Software Development Process

- A general model of the software development process is called a *software life cycle* software life cycle model represents all the activities and work products necessary to develop a software system.
- Life cycle models enable managers and developers to deal with the complexity of the process of developing software
- Models of the software development process, called **software life cycle models**.

SDLC-Activity vs Entity Centered

- Most proposed life cycle models have focused on the activities of software development and represented them explicitly as class objects. This view of the software life cycle is called **activity centered**.
- Another view of the software life cycle is to focus on the work products created by these activities. This other view is called **entity centered**.
- The activity-centered view leads participants to focus on how work products are created.
- The entity-centered view leads participants to focus on the content and structure of the work products.

Note: A work product is a visible outcome of a task Examples A document A review of a document A presentation A piece of code A test report Work products delivered to the customer are called deliverables

- Figure depicts a simple life cycle for software development using three activities: Problem definition, System development, and System operation.

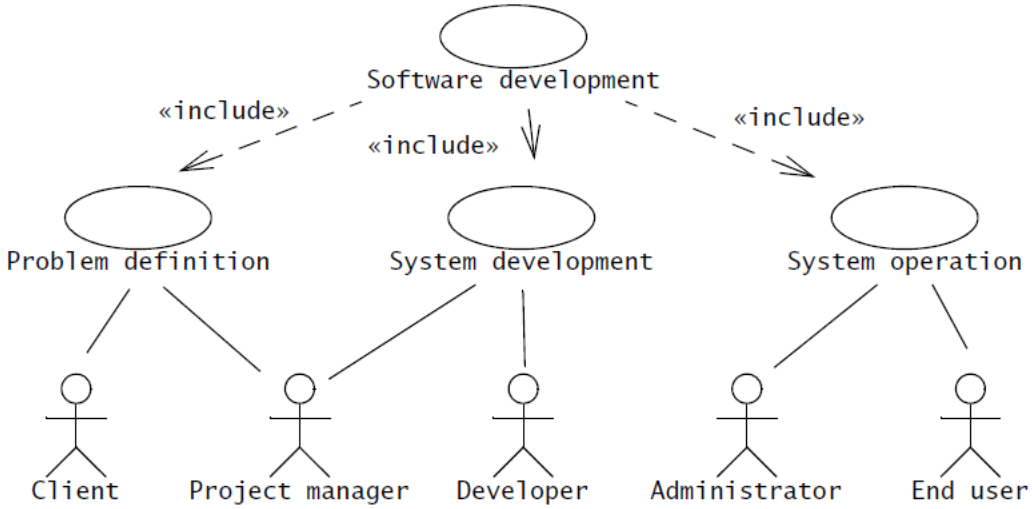


Figure 15-1 Simple life cycle for software development (UML use case diagram).

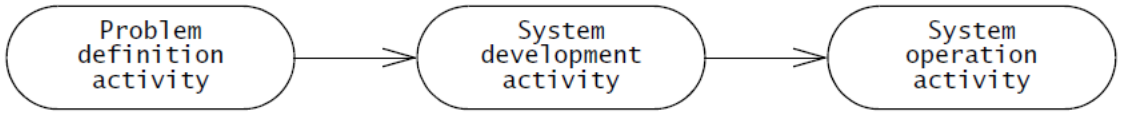


Figure 15-2 Simple life cycle for software development (UML activity diagram).

