Software Lifecycle Models and Software Process

- Software lifecycle basics
- Software lifecycle models
  - build-and-fix
  - waterfall
  - rapid prototype
  - incremental and iterative
  - spiral
- Software process examples
  - XP and RUP
- Process improvement
  - CMM & ISO9000

Software Lifecycle

- A series of steps through which a software product progresses
- Lifetimes vary from days to months to years
- Consists of
  - people!
  - overall process
  - intermediate products
  - stages of the process

What is a process?

- "Device" for producing a product (get job done)
- Level of indirection
  - Process description describes wide class of instances
- Humans create process descriptions to solve classes of problems
- Thus
  - software processes are "devices" for creating and evolving software products

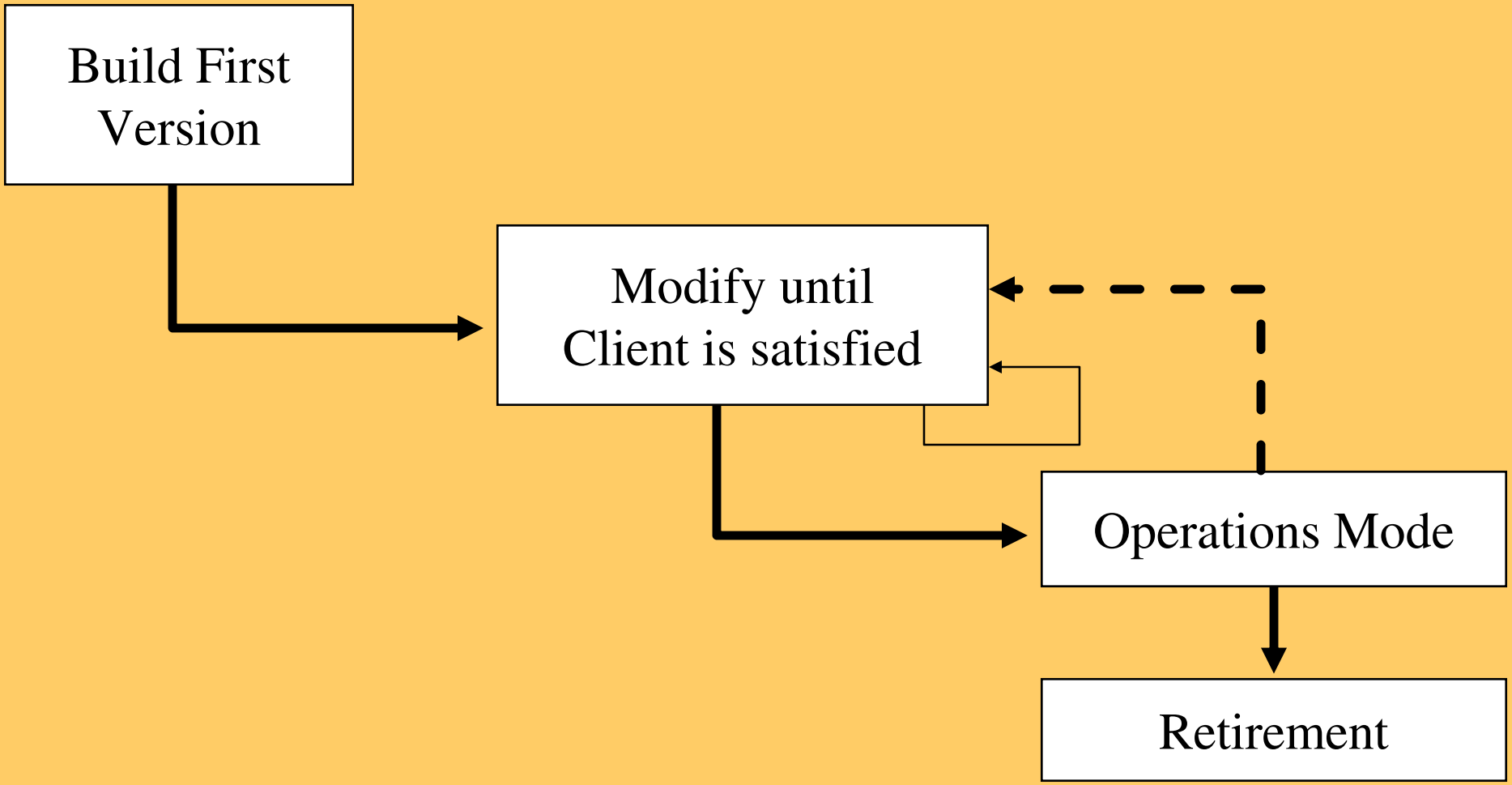
Intermediate Software Products

- Objectives
  - Mark the “end” of phases
  - Enable effective reviews
  - Specify requirements for next phase
    - Note the abstract requirements/design cycle
- Form
  - Rigorous
  - Machine processible (highly desirable)
- Content
  - Specifications, Tests, Documentation

