation (V & V) is intended to show that a system conforms to its specification and meets the requirements of the system customer.
- ✧ Involves checking and review processes and system testing.
- ✧ System testing involves executing the system with test cases that are derived from the specification of the real data to be processed by the system.
- ✧ Testing is the most commonly used V & V activity

Stages of testing



Stages of Testing

✧ **Component testing**

- Individual components are tested independently;
- Components may be functions, objects, or coherent groupings of these entities.

✧ **System testing**

- Testing of the system as a whole. Testing of emergent properties is particularly important.

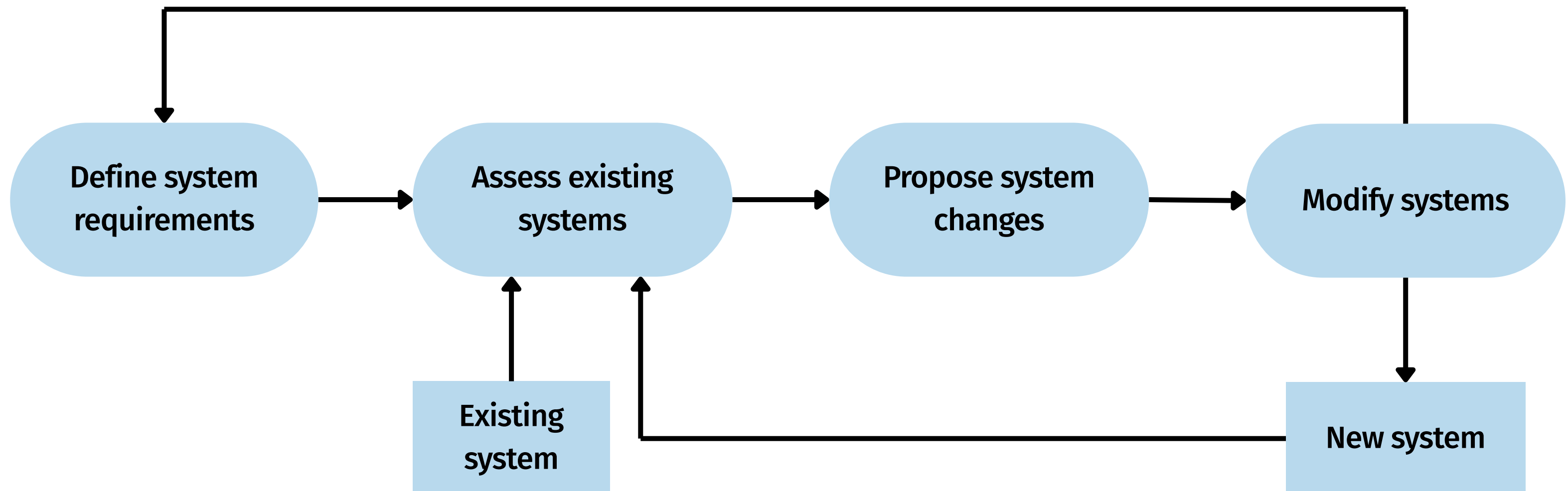
✧ **Acceptance testing**

- Testing with customer data to check that the system meets the customer's needs and expectations.

Software evolution

- ✧ **Software is Inherently Flexible:** Unlike hardware, software can be constantly modified and updated to meet changing business needs.
- ✧ **The Death of "Maintenance":** Historically, companies separated "Development" (building new software) from "Maintenance" (fixing old software). Today, this distinction is irrelevant.
- ✧ **Why It Changed:** Fewer and fewer systems are built completely from scratch. Modern software engineering is almost always about evolving, modifying, and integrating with existing systems.

System evolution



That's a wrap!

Thank you!

.....

June 22, 2025

