

Development Process

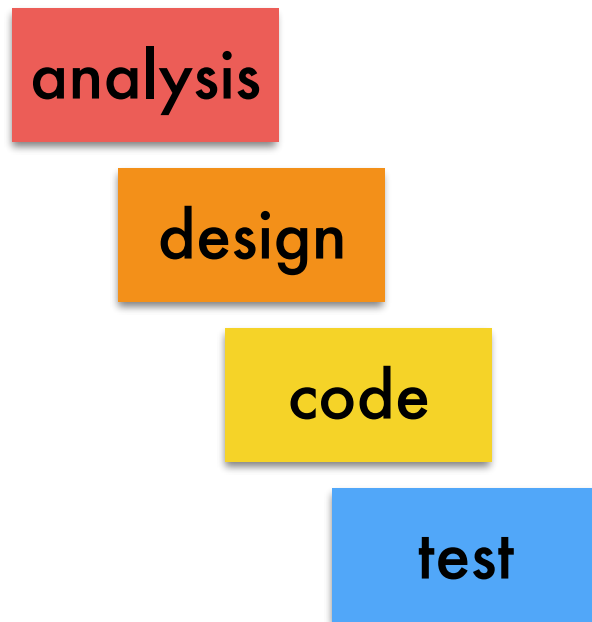
Chapter 2 of **UML Distilled** by Martin Fowler

Waterfall

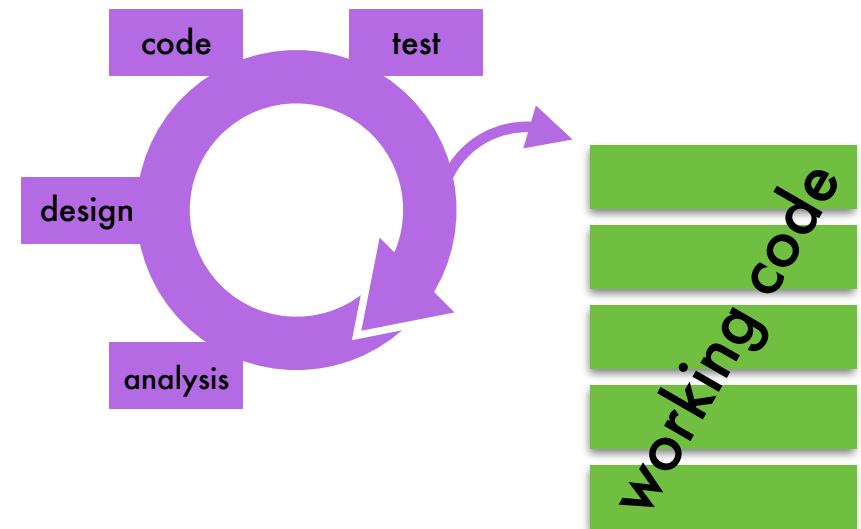
vs

Iterative

Waterfall breaks down a project based on activities...



Iterative breaks down a project based by subsets of functionality...



Each iteration does all activities, but delivers a subset of requirements

Iterative
development has come
under many names:
incremental, spiral,
evolutionary, and so
on...

These distinctions are
neither widely agreed upon
nor that important
compared to the iterative vs
waterfall dichotomy



Fowler
UML Distilled

You should use
iterative development
only on projects that
you want to
succeed.



Fowler
UML Distilled

Time Boxing

An iteration must be a fixed length of time

If you can't build all you intended:

- you **must slip some functionality** from the iteration
- you **can't slip the delivery date**



Rework

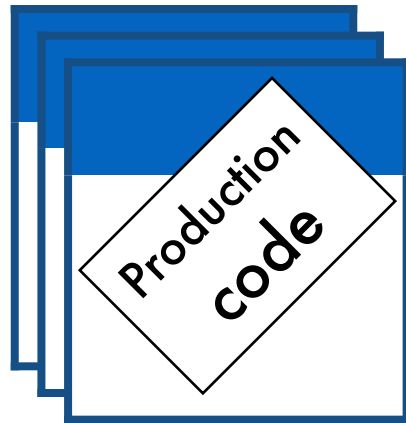
- Iterative development explicitly assumes you will be reworking and deleting existing code during later iterations

Managing Rework

- Automated regression tests
- Refactoring
- Continuous integration

Regression Testing

Automated tests suite used to quickly detect bugs when you change things



Rule of thumb:

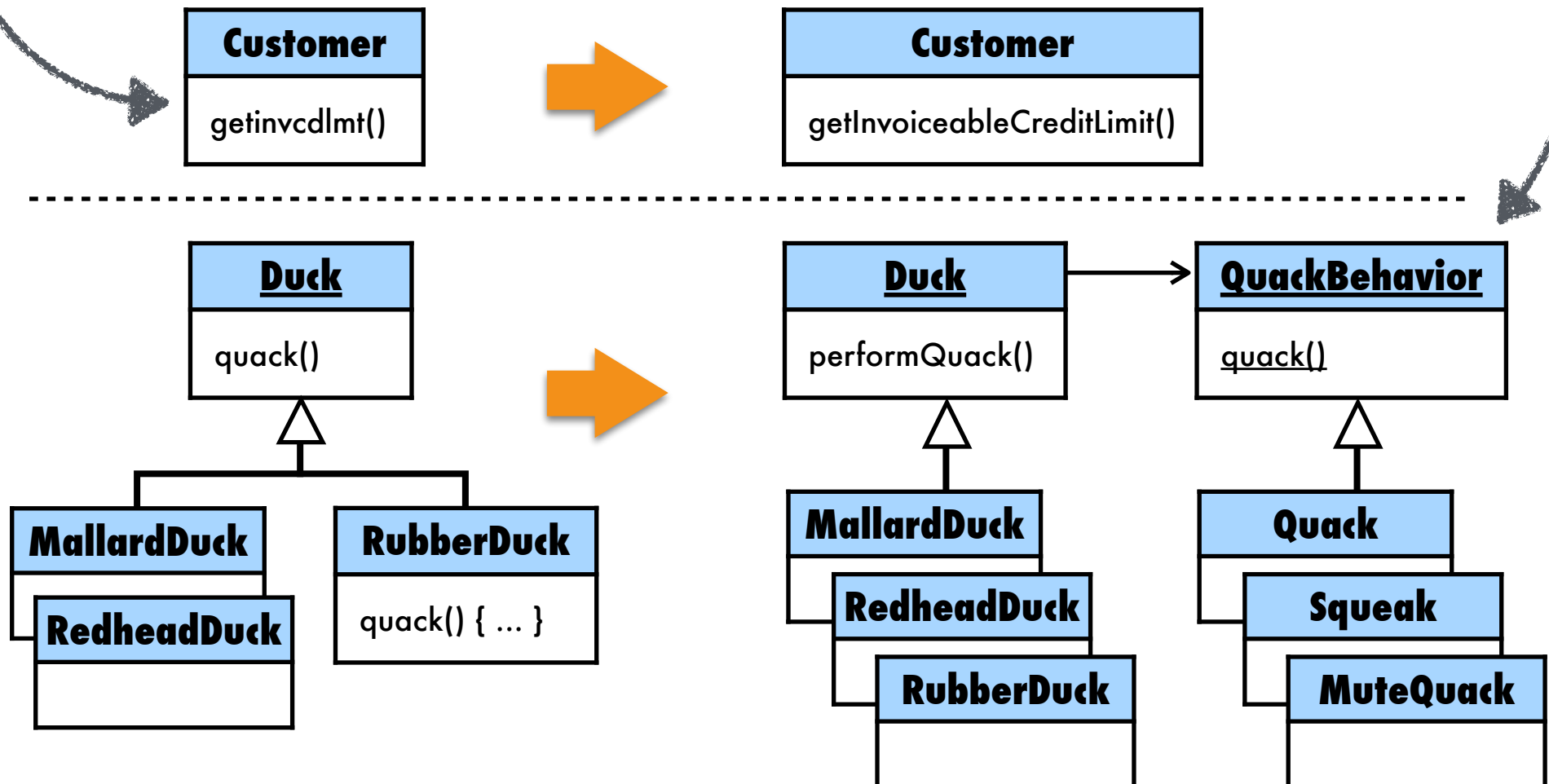
$| \text{production code} | = | \text{unit test code} |$

