COMP41670 Software Engineering

11. Software Engineering

Dr Avishek Nag

UCD School of Computer Science.   Scoil na Ríomheolaíochta UCD.

Table of Contents

Introduction

What Is Software Engineering?

- A program is a list of instructions that are encoded in such a way that they can be executed by a computer so as to perform a specific task.
- Software is a set of computer programs and associated data and documentation.
- **Software engineering** is the systematic development of software.
- Software engineering is concerned with all aspects of software development.
- It is essential for development and maintenance of large scale systems by teams.

How Big Is Big?

- Lines of Code (LOC) is a rudimentary metric but…
- 1 million lines of code = 18,000 printed pages = 14x War and Peace
- https://www.informationisbeautiful.net/visualizations/million-lines-of-code/

- Linux Kernel Development Visualisation (1991-2015)
- https://youtu.be/5iFnzr73XXk

Whose Idea Was This Anyway?

- Margaret Hamilton in 1965
- MIT Instrumentation Laboratory
- Lead the team that developed the guidance and navigation system for the Apollo spacecraft
- https://www.computer.org/publications/tech-news/events/what-to-know-about-the-scientist-who-invented-the-term-software-engineering

Margaret Hamilton, MIT,1969

Aspects of Software Engineering

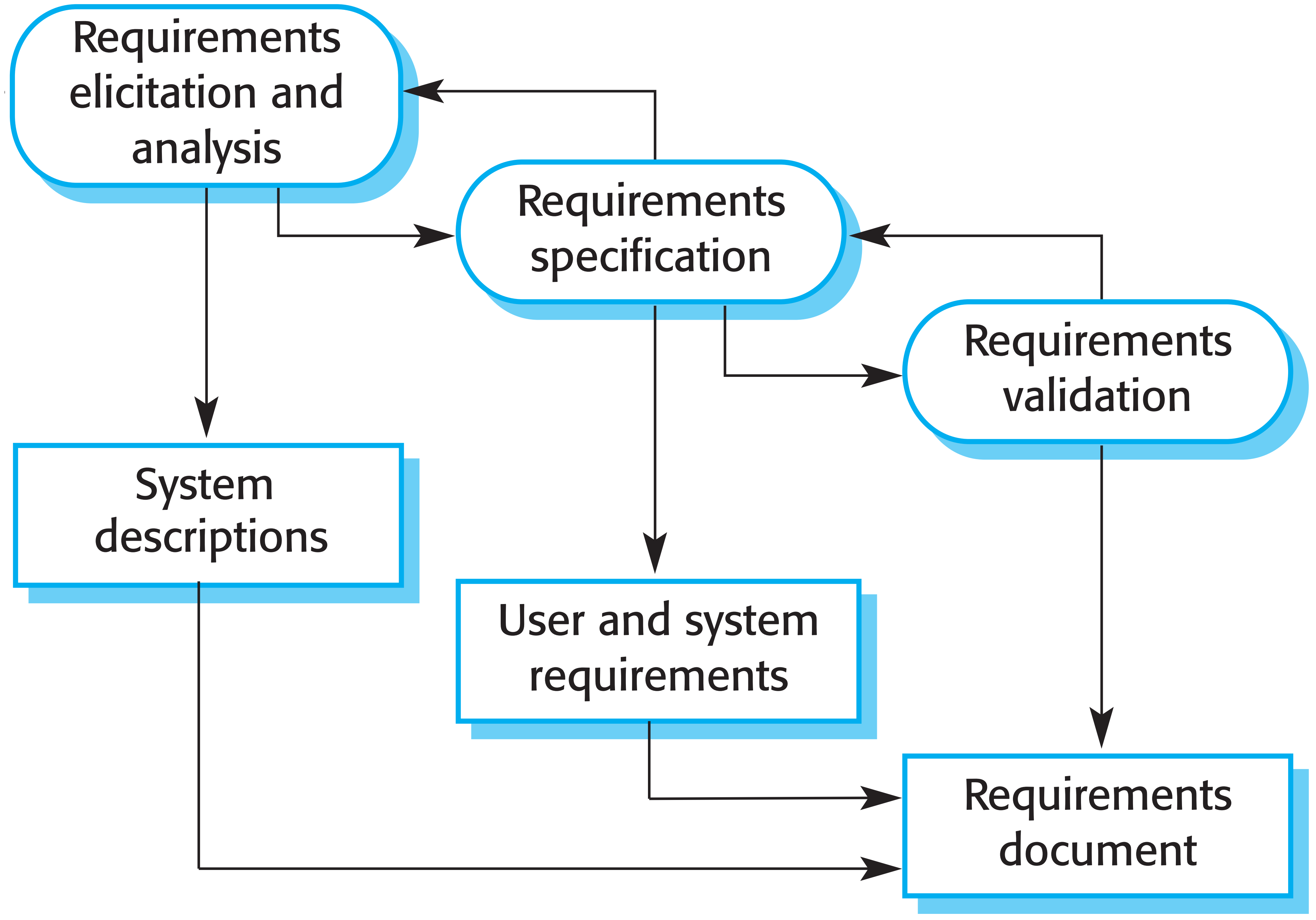
- The **Software Development Lifecycle** (SDLC) is the typical sequence of activities which are performed during software development project.
    1. **Specification**: customers and engineers define the software that is to be produced and the constraints on it
    2. **Development**: the software is designed and programmed.
    3. **Validation**: the functionality and performance of the software is checked to make sure that it is what the customer wanted.
    4. **Evolution**: the software is modified to fix bugs and to meet changing customer and market requirements

