CS 4530 & CS 5500 Software Engineering

Lecture 10.1: Software Processes and Continuous Development

Jonathan Bell, John Boyland, Mitch Wand
Khoury College of Computer Sciences

Learning Objectives for this Lesson

By the end of this lesson, you should be able to…

- Relate continuous development to classic software process models (waterfall and agile)
- Identify the stages of a continuous development pipeline and describe how they relate to improving code velocity and quality

What is a software process?

- A structured set of activities required to develop a software product
    - Specification
    - Design and implementation
    - Validation
    - Evolution (operation and maintenance)
- Goal: Minimize Risk
    - Falling behind schedule
    - Changes to requirements
    - Bugs/unintended effects of changes

Software Verification and Validation

Quality Assurance

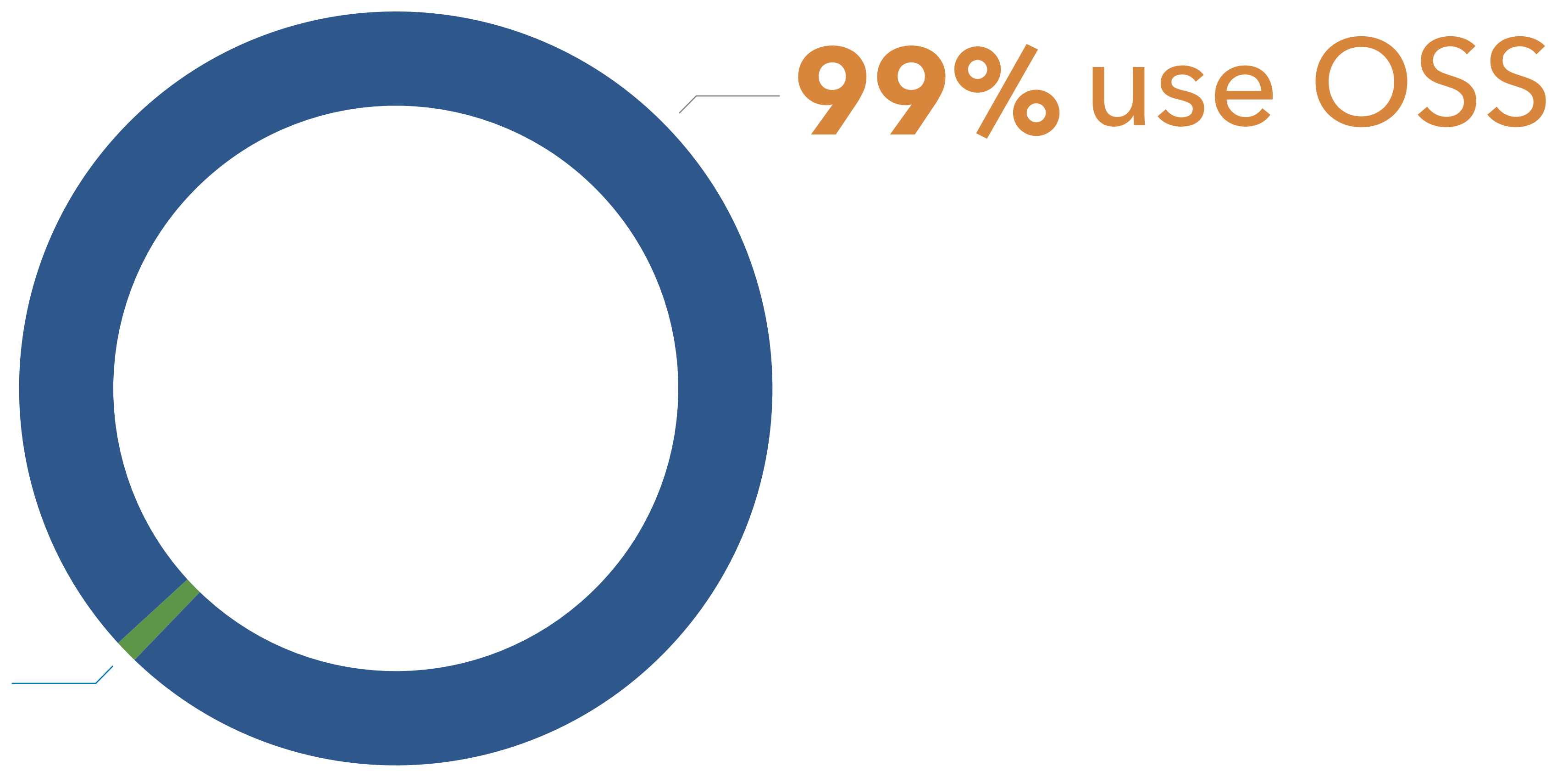
- Verification and validation (V & V) is intended to show that a system conforms to its specification and meets the requirements of the customer(s).
- Involves checking and review processes, and acceptance or beta testing.
- Custom software: Acceptance testing involves executing the system with test cases that are derived from the real data to be processed by the system in the customer's environment.
- Generic software: Beta testing executes the system in many customers' environments under real use.

