Driving Development with Tests

Emily Bache
@emilybache
emily@bacheconsulting.com

material copyright Emily Bache consulting, emily@bacheconsulting.com

Driving Development with Tests

Emily Bache
@emilybache
emily@bacheconsulting.com

material copyright Emily Bache consulting, emily@bacheconsulting.com

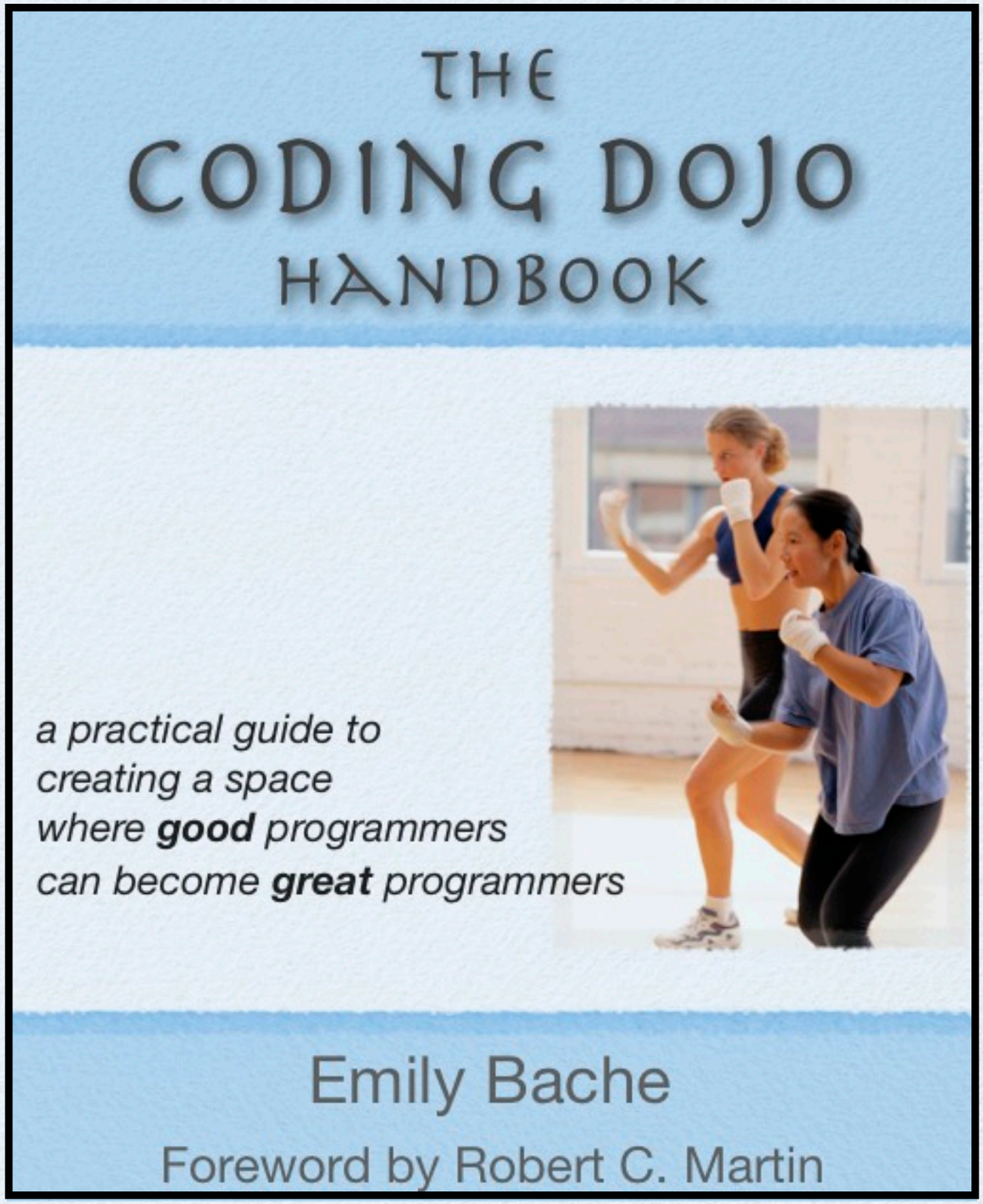
The Coding Dojo

- Coding Dojo: better coding skills
- Practice in a safe environment
- Discuss actual code
- have fun!

