Object-Oriented Analysis & Design

Applying UML and Patterns by Larman, chapter 1

What is OOA/D?

- object-oriented analysis – emphasis on finding and describing the objects (or concepts) in the problem domain.
- object-oriented design – emphasis on defining software objects and how they collaborate to fulfill the requirements.

OO Analysis (flight info system)

- Domain concepts include:
  - Plane
  - Flight
  - Pilot

OO Design (flight info system)

domain concept

Plane
tailNumber

visualization of domain concept

```
public class Plane {
    private String tailNumber;
    public List getFlightHistory() {...}
}
```

representation in an object-oriented programming language

OOA/D – a short example

A “dice game” in which software simulates a player rolling two dice. If the total is seven, they win; otherwise, they lose.

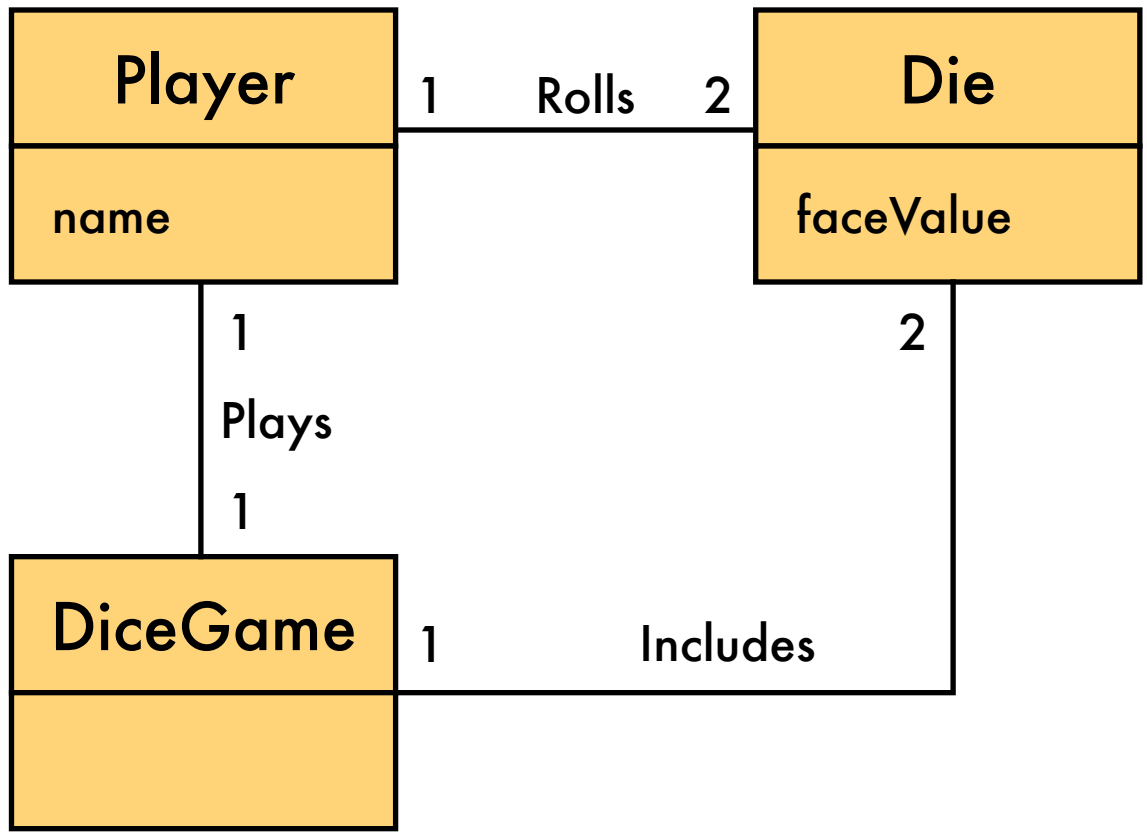
OOA/D Steps

1. Define use cases
2. Define domain model
3. Define interaction diagrams
4. Define design class diagrams

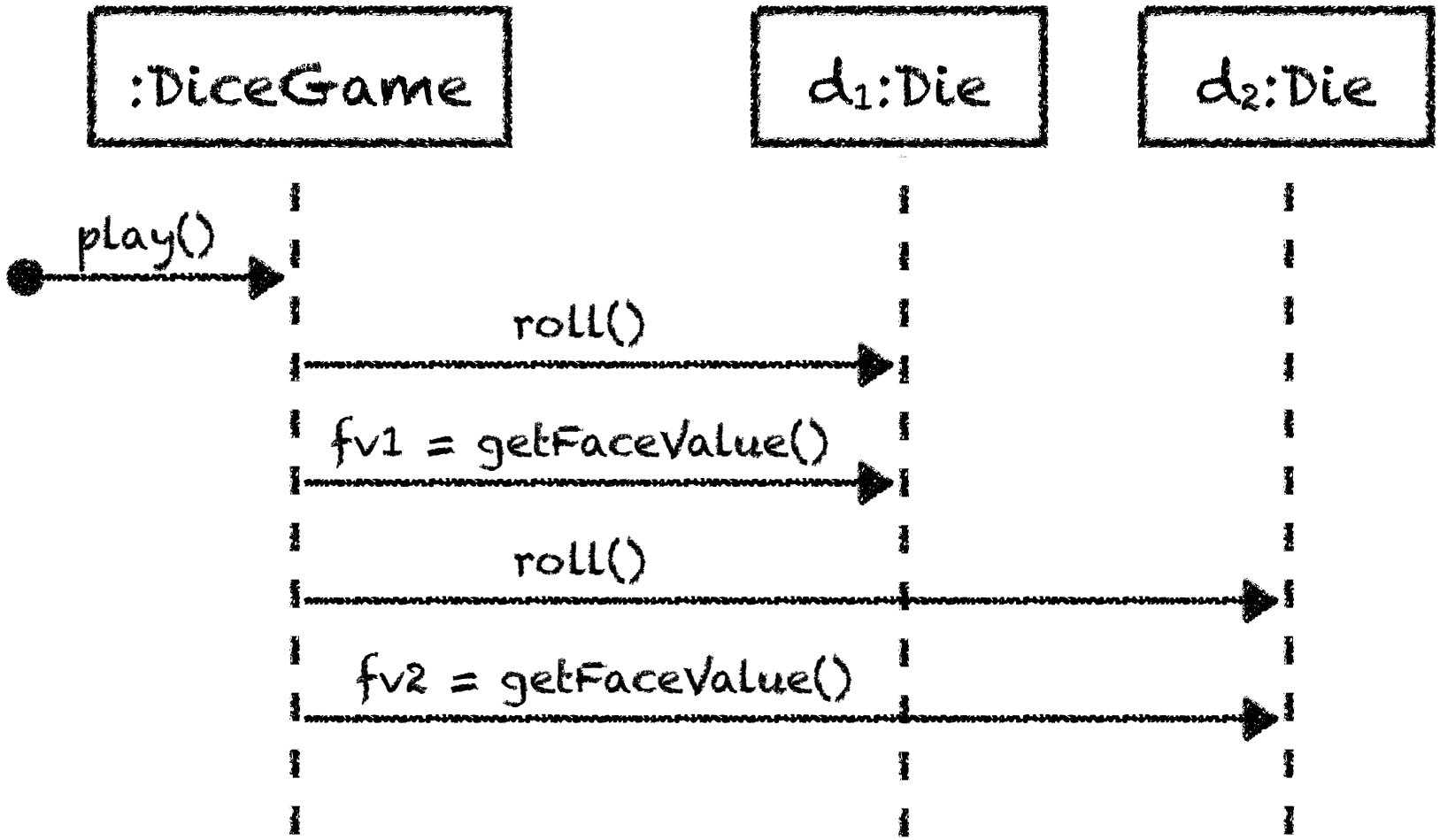
1. Define use cases

- Use cases are simply written stories
- **Play a Dice Game** – Player requests to roll the dice. System presents results: If the dice face value totals seven, player wins; otherwise, player loses.