Process Activities

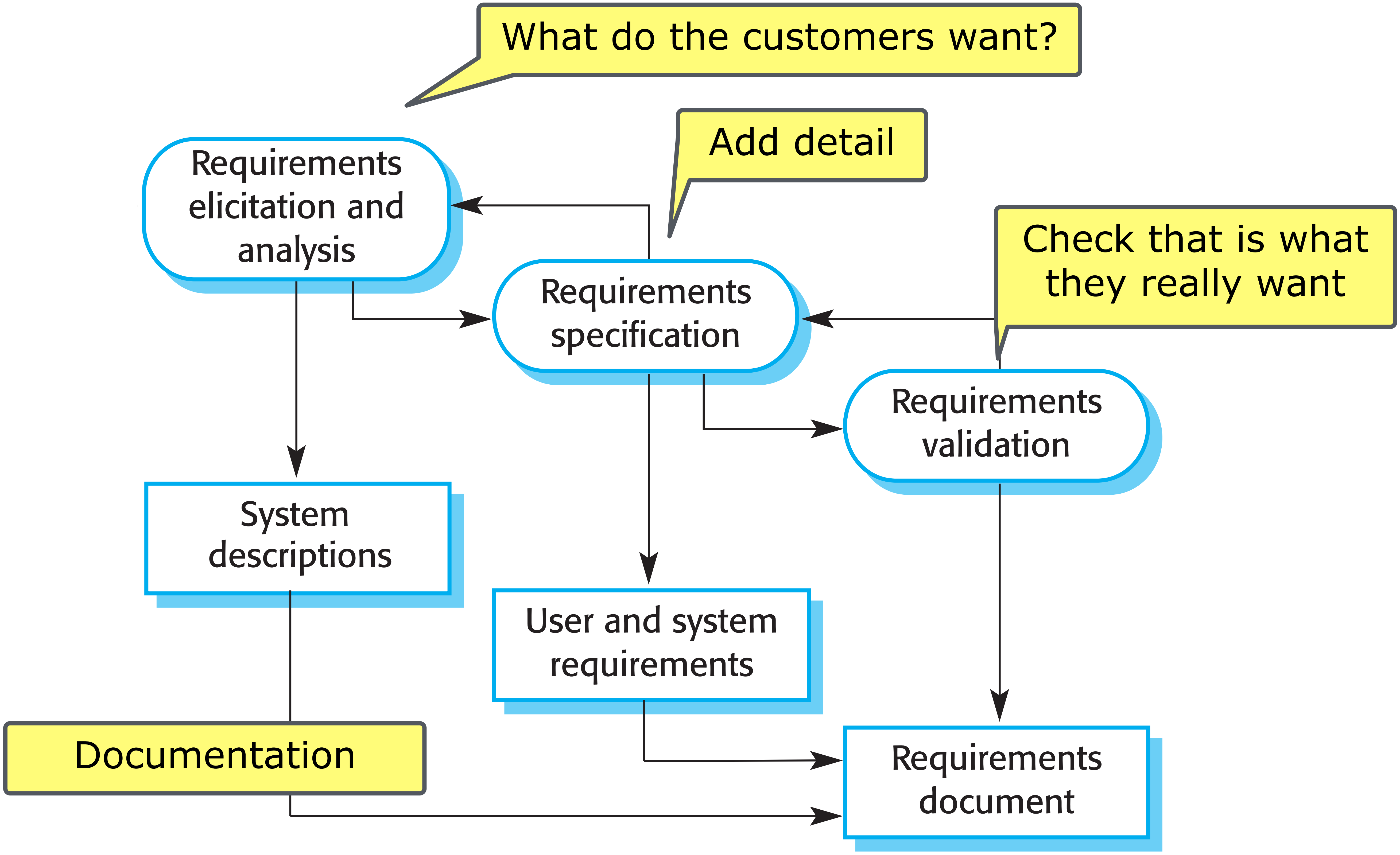
Software Specification

- … is the process of establishing what services are required and the constraints on the system’s operation and development.

Requirements Engineering


Requirements elicitation and analysis
Requirements specification
Requirements validation
System descriptions
User and system requirements
Requirements document

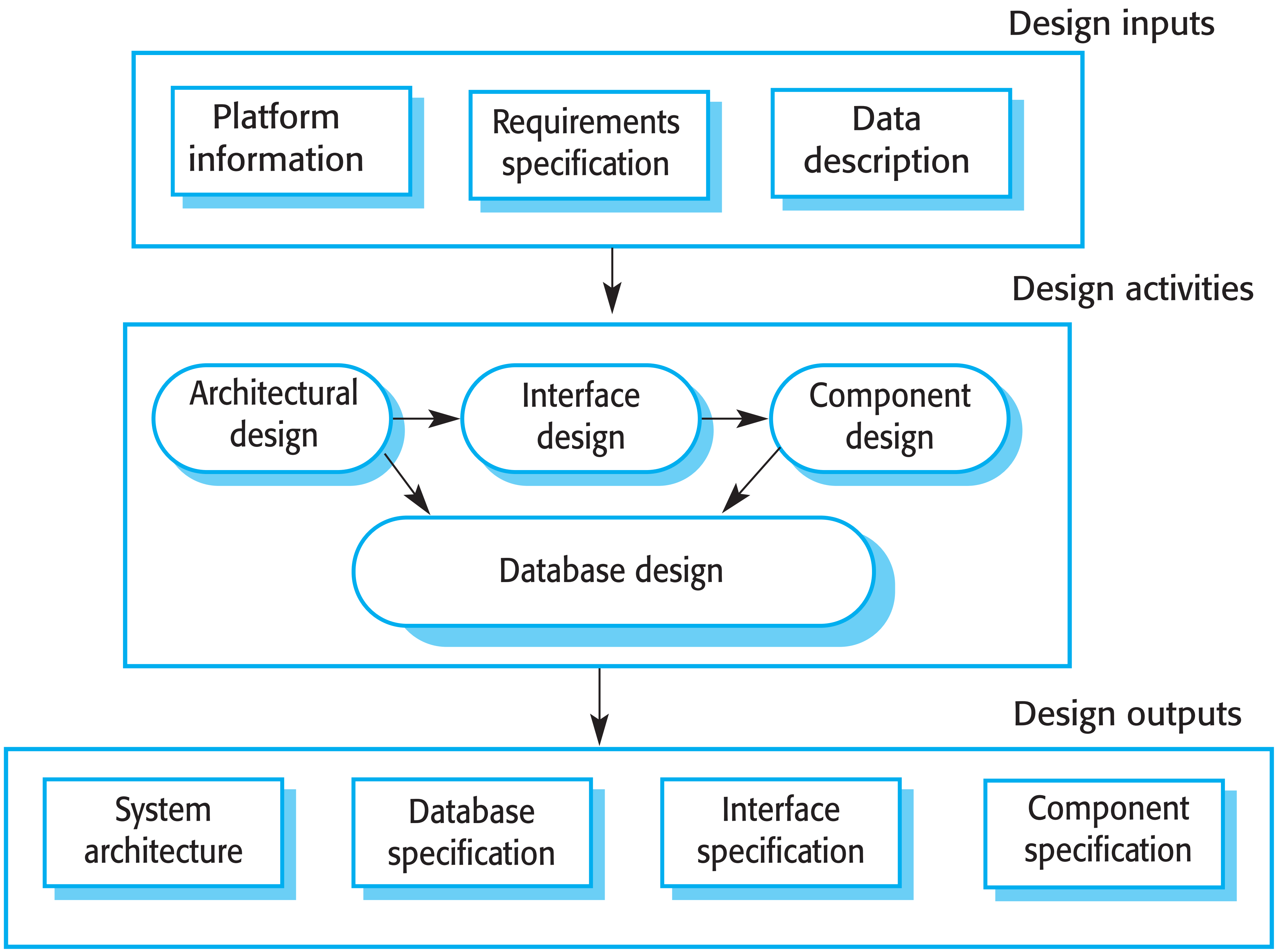
Requirements Engineering


What do the customers want?
Add detail
Check that is what they really want
Requirements elicitation and analysis
Requirements specification
Requirements validation
System descriptions
User and system requirements
Requirements document
Documentation

