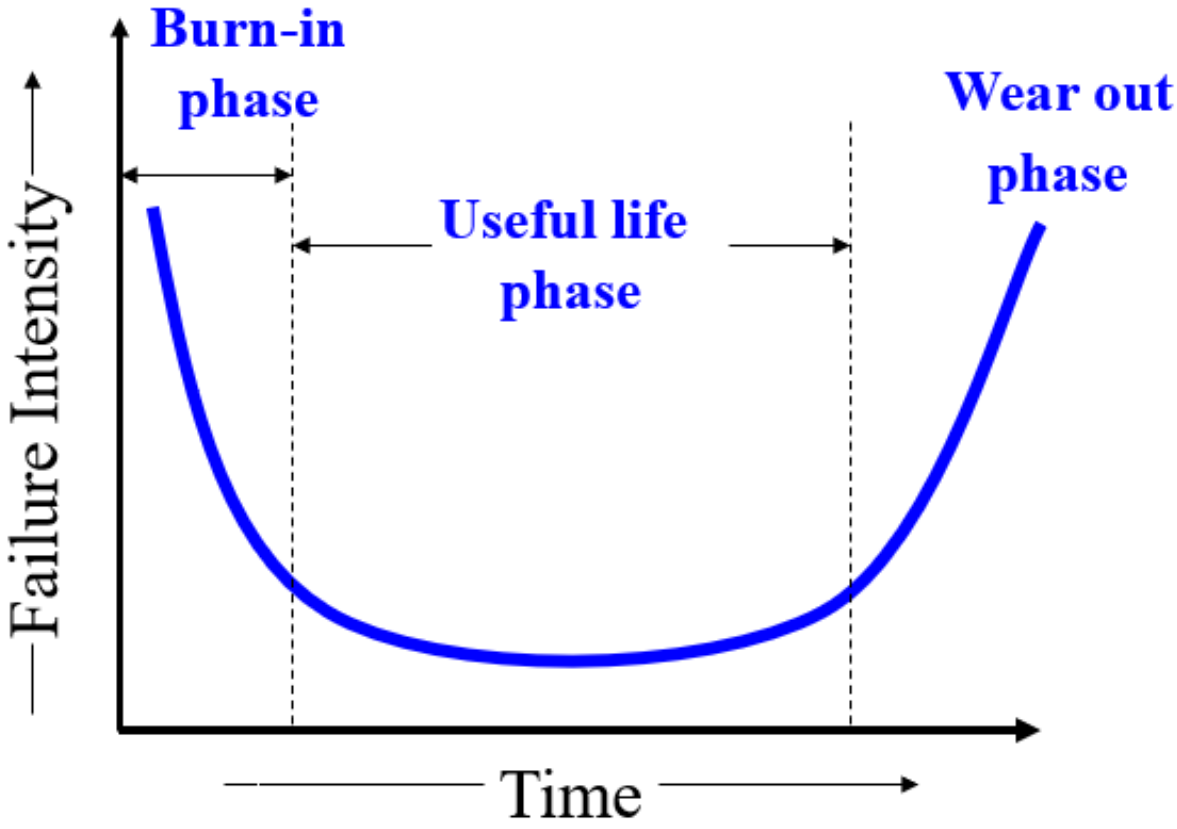
Software Characteristics:

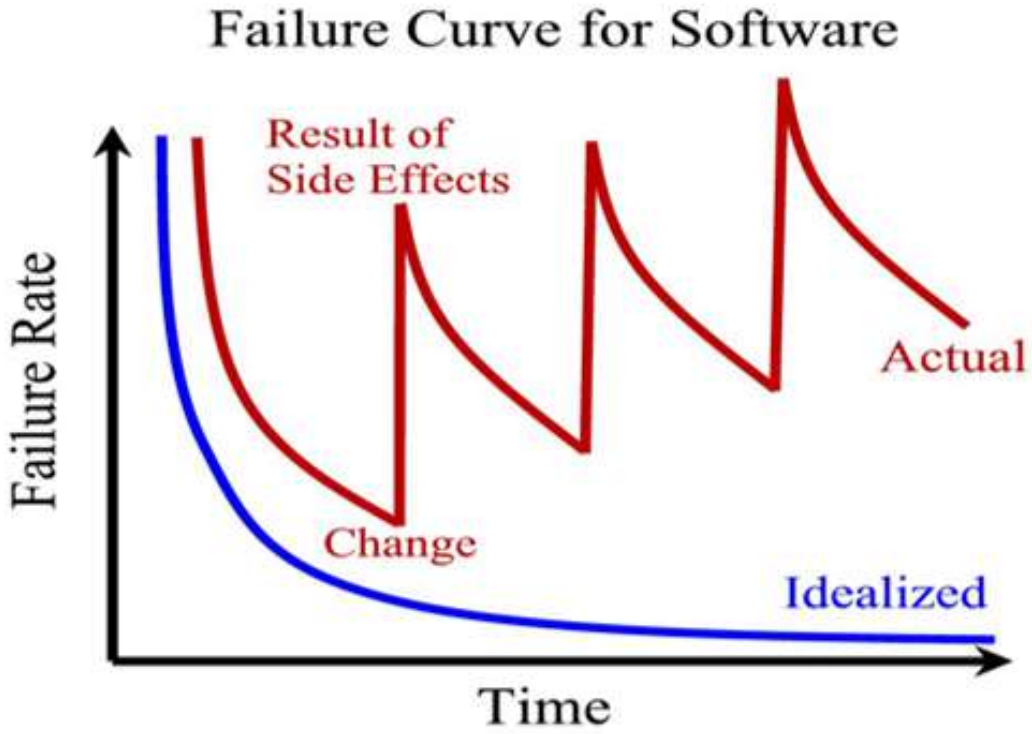
✓Software does not wear out.