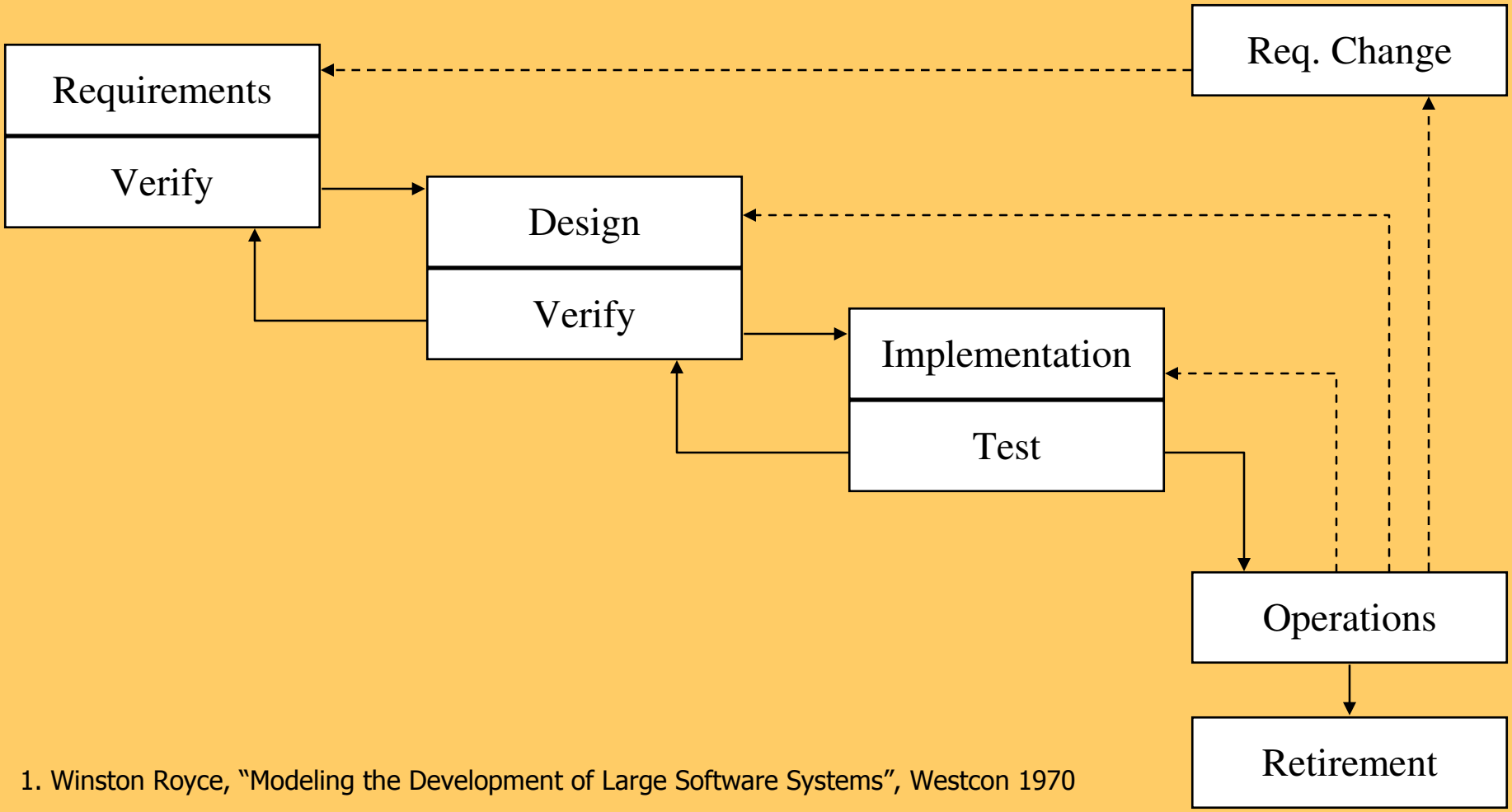
Phases of a Software Lifecycle

- Standard Phases
  - Requirements Analysis & Specification
  - Design
  - Implementation and Integration
  - Operation and Maintenance
  - Change in Requirements
  - Testing throughout
- Phases promote manageability and provide organization

Build-and-Fix



Waterfall[1]

