

2008 Fall

# Software Modeling & Analysis

## Part 1. Overview

- Software Processes
- Project Management

Lecturer: JUNBEOM YOO  
jbyoo@konkuk.ac.kr

Chapter 4.  
Software Processes

# Objectives

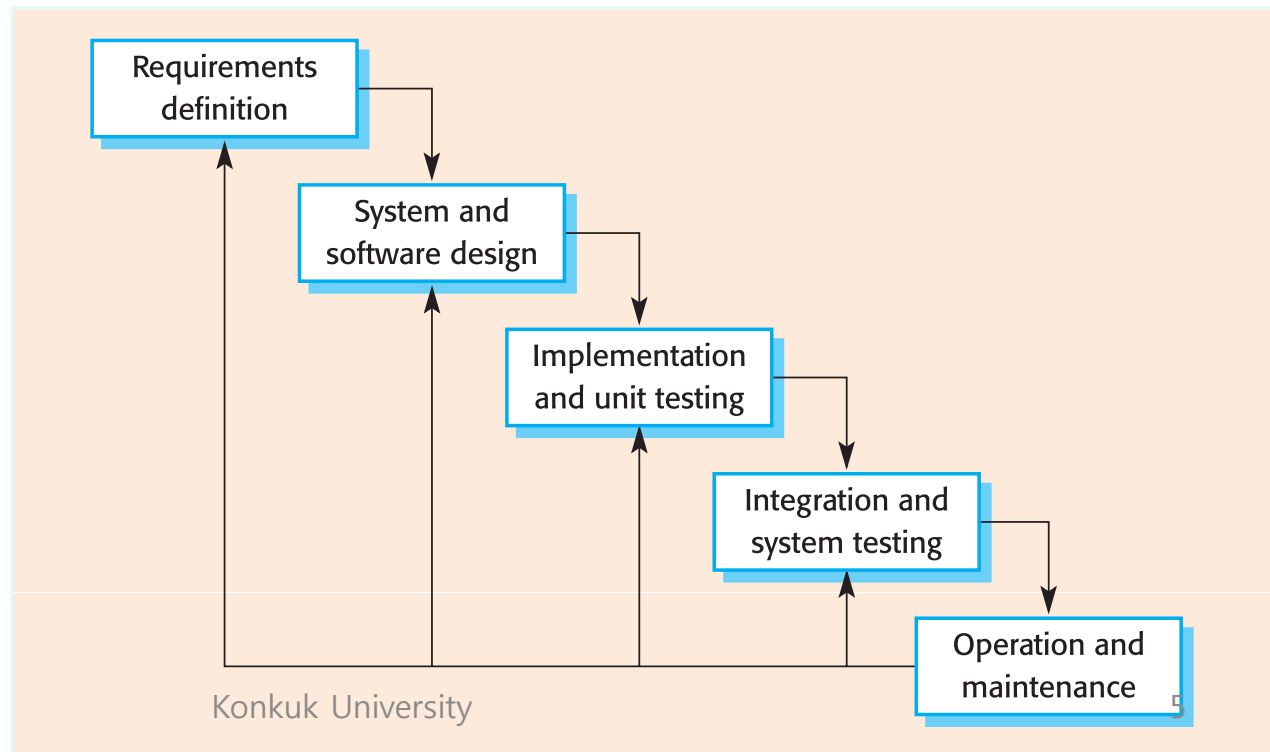
- To introduce software process models
- To describe three generic process models and when they may be used
- To describe process activities
- To explain the Rational Unified Process(RUP) model

# Software Process

- A structured set of activities required to develop a software system
  - Specification
  - Design
  - Validation
  - Evolution
- A software process model is an abstract representation of a process.
  - Waterfall model
    - Separate and distinct phases of specification and development
  - Evolutionary development
    - Specification, development and validation are interleaved.
  - Component-based software engineering
    - System is assembled from existing component.
  - Many variants

