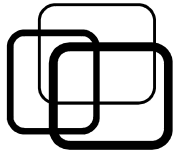
Software Process

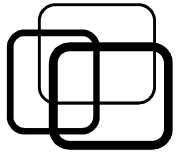
Inst. Nguyễn Minh Huy

Contents



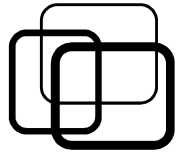
- Basic concepts.
- Software process models.

Contents



- **Basic concepts.**
- Software process models.

Basic concepts



- Working with no-process:.
  - No well-organized steps.
  - No pre-defined policies.
  - Emotional decisions.

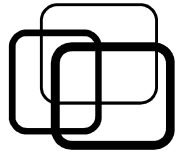
➔ Unpredictable.

➔ Uncontrollable.

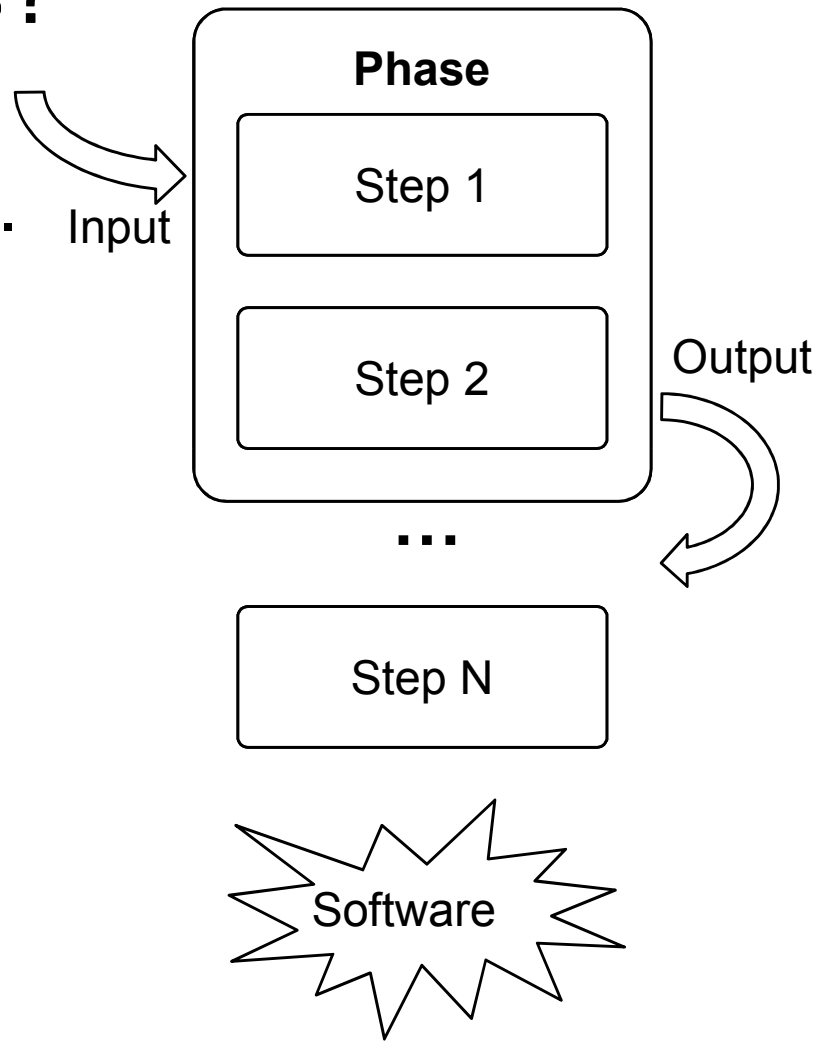
➔ Unrepeatable.

➔ Chaotic and risky!!

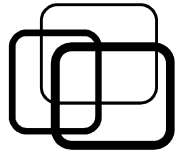
Basic concepts



- What is software process?
  - A series of steps.
  - Pre-defined, well-organized.
  - Produce software product.
- Process phase:
  - A group of related steps.
  - Produce specific products.
  - Each phase defines:
    - WHAT: to do.
    - WHO: will join.
    - INPUT: resources.
    - OUTPUT: products.