Bath Tub Curve (for hardware products)

Software Characteristics:

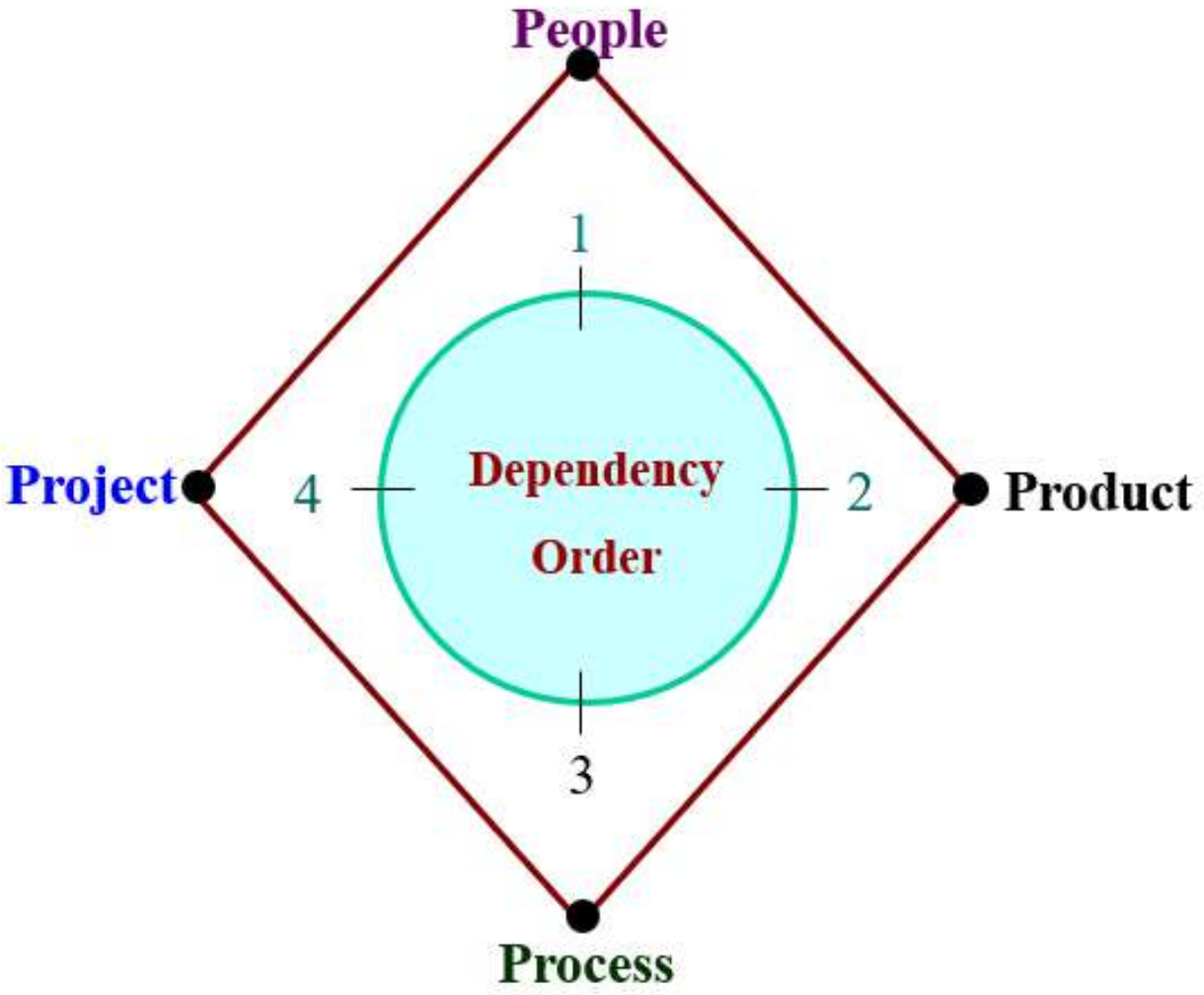
- ✓ Software is not manufactured
- ✓ Reusability of components
- ✓ Software is flexible



Factors Contributing to the Software Crisis

- Larger problems,
- Lack of adequate training in software engineering,
- Increasing skill shortage,
- Low productivity improvements.

Role of Management in Software Development



Sure! Here are concise points on software engineering myths from each perspective:

Management Perspective

1. **Easy Development:** Misjudging complexity leads to unrealistic timelines.
2. **Completion Myth:** Projects require ongoing maintenance, not just a launch.
3. **Latest Technology:** New tools don't guarantee success; fit matters more.

Customer Perspective

1. **Instant Gratification:** Quality software needs time for proper development.
2. **Feature Abundance:** More features don't equal better value; focus on essentials.
3. **Final Testing:** Testing should be continuous, not just at the end.

Developer Perspective

1. **Coding Focus:** Effective planning and design are critical, not just coding.
2. **Solo Effort:** Collaboration and communication are key to success.
3. **Static Knowledge:** Continuous learning is essential to keep up with technology.

Process metrics quantify the attributes of software development process and environment;

whereas product metrics are measures for the software product.