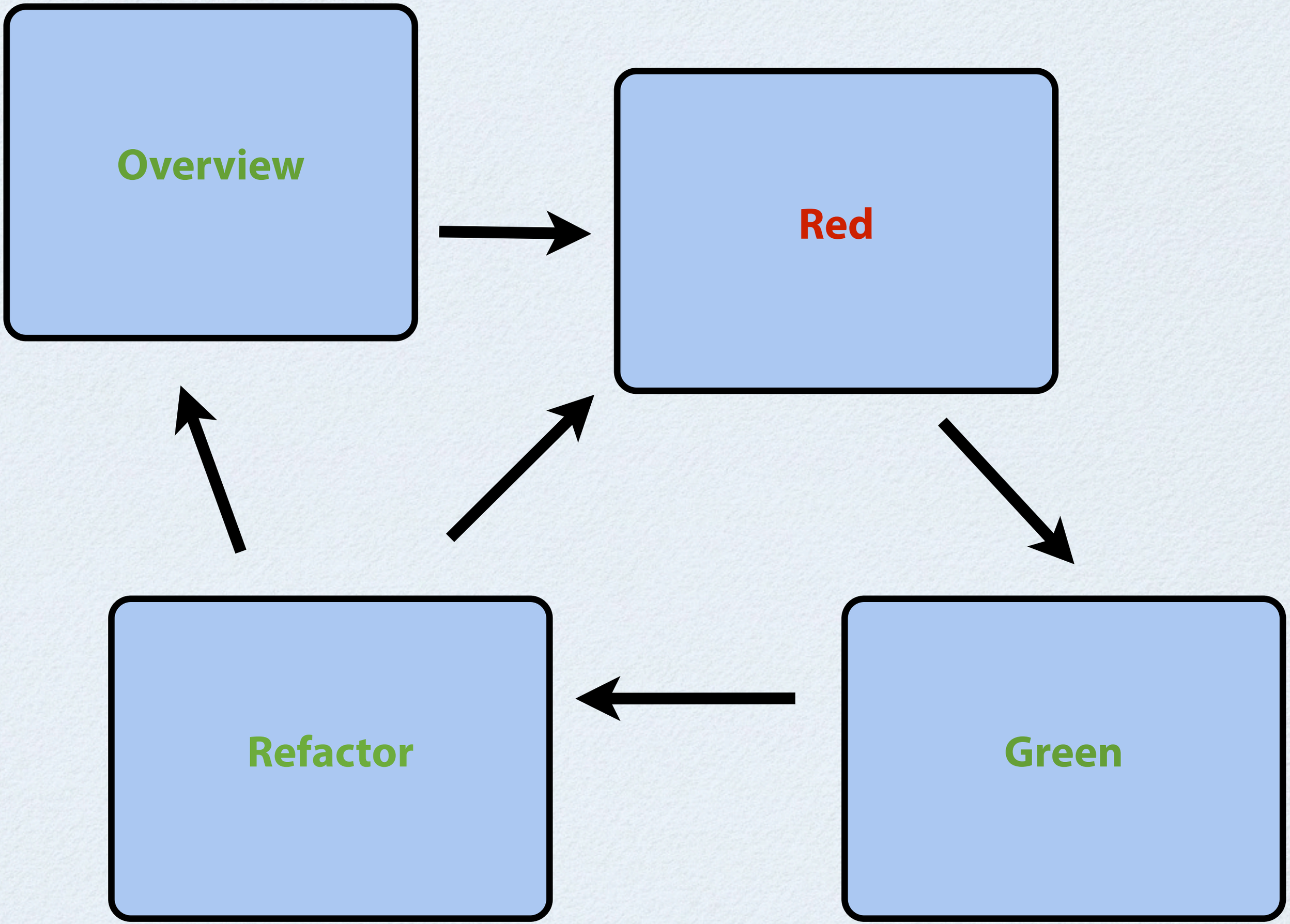
material copyright Bache consulting, emily@bacheconsulting.com

Available from:

http://leanpub.com



material copyright Bache consulting, emily@bacheconsulting.com



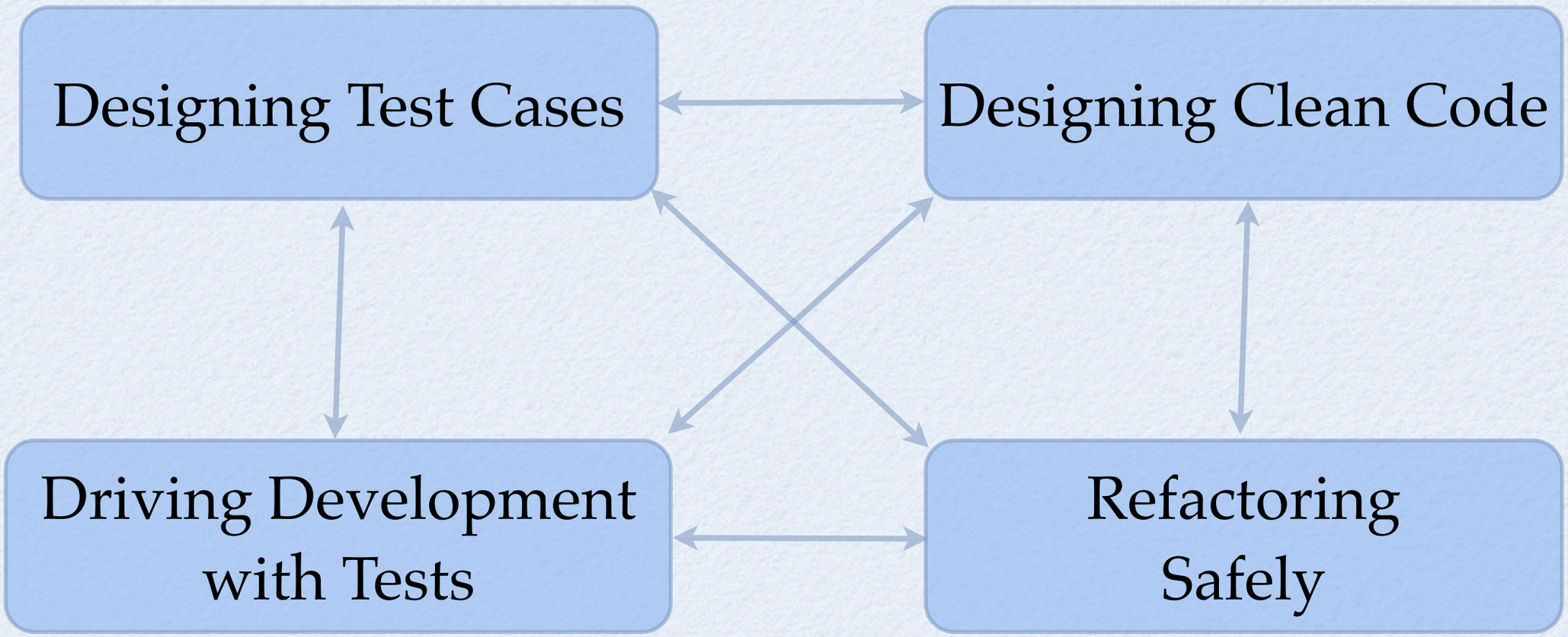
material copyright Bache consulting, emily@bacheconsulting.com

