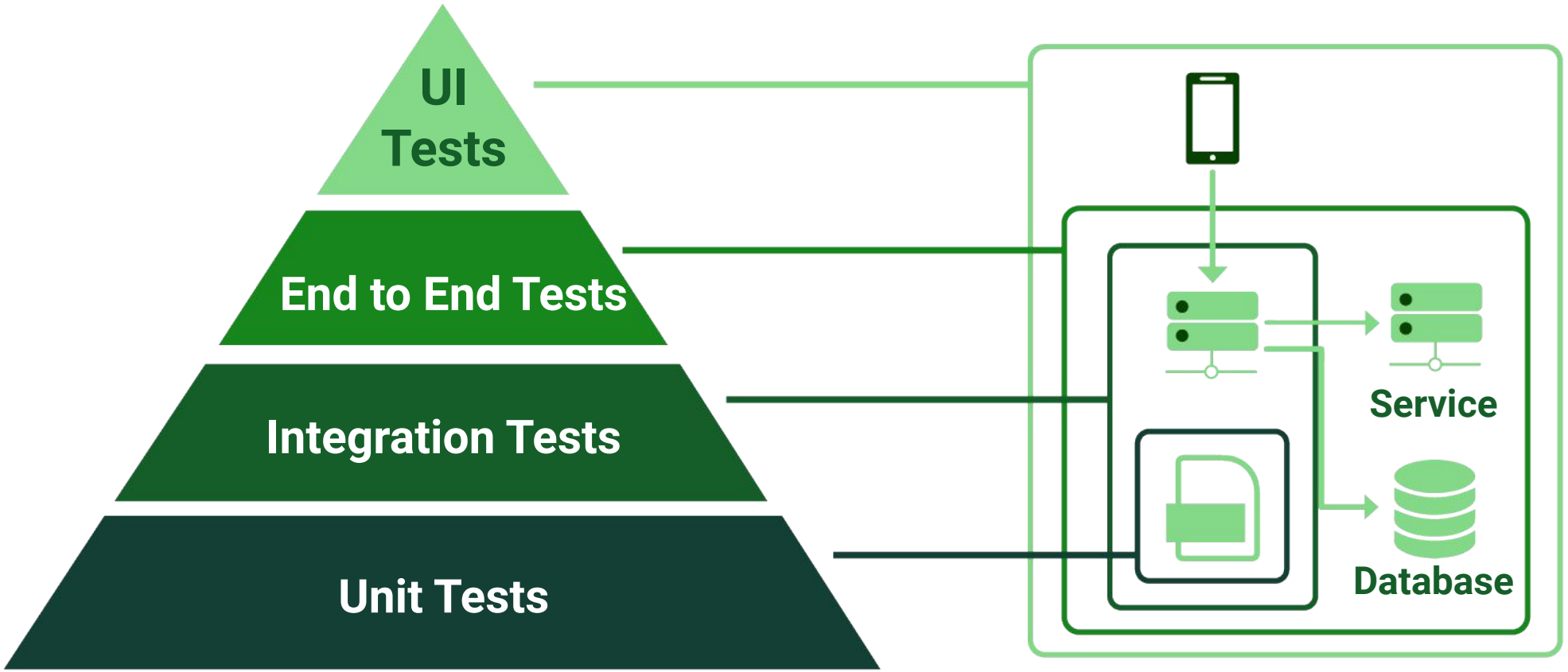
FUNDAMENTOS DE TESTING

#NuncaParesDeAprender

JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

Nicolas Molina

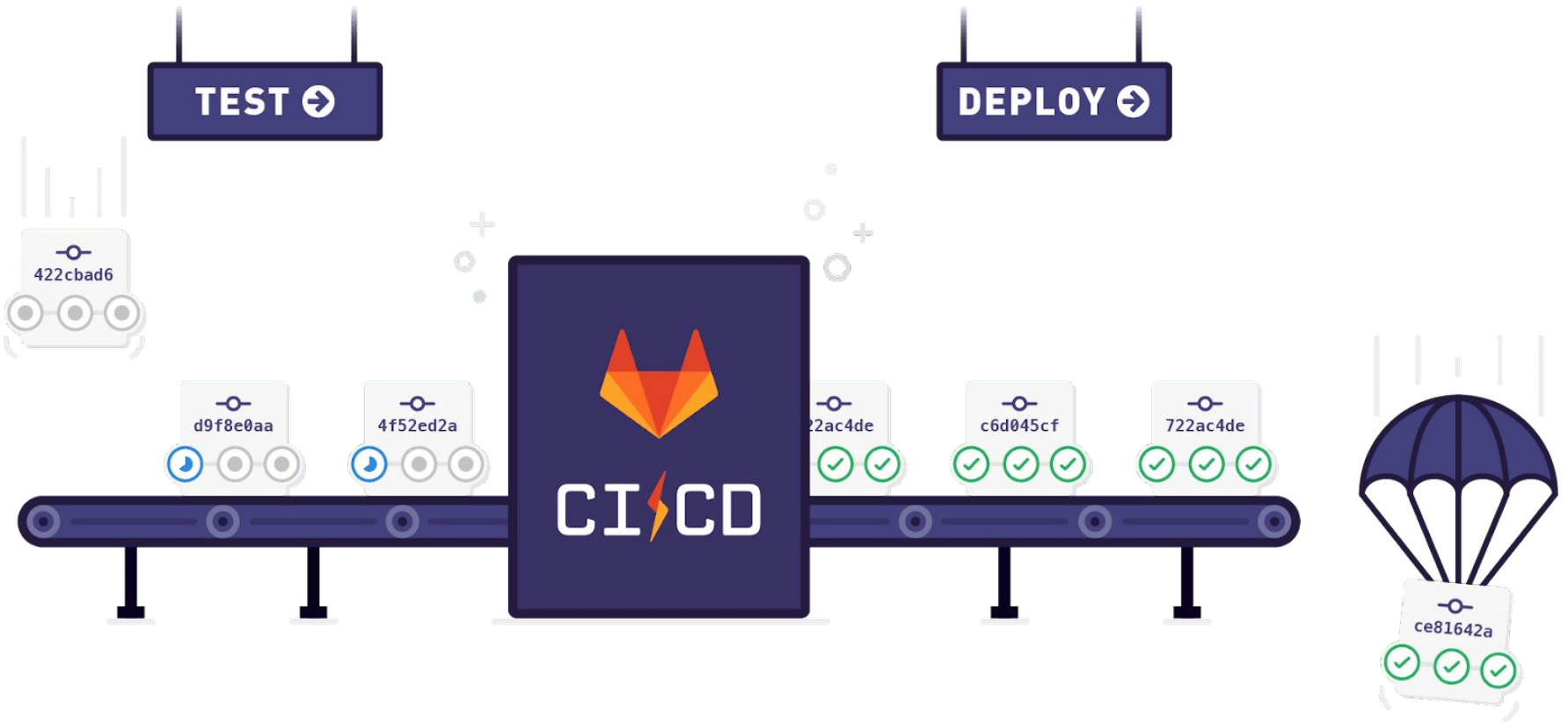
@nicobytes

Google Dev Expert

¿QUÉ ES EL TESTING?

Definamos el concepto.

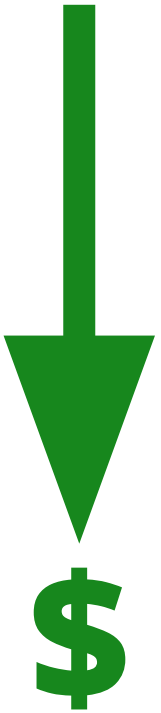
JS


TEST
DEPLOY
422cbad6
d9f8e0aa
4f52ed2a
c6d045cf
722ac4de
ce81642a
CI/CD

JS

¿Por qué?

- Diseño
- Desarrollo
- Pruebas
- Producción



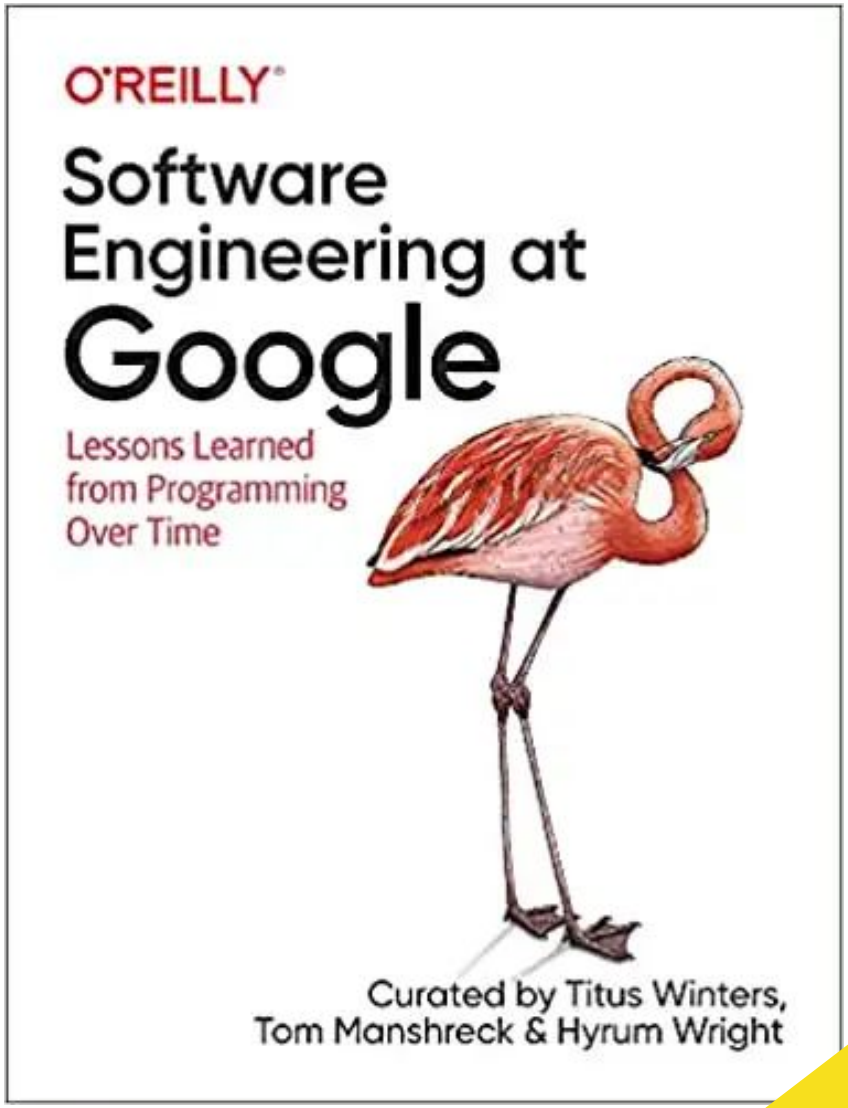
Demostrar que **el software funciona.**

Demostrar que **el software funciona.**

Los problemas siempre van a existir, lo que hacemos es gestionar el **riesgo**.

JS

“The earlier you find a mistake, ***the easier it is to fix****”*.



*“**Static analysis** runs in your editor. Finds typos, incorrect function calls, autocompletes code”.*

*“**Unit tests** take a few seconds to verify your code does what you think it does”.*

*“**Integration tests** take a few minutes to validate your system works. May catch fun edge cases”.*

JS

2 Unit tests 0 Integration tests

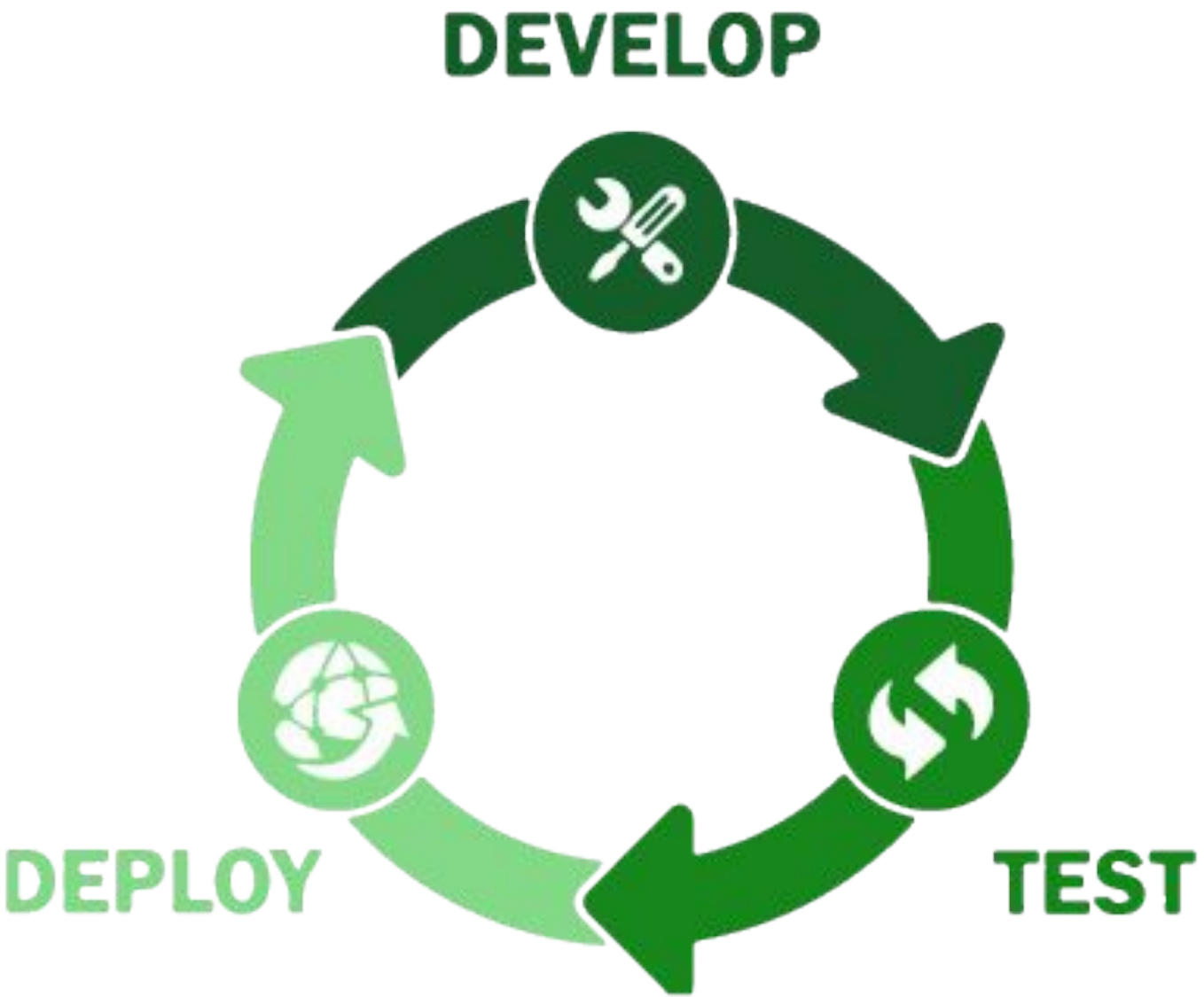
*“**Code review** takes a few hours to validate you're following standard norms and practices of your team”.*

*“**QA** takes a few hours or days to ensure everything works together as expected”.*