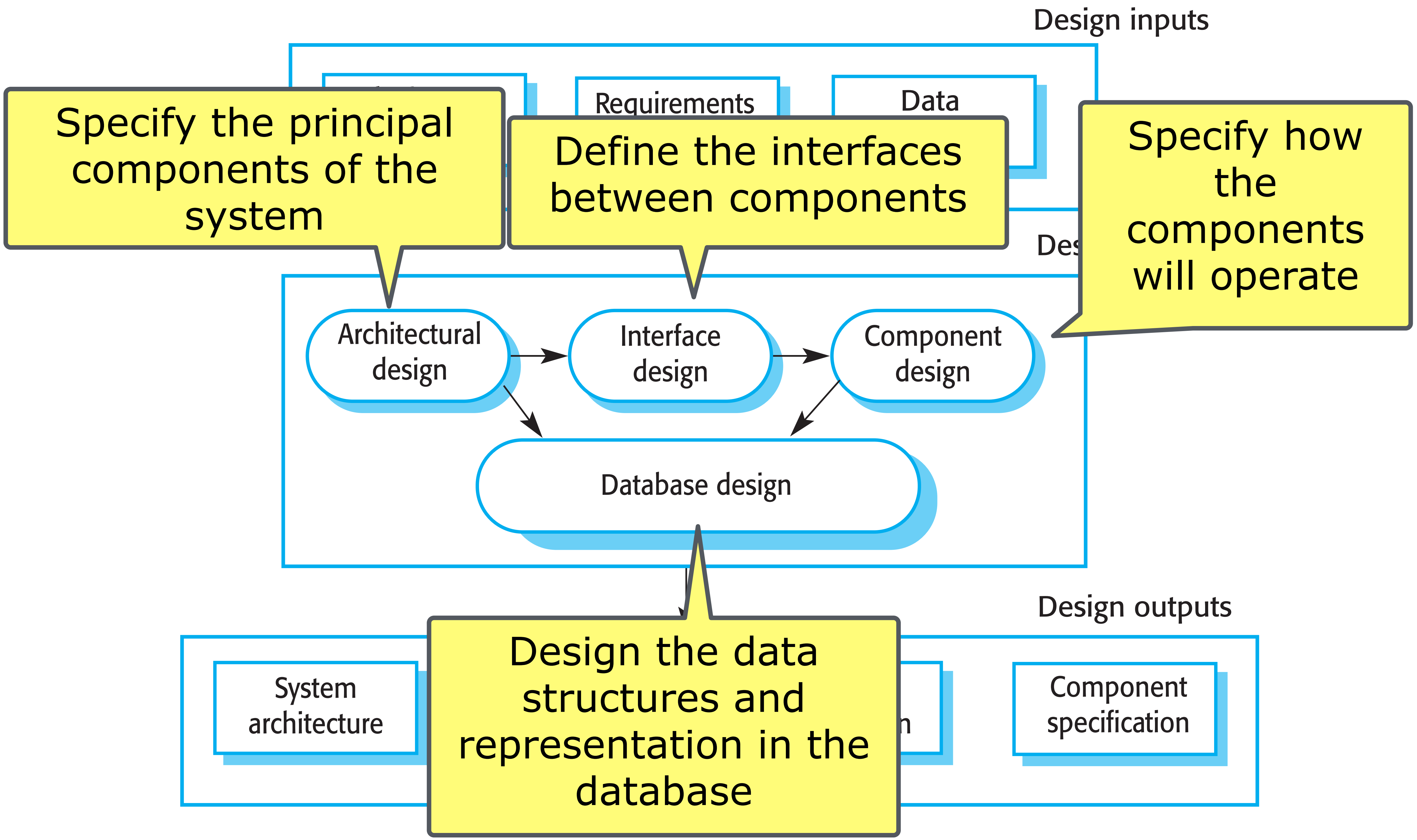
Software Design and Implementation

- … is the process of converting the system specification into an executable system.

Software Design and Implementation



Software Design and Implementation


Design inputs
Requirements
Data
Specify the principal components of the system
Define the interfaces between components
Specify how the components will operate
Des
Architectural design
Interface design
Component design
Database design
Design the data structures and representation in the database
Design outputs
System architecture
n
Component specification

System implementation

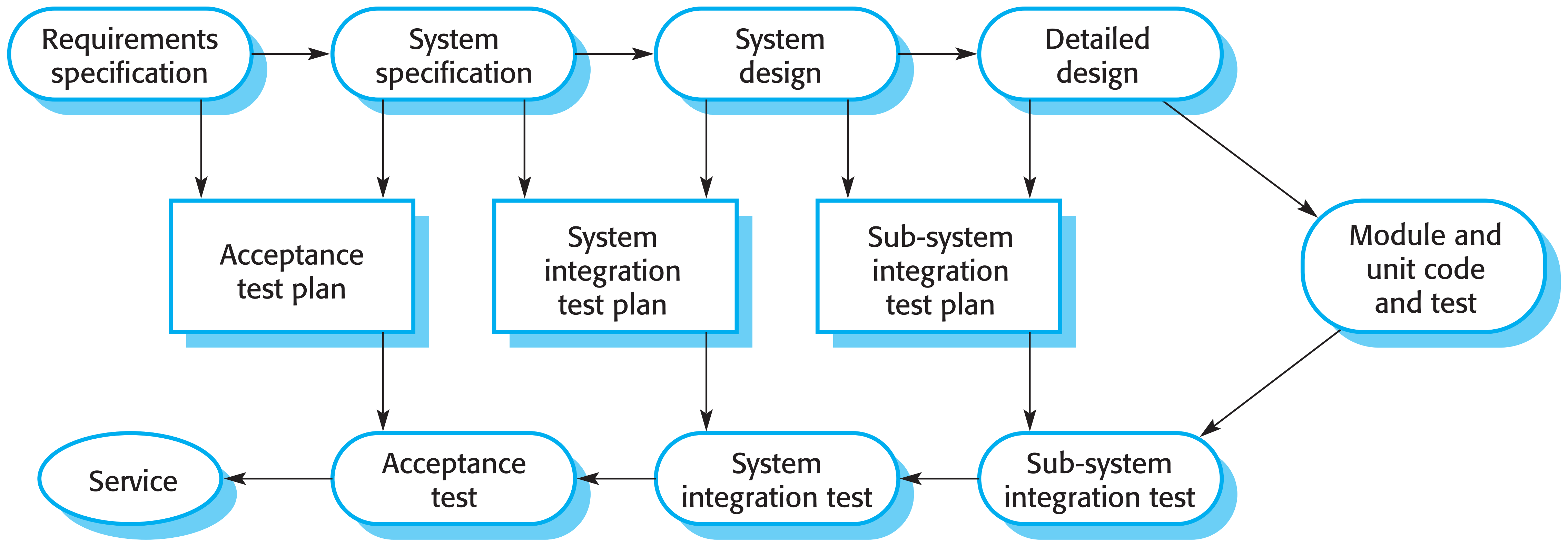
- The software is implemented either by developing a program or programs or by configuring an application system.
- Design and implementation are interleaved activities for most types of software system.
- Programming is an individual activity with no standard process.
- Debugging is the activity of finding program faults and correcting these faults.

