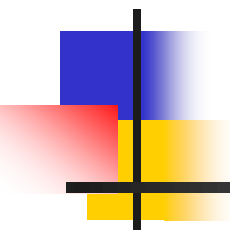


# SWE 205 - Introduction to Software Engineering



---

## Lecture 4



# Lecture Objectives

---

- Software processes and their principle characteristics.
- Generic software process models
  - Waterfall model
  - Evolutionary development
  - Component based system development



# What is a software process?

---

- A set of ordered tasks to produce indented output of some kind
  - Involving activates, constraints; and
  - Resources
- The process of building a software product
  - Life cycle - describes the life of the software from conception through its implementation, delivery, use and maintenance.



# Why we need software process?

---

- Common understanding of the activities, resources and constraints involved in software development.
- Creating processes helps
  - Find inconsistencies,
  - Redundancies; and
  - Omissions

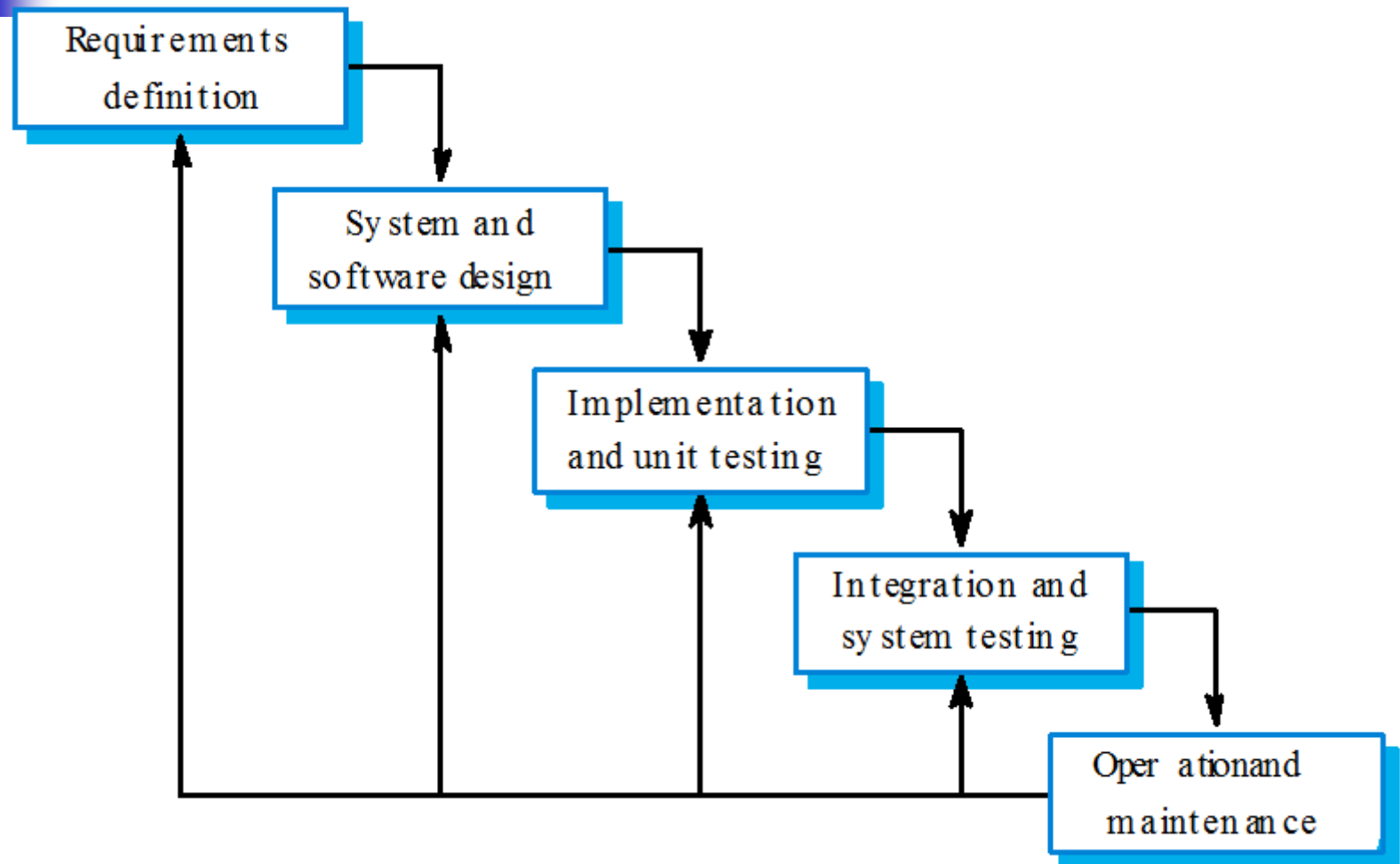


# Generic software process models

---

- The waterfall model
  - Separate and distinct phases of specification and development.
- Evolutionary development
  - Specification, development and validation are interleaved.
- Component-based software engineering
  - The system is assembled from existing components.

# Waterfall Model





# Waterfall Model

---

- Inflexible partitioning of the project into distinct stages
  - Makes it difficult to respond to changing customer requirements.
- Model is only appropriate when the requirements are well-understood; and
  - changes will be fairly limited during the design process.

Few business systems have stable requirements.



