Cornell University
Computing and Information Science

CS 5150 Software Engineering

3. Examples of software development processes

William Y. Arms

Definitions: Activity and Sprint

Activity

An activity is a general term for any part of a project that takes place over time (also known as a **task**)

- Each step in the software development process can be broken down into several activities.

Sprint

A sprint is a set period of time during which a team completes part of a software project.

- Each sprint will go through most or all of the process steps.

- A typical sprint might have a team of 6 to 8 people working for 2 to 4 weeks.

Iterative Refinement

Concept

- Create a **prototype** system early in the development process.

- Review the prototype with clients and test it with users, to improve the understanding of the requirements and clarify the design.
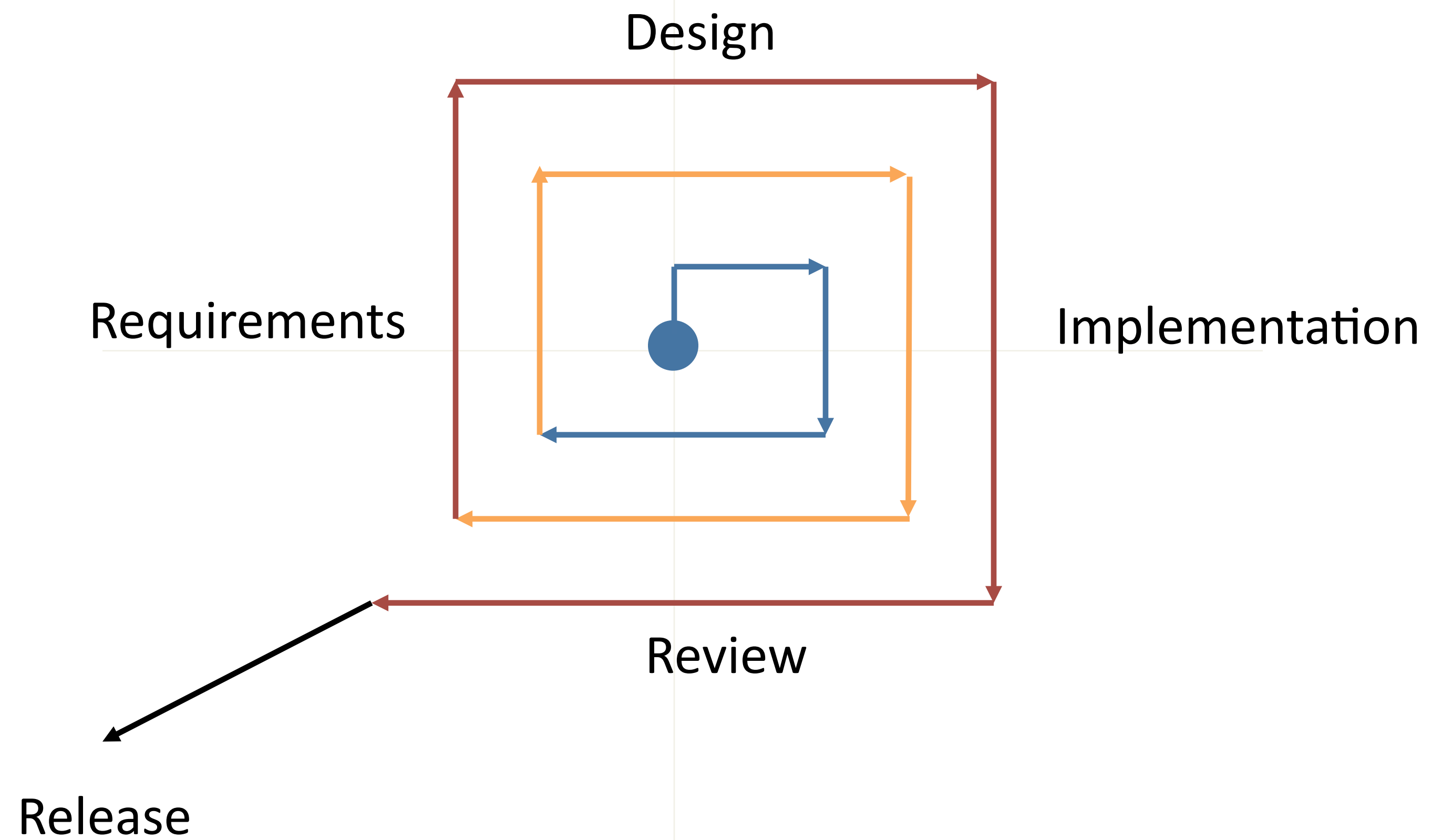
- Refine the prototype in a series of iterations.

Requirements are hard to understand until there is an operational system, particularly with user interfaces.

Mistakes in the requirements are the most expensive to correct.

Example:

- Converting a national archive from paper based to computer based.

