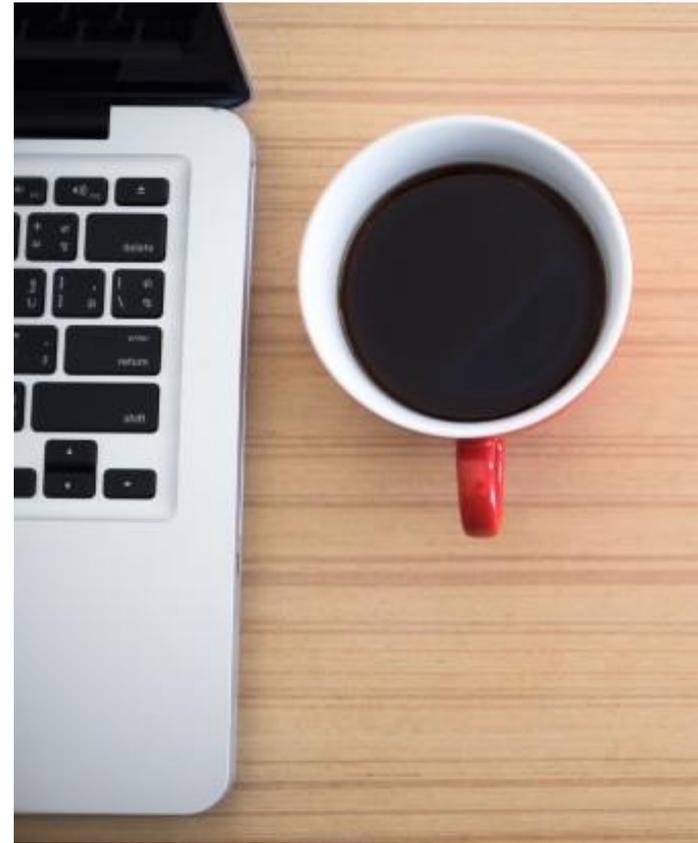


Course Topics

- Introduction
- Software Process Models
- Requirements Engineering
- Modeling
- Programming Languages
- Software Construction Techniques
- Testing
- Project Management
- Refactoring
- Ethical Issues

Lecture Objectives

- ✓ Four basic process activities
 - Specification
 - Development
 - Validation
 - Evolution



Process activities



- Real software processes are **interleaved** sequences of
 - technical,
 - collaborative and
 - managerial activities
- with the overall goal of specifying, designing, implementing and testing a software system.

Process activities



- The four basic process activities of
 - **specification,**
 - **development,**
 - **validation** and
 - **evolution**
 - They are organized differently in different development processes.
 - In the **waterfall** model, they are organized in sequence, whereas in **incremental** development they are interleaved.
- 

Software specification

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

- Requirements engineering process
 - Feasibility study
 - Is it technically and financially feasible to build the system?
 - 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

