Applying UML & Patterns (3rd ed.)

Chapter 2

Iterative Processes

Dr. Glenn L. Ray

School of Information Sciences

University of Pittsburgh

gray@sis.pitt.edu 412-624-9470

Fig. 2.1

Iterative SW Development Process

Each iteration is a mini-waterfall process

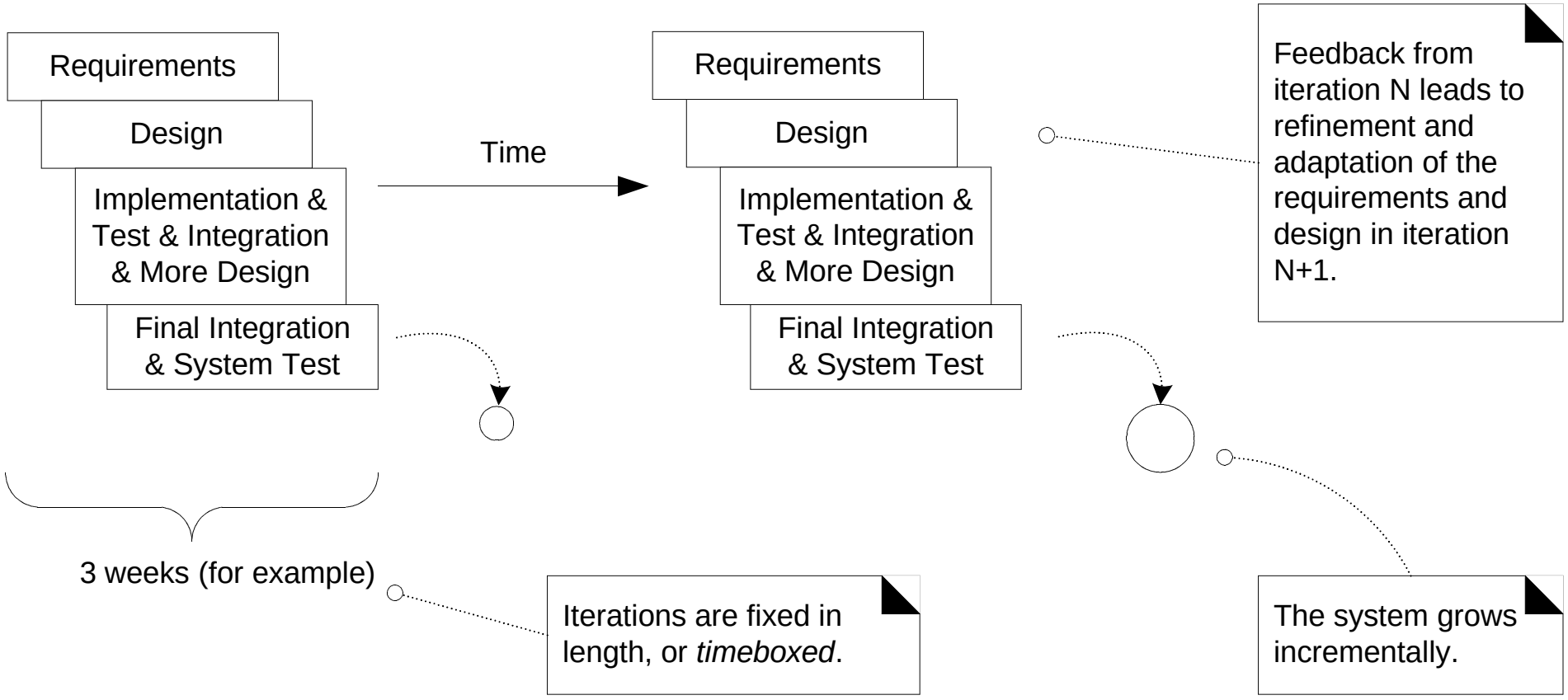


Fig. 2.2

Iterative development entails the refactoring of previously written code in subsequent iterations. Requires good architecture, automated tools and library of unit tests to be successful

