# Com S 362
# Object-Oriented Analysis & Design

UP (Universal Process) and Agile

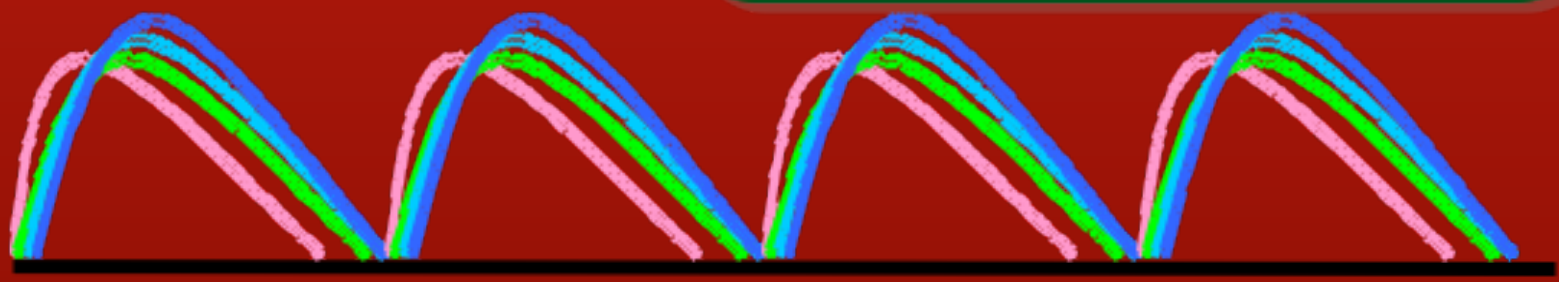
# Iterative Development vs. Waterfall

- Waterfall – complete each step before proceeding to the next
  - Requirements
  - Design
  - Implementation
  - Verification
  - Maintenance
- Fails in practice because requirements change 25 to 50% (Larman p. 24)

# Scrum: 'All At Once' with CI
## Deliver Early and Often

Rather than doing one thing at a time...

Scrum teams do a little of everything all the time.




Scrum is *not* just mini waterfall cycles!

# 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.

# UP Phases

- **Inception —** ap[illegible] NOT Requirements Phase [illegible]pe, vague estimates.
- **Elaboration —** refined vision, iterative implementation of the core archit[illegible] NOT Design Phase [illegible]*sks*, identification of r[illegible]e, more realistic estimates.
- **Construction —** iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.
- **Transition —** beta tests, deployment.

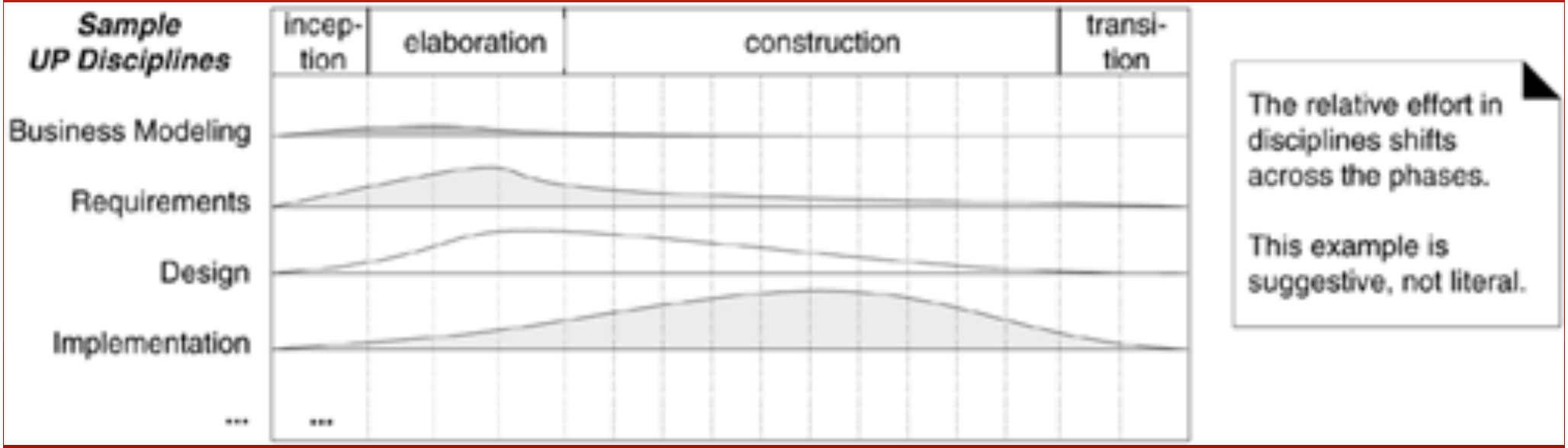
# UP Phases

- **Inception —** ap…pe, vague estimates.
- **Elaboration —** refined vision, iterative implementation of the core archit…*sks*, identification of r…e, more realistic estimates.
- **Construction —** iterative implementation of the remaining lower risk and easier elements, and preparation for deployment.
- **Transition —** beta tests, deployment.