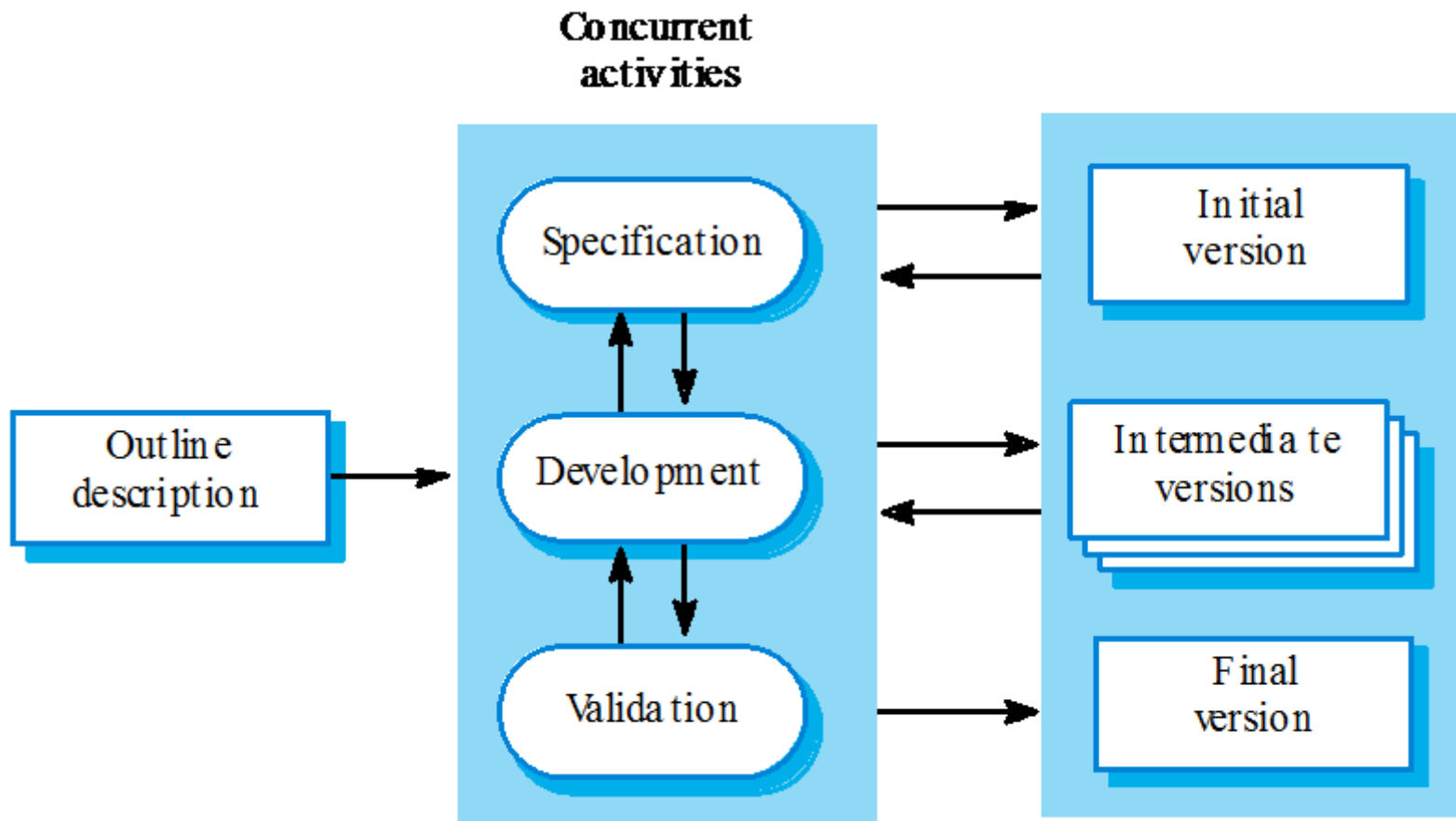
# Evolutionary development

---

- Exploratory development
  - Objective is to work with customers and to evolve a final system from an initial outline specification.
  - Starts with well-understood requirements and add new features as proposed by the customer.
- Throw-away prototyping
  - Objective is to understand the system requirements.
  - Starts with poorly understood requirements to clarify what is really needed.



# Evolutionary Development





# Evolutionary Development

---

- Problems

- Lack of process visibility;
- Systems are often poorly structured;
- Special skills (e.g. in languages for rapid prototyping) may be required.

- Applicability

- For small or medium-size interactive systems;