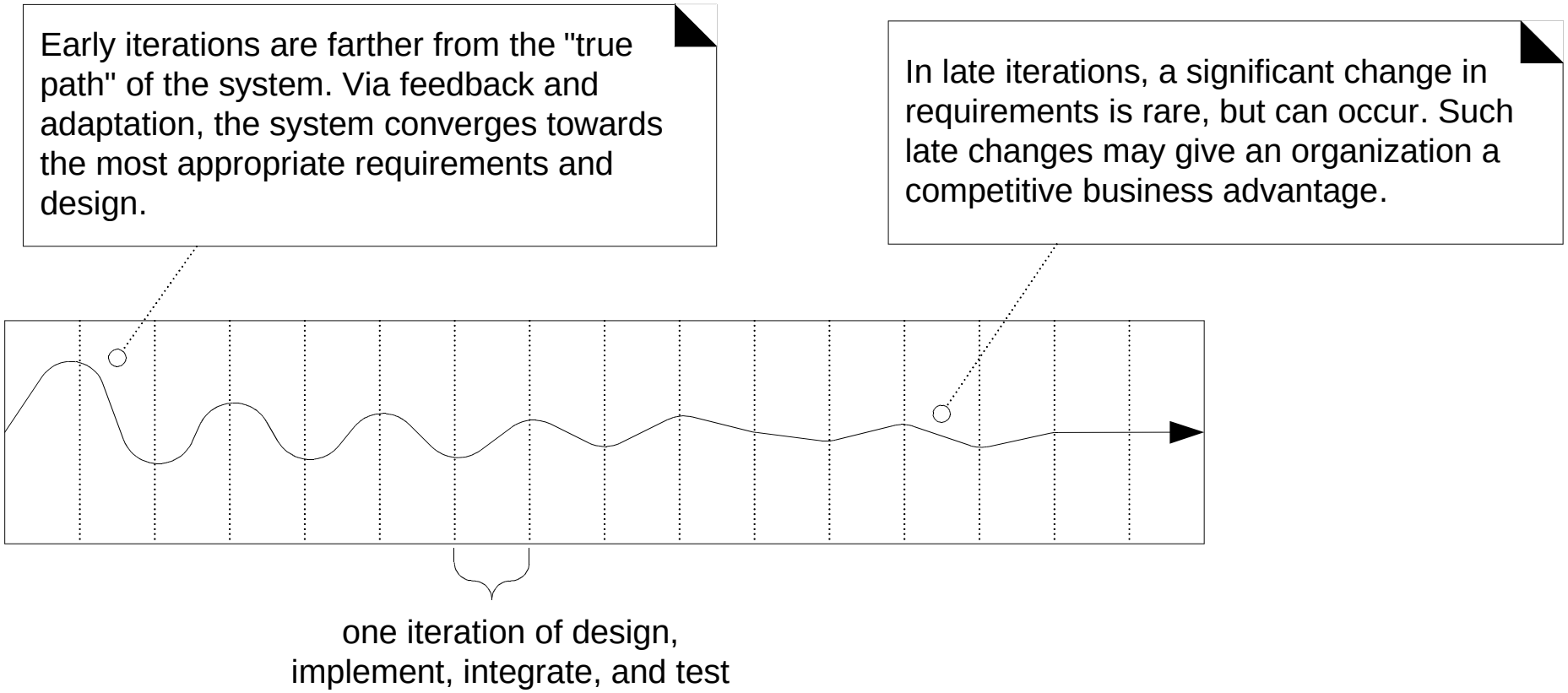