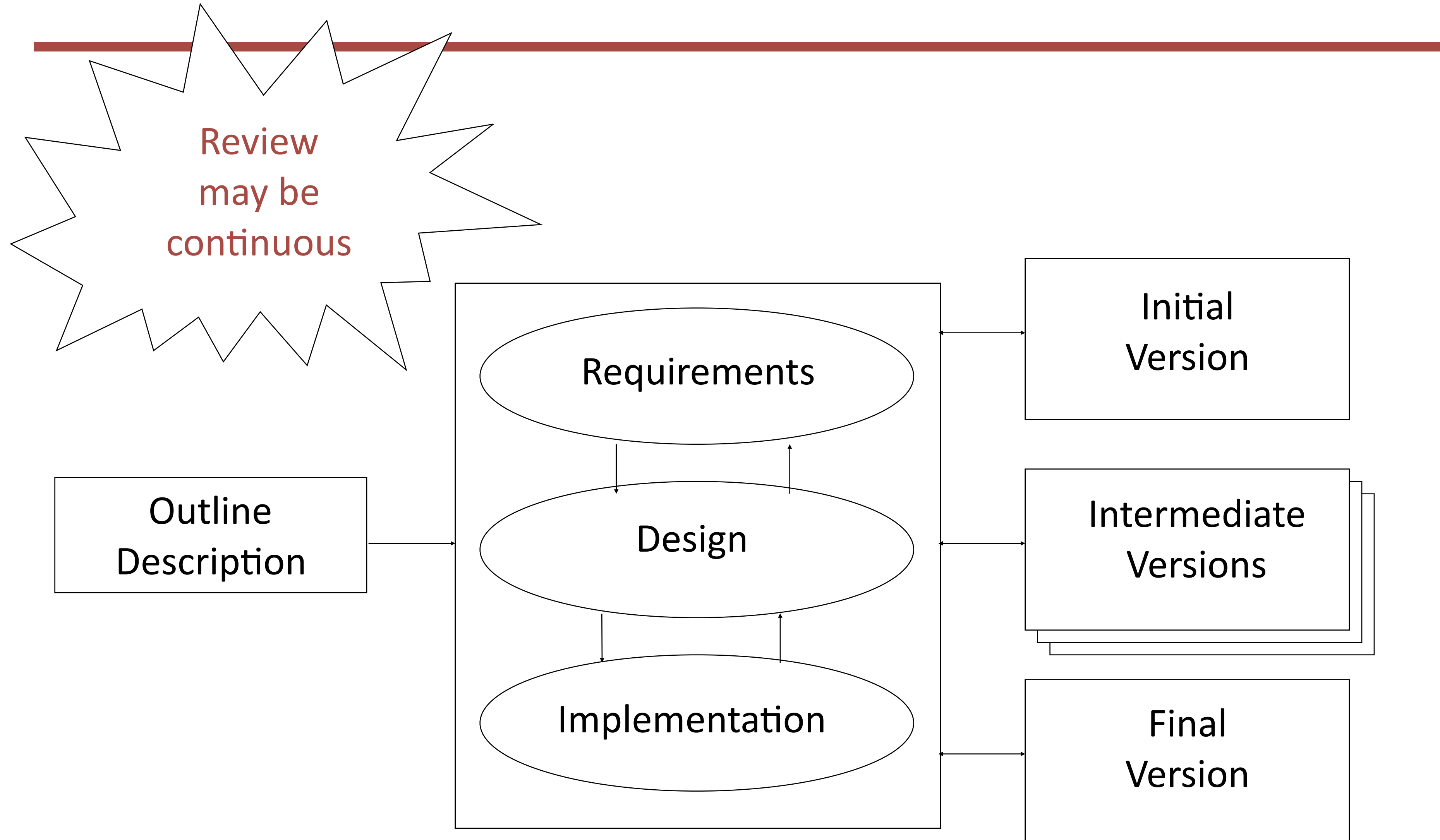# Iterative Refinement

Discussion of Iterative Refinement

This is a medium weight process with documentation created during the process.

Iterative refinement uses various techniques that enable the client to review the the planned system early during development:

- User interface mock-up

- Throw-away software components

- Dummy modules

- Rapid prototyping

- Successive refinement

Get something working as quickly as possible, for client and user evaluation, but do not release it.

Iterative Refinement with a Large System

Review may be continuous

Outline Description

Requirements

Design

Implementation

Initial Version

Intermediate Versions

Final Version

Spiral Development

Spiral development

- Create a base system that has the overall structure of the final product with dummy stubs for missing components.

- Create a comprehensive set of test cases for all completed components

- Use a succession of sprints to develop new or improved components, each with a set of test cases. Add these components to the source code library.

- On a daily cycle, build the entire system from the source code library and run the complete set of test cases.

With spiral development there is always a fully tested system, but the functionality is incomplete.

Example

Developing a new version of an operating system.