Rhythm of TDD

- Small increments of functionality, "pulled" into existence by a new test
- All existing tests keep passing
- Refactoring often & minor
- When your guiding test passes, you feel confident the code works

material copyright Emily Bache consulting, emily@bacheconsulting.com

TDD skills



In the dojo we can focus on one at a time

material copyright Emily Bache consulting, emily@bacheconsulting.com

Overview
Analyse Problem
Test List
Guiding Test

next test chosen

Red
Declare & Name
Arrange-Act-Assert
Satisfy compiler

test runs and fails

Green
Implement solution
Fake it
Start over

all tests pass

Refactor
Remove Fake
Remove Code Smell
(No new functionality)
Note new test cases

next test unclear

next test chosen

material copyright Bache consulting, emily@bacheconsulting.com

Overview

Analyse Problem

Test List

Guiding Test

next test chosen

Red

Declare & Name

Arrange-Act-Assert

Satisfy compiler

test runs and fails

Green

Implement solution

Fake it

Start over

all tests pass

Refactor

Remove Fake

Remove Code Smell

(No new functionality)

Note new test cases

next test chosen

next test unclear

material copyright Bache consulting, emily@bacheconsulting.com

Overview
Analyse Problem
Test List
Guiding Test

next test chosen

Red
Declare & Name
Arrange-Act-Assert
Satisfy compiler

next test unclear

next test chosen

test runs and fails

Test too hard

Refactor
Remove Fake
Remove Code Smell
(No new functionality)
Note new test cases

all tests pass

Green
Implement solution
Fake it
Start over

material copyright Bache consulting, emily@bacheconsulting.com

Overview

Analyse Problem
Test List
Guiding Test

next test chosen

Red

Declare & Name
Arrange-Act-Assert
Satisfy compiler

test runs and fails

Green

Implement solution
Fake it
Start over

all tests pass

Refactor

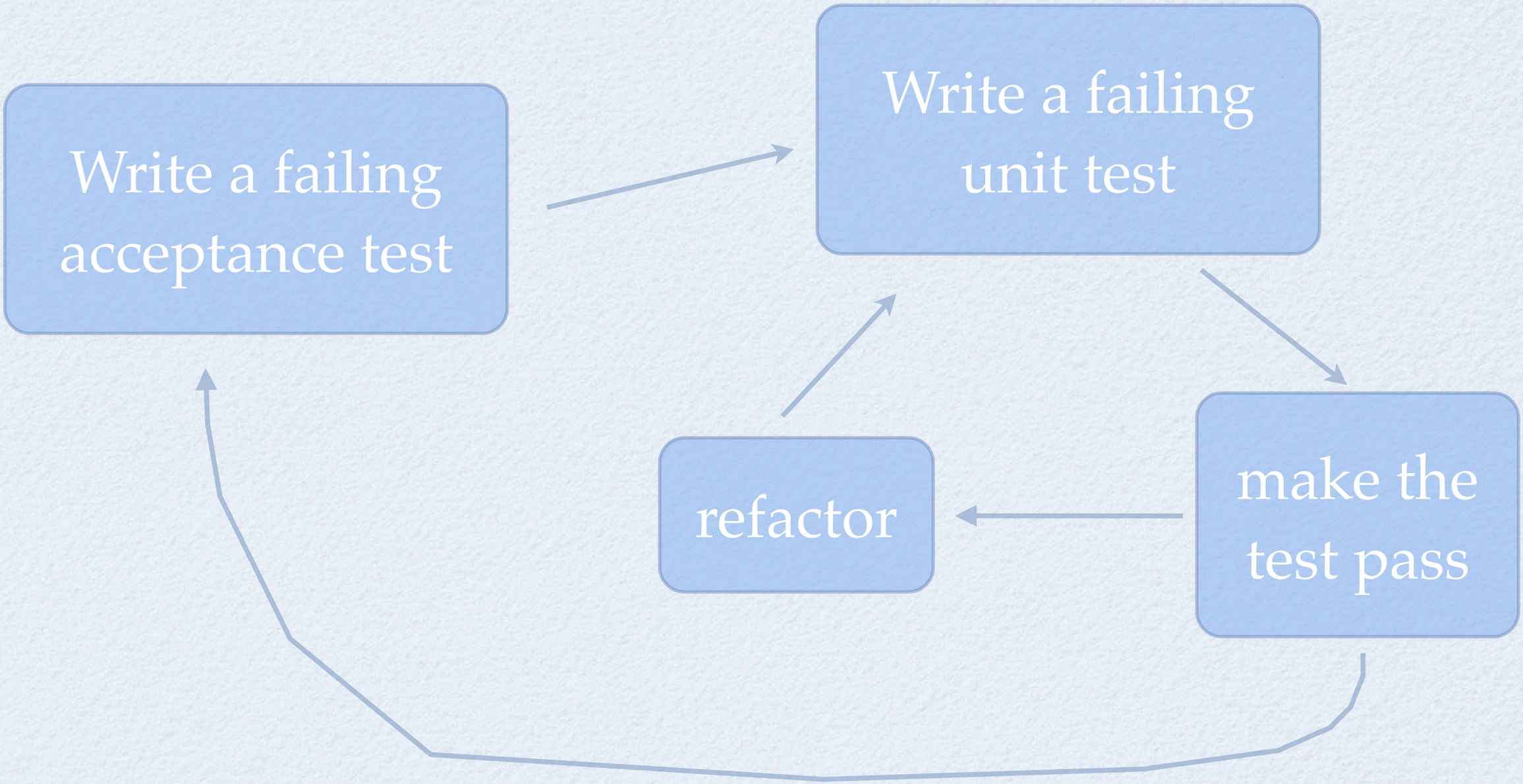
Remove Fake
Remove Code Smell
(No new functionality)
Note new test cases