# Waterfall Model

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



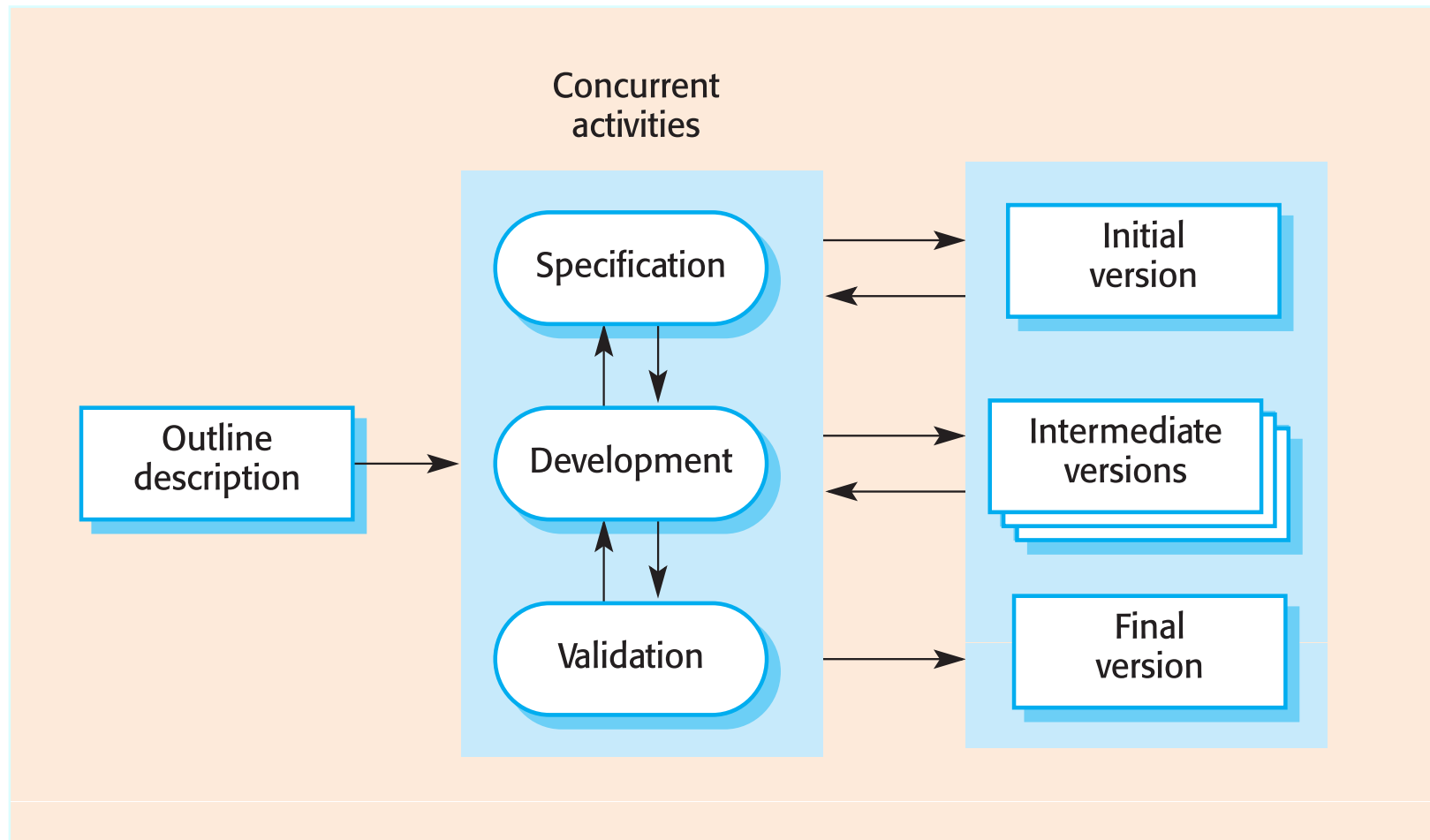
# Waterfall Model Problems

- Inflexible partitioning of project into distinct stages makes it difficult to respond to changing customer requirements.
- Therefore, this model is only appropriate when
  - Requirements are well-understood.
  - Changes will be fairly limited during design process.
  - However, few business systems have stable requirements.
- Waterfall model is mostly used for large system engineering projects where a system is developed at several sites.

# Evolutionary Development

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

# Evolutionary Development Phases



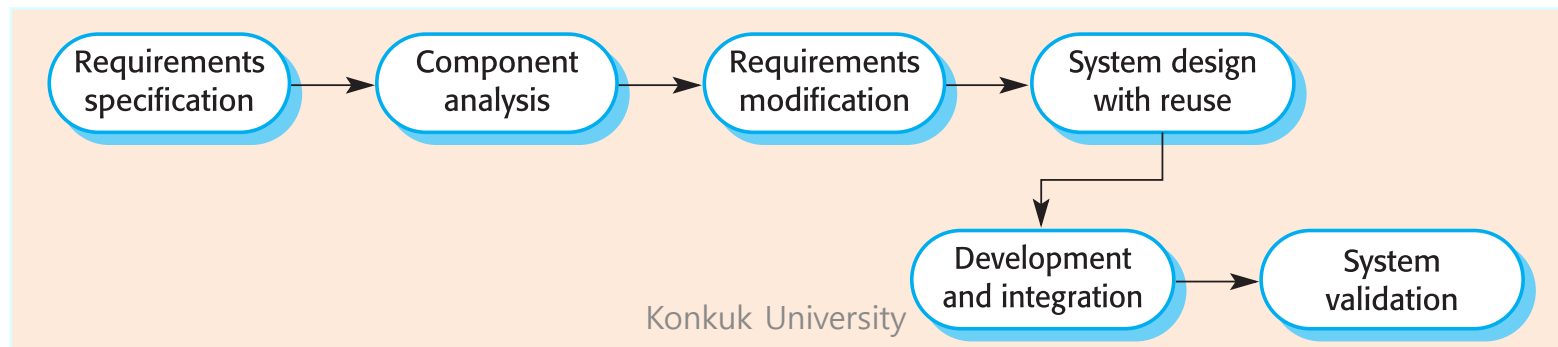


# Evolutionary Development Problems

- 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

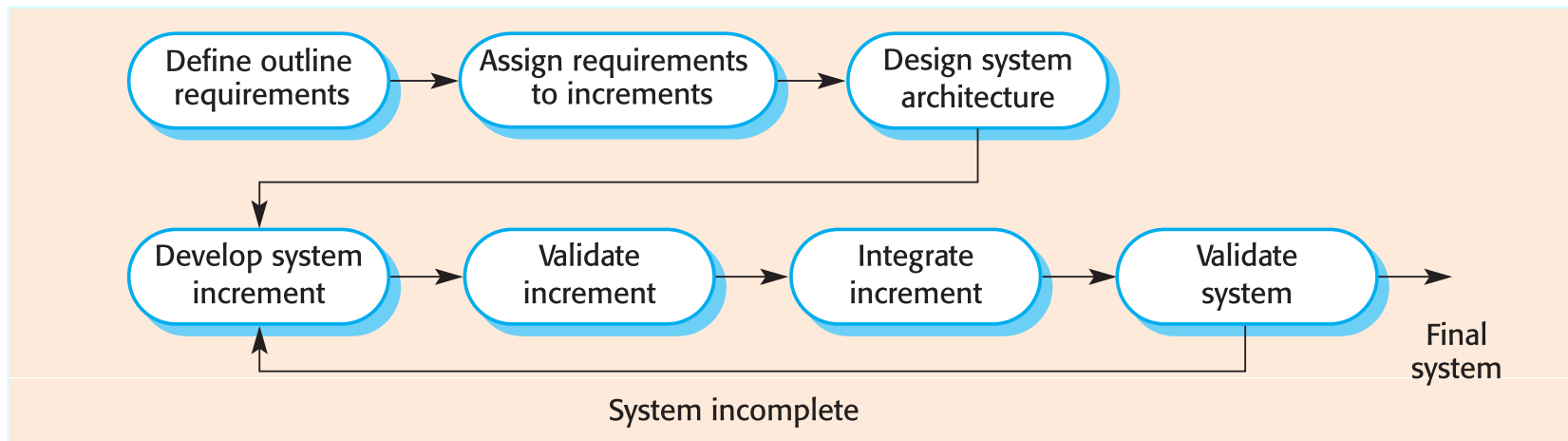
# Component-Based Software Engineering

- Based on systematic reuse.
- Systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems.
- Process stages
  - Component analysis
  - Requirements modification
  - System design with reuse
  - Development and integration
- This approach is becoming increasingly used as component standards have emerged.