- For parts of large systems (e.g. the user interface);
- For short-lifetime systems.



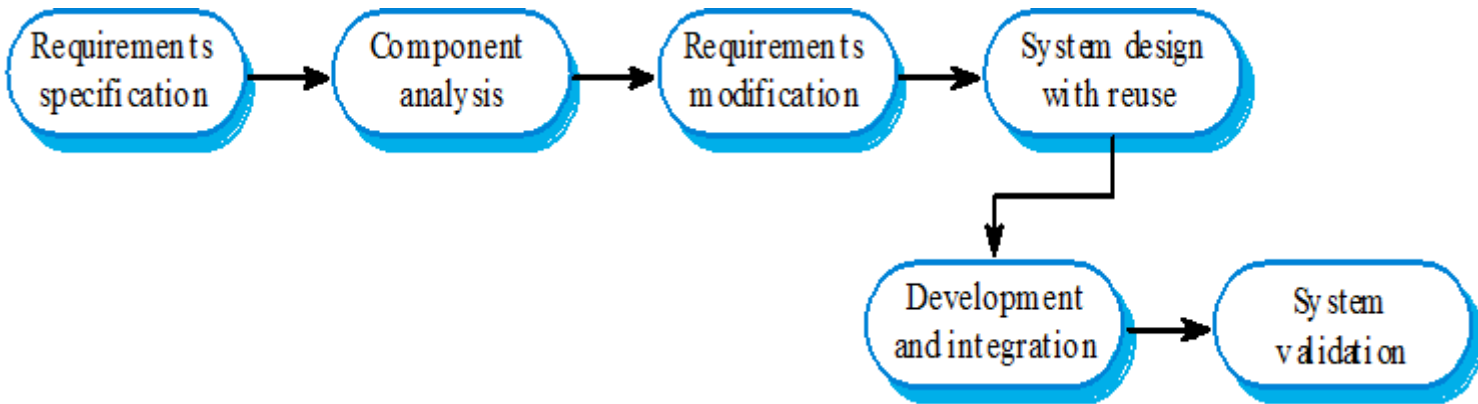
# Component Based System Engineering

---

- Systematic reuse where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.

This approach is becoming increasingly used as component standards have emerged.

# Component Based System Engineering





# Process Iterations

---

- System requirements ALWAYS evolve in the course of a project
  - Process iteration where earlier stages are reworked is always part of the process for large systems.
- Iteration can be applied to any of the generic process models.
- Two (related) approaches
  - Incremental delivery;
  - Spiral development.