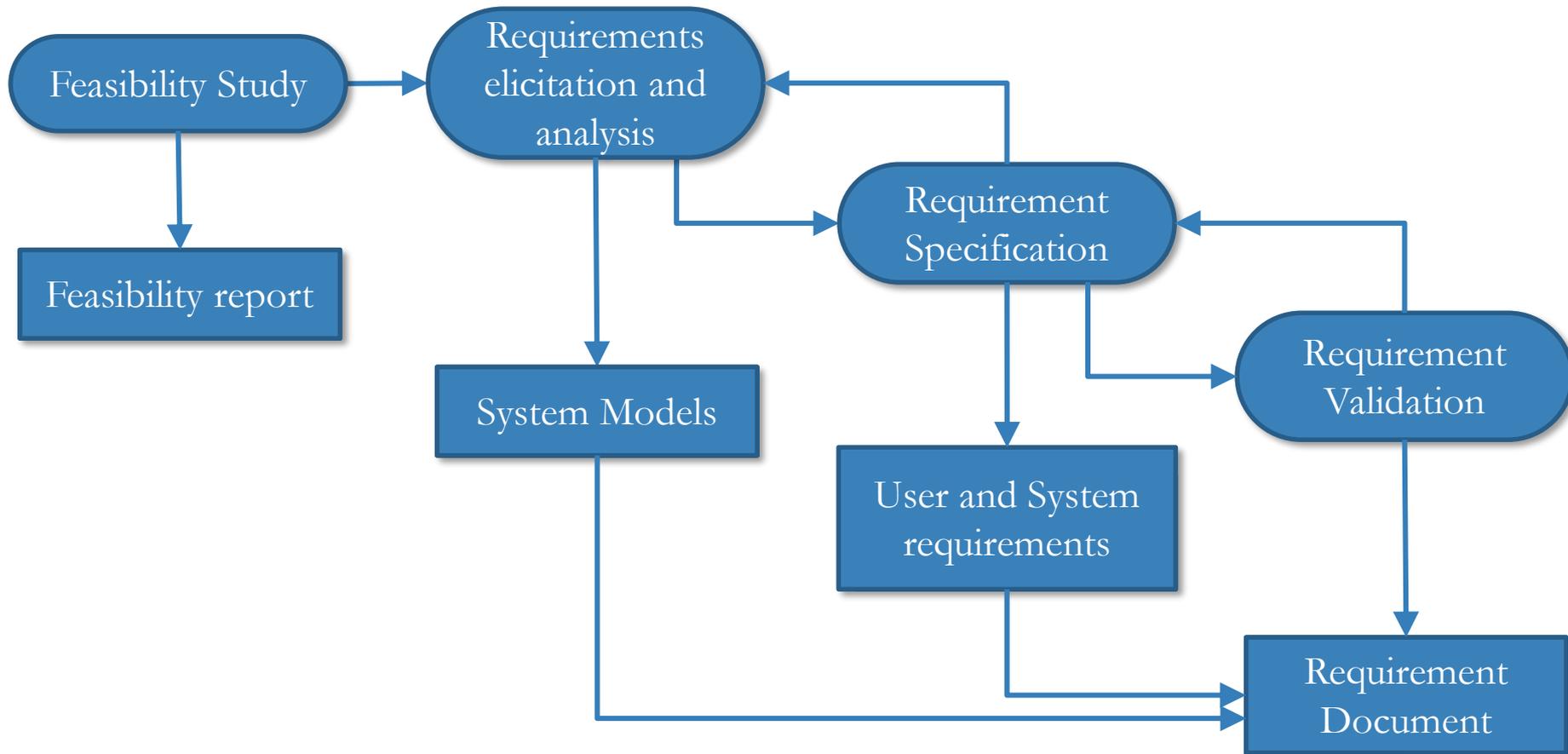
Software specification

- For those who think that requirements are easy



<http://www.dilbert.com/>

The requirements engineering process

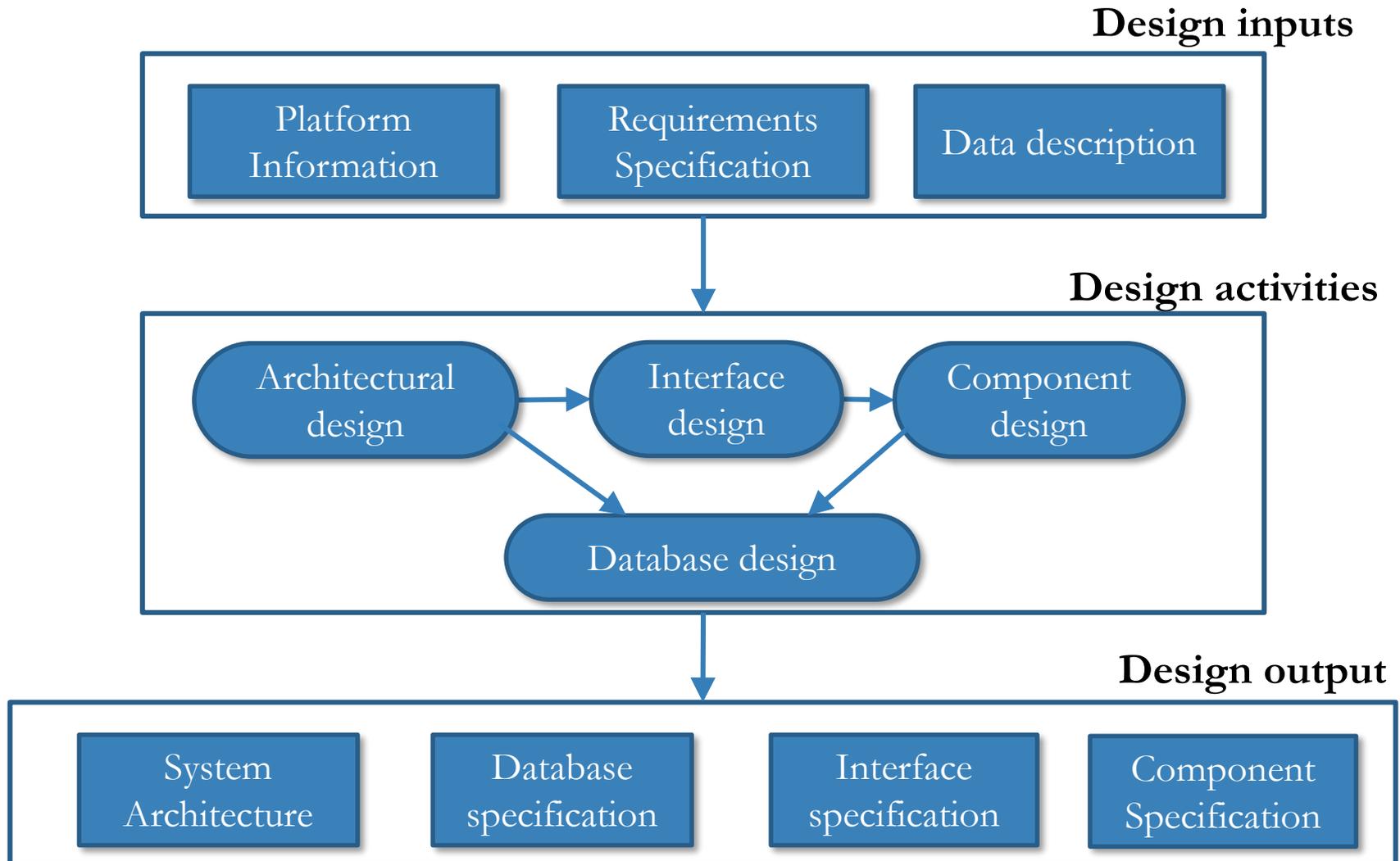


Software design and implementation



- 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 interleaved.