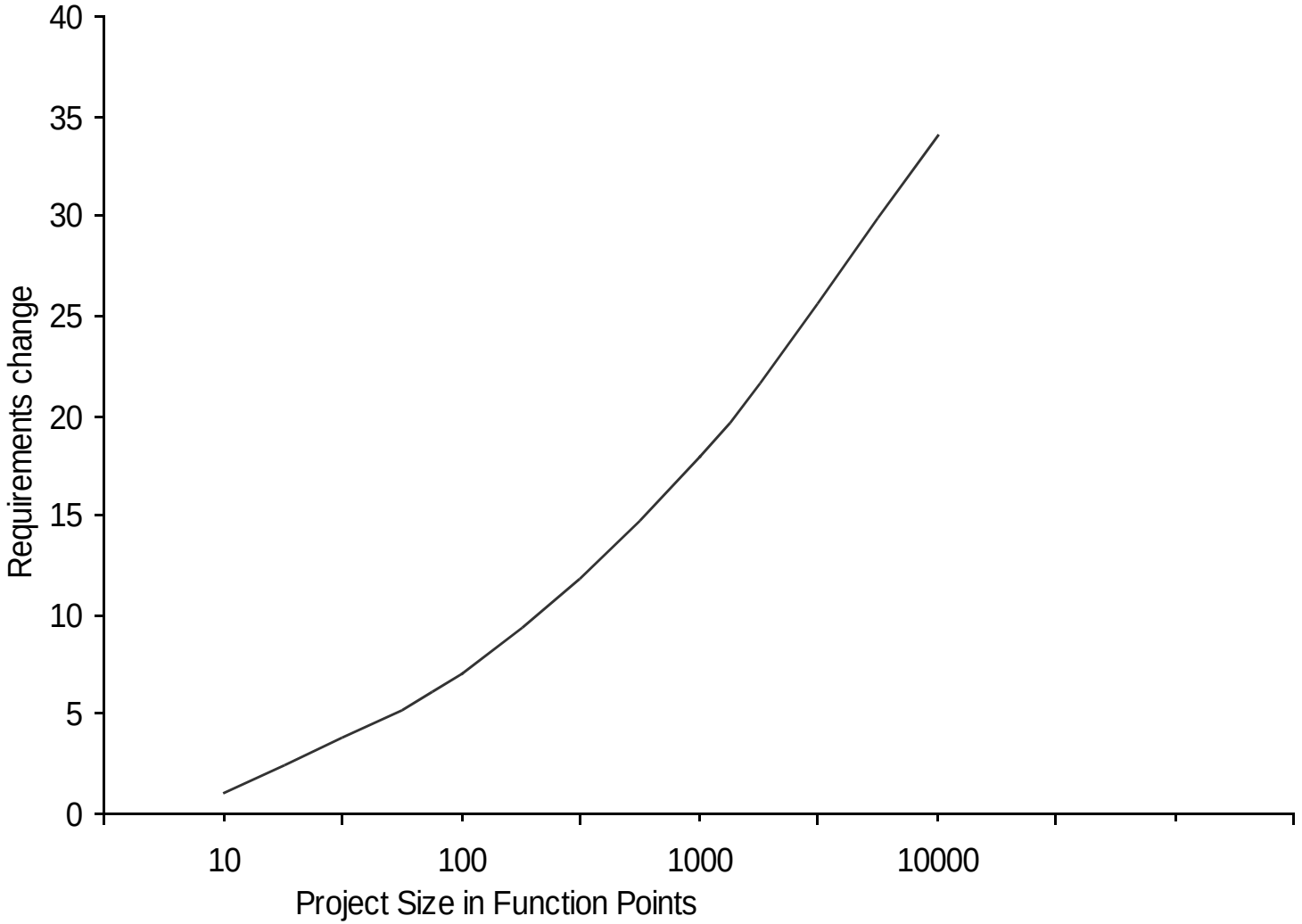