Examples

Process metrics: Productivity, Quality, Efficiency etc.

Product metrics: Size, Reliability, Complexity etc.

Software Life Cycle Models

The goal of Software Engineering is to provide models and processes that lead to the production of well-documented maintainable software in a manner that is predictable.

Waterfall Model

Problems of waterfall model

i. It is difficult to define all requirements at the beginning of a project

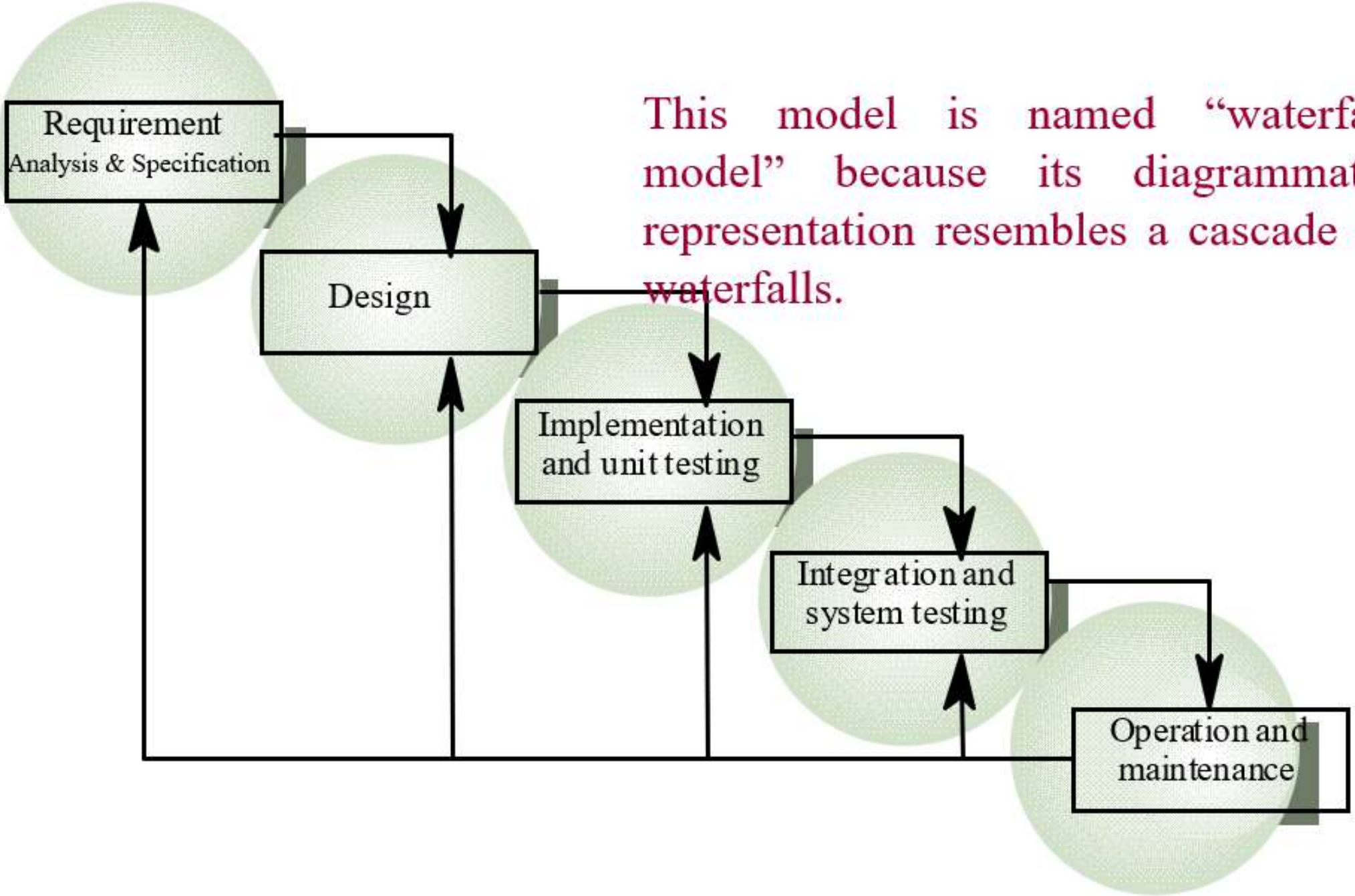
ii. This model is not suitable for accommodating any change

iii. A working version of the system is not seen until late in the project's life

iv. It does not scale up well to large projects.

v. Real projects are rarely sequential.

Waterfall Model



This model is named "waterfall model" because its diagrammatic representation resembles a cascade of waterfalls.