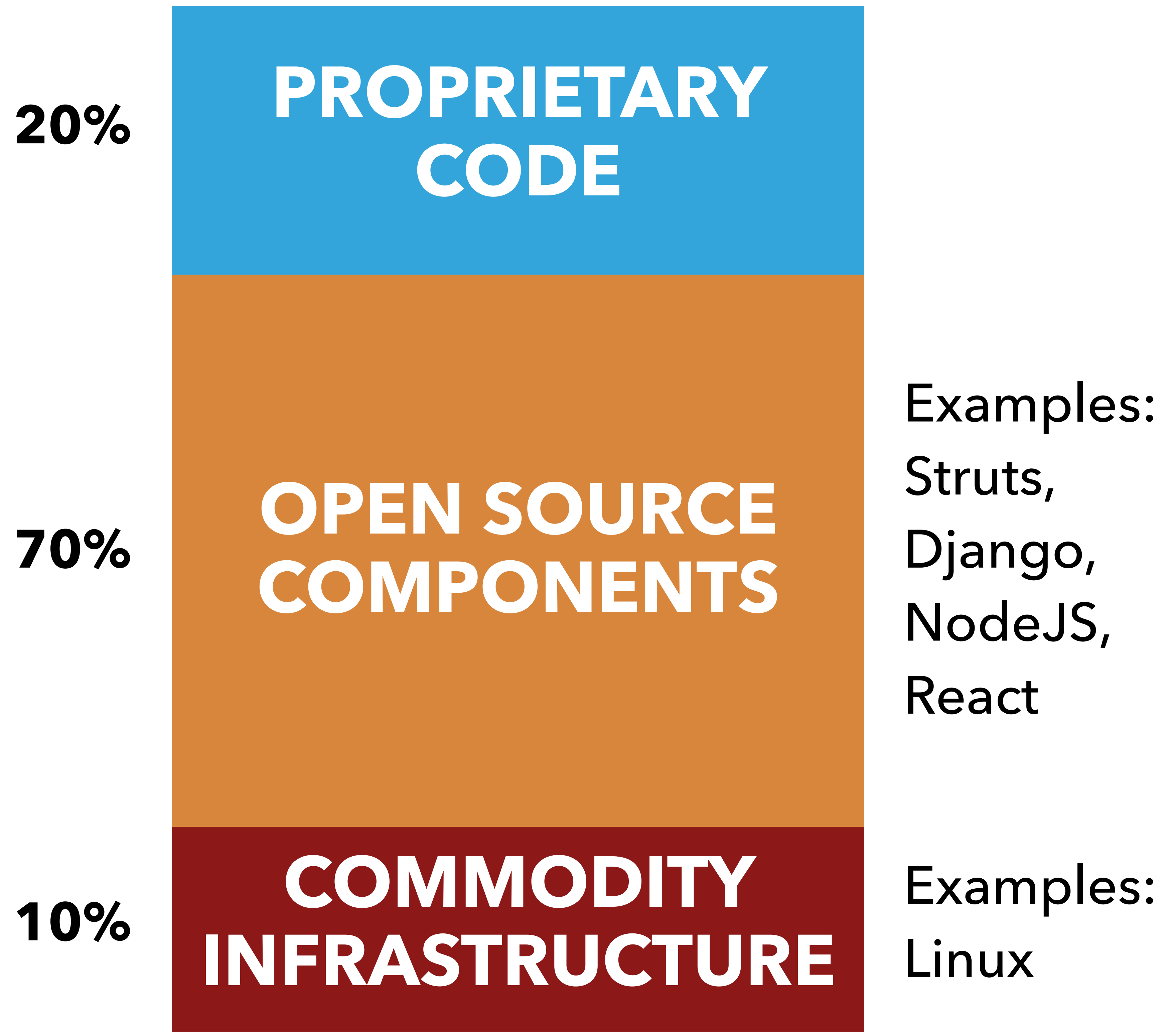
Software Evolution

Software is inherently flexible: we want high development velocity!