# Process Iteration

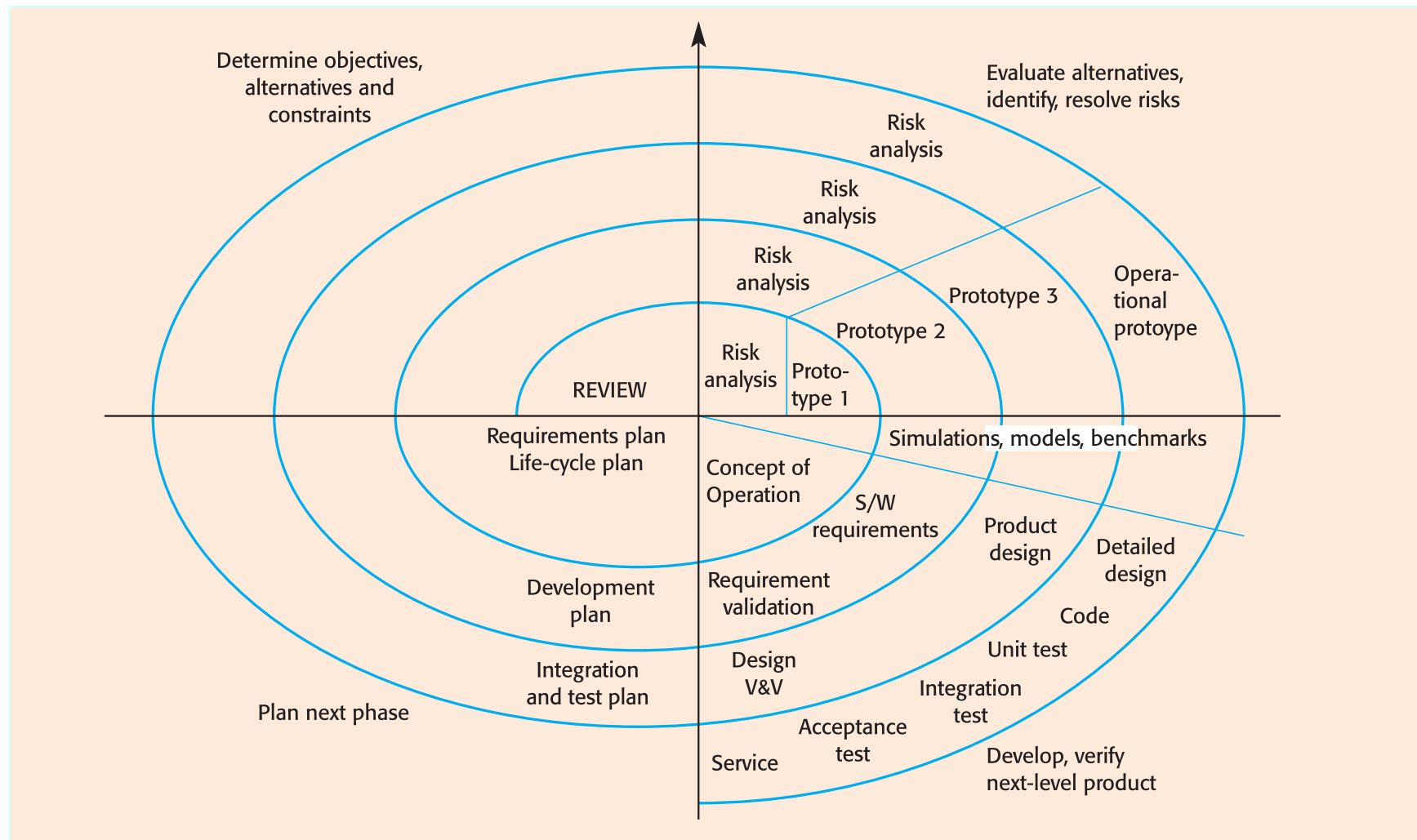
- System requirements ALWAYS evolve in the course of a project.
- Process iteration is always part of the process for large systems.
- Iteration can be applied to any of the generic process models.
  
- Two (related) approaches
  - Incremental delivery
  - Spiral development



# Spiral Development

- Process is represented as a spiral.
- 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.
- Risks are explicitly assessed and resolved throughout the process.
- Spiral model sectors
  - Objective setting
    - Specific objectives for the phase are identified.
  - Risk assessment and reduction
    - Risks are assessed and activities put in place to reduce the key risks.
  - Development and validation
    - A development model for the system is chosen which can be any of the generic models.
  - Planning
    - The project is reviewed and the next phase of the spiral is planned.

# Spiral Development Model

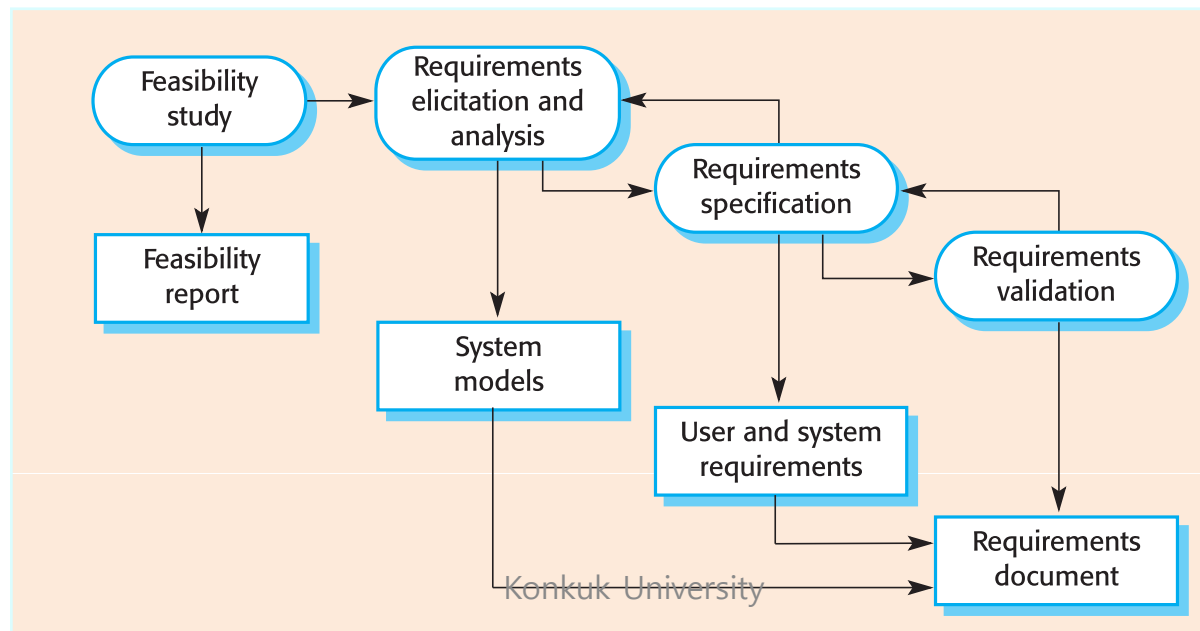


# Process Activities

- Software specification
- Software design and implementation
- Software validation
- Software evolution