Simple example:
Rename method

Refactoring

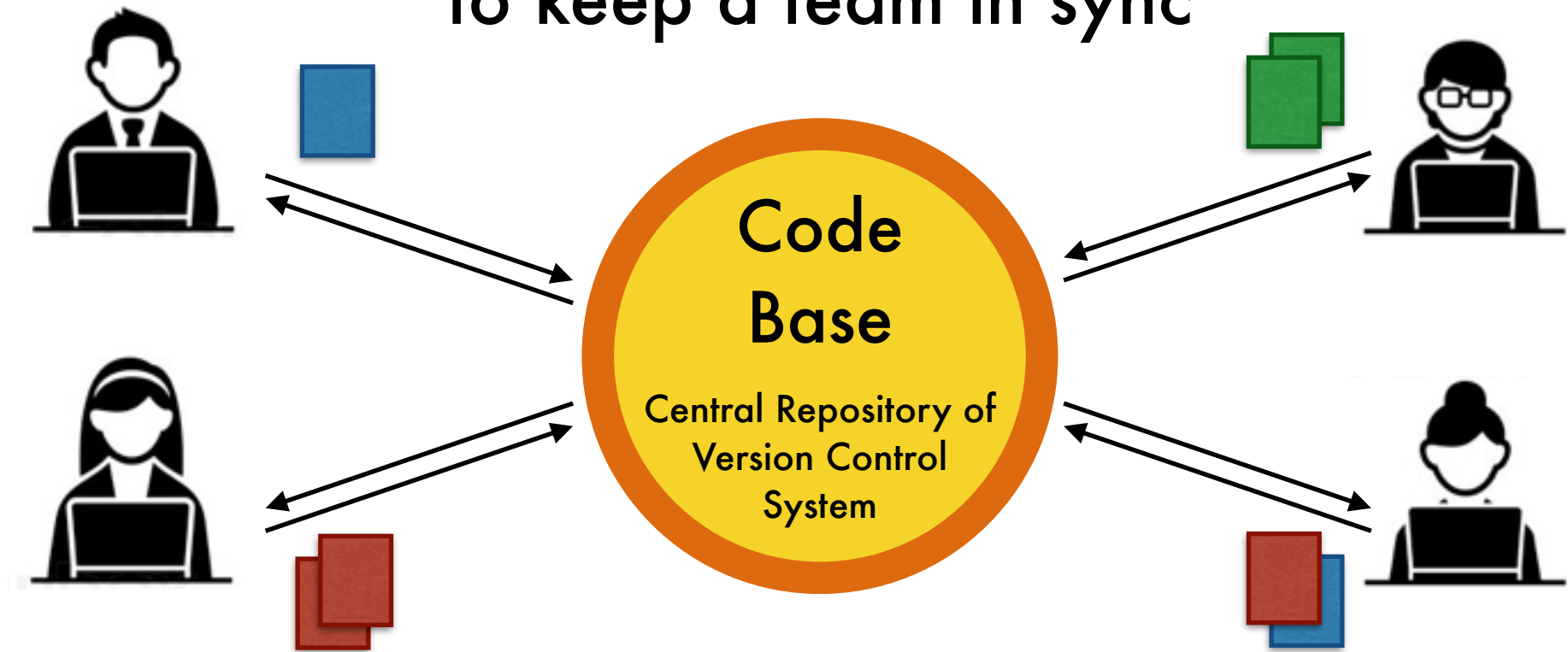
Complex example:
Introduce Strategy
Pattern

Strengthening design without changing behavior



Continuous Integration

Automated build process used
to keep a team in sync



Build process – including regression tests – is automatically run
when any team member checks code into central repository

Waterfall Problems

- Hard to tell if project is on track
- Tested, integrated software not produced
- Can't adapt to requirement changes

until
the end



but...

- Waterfall is still common in industry!