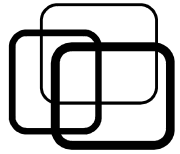


Basic concepts

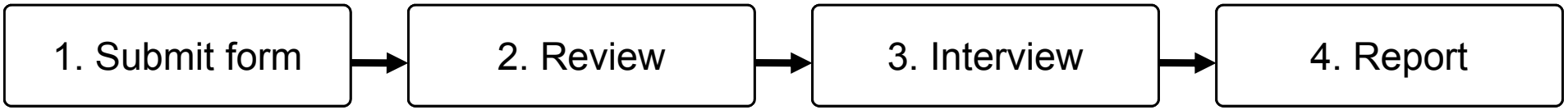


- Process description:
  - Workflow:
    - Follow step-by-step.
    - Popular way to describe process.
  - Dataflow:
    - Follow one data unit.
  - Role/Action:
    - Follow one role.

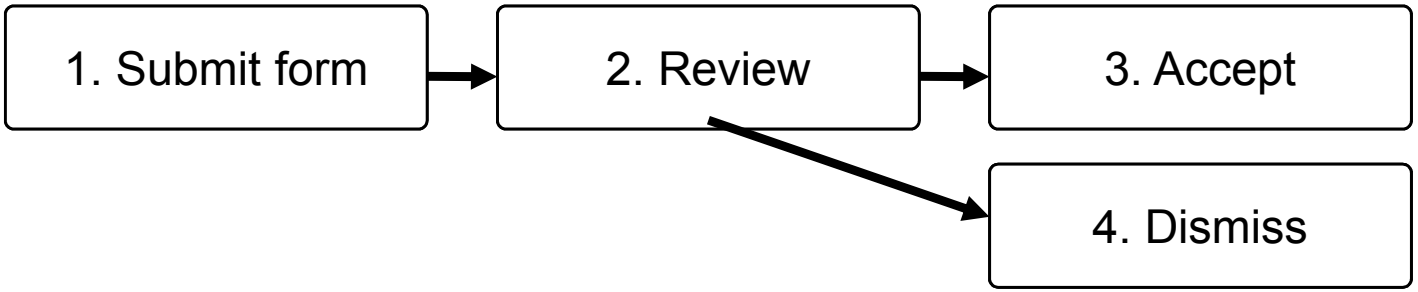
Basic concepts



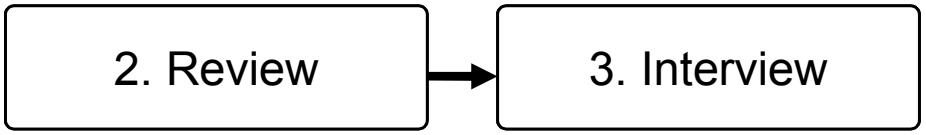
- Job interview process:
  - Workflow:



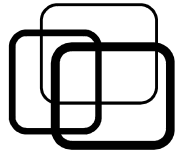
  - Dataflow: follow “Form”