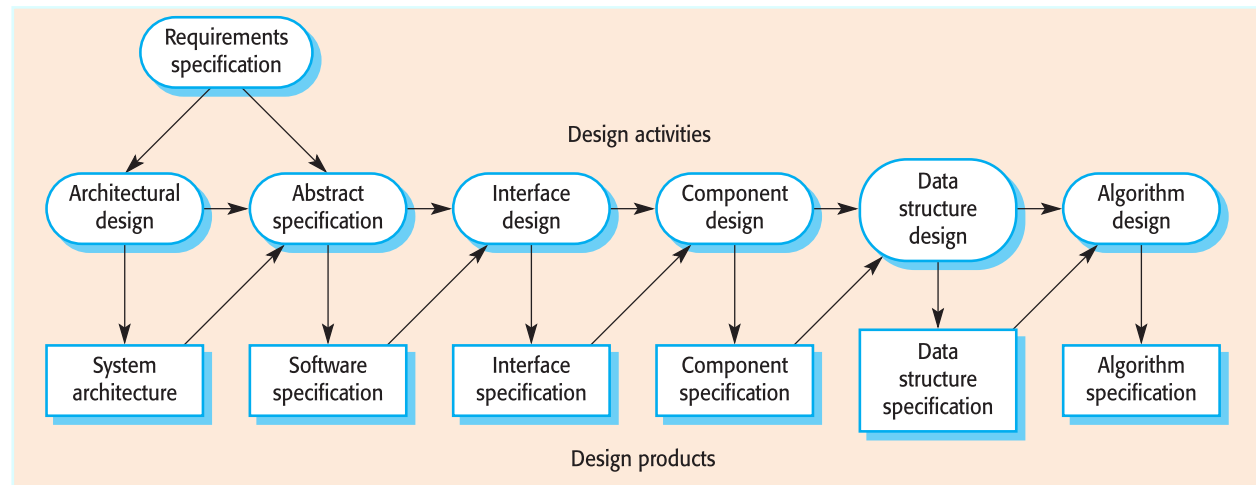
# Software Specification

- Process of establishing what services are required and constraints on the system's operation and development. (Requirements engineering)
- Requirements engineering process
  - Feasibility study
  - Requirements elicitation and analysis
  - Requirements specification
  - Requirements validation



# Software Design and Implementation

- Process of converting system specification into executable system.
- Software design
  - Design a software structure that realises the specification
  - Software design process



- Implementation
  - Translate this structure into an executable program.
  - Programming is a personal activity - there is no generic programming process.

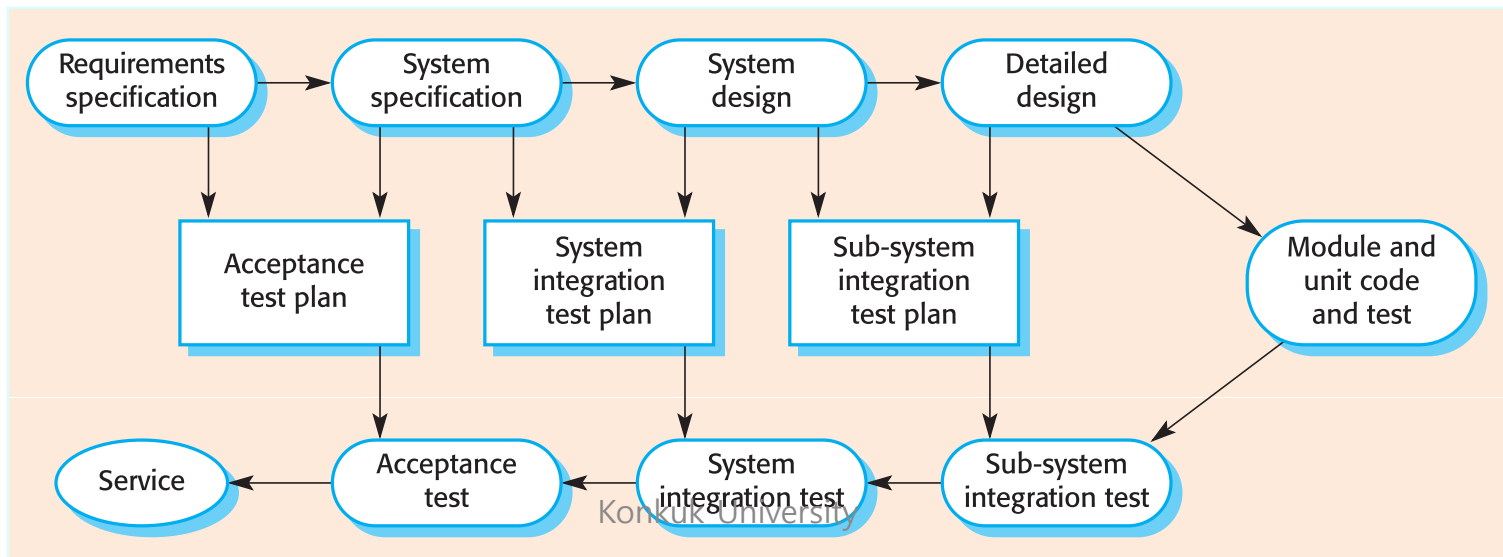


# Software Validation

- Verification and validation (V & V) is intended to show that
  - System conforms to its specification.
  - System meets requirements of the system customer.
  - Involves
    - Checking
    - Review processes
    - System testing
- System testing involves executing the system with test cases that are derived from its specification.

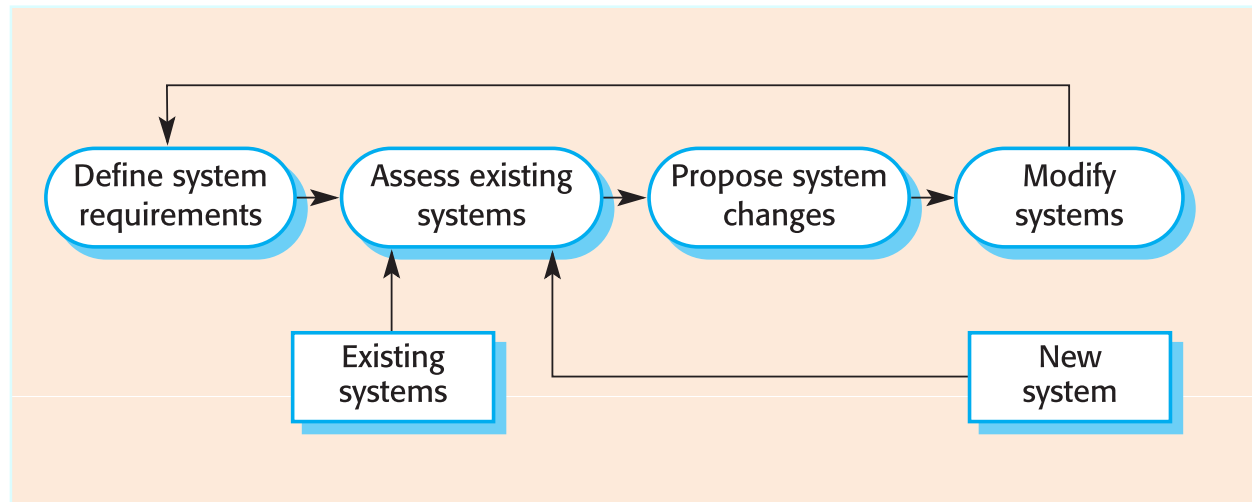
# Testing Stages and Phases

- Unit or Component testing
  - Individual components are tested independently.
  - Components may be functions, objects, or 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.



