CSE 331: Software Engineering

Lecture 2

Software Processes

Software Process

Fundamental Assumption:

Good processes lead to **good software**

Good processes reduce **risk**

Good processes enhance **visibility**

Variety of Software Processes

Software products are very varied...

Therefore, there is no standard process for all software engineering projects

BUT successful software development projects all need to address similar issues.

This creates a number of **process steps** that must be part of all software projects

Basic Process Steps in all Software Development

- Feasibility and planning
- Requirements
- Design
- Implementation
- Acceptance and release
- Operation and maintenance

It is essential to distinguish among these aspects and to be clear which you are doing at any given moment.

Do not confuse requirements and design.

Feasibility and Planning

A **feasibility study** precedes the decision to begin a project.

- What is the scope of the proposed project?
- Is the project technically feasible?
- What are the projected benefits?
- What are the costs, timetable?

A feasibility study leads to a **decision**: go or no-go.

Requirements Analysis and Definition

The **requirements** analysis and definition establish the system's services, constraints and goals by consultation with users. They are then **defined** in a manner that is understandable by both users and development staff.

This phase can be divided into:

- Requirements analysis
- Requirements definition
- Requirements specification

Requirements define the function of the system FROM THE CLIENT'S VIEWPOINT.

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

System and Program Design

System design: Partition the requirements to hardware or software systems. Establishes an overall system architecture

Software design: Represent the software system functions in a form that can be transformed into one or more executable programs

- Unified Modeling Language (UML)

The design describes the system FROM THE SOFTWARE DEVELOPERS' VIEWPOINT

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.

Design activities

- *Architectural design,* where you identify the overall structure of the system, the principal components (sometimes called sub-systems or modules), 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.

Implementation

Programming

The software design is realized as a set of programs or program units. (Written specifically, acquired from elsewhere, or modified.)

Testing

Individual components are tested against specifications.

The individual program units are integrated and tested *against the design* as a complete system.

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.

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.

Acceptance and Release

Acceptance

The complete system is tested *against the requirements by the client*.

Delivery and release

The complete system is delivered to the client and released into production.