Software Validation

- Verification and validation (V&V) is intended to demonstrate that the system conforms to the the specification and meets the requirements of the customer.
- Testing is the main V&V activity.

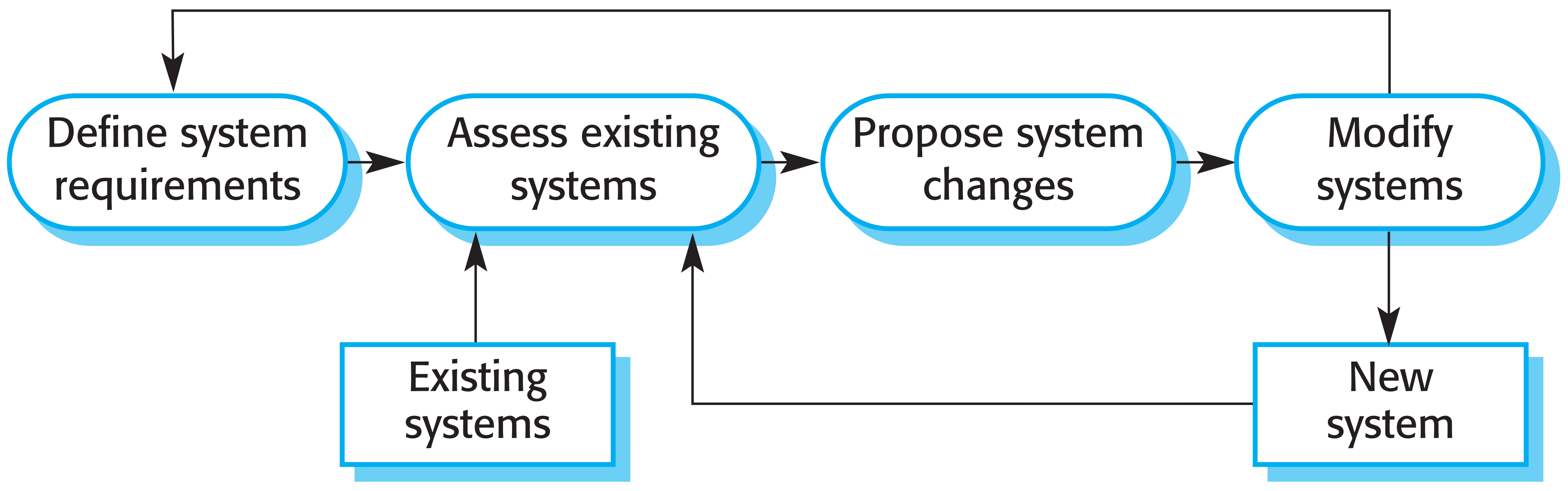
Testing Phase

- In testing, you should check that the product matches the specification, independently of the code (i.e., spec -> test).


Requirements specification
System specification
System design
Detailed design
Acceptance test plan
System integration test plan
Sub-system integration test plan
Module and unit code and test
Service
Acceptance test
System integration test
Sub-system integration test