next test chosen

next test unclear

material copyright Bache consulting, emily@bacheconsulting.com

Double-Loop TDD



p40 GOOS, Freeman, Pryce

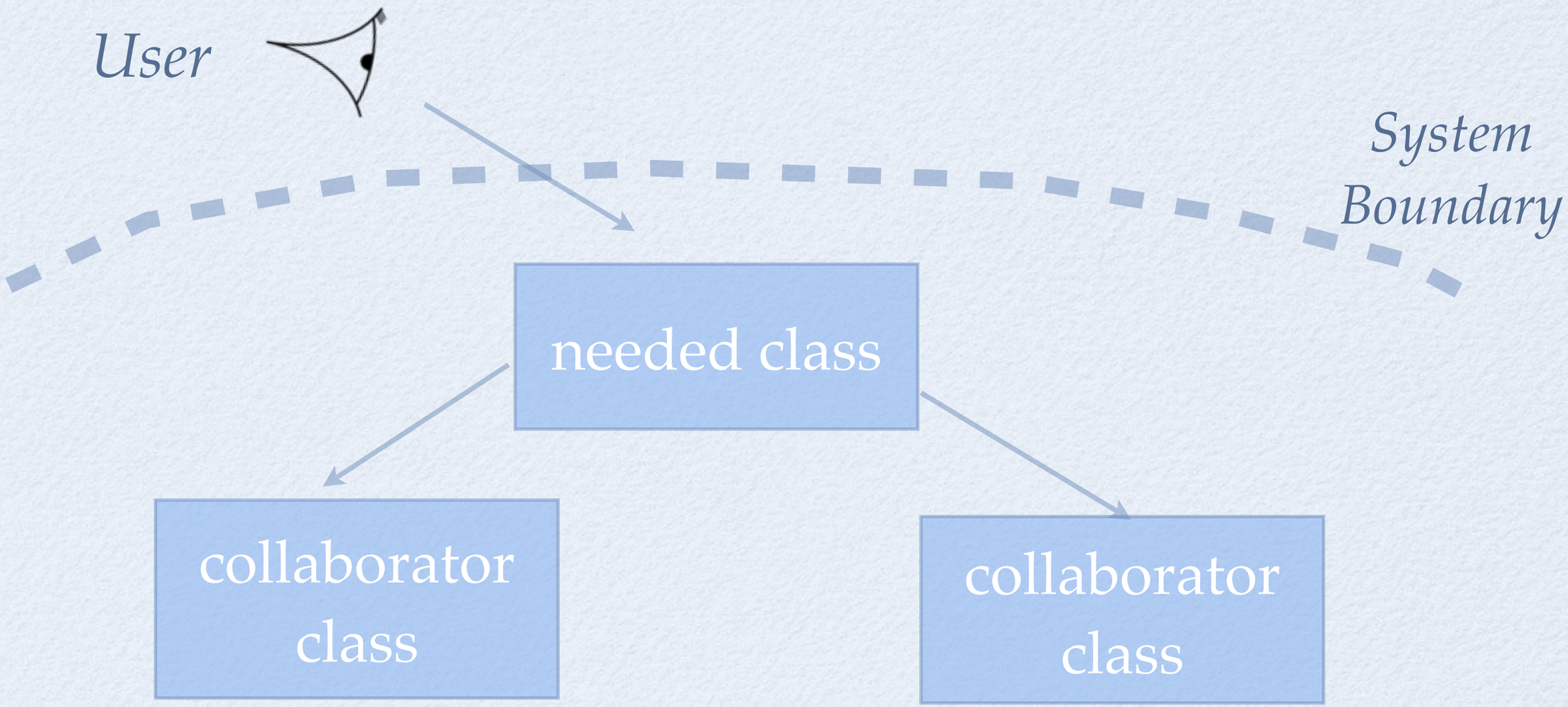
material copyright Bache consulting, emily@bacheconsulting.com

Train Reservation

- A code kata I designed.
- Needs more than one class to implement.
- Potential to use Mocks and/or Stubs