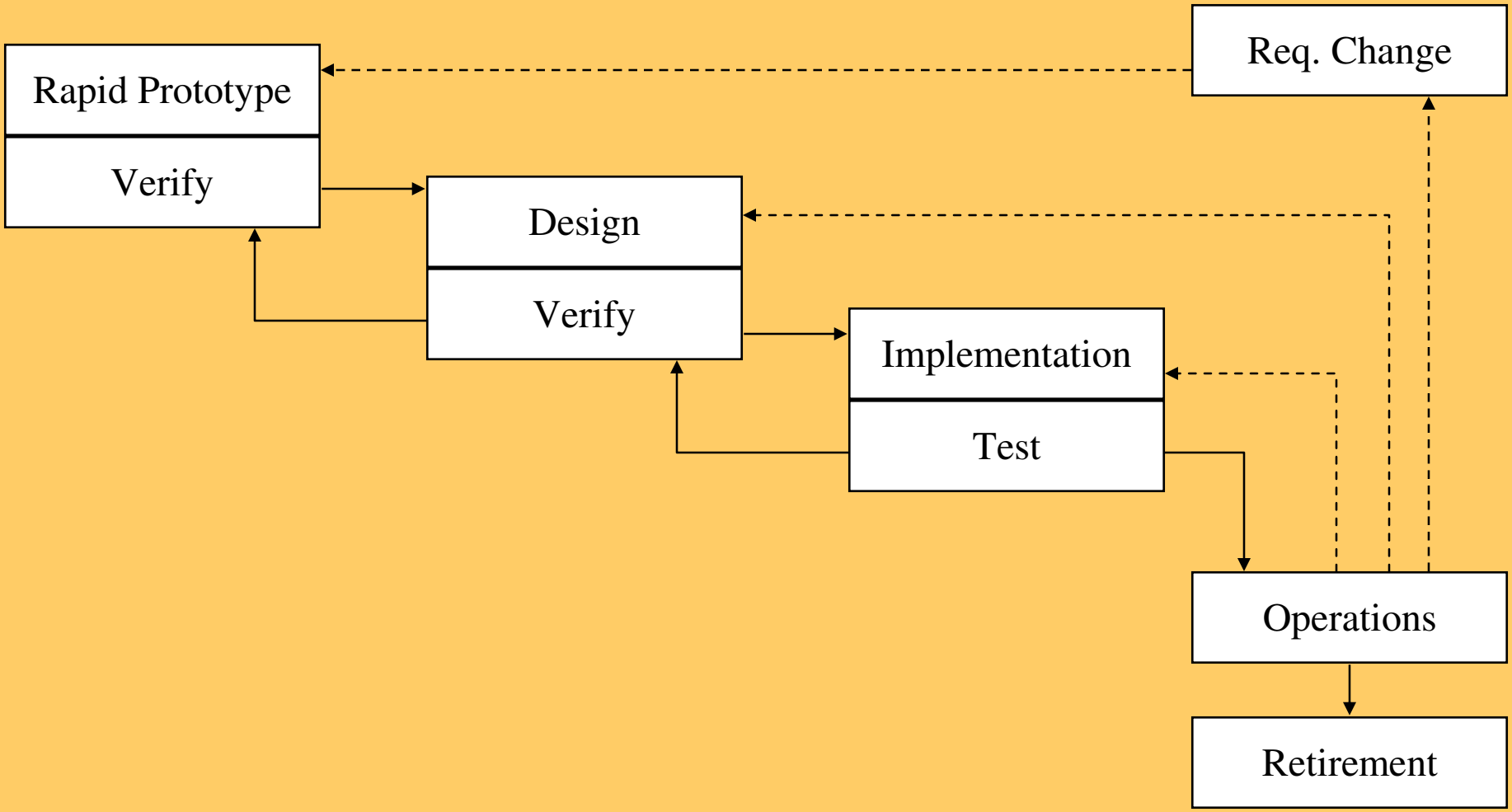


1. Winston Royce, "Modeling the Development of Large Software Systems", Westcon 1970

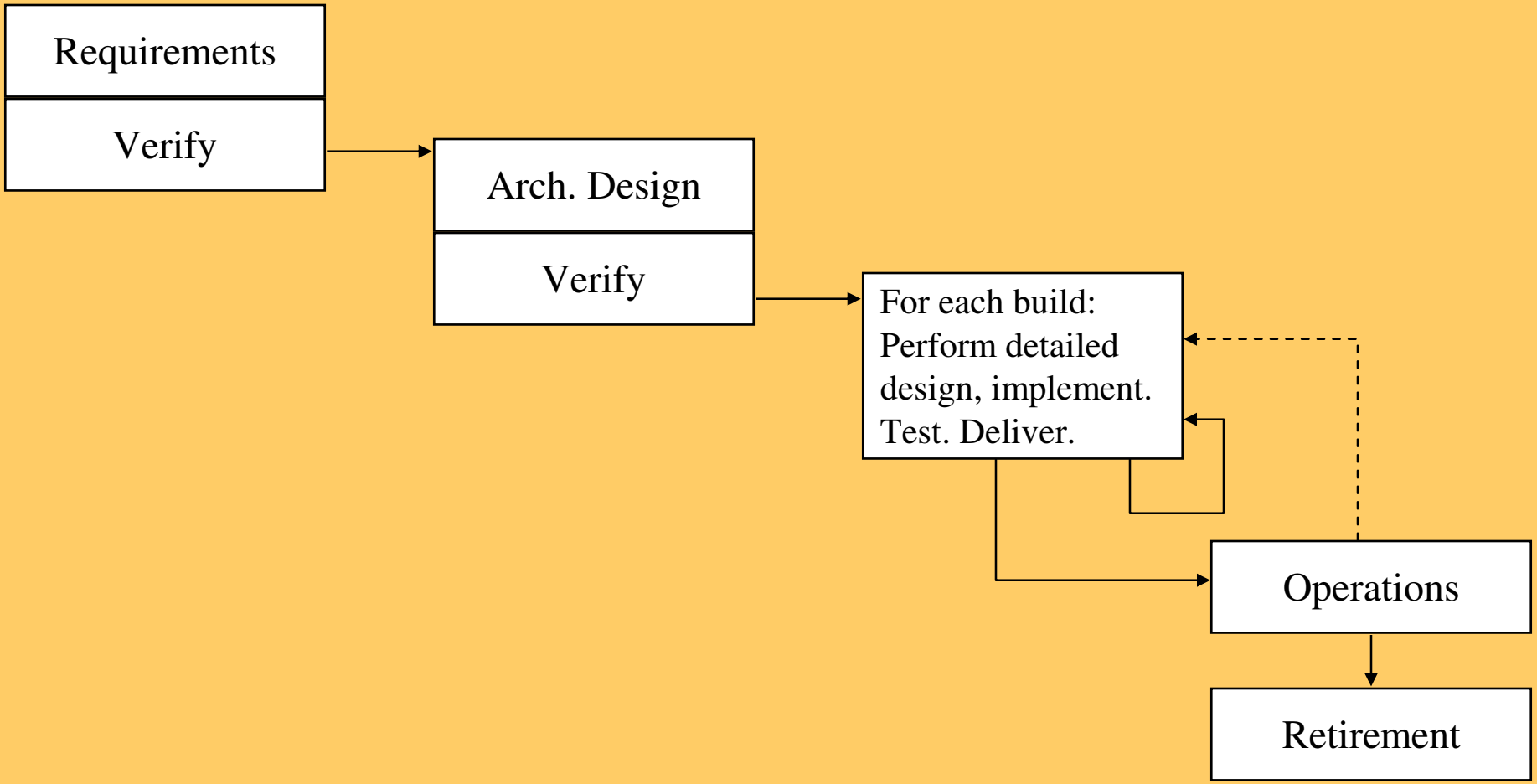
Gantt Chart

- Plan activities
- Allocate resources
- Define dependencies
- MS Project

Rapid Prototyping



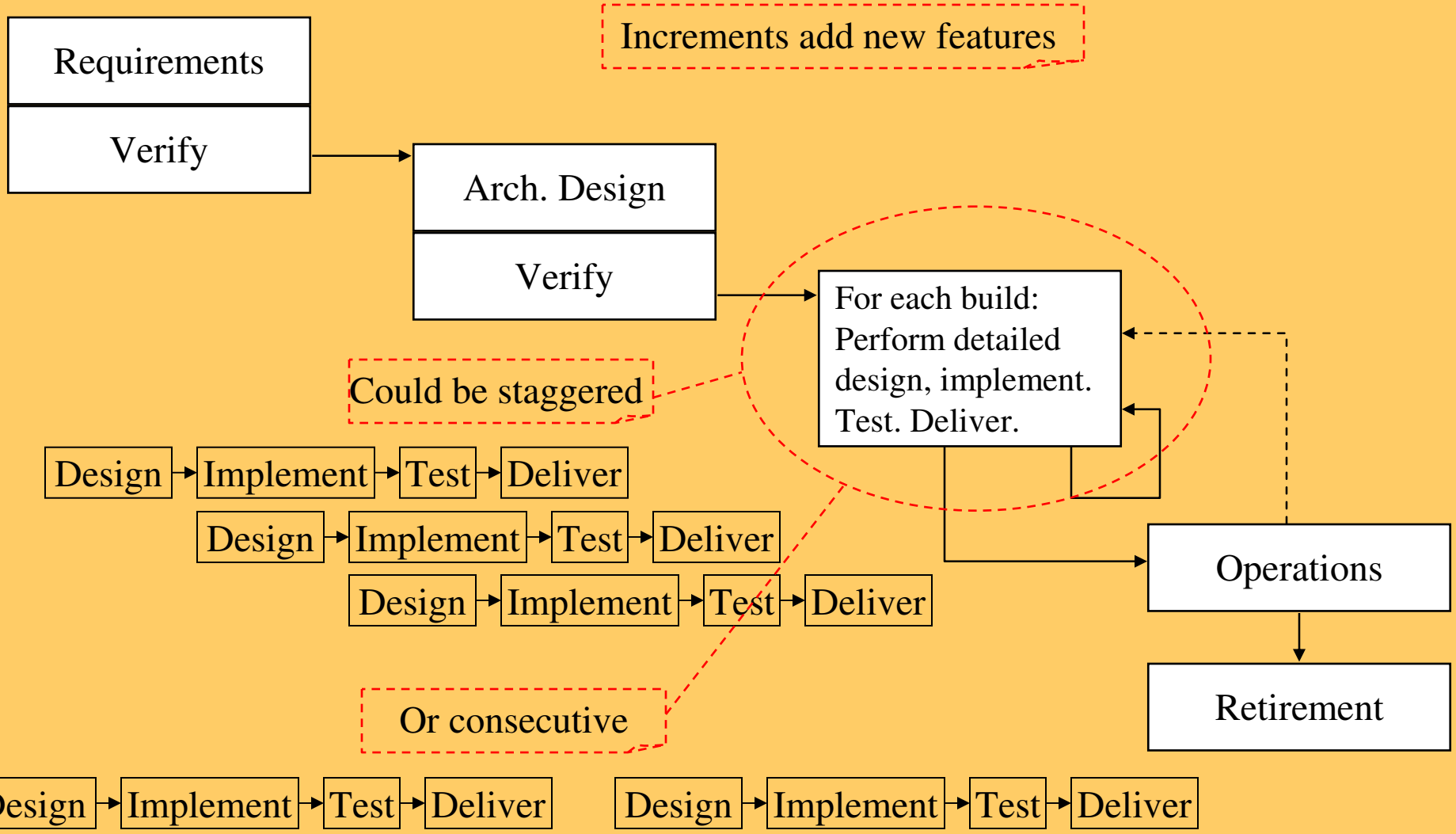
Incremental[1] and Iterative[2]