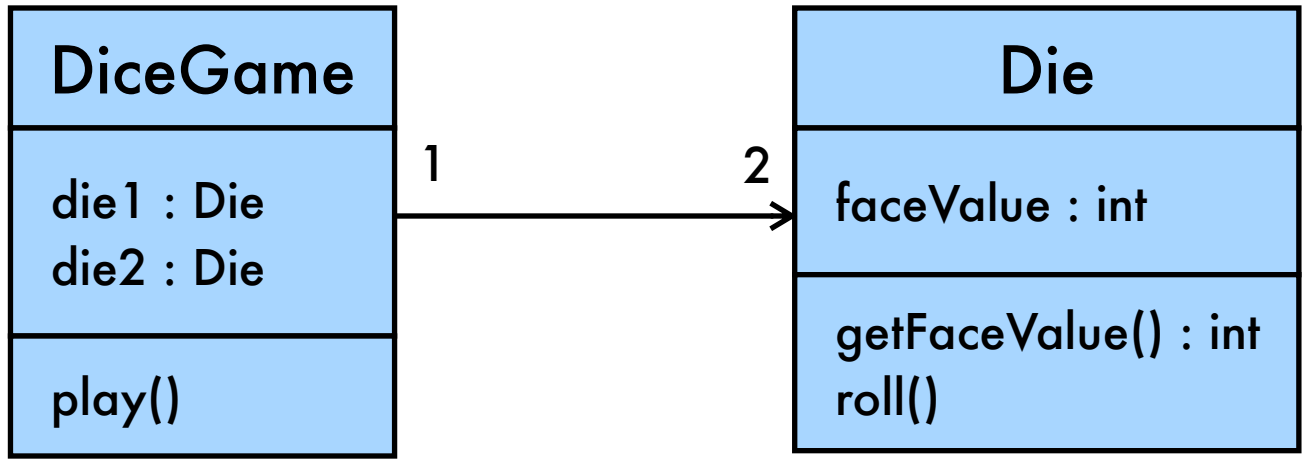
2. Define domain model



3. Define interaction diagrams



4. Define design class diagrams



Agile Modeling

The purpose of modeling is primarily to ***understand***, not to ***document***.