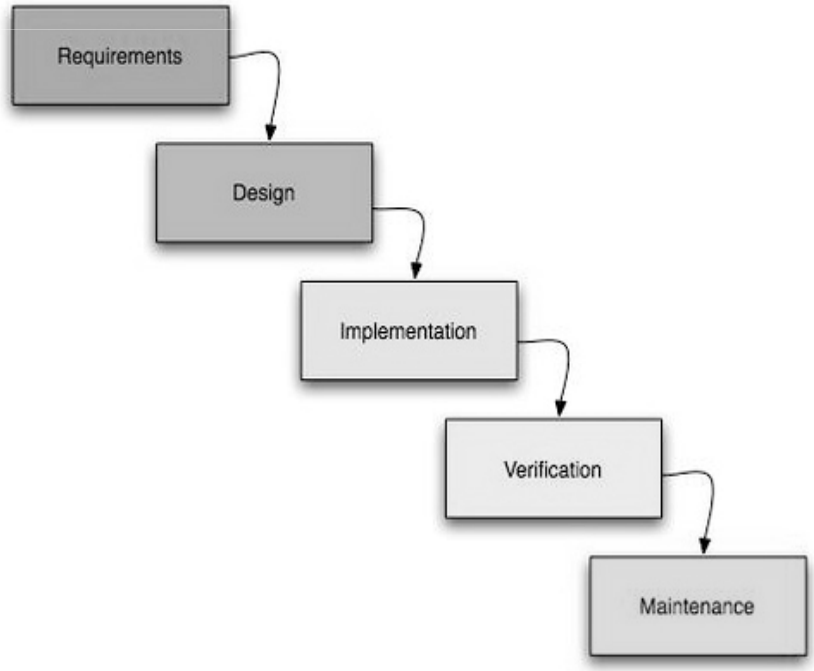
  - Role/Action: follow “Interviewer”,



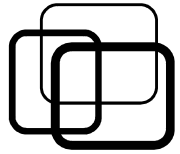
Basic concepts



- Main phases:
  - Requirement Analysis.
  - Architecture & Design.
  - Code Implementation.
  - Testing.
  - Maintenance.

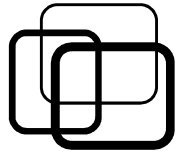


Basic concepts



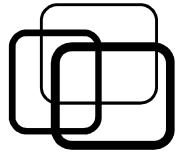
- Requirement Analysis:
  - Answer question: **WHAT**.
  - Identify what to do.
  - Role: BA (**B**usiness **A**nalyst).
  - Activities:
    - Gather user needs.
    - Write software requirement specifications.
    - Verify requirements.
    - Model requirements.

Basic concepts



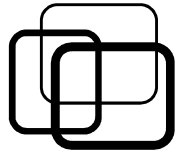
- Architecture & Design:
  - Answer question: **HOW**.
  - Find solutions and how to do.
  - Role: software architect.
  - Activities:
    - High-level design.
      - Architecture design.
    - Low-level design.
      - Data design.
      - User interface design.
      - Interaction design.

Basic concepts



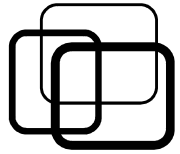
- Code Implementation:
  - Produce code from design.
  - Role: developer.
  - Activities:
    - Coding/ Debugging.
    - Implement database.

Basic concepts



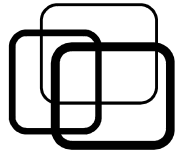
- Testing:
  - Find errors from code:
    - Based on design & requirements.
    - Based on users.
  - Role: tester.
  - Activities:
    - Integration test.
    - System test.

Basic concepts



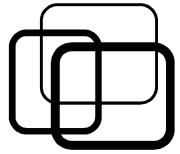
- Maintenance:
  - Operate, update, extend.
  - Role:
    - Technical support team.
    - Developer.
  - Activities:
    - Set up computers & network.
    - Install & set up system.
    - Report & fix bugs.