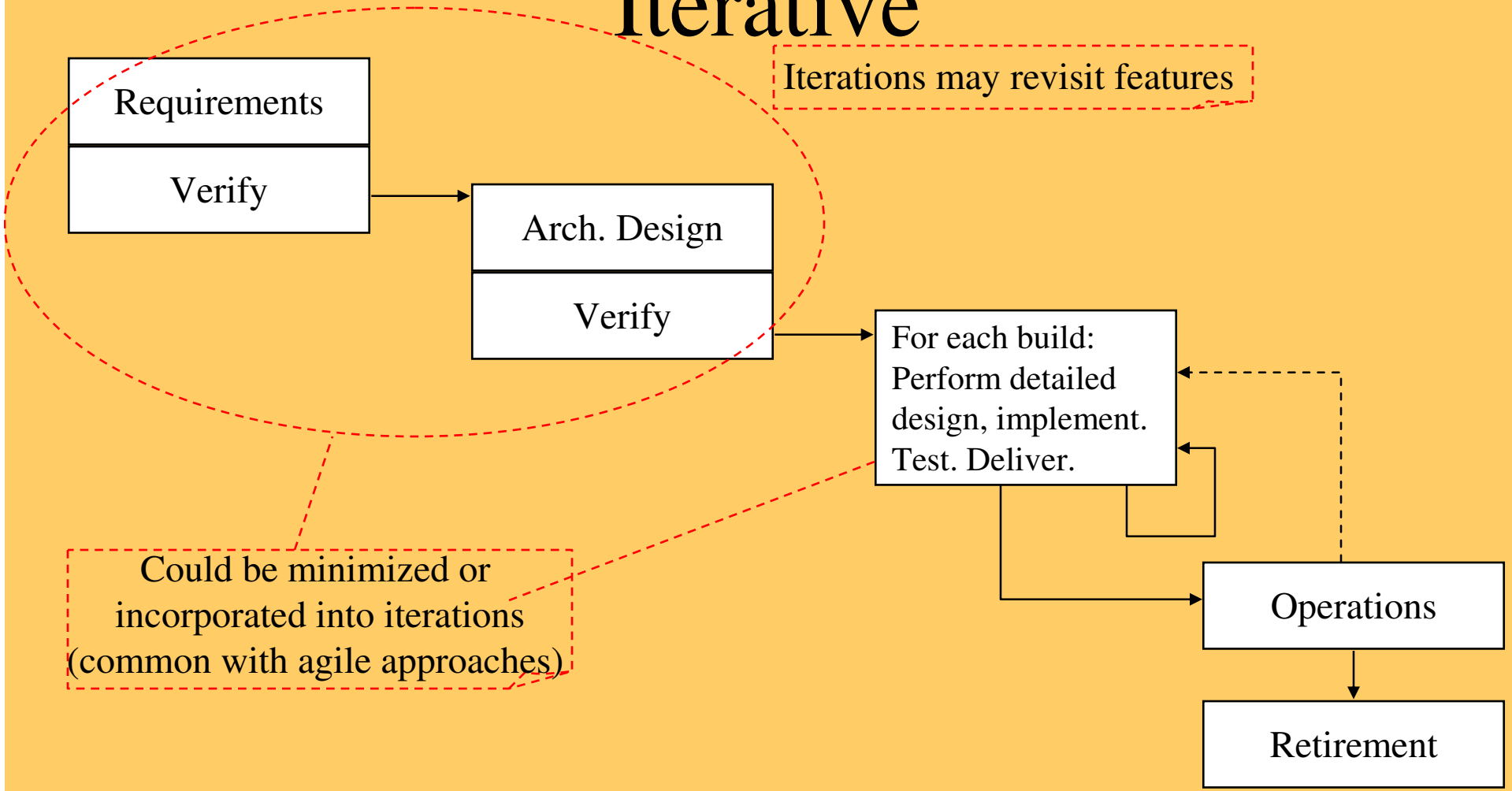
1. Harlan Mills, "Cleanroom Software Engineering", IEEE Software, 1987