Software Evolution

- Aka Operations and Maintenance
- Bug fixes and new features

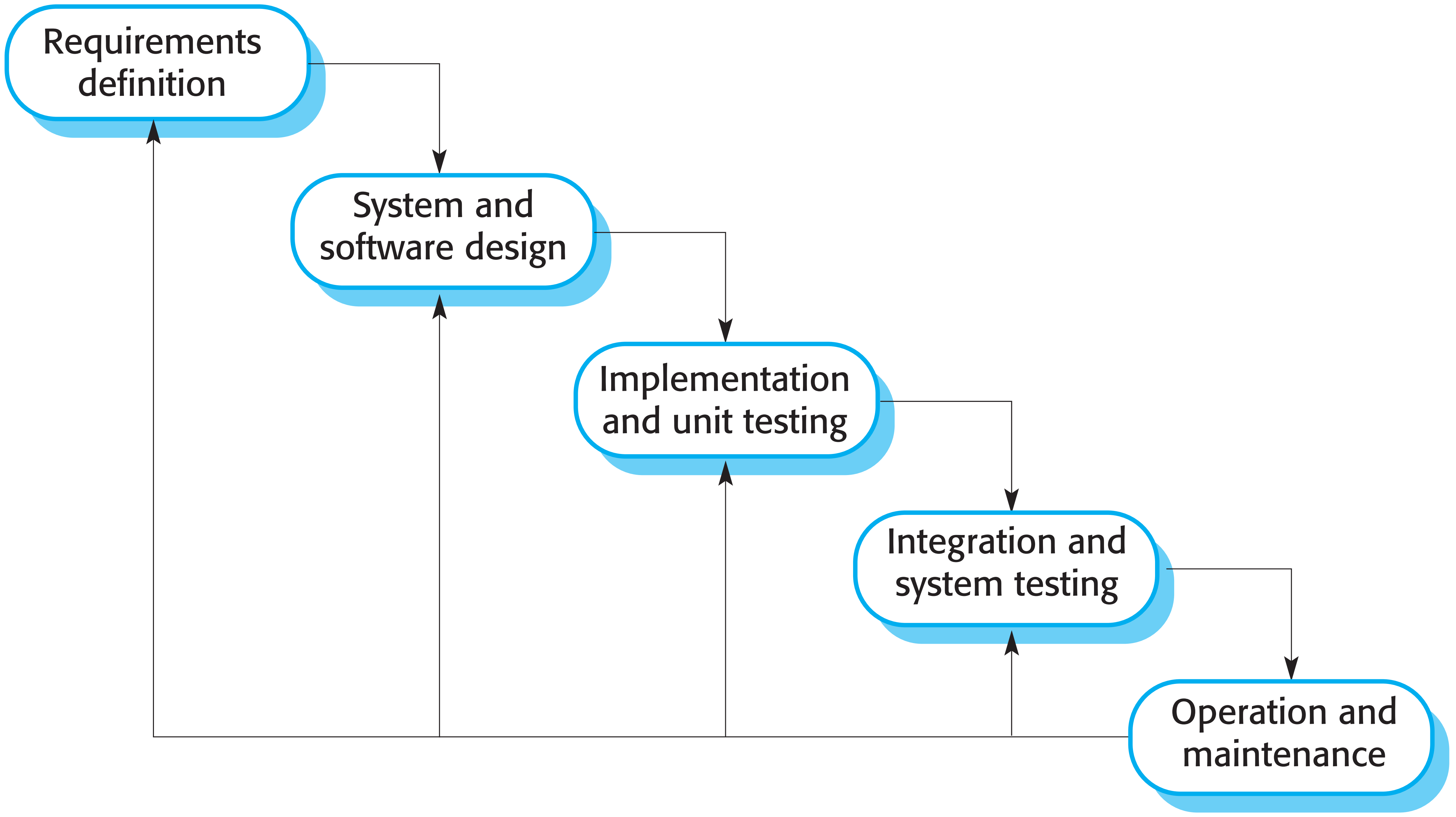


Software Process Models

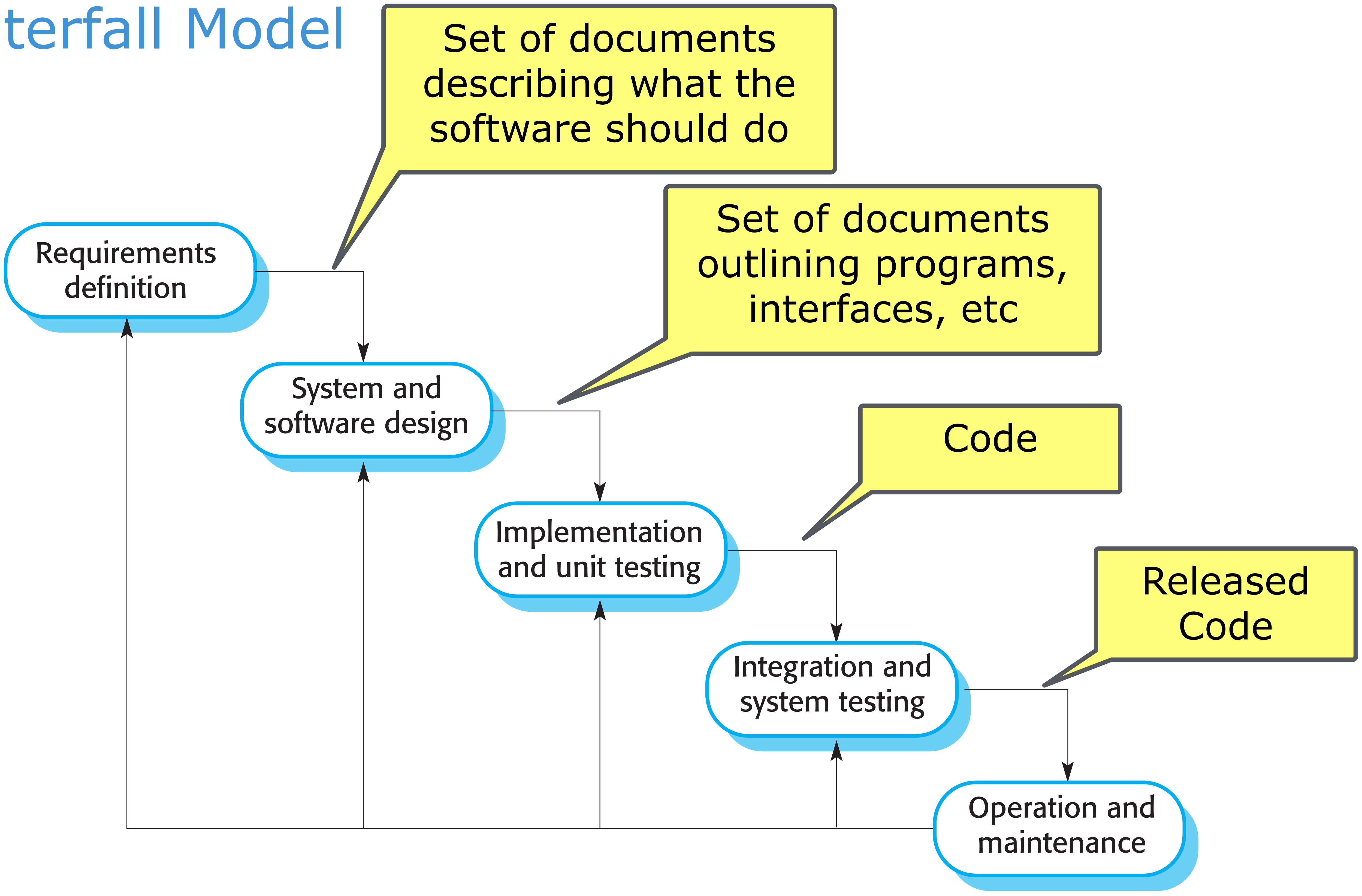
Software Process Model

- A **software process model** (or SDLC) is a simplified representation of a software development process.
  1. **Waterfall model**: views the activities as separate and sequential
  2. **Incremental development**: interleaves the activities, the system is built in a series of versions
  3. **Integration and configuration**: configuring reusable components for a new setting and integrating them into a system