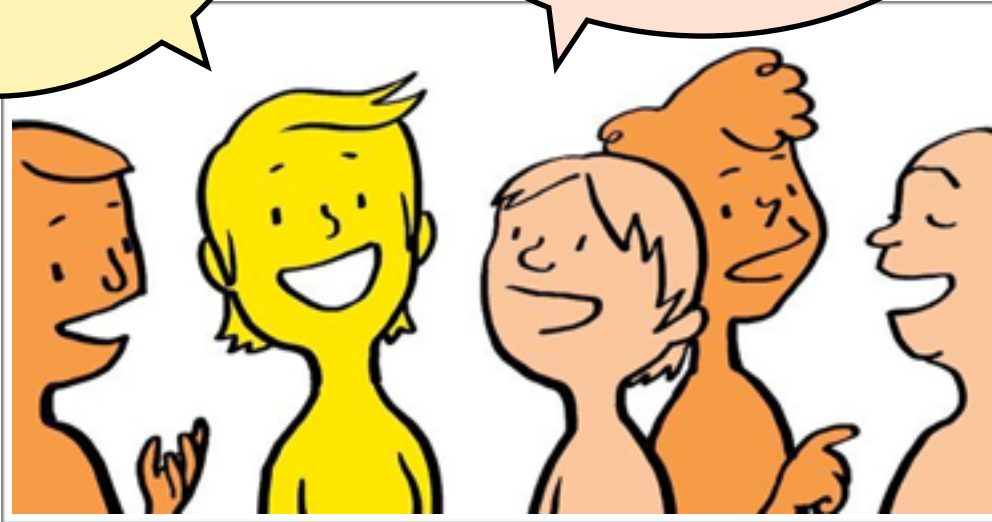
Pause and Think

Why is the waterfall model still
so common in industry?

Why is Waterfall so common?

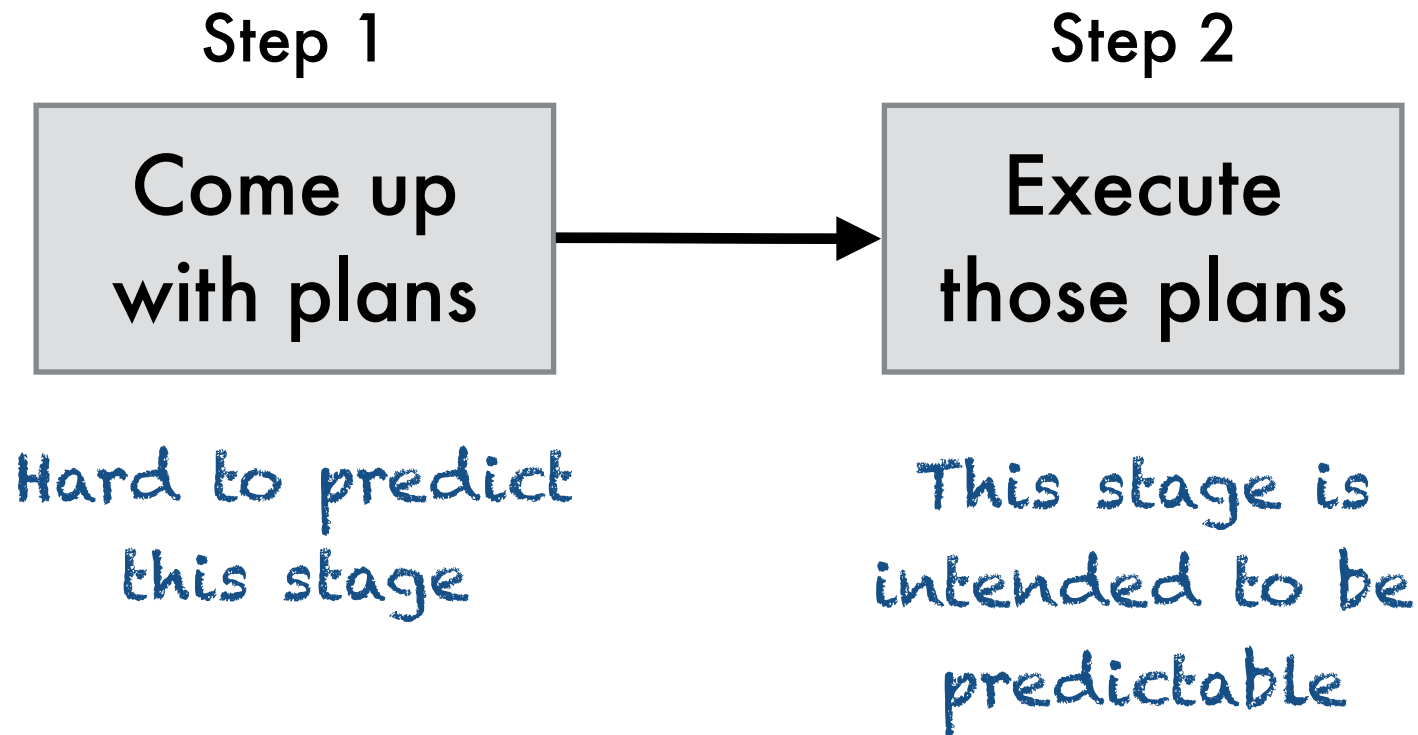
We are doing
one analysis
iteration followed
by two design
iterations!

This iteration's
code is very buggy,
but we'll clean it up
at the end.