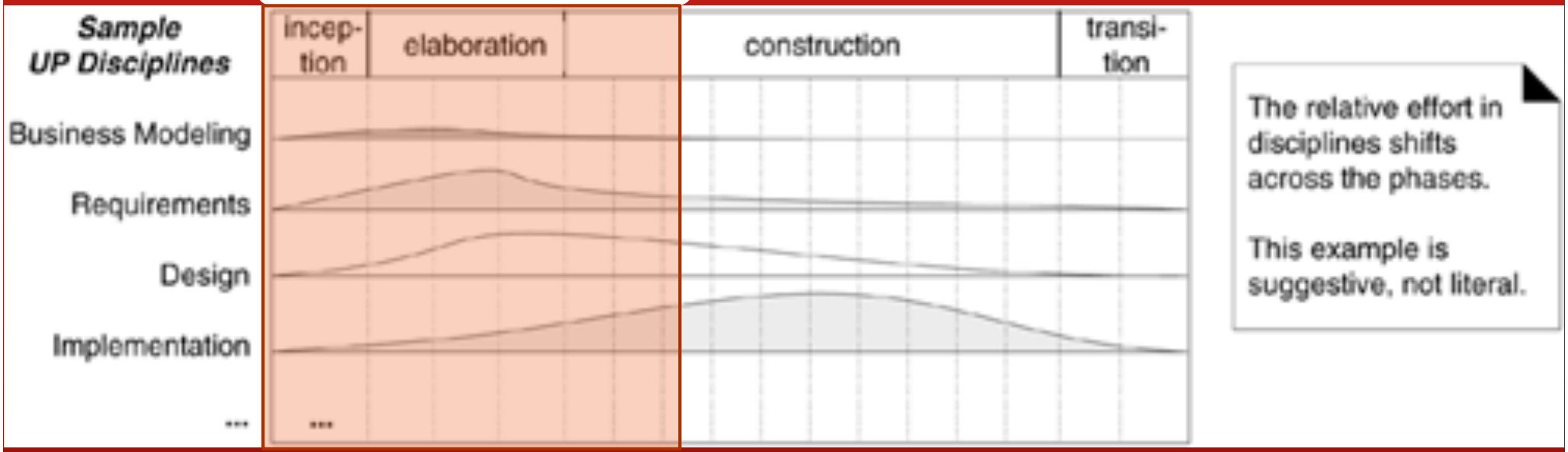
Determine feasibility

Mitigate risks

- Intensity of disciplines changes over the course of the project
- But most disciplines occur in each iteration



Focus of this class




The relative effort in disciplines shifts across the phases.

This example is suggestive, not literal.

# What is Agile?

- 'Agile' as a software development concept was introduced in the "Agile Manifesto."
- The Manifesto was produced at a meeting of 17 developers and consultants in 2001 who were promoting and exploring various "light-weight" methods.

# Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

**Individuals and interactions** over processes and tools
**Working software** over comprehensive documentation
**Customer collaboration** over contract negotiation
**Responding to change** over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

# Agile Values ≠ A Methodology

- The manifesto describes 'values' and principles (decision criteria) which the authors thought would lead to better methodologies.
- The manifesto does not describe a methodology.
  - The authors were unable to agree upon methodological specifics. After signing the manifesto, the major authors continued to promote their own methods.
- The manifesto avoids:
  - Roles, Artifacts, Rituals, Tools, and Processes.

# How should we interpret Agile?

Most useful as a historical demarcation

- 'Agile' is typically applied to any light-weight method or practice introduced after 2001.
  - Even though the method may have little in common with other 'agile' methods, and
  - Even though the practice may be used in a heavyweight, plan-driven operation.