Contents

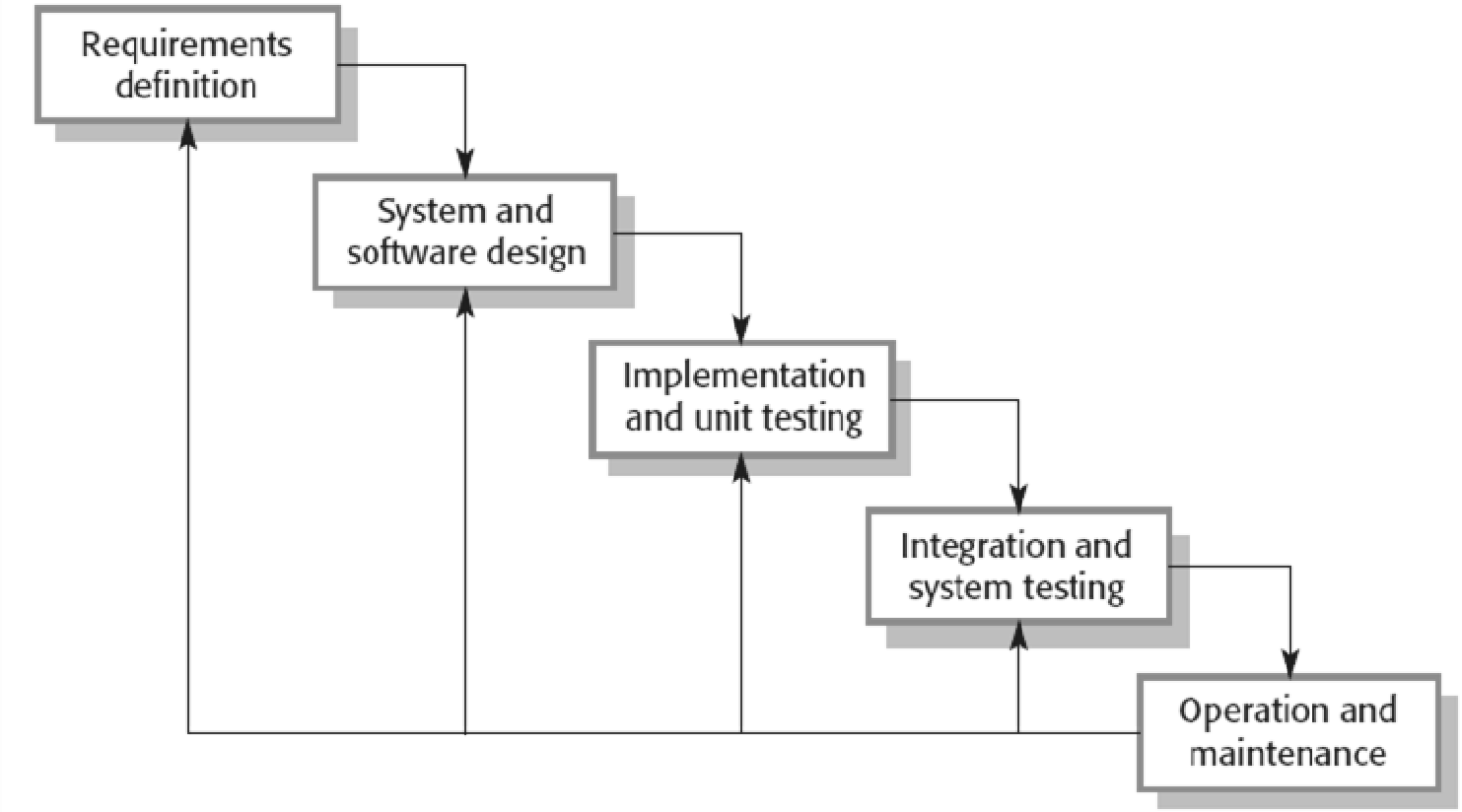


- Basic concepts.
- **Software process models.**

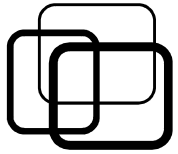
Software process models



- Waterfall Model:
  - Winston Royce,1970.



