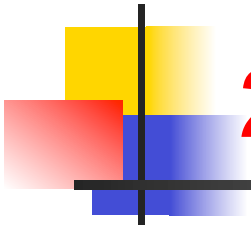
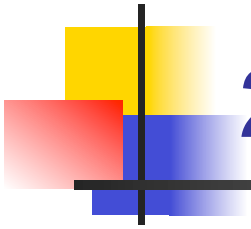


2. Life-Cycle Perspective

Overview

- 2.1 Motivation
- 2.2 Waterfall Model
- 2.3 Requirements in Context

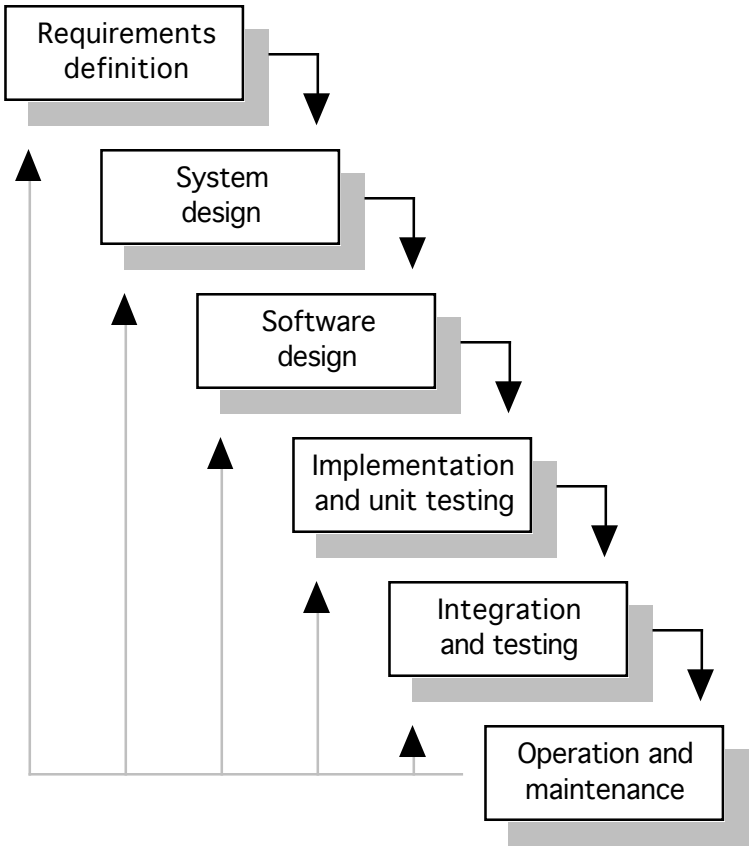


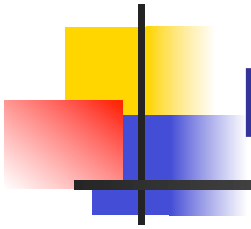
2.1 Motivation

- Modern software engineering is based on the premise that design decisions and planning must consider the entire life of the product
- A narrow (development only) perspective is likely to lead to failures, lack of dependability and later expenses much greater than the development costs
- Maintainability, enhanceability, portability, etc. are fundamental life-cycle concerns
- The life cycle starts with the requirements definition

2.2 Waterfall Model

- Understanding requirements presupposes a good grasp of the development process as a whole
- This model remains one of the best abstractions for the software development process





Multiple Perspectives

- Waterfall model
  - product focused
- Evolutionary
  - increment driven
  - rapid prototyping
- Spiral (Boehm)
  - risk analysis driven
  - planning, risk assessment, engineering, customer evaluation
- Transformational
  - specification driven
  - formal methods

2.3 Requirements in Context

- Requirements may vary in level of abstraction and contents from one context to another
- System requirements are the result of an analysis or discovery process
- Software requirements are the result of a design process involving requirements allocation
- Sometimes there is no distinction between them