- As requirements change due to changing business circumstances, the software that supports the business must also evolve and change.
- Most software today is built on large (and old) codebases

Synopsys (BlackDuck) 2019 audit of 1,200 proprietary codebases:





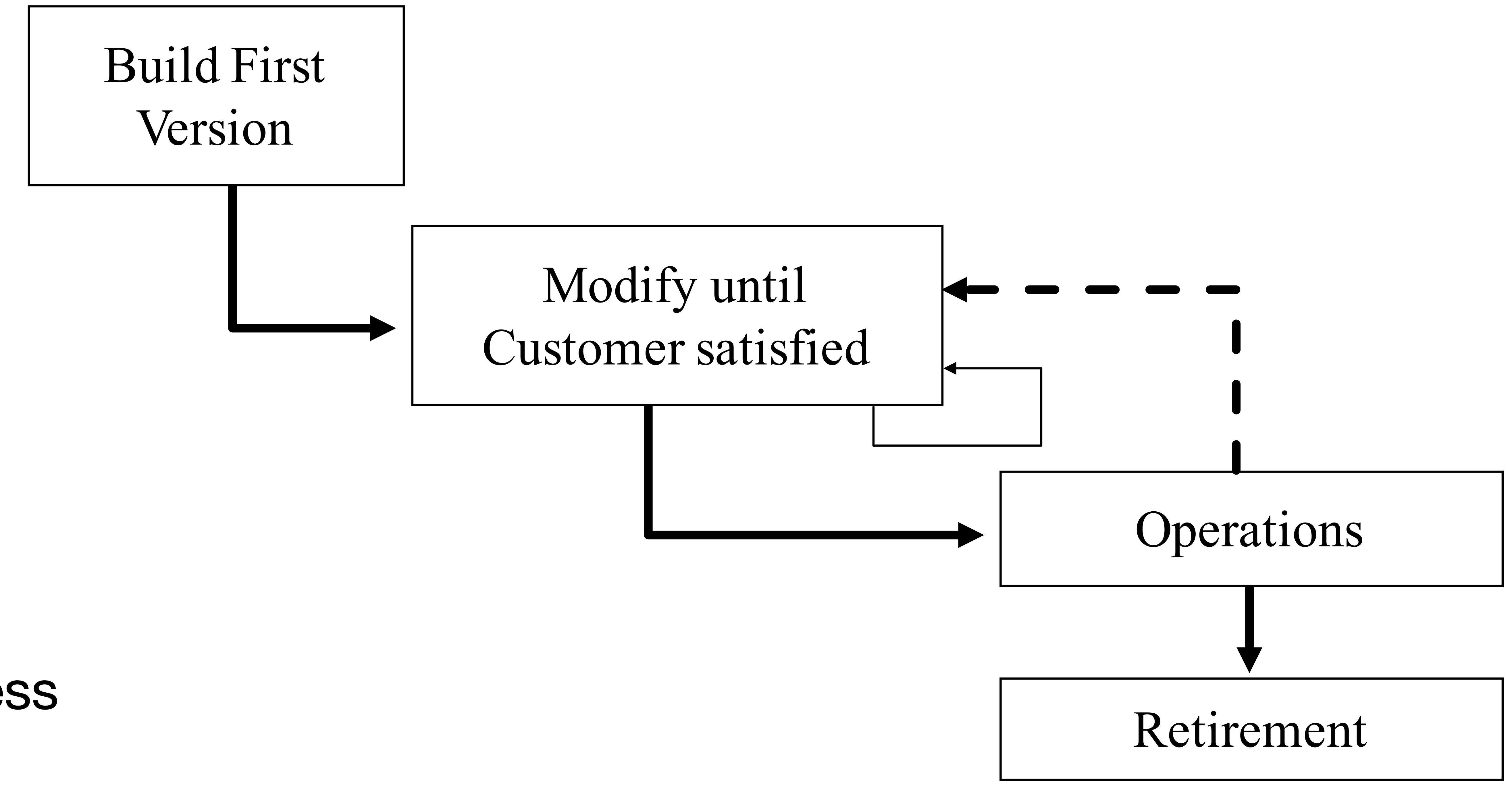
Process Models

- If we say that building software requires:
  - Specification
  - Design/Implementation
  - Validation
  - Evolution
- How do we structure our organization/development teams/tasks to do this most efficiently?