OOA/D, Larman

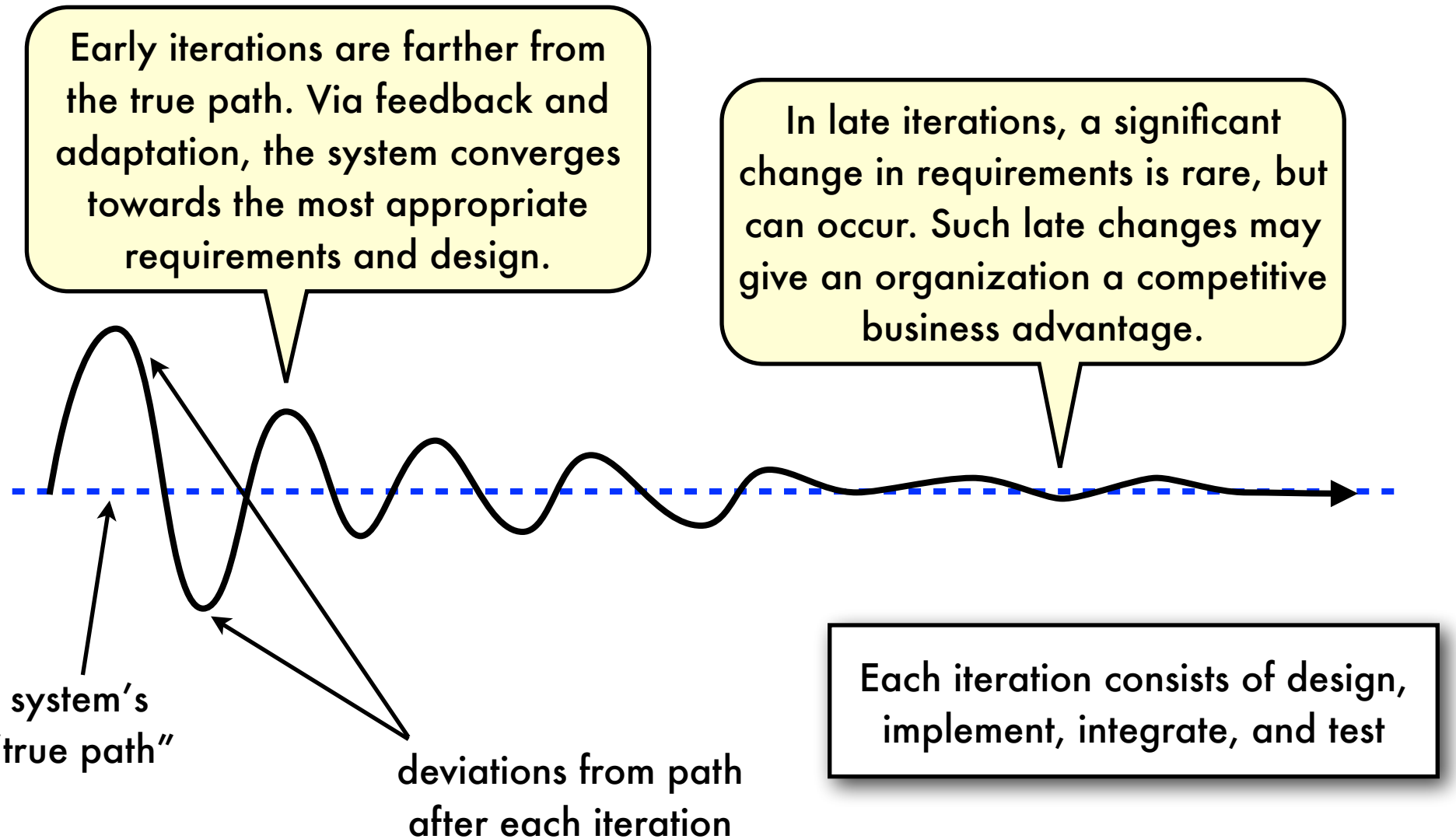
Iterative Development

Applying UML and Patterns by Larman, chapter 2

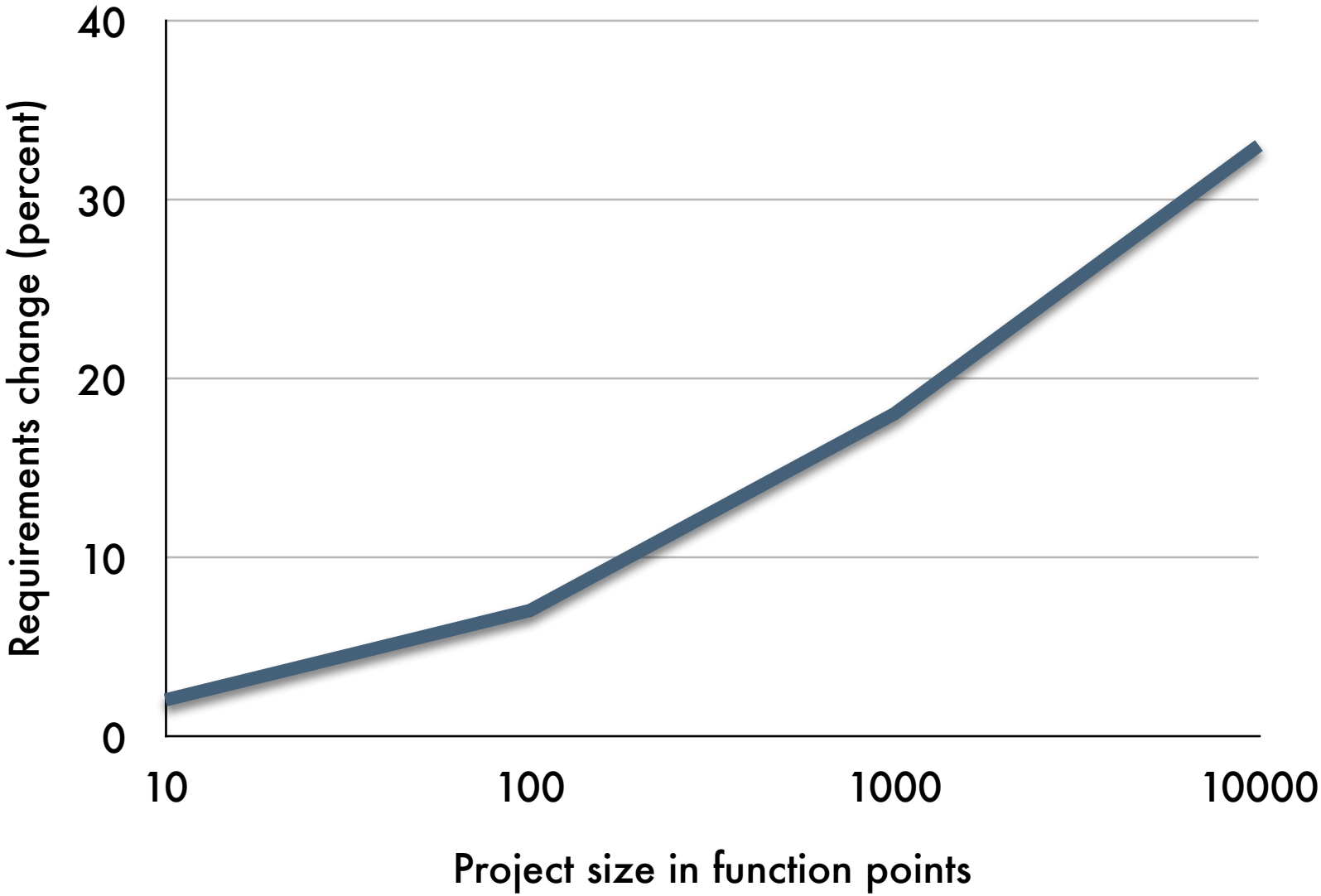
Iterative Development

- Contrasted with sequential (waterfall) development
- Involves early programming and testing of partial systems, in repeating cycles
- Assumes development starts before all requirements are defined in detail
- Feedback used to clarify and improve evolving specifications