Operation and Maintenance: Software Life Cycle

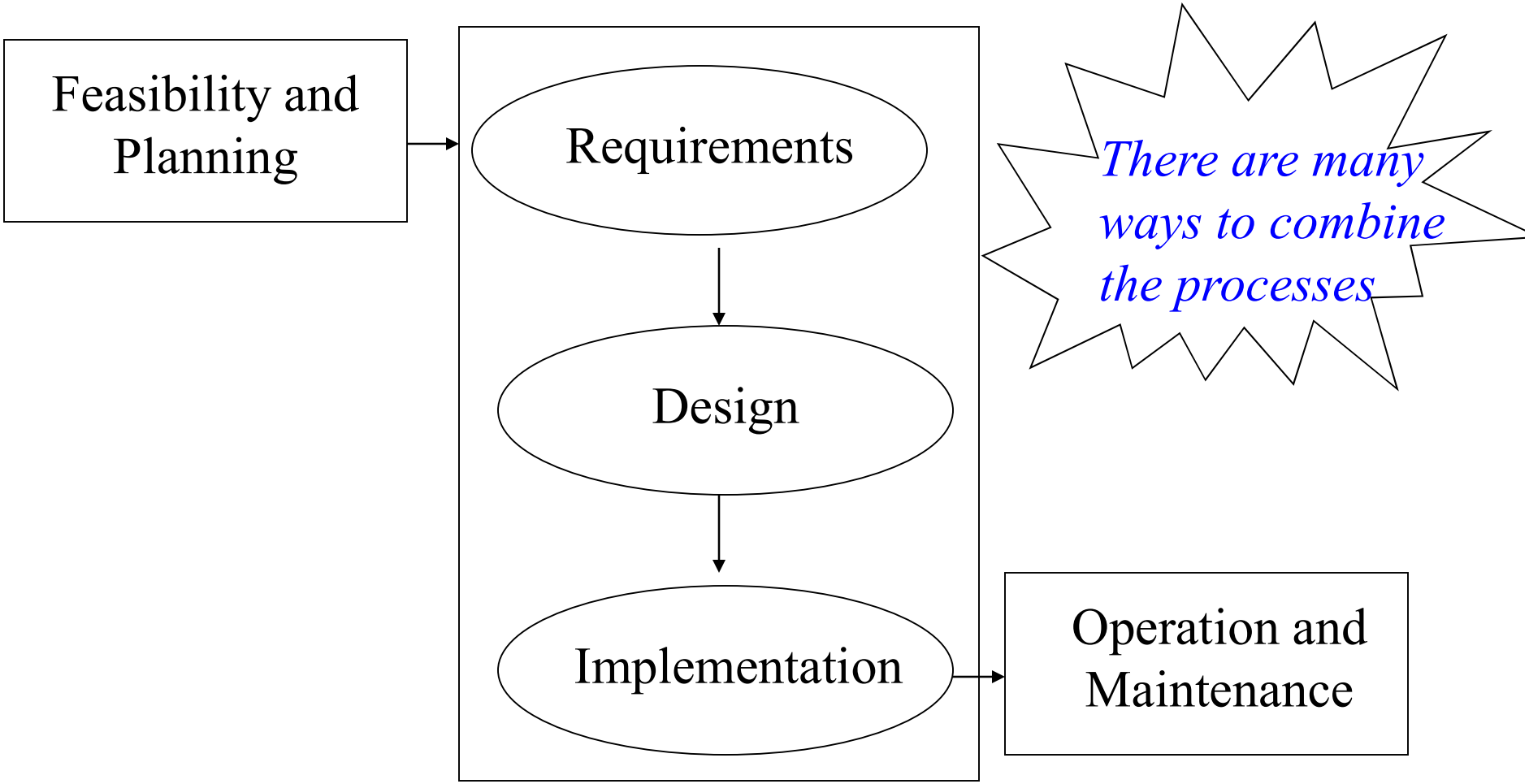
Operation: The system is put into practical use.

Maintenance: Errors and problems are identified and fixed.

Evolution: The system evolves over time as requirements change, to add new functions or adapt the technical environment.

Phase out: The system is withdrawn from service.