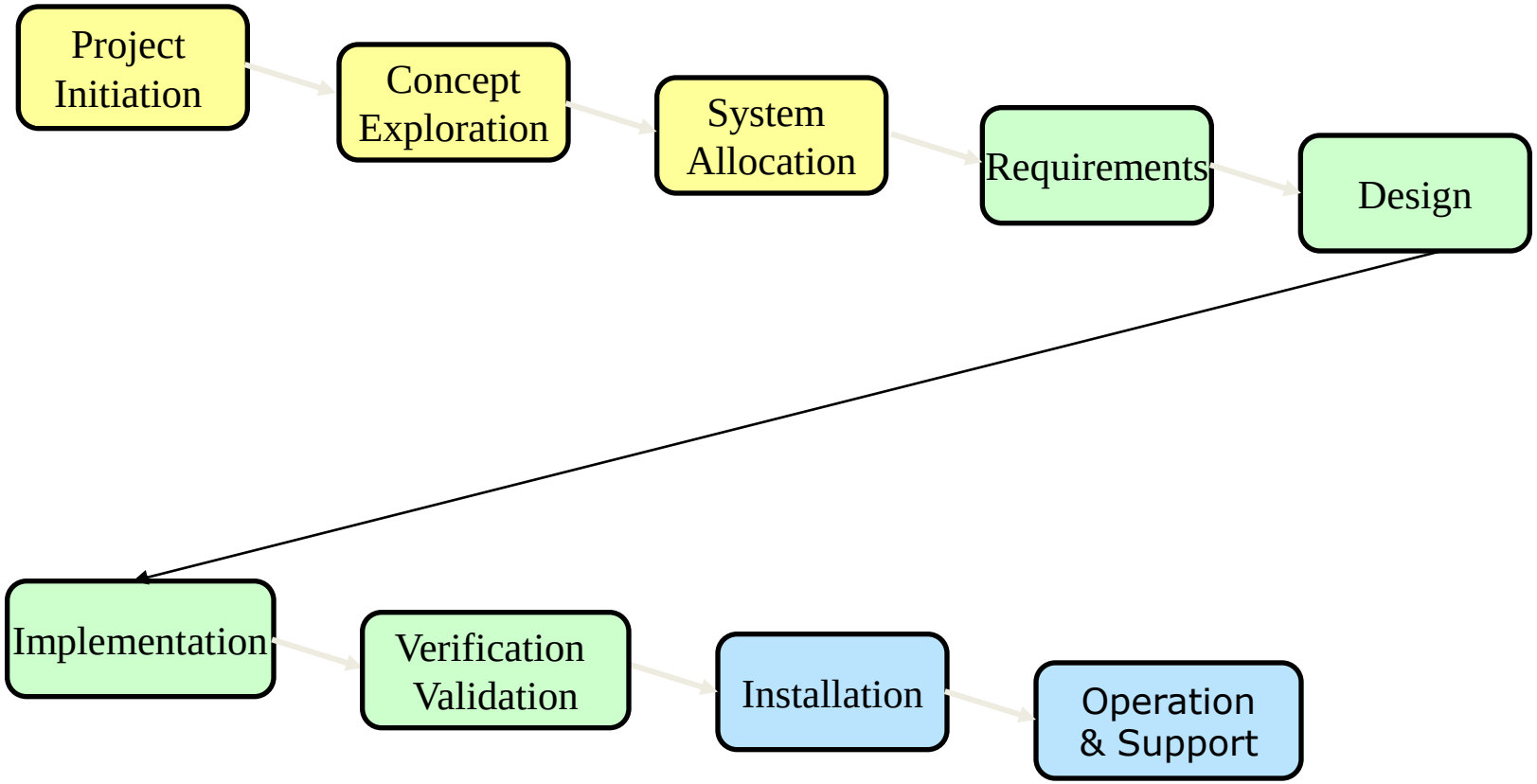
Generic Life Cycle models

- Sequential
  - The waterfall method
- Iterative
  - Spiral

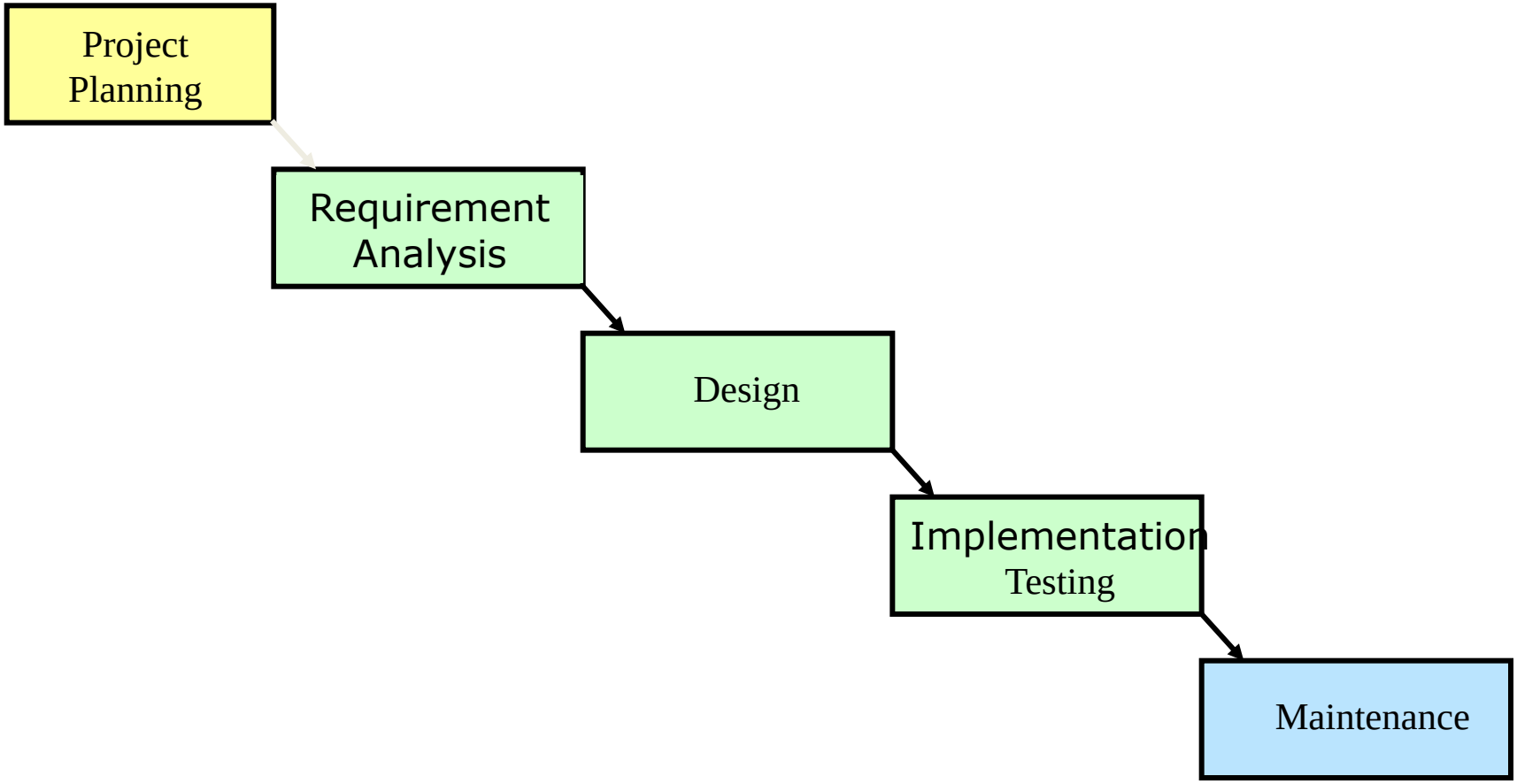
Sequential Activity-Centered Models

- The oldest software life cycle model is the waterfall model, which prescribes a sequential execution of the activities.
- All requirements activities are completed before the system design activity starts.
- The goal is never to turn back once an activity is completed.
- The key feature of this model is the constant verification activity which ensures that each development activity does not introduce unwanted or delete mandatory requirements.
- This model provides a simple view of software development that measures progress by the number of tasks that have been completed.
- The model assumes that software development can be scheduled as a step-by-step process that transforms user needs into code.