Waterfall Model


Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

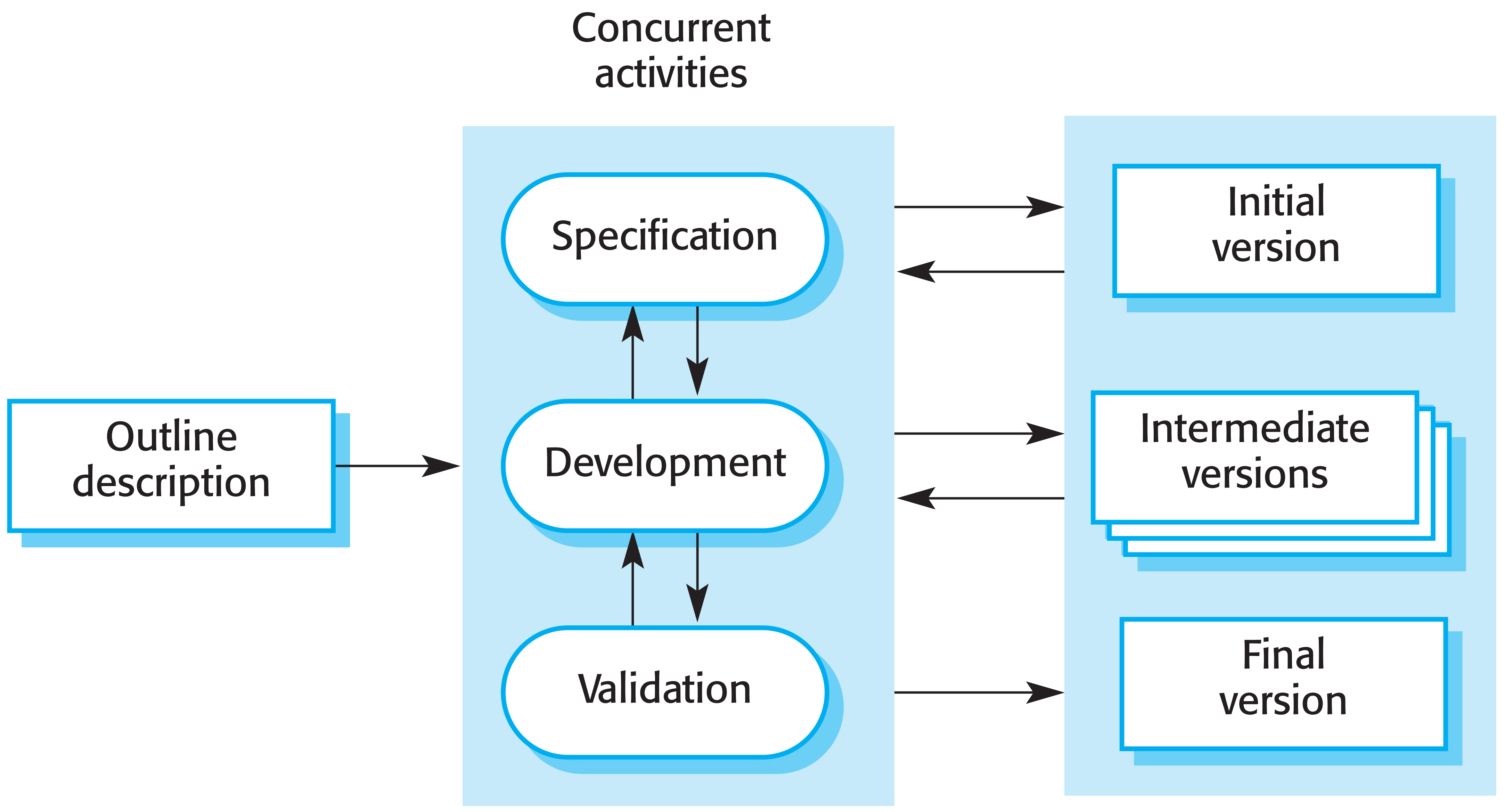
Waterfall Model


Set of documents describing what the software should do
Set of documents outlining programs, interfaces, etc
Code
Released Code
Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Waterfall Model

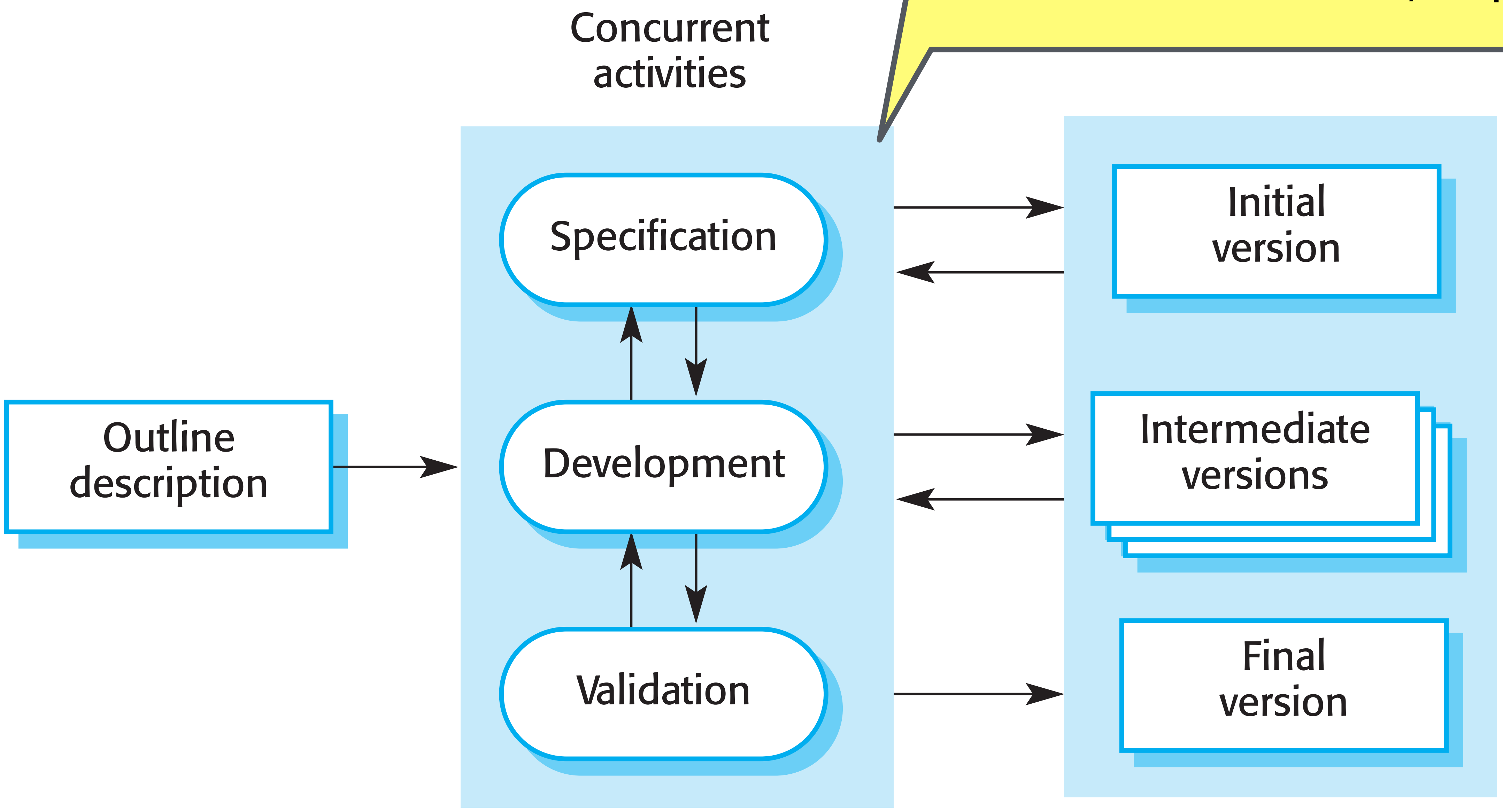
- Works well for predictable, well understood projects.
- Breakdowns down when a large problem is found late and the team has to go back to re-specify or re-design the software.