Spiral Development
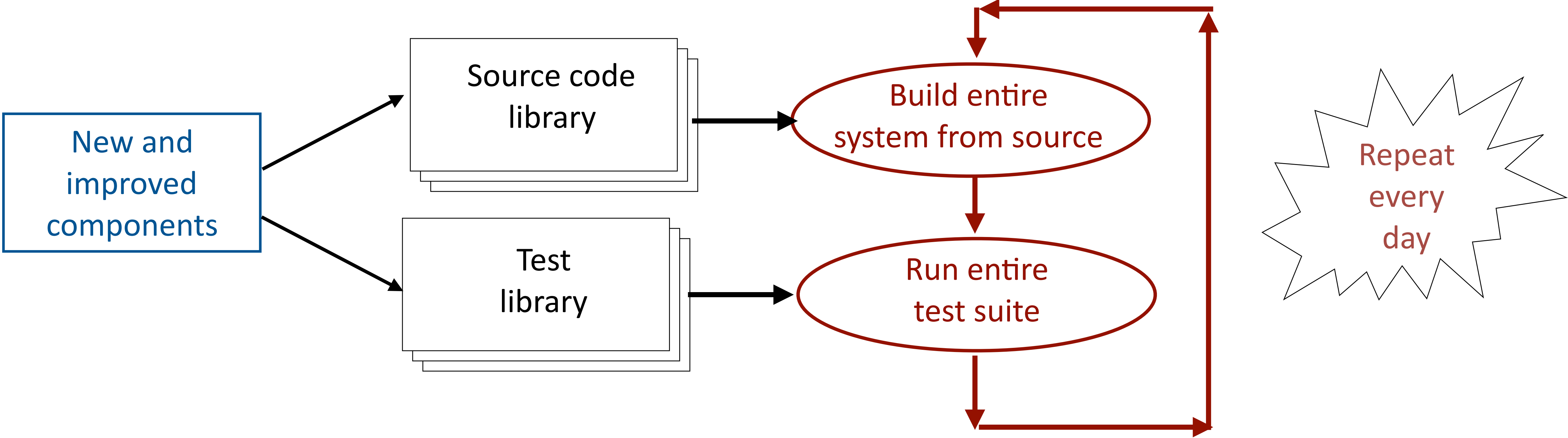
Incremental Release of Online Systems

When **software is released online** it is often possible to divide the development into a sequence of sprints that are developed and released in quick succession.

Example:

• Start-up company developing a web based shopping service.

Advantages

• Pay-back on investment begins soon.

• Requirement are more clearly understood in developing subsequent sprints – minimize waste.

• Feedback from customers and clients can be incorporated in later phases.

It is easier for a small team to develop a small sprint correctly than to coordinate large projects with many ramifications.

Agile Development (Original Version with Immediate Release)
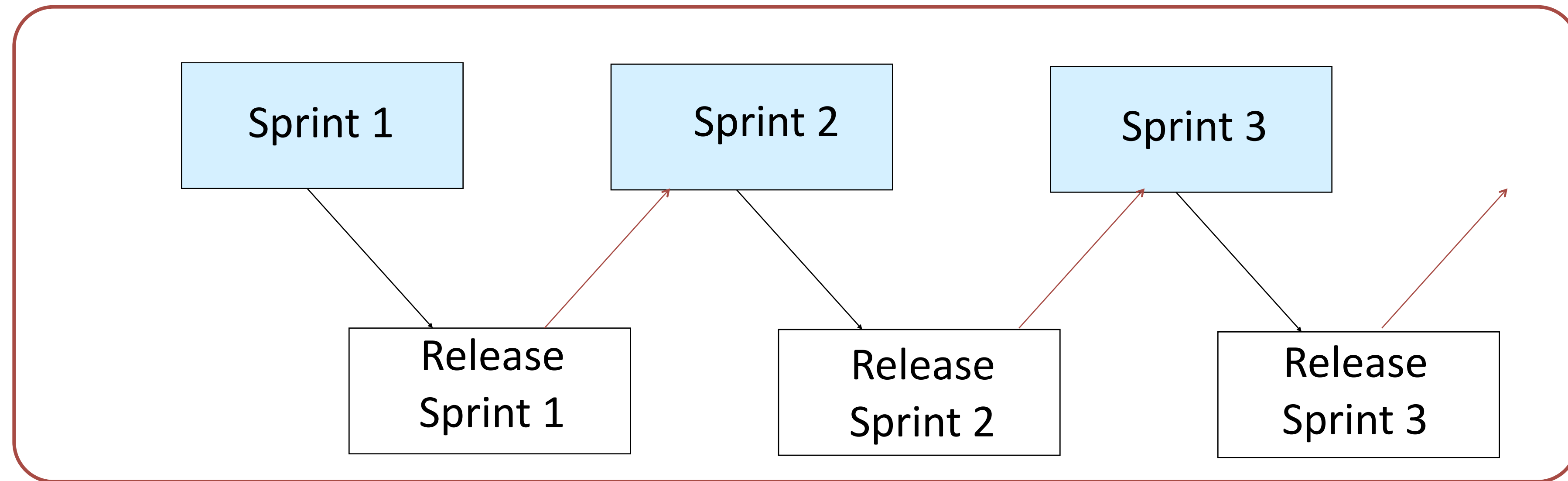
- The project is divided into a large number of small tasks, known as **sprints**.

- For each sprint, a team works through a full software development cycle including planning, requirements analysis, design, coding, testing, and acceptance testing, and release.

- Each sprint is completed in a fixed time period, e.g., four weeks.

- The size of an sprint is based on team size, e.g., 5-10 people.

Discussion of Agile Development

Variations on agile software development

In practice it is rarely possible for every sprint to end with released software, but software development based on sprints has many advantages.

Modern software development includes a wide range of processes that are called "agile". Other processes with names such as "heroic programming" or "scrum" use many of the same concepts.

Characteristics

- Development of a project is divided into a large number of sprints.

- Each sprint ends with fully tested code.

