Common Techniques for Testing Code

Jason Olson
Staff Software Engineer

@jolson88 www.jolson88.com

Overview

Dependency injection

Test doubles

Best practices

Getting a little taste for TDD techniques...

Dependency Injection

The Testing Problem

```
class Greeter {

    nameService: NameService;

    constructor() {
        this.nameService = new NameService();
    }

    sayHello(): string {
        return `Hello, ${this.nameService.getName()}`;
    }
}
```

Dependency

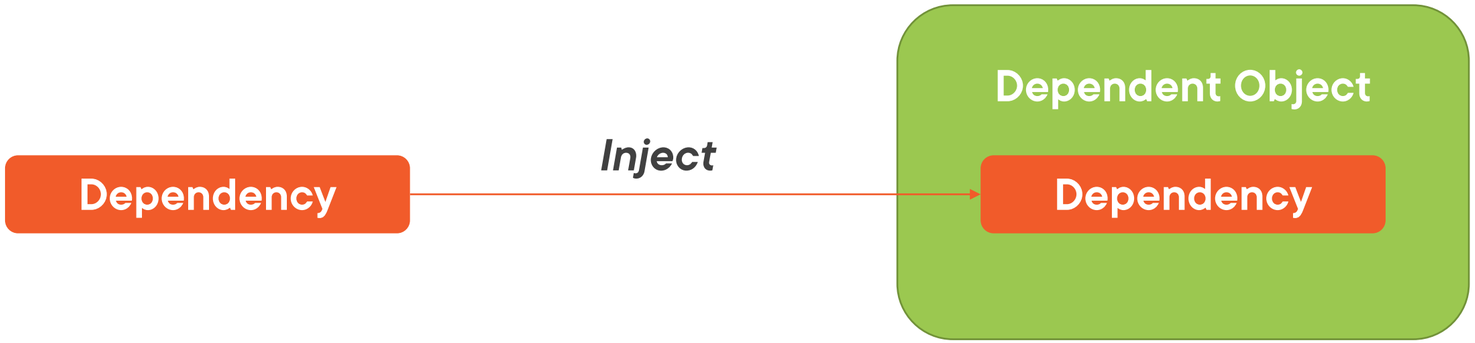
What if we want this to fail?

What Is Dependency Injection?

In software engineering, dependency injection is a technique whereby one object supplies the dependencies of another object.

- A *dependency* is an object that can be used.

- An *injection* is the passing of a dependency to a dependent object that would use it.



Wikipedia - https://en.wikipedia.org/wiki/Dependency_injection

Dependency Injection Mechanisms

Constructor (or Method) Injection

Providing dependencies through a class constructor or method.

Property/Setter Injection

Using a property or setter method to inject a dependency.

Constructor Injection

```
class Greeter {
  nameService: NameService;

  constructor(nameService) {
    this.nameService = nameService;
  }

  sayHello(): string {
    return `Hello, ${this.nameService.getName()}`;
  }
}
```

Test

```
class BadNameService {
  getName(): string {
    throw new Error('Could not get name');
  }
}
```

```
var service = new BadNameService();
var greeter = new Greeter(service);
expect(() => greeter.sayHello()).toThrow();
```

Is Dependency Injection Only for Object-oriented Programming?

```
function getName() {
  return "Jason";
}

function greet() {
  return `Hello, ${getName()}`;
}

greet();
//=> Hello, Jason
```

```
function greet(getName) {
  return `Hello, ${getName()}`;
}

greet(() => {
  return "Jason";
});
//=> Hello, Jason
```