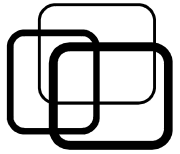
Software process models



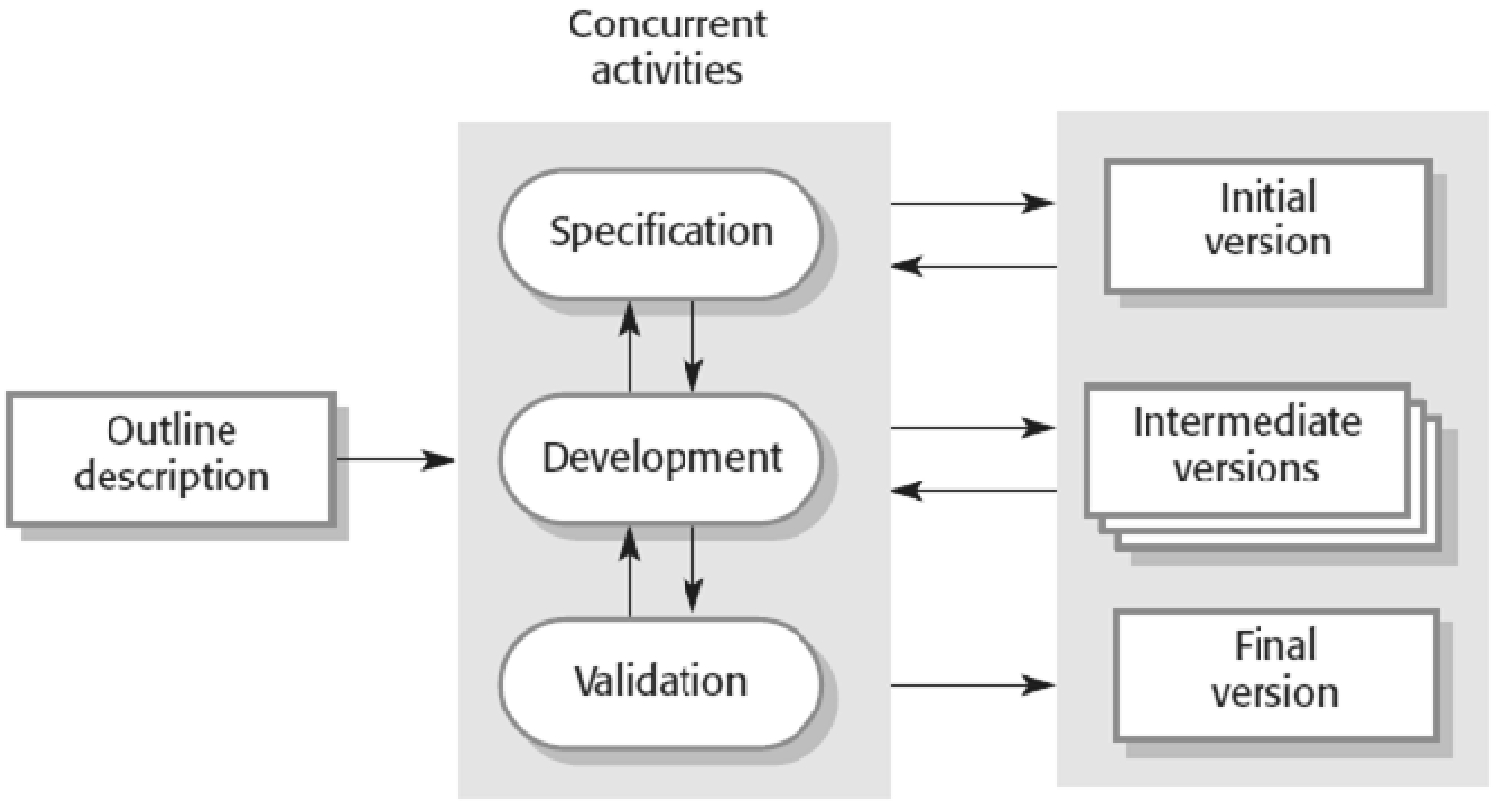
- Waterfall Model:
  - Characteristics:
    - SEQUENTIAL and INDEPENDENT phases.
    - Drafting before implementation.
    - Testing after implementation.
  - Advantages:
    - Pre-defined and well-organized.
    - Planning.
    - Predictive process.
  - Disadvantages:
    - Not welcome changes.
    - Suitable for well-defined and stable requirements.

    ➔ Improve: Backward Waterfall.

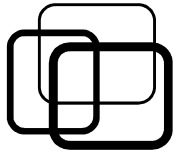
Software process models



- Evolutionary Model:
  - Improve Code & Fix.

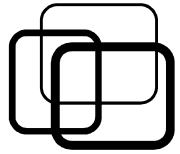


Software process models



- Evolutionary Model:
  - Characteristics:
    - ITERATIVE phases.
    - Fast draft and implementation.
    - Improve through versions and feedback.
  - Advantages:
    - Requirements can be changed and unclear.
    - Adaptive process.
  - Disadvantages:
    - Unpredictable.
    - Low quality design.