2. Victor Basili and Joe Turner, "Iterative Enhancement", IEEE Trans. Software Eng., 1975

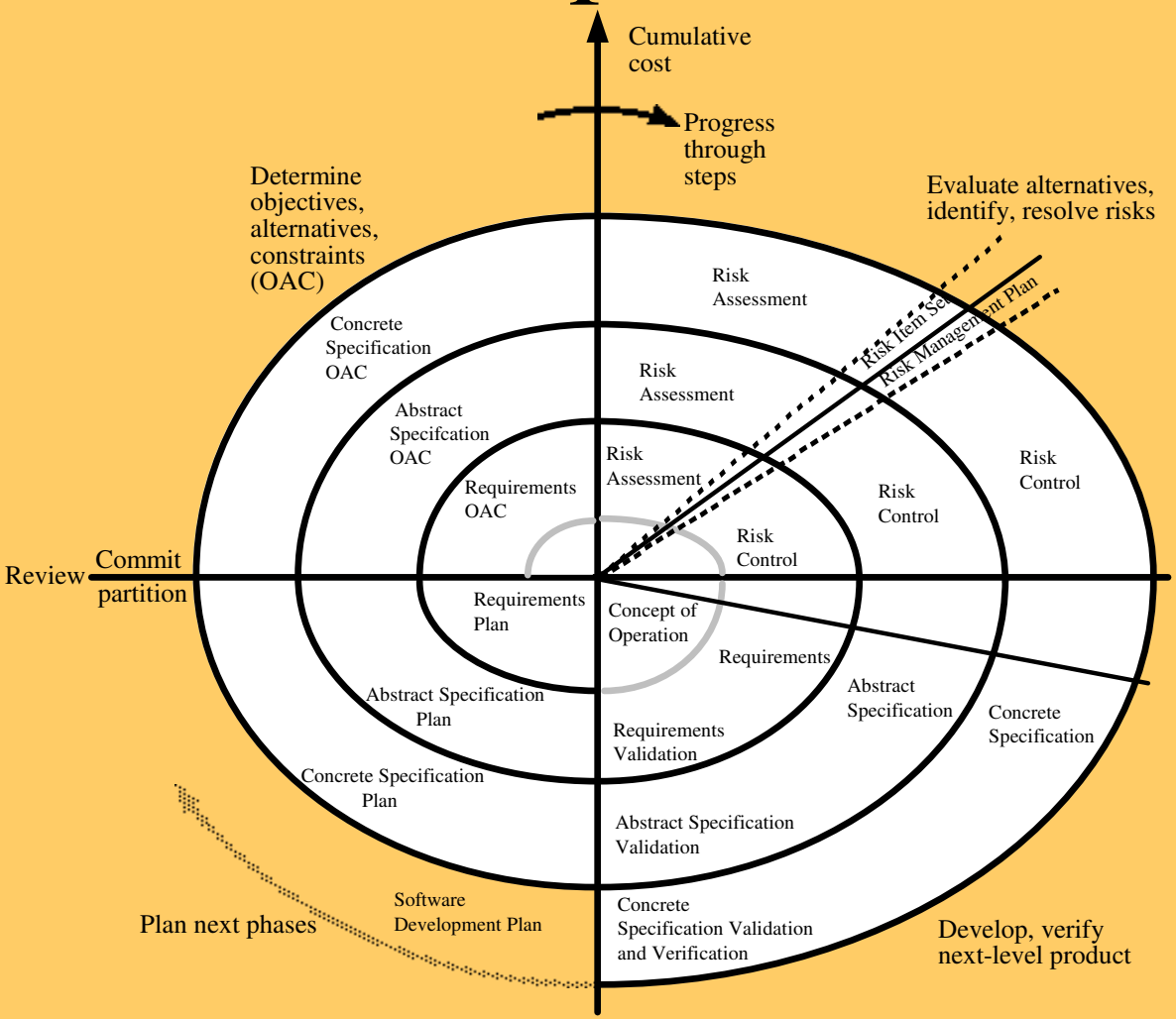
Incremental



Iterative

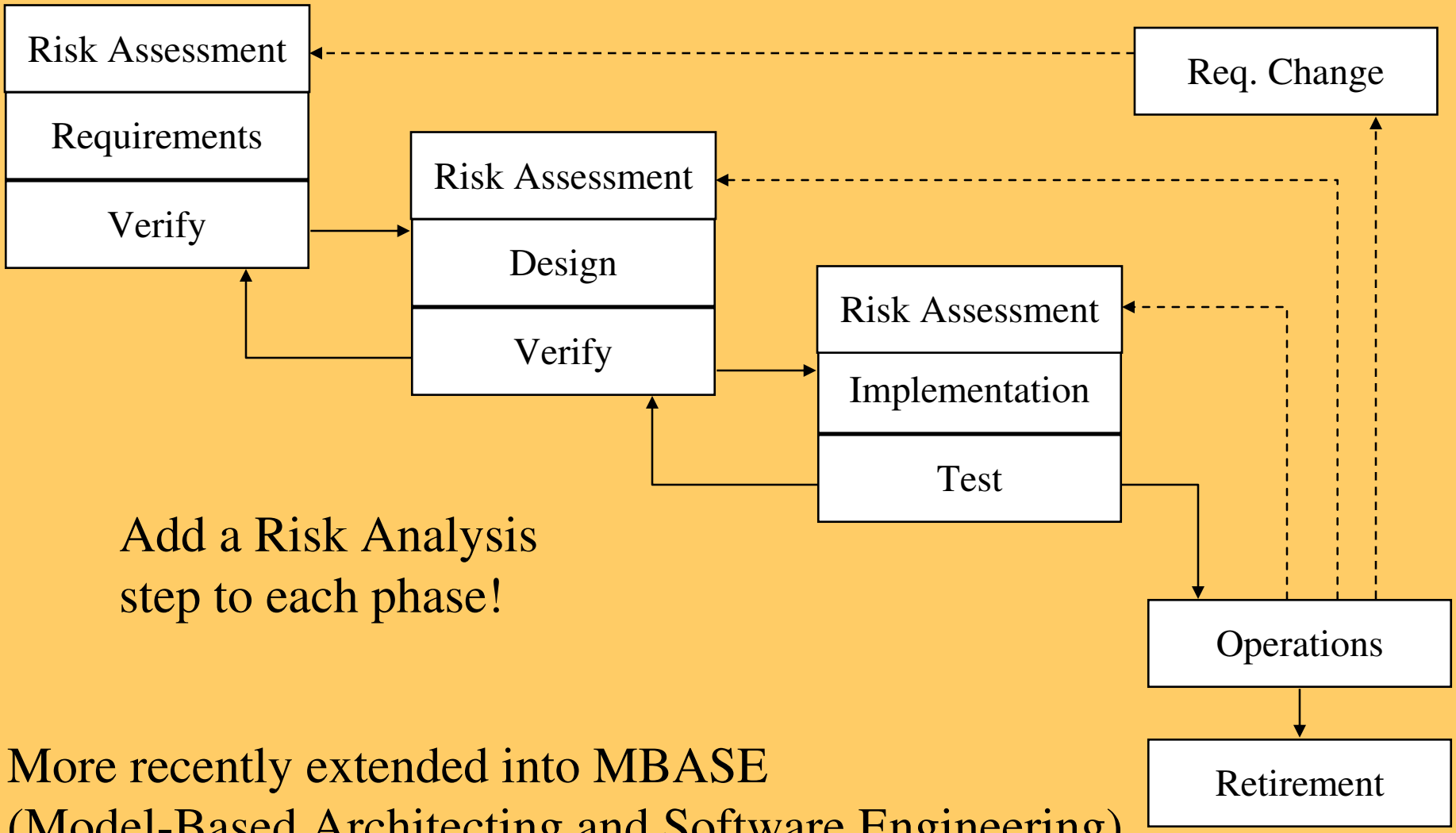

Iterations may revisit features
Requirements
Verify
Arch. Design
Verify
For each build:
Perform detailed
design, implement.
Test. Deliver.
Operations
Retirement
Could be minimized or
incorporated into iterations
(common with agile approaches)

The Spiral Model[1]



1. Barry Boehm, "A Spiral model of Software Development and Enhancement", SPSE 1985

(Extremely) Simplified Spiral Model



Add a Risk Analysis step to each phase!