Software Processes

Code-and-fix

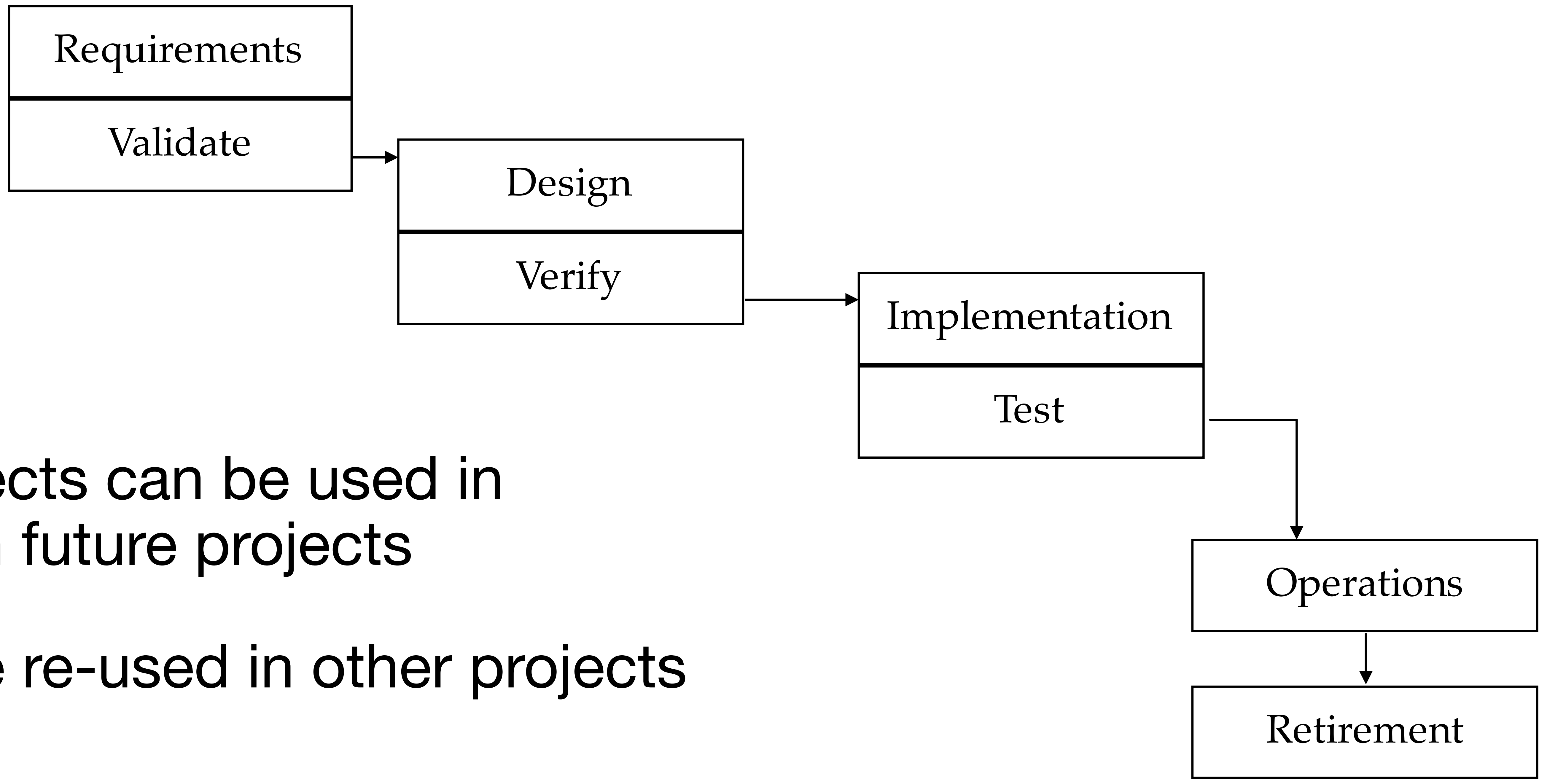
- Really Bad
- Really Common
- Advantages
  - No Overhead
  - No Expertise
- Disadvantages
  - No means of assessing progress
  - Difficult to coordinate multiple programmers
- Useful for “hacking” single-use/personal-use programs: start with empty program and debug until it works