Higher-Order Functions

Trade-offs of Dependency Injection

Self-contained Code

Dependency-injected Code

Easier to Understand

Harder to Understand

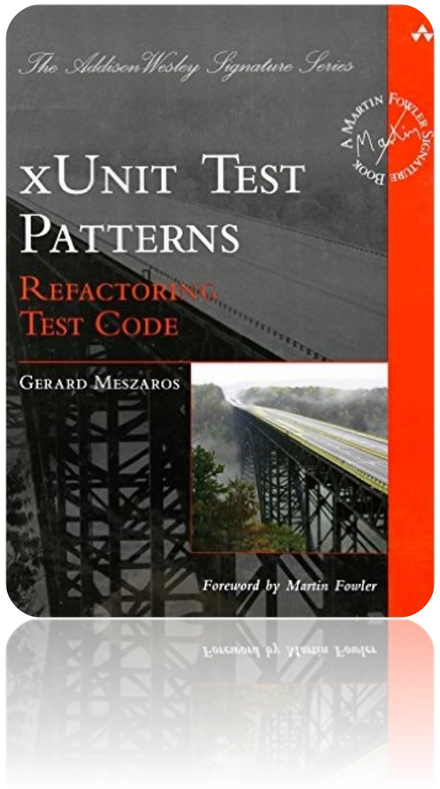
Harder to Test

Easier to Test

Test Doubles

Test Doubles

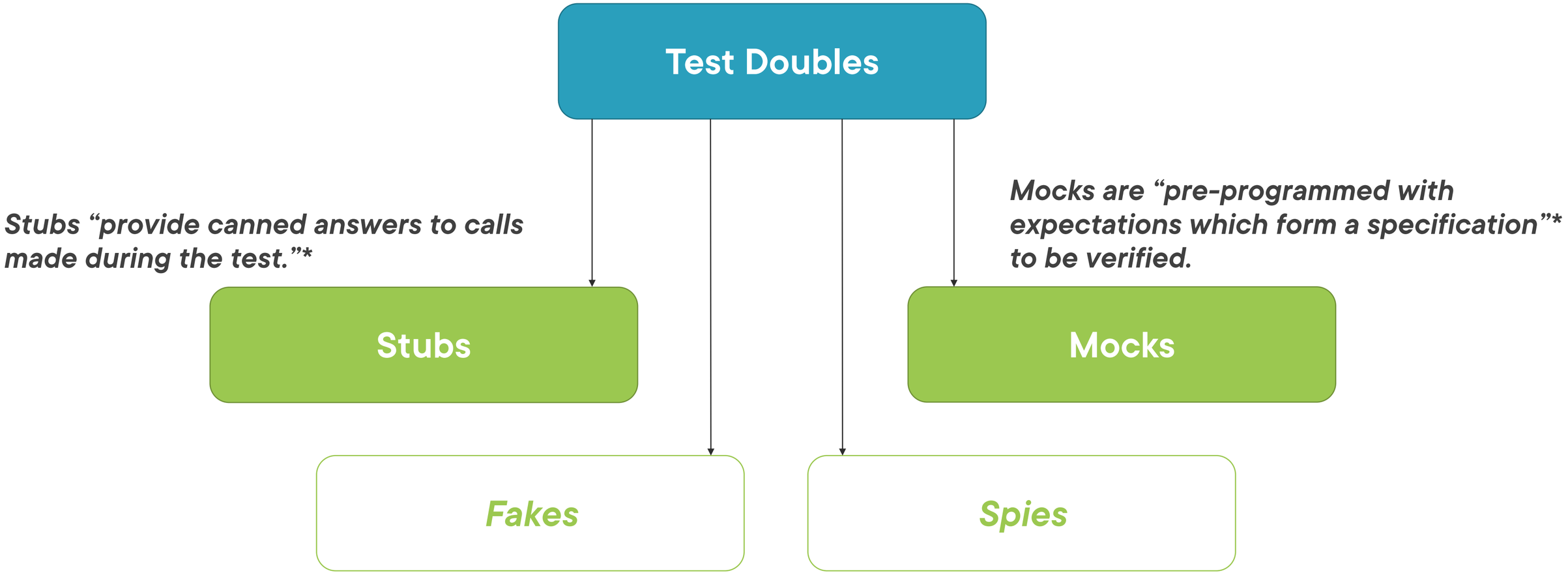
Gerard Meszaros



Test Double is a "generic term for any kind of pretend object used in place of a real object for testing purposes."