material copyright Emily Bache consulting, emily@bacheconsulting.com

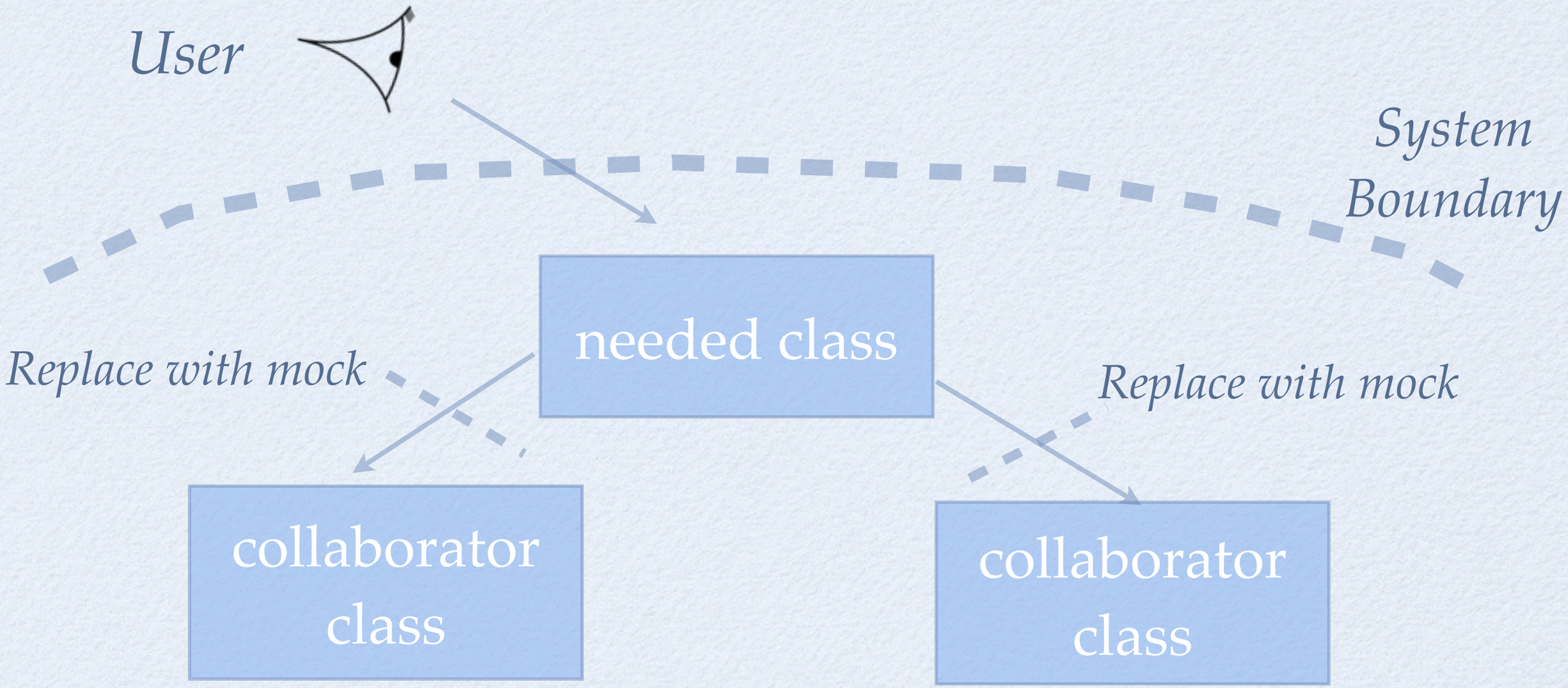
London School Outside - In



p 61, GOOS, Freeman, Pryce

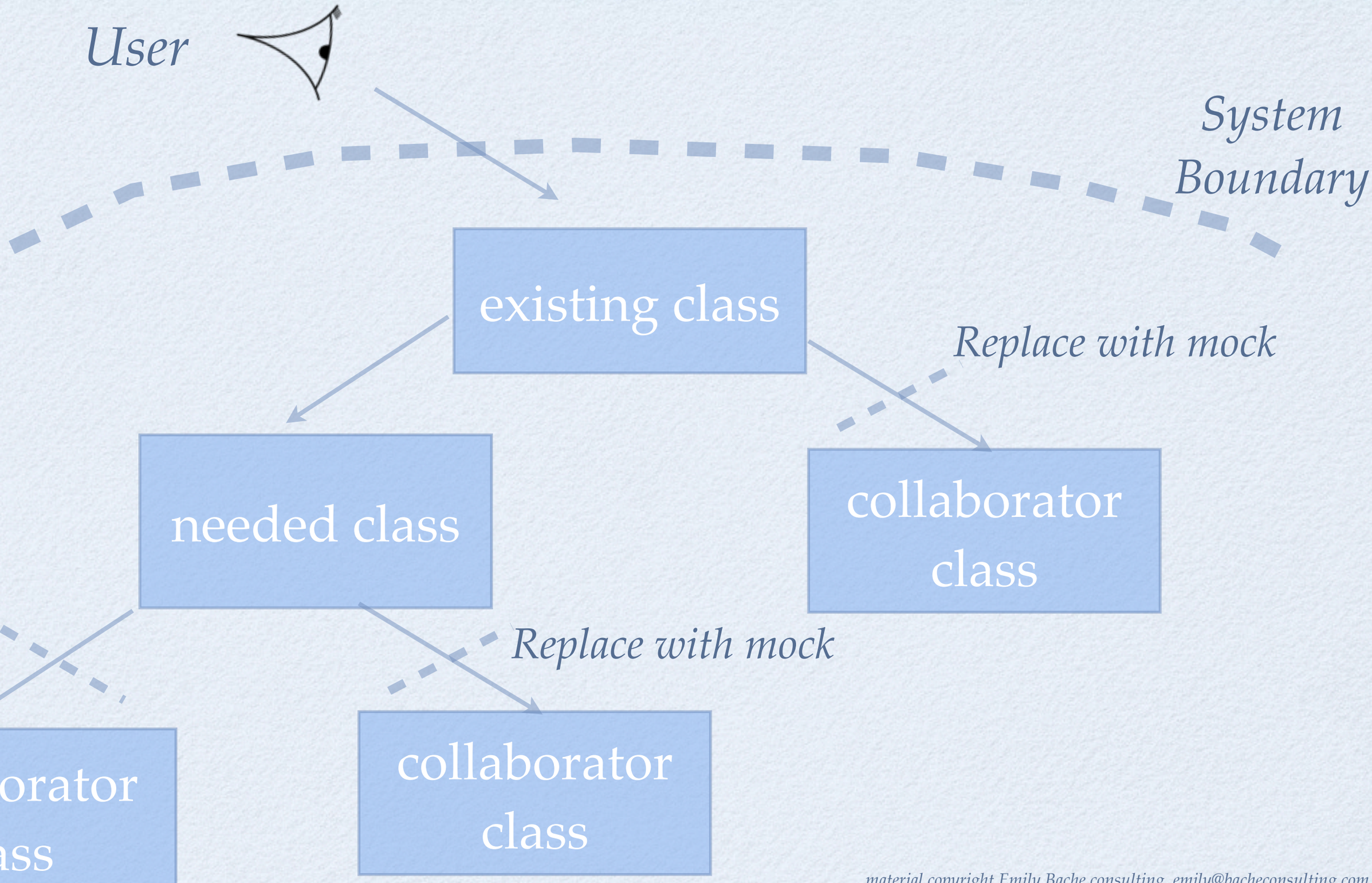
material copyright Emily Bache consulting, emily@bacheconsulting.com