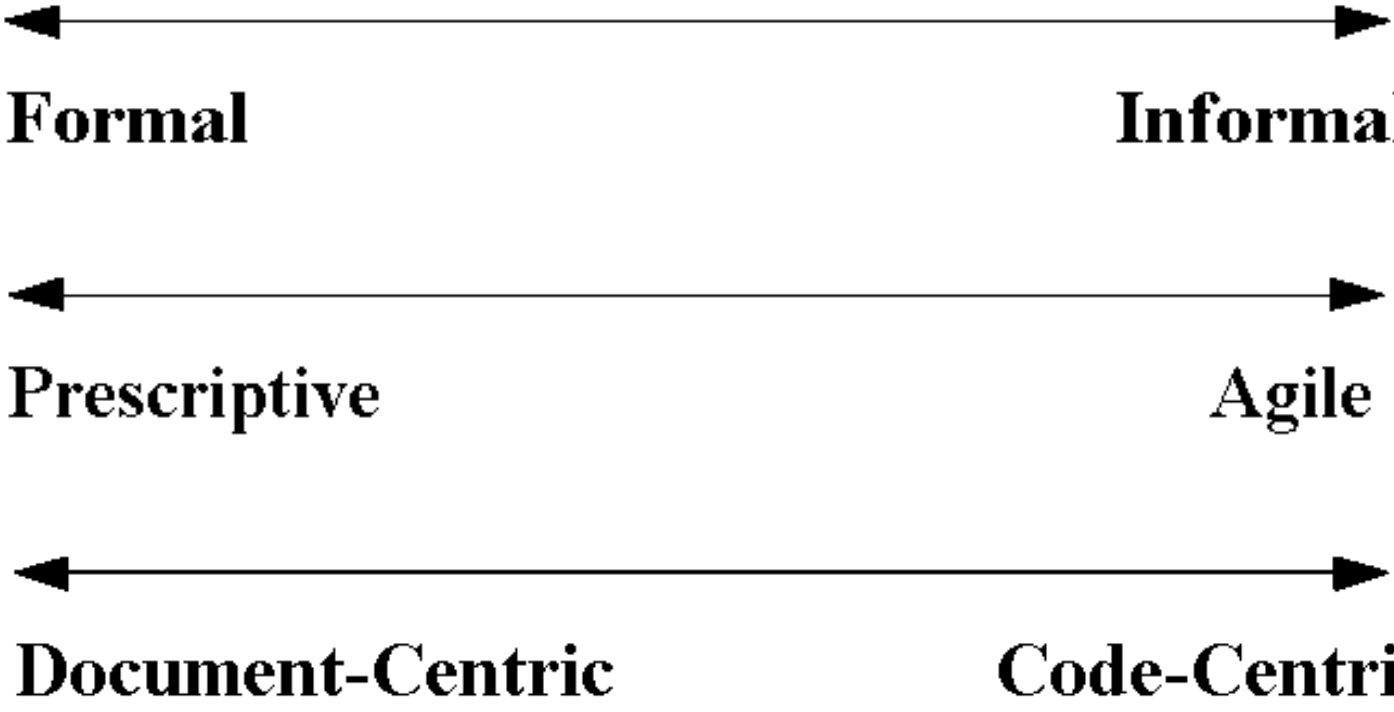
More recently extended into MBASE (Model-Based Architecting and Software Engineering)

Comparing Engineering Disciplines

- What is design? What is construction?
  - In civil and mechanical engineering, design makes up about 10% of the project
  - Design completed by creative, intelligent people
  - Construction completed by skilled but often less educated people
- Jack Reeves suggested that source code is the design document and compilers/linkers do the construction for free.
  - What are the implications of this view?

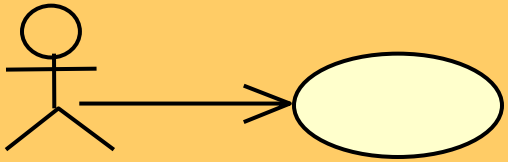
Comparing SD Processes

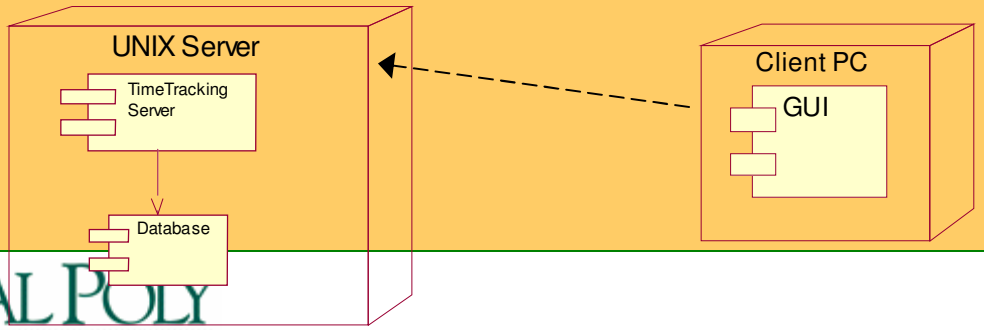
- Processes vary in complexity and control

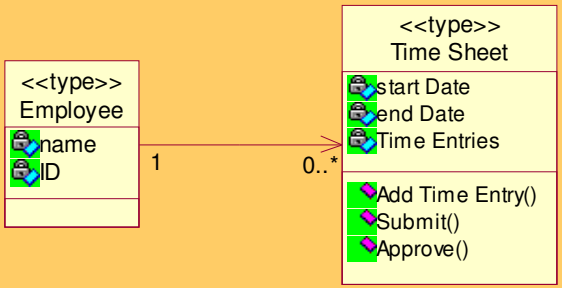


Modern Process 1

- Rational Unified Process (RUP)
  - iterative and incremental
  - use-case driven
  - component-based architectures
  - visually modeled with UML
  - quality verification
  - tools based