- 

A general model of the design process



Design activities



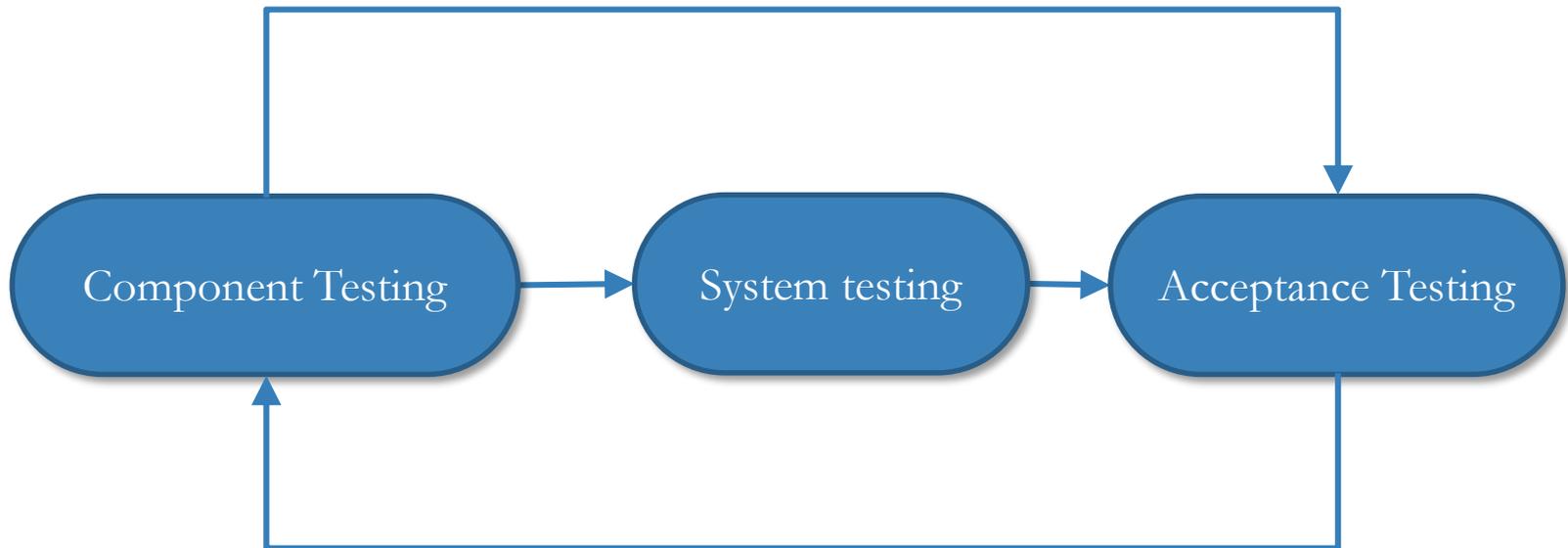
- **Architectural design**, where you identify the overall structure of the system, the principal components, their relationships and how they are distributed.
 - **Interface design**, where you define the interfaces between system components.
 - **Component design**, where you take each system component and design how it will operate.
 - **Database design**, where you design the system data structures and how these are to be represented in a database.
- 