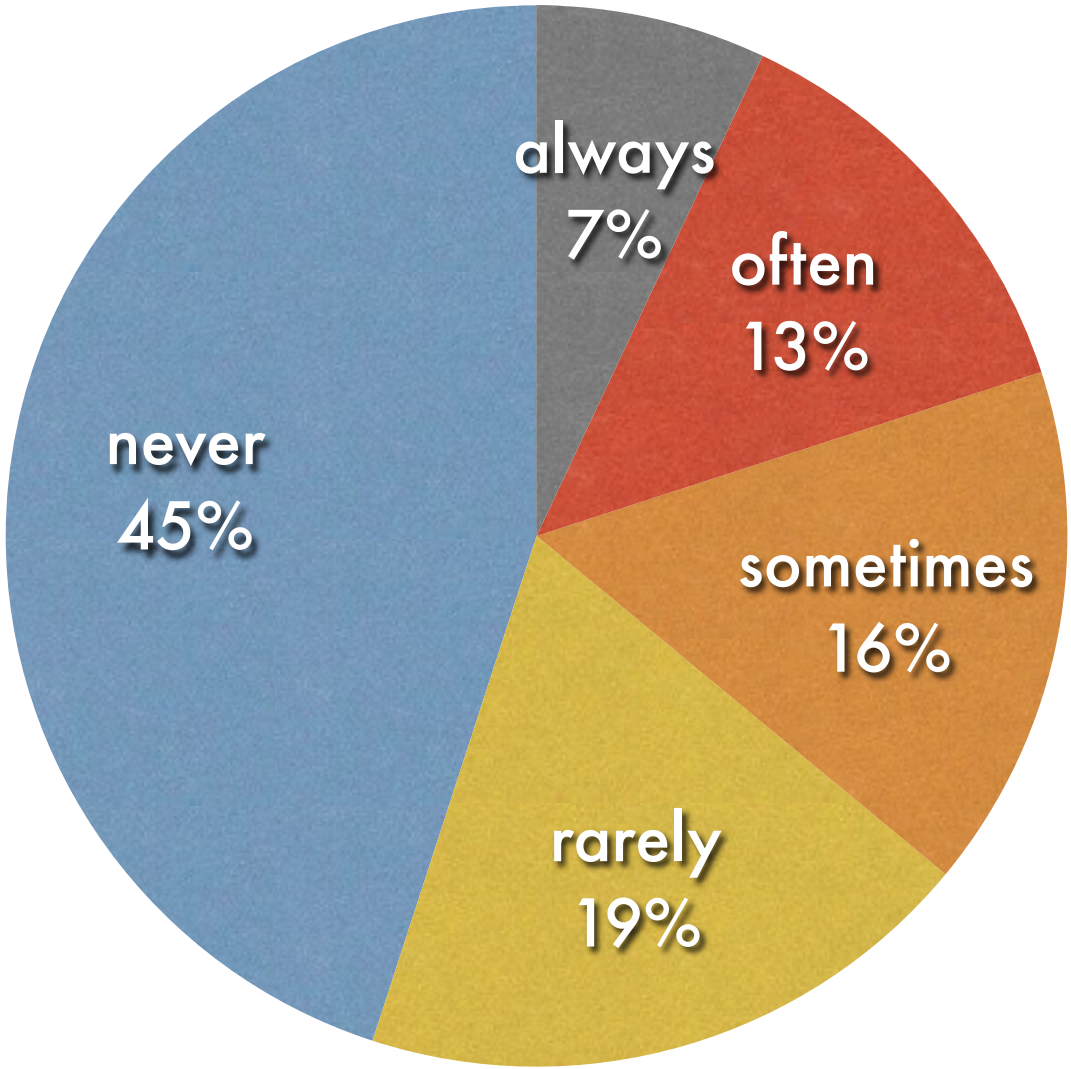
Finding the "true path"



Waterfall problems



More problems...