Software Processes

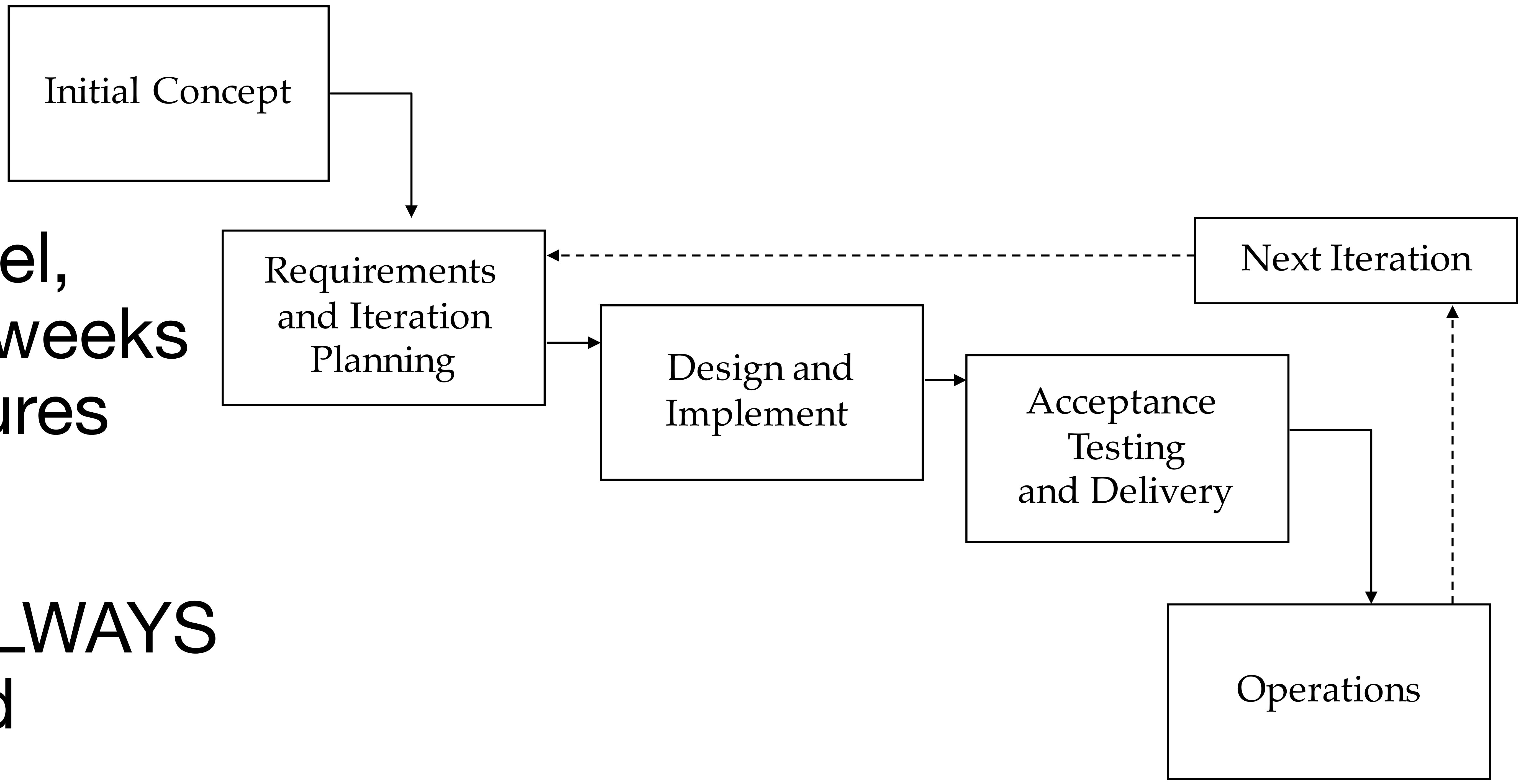
Waterfall Model

- Widely used today
- Advantages
  - Measurable progress
  - Experience applying steps in past projects can be used in estimating duration of “similar” steps in future projects
  - Produces software artifacts that can be re-used in other projects
- Disadvantages
  - Difficulty of accommodating change after the process is underway: One phase has to be complete before moving onto the next phase.