London School Outside - In



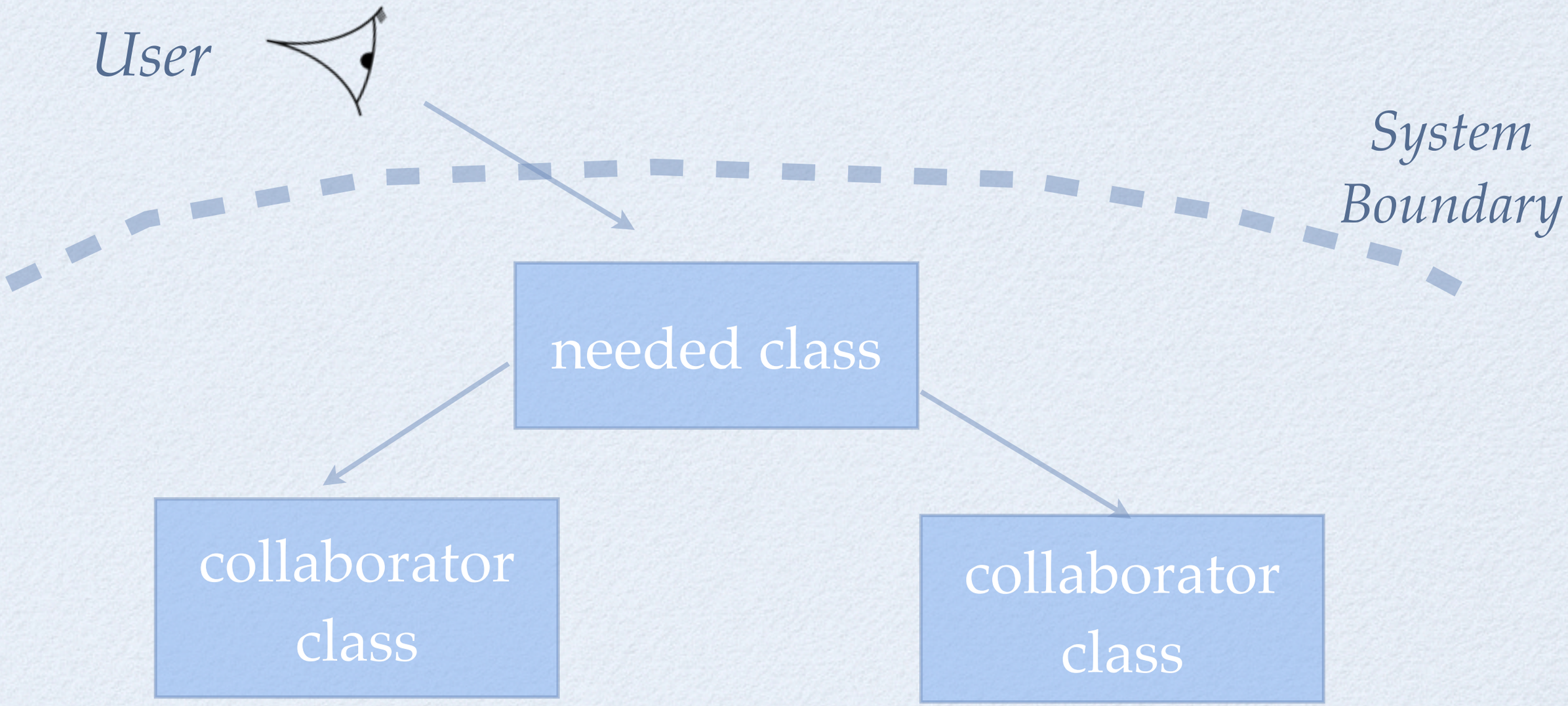
material copyright Emily Bache consulting, emily@bacheconsulting.com