Combining the Process Steps



Sequence of Processes

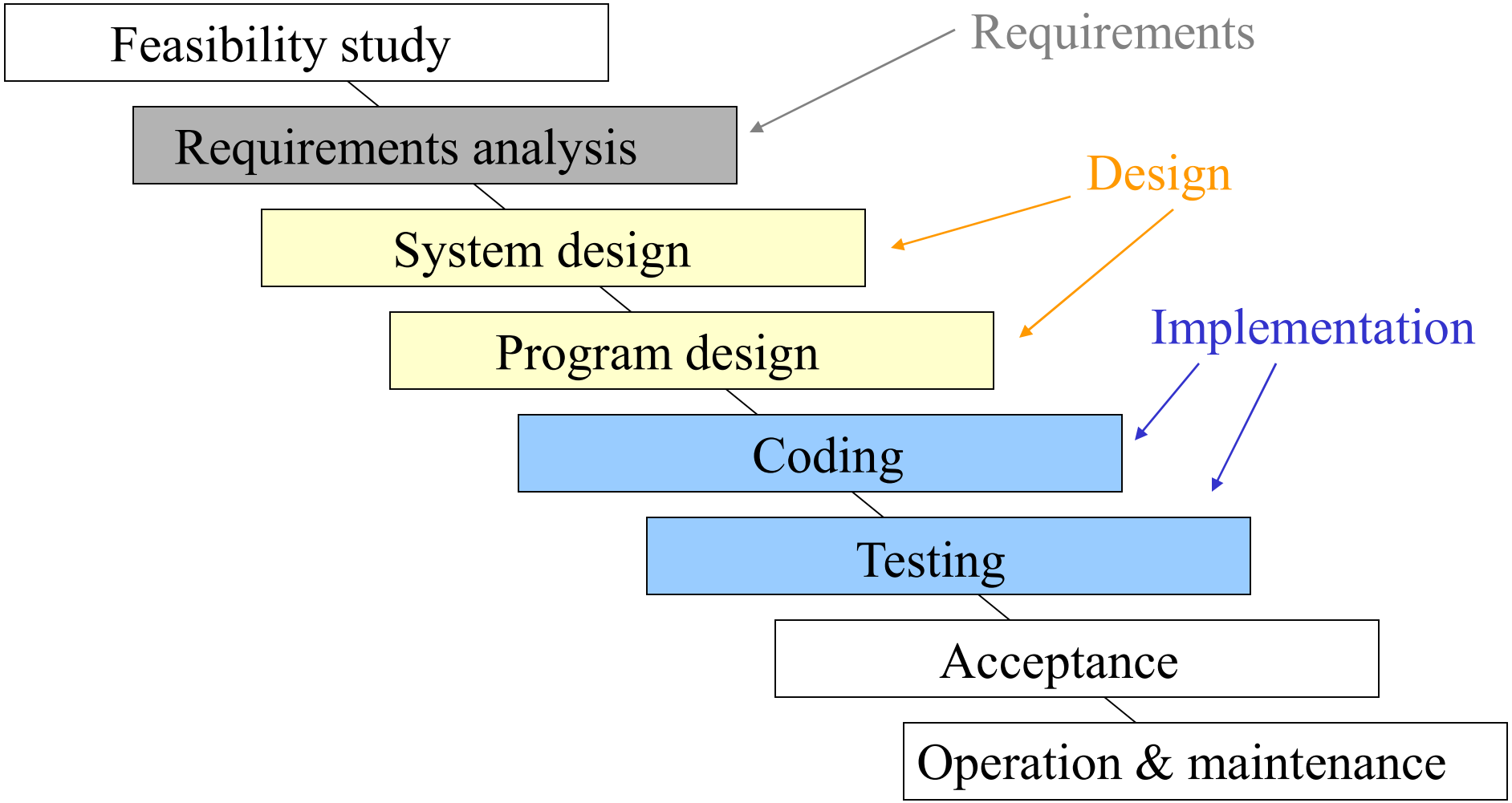
Every software project will include these basic processes, **in some shape or form**, but:

- They may be formal or informal
- They may be carried out in various sequences

Examples:

- A feasibility study cannot create a proposed budget and schedule without a preliminary study of the requirements and a tentative design.
- Detailed design or implementation usually reveals gaps in the requirements specification.

Process 1: Sequential
The Waterfall Model



Discussion of the Waterfall Model

Advantages:

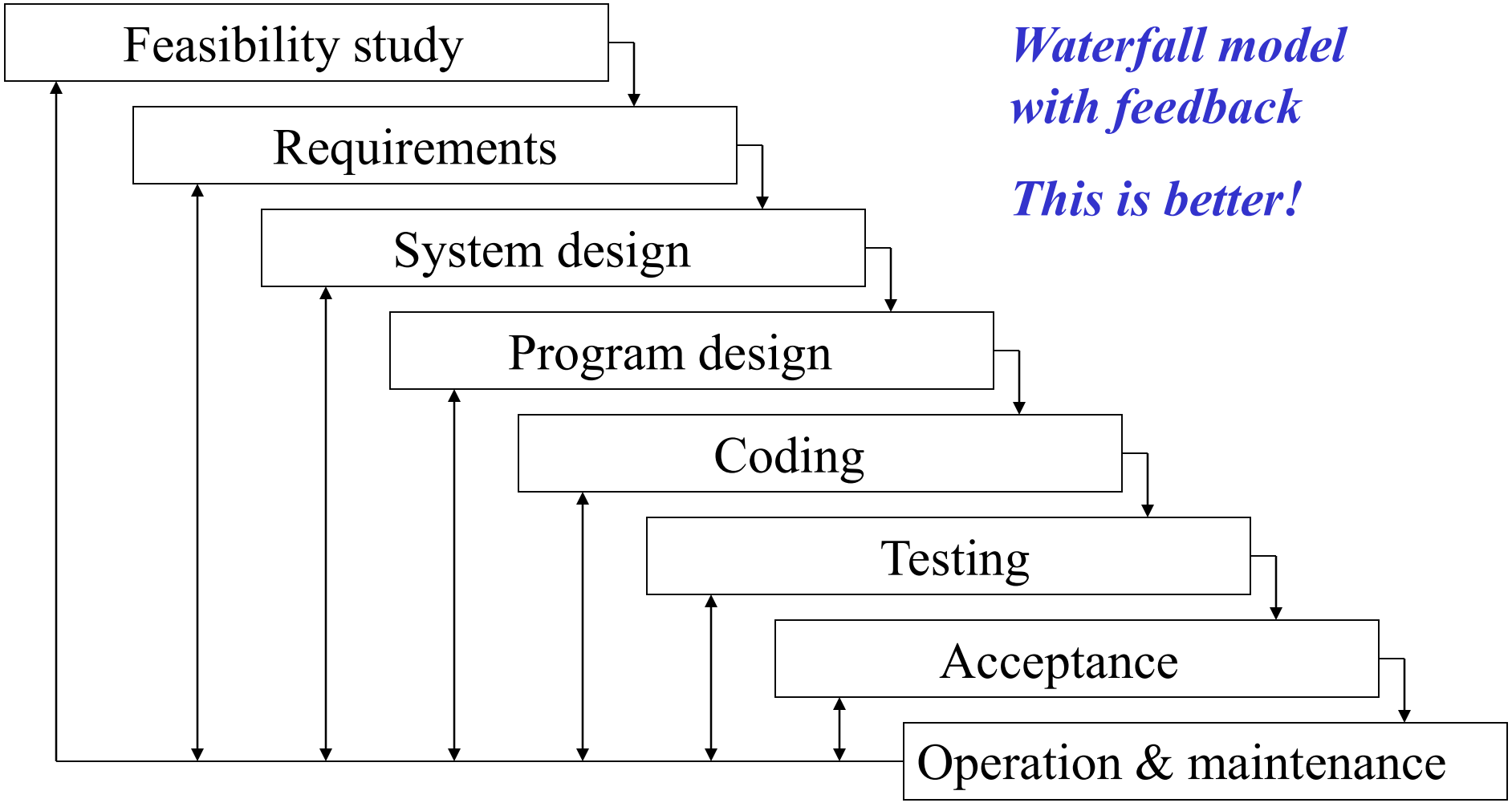
- Process visibility
- Separation of tasks
- Quality control
- Cost control

Disadvantages:

Each stage in the process reveals new understanding of the previous stages, that requires the earlier stages to be revised.

The Waterfall Model is not enough!

Modified Waterfall Model



Waterfall model with feedback

This is better!