Software Processes

Agile Model

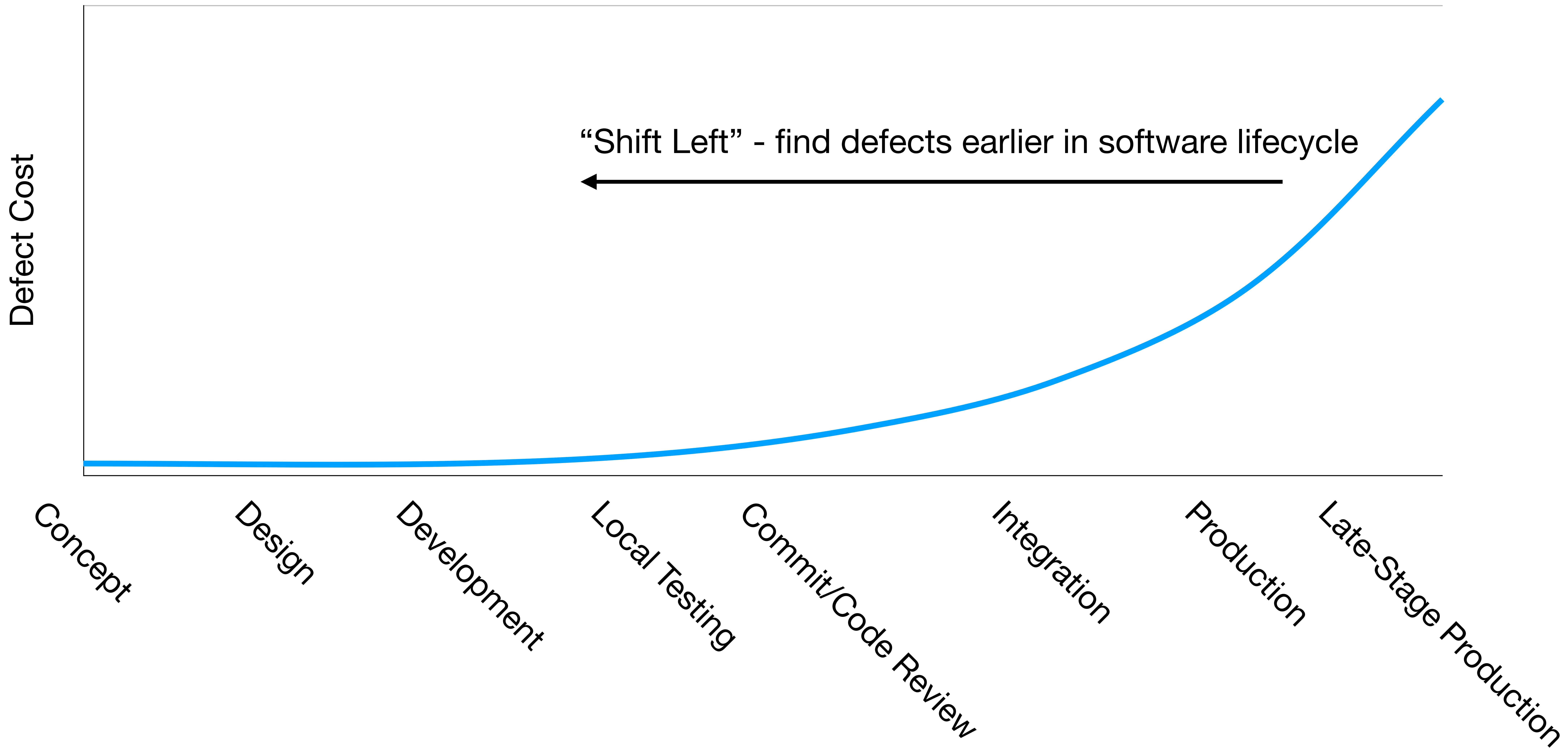


Initial Concept
Requirements and Iteration Planning
Design and Implement
Acceptance Testing and Delivery
Operations
Next Iteration

- Agile results in an *iterative* model, where each iteration is several weeks long and results in several features being built
- Recognize that requirements ALWAYS evolve as you are trying to build something
- Plus, maybe you can get useful feedback by delivering a partial app early