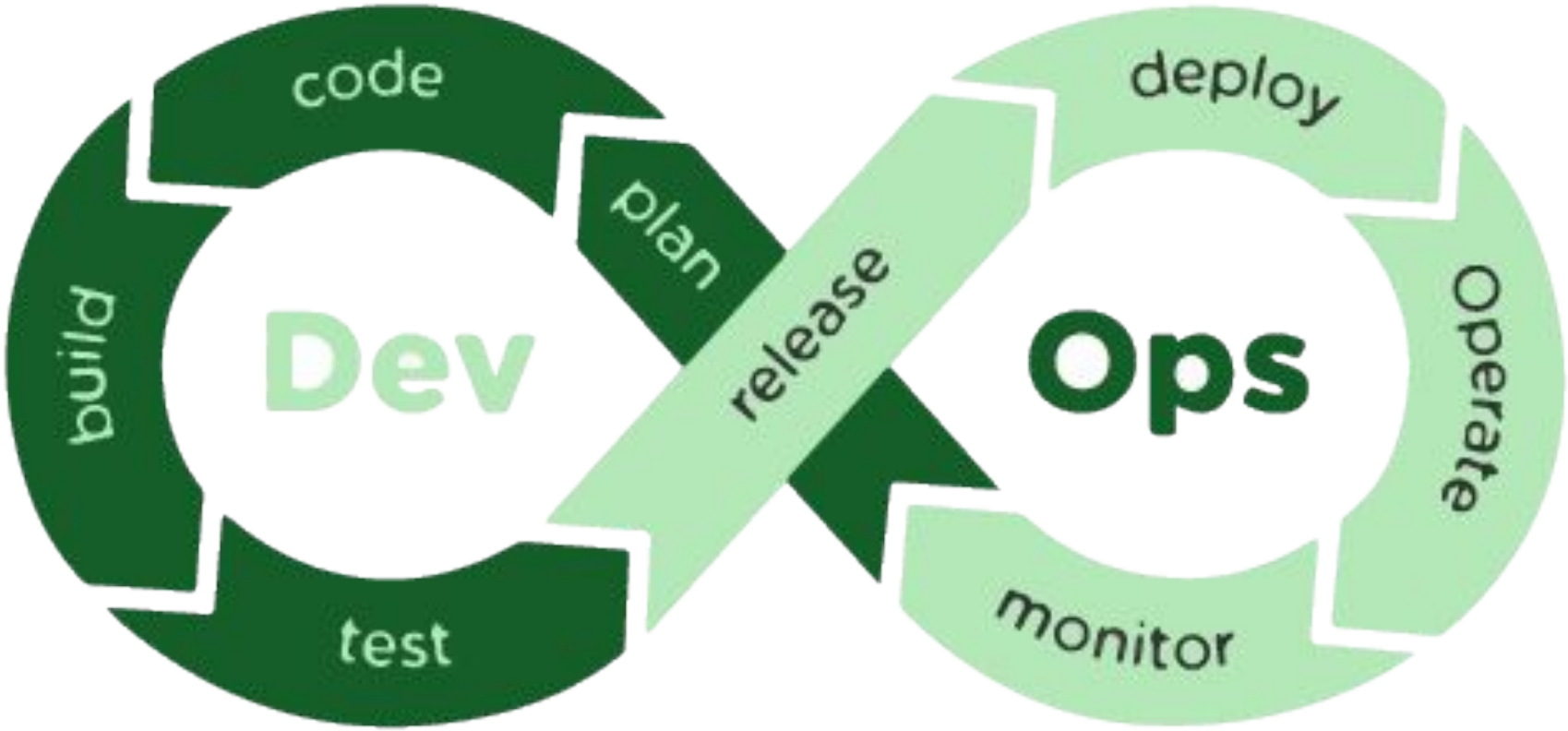
Demostrar que **el software funciona**

Reducir el riesgo

JS



JS

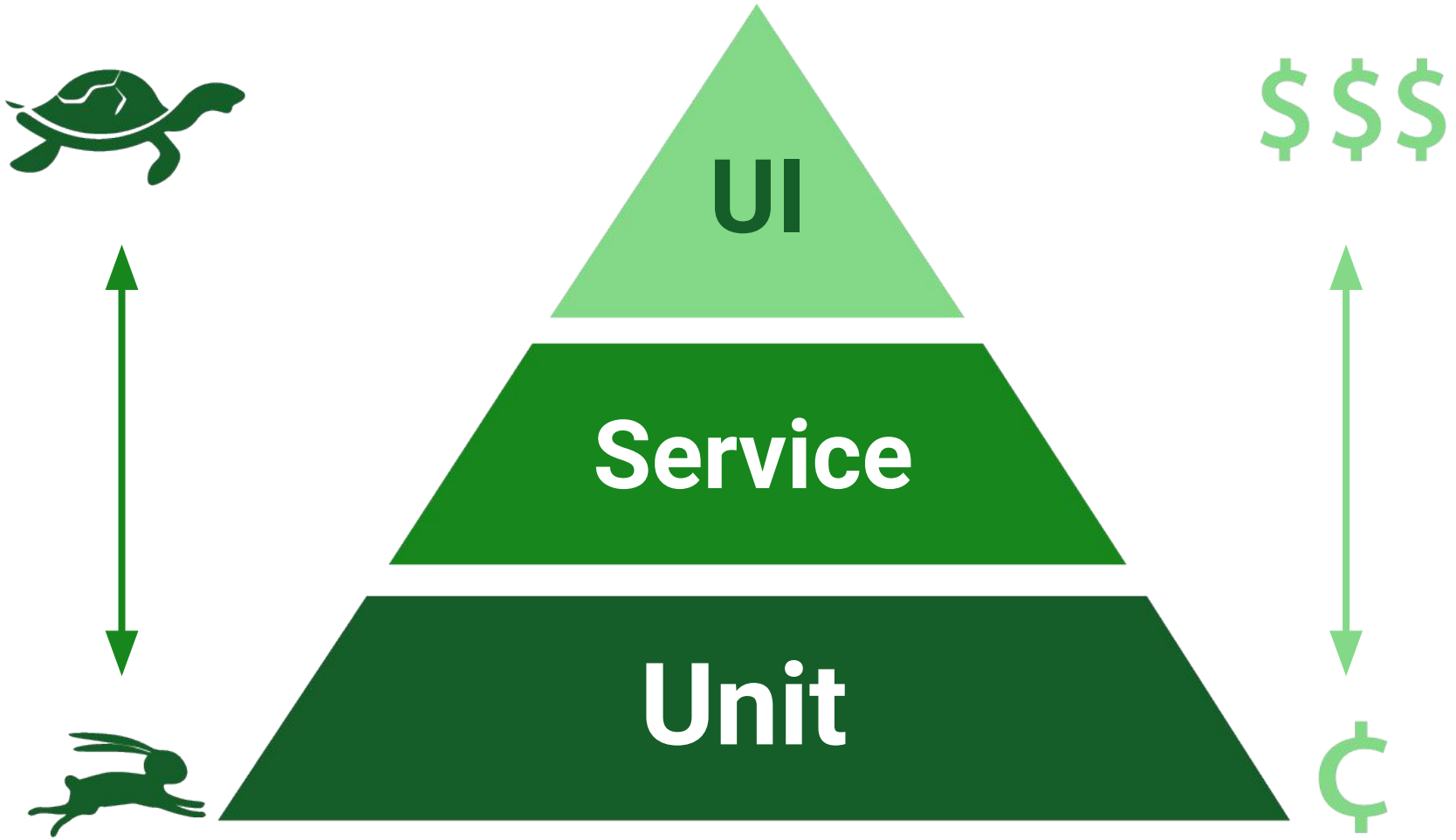

code
plan
build
Dev
test
release
deploy
Ops
operate
monitor

Un código que no está bien creado es complejo de probar

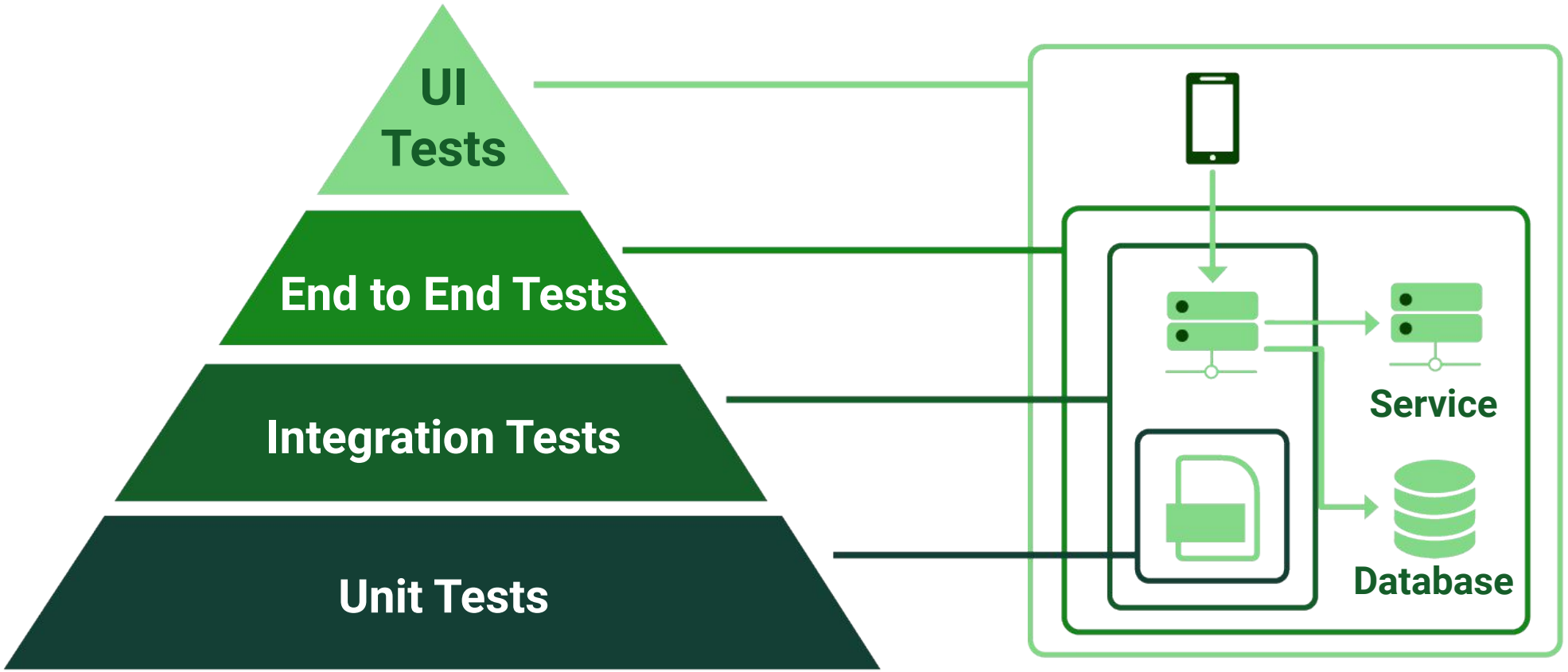
LA PIRÁMIDE

Clasificación y cantidad

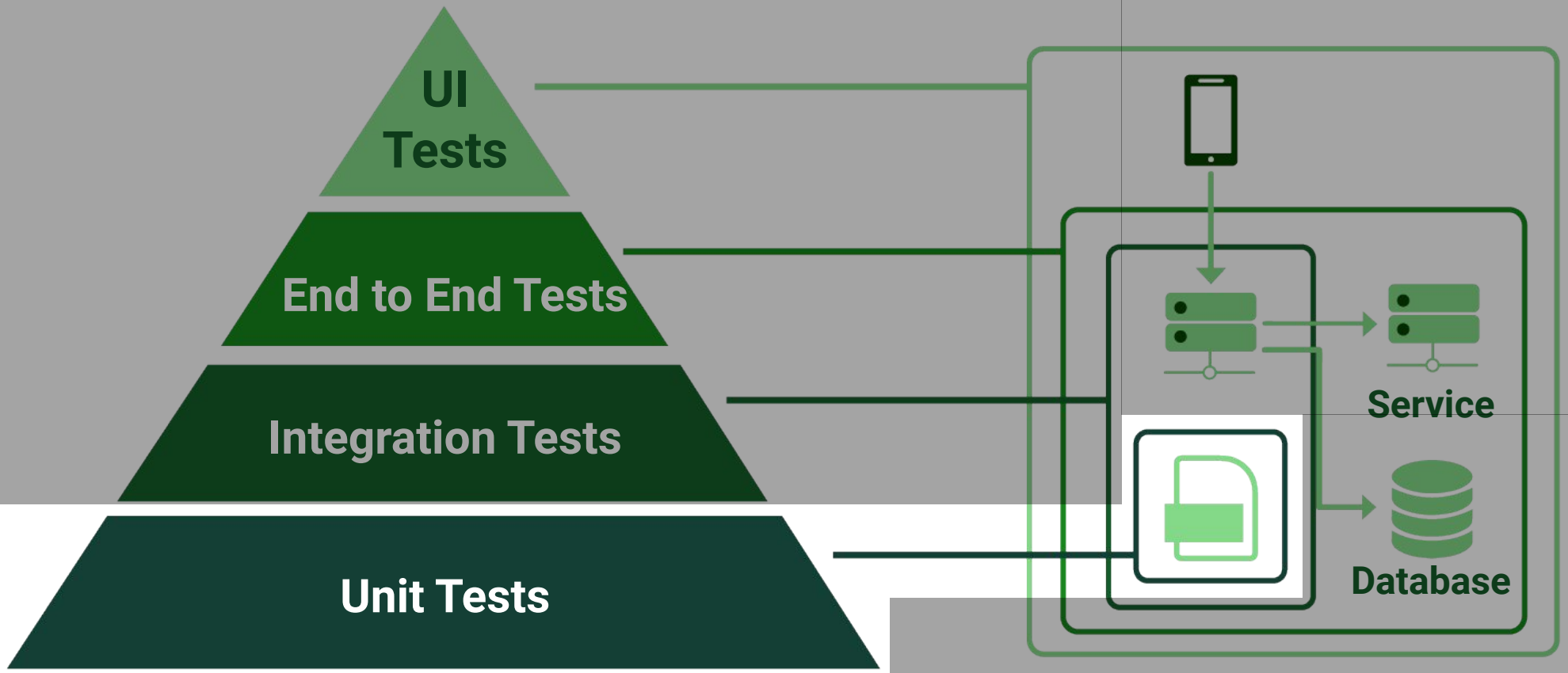
JS



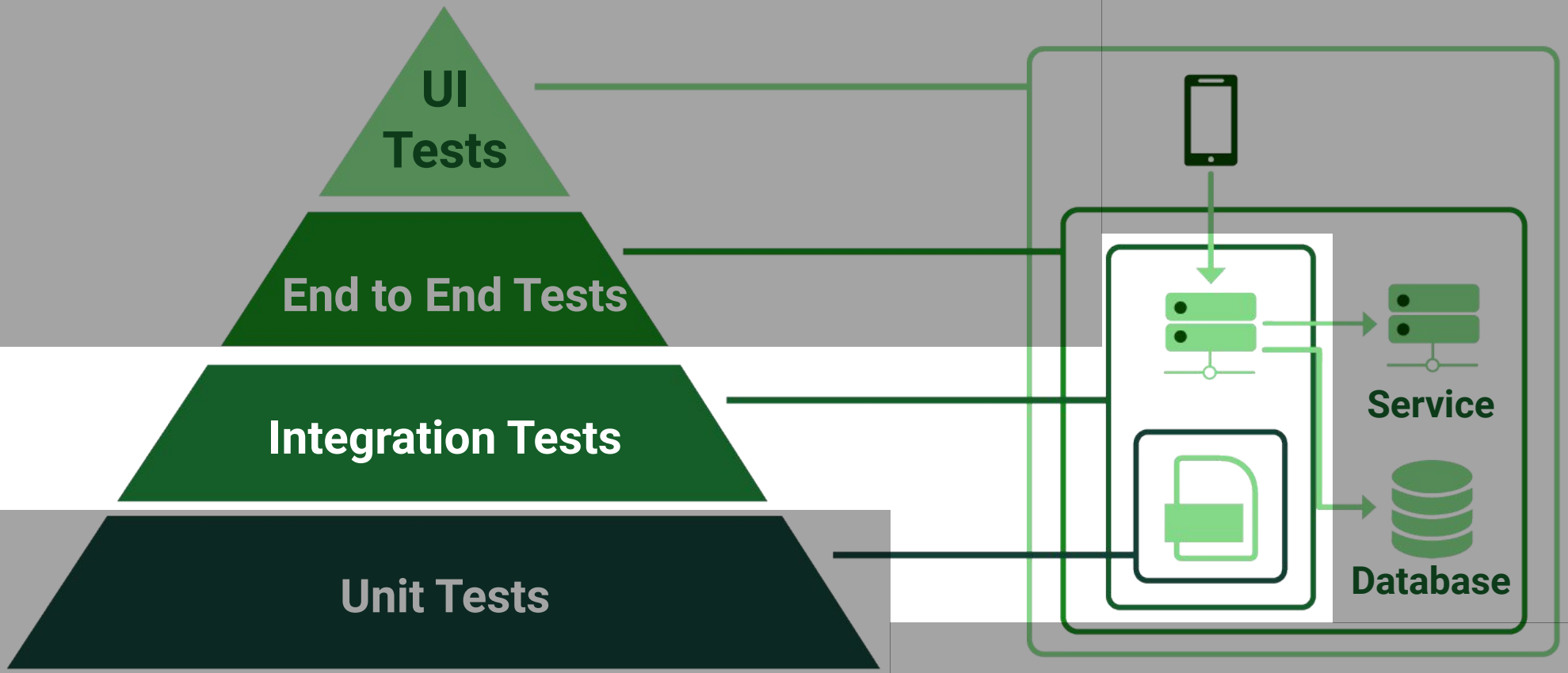
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

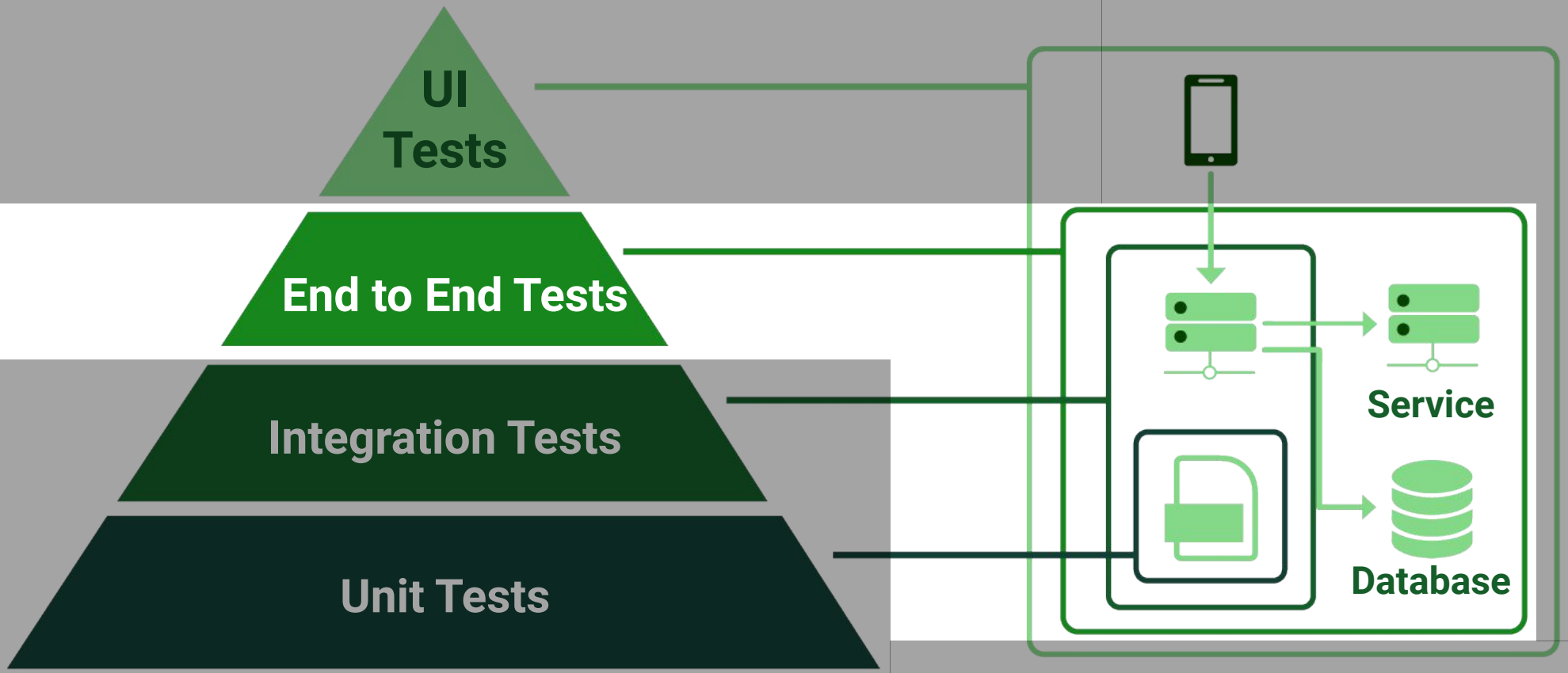
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