# A more precise view

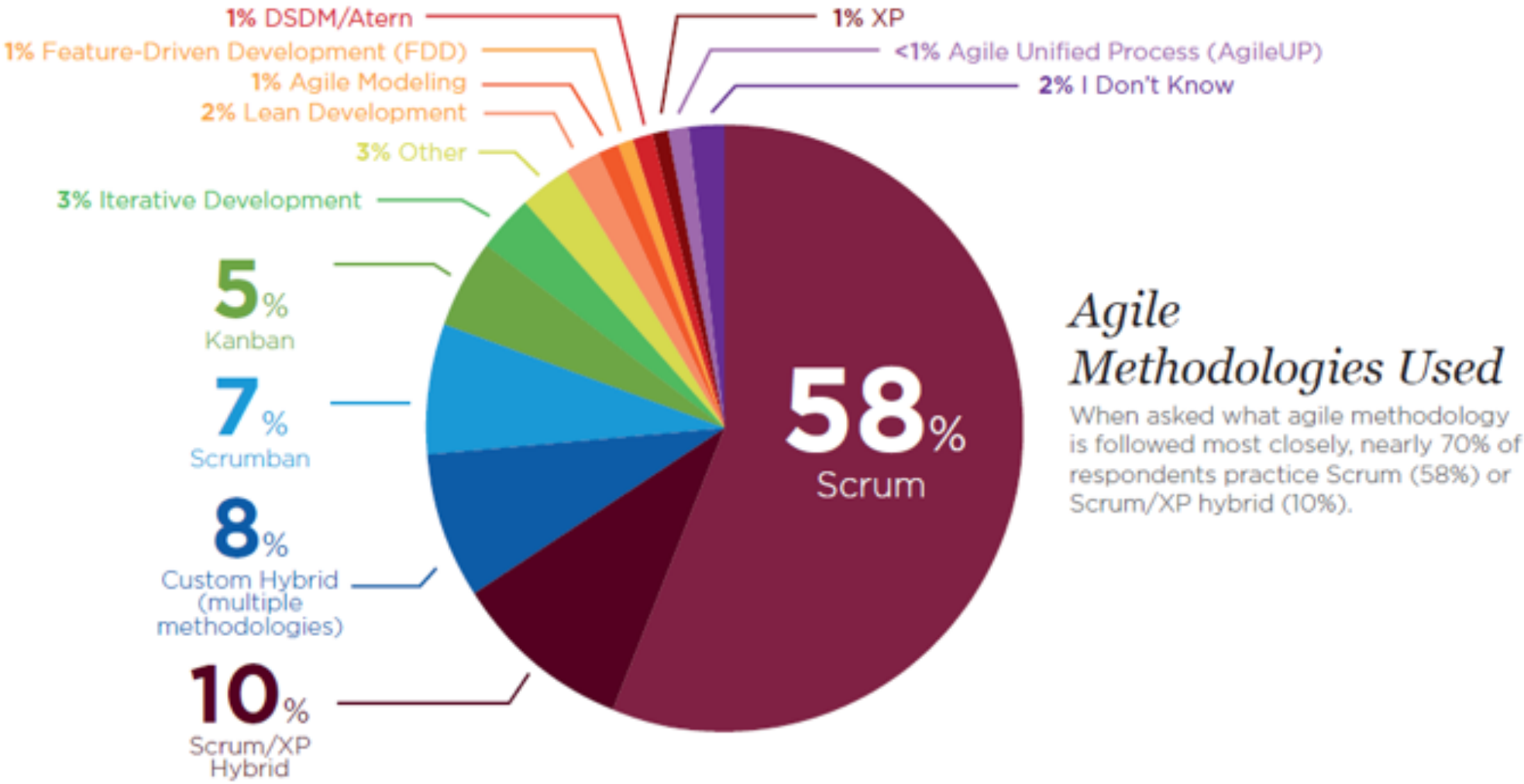
'agile' usually implies that the methodology:

- Is iterative and incremental (IID) with short time-boxed iterations
- Is "light-weight" (avoids extensive analysis, documentation, and prescriptive processes.)
- Delays many product, design, and implementation decisions to the "last responsible moment."
- Prioritizes work according to perceived customer value.
- Uses a pull-mode work-release strategy (developers decide how much work they can do.)
- Uses "shippable code" as the only measure of progress.

# Reality Check

- Textbook Says: “The Unified Process has emerged as a popular iterative software development process ...”

- Survey Says ...



# Scrum

- Most popular of leading agile approaches.
- 71% of organizations report using agile practices always, often, or sometime.
- But we don’t know what they consider agile!
- 75% of those using agile, report they are using Scrum.
- As originally described, scrum is focused mostly on team activities.

# Scrum

- Team consists of Product Owner, Scrum Master, and Dev Team
- Prescribed meetings:
  - Sprint planning – decide what to work on
  - Daily stand-up – commit to each other on particular issues
  - Sprint demo/Sprint review – demo the sprint's work
  - Sprint retrospective – what went right and wrong