Incremental Development



Incremental Development

Design and build a few features, check that the customer likes it, repeat



Incremental Development

- Works well when either what the customer wants or how best to build it are unclear at the outset (which is most of the time).
- Frequent checks that the customer is happy with the product.
- Can quickly respond to changing requirements.
- Can lead to a lot of tidying up towards the end of the project.

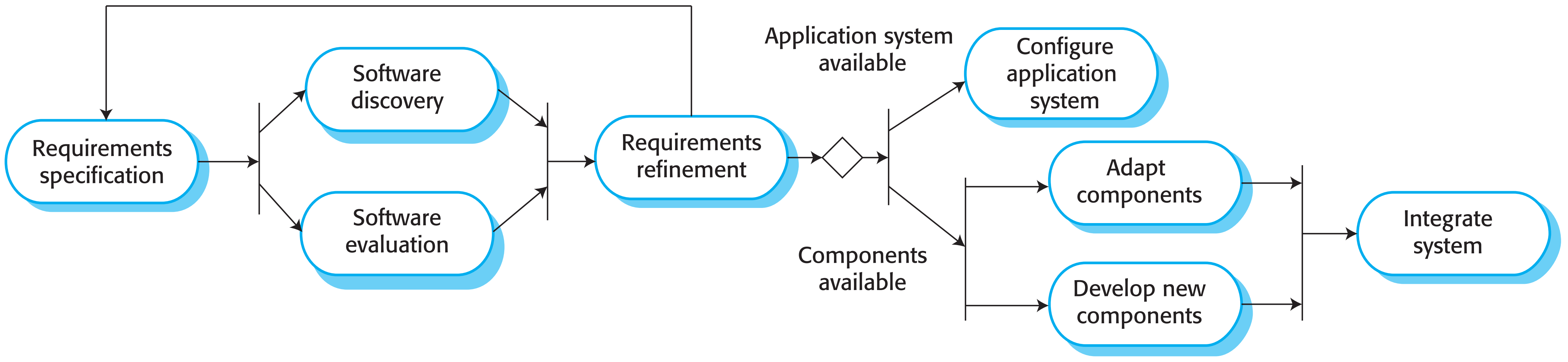
Incremental Development

- Agile methods are a form of incremental development.
- A planned agile approach is now the most common software development methodology.

Integration and Configuration

- Stand-alone application systems that are configured for use in a particular environment.
- Collections of objects that are developed as a package to be integrated with a component framework (e.g .NET or J2EE)
- Web services that are developed according to service standards and which are available for remote invocation.

Integration and Configuration



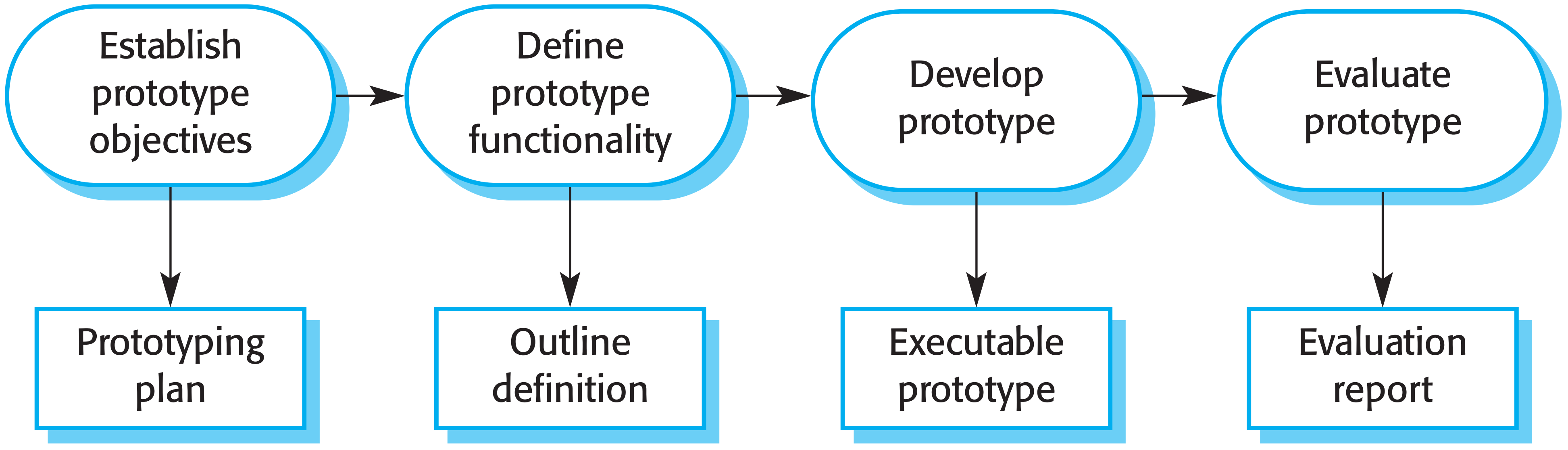
Integration and Configuration

- Cheap and fast.
- Low risk.
- Only viable when there is a market for a large number of very similar systems.
- Products tend to be clunky because there are not custom designed (one size fits all).

Prototyping

- A prototype is an initial version of a system used to demonstrate concepts and try out design options.
- A prototype can be used in:
  - The requirements engineering process to help with requirements elicitation and validation
  - In design processes to explore options and develop a UI design
- Early checks that what you are building is what the customer wants.