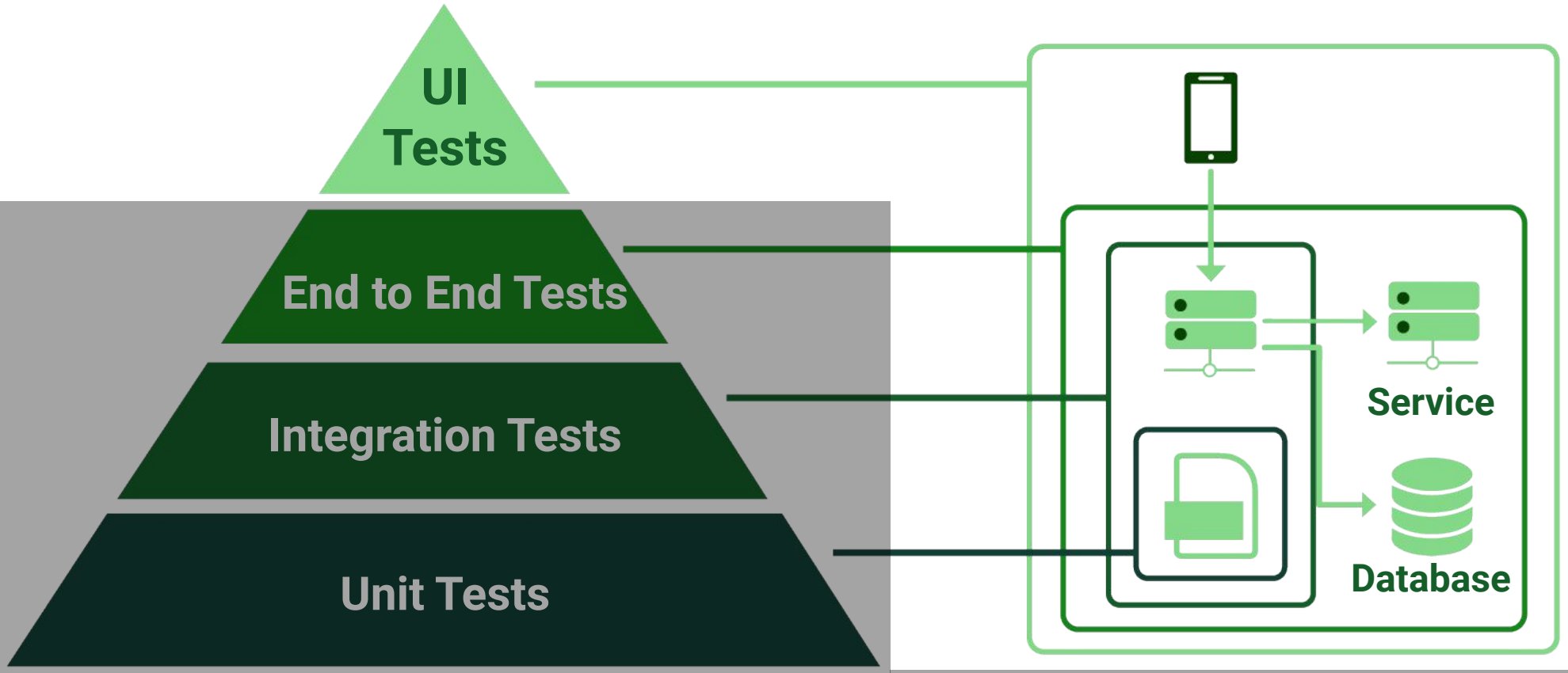
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

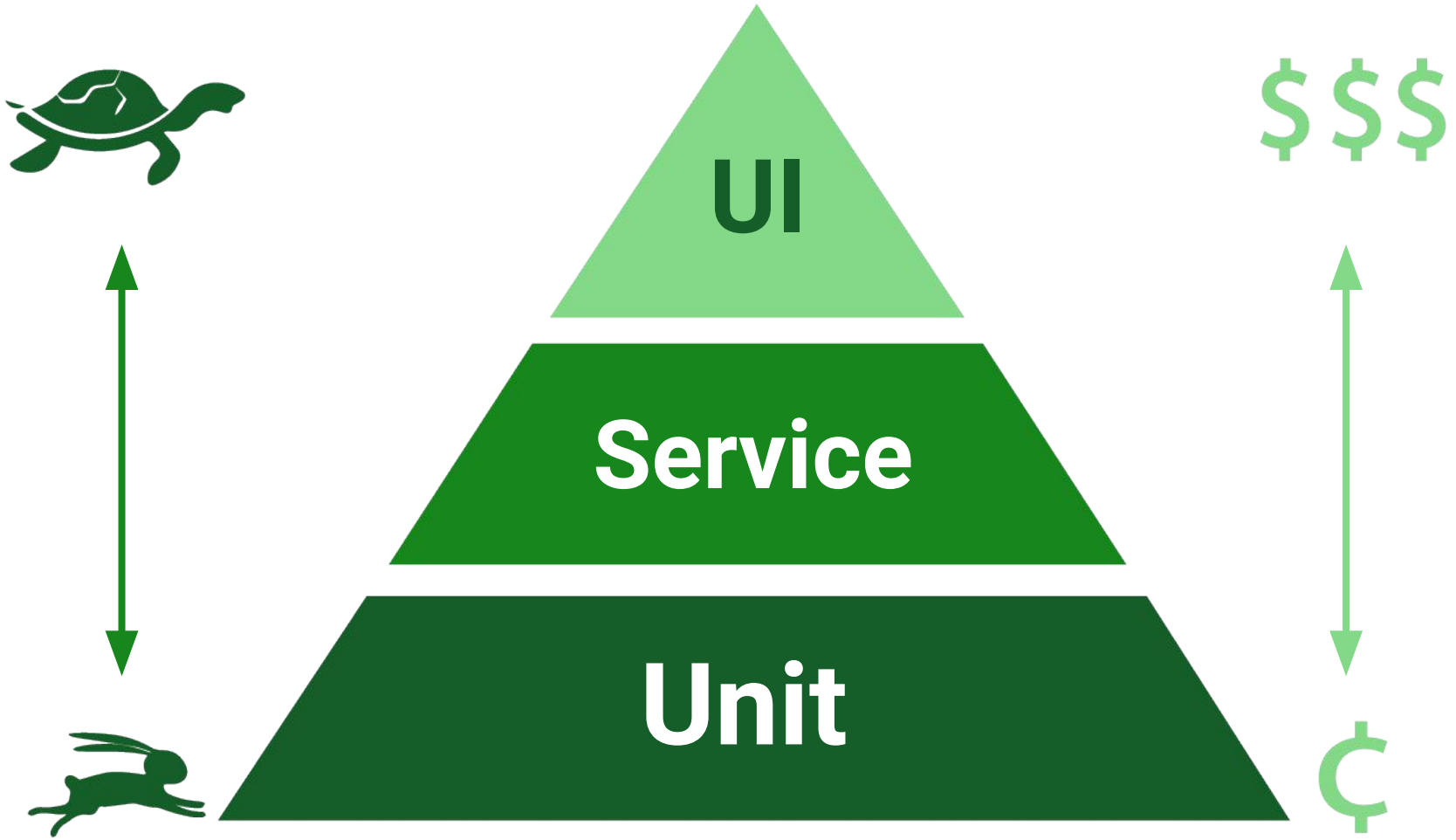
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

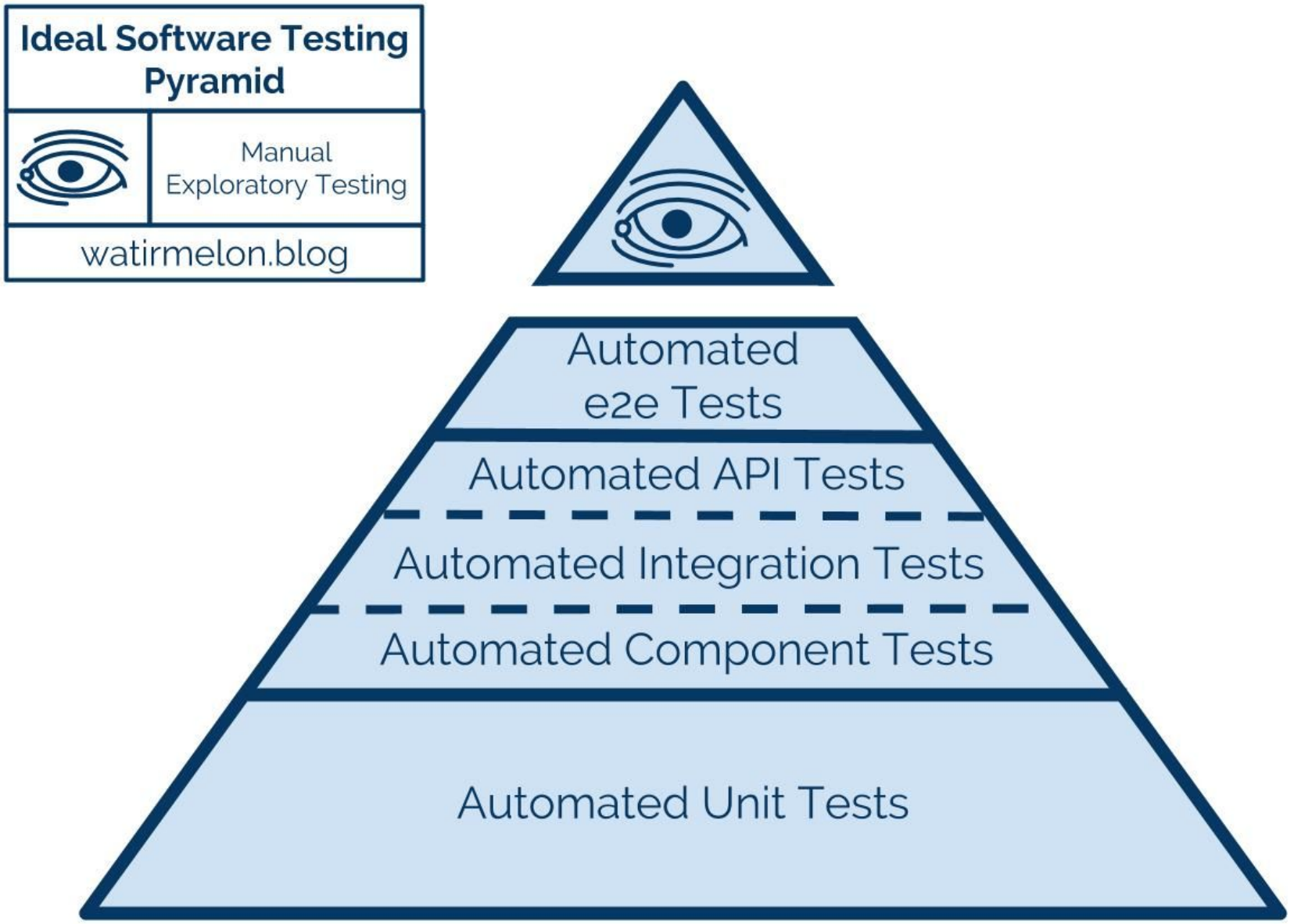
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS



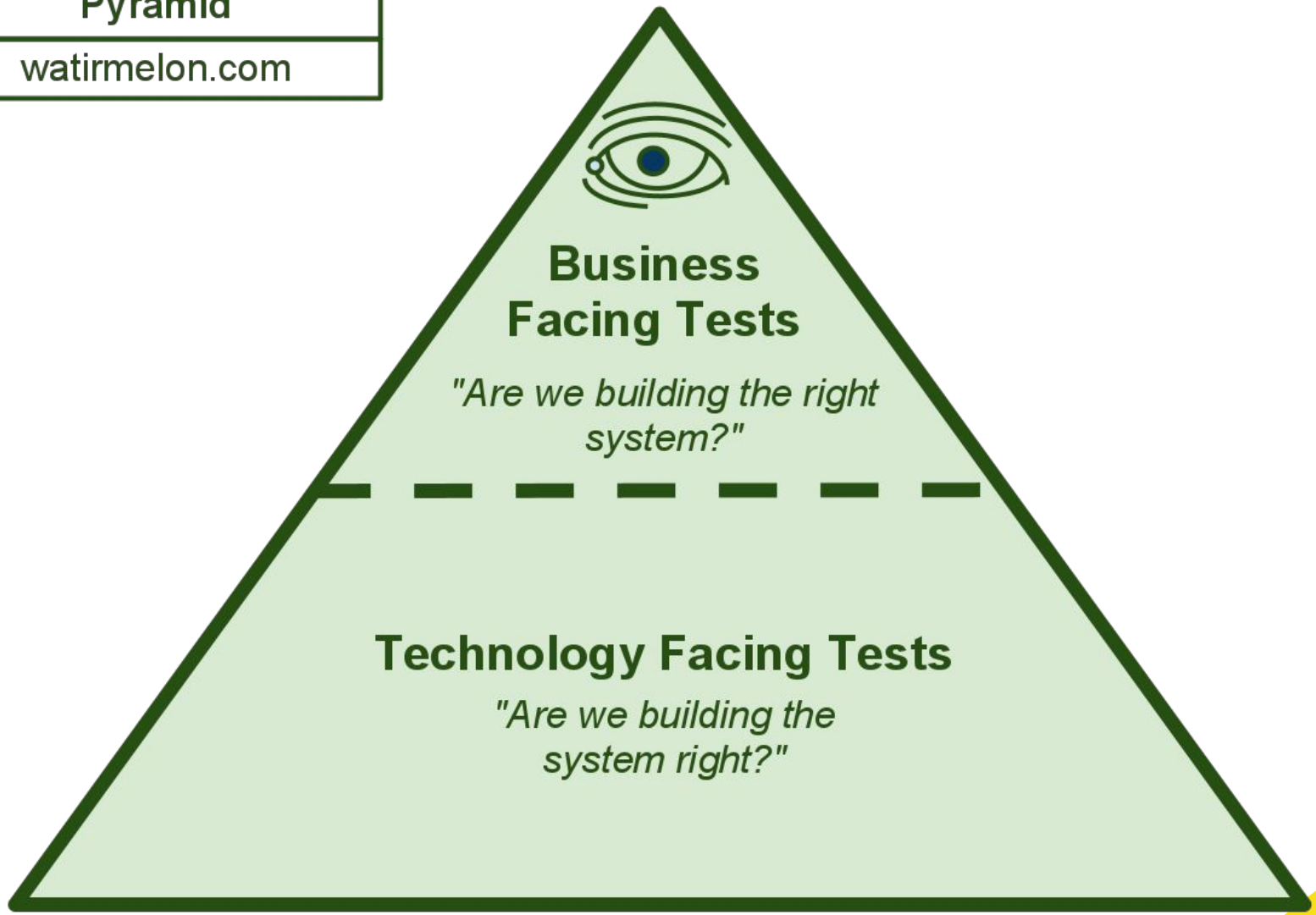
JS


Ideal Software Testing Pyramid
Manual Exploratory Testing
watirmelon.blog
Automated e2e Tests
Automated API Tests
Automated Integration Tests
Automated Component Tests
Automated Unit Tests