- This is a lightweight process with minimal documentation created during the process.

A general definition (from Wikipedia)

Agile software development describes a set of principles for software development under which requirements and solutions evolve through the collaborative effort of self-organizing cross-functional teams.

Agile Development
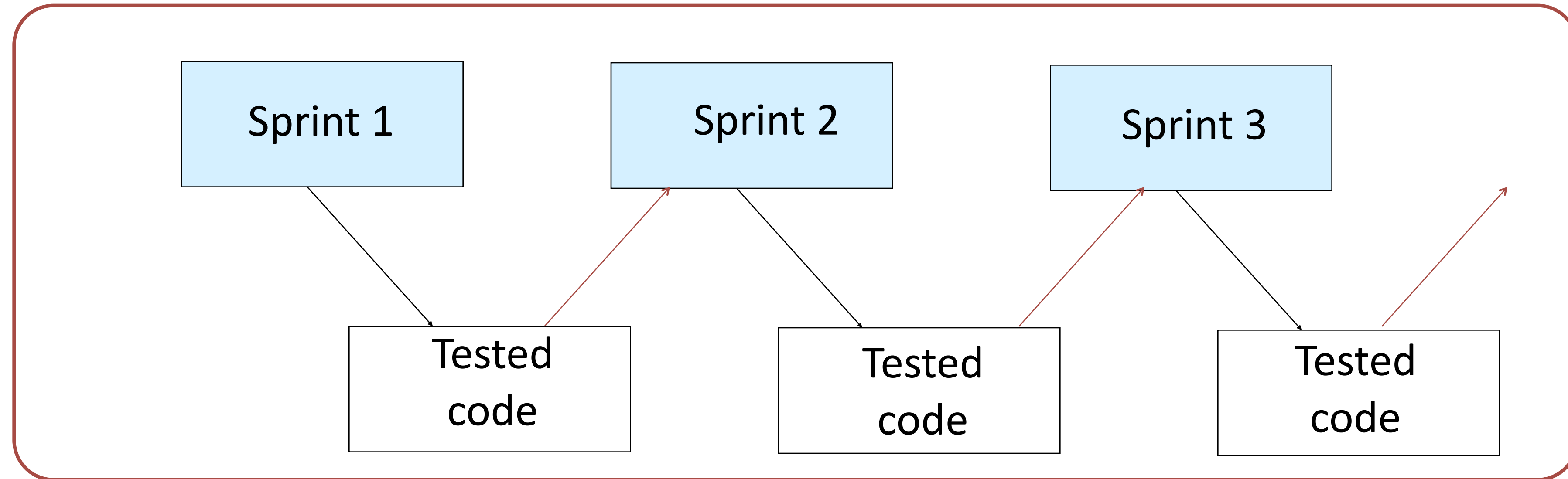
After each sprint the code may be:

- released (original agile method)

- combined with code from other sprints for subsequent release

- incorporated into a larger code base (spiral development)

Agile Development: Rework

The challenge of agile development

The agile approach is excellent for the development or continual enhancement of a system within an established architecture.

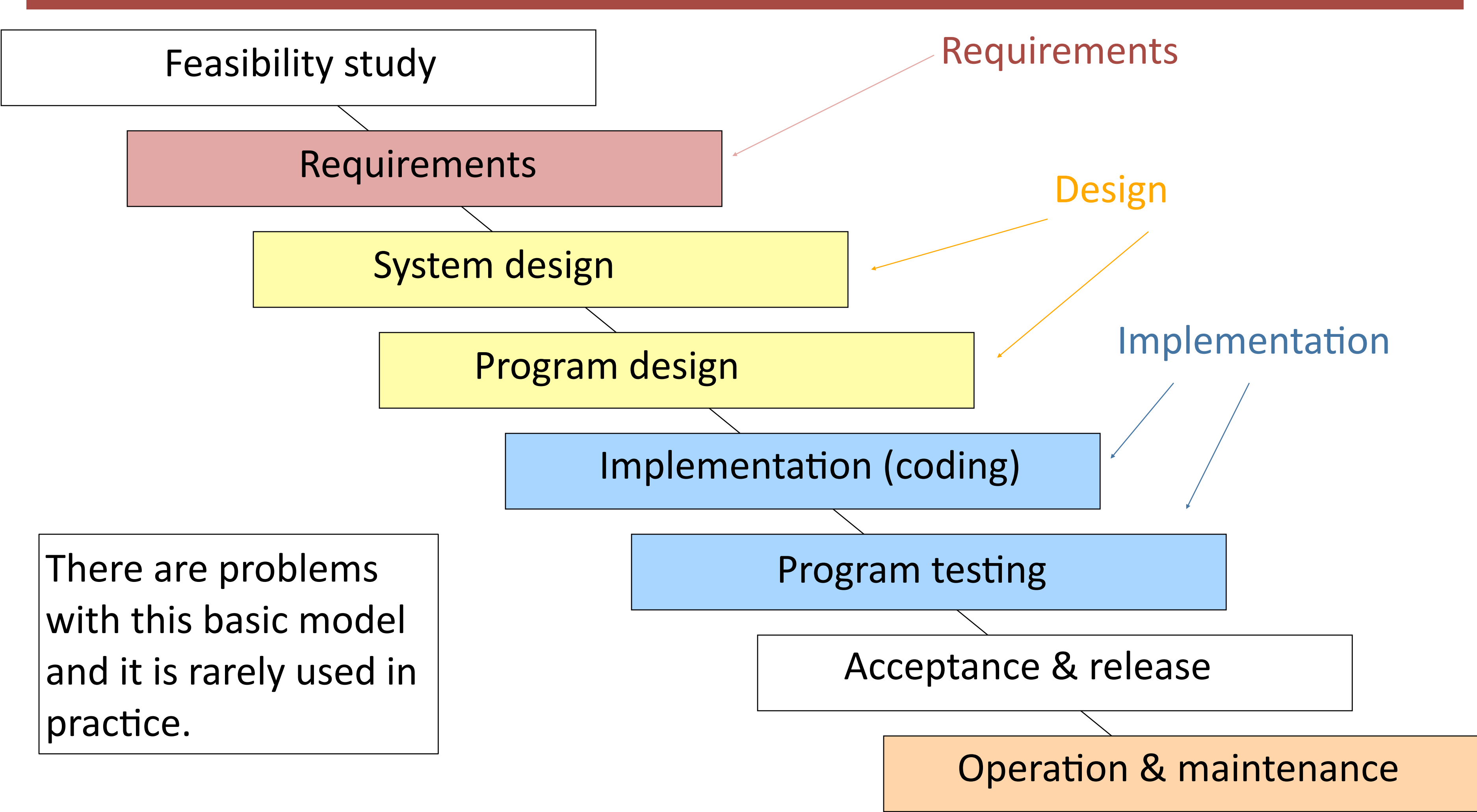A high-level team must establish the overall architecture and coordinate the sprints.

