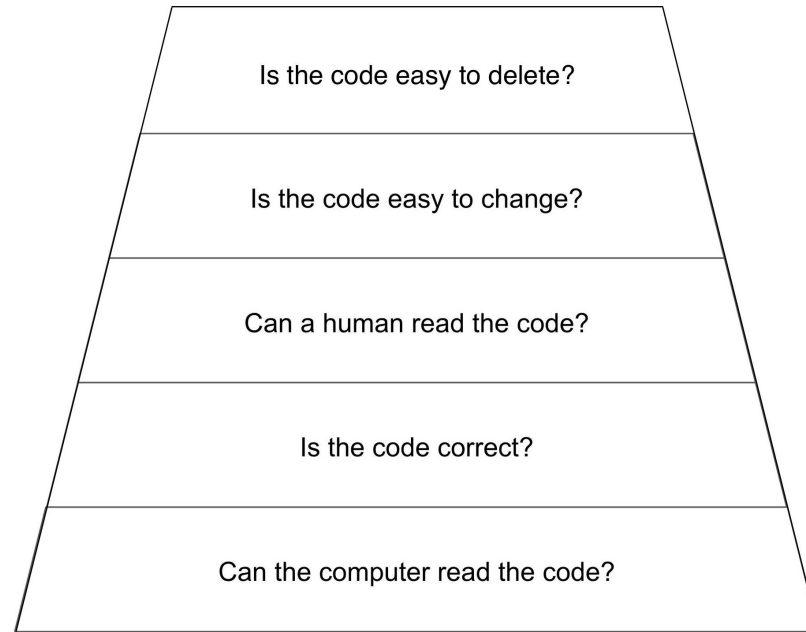


Pipelines, Wrappers, Middlewares? And Deletable Code

A dramatic interpretation and retelling of
<https://kellysutton.com/2017/05/29/deletability.html>
And some patterns of deletable code

Code that easy to delete is high quality code

Few Dependencies
Single Responsibility
Probably easy to Reuse



Fundamental Friction

Small Files are Worse

1. Small files have less code
2. Small files have less context
3. Small files require more navigation
4. Small files require more effort to work with
5. Small files make it harder to understand the system
6. Small files are worse

Small Files are Better

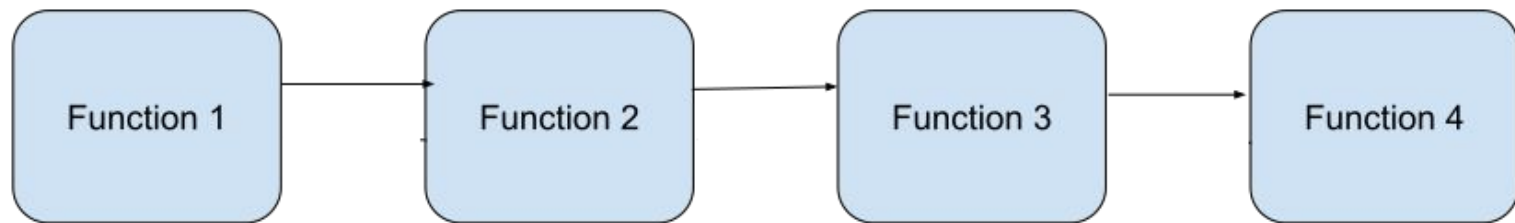
1. Small files have less code
2. Small files have fewer responsibilities
3. Small files are simpler
4. Small files are more modular
5. Small files are more easily deleted
6. Small files are better

TDD Design Pressure

Easy to test correlates to system being modular

Testing behavior not implementation

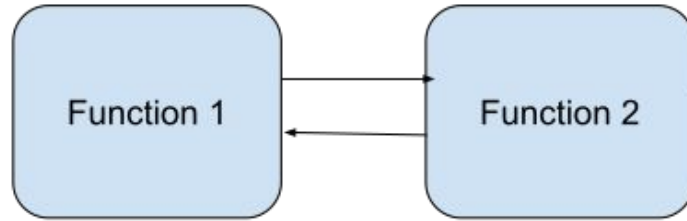
Pipeline



When to use pipelines

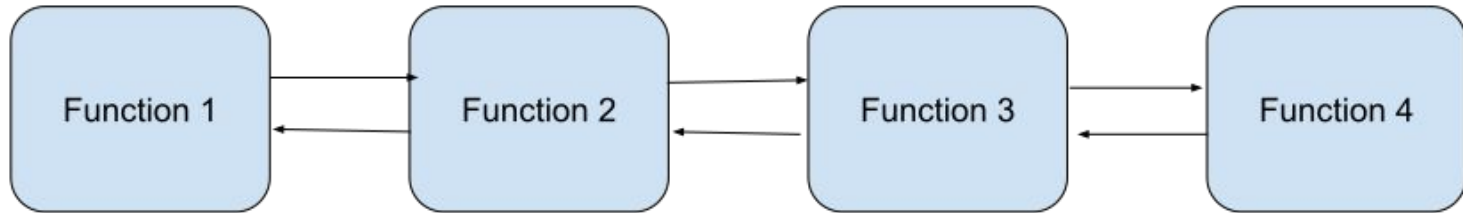
- ETL Jobs
- Reports
- Multi step processing
- Dry runs or staging operations
- Everywhere?

Wrapper



Middleware

Middleware Stack



When to use wrappers/middleware

API boundaries

Transactions

Logging

Usually not business logic