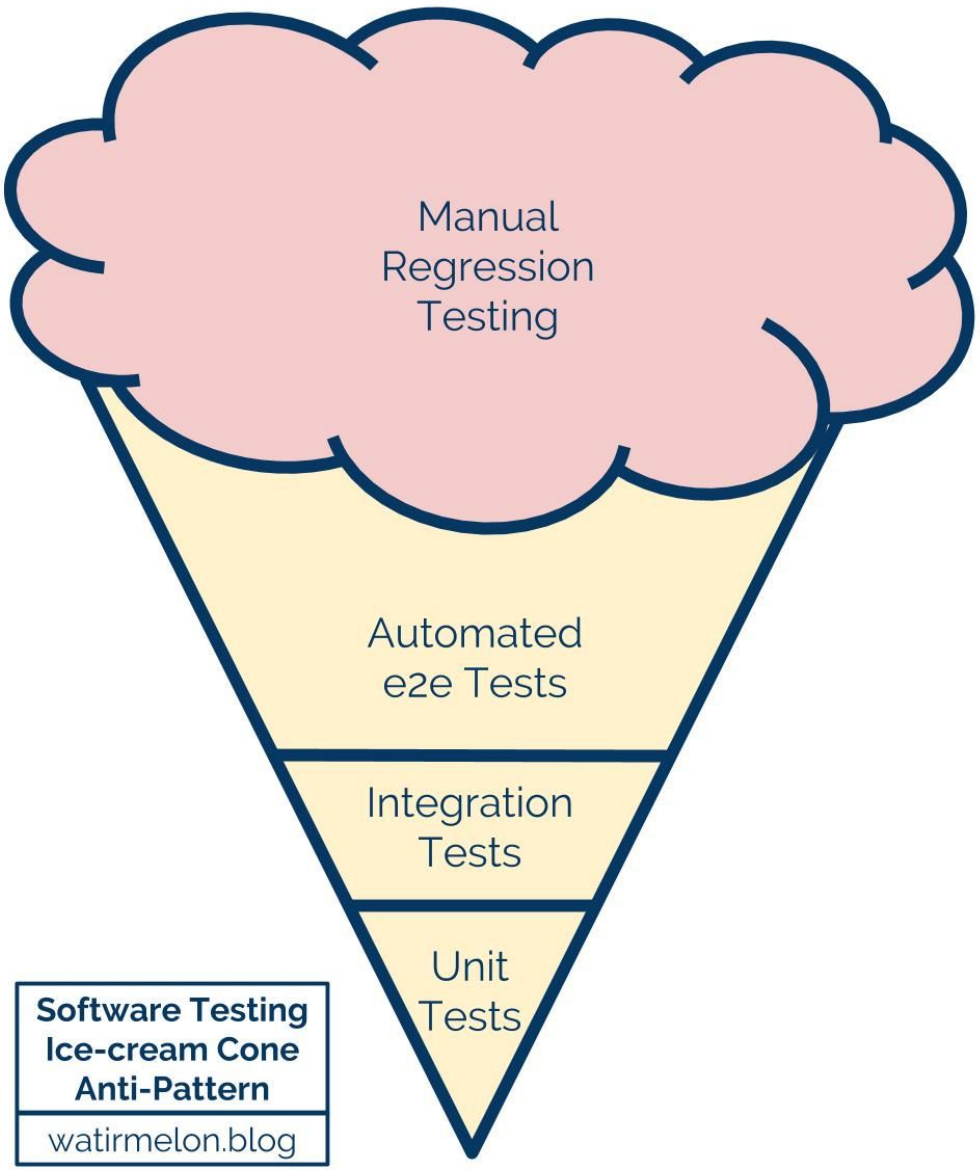
JS

Software Testing Pyramid

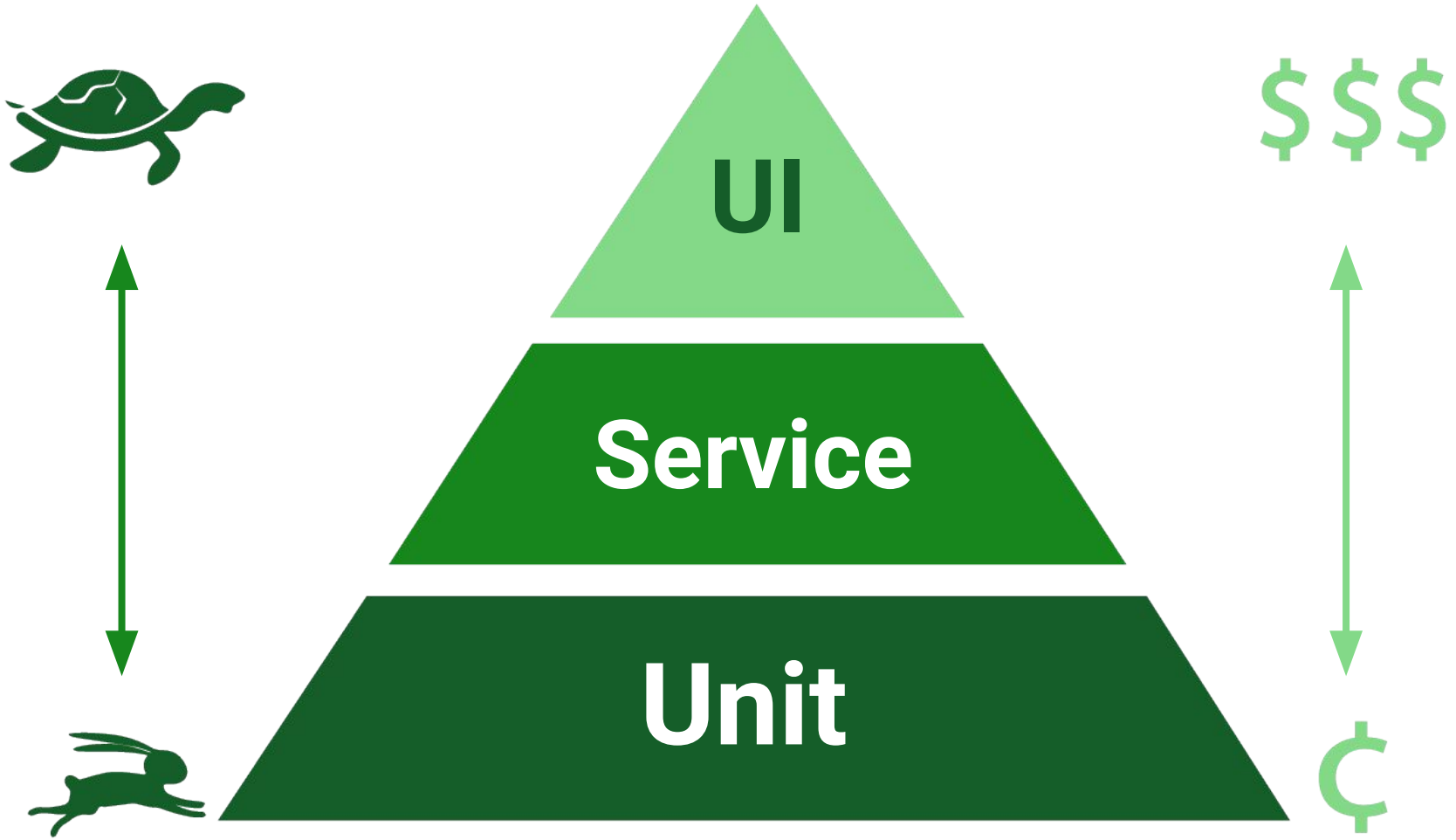
watirmelon.com



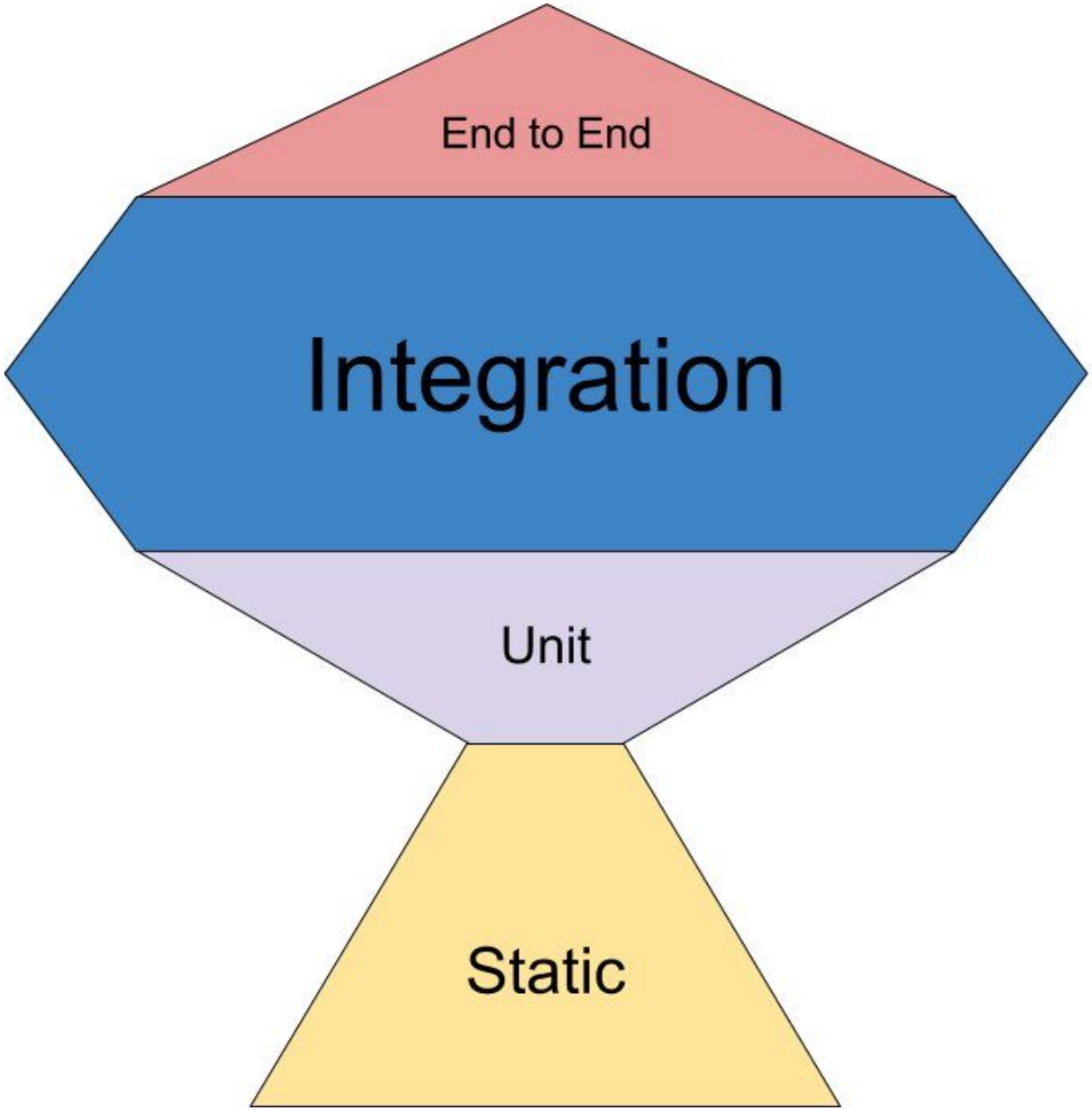
JS


Manual
Regression
Testing
Automated
e2e Tests
Integration
Tests
Unit
Tests
Software Testing
Ice-cream Cone
Anti-Pattern
watirmelon.blog

JS



JS



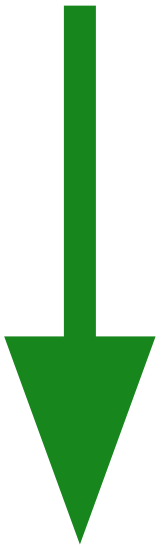
DEUDA TÉCNICA

Una ayuda estratégica.

JS

Riesgos

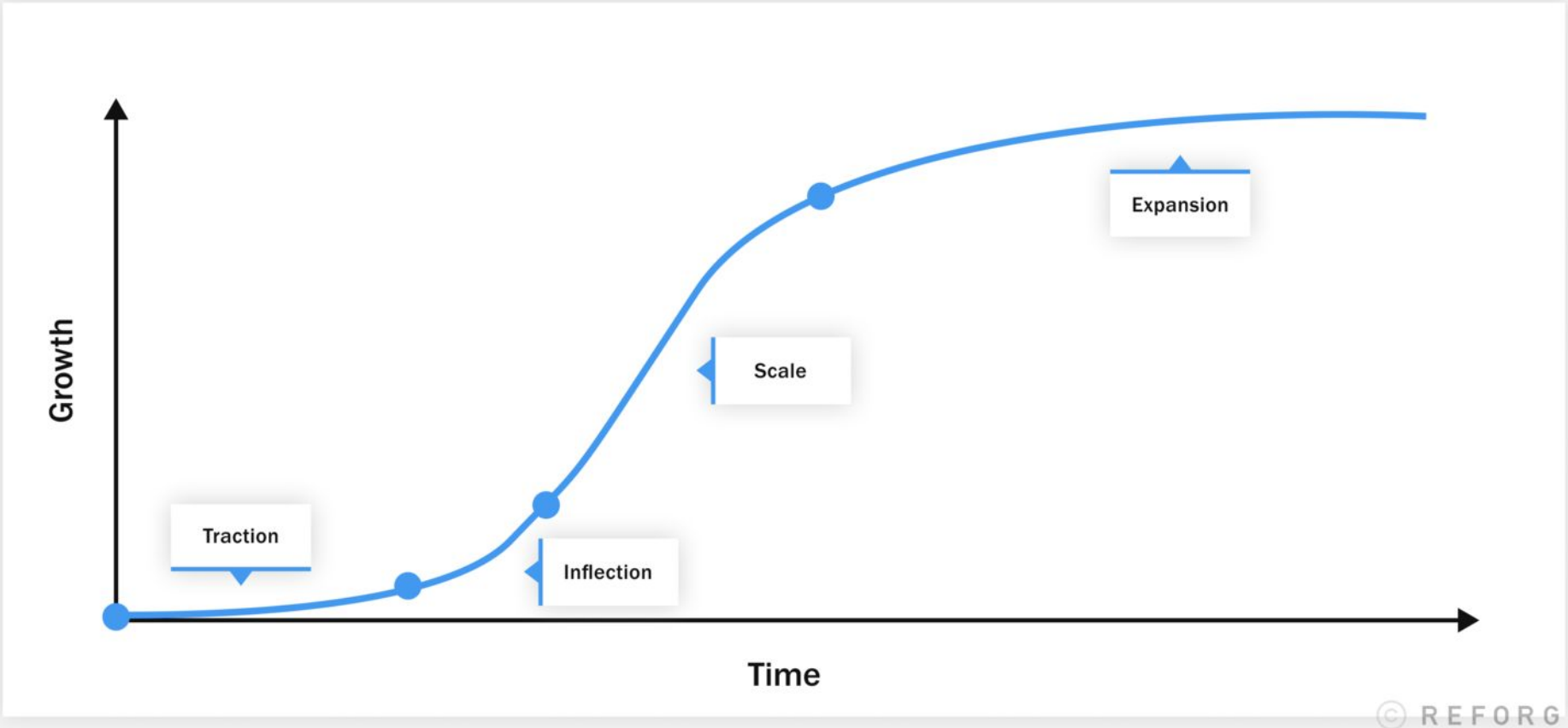
- Diseño
- Desarrollo
- Pruebas
- Producción