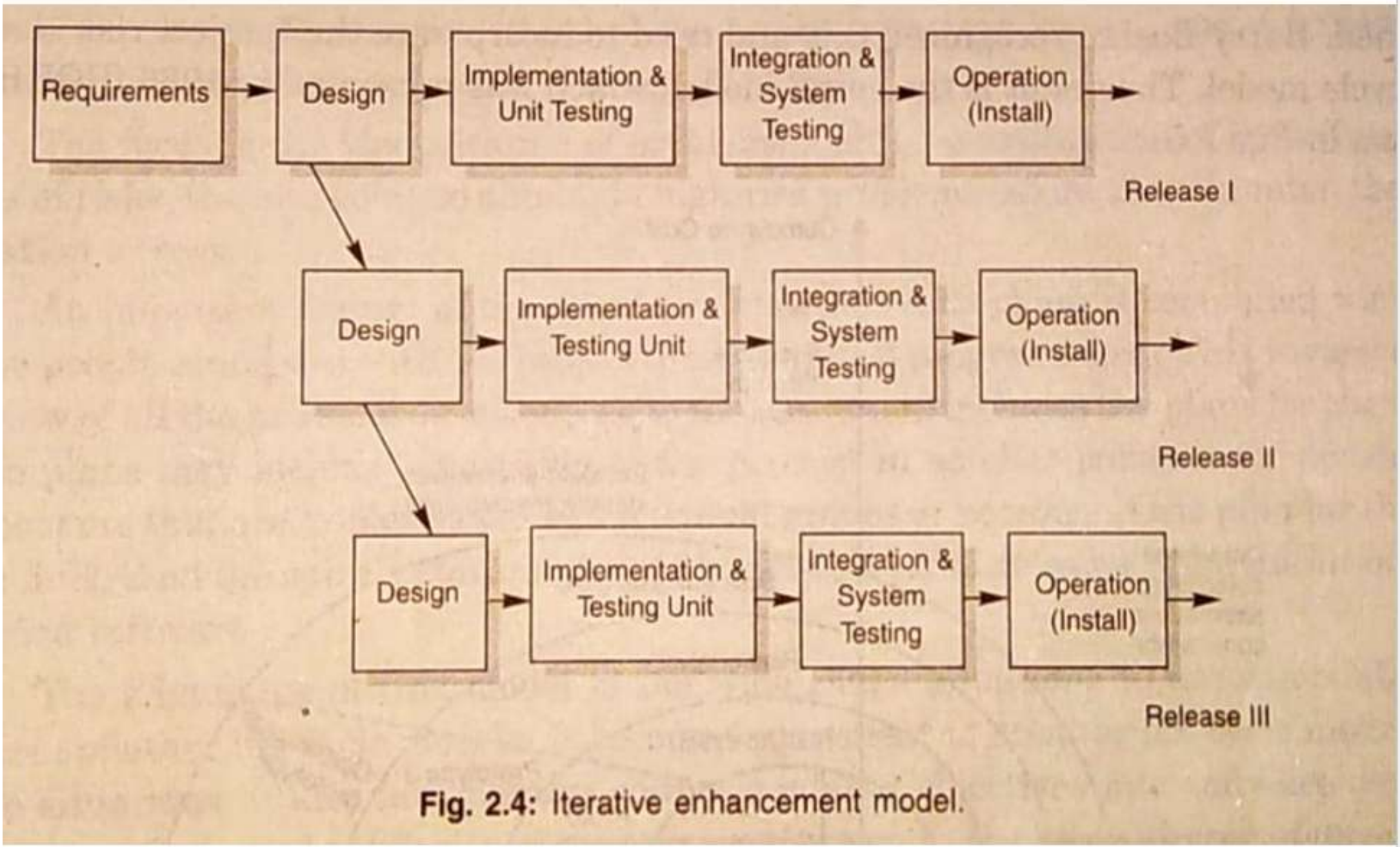
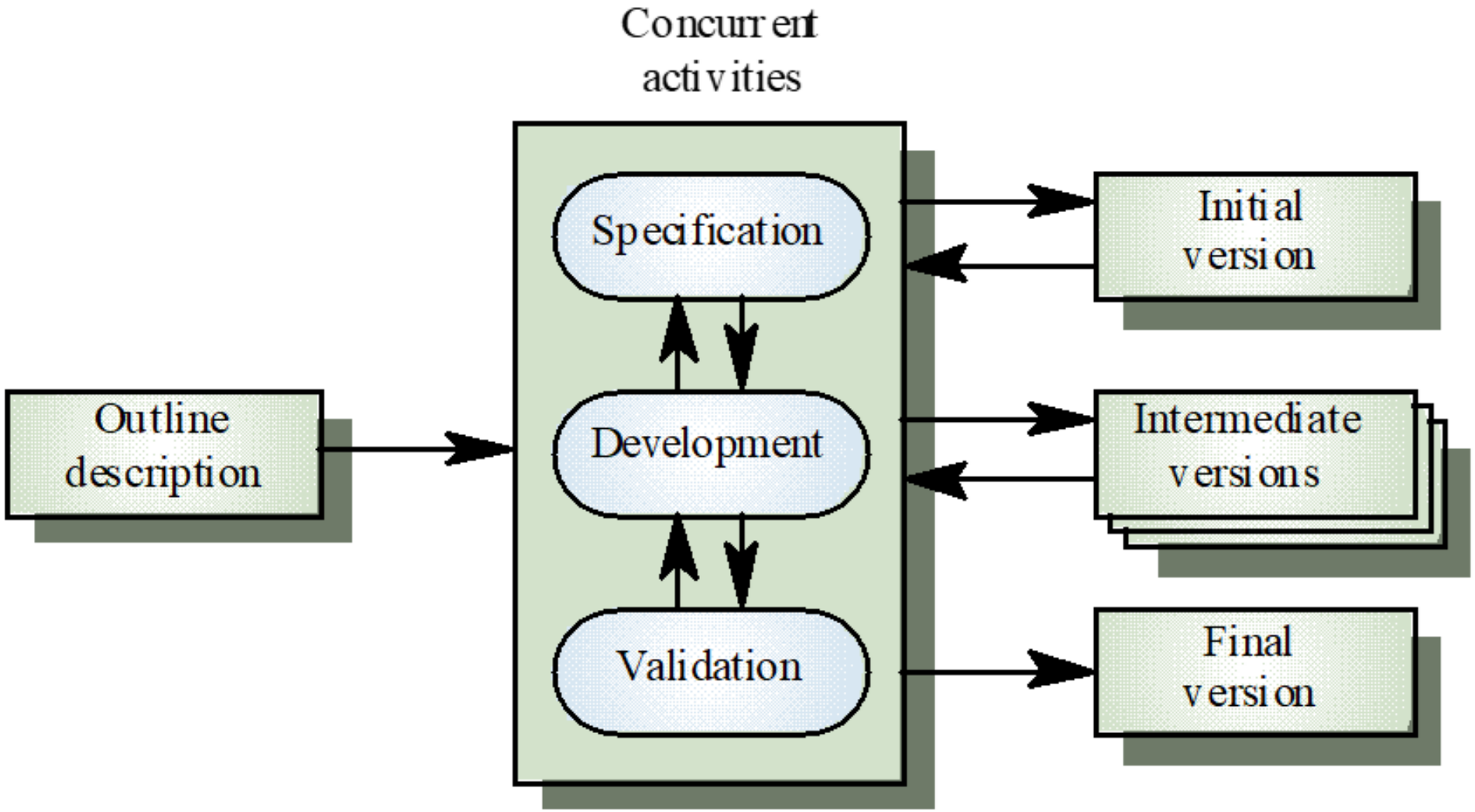