Rational Unified Process 5.5 - Netscape

File Edit View Go Communicator Help

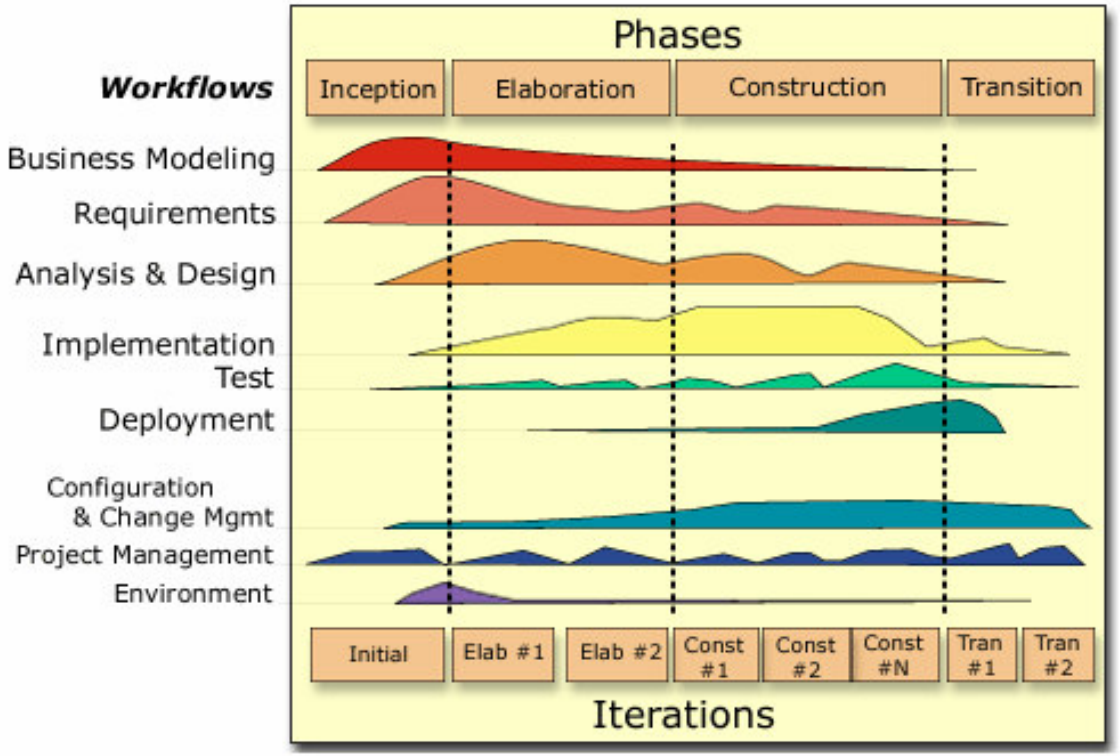
Back Forward Reload Home Search Netscape Print Security Stop

Bookmarks Location: file:///C|/Program Files/Rational/RationalUnifiedProcess5.5/index.htm What's Related

Instant Message WebMail Contact People Yellow Pages Download Find Sites Channels

Index Search Where am I Getting Started Glossary

- Overview
  - Site Map
  - Roadmaps
  - Navigation Tools
  - Key Concepts
  - Glossary
  - References
  - Best Practices
  - What's New
- Core Workflows
- Iteration Workflows
- Workers and Activities
- Artifacts
- Tool Mentors
- Templates
- White Papers
- Work Guidelines
- Resource Center
- About the Unified Process



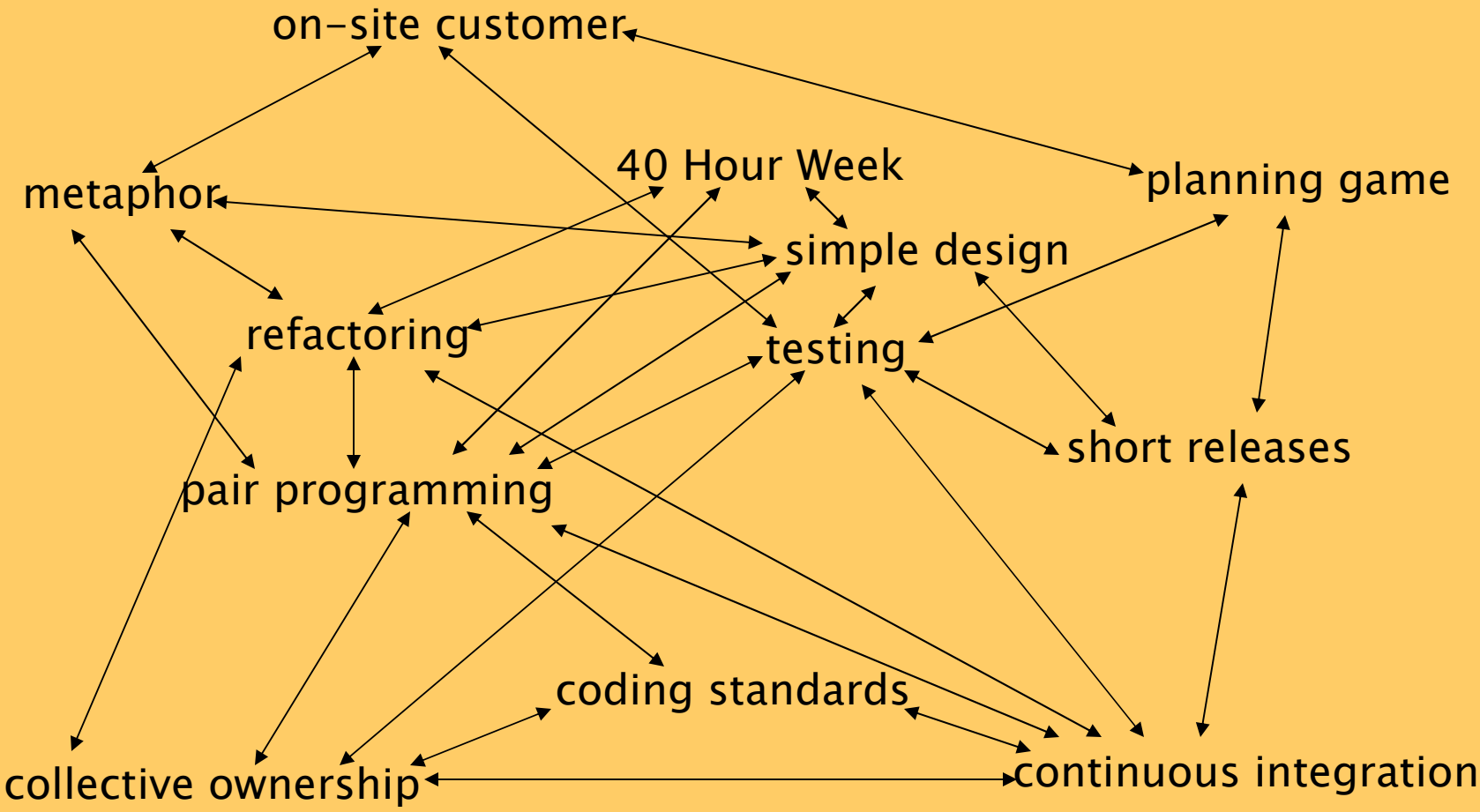
Click on an element for more information.