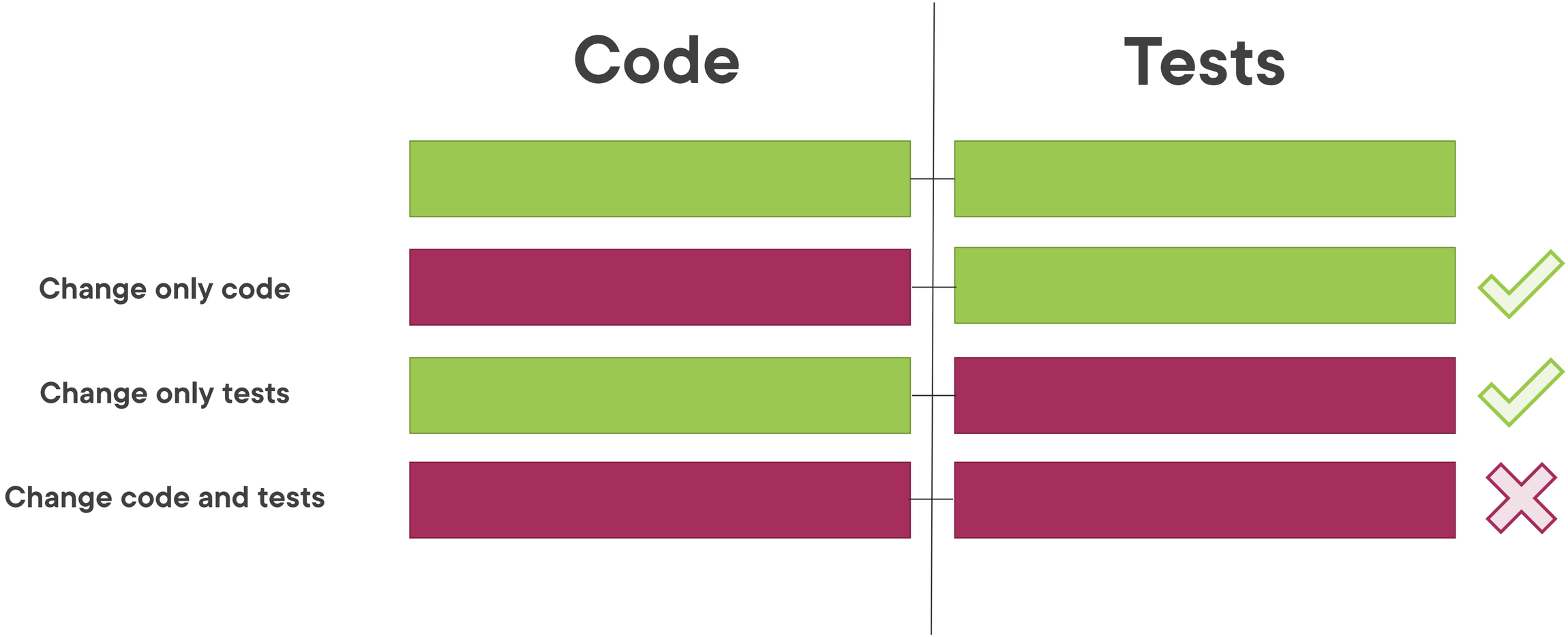
Martin Fowler
martinfowler.com/articles/mocksArentStubs.html