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



Testing stages

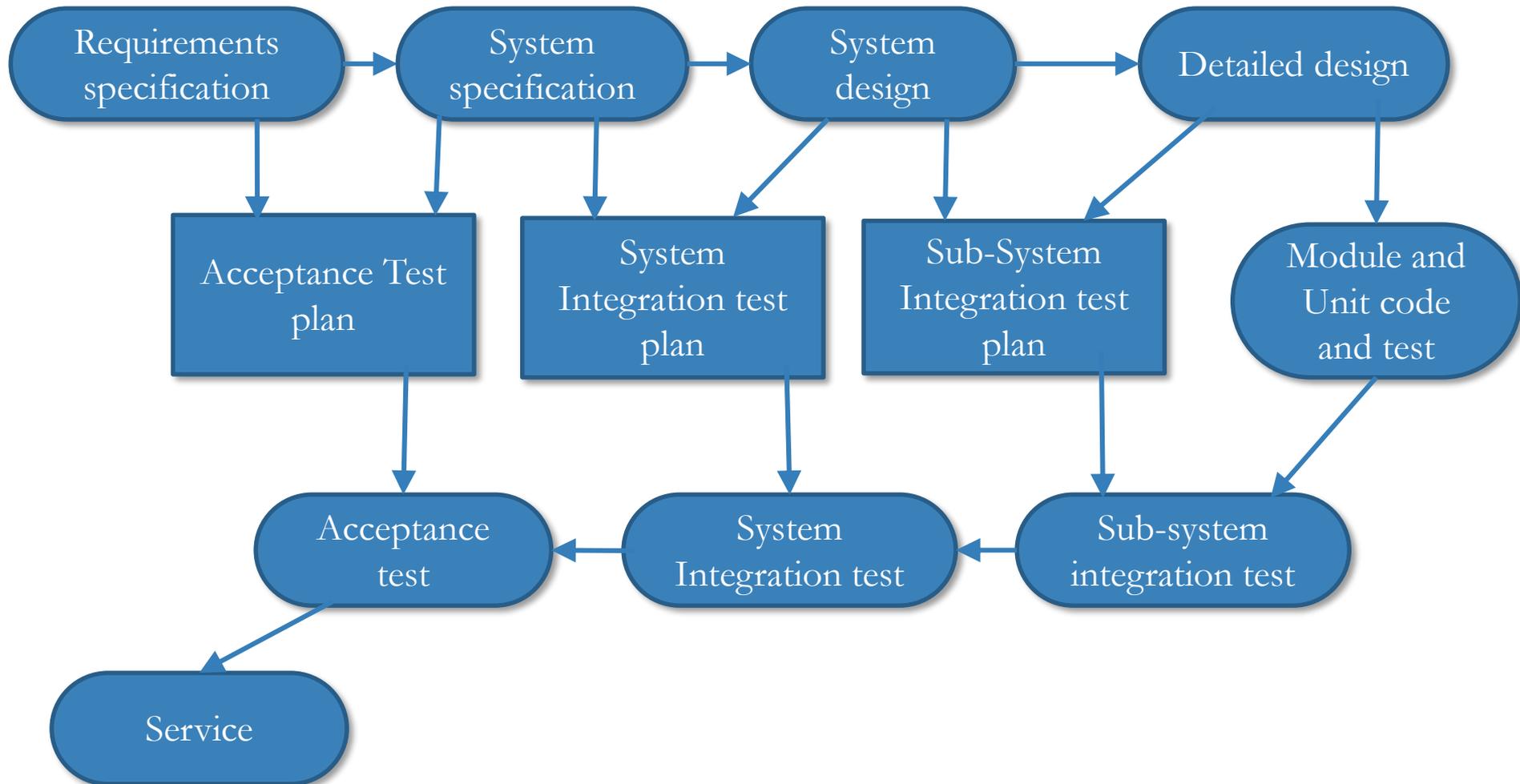


- Development or component testing
 - Individual components are tested independently;
 - Components may be functions or objects or coherent groupings of these entities.