Prototyping Process



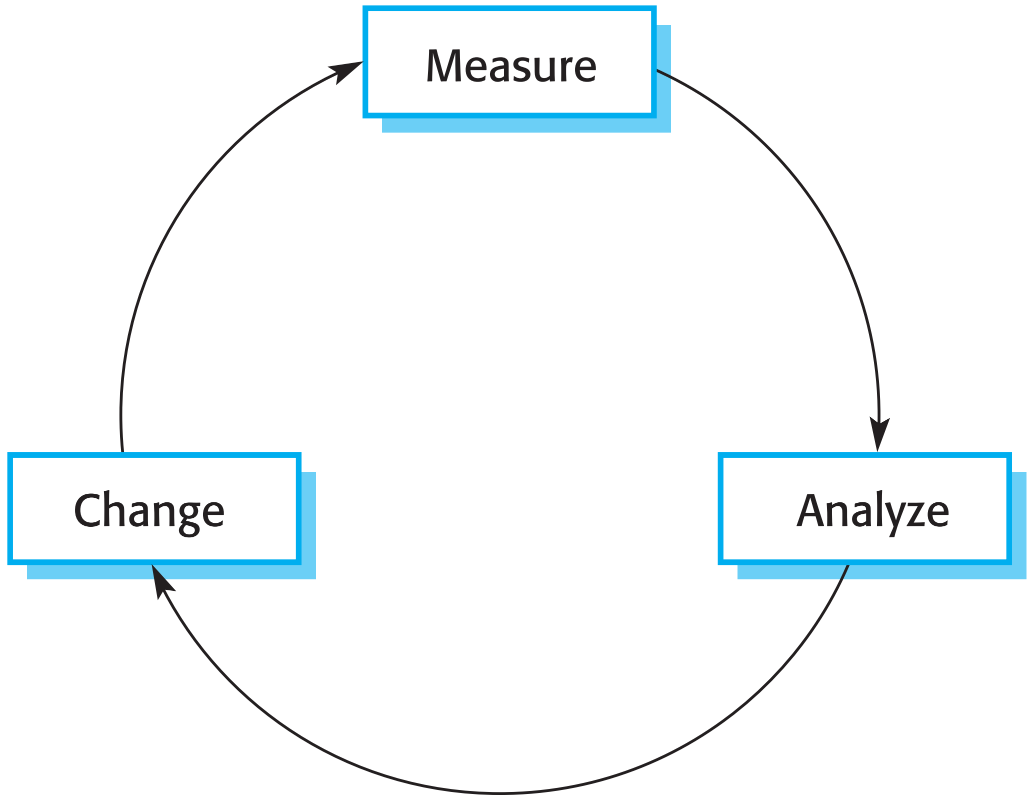
Prototyping

- Rapid prototyping tools or languages.
- Throw-away prototypes
- Incremental delivery builds in prototyping.

Process Improvement

Process Improvement

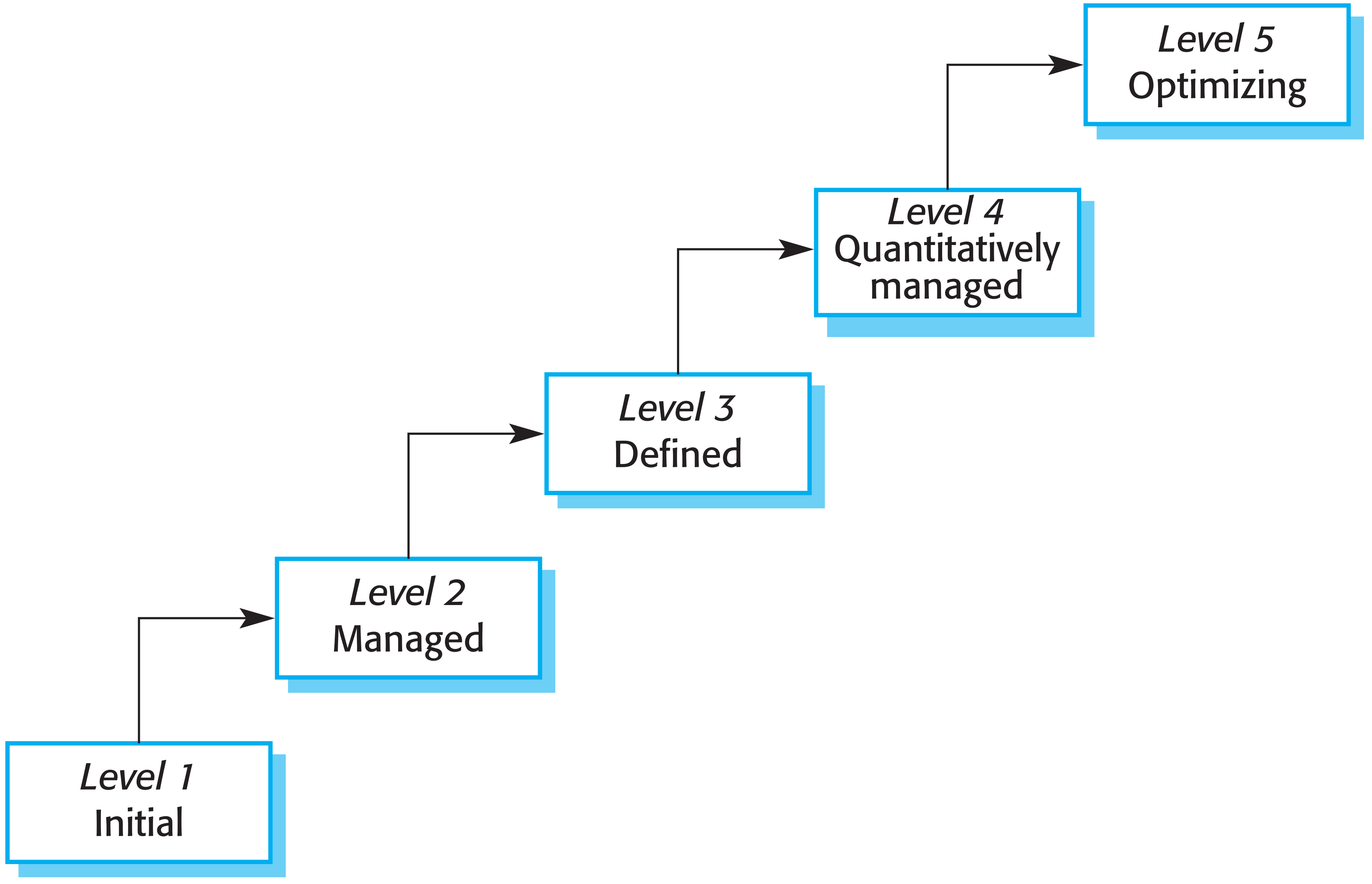
- … is reviewing how projects went and enhancing the software engineering process for future projects.
- The process improvement cycle:



Process Metrics

- Time taken for process activities to be completed
  - E.g. Calendar time or effort to complete an activity or process.
- Resources required for processes or activities
  - E.g. Total effort in person-days.
- Number of occurrences of a particular event
  - Number of defects discovered.

Capability Maturity Levels



Summary

- Software Engineering is needed to develop software at scale
- Software Development Lifecycle (SDLC) are the main activities that need to be done.
- Software Process Models describe the sequencing and relationships between those activities
- Process Improvement reviews projects and puts in place recommendations for improving processes.