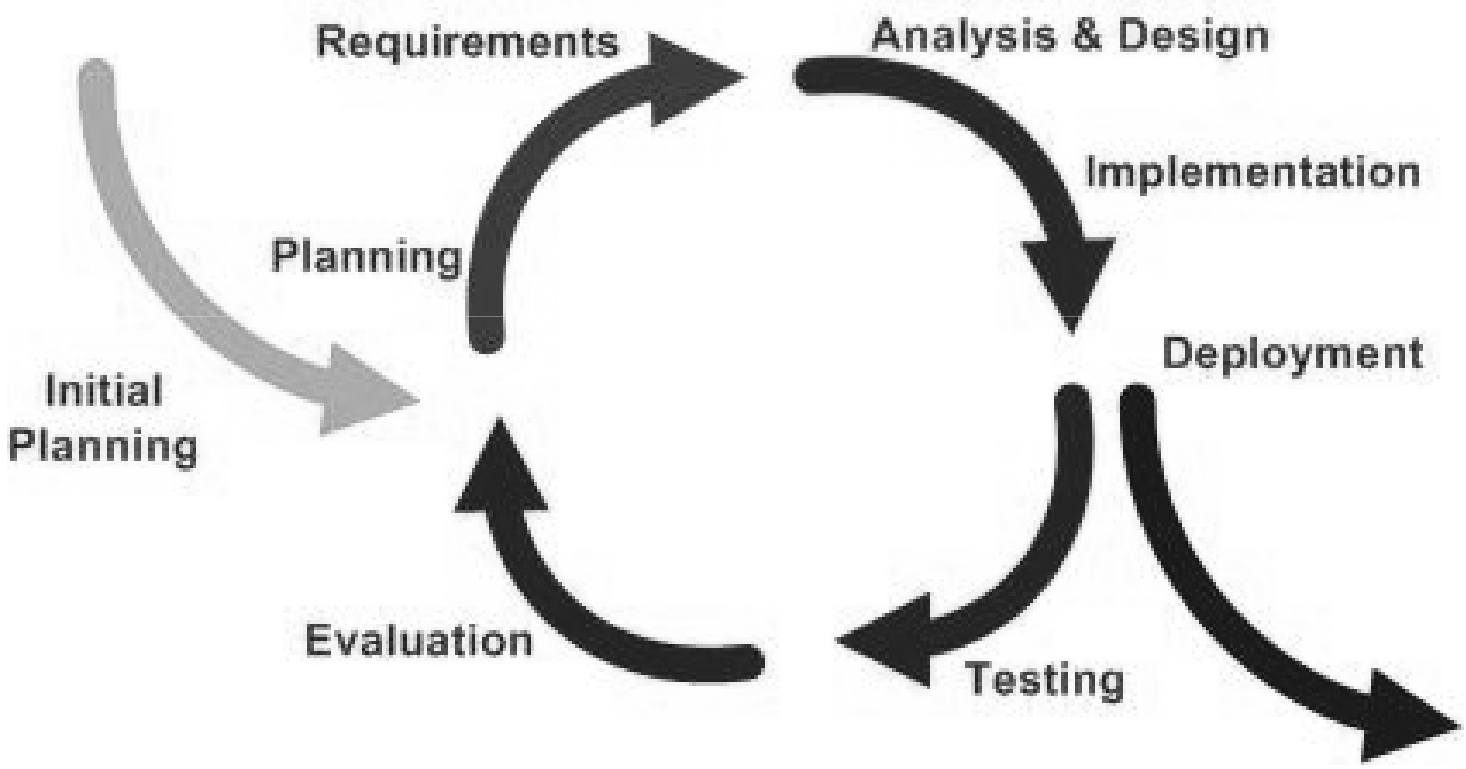
Software process models



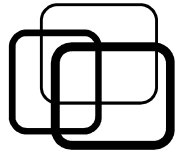
- Used Evolutionary Models:
  - Iterative Waterfall.
  - Spiral.
  - RUP.
  - Agile.