Rework

With agile development the requirements and design of the overall system emerge incrementally.

- Inevitably parts of some early sprints will need to be reworked.

- This requires changes to code that has already been fully tested and may have been released. This is always awkward.

If the volume of rework is large, it is more efficient not to fully polish each component, but to use iterative refinement to minimize the amount of rework.

Sequential Development: The Waterfall Model

Feasibility study

Requirements

Requirements

System design

Design

Program design

Implementation

Implementation (coding)

Program testing

There are problems with this basic model and it is rarely used in practice.

Acceptance & release

Operation & maintenance

Discussion of the Waterfall Model

The waterfall model is a heavyweight process with full documentation of each process step.

Advantages:

- Process visibility
- Separation of tasks
- Quality control at each step
- Cost monitoring at each step

Disadvantages:

In practice, each stage in the process reveals new understanding of the previous stages, which often requires the earlier stages to be revised.

The Waterfall Model is not flexible enough.
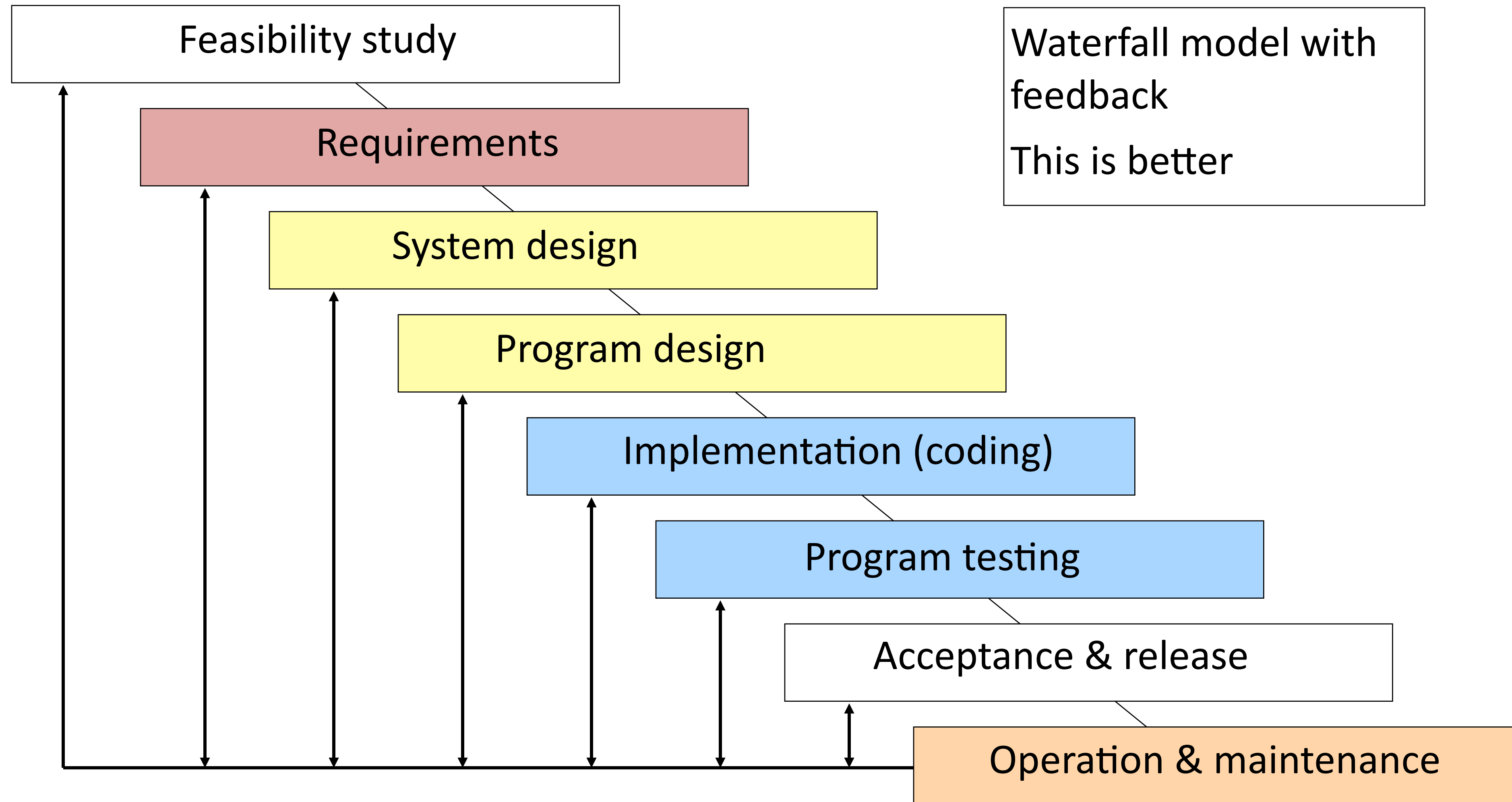
Discussion of the Waterfall Model