 - 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.
- 

Testing phases in a plan-driven software process

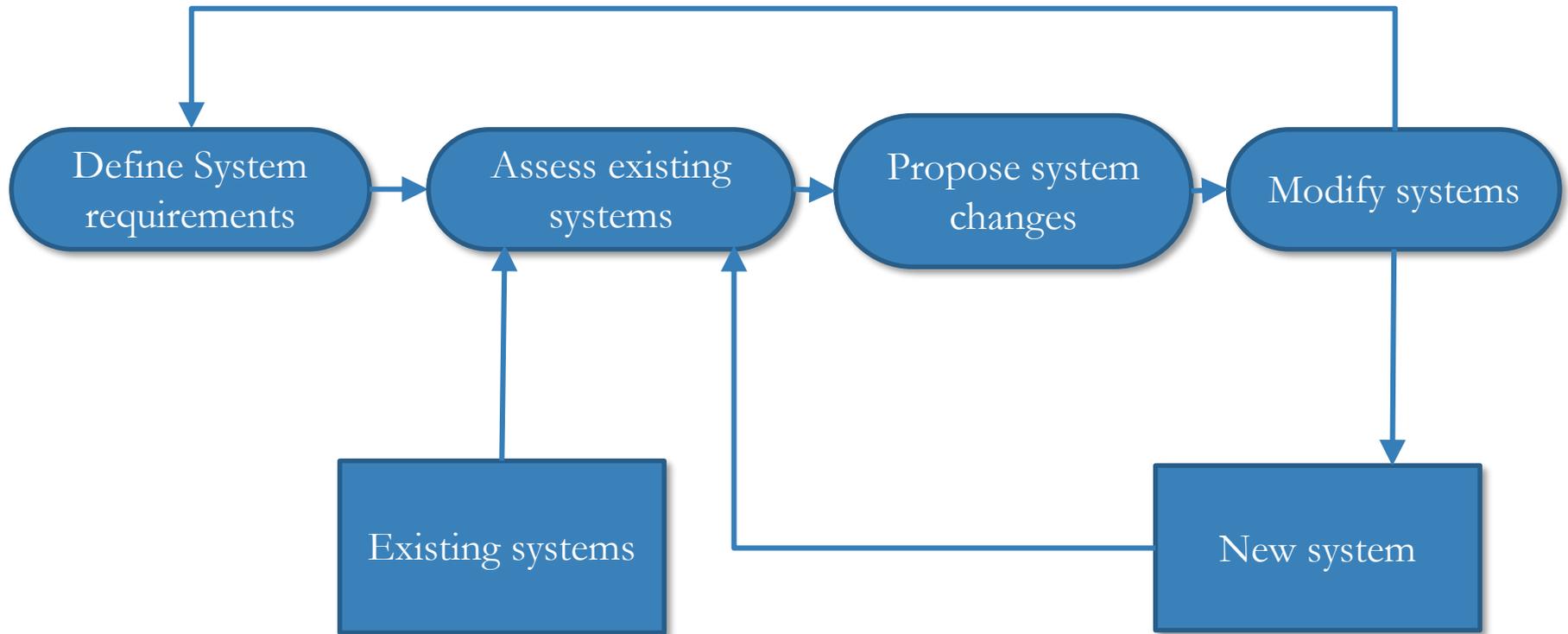


Software evolution



- Software is inherently flexible and can change.
 - As requirements change through changing business circumstances, the software that supports the business must also evolve and change.
 - Although there has been a demarcation between development and evolution (maintenance) this is increasingly irrelevant as fewer and fewer systems are completely new.
- 

System evolution



Key points



- **Requirements engineering** is the process of developing a software specification.
 - **Design and implementation** processes are concerned with transforming a requirements specification into an executable software system.
- 

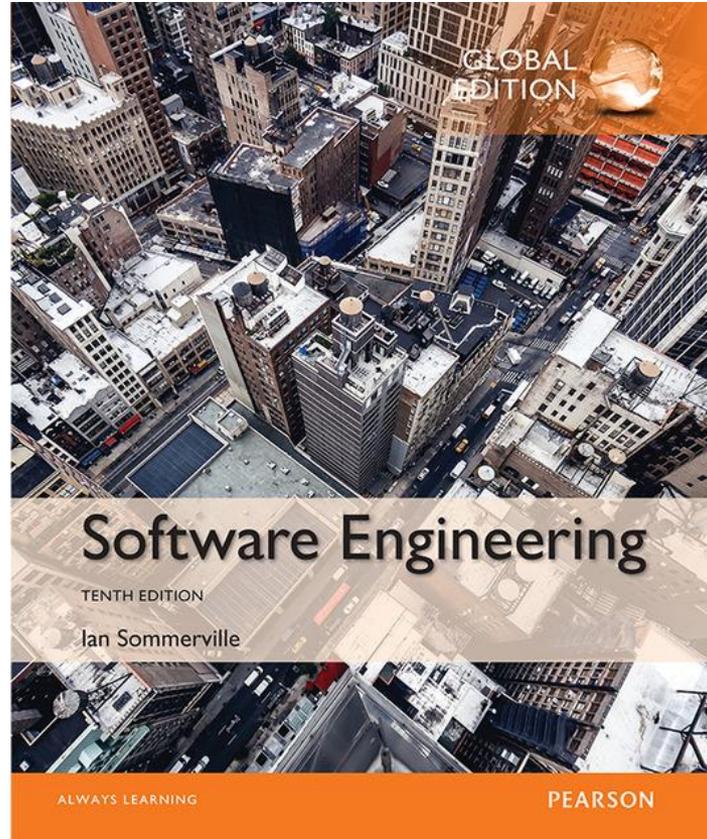
Key points



- Software **validation** is the process of checking that the system conforms to its specification and that it meets the real needs of the users of the system.
 - Software **evolution** takes place when you change existing software systems to meet new requirements. The software must evolve to remain useful.
- 

Read

Chapter 2



References



- Ian Sommerville, “Software Engineering”, 10th Edition, Addison-Wesley, 2015.
- Timothy C. Lethbridge and Robert Laganière, “Object-Oriented Software Engineering: Practical Software Development using UML and Java”, 2nd Edition, McGraw Hill, 2001.
- R. S. Pressman, Software Engineering: A Practitioner’s Approach, 10th Edition, McGraw-Hill, 2005.