$

JS

Four Stages of Company Growth



Credits: Reforge

JS

Classifying Tech Debt

Maintenance Debt

Not keeping up with updates to tech work

Developer Efficiency Debt

Don't have the right tests, monitoring, or alerting in place

Stability Debt

Builds up different tech debt which affects infrastructure stability

Security Debt

Issues in tech stack leaving open security vulnerabilities

Technical Product Debt

Visible negative product impact

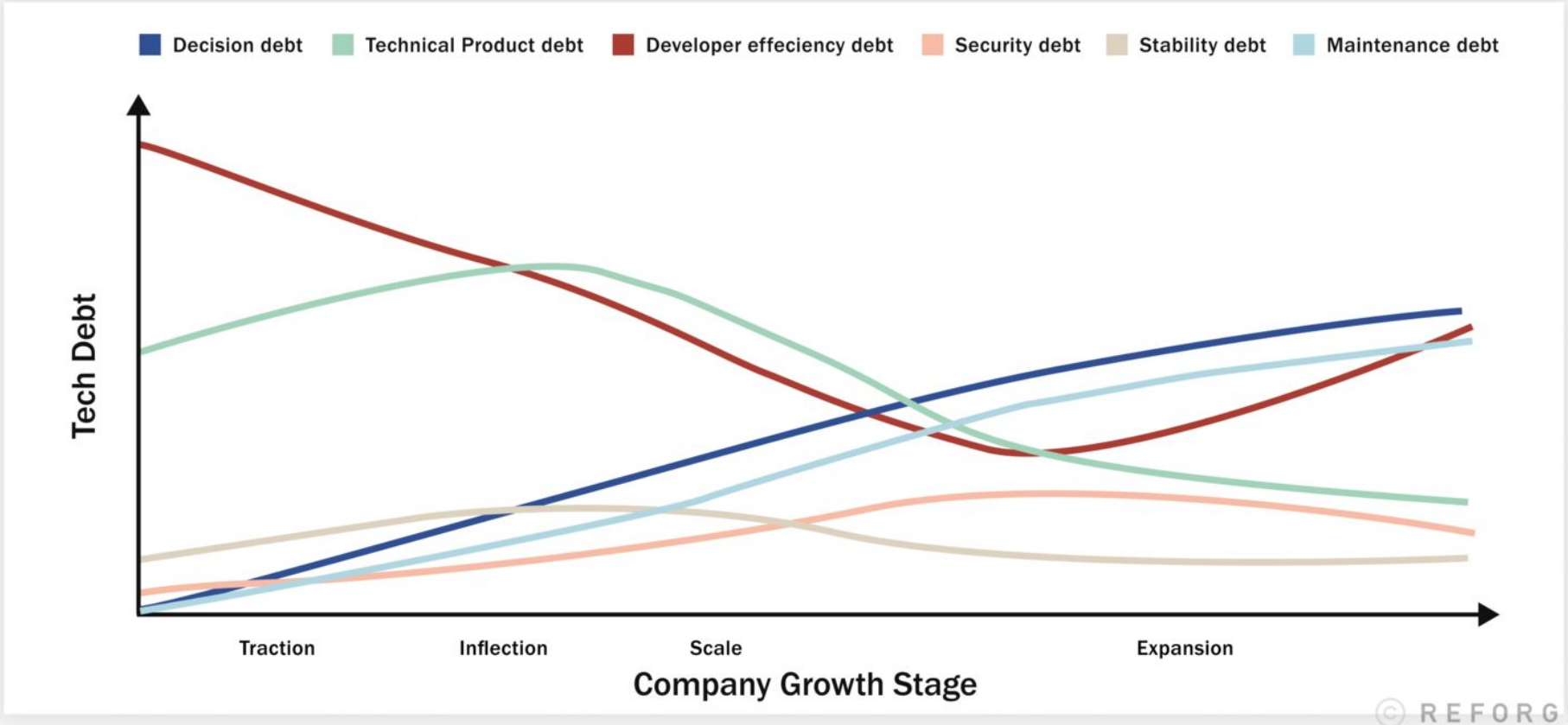
Decision Debt

Past tech decision was wrong or had tradeoffs we're now paying for

Credits: Reforge

JS

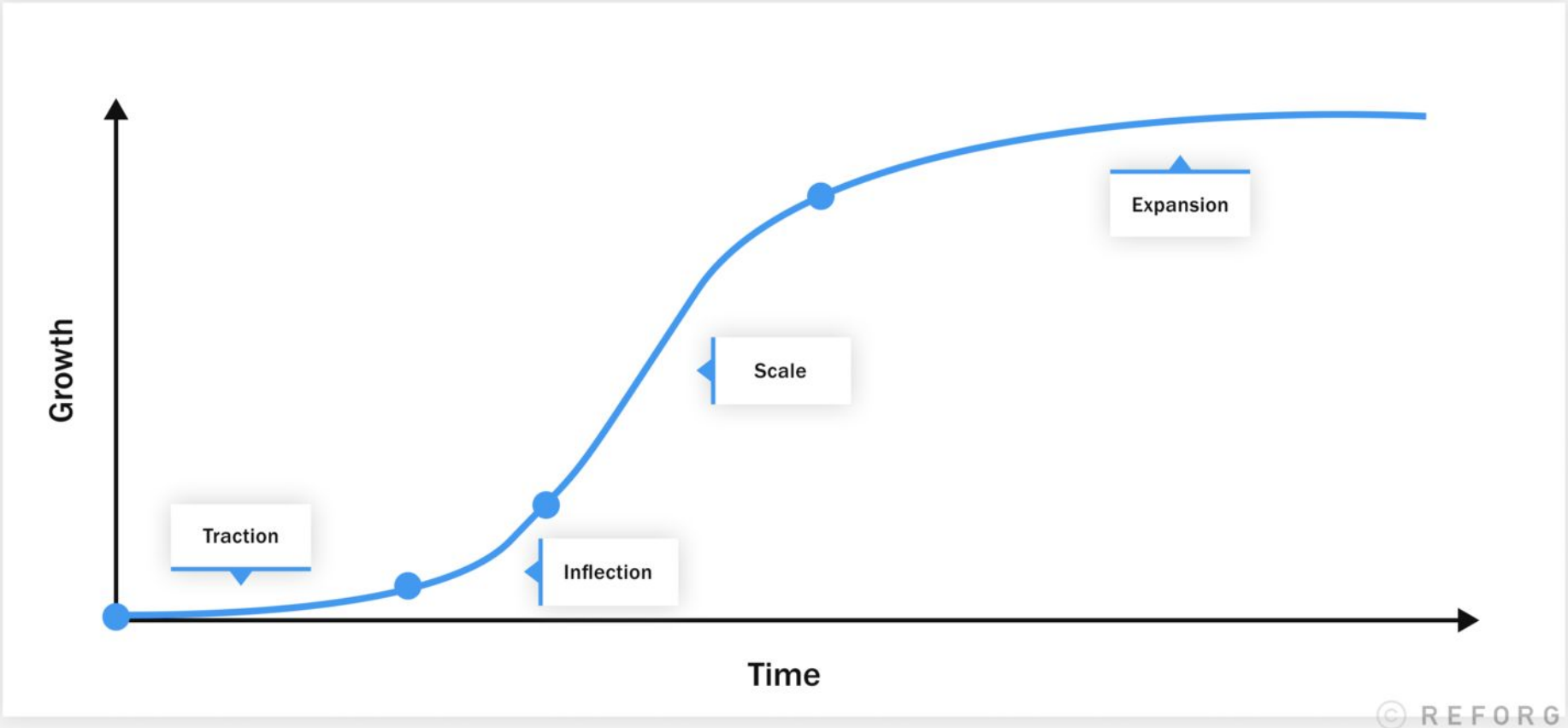
Tech Debt Portfolio as Company Grows



Credits: Reforge

JS

Four Stages of Company Growth



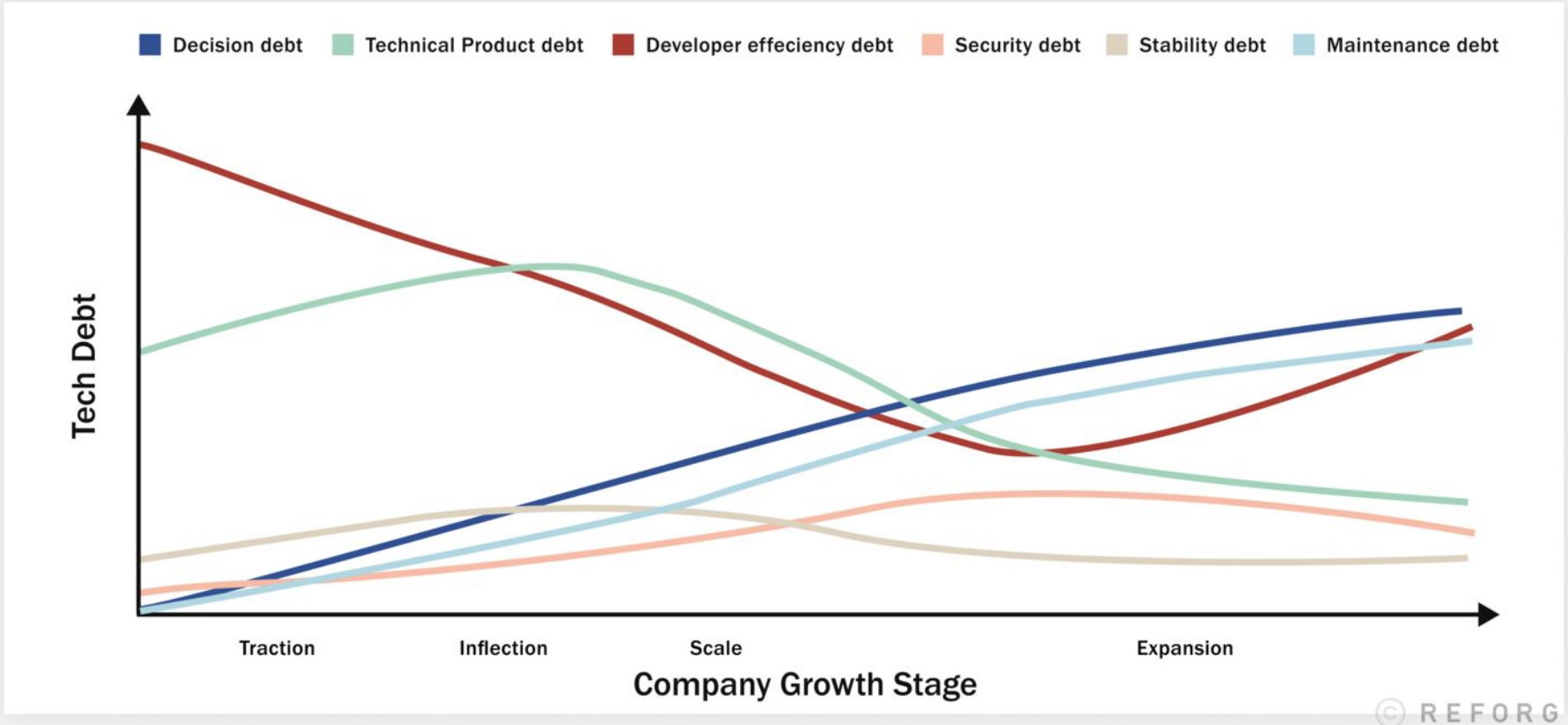
Credits: Reforge

Traction

Speed > accuracy

JS

Tech Debt Portfolio as Company Grows



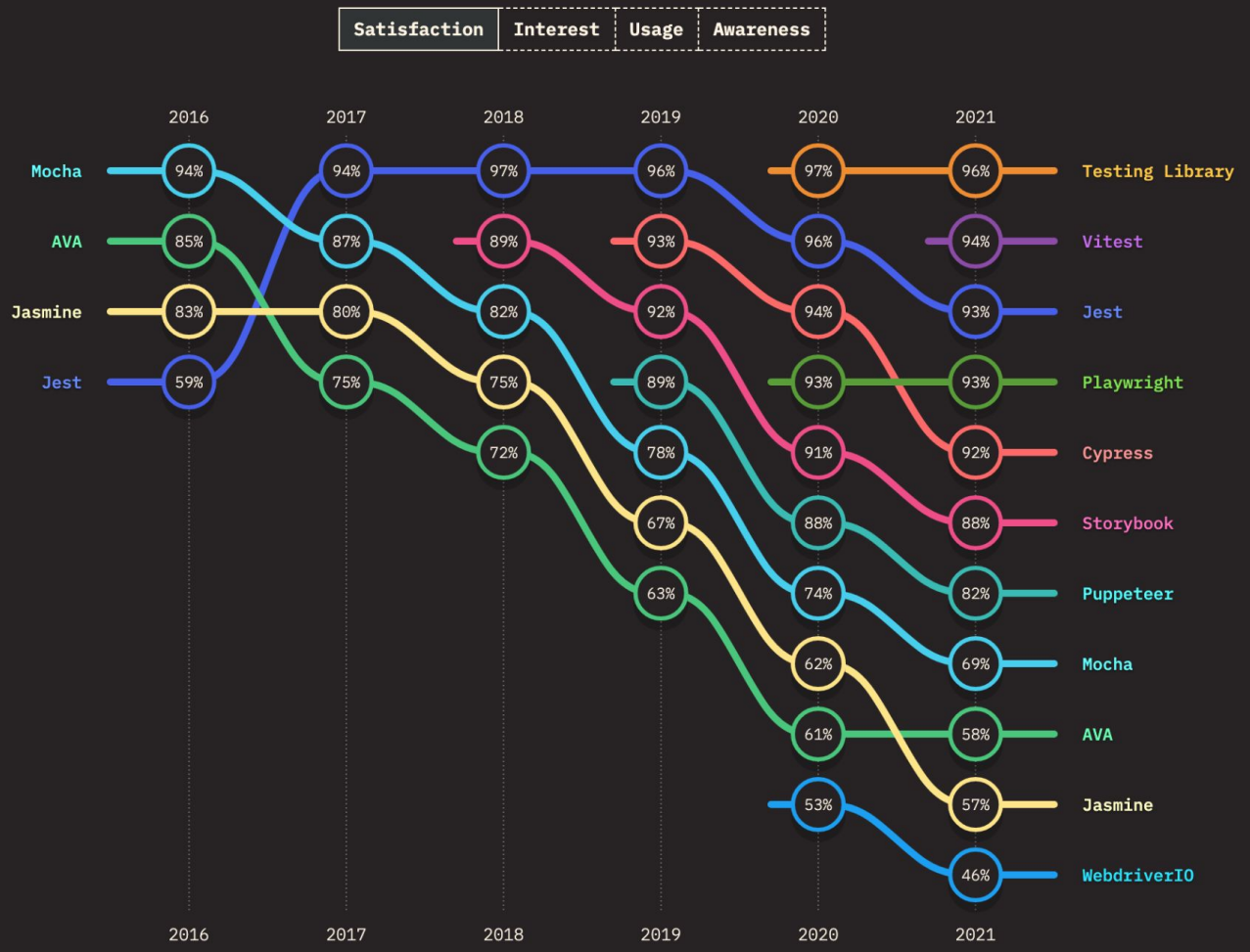
Credits: Reforge

HERRAMIENTAS

Tipos y clasificación.

RANKINGS

Satisfaction, interest, usage, and awareness ratio rankings.



Technologies with less than 10% awareness not included. Each ratio is defined as follows:

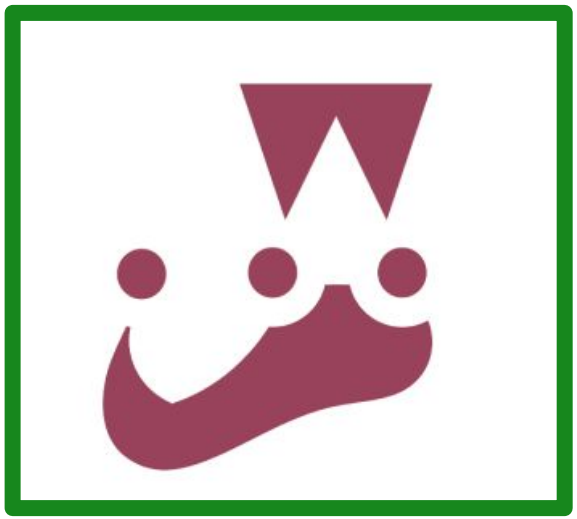
- Satisfaction: **would use again** / (**would use again** + **would not use again**)
- Interest: **want to learn** / (**want to learn** + **not interested**)
- Usage: (**would use again** + **would not use again**) / **total**
- Awareness: (**total** - **never heard**) / **total**

JS

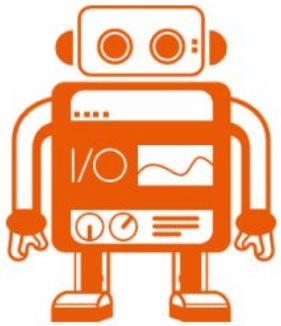
Tools

JS

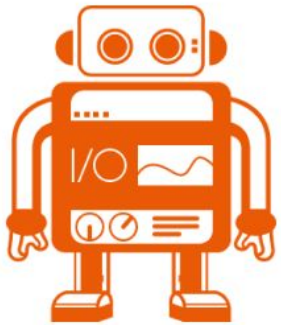
Tools



UI Testing



UI Testing



JS

API Testing

JS

API Testing

JS

Pruebas estáticas

```
describe("A suite is just a function", () => {
  const a;
  it("and so is a spec", () => {
    a = true;

    expect(a).toBe(true);
  });
});
```

```
describe("A suite is just a function", () => {
  const a;
  test("and so is a spec", () => {
    a = true;

    expect(a).toBe(true);
  });
});
```

```
it('null', () => {
  const n = null;
  expect(n).toBeNull();
  expect(n).toBeDefined();
  expect(n).not.toBeUndefined();
  expect(n).not.toBeTruthy();
  expect(n).toBeFalsy();
});
```

```
test('null', () => {
  const n = null;
  expect(n).toBeNull();
  expect(n).toBeDefined();
  expect(n).not.toBeUndefined();
  expect(n).not.toBeTruthy();
  expect(n).toBeFalsy();
});
```

CREANDO PROYECTO

Manos a la obra.

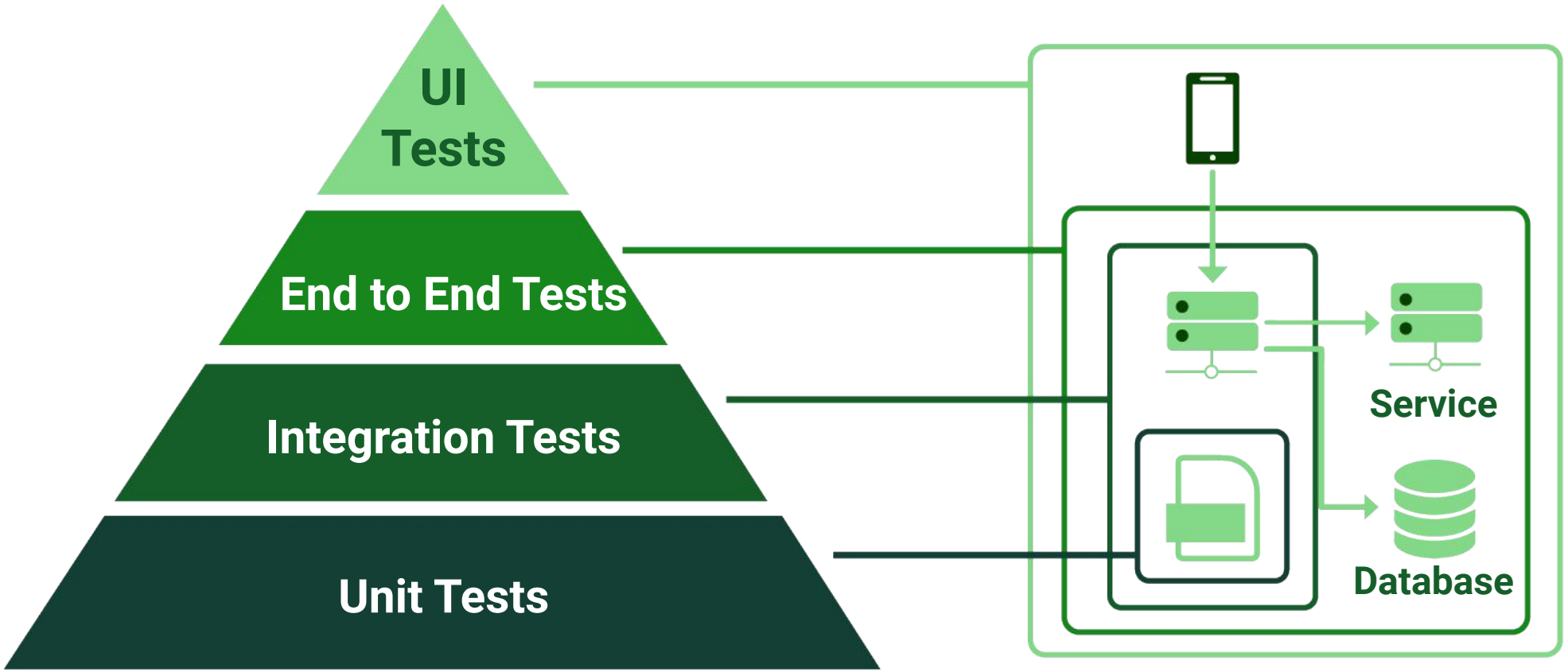
TU PRIMER TEST

Comprendiendo el flujo.