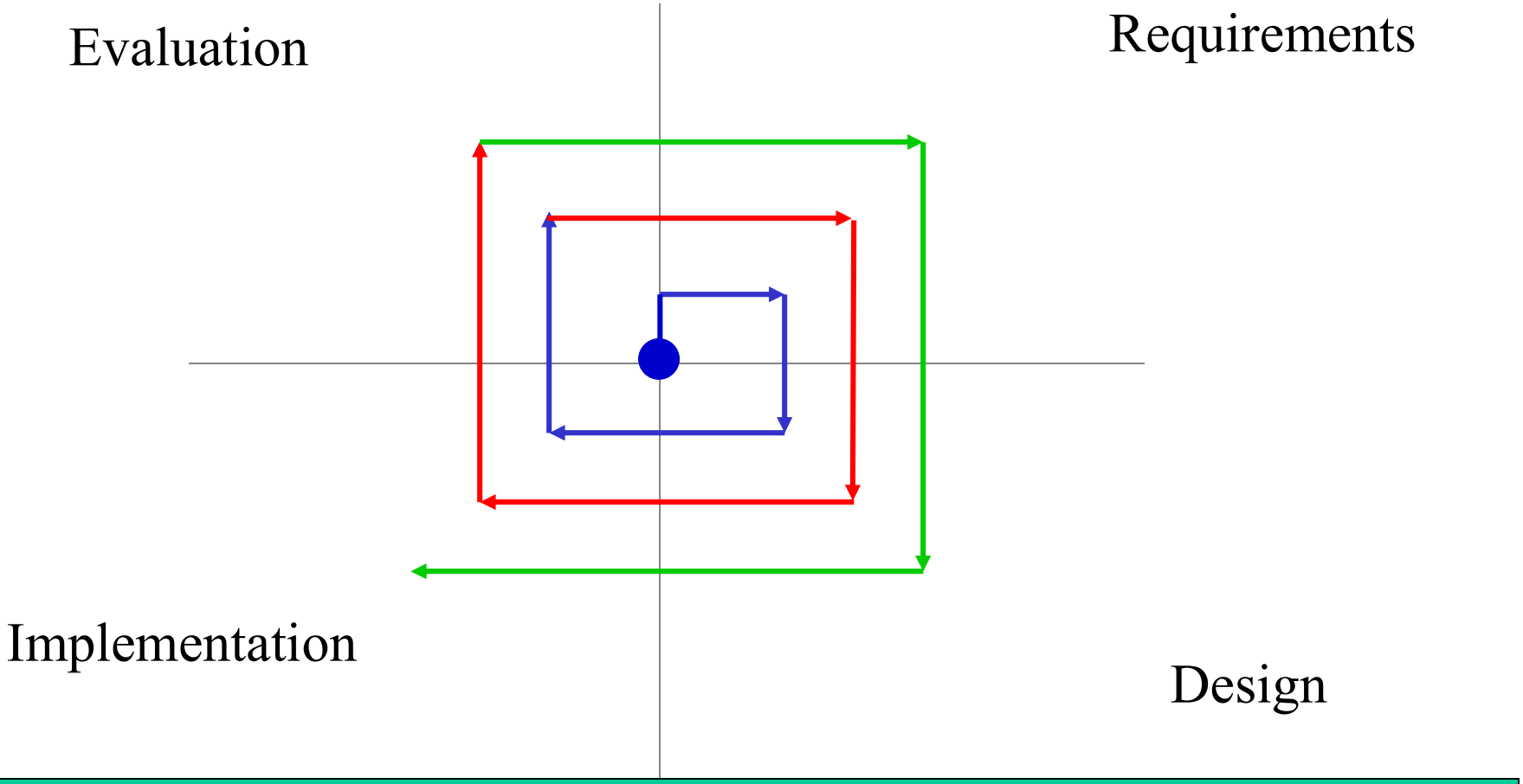
Process 2: Iterative Refinement (Evolutionary Development)

Concept: Initial implementation for user comment, followed by refinement until system is complete.

- Vaporware: user interface mock-up
- Throw-away software components
- Dummy modules
- Rapid prototyping
- Successive refinement

Get something working as quickly as possible!