Start Inbox - Micro... Exploring - R... Rational ... Microsoft Po... Getting Start... 3:34 PM

Modern Process 2

- eXtreme Programming (XP)
  - lightweight process
  - frequent iterations
  - Best practices "in the extreme"
    - continuous integration
    - pair programming
    - test-driven development
    - refactoring
  - popular agile method

XP Practice Coupling[1]



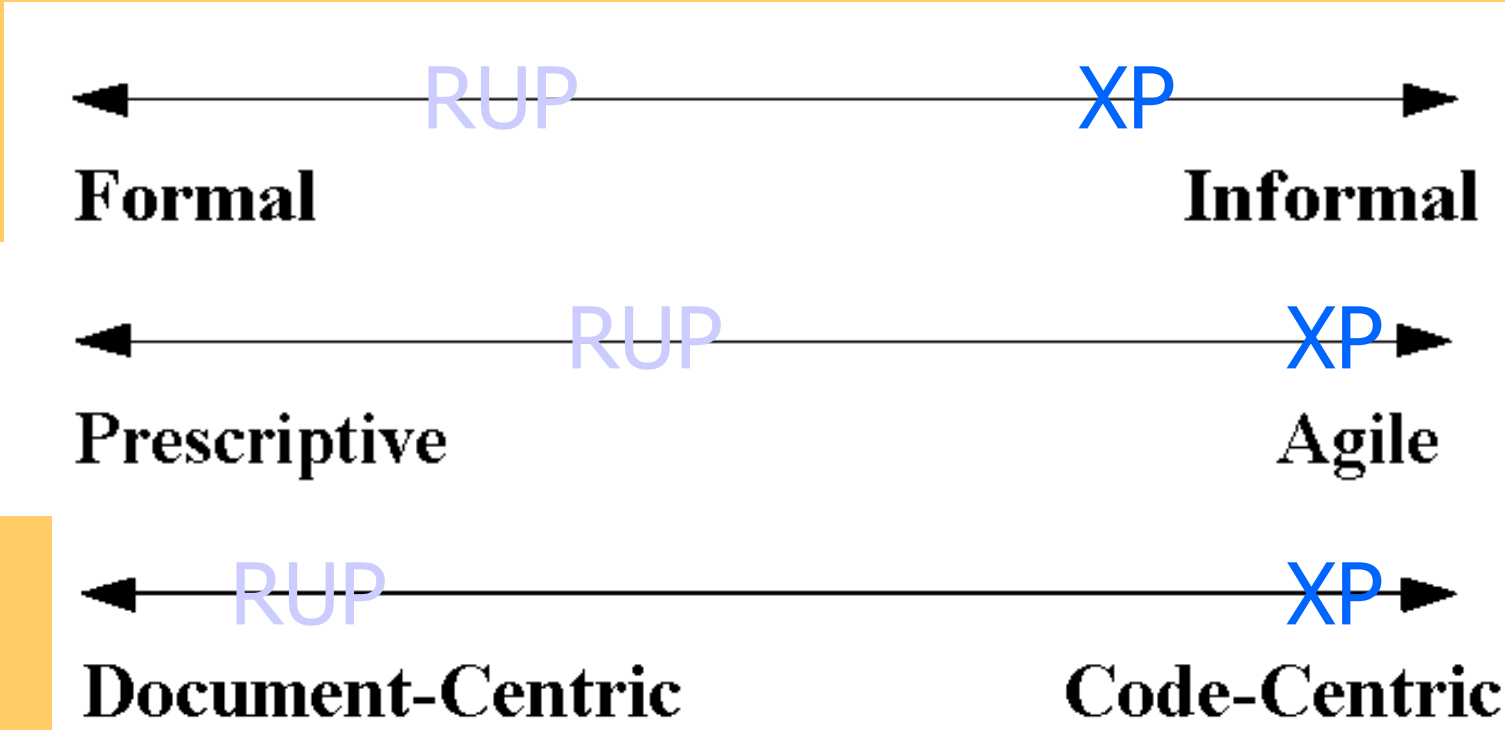
1. Beck, Extreme Programming Explained: Embrace Change, 2000

Refactoring

- Improving the structure of code without changing its behavior
- Removing code smells

Comparing Modern Processes

- RUP vs. XP



Capability Maturity Model (CMM)

- CMM is not a software lifecycle model ...
  - Strategy for improving the software development process regardless of the process "model" followed
    - Basic premise: the use of new software methods alone will not improve productivity and quality, because software management is, in part, the cause of problems
  - CMM assists organizations in providing the infrastructure required for achieving a disciplined and mature process ($$)
- Includes
  - technical aspects of software production
  - managerial aspects of software production

Capability Maturity Model (continued)

- Five maturity levels
  - 1. initial – ad hoc process
  - 2. repeatable process – basic project management
  - 3. defined process – process modeling and definition
  - 4. managed process – process measurement
  - 5. optimizing process – process control and dynamic improvement
- to move from one stage to the next, the SEI provides a series of questionnaires and conducts process assessments that highlight current shortcomings

ISO 9000

- Further attempt to improve software quality based on International Standards Organization (ISO)
- ISO 9000 = series of five related standards
  - within ISO 9000 standard series ISO 9000-3 focuses on software and software development
- Basic features:
  - stress on documenting the process in both words and pictures
  - requires management commitment to quality
  - requires intensive training of workers
  - emphasizes measurement

ISO 9000

- Adopted by over 60 countries (USA, Japan, European Union, ...)
- To be ISO 9000 compliant, a company's process must be certified

Process Measures

- Quality (e.g. faults per KLOC)
- Functionality (e.g. function points)
- Productivity (e.g. function points per hour)
- Maintainability (e.g. effort to make change)
- Customer satisfaction (e.g. survey results)

Product Measures

- External Quality (e.g. faults)
- Internal Quality
  - Coupling, cohesion, complexity
- Size (e.g. LOC, #classes, #methods)
- Testability (e.g. tests per method)
- Scalability (e.g. users supported)
- Performance (e.g. transactions per second)