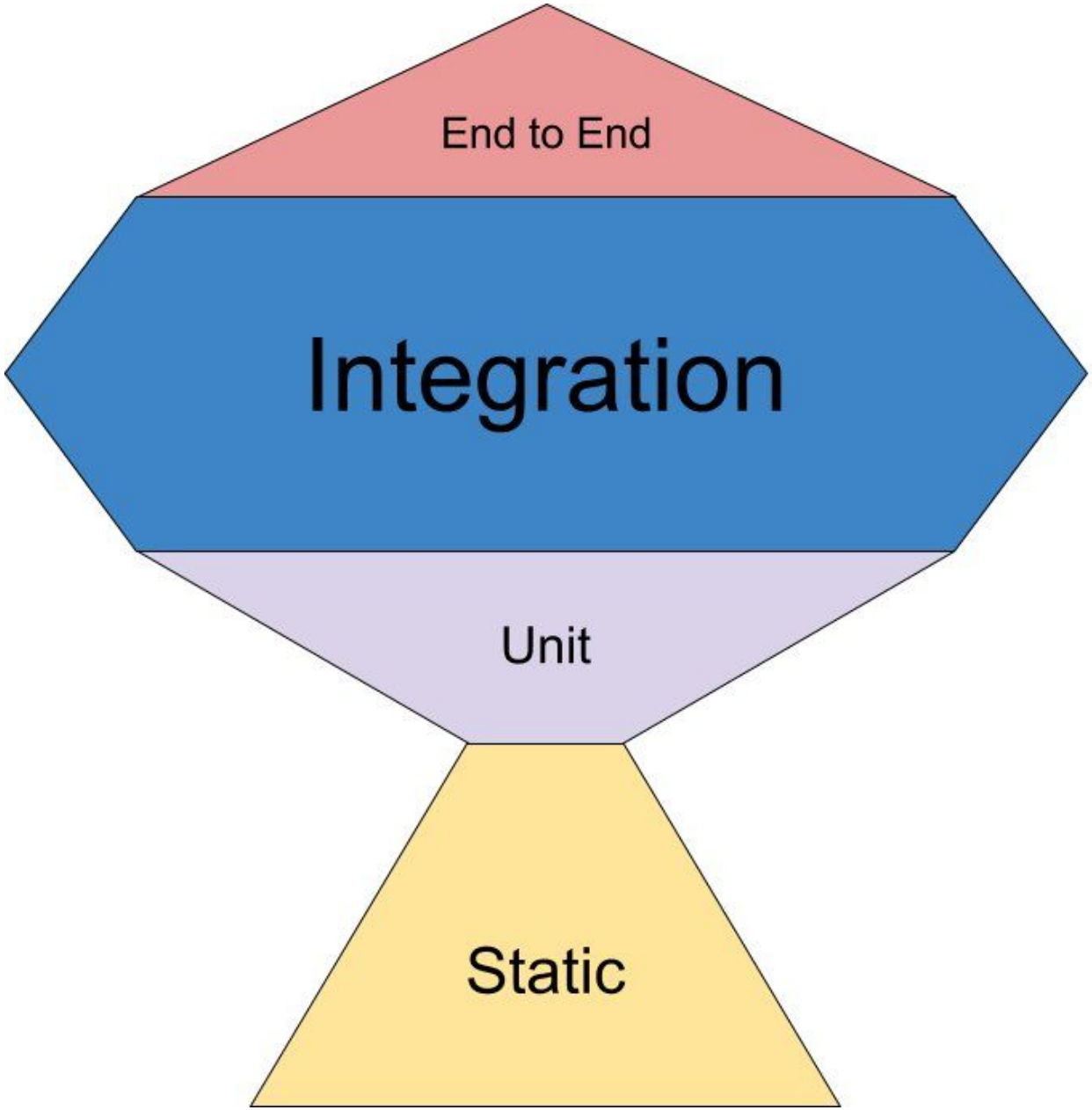
PRUEBAS ESTÁTICAS

Pruebas sin ejecutar el código.

JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS



JS

Pruebas estáticas

ASSERTIONS

Resolver lo esperado.

```
it('null', () => {
  const n = null;
  expect(n).toBeNull();
  expect(n).toBeDefined();
  expect(n).not.toBeUndefined();
  expect(n).not.toBeTruthy();
  expect(n).toBeFalsy();
});
```

```
test('null', () => {
  const n = null;
  expect(n).toBeNull();
  expect(n).toBeDefined();
  expect(n).not.toBeUndefined();
  expect(n).not.toBeTruthy();
  expect(n).toBeFalsy();
});
```

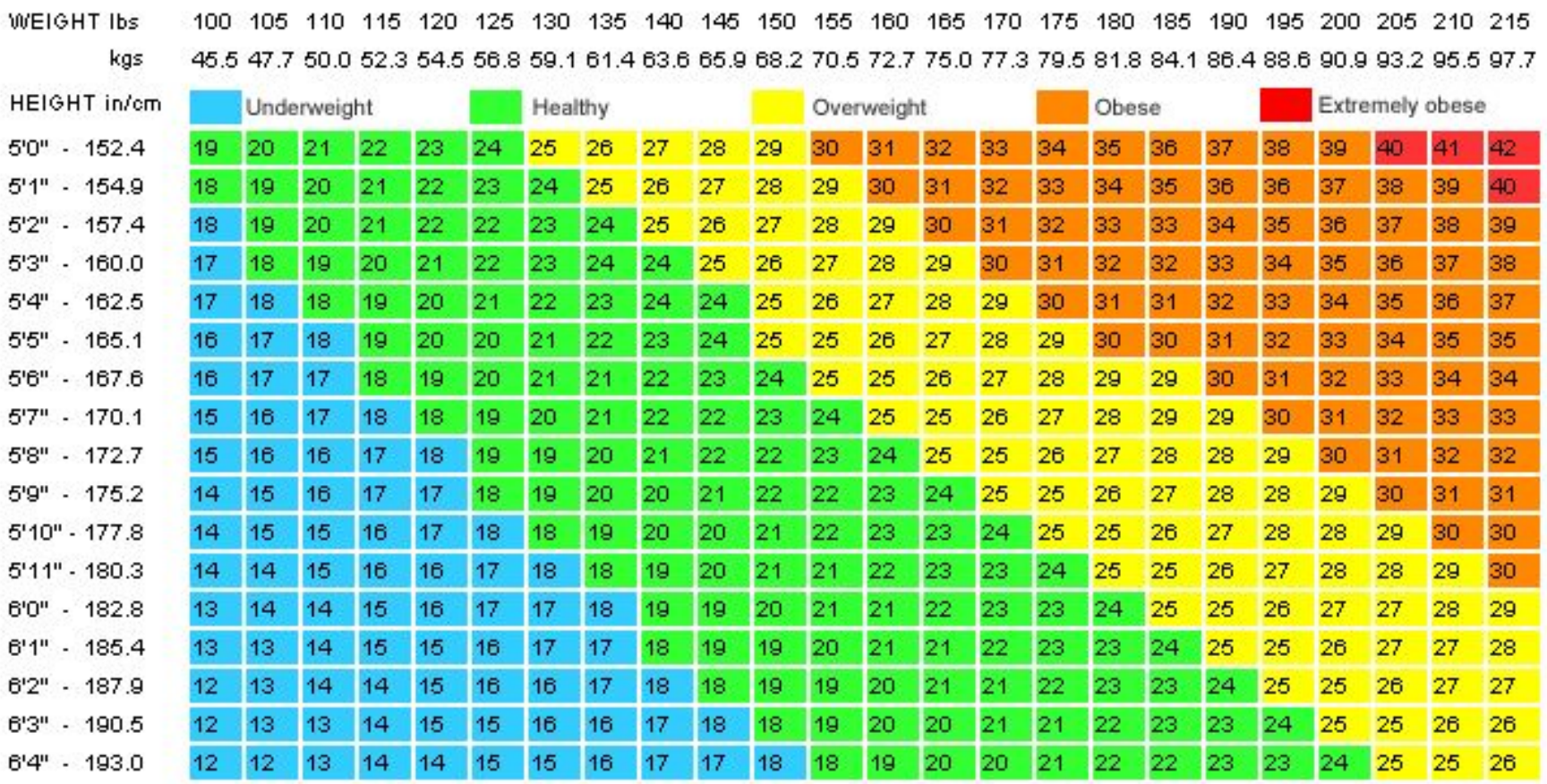
SETUP / TEARDOWN

Configuración y scope de pruebas

IMPLEMENTACIÓN

Pruebas a Person Class

JS


WEIGHT lbs 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215
kgs 45.5 47.7 50.0 52.3 54.5 56.8 59.1 61.4 63.6 65.9 68.2 70.5 72.7 75.0 77.3 79.5 81.8 84.1 86.4 88.6 90.9 93.2 95.5 97.7
HEIGHT in/cm
Underweight
Healthy
Overweight
Obese
Extremely obese
5'0" - 152.4 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
5'1" - 154.9 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 36 37 38 39 40
5'2" - 157.4 18 19 20 21 22 22 23 24 25 26 27 28 29 30 31 32 33 33 34 35 36 37 38 39
5'3" - 160.0 17 18 19 20 21 22 23 24 24 25 26 27 28 29 30 31 32 32 33 34 35 36 37 38
5'4" - 162.5 17 18 18 19 20 21 22 23 24 24 25 26 27 28 29 30 31 31 32 33 34 35 36 37
5'5" - 165.1 16 17 18 19 20 20 21 22 23 24 25 25 26 27 28 29 30 30 31 32 33 34 35 35
5'6" - 167.6 16 17 17 18 19 20 21 21 22 23 24 25 25 26 27 28 29 29 30 31 32 33 34 34
5'7" - 170.1 15 16 17 18 18 19 20 21 22 22 23 24 25 25 26 27 28 29 29 30 31 32 33 33
5'8" - 172.7 15 16 16 17 18 19 19 20 21 22 22 23 24 25 25 26 27 28 28 29 30 31 32 32
5'9" - 175.2 14 15 16 17 17 18 19 20 20 21 22 22 23 24 25 25 26 27 28 28 29 30 31 31
5'10" - 177.8 14 15 15 16 17 18 18 19 20 20 21 22 23 23 24 25 25 26 27 28 28 29 30 30
5'11" - 180.3 14 14 15 16 16 17 18 18 19 20 21 21 22 23 23 24 25 25 26 27 28 28 29 30
6'0" - 182.8 13 14 14 15 16 17 17 18 19 19 20 21 21 22 23 23 24 25 25 26 27 27 28 29
6'1" - 185.4 13 13 14 15 15 16 17 17 18 19 19 20 21 21 22 23 23 24 25 25 26 27 27 28
6'2" - 187.9 12 13 14 14 15 16 16 17 18 18 19 19 20 21 21 22 23 23 24 25 25 26 27 27
6'3" - 190.5 12 13 13 14 15 15 16 16 17 18 18 19 20 20 21 21 22 23 23 24 25 25 26 26
6'4" - 193.0 12 12 13 14 14 15 15 16 17 17 18 18 19 20 20 21 22 22 23 23 24 25 25 26

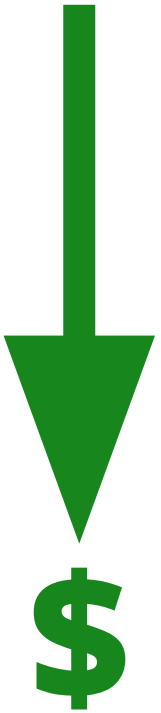
Tipos de Pruebas

Clasificación y cantidad

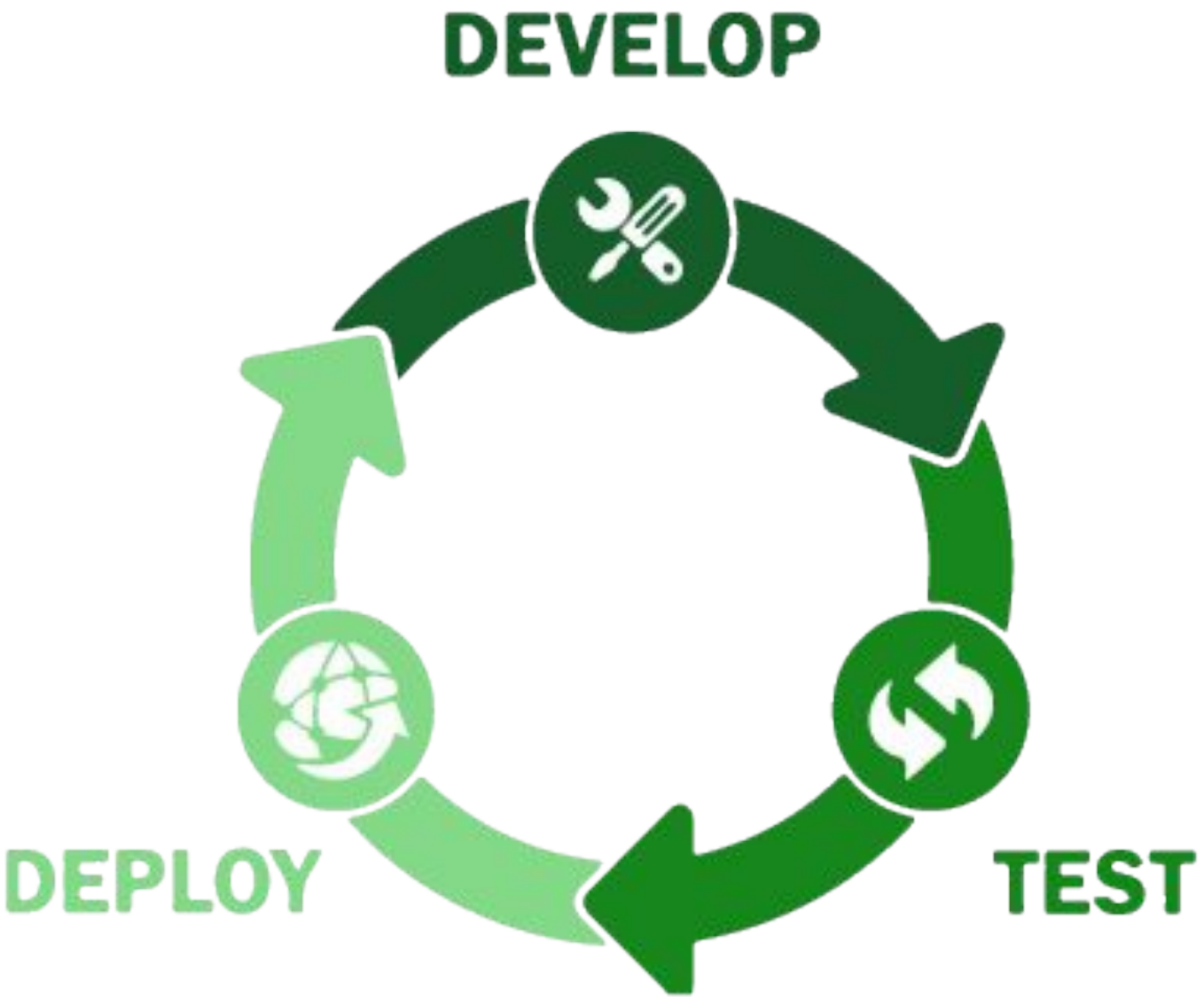
JS

Fases

- Requerimientos
- Diseño / Arq.
- Desarrollo
- QA
- Producción



JS



SUT

System Under Test

Validar

¿Estamos construyendo el sistema correcto?

Verificar

¿Estamos construyendo el sistema correctamente?

JS

Requerimientos

- **SUT:** Requerimientos
- **Ejecución:** Manuales
- **Objetivo:** Validar
- **Herramientas:** Inspección

JS

Diseño / Arquitectura

- **SUT:** Sistema
- **Ejecución:** Manuales / Automaticas
- **Objetivo:** Verificar
- **Herramientas:** Case Tools