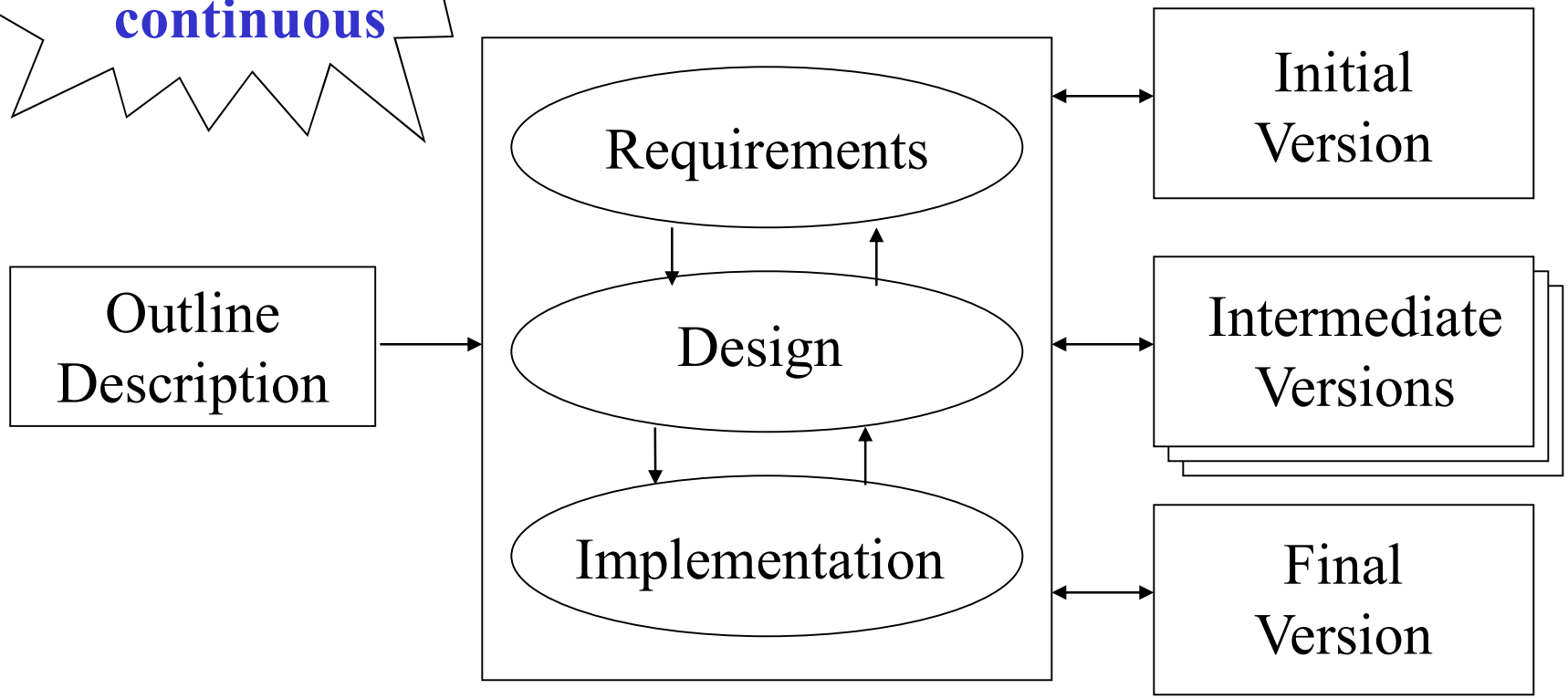
Iterative Refinement



Iterative Refinement

The feasibility study is continuous

Concurrent Activities



Process 3: Phased Development

Concept

A simple system with basic functionality is brought quickly into production (Phase 1).

Subsequent phases are based on experience gained from users of each previous phase.

Advantages

- Pay-back on investment begins soon.
- Requirement are more clearly understood in developing subsequent phases

Iterative Refinement + Waterfall Model: Graphics for Basic

Outline Description: Add vector graphics to Dartmouth Basic.

Phase 1: Extend current language with a preprocessor and run-time support package. (1976/77)

Phase 2: Write new compiler and run-time system incorporating graphics elements. (1978/80)

Iterative Refinement + Waterfall Model: Graphics for Basic

Phase 0: Iterative Refinement

Design Issues:

- Pictorial subprograms: coordinate systems, window/viewport
- User specification of perspective

Design Strategy: (Iterative Refinement)

- Write a series of prototypes with various proposed semantics
- Evaluate with a set of programming tasks

Iterative Refinement + Waterfall Model: Graphics for Basic

Phase 1: Implementation

- When the final specification was agreed, the entire preprocessor and run-time support were coded from new.
- The system was almost entirely bug-free.

Phase 2: New compiler (Waterfall)

Phase 1 was used as the requirements definition for the final version.

Observations about Software Processes

Completed projects should have the basic process steps *but* ... the development process is always partly evolutionary.