London School Outside - In



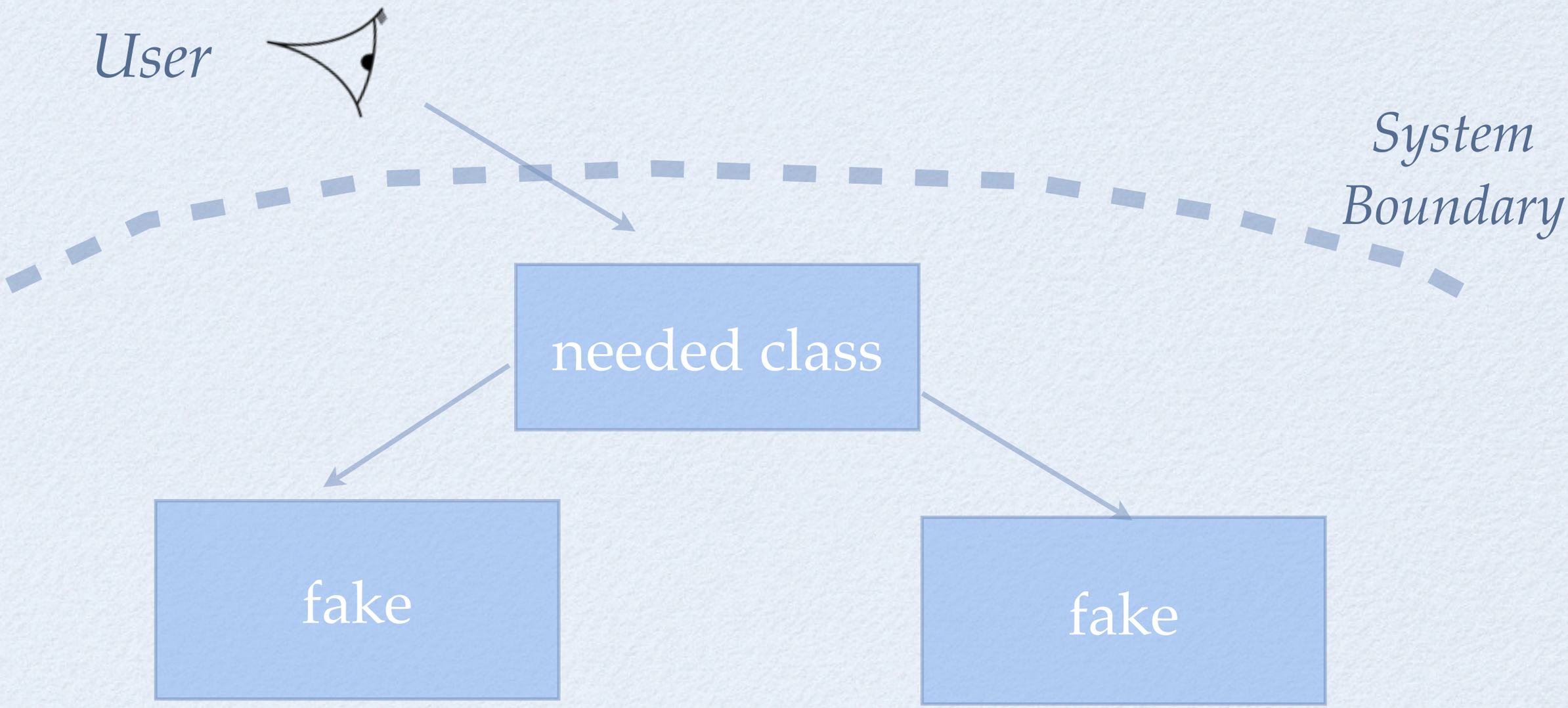
material copyright Emily Bache consulting, emily@bacheconsulting.com

Classic TDD Outside-In



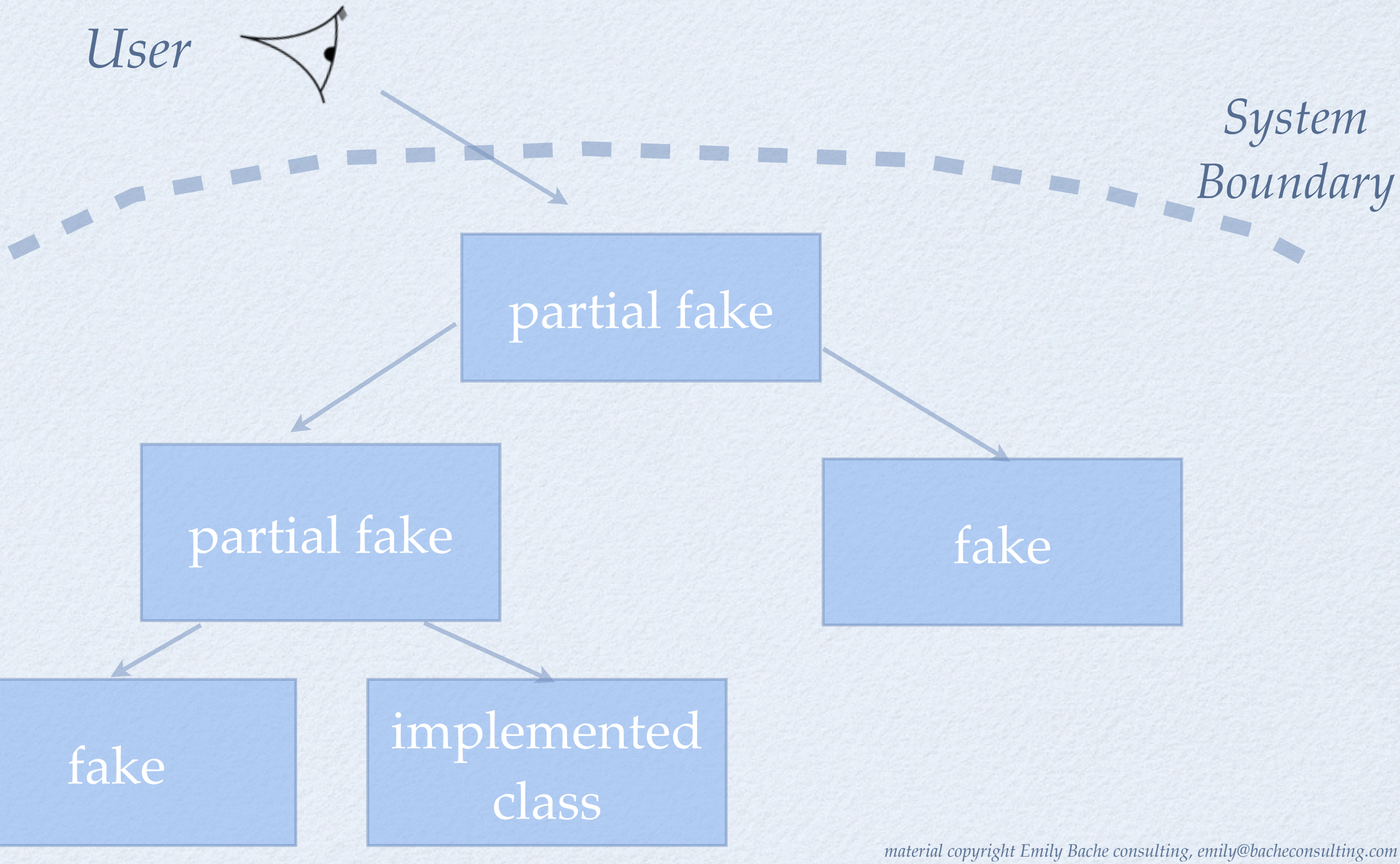
material copyright Emily Bache consulting, emily@bacheconsulting.com