Software process models

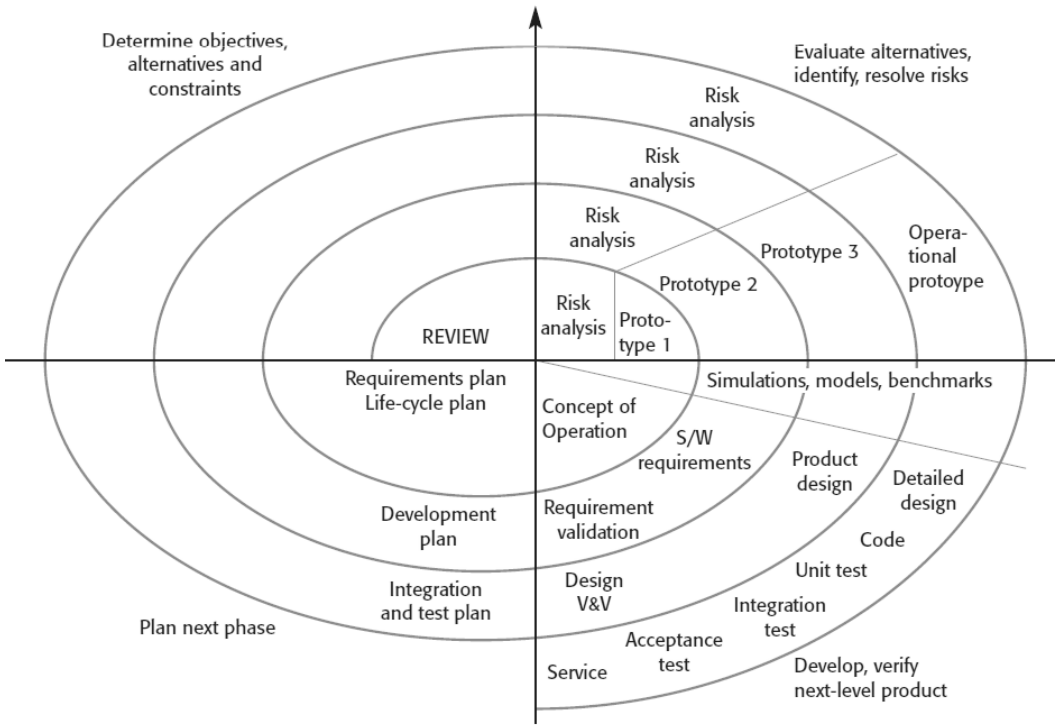
- Iterative Waterfall:



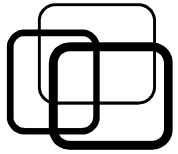
Software process models



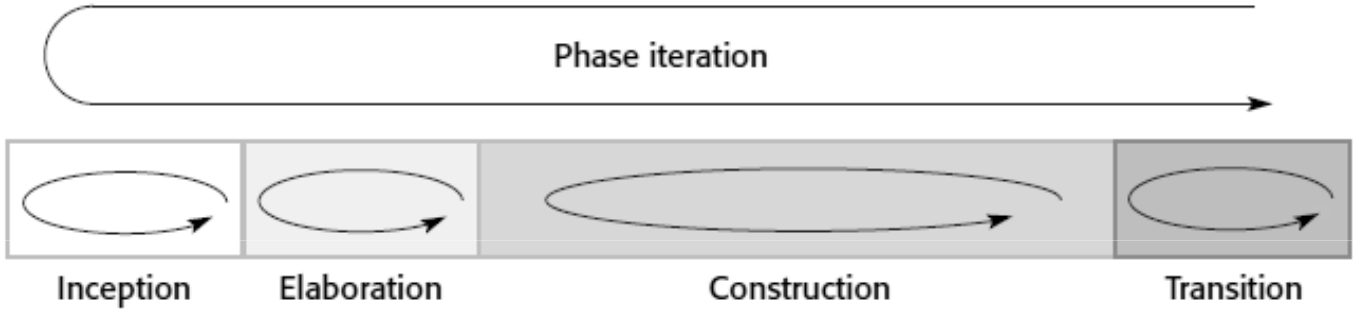
- Spiral Process:
  - Barry Boehm, 1986.
  - Each phase is a spiral:
    - Objective identification.
    - Risk evaluation.
    - Implementation.
    - Plan next phase.



Software process models

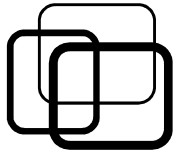


- RUP (**R**ational **U**nified **P**rocess):
  - IBM-Rational, 2003.



  - Four phases:
    - Inception: planning.
    - Elaboration: analysis & design.
    - Construction: implementation & testing.
    - Transition: maintenance.
  - Loop in phase or whole process.

Role play game



- Description:
  - Your team is assigned a “PROJECT”.
  - Each member play a ROLE.
    - Analyst.
    - Architect.
    - Developer.
    - QC Person.
  - There are 4 phases:
    - Requirement analysis.
    - Drafting.
    - Implementation.
    - Verification.