A pure sequential model is impossible

Examples:

- A feasibility study cannot create a proposed budget and schedule without a preliminary study of the requirements and a tentative design.

- Detailed design and implementation reveal gaps in the requirements specification.

- Requirements and/or technology may change during the development.

The plan must allow for some form of iteration.

Modified Waterfall Model

Feasibility study

Requirements

System design

Program design

Implementation (coding)

Program testing

Acceptance & release

Operation & maintenance

Waterfall model with feedback

This is better

Sequential Development

Sequential processes work best when the requirements are well understood and the design is straightforward, e.g.,

- Conversions of manual data processing systems where the requirements were well understood and few changes were made during the development (e.g., electricity billing).
- New models of a product where the functionality is closely derived from an earlier product (e.g. automatic braking system for a car).
- Portions of a large system where some components have clearly defined requirements and are clearly separated from the rest of the system.

Contracts

Note about contracts for software development

Some organizations contract for software development by placing separate contracts for each stage of the Waterfall Model or arrange for payment after each stage. This is a very bad practice.

Mixed Processes

In practice, many large projects use processes that mix aspects of the four types of software process. For example:

- With spiral development, new components may be developed using any of the three other methods.

- User interfaces have to be tested with users. This forces iterative development, even within an agile or sequential process.

Mixed Processes: Phased Development

Combine sequential and iterative elements

A simple system with basic functionality is brought quickly into production (Phase 1). This system may be developed using a sequential or iterative refinement.

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

Advantages

• Pay-back on investment begins soon.

• Requirement are more clearly understood when developing subsequent phases

Examples of Mixed Processes:
Iterative Refinement + Waterfall Model

Problem: Add graphics package to a programming environment

Phase 1: Iterative refinement

Make several prototype versions by extending the current environment with a preprocessor and run-time support package. Test with users until users are pleased with function. Throw the code away.

Phase 2: Modified waterfall

Use the results of Phase 1 to specify a formal set of requirements. Write new compiler and run-time system incorporating graphics elements. Make minor adjustments to requirements as needed.

Corporate Processes

Large software development organizations have their own internal processes that are designed for their needs. For example:

- Amazon.com (Internet commerce) makes extensive use of sprints. Most software development is divided into increments of about four weeks elapsed time.

- Lockheed Martin (government contractor) follows a sequential process that fits with the way that the US government manages software contracts.

- SAP (business software) emphasizes the functionality that is seen by their business customers. Much of the development is suitable for a sequential process.

- Microsoft (PC software) places great emphasis on testing with a very wide variety of equipment and backward compatibility. Much of the development uses a spiral process.

Choosing a Software Process

Changes during the software development process are expensive.

- If the requirements are poorly understood, or expected to change, select a process that keeps flexibility. Iterative refinement, agile sprints, phased implementation.

- If a big software system has many inter-related components, avoid major changes to the design of a system during development. Sequential process, such as the modified waterfall model.