Classic TDD Outside-In



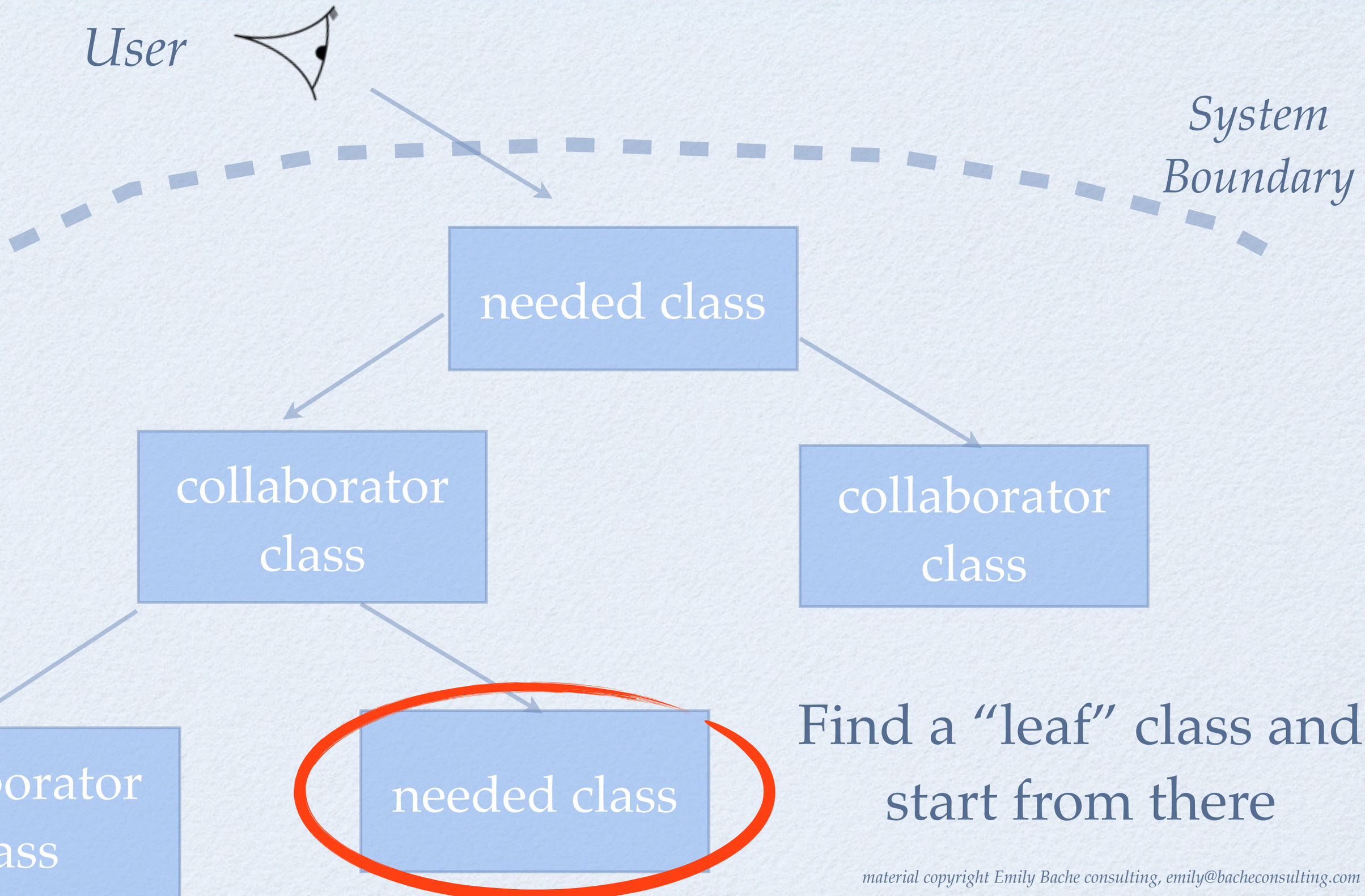
material copyright Emily Bache consulting, emily@bacheconsulting.com

Classic TDD Outside-In



material copyright Emily Bache consulting, emily@bacheconsulting.com

Classic TDD bottom-up



Find a "leaf" class and start from there

material copyright Emily Bache consulting, emily@bacheconsulting.com

Let me follow your tests

- Use github - fork my project and push to your user
- OR Keep the tests in a file in the order you wrote them
  - Don't delete tests, set @Ignore on them or comment out

material copyright Emily Bache consulting, emily@bacheconsulting.com