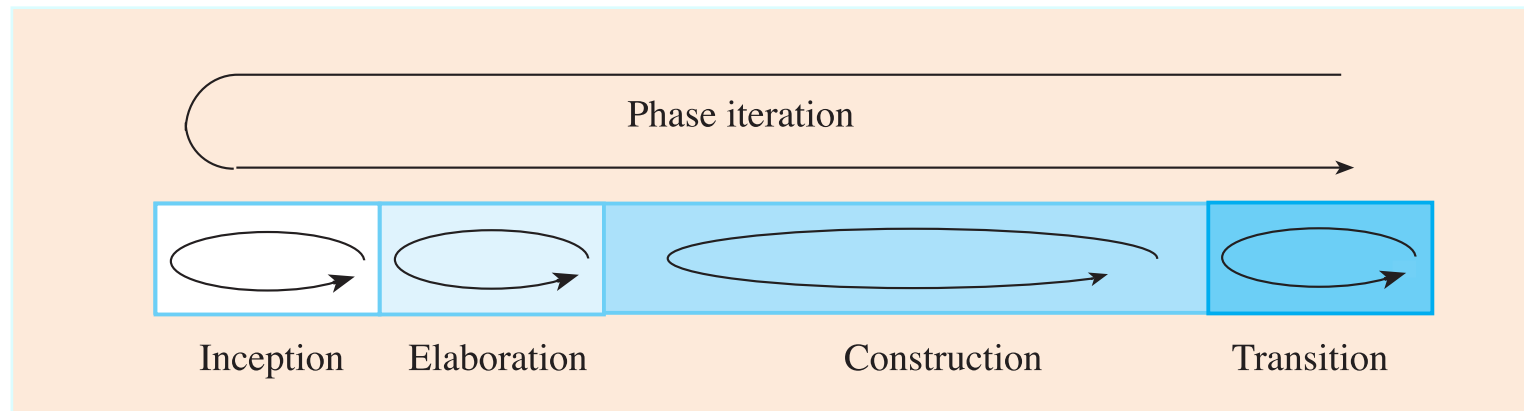
# Software Evolution

- Software is inherently flexible and can change.
- As requirements change through changing business circumstances, the software that supports the business must also evolve and change.



# The Rational Unified Process

- A modern process model derived from working groups on the UML.
- Normally described from 3 perspectives
  - Dynamic perspective that shows phases over time
  - Static perspective that shows process activities
  - Practice perspective that suggests good practice.
- 4 RUP phases : (Dynamic perspective)



# Workflows of the RUP (Static Perspective)

<b>Workflow</b>	<b>Description</b>
Business modelling	The business processes are modelled using business use cases.
Requirements	Actors who interact with the system are identified and use cases are developed to model the system requirements.
Analysis and design	A design model is created and documented using architectural models, component models, object models and sequence models.
Implementation	The components in the system are implemented and structured into implementation sub-systems. Automatic code generation from design models helps accelerate this process.
Test	Testing is an iterative process that is carried out in conjunction with implementation. System testing follows the completion of the implementation.
Deployment	A product release is created, distributed to users and installed in their workplace.
Configuration and change management	This supporting workflow managed changes to the system (see Chapter 29).
Project management	This supporting workflow manages the system development (see Chapter 5).
Environment	This workflow is concerned with making appropriate software tools available to the software development team.

# RUP Good Practice (Practice Perspective)

- Develop software iteratively.
- Manage requirements.
- Use component-based architectures.
- Model software Visually.
- Verify software quality.
- Control changes to software.

# Summary

- Software processes are the activities involved in producing and evolving a software system.
- Software process models are abstract representations of these processes.
- Generic process models describe organisation of software processes. Examples include the waterfall model, evolutionary development and component-based software engineering.
- Iterative process models describe the software process as a cycle of activities.
- General software process activities are specification, design and implementation, validation and evolution.
- The Rational Unified Process is a generic process model based on UML.

Chapter 5.  
Project Management



# Objectives

- To explain main tasks undertaken by project managers
- To introduce software project management and to describe its distinctive characteristics
- To discuss project planning and planning process
- To show how graphical schedule representations are used by project management
- To discuss the notion of risks and risk management process

# Software Project Management

- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and procuring the software.
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software.
- Project management distinctions:
  - Product is intangible.
  - Product is uniquely flexible.
  - Software engineering is not recognized as an engineering discipline with the same status as mechanical, electrical engineering, etc.
  - Software development process is not standardised.
  - Many software projects are 'one-off' projects.

# Project Management Activities

- Proposal writing
- Project staffing
- Project planning and scheduling
- Project costing
- Project monitoring and reviews
- Personnel selection and evaluation
- Report writing and presentations

# Project Staffing

- May not be possible to appoint ideal people to work on a project
  - Project budget may not allow for the use of highly-paid staff.
  - Staff with appropriate experience may not be available.
  - Organisation may wish to develop employee skills through performing software projects.
- Managers have to work within these constraints especially when there are shortages of trained staff.