Types of Test Doubles



* martinfowler.com/articles/mocksArentStubs.html

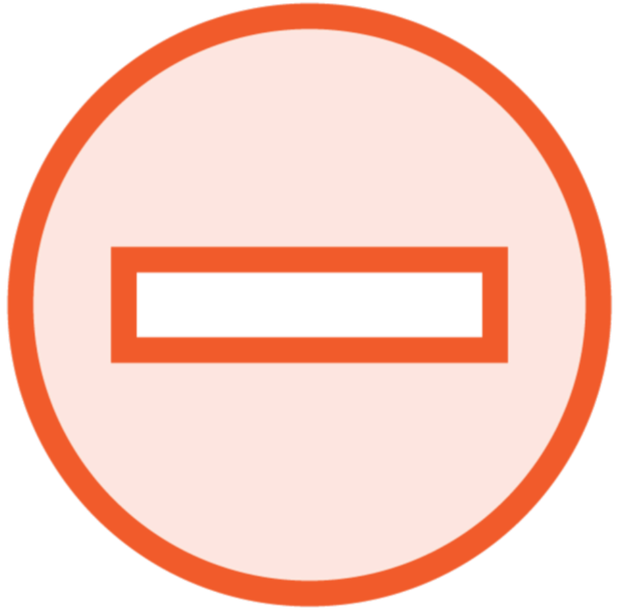
Brittle Code



Testing Best Practices

Treat Test Code Like Production Code

Write readable and maintainable test code



Address both positive and negative test cases

Separate common set-up and teardown logic

Focus Only on Necessary Values and Results

Review Tests and Test Practices with Team

Effective techniques

Catching bad habits

Common challenges

Leave the Code Better Than You Found It

Summary

Dependency injection

Test doubles

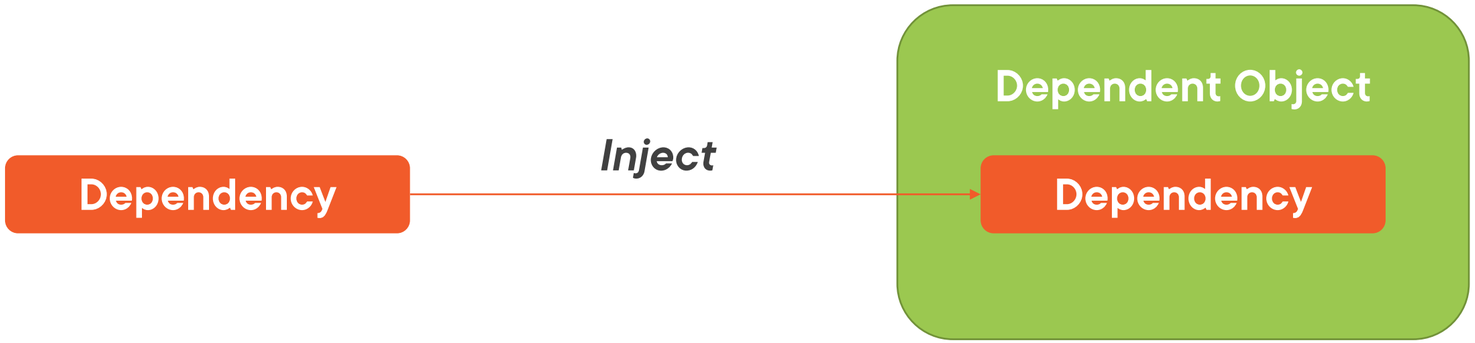
Best practices

What Is Dependency Injection?

In software engineering, dependency injection is a technique whereby one object supplies the dependencies of another object.

- A *dependency* is an object that can be used.

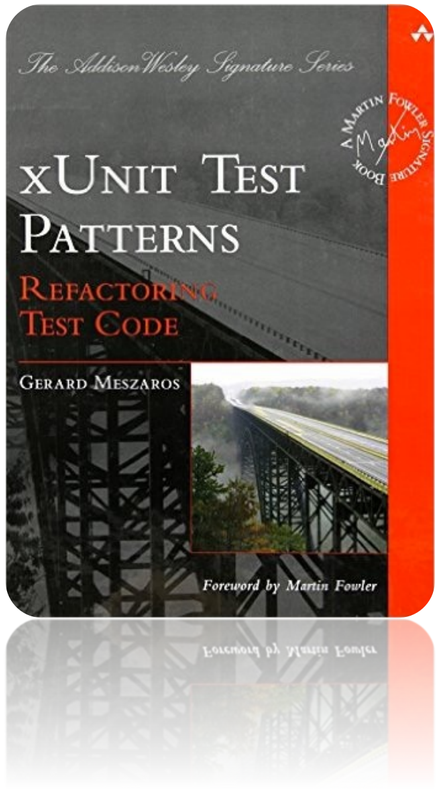
- An *injection* is the passing of a dependency to a dependent object that would use it.