Cost to Fix a Defect Over Time

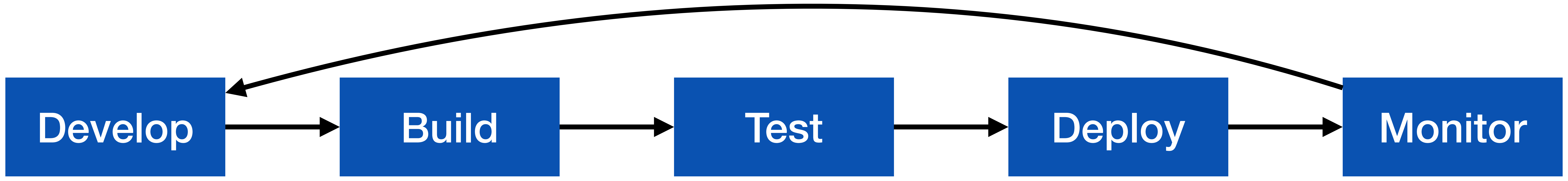
Rough Estimate



Software Processes

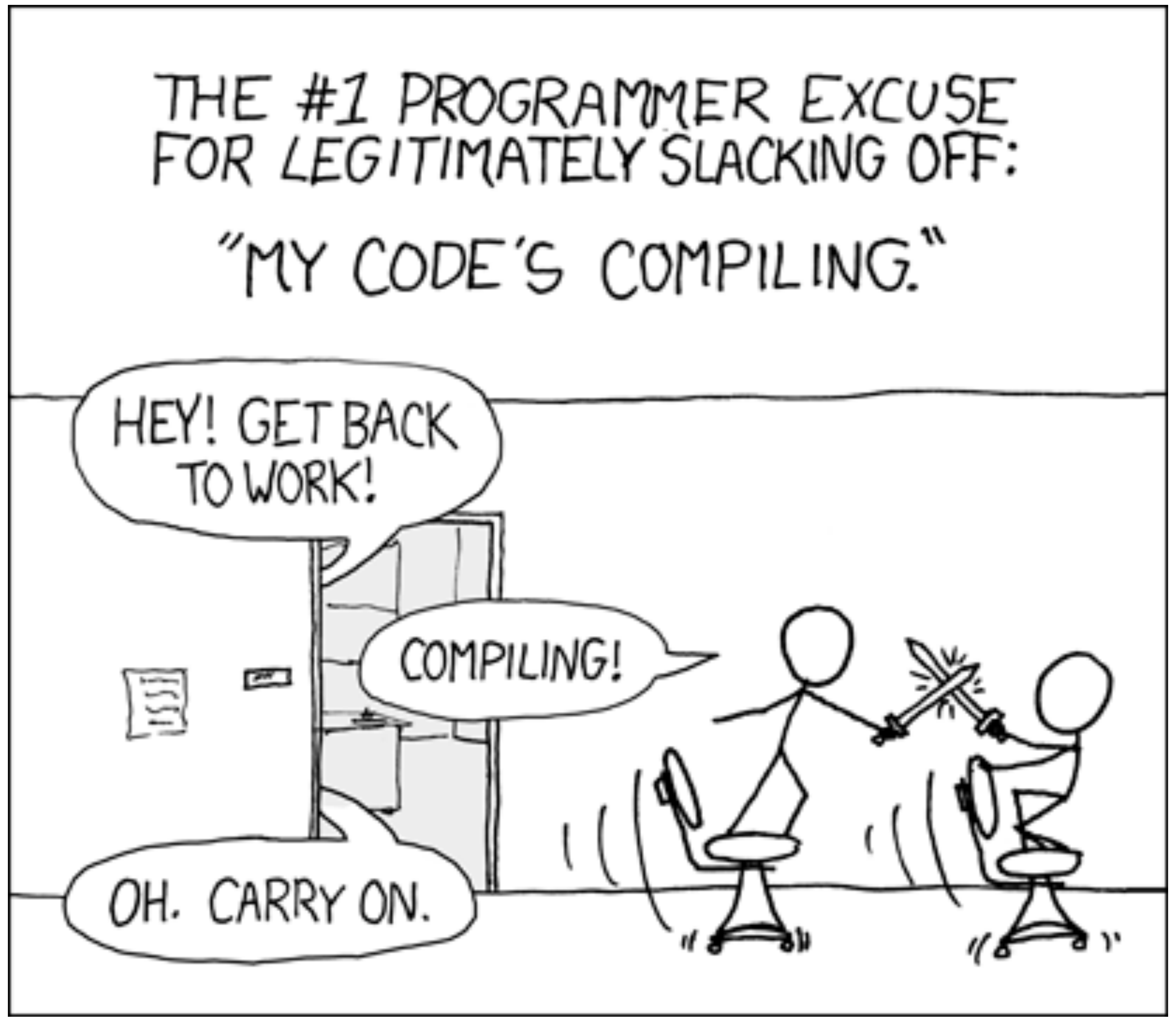
Continuous Development

- Like agile, but…
  - Fast feedback loops
  - We have a formal mechanism for deploying new versions of code and validating (test/staging/production)



Why Continuous Development?

Unblocking developers and increasing velocity



Why Continuous Development?

