

Recognizing the Dangers of Over-abstraction



Richard Warburton

@richardwarburton www.monotonic.co.uk



Module Contents

False Abstractions

Poor Naming

Single
Implementation

Incomplete
Implementation

YAGNI



False Abstraction





Every abstraction has a cost

Goal: Improving our codebase

- The benefits must outweigh the cost

Principle driven judgement



What Are the Costs?

**Hard to find
+ read code**

Indirection, Proliferation
of Concepts

**Hard to
evolve code**

Need to change
multiple places

More code

Rule of thumb: less
code is better



False Abstraction

An interface or abstract class that should not exist



Poor Naming

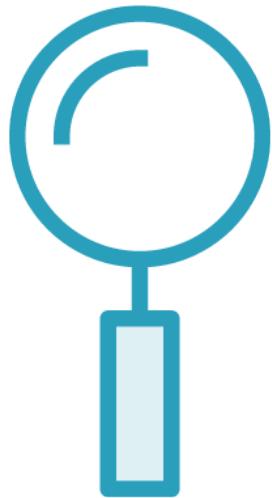


“There are two hard things in Computer Science: cache invalidation and naming things.”

Phil Karlton



Issues with Naming



Discovery

Difficult to find code with lots of
similarly named classes



Indicative of Unclear Thinking

Heuristic for other problems



Example of Poor Naming

ClientRepository

ClientRepositoryImpl

IClientRepository

ClientRepository

ClientRepositoryInterface

ClientRepository



Example of Good Naming

List

ArrayList

ClientRepository

SqlClientRepository



Single Implementation



Single Implementation

Anti-pattern where an abstract class or interface only has a single implementing class





Incomplete Implementation



Incomplete Implementation

Anti-pattern where an interface or abstract class's abstract methods are often not implemented. Eg: empty method bodies or UnsupportedOperationException



Problems with Incomplete Implementation

Abstraction Isn't Solid

Is it really modelling
something in your
domain or app?

Difficult to Evolve

Often end up with more
incompleteness in future



YAGNI

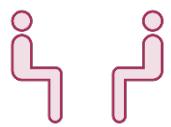


YAGNI

You Ain't Gonna Need It.



YAGNI



Originally an Agile/XP Principle



Don't add abstractions until they are needed



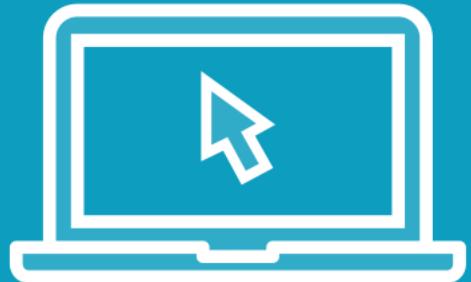
Do the simplest thing that could possibly work



Continuously refactor to Evolve/Add/Remove abstractions



Demo



File Type Detection and Extraction

Refactor adding an abstraction

More Complex Client Engagement

Abstract over ClientEngagements

**Demonstrate how this makes things
more complex**



Summary



Learned about the dangers of over-abstraction

- False abstractions
- Poor naming
- Single or incomplete implementations
- YAGNI

Similar to premature optimization