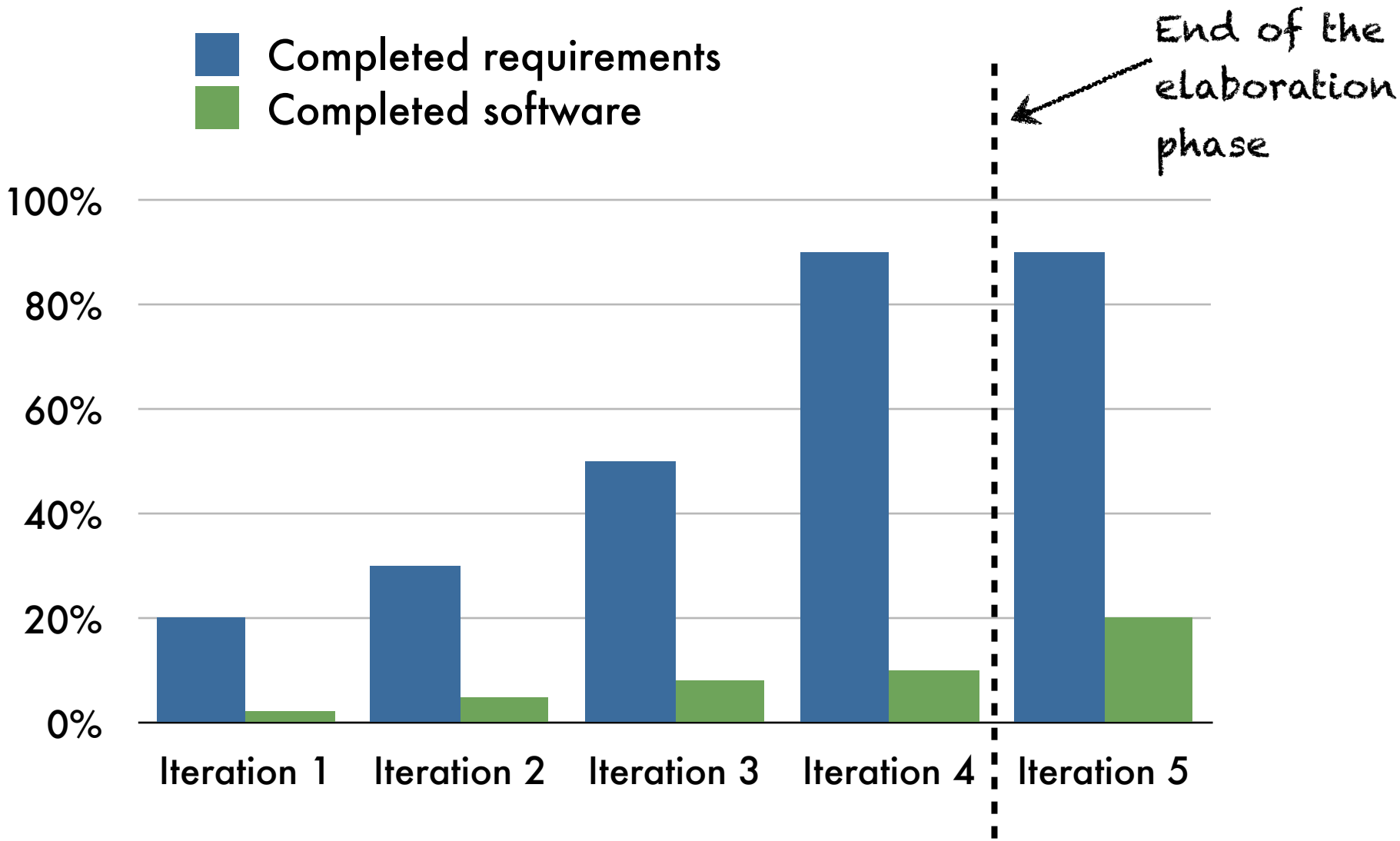
Portions indicate actual use of waterfall-specified features (Larman, §5.2)

Warning: Superimposing Waterfall on Iterative

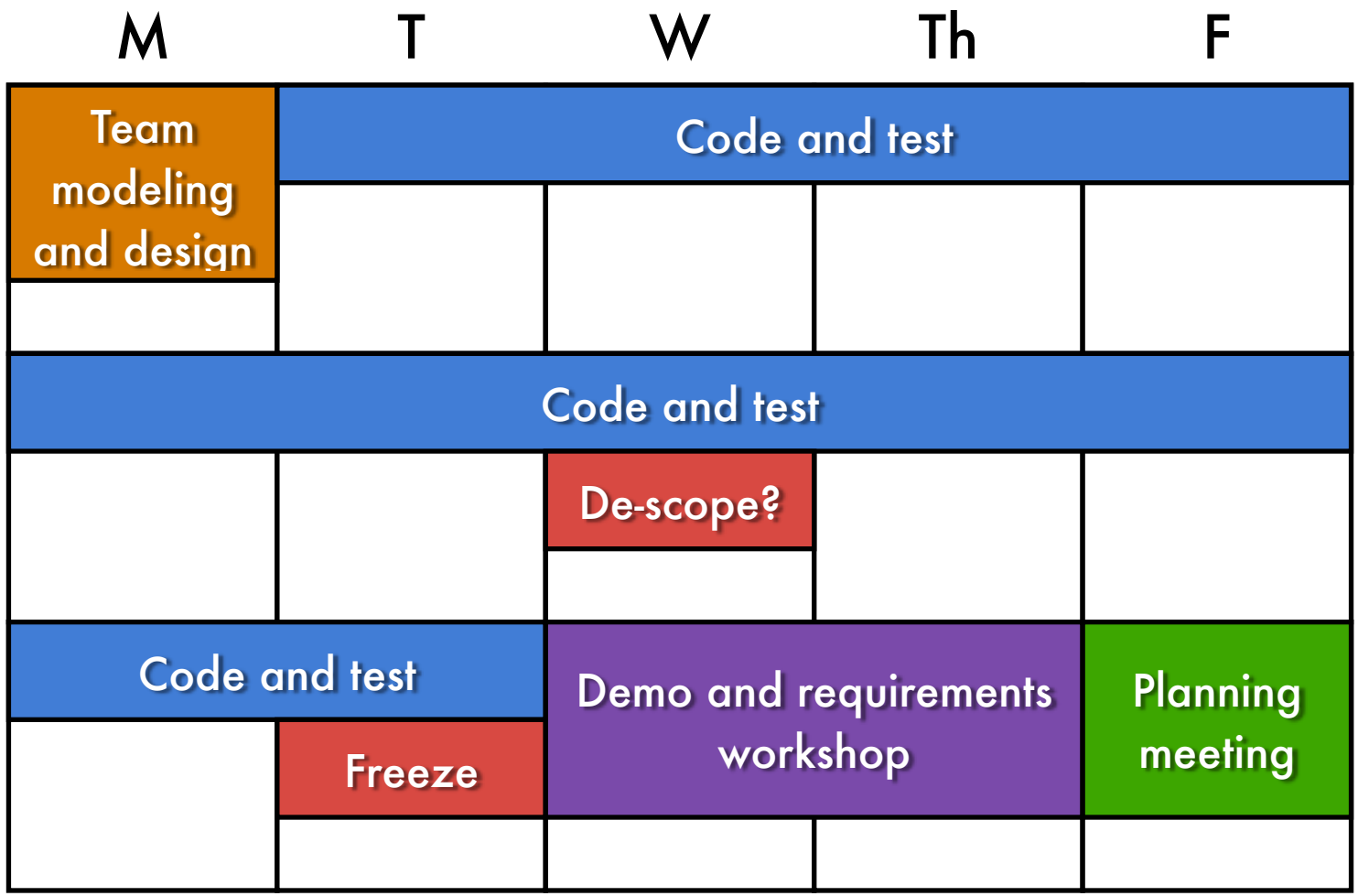
If you find yourself on an iterative project where most of the requirements are written before development begins, or there is an attempt to create many detailed UML models before programming, waterfall thinking has unfortunately afflicted the project.