JS

Desarrollo / Implementación

- **SUT:** Depende
- **Ejecución:** Automaticas
- **Objetivo:** Verificar

JS

Pruebas estáticas

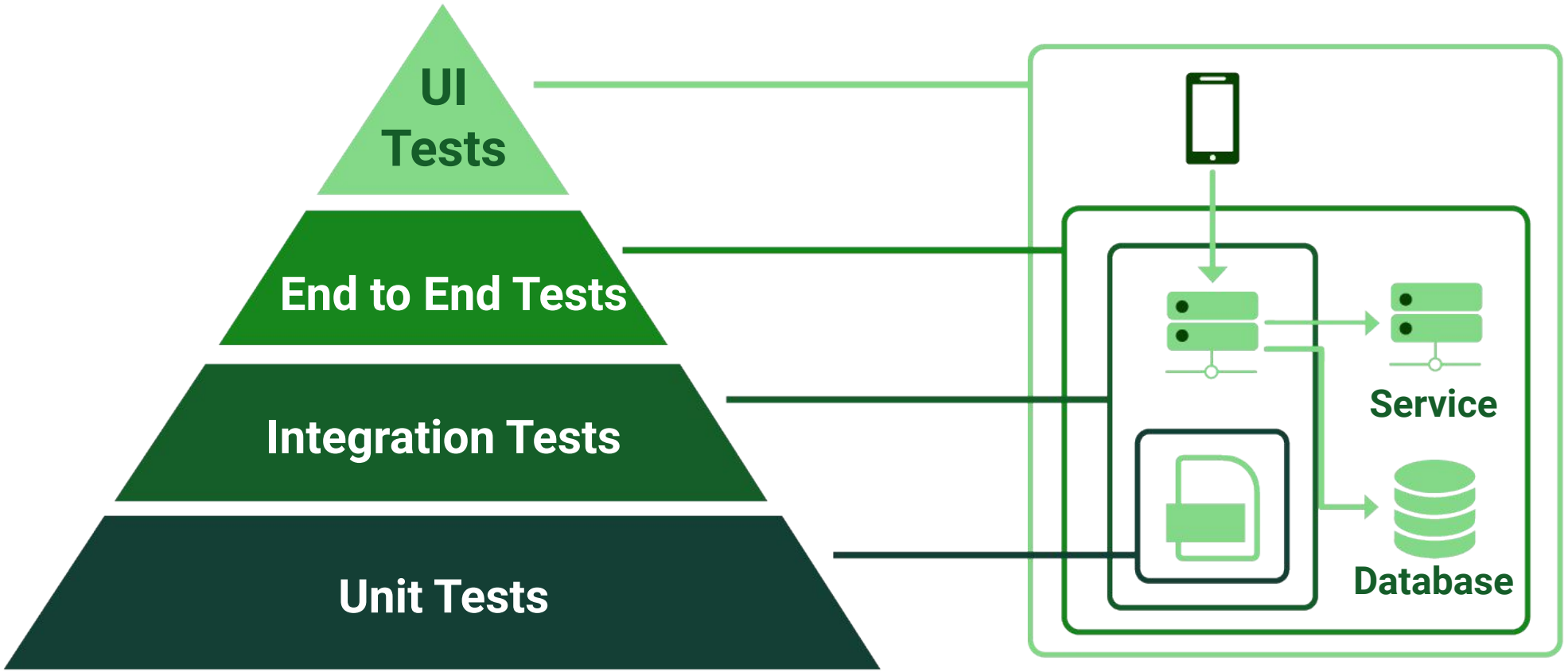
JS

Pruebas estáticas

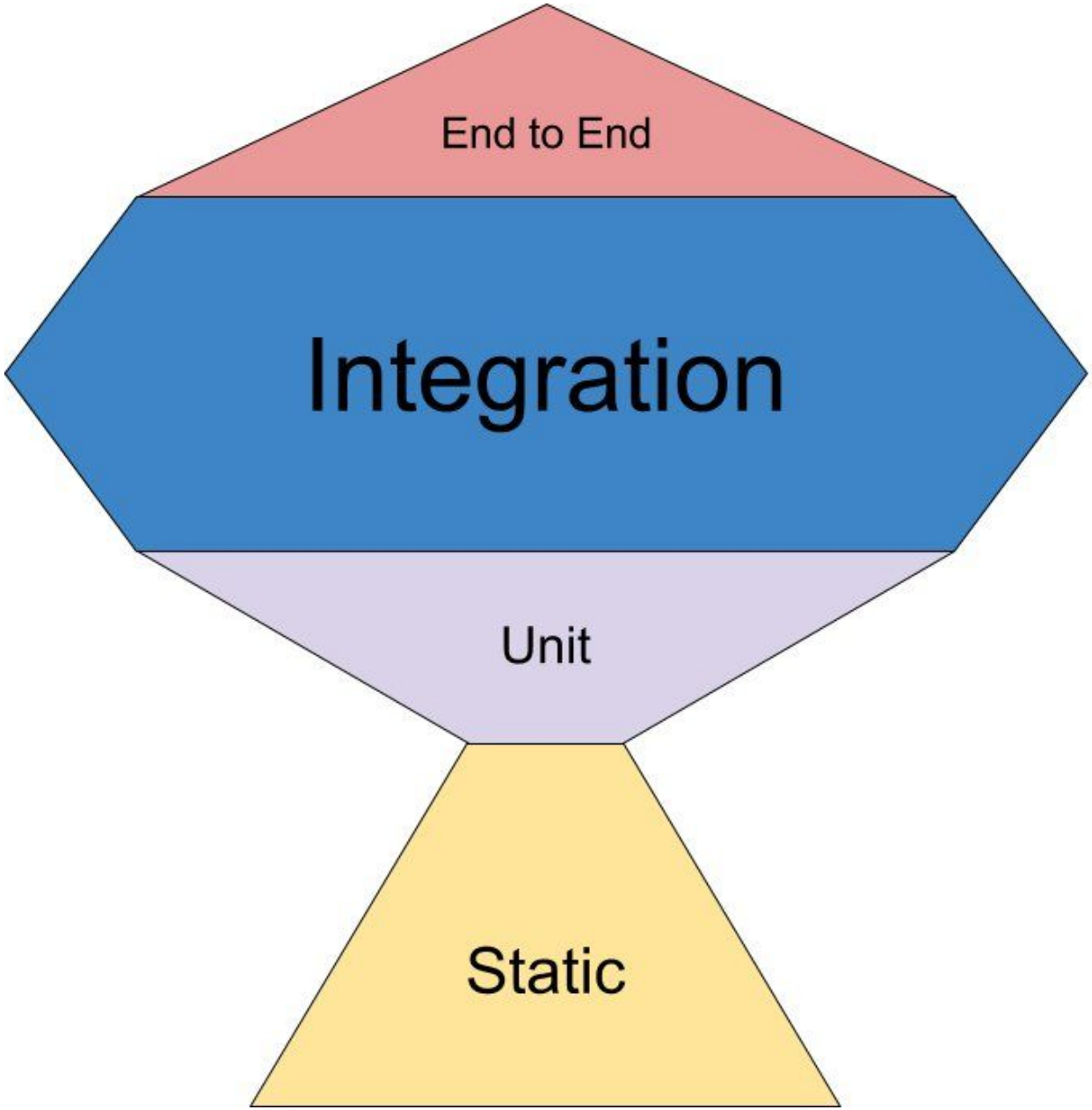
Pruebas estáticas

Pruebas Funcionales

JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS

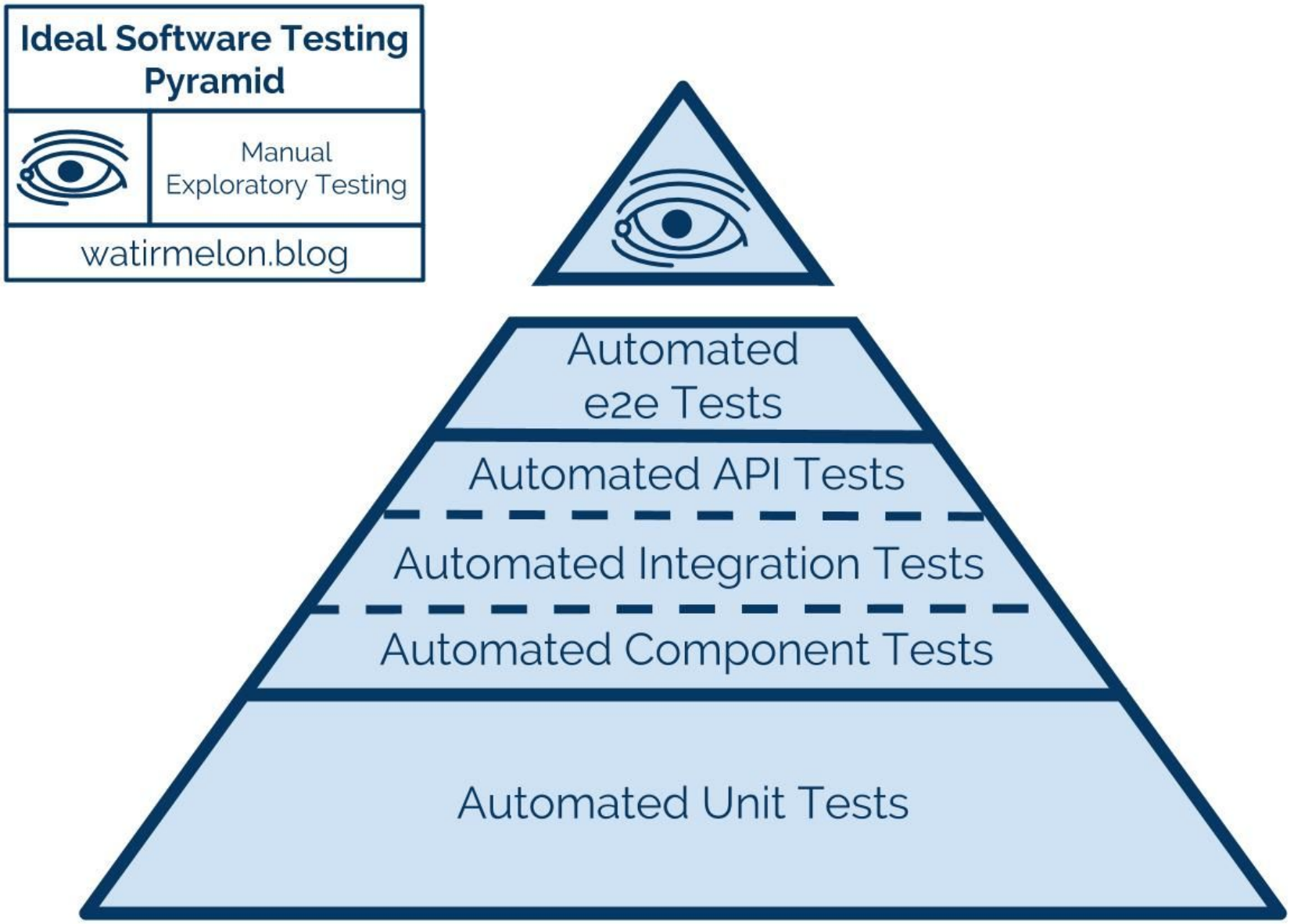


¿Quién hace las pruebas?

Unit test

Si escribes el código tienes que escribir el nit test

JS


Ideal Software Testing Pyramid
Manual Exploratory Testing
watirmelon.blog
Automated e2e Tests
Automated API Tests
Automated Integration Tests
Automated Component Tests
Automated Unit Tests

Pruebas no funcionales

JS

No funcionales

- Rendimiento
- De carga o estabilidad
- Pruebas de estrés
- Usabilidad
- Seguridad

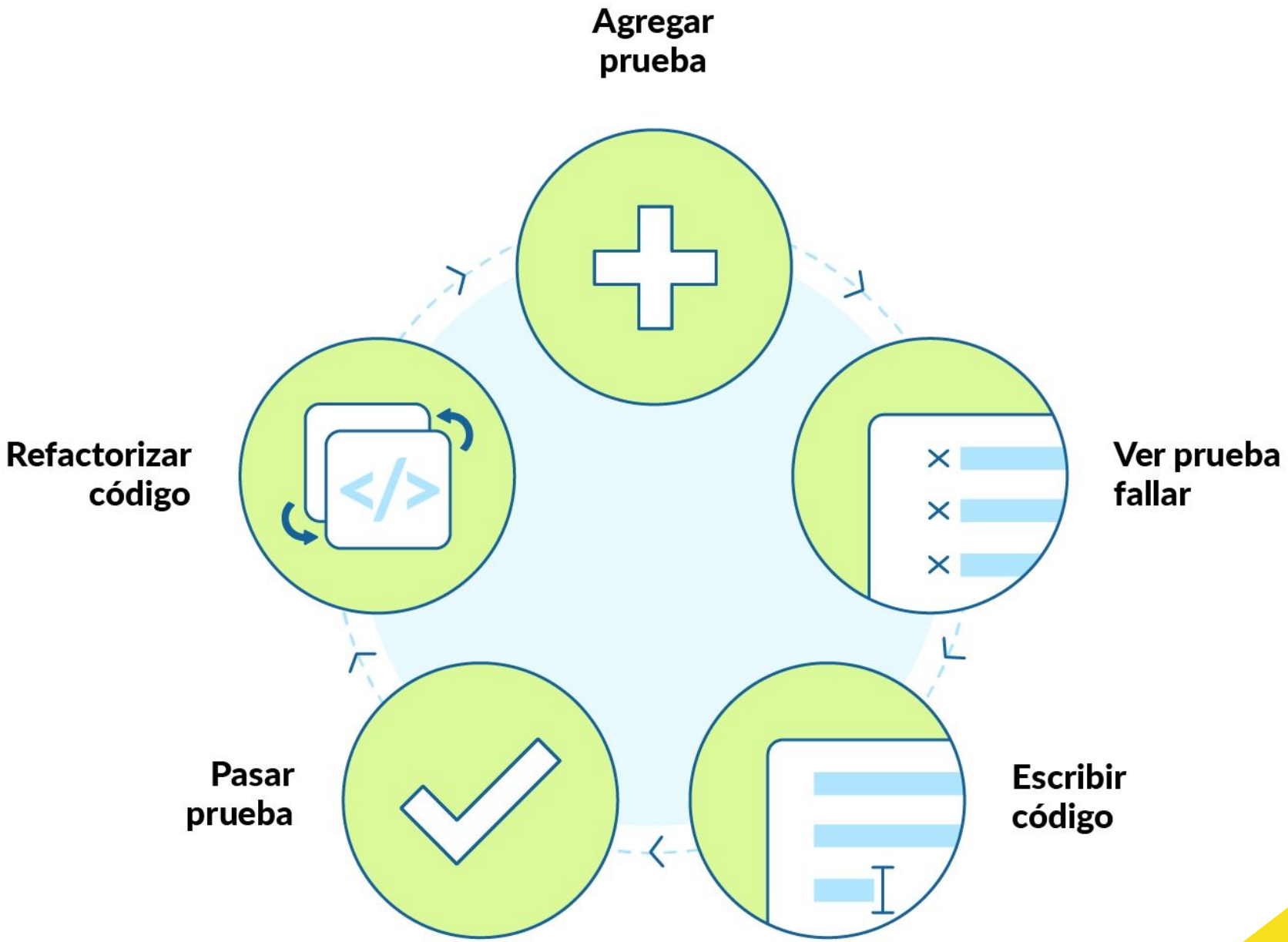
JS


Chaos
Engineering
The history, principles, & practice
Gremlin

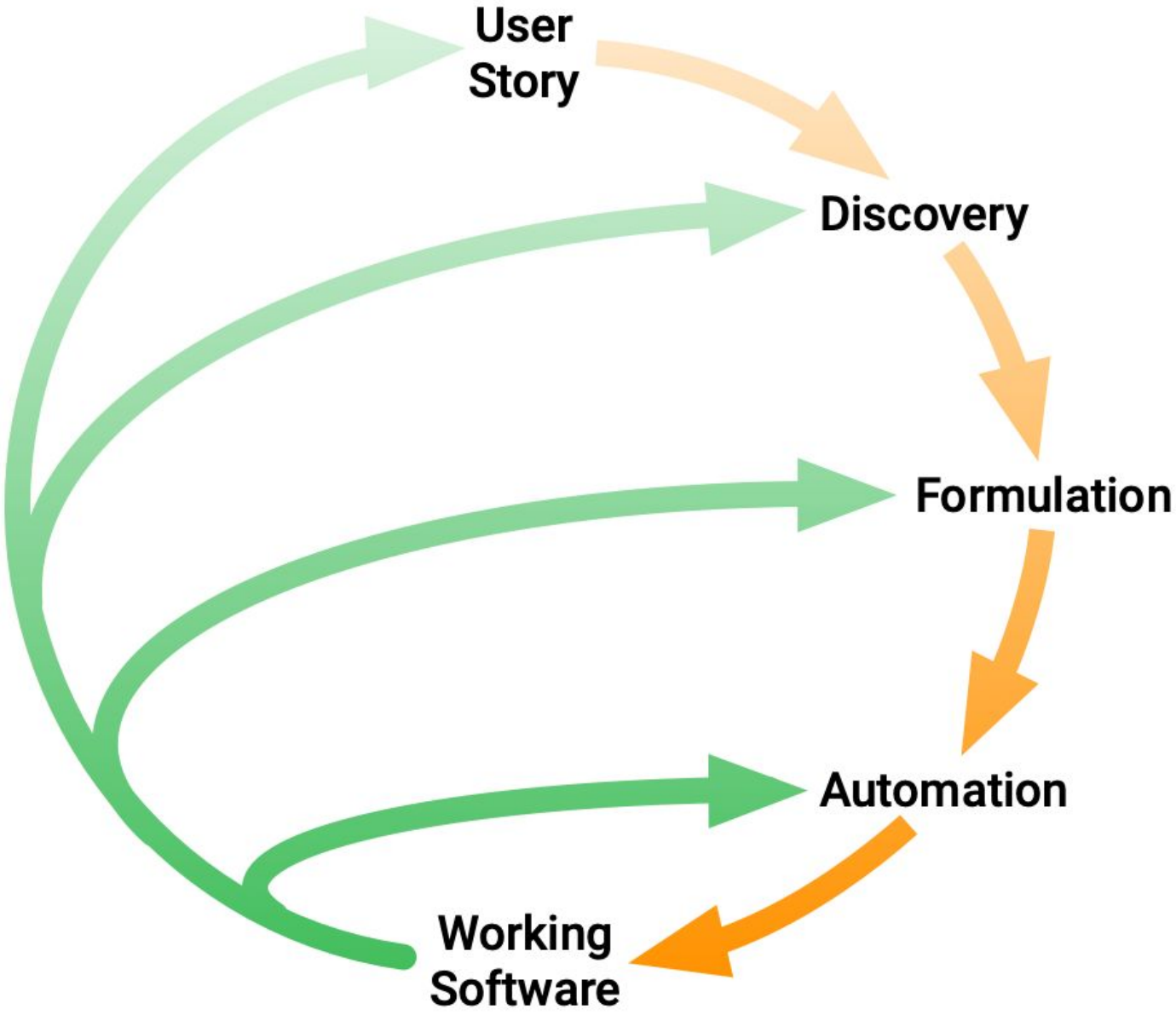
METODOLOGÍAS

Y otros conceptos.