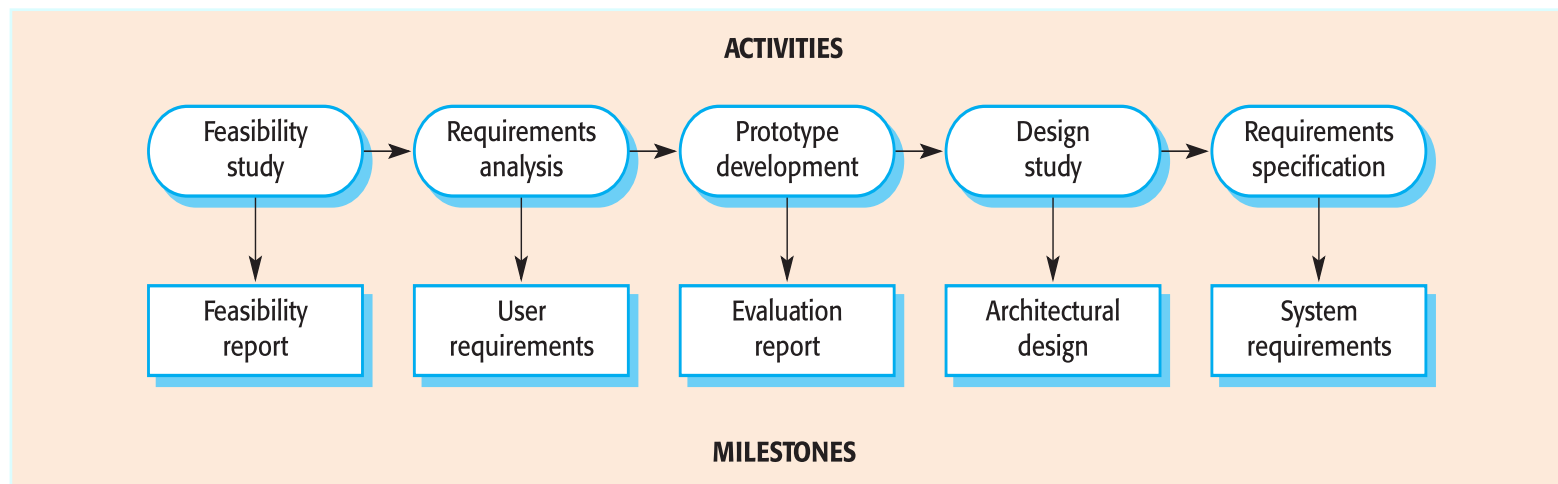
# Project Planning

- Probably the most time-consuming project management activity
- Continuous activity from initial concept through to system delivery
- Plans must be regularly revised as new information becomes available.
- Various different types of plan may be developed to support main software project plan that is concerned with schedule and budget.

<b>Plan</b>	<b>Description</b>
Quality plan	Describes the quality procedures and standards that will be used in a project. See Chapter 27.
Validation plan	Describes the approach, resources and schedule used for system validation. See Chapter 22.
Configuration management plan	Describes the configuration management procedures and structures to be used. See Chapter 29.
Maintenance plan	Predicts the maintenance requirements of the system, maintenance costs and effort required. See Chapter 21.
Staff development plan.	Describes how the skills and experience of the project team members will be developed. See Chapter 25.

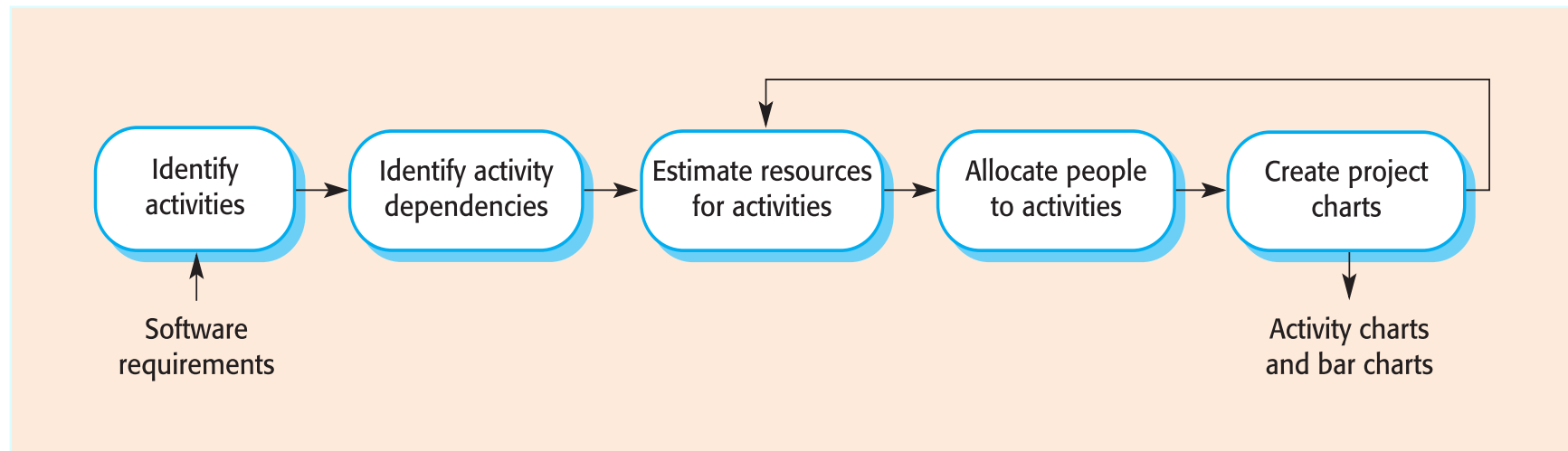
# Project Planning Process

- Activities: produce tangible outputs for management to judge progress
- Milestones : end-point of a process activity
- Deliverables : project results delivered to customers
- Waterfall process allows straightforward definition of progress milestones.