OOA/D, Larman

Sample first five iterations (of a 20 iteration project)



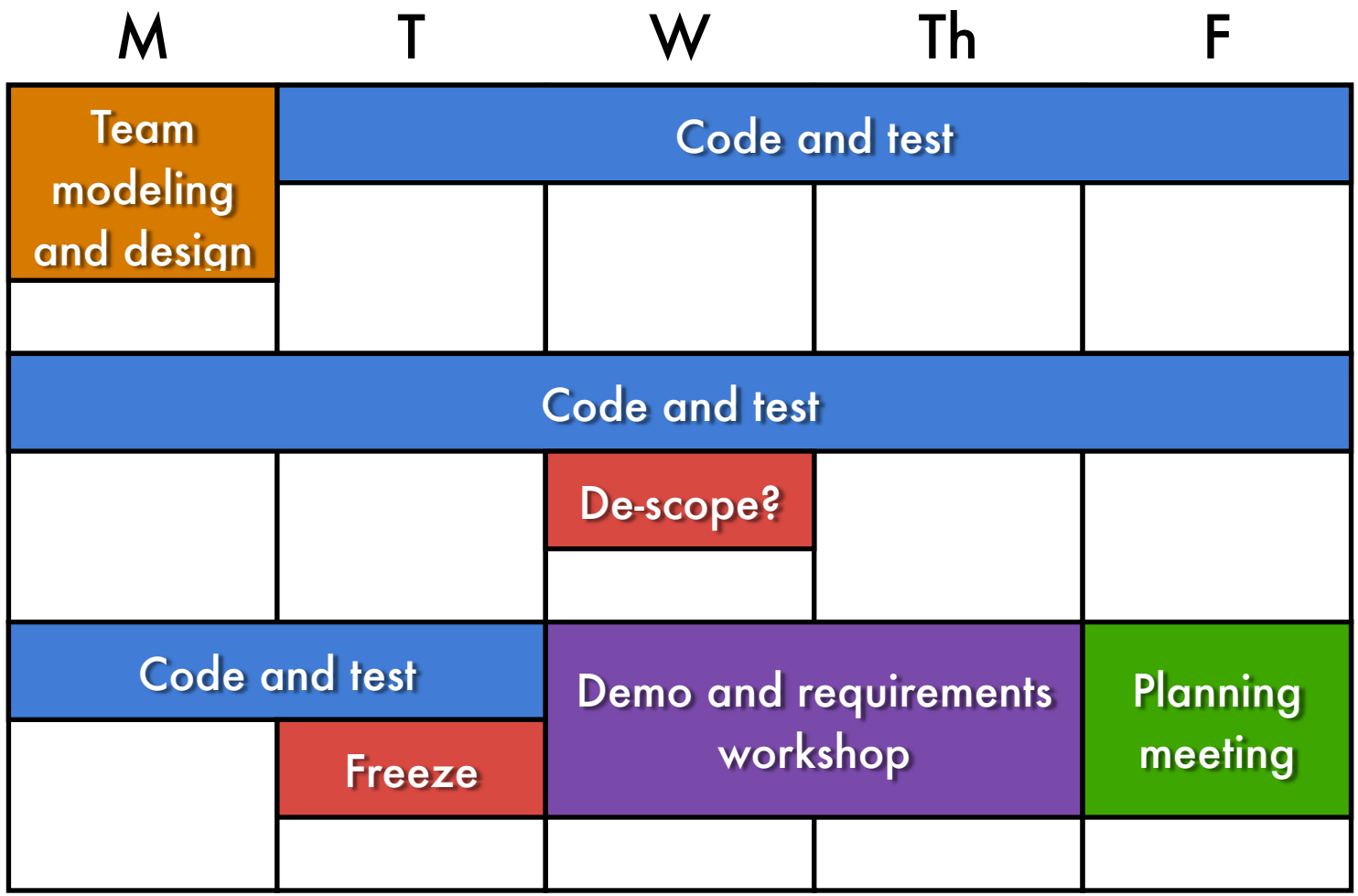
Sample 3-week iteration



Pause and Think

Why should you consider de-scoping
in the middle of an iteration and
what exactly does it mean?

Sample 3-week iteration



Unified Process

The UP was not meant by its creators to be heavy or un-agile, although its large optional set of activities and artifacts have understandably led some to that impression.

Rather, it was meant to be adopted and applied in the spirit of adaptability and lightness – an **agile UP**.

OOA/D, Larman

Adopting an Agile UP

- Prefer a small set of UP activities and artifacts
- Requirements and designs are not completed before implementation
- Apply the UML with agile modeling practices
- There is no detailed plan for the entire project

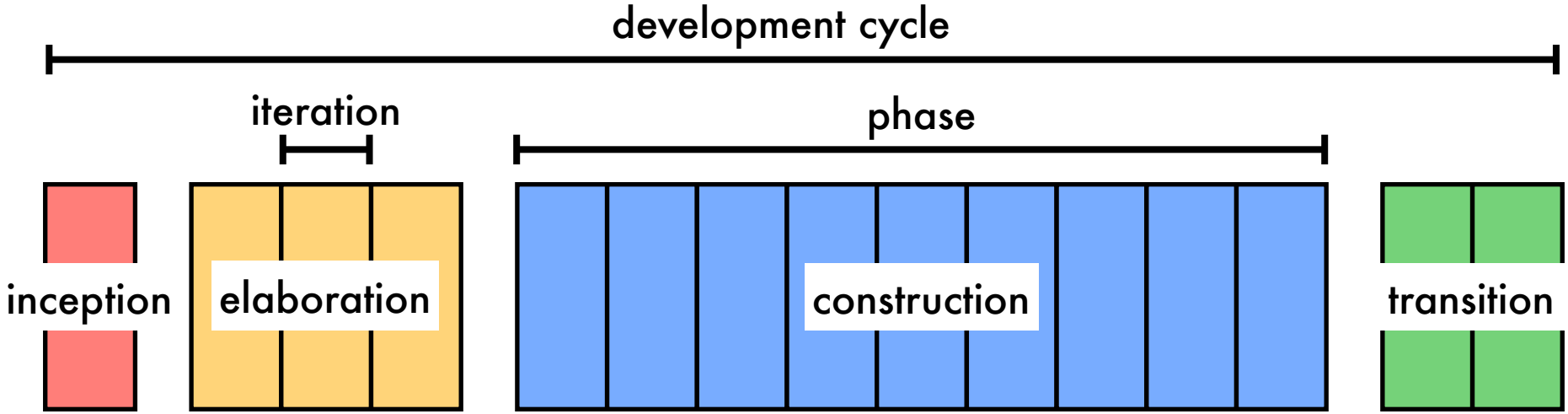
Other Critical UP Practices

- tackle high-risk and high-value issues in early iterations
- continuously engage users for evaluation, feedback, and requirements
- build a cohesive, core architecture in early iterations
- continuously verify quality; test early, often and realistically
- apply use cases where appropriate
- do some visual modeling (with the UML)
- carefully manage requirements
- practice change request and configuration management

UP Phases

- **Inception** – approximate vision, business case, scope, vague estimates.
- **Elaboration** – refined vision, iterative implementation of the core architecture, resolution of high risks, identification of most requirements and scope, more realistic estimates.
- **Construction** – iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.
- **Transition** – beta tests, deployment.

Unified Process



milestone
An iteration endpoint 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.

final production realease
At this point the system is released for production use.

Rational Unified Process