Wikipedia - https://en.wikipedia.org/wiki/Dependency_injection

Test Doubles

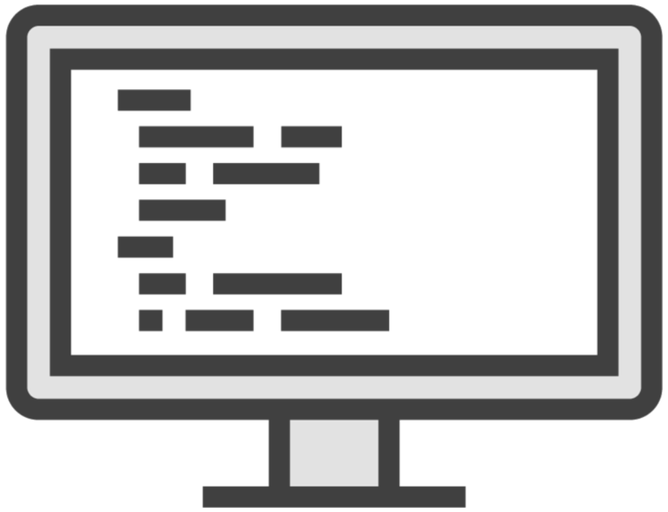
Gerard Meszaros



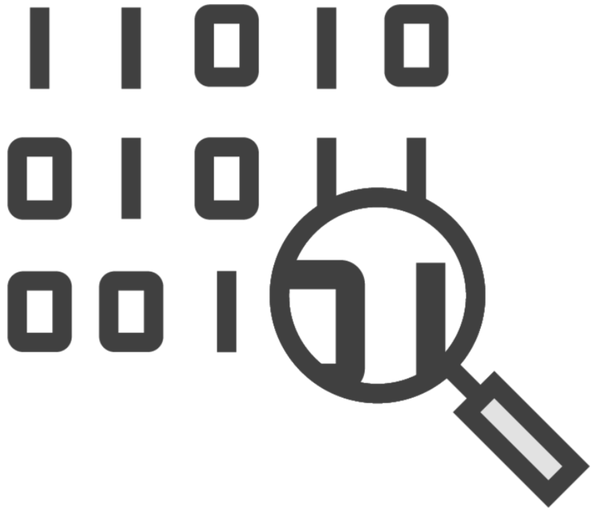
Test Double is a "generic term for any kind of pretend object used in place of a real object for testing purposes."

Martin Fowler
martinfowler.com/articles/mocksArentStubs.html

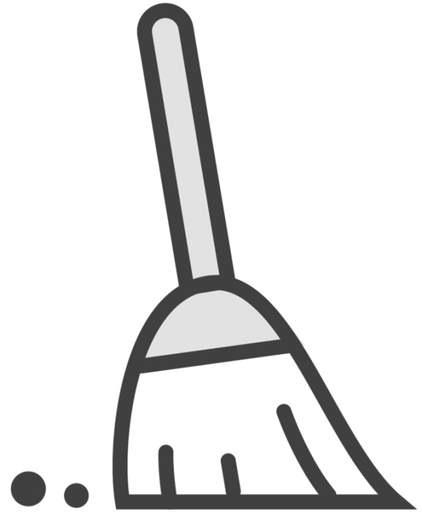
Best Practices



Treat test code like production code



Focus only on necessary values and results



Leave the code better than you found it