People claim to be using iterative
development when they are not

Predictive Planning



Nothing is more frustrating than not having a clear idea how much it will cost to build some software and how long it will take to build it.

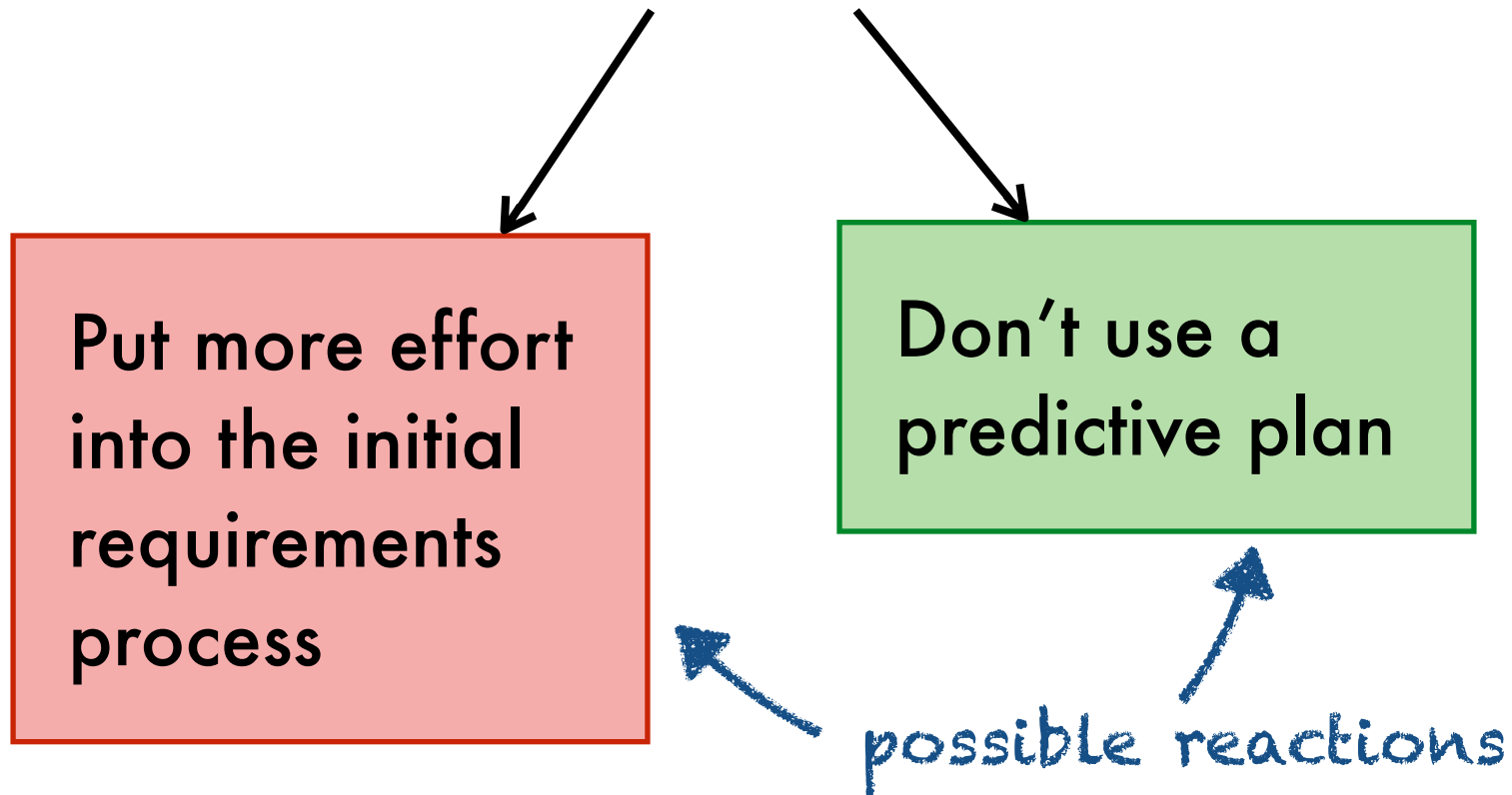
However, there is a considerable debate about whether many software projects can ever be predictable.



Fowler
UML Distilled

Requirements churn

Changes in requirements in the later stages of the project



Adaptive Planning

- Predictability is seen as an illusion
- Treats change as a constant
- Project is controllable, not predictable
- Nothing goes “according to plan”
- Can fix budget and delivery time, but not delivered functionality