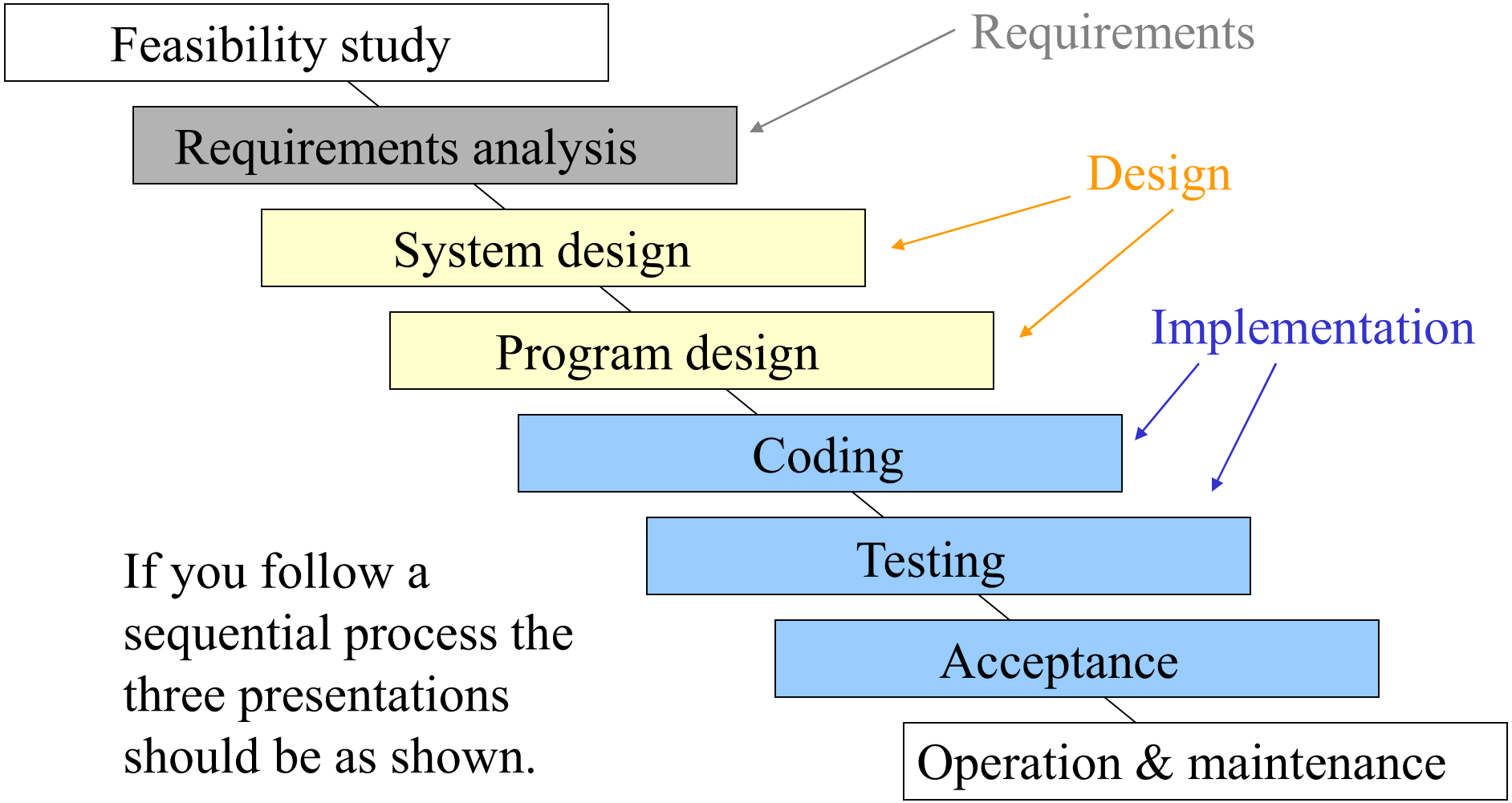
Risk is lowered by:

- **Prototyping** key components
- Dividing into **phases**
- Following a **visible** software process
- Making use of reusable **components**

Conclusion

It is not possible to complete each step and *throw it over the wall.*

Project Presentations: Sequential Option



If you follow a sequential process the three presentations should be as shown.

Project Presentations: Iterative Option