Fig. 2.3

- Key benefits of iterative process
  - Higher project success rate
    - Better management of complexity
    - Early mitigation of high risks
    - Easier adaptation to changing reqs.
  - Higher productivity & lower defect rate
  - Client visibility into project status
  - Early lessons learned applied to later iterations

Fig. 2.3

Requirements change increases with project size
Manufacture of a SW ‘product’ is not predictable
Waterfall process too rigid

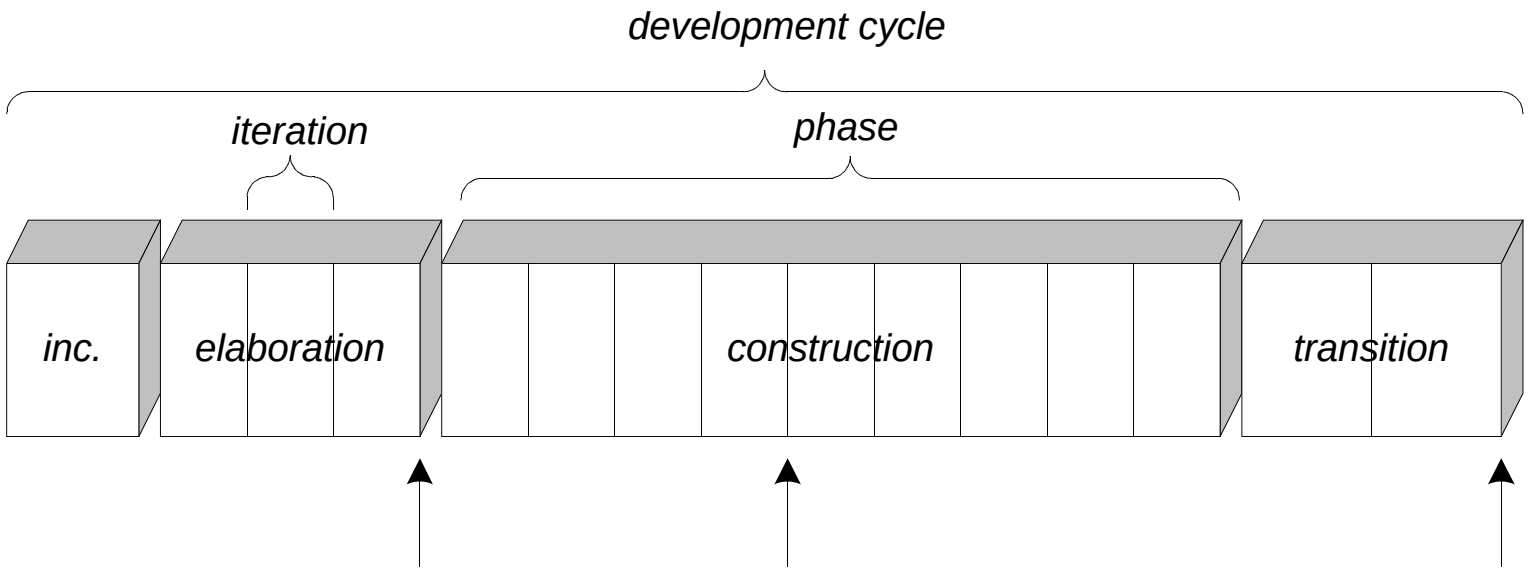


Introduction

- Important UP practices
  - Tackle hi-risk/hi-priority early
  - Continuously engage users
  - Build core architecture early
  - Verify quality by testing throughout
  - Focus on essential models using UML
  - Manage reqs. with use cases
  - Implement sound change & configuration mgmnt.

Fig. 2.6

Unified Process (UP)



milestone

An iteration end-point when some significant decision or evaluation occurs.

release

A stable executable subset of the final product. The end of each iteration is a minor release.

increment

The difference (delta) between the releases of 2 subsequent iterations.

final production release

At this point, the system is released for production use.