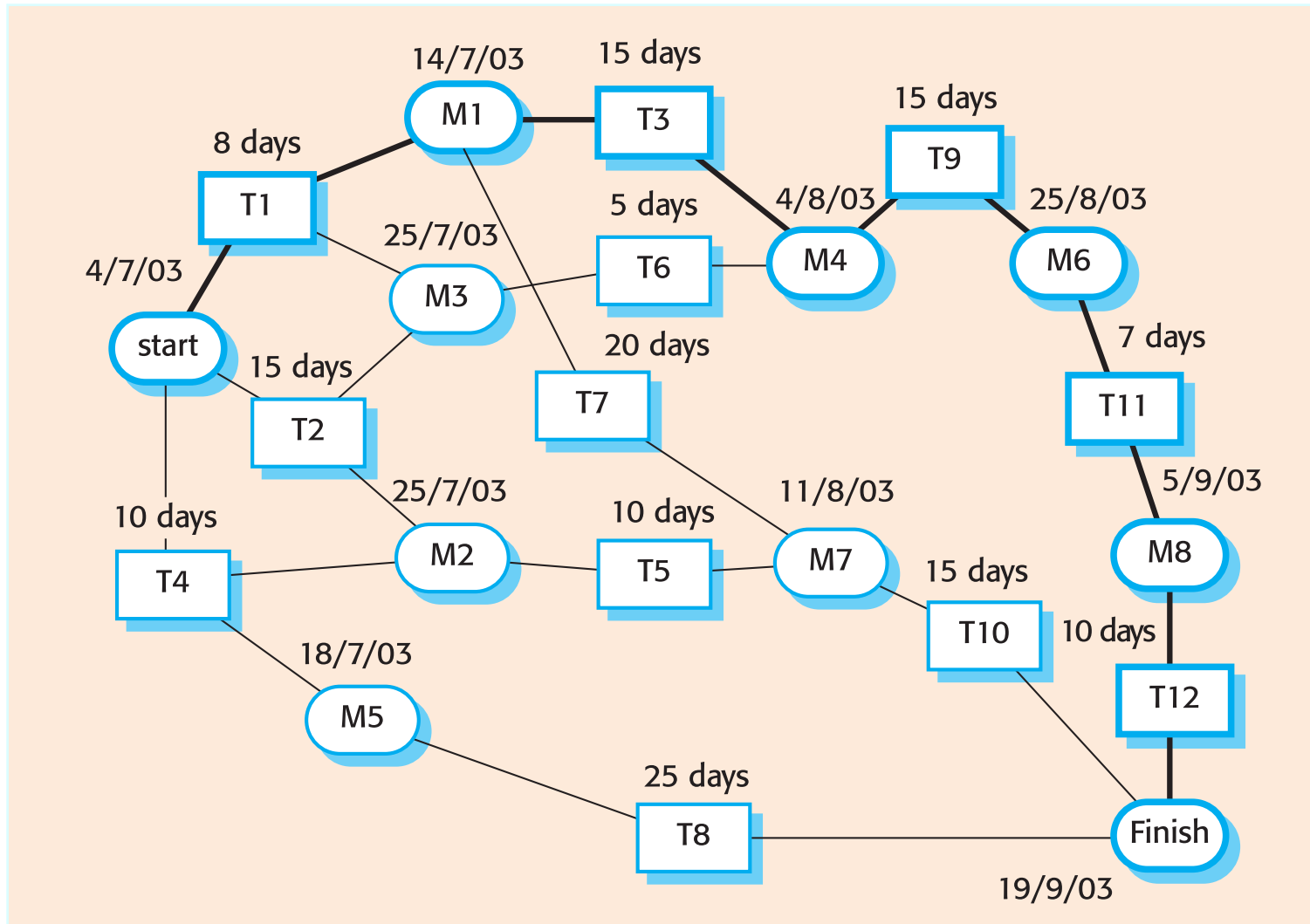


# Project Scheduling Process

- Split project into tasks and estimate time and resources required to complete each task.
- Organize tasks concurrently to make optimal use of workforce.
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete.
- Depend on project manager's intuition and experience.

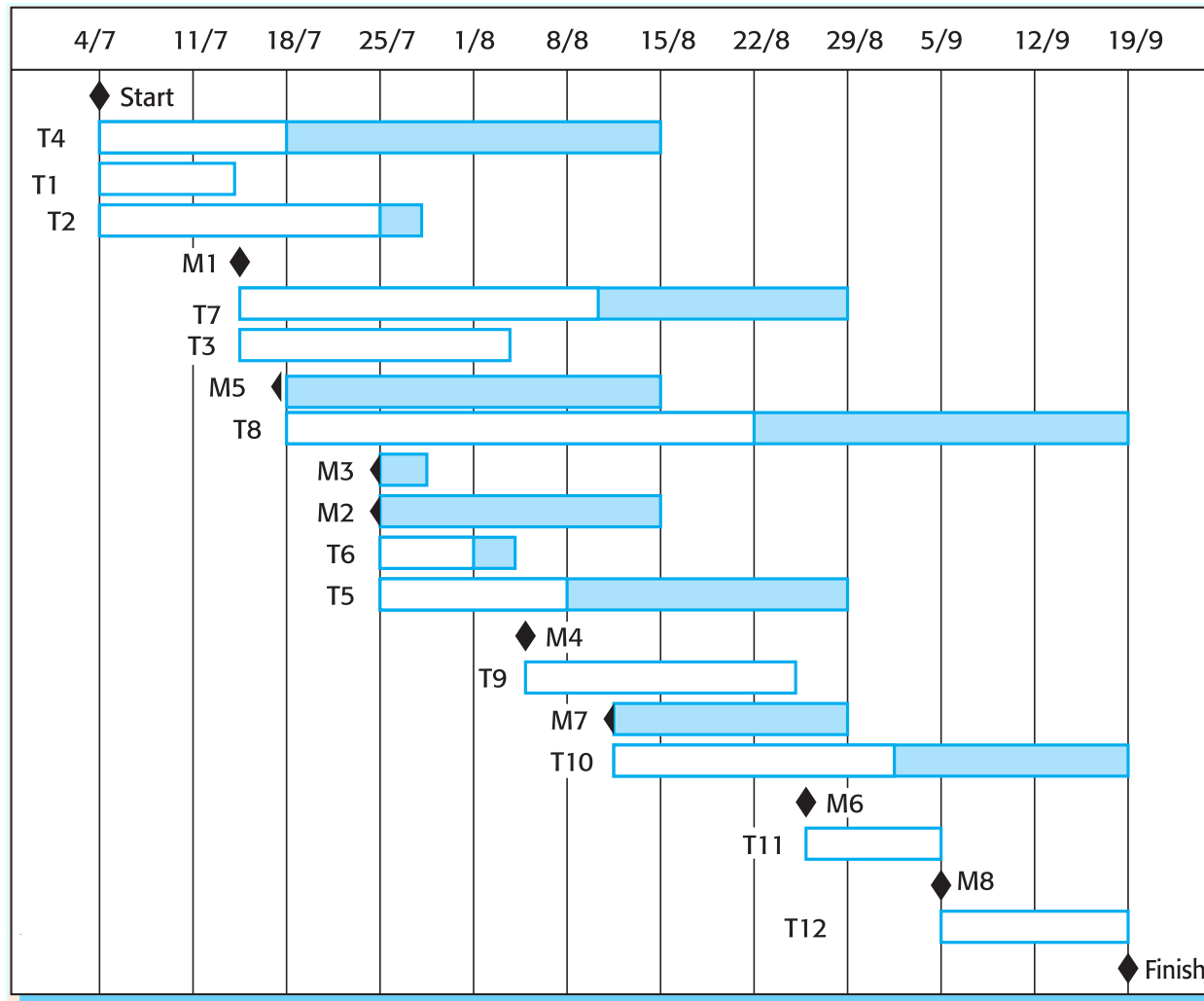


# Activity Network

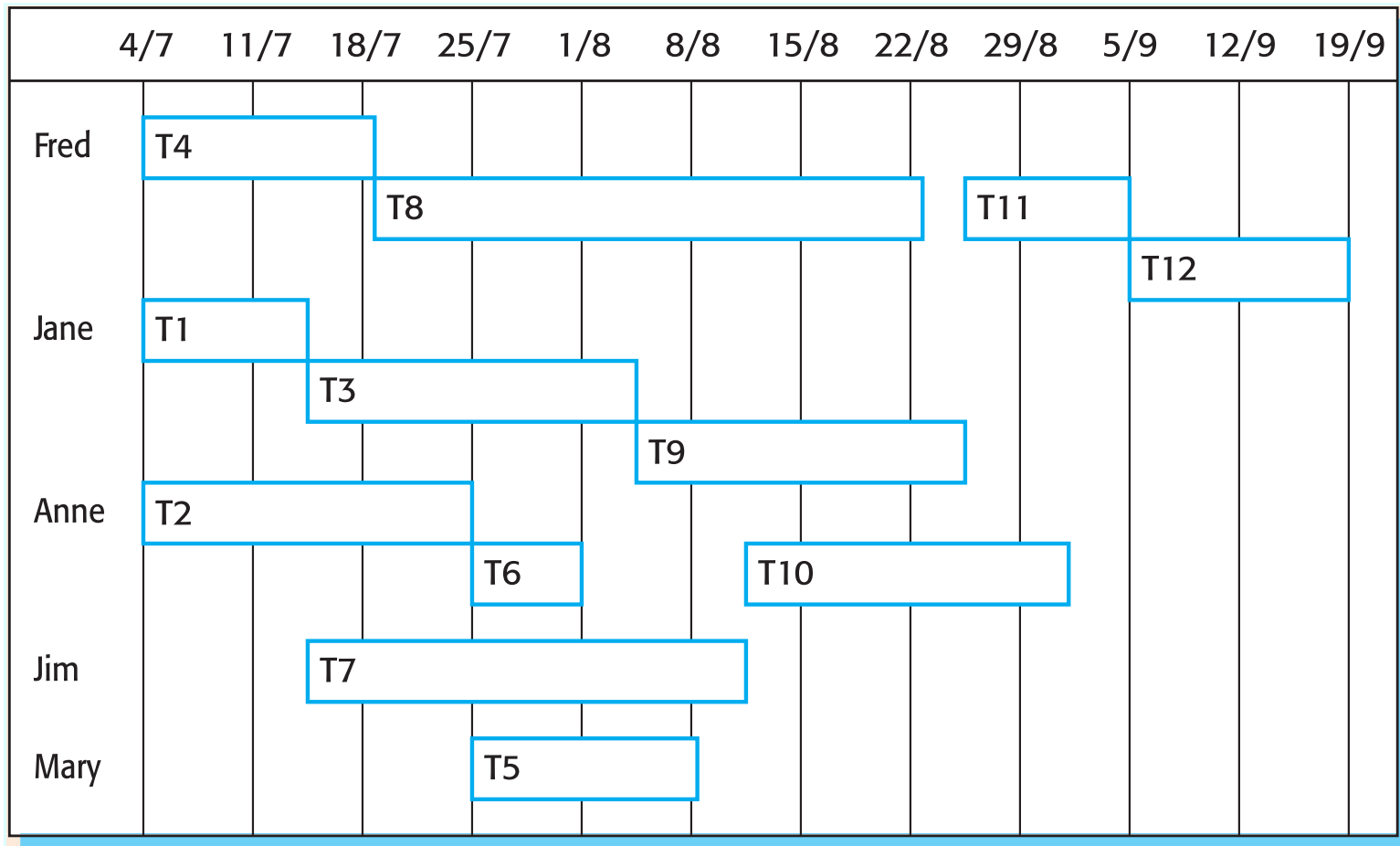




# Activity Timeline

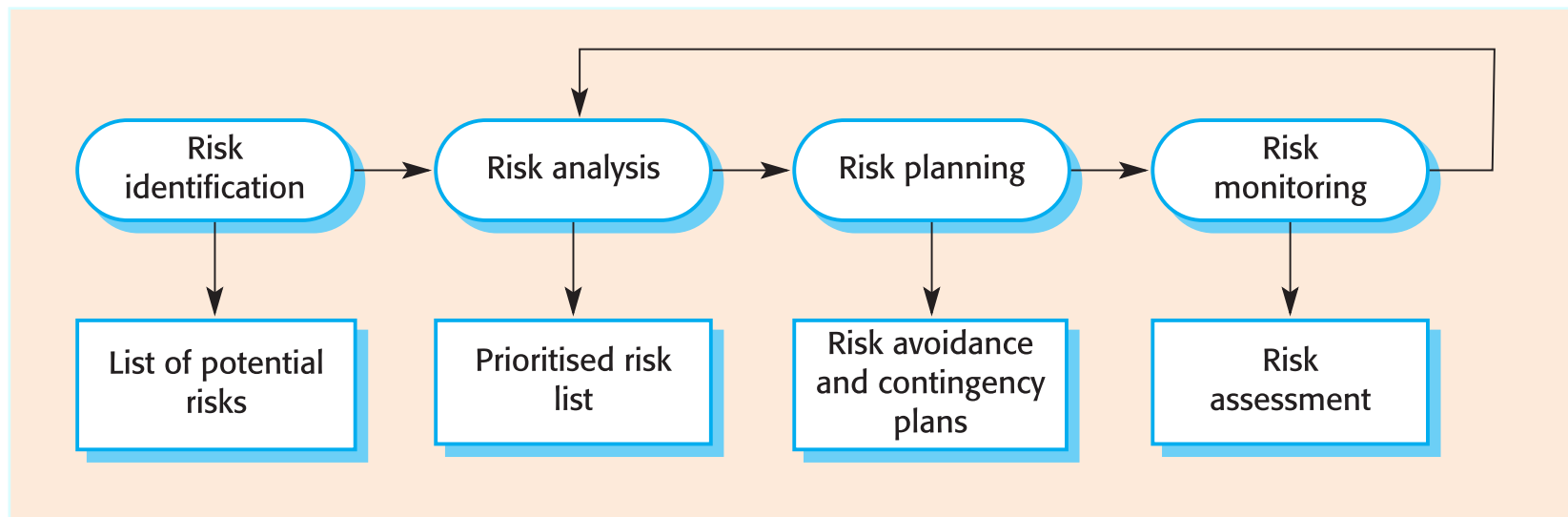


# Staff Allocation



# Risk Management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- A risk is a probability that some adverse circumstance will occur
  - Project risk affects schedule or resources.
  - Product risk affects quality or performance of the software being developed.
  - Business risk affects the organisation developing or procuring the software.
- Risk management process



# Summary

- Good project management is essential for project success.
- Intangible nature of software causes problems for management.
- Managers have diverse roles but their most significant activities are planning, estimating and scheduling.
- Project milestone is a predictable state where a formal report of progress is presented for management.
- Project scheduling involves preparing various graphical representations showing project activities, their durations and staffing.
- Risk management is concerned with identifying risks which may affect the project, and planning to ensure that these risks do not develop into major threats.