- If the market for the software is poorly understood, use a process that gets operational software in front of customers as quickly as possible. Incremental, agile sprints.
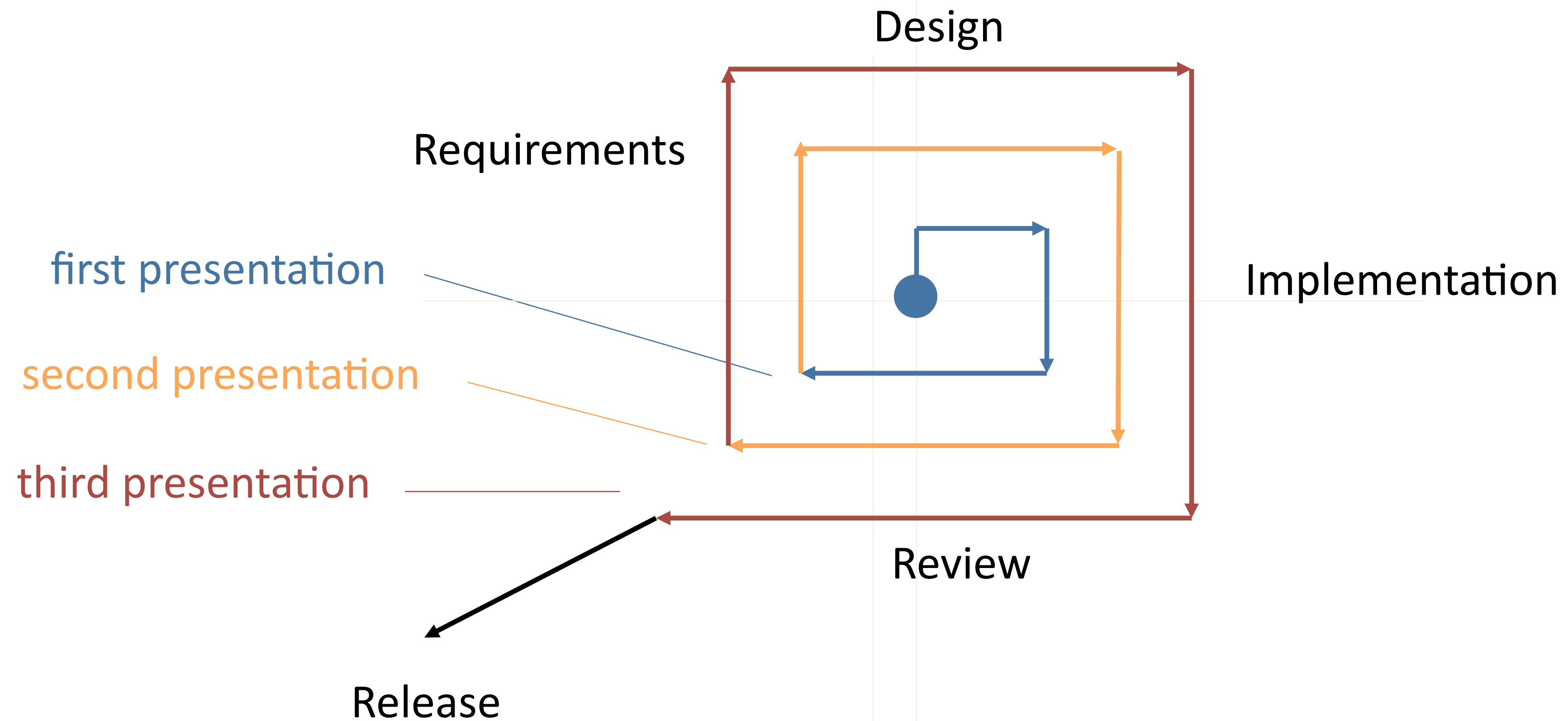
Observations about Software Processes

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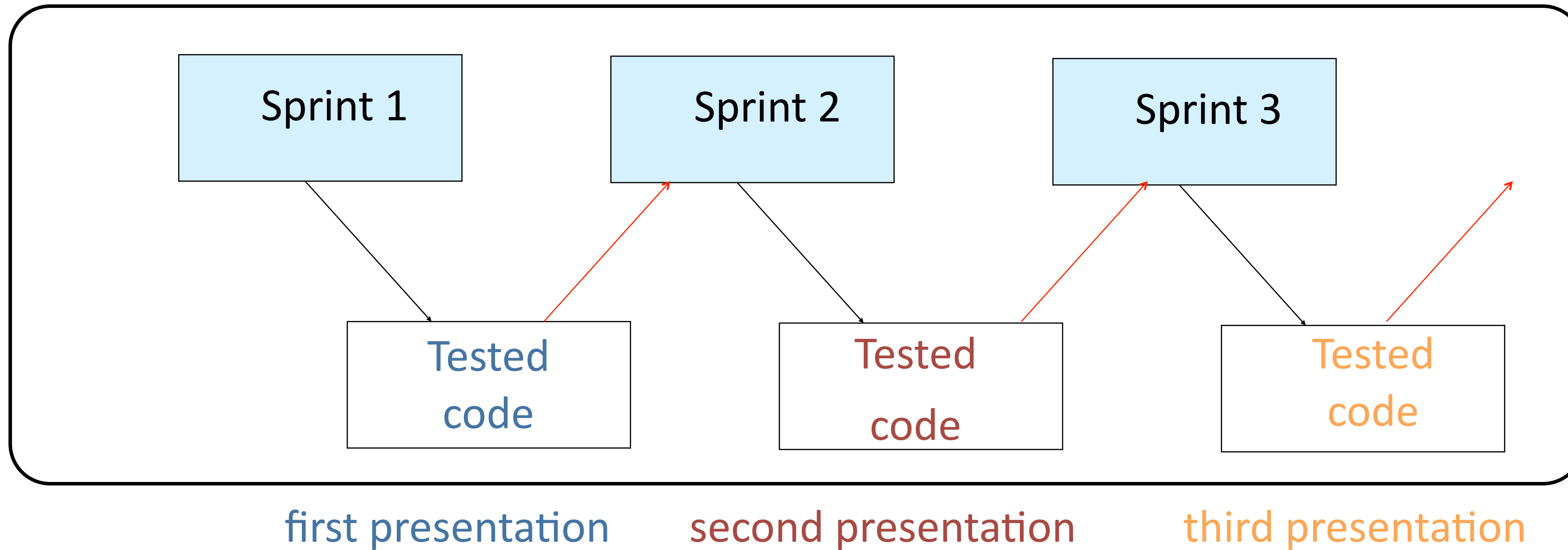*but ...* the development process is always partly evolutionary.