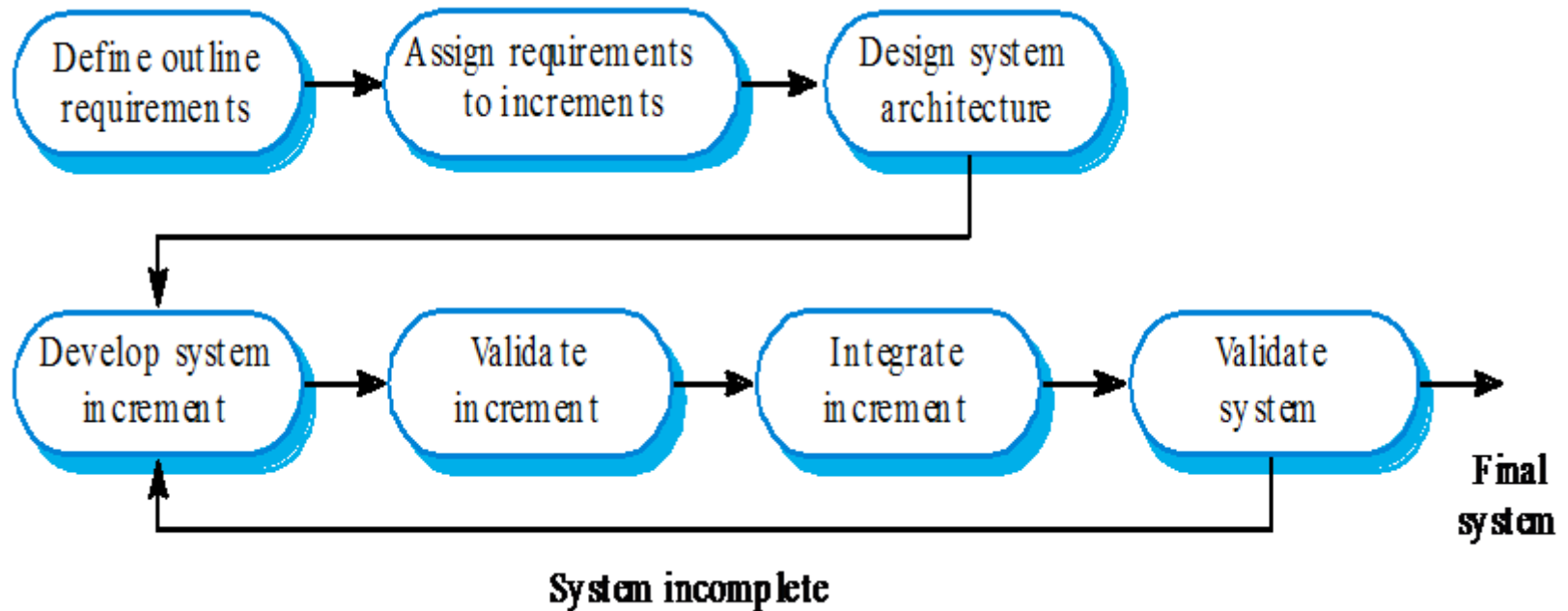


# Incremental Delivery

---

- Rather than deliver the system as a single delivery,
  - Development and delivery is broken down into increments;
  - With each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments.

# Incremental Delivery



Requirements are frozen though requirements for later increments can continue to evolve.



# Incremental Delivery - Advantages

---

- Customer value can be delivered with each increment so system functionality is available earlier.
- Early increments act as a prototype to help elicit requirements for later increments.
- Lower risk of overall project failure.
- The highest priority system services tend to receive the most testing.