Fig. 2.7

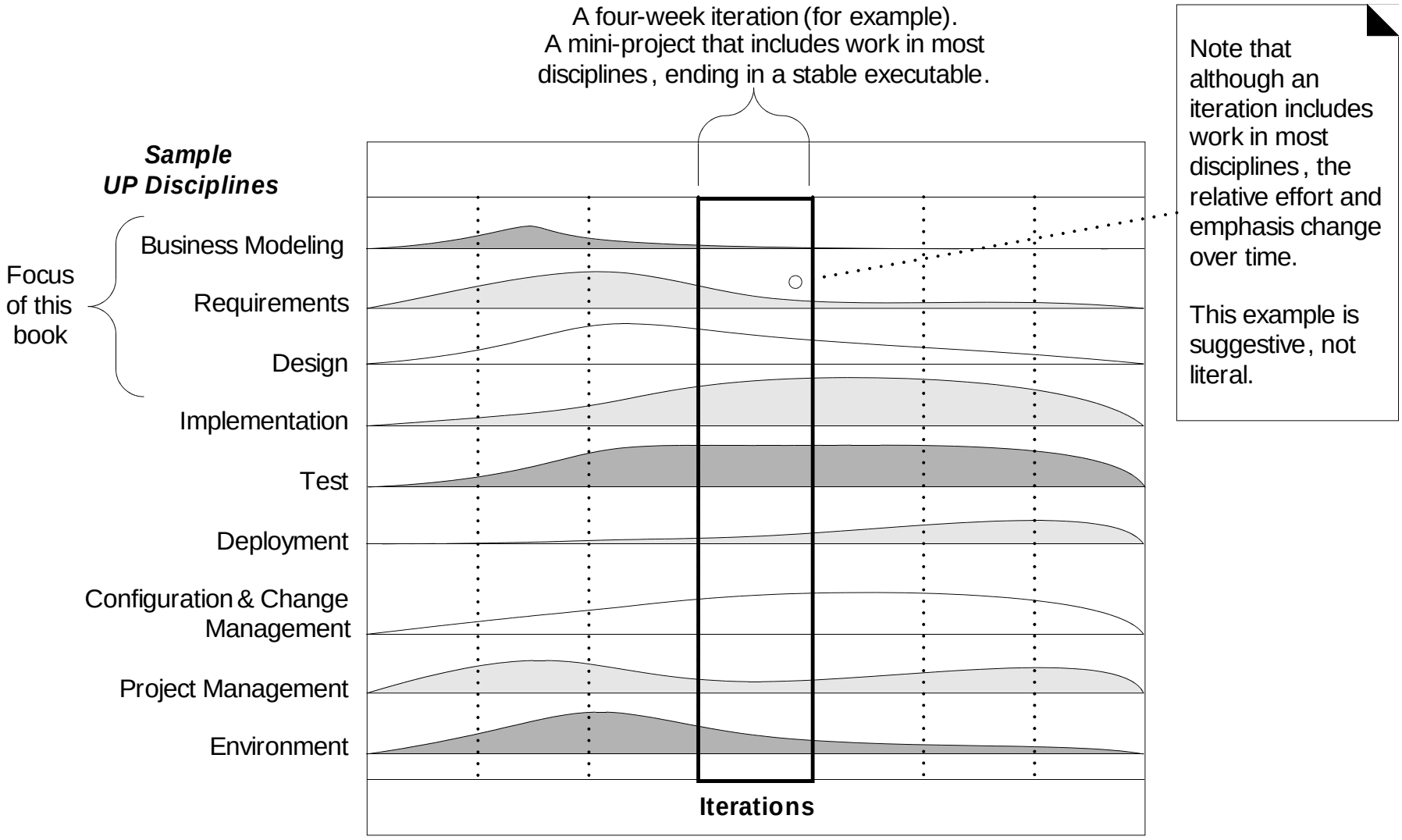
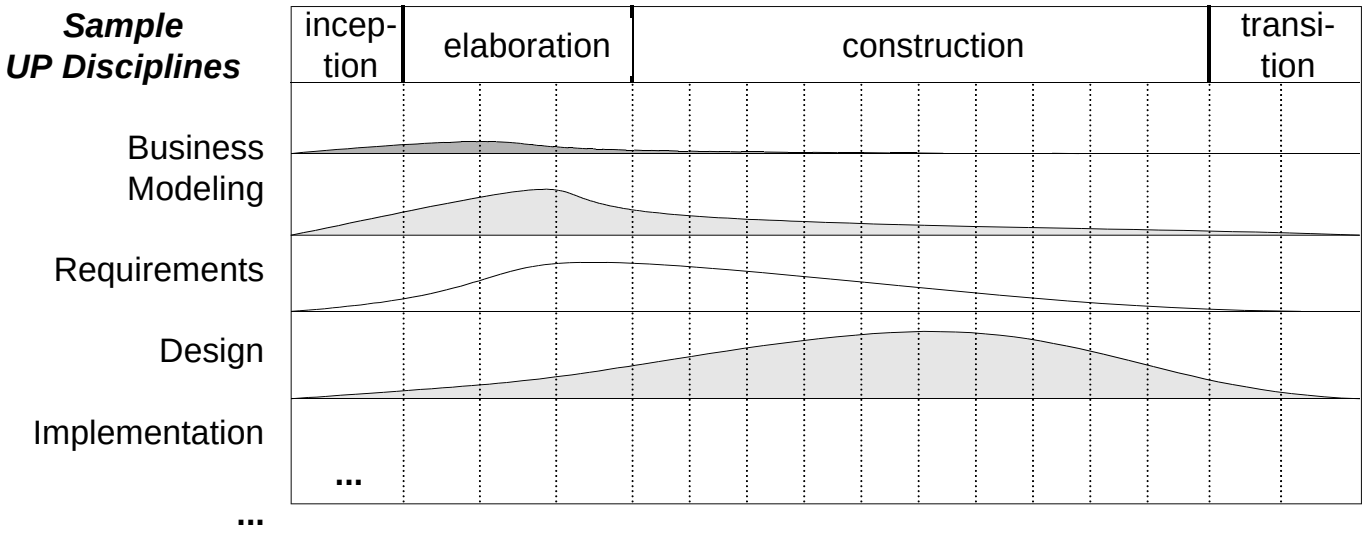


Fig. 2.8



The relative effort in disciplines shifts across the phases.