Improving the end-user experience

If you have:

1
5
10
100
1,000

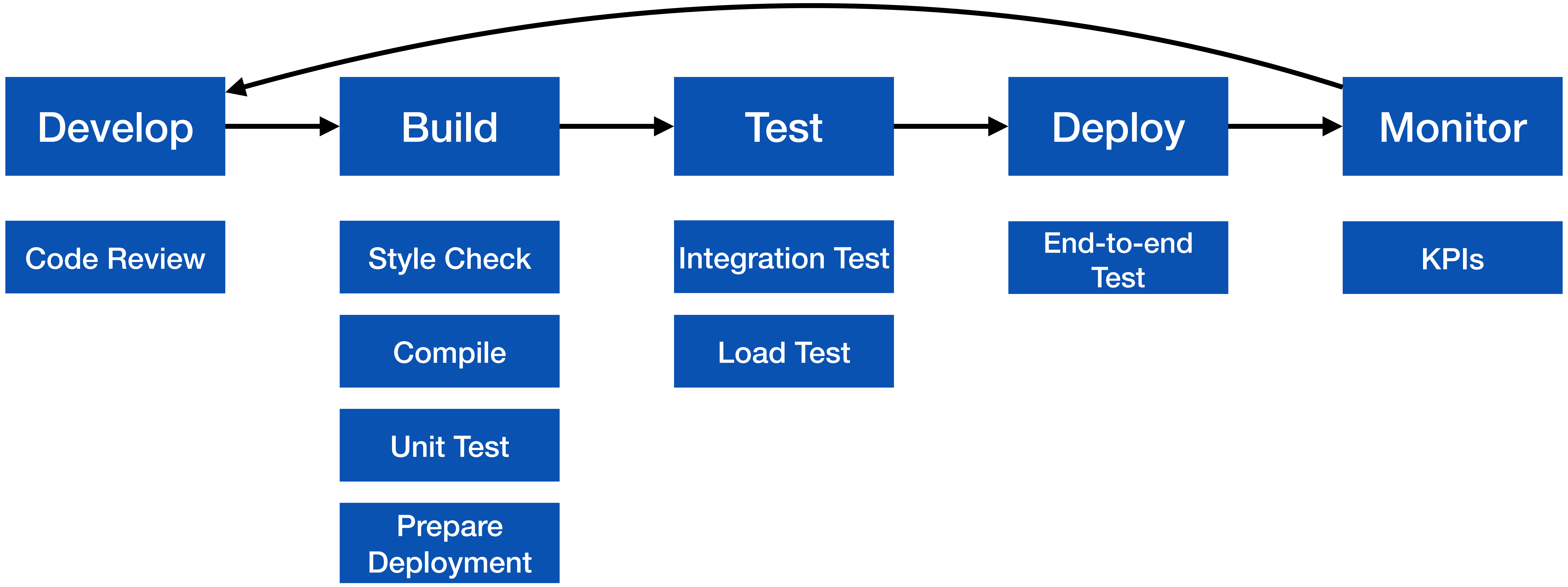
developers working on your product

How often can you **deliver** your customers:

- Bug fixes
- Security patches
- Feature enhancements
- New features

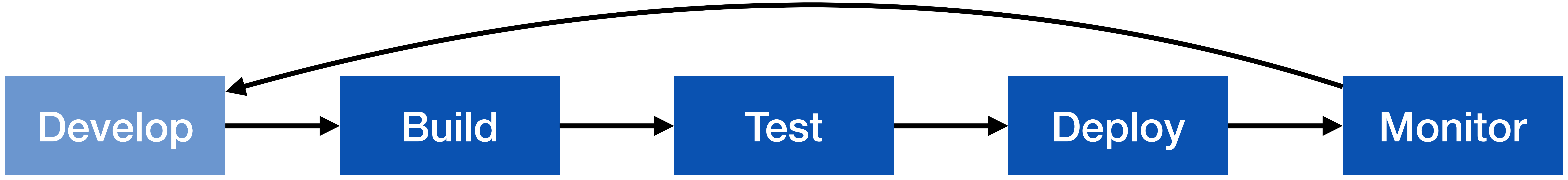
Continuous Development

Improving quality & velocity with frequent, fast feedback loops



Roadmap for this week

- Continuous development overview (this lesson)
- “Shifting left” with continuous integration
- Deployment infrastructure
- Continuous delivery



This work is licensed under a Creative Commons Attribution-ShareAlike license