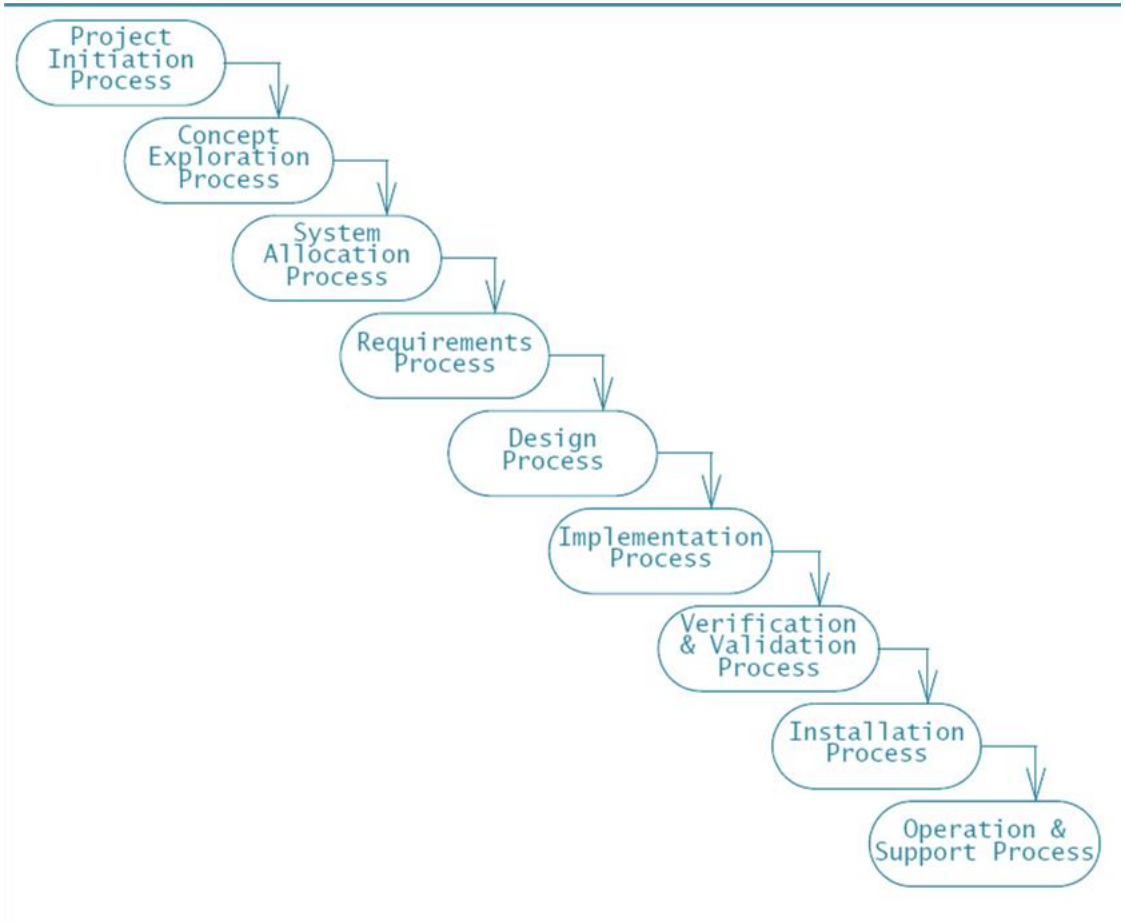
Waterfall Process Model



Waterfall Process Model



waterfall model



Concept Exploration

- Identifying idea and need for a product
- Formulate possible approaches to solution
- Determining feasibility, identifying and mitigating risks
- Setting up funding/support

System allocation

- Estimating, planning and identifying staffing needs for each phase

Requirements Analysis

- gathering, specifying and analyzing the requirements.
- Defining functionality and scope
- Define interface requirements
- Prioritize requirements
- Validate and Verify requirements

System and Software Design

- Design how software will be structured internally
- Establish system architecture
- Analysis of the design and verification that the design meets the requirements

Implementation

- Programming
- Verification that implemented software meets all requirements
  - Plan unit implementation and integration (implementation/test)
- Create user manual (operating instructions)

Testing (verification of implementation phase)

- Testing modules (unit test)
- Testing combinations of modules (integration test)
- Testing the whole system (system test)

Installation

- Installation at user site (plan test accept)
- User testing, verifying user needs are met
- Errors corrected, bugs fixed

Operation and Support (maintenance)

- Software enhanced (new features, new requirements)
- Errors corrected, bugs fixed
- Evolution to newer platforms (newer OS, hardware)

Advantages: Waterfall Process Model

- Phases (processes) are performed once
  - Makes the process easy to follow
  - One phase should not be started until the previous one is complete
  - Makes the process easy to manage
  - Documents need only be produced and approved once
- The waterfall method should only be used if the requirements are well understood