This example is suggestive, not literal.

Fig. 2.4

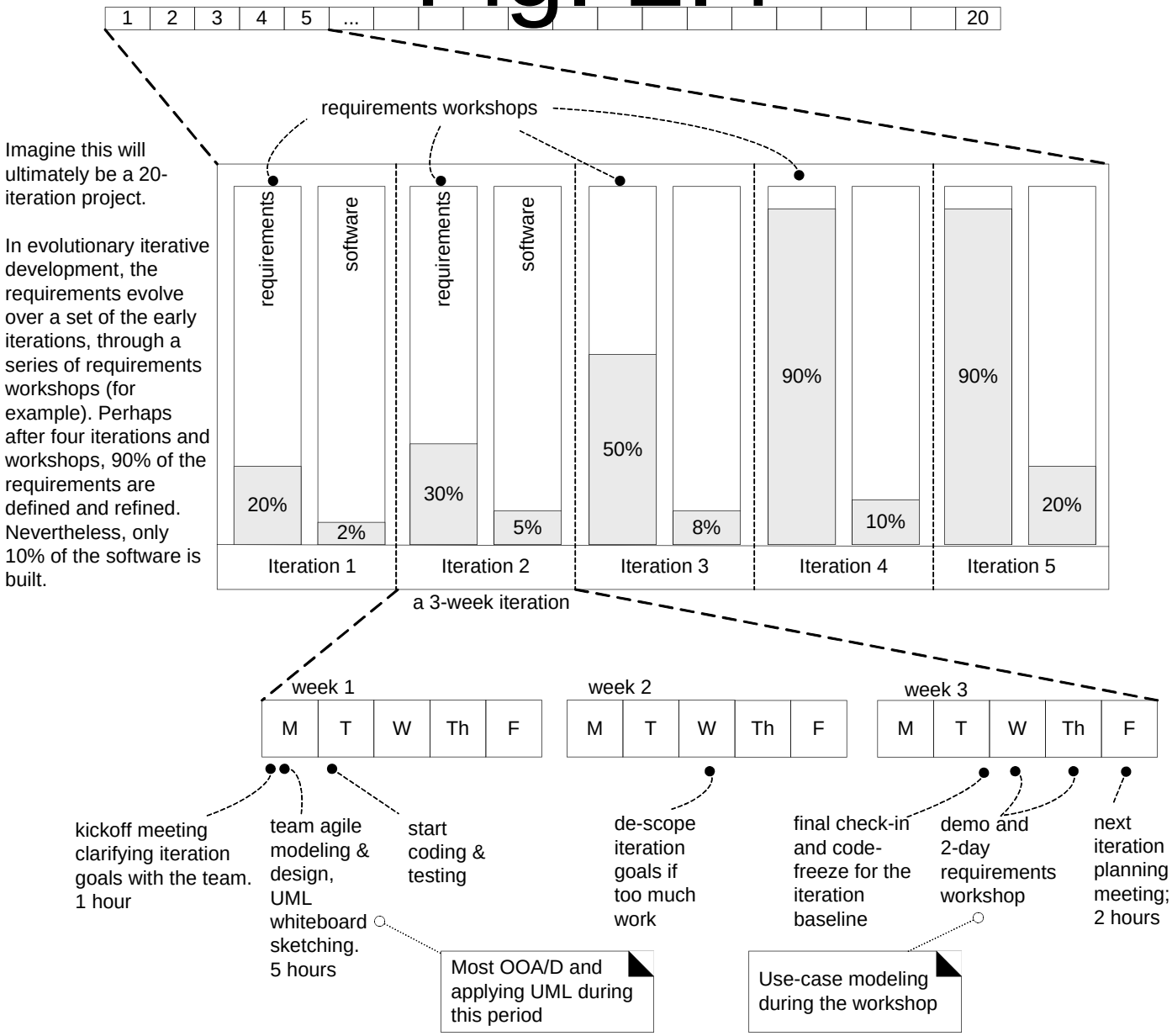


Fig. 2.9

In this course, we're primarily in the
Inception & early Elaboration phases