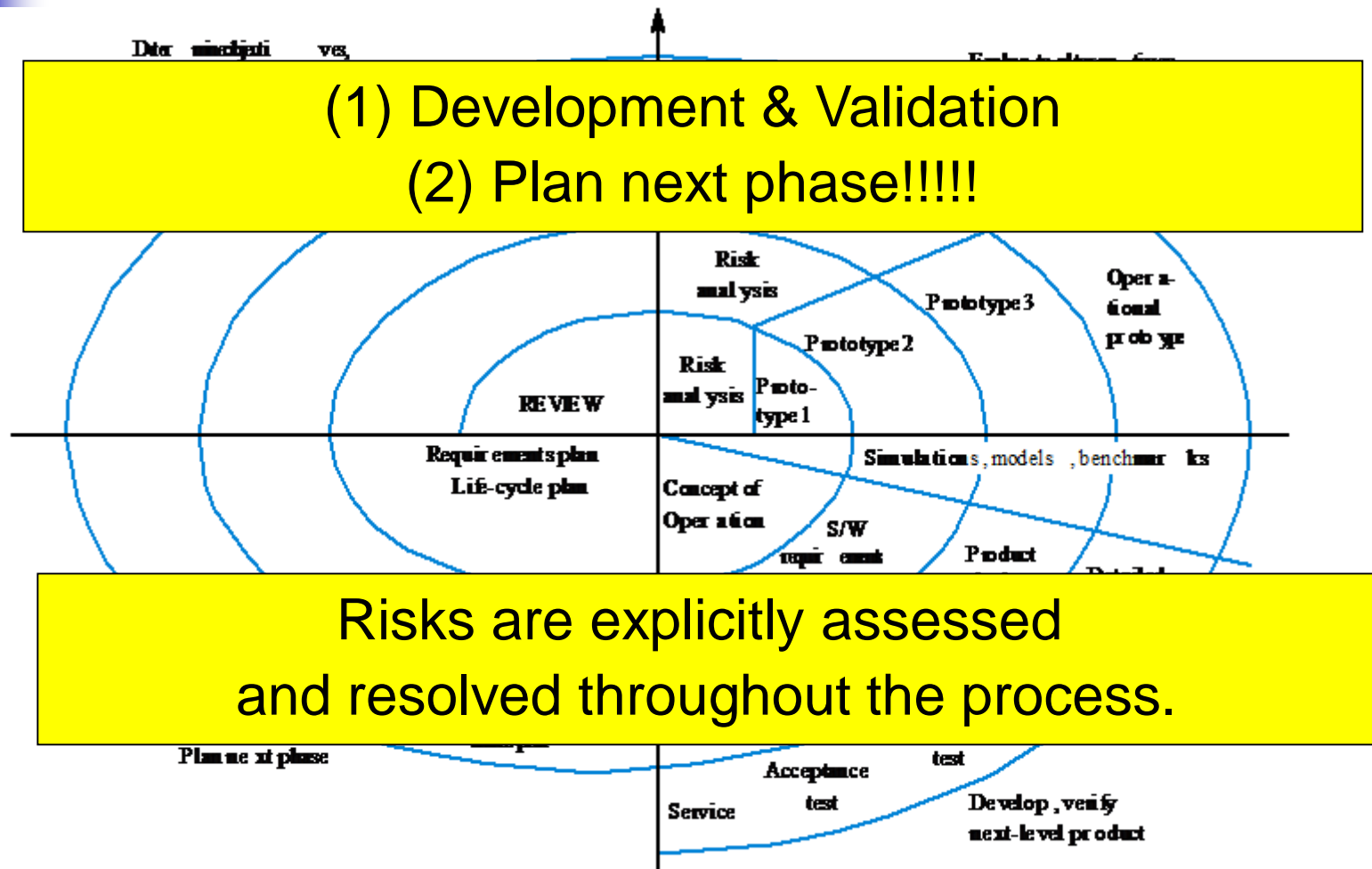


# Spiral Development

---

- Process is represented as a spiral rather than as a sequence of activities with backtracking.
- Each loop in the spiral represents a phase in the process.
- No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.

# Spiral Development Model





# Key Points

---

- Software processes are the activities involved in producing and evolving a software system.
- General activities are specification, design and implementation, validation and evolution.
- Generic process models describe the organisation of software processes. For example
  - waterfall model,
  - evolutionary development; and
  - component-based software engineering.
- Iterative process models describe the software process as a cycle of activities.