Risk is lowered by:

- **Prototyping** key components

- Frequent **releases**, or dividing large projects into **phases**

- Early and repeated testing with **users** and **customers**

- Following a **visible** software process

- Making use of reusable **components**

It is never possible to complete each step without provision for revision. This is known as throwing it over the wall.
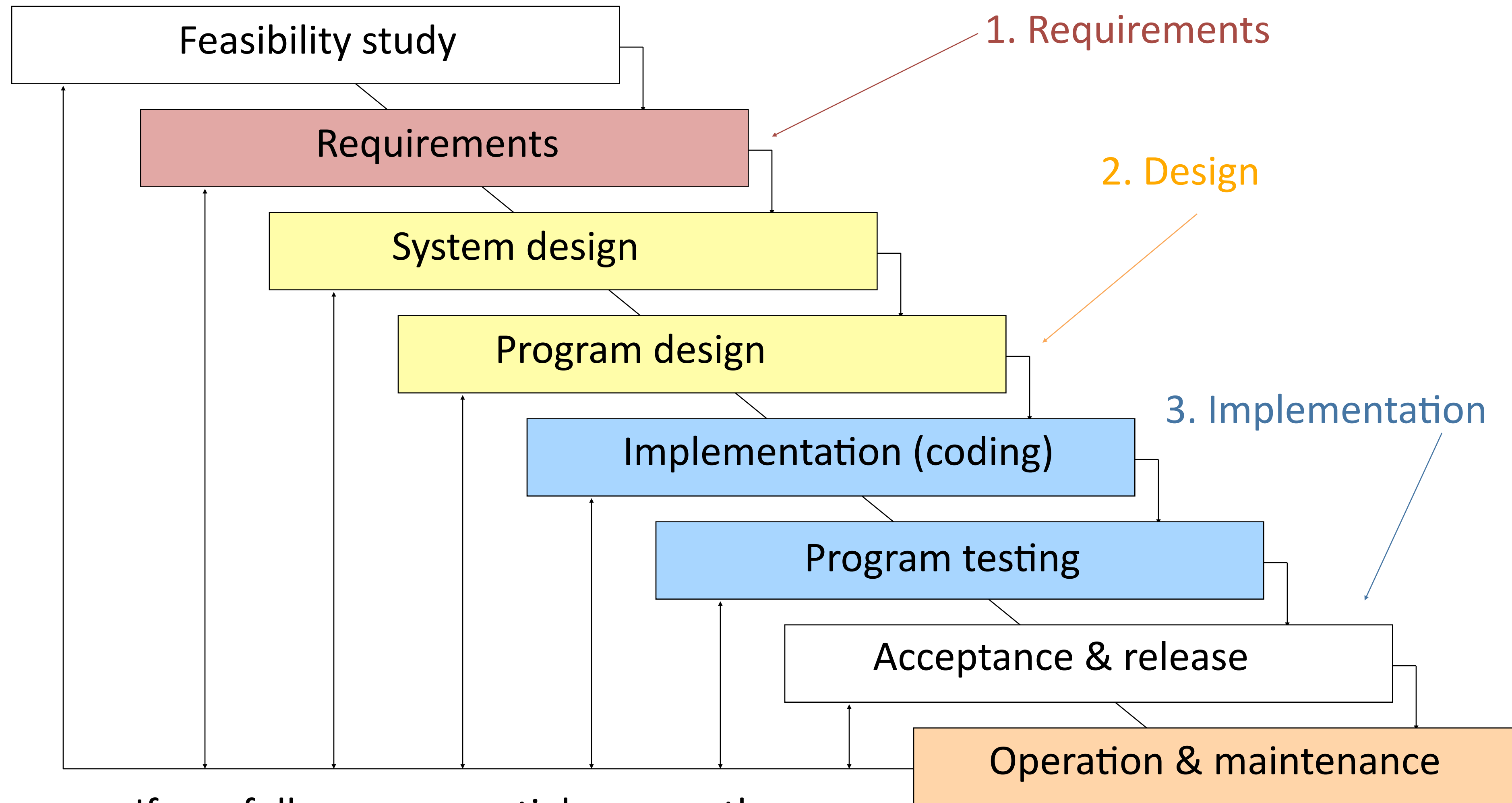
CS 5150 Projects: Iterative Refinement

Design

Requirements

Implementation

first presentation

second presentation

third presentation

Review

Release

CS 5150 Project: Agile Development

For each sprint aim to complete a section of the code.

CS 5150 Projects: Sequential Development

Feasibility study

Requirements

System design

Program design

Implementation (coding)

Program testing

Acceptance & release

Operation & maintenance

1. Requirements

2. Design

3. Implementation

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