Fig. 2.4: Iterative enhancement model.

Evolutionary Process Model



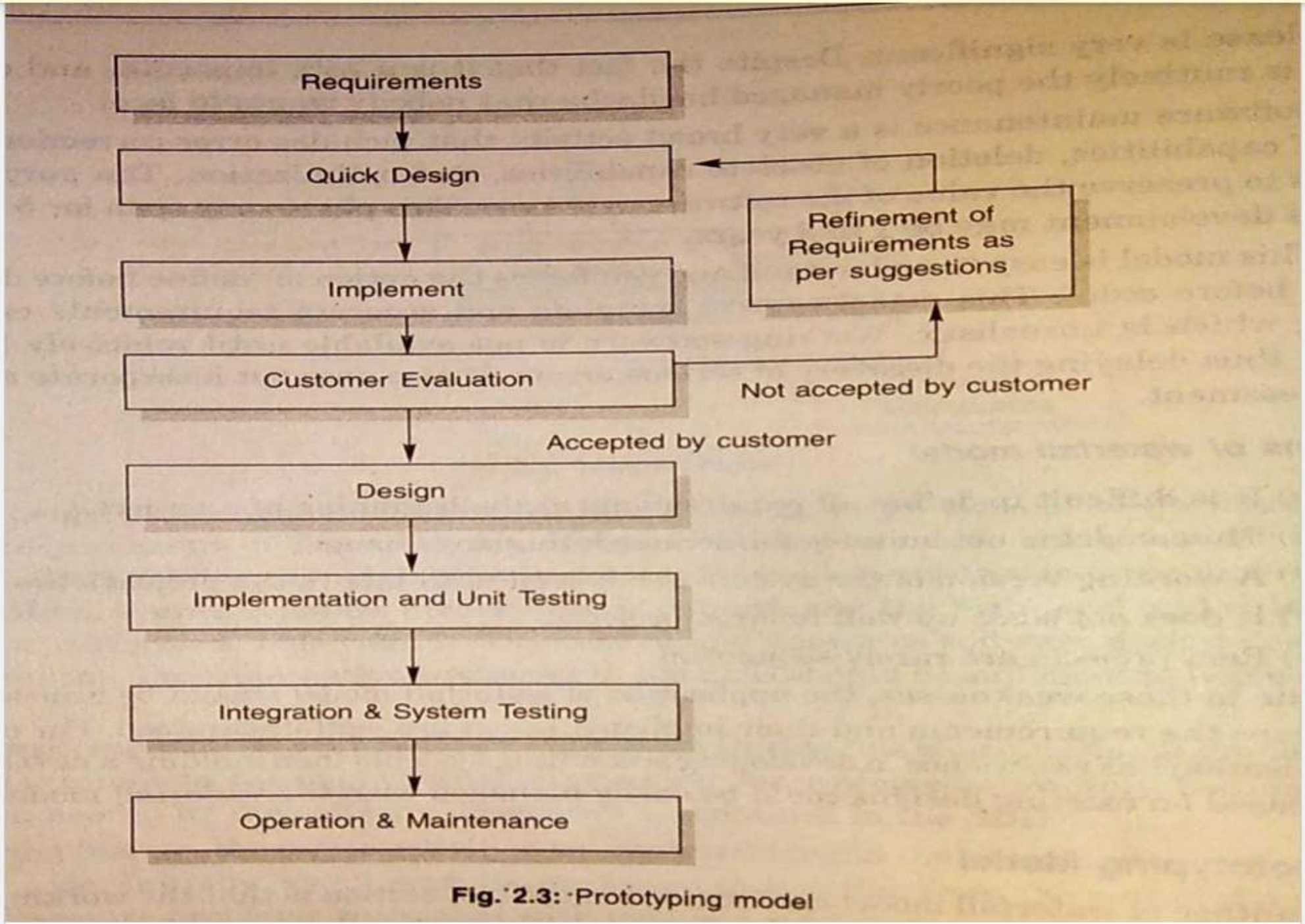


Fig. 2.3: Prototyping model

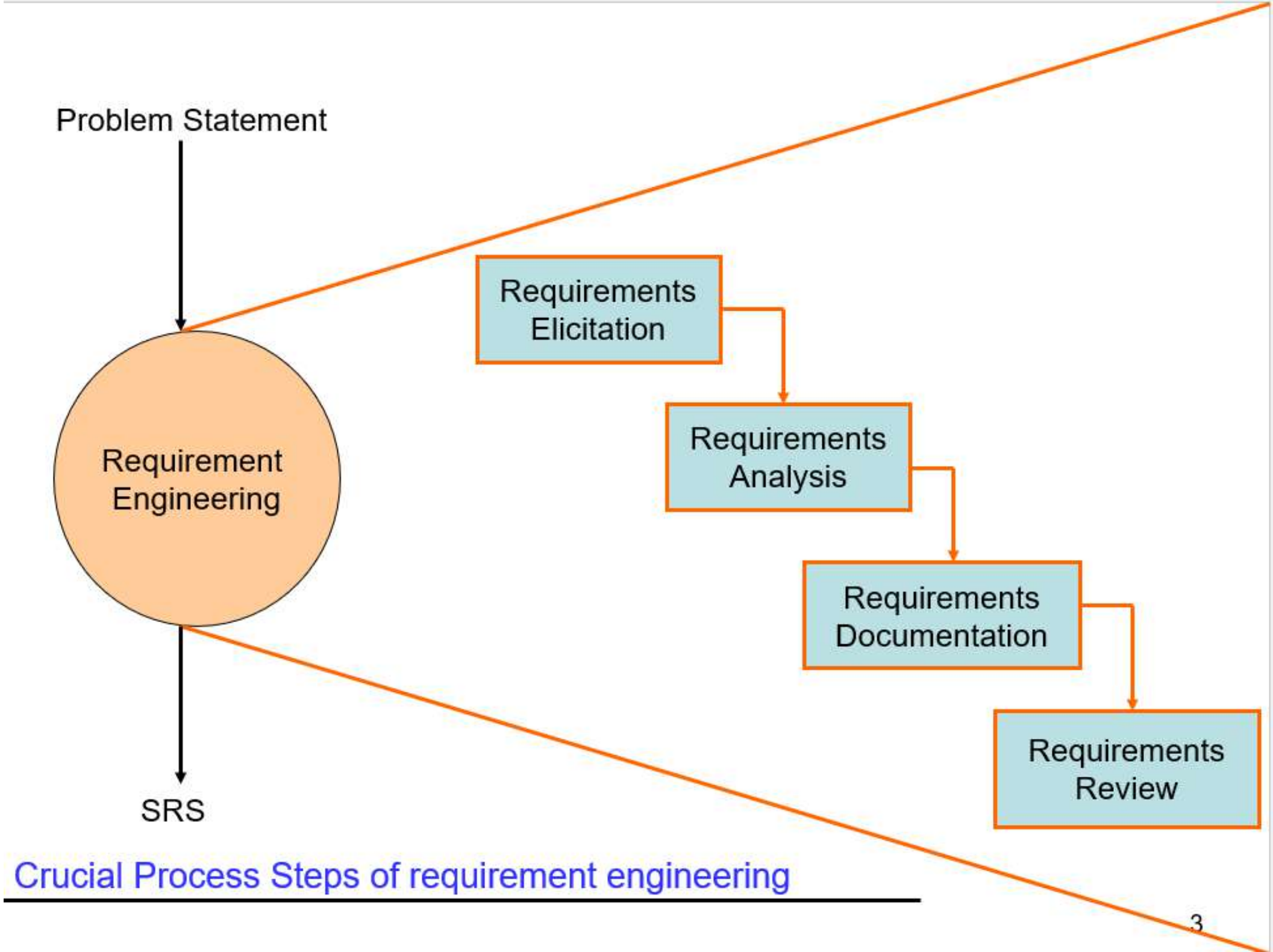
Spiral Model

The radial dimension of the model represents the cumulative costs. Each path around the spiral is indicative of increased costs. The angular dimension represents the progress made in completing each cycle. Each loop of the spiral from X-axis clockwise through 360º represents one phase. One phase is split roughly into four sectors of major activities.

- **Planning:** Determination of objectives, alternatives & constraints.
- **Risk Analysis:** Analyze alternatives and attempts to identify and resolve the risks involved.
- **Development:** Product development and testing product.
- **Assessment:** Customer evaluation

The Unified Process Model consists of four main parts:

1. **Inception**: Defines the project's scope, feasibility, and initial requirements, focusing on business needs and stakeholder input.
2. **Elaboration**: Refines the requirements and architecture, addressing risks and creating a more detailed design.
3. **Construction**: Involves actual development and testing of the software, focusing on building the system incrementally.
4. **Transition**: Ensures the software is delivered and deployed, gathering user feedback and making necessary adjustments before final release.



Crucial Process Steps of requirement engineering

Requirements Documentation

SRS Should

-- Correctly define all requirements

-- not describe any design details

-- not impose any additional constraints

Characteristics of a good SRS

An SRS Should be

✓ Correct

✓ Unambiguous

✓ Complete

✓ Consistent

Requirements Validation

Problem actions

- Requirements clarification
- Missing information
  - find this information from stakeholders
- Requirements conflicts
  - Stakeholders must negotiate to resolve this conflict
- Unrealistic requirements
  - Stakeholders must be consulted
- Security issues
  - Review the system in accordance to security standards

Software Design

The design needs to be

- Correct & complete
- Understandable
- At the right level
- Maintainable

Consider the example of editing a student record in a 'student information system'.