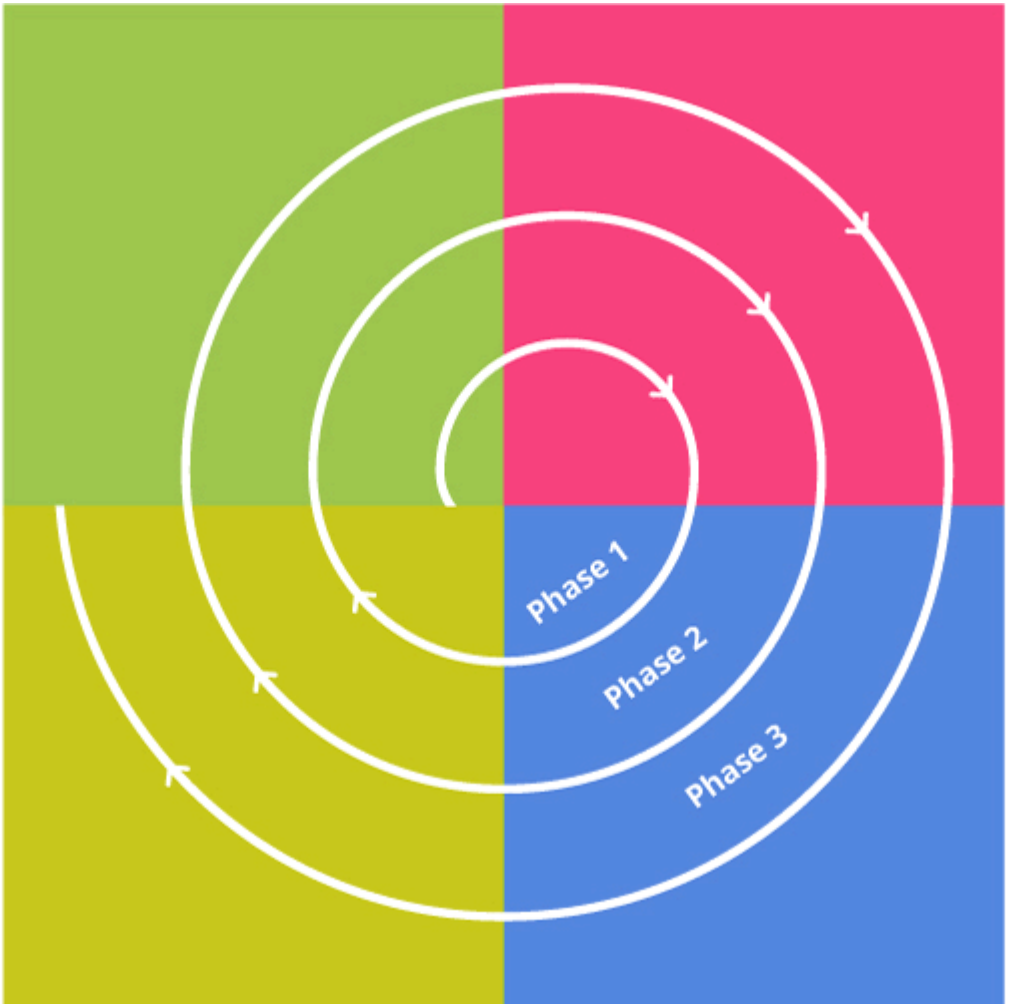
Waterfall Model:Disadvantages

- Impractical to fully complete any one of the phases before starting the next one
  - difficult to capture all requirements before proceeding to design …
- Users don't get to see the results until the end
  - problems may emerge only at the end when user realizes the product is not what was needed