JS


Agregar prueba
Ver prueba fallar
Escribir código
Pasar prueba
Refactorizar código

JS


User Story
Discovery
Formulation
Automation
Working Software

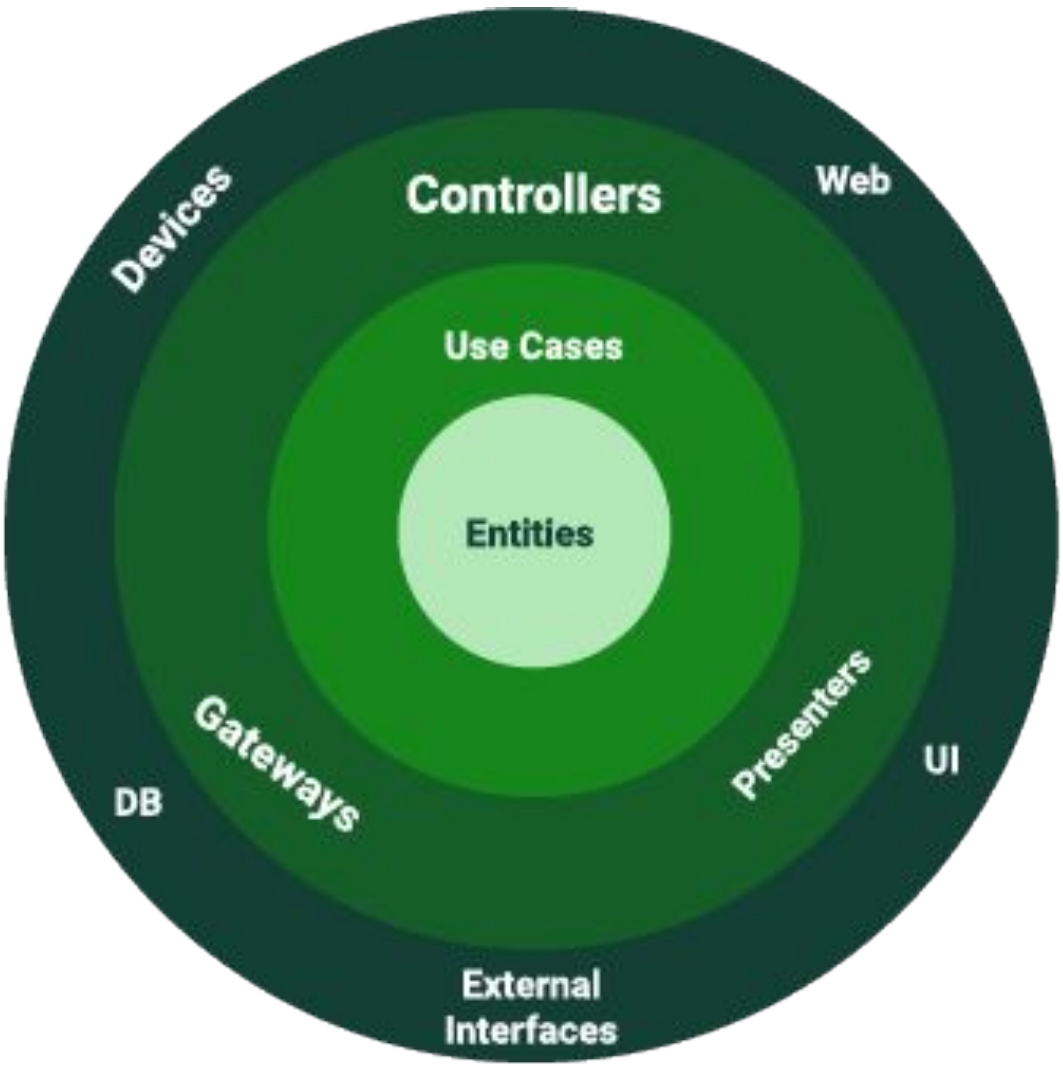
Falso positivo

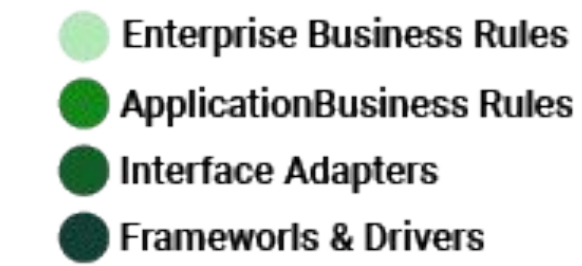
Un examen indica una enfermedad cuando no la hay

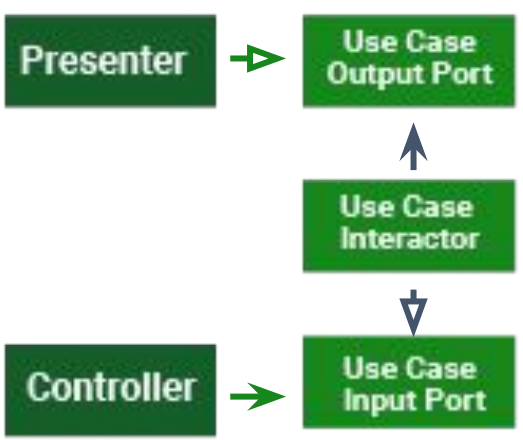
Falso negativo

Un examen indica que todo es normal cuando en realidad el paciente está enfermo.

JS The Clean Architecture



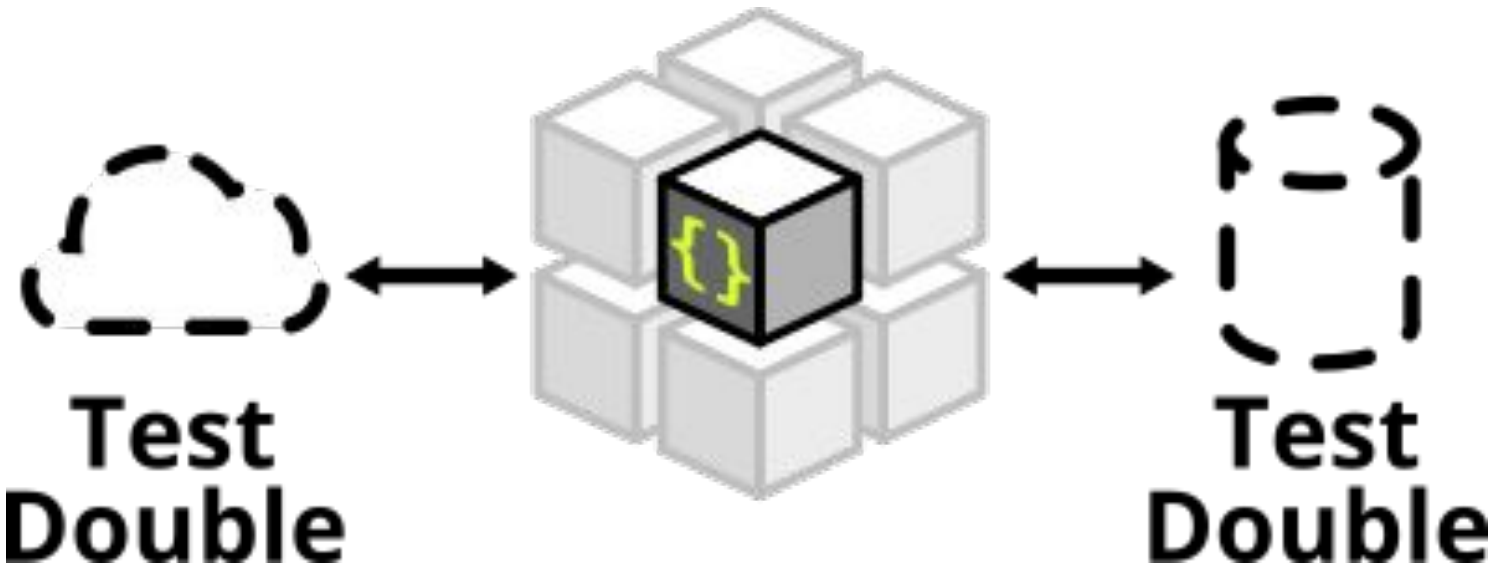




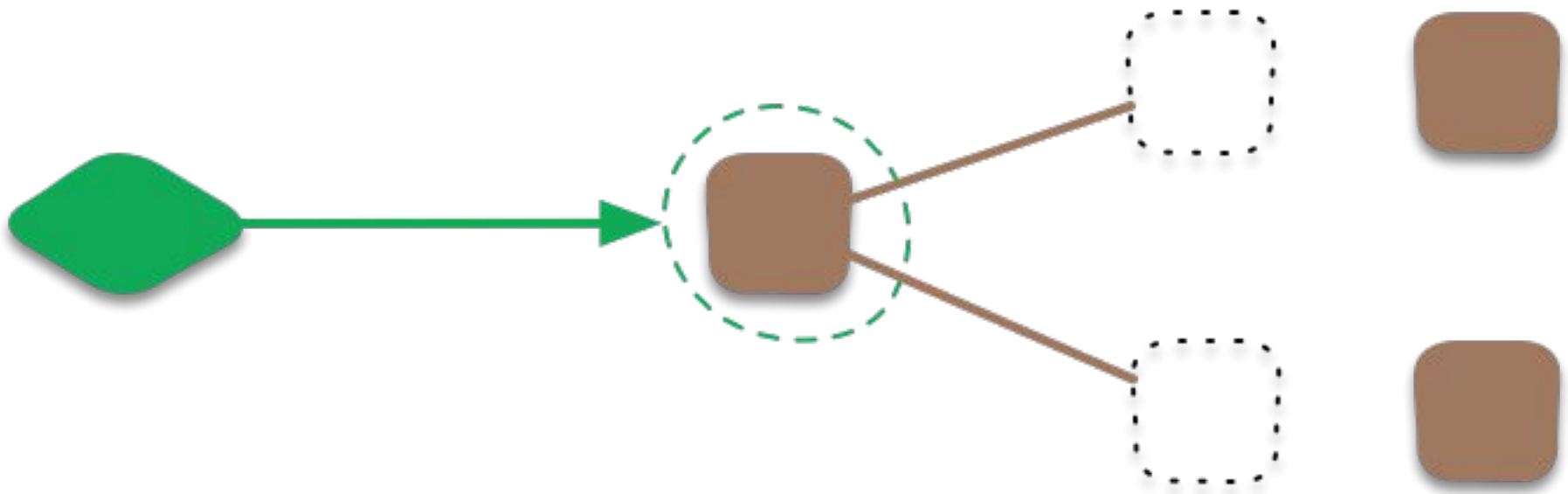
UNIT TESTS

Pruebas aisladas

JS

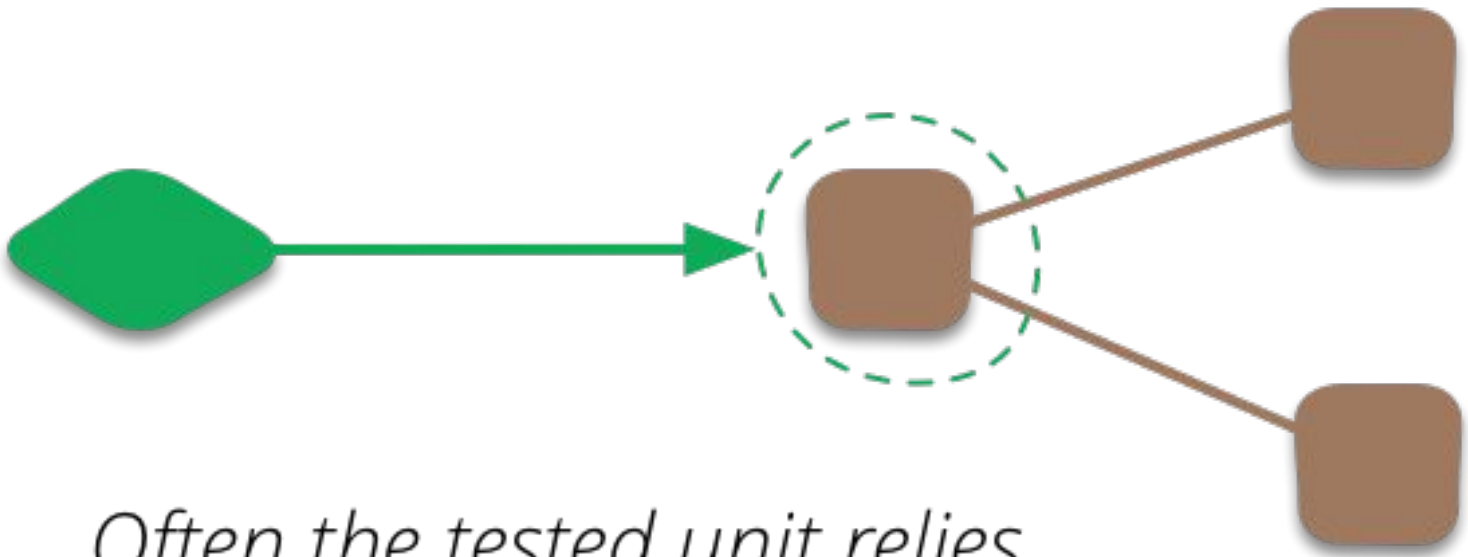


JS



Some unit testers prefer to isolate the tested unit

JS



Often the tested unit relies on other units to fulfill its behavior

REPORTE DE COBERTURA

Una medida porcentual.

JS

File	Statements
src	100%
src/app	100%
src/app/models	72.73%

JS

```
1  1x  export class Person {
2
3          constructor(
4  1x       public name: string,
5  1x       public lastname: string,
6  1x       public age: number
7          ) {}
8
9  1x      getFullName(): string {
10 1x       return `${this.name} ${this.lastname}`;
11         }
12
13 1x      getAgeInYears( years: number ): number {
14           if ( years > 0 ) {
15             return this.age + years;
16           }
17           return this.age;
18         }
19 1x   }
20
```

JS

JS

International Organization for Standardization

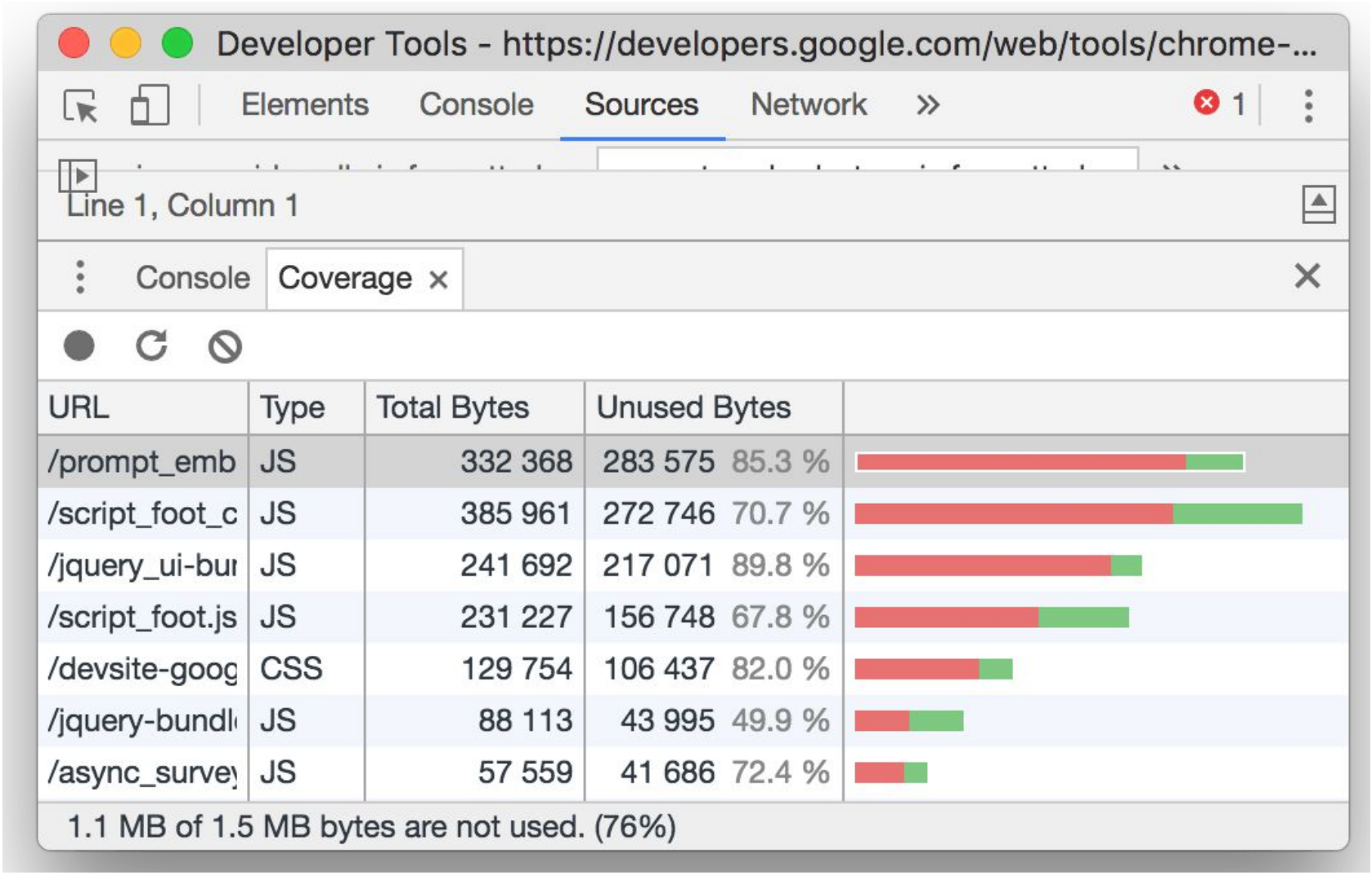
Es una medida ***porcentual*** *que evalúa el grado que un código ha sido ejecutado.*

100% Cobertura es igual a no tener errores

Un equilibrio para no hacer pruebas innecesarias

También es usada como técnica para eliminar código que ***nunca*** *va a ser ejecutado.*

JS


Developer Tools - https://developers.google.com/web/tools/chrome-...
Elements
Console
Sources
Network
1
Line 1, Column 1
Console
Coverage
URL
Type
Total Bytes
Unused Bytes
/prompt_emb JS 332 368 283 575 85.3 %
/script_foot_c JS 385 961 272 746 70.7 %
/jquery_ui-bur JS 241 692 217 071 89.8 %
/script_foot.js JS 231 227 156 748 67.8 %
/devsite-goog CSS 129 754 106 437 82.0 %
/jquery-bundl JS 88 113 43 995 49.9 %
/async_survey JS 57 559 41 686 72.4 %
1.1 MB of 1.5 MB bytes are not used. (76%)

API PROJECT

Proyecto

DOUBLES

Mocking, Stub, fake, dummies, doubles

Dummy

Son datos ficticios y se usan para rellenar información.

Fake

Simulan un objeto real y sirven para suplantar ciertos datos y comportamientos.

Stubs

Proveen respuestas