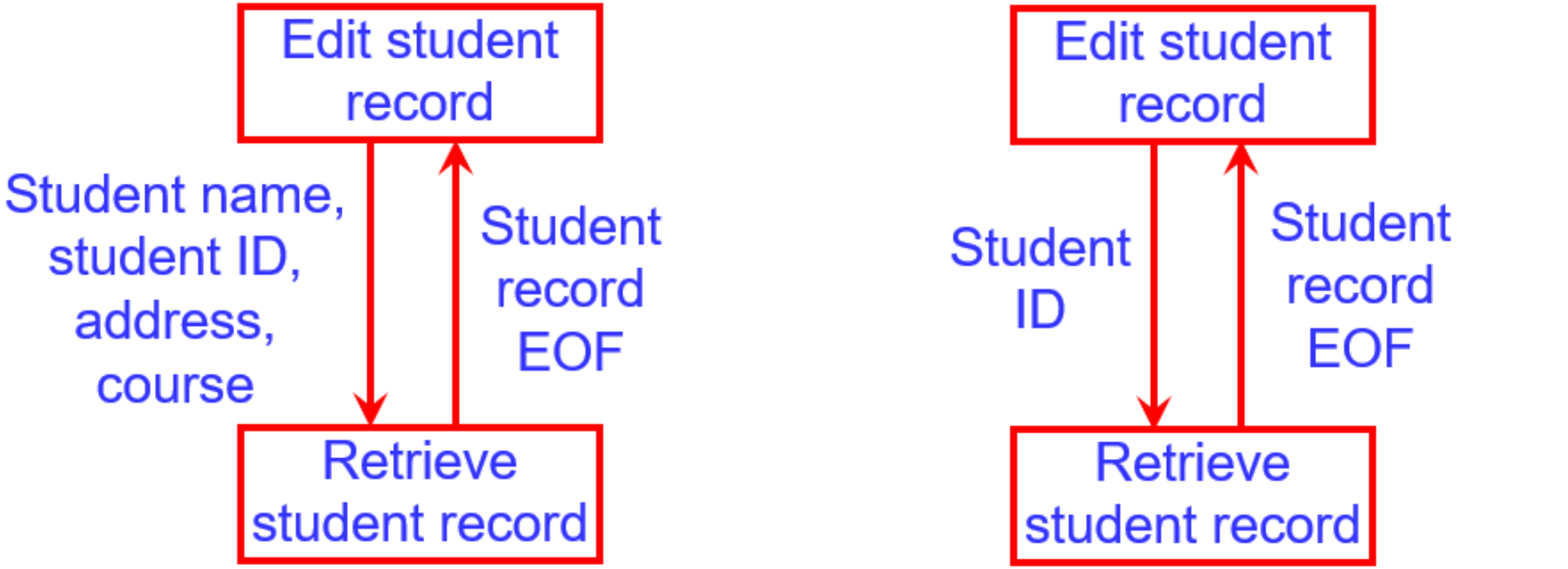


Fig. 6 : Example of coupling

Software Design

Data coupling	Best
Stamp coupling	
Control coupling	
External coupling	
Common coupling	
Content coupling	Worst

Fig. 7 : The types of module coupling

Given two procedures A & B, we can identify number of ways in which they can be coupled.

Module Cohesion

Cohesion is a measure of the degree to which the elements of a module are functionally related.

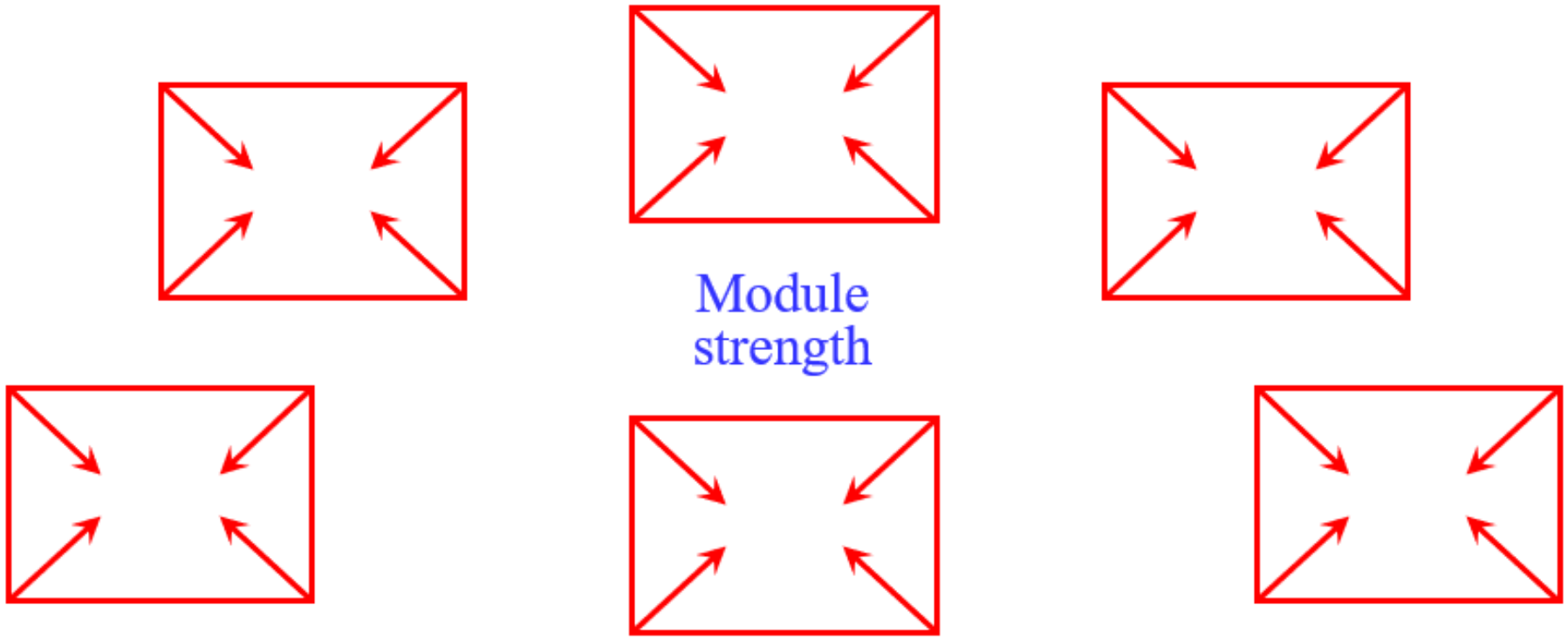


Fig. 10 : Cohesion=Strength of relations within modules

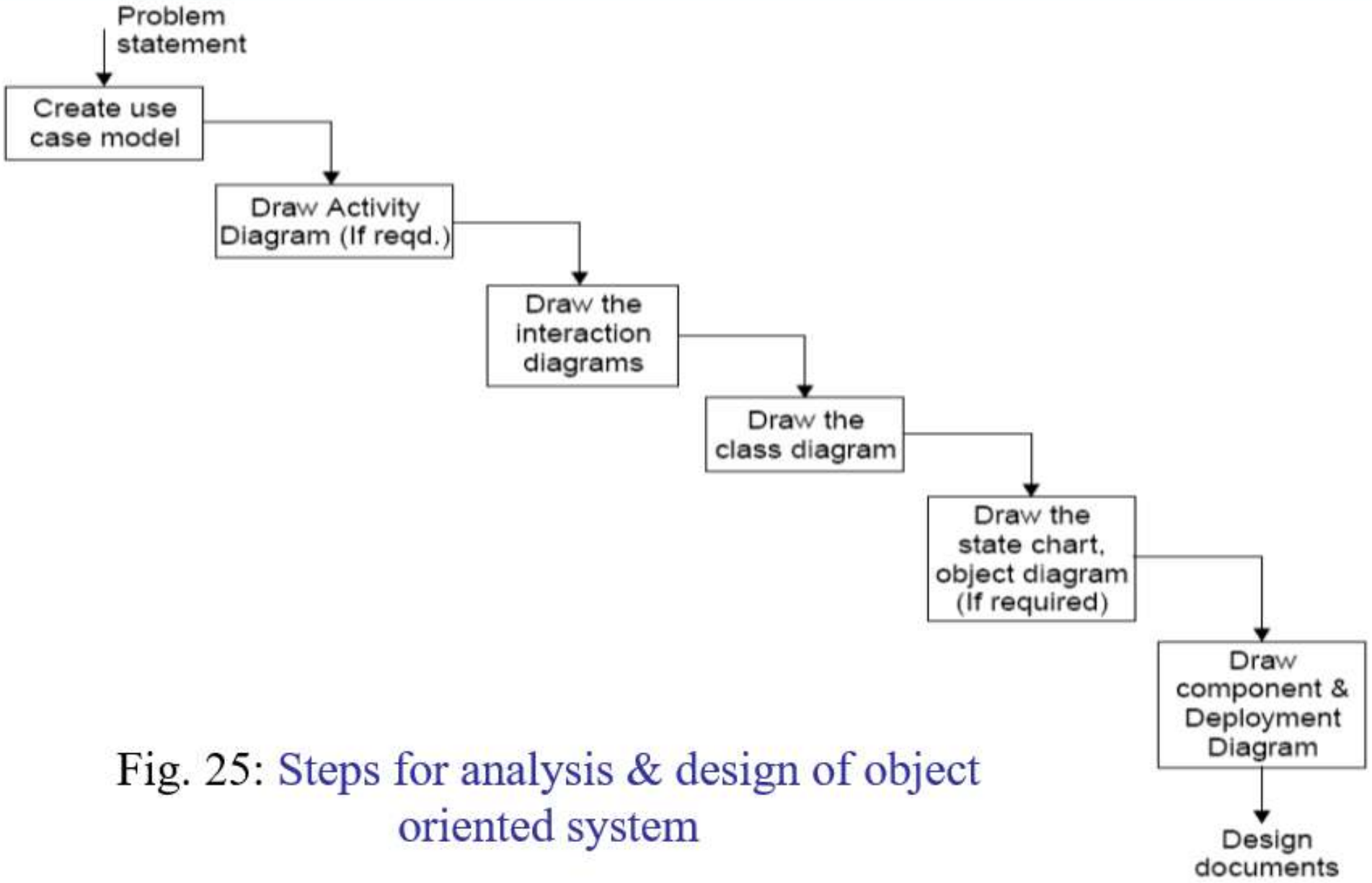


Fig. 25: Steps for analysis & design of object oriented system