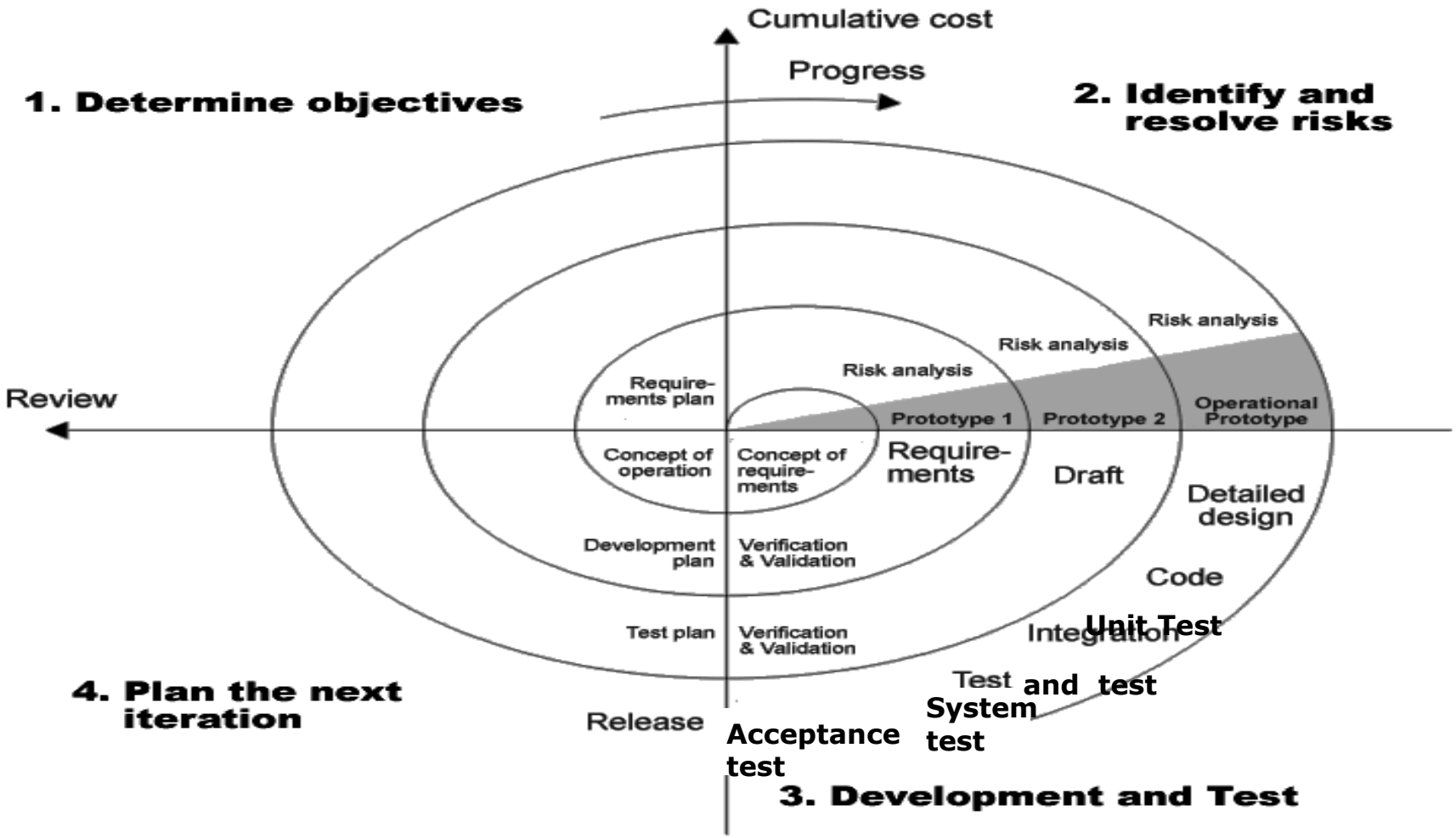
Spiral Model

- To address the source of weaknesses in the waterfall model
- In particular, to accommodate infrequent change during the software development.
- It is based on the same activities as the waterfall model; however, it adds several activities such as risk management, reuse, and prototyping to each activity. These extended activities are called *cycles* or *rounds*.
- Each round is composed of four phases

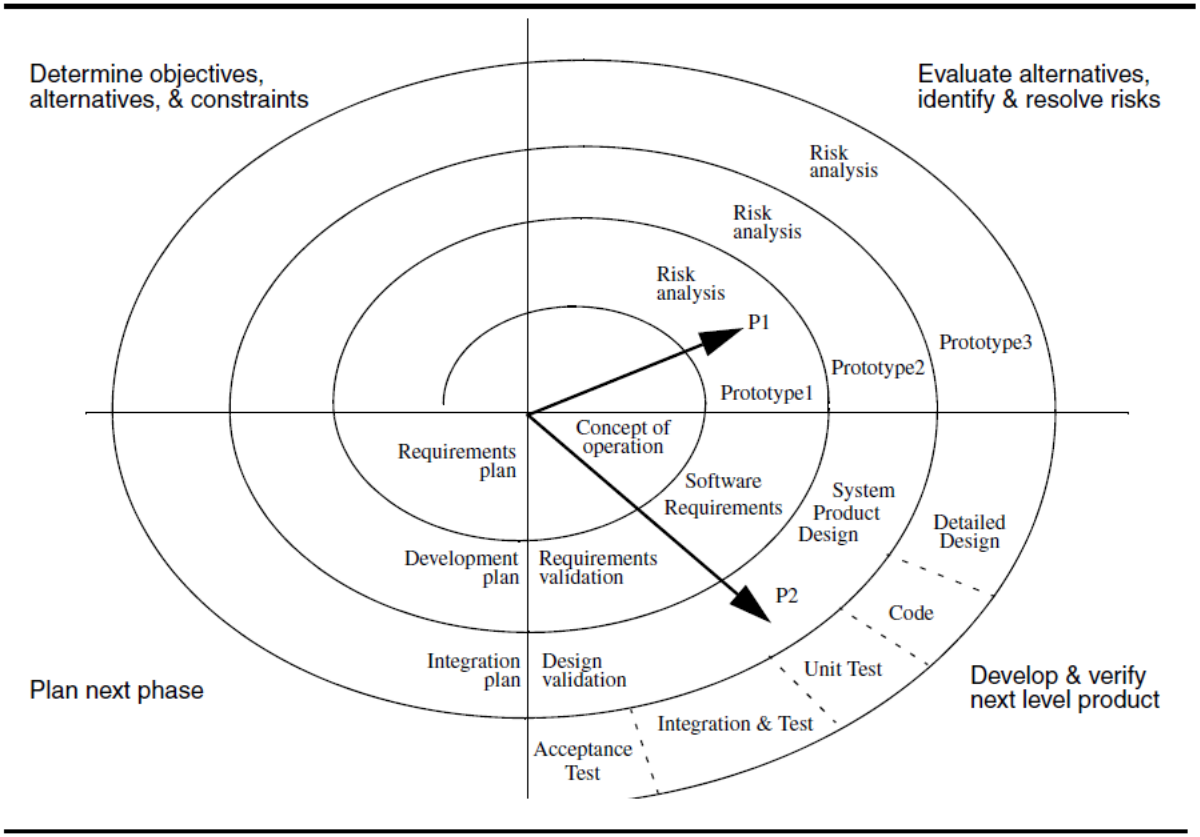
Understanding Each Quadrant of the Spiral



Boehm's spiral model



Iterative Activity-Centered Models
Spiral Model



- During the first phase (upper left quadrant), developers explore alternatives, define constraints, and identify objectives.
- During the second phase (upper right quadrant), developers manage risks associated with the solutions defined during the first phase.
- During the third phase (lower right quadrant), developers realize and validate a prototype or the part of the system associated with the risks addressed in this round.
- The fourth phase (lower left quadrant) focuses on planning the next round based on the results of the current round.
- The last phase of the round is usually conducted as a review involving the project participants, including developers, clients, and users.
- This review covers the products developed during the previous and current rounds and the plans for the next round.

Each round follows the waterfall model and includes the following activities:

- 1. Determine objectives
- 2. Specify constraints
- 3. Generate alternatives
- 4. Identify risks
- 5. Resolve risks
- 6. Develop and verify next-level product
- 7. Plan.

- The first two activities define the problem addressed by the current cycle.
- The third activity, *Generate alternatives*, defines the solution space.
- The activities *Identify risks* and *Resolve risks* identify future problems that may result in high cost or cancellation of the project.
- The activity *Develop and verify next-level product* is the realization of the cycle.
- The activity *Plan* is a management activity to prepare for the next cycle.
- These rounds can be viewed in a polar-coordinate system shown in Figure.
- The first round, *Concept of Operation*, starts in the upper left quadrant. Subsequent rounds are represented as additional layers on the spiral.
- The notation makes it easy to determine the status of the project at any time.

Each round follows the waterfall model and includes the following activities:

1. Determine objectives
2. Specify constraints
3. Generate alternatives
4. Identify risks
5. Resolve risks
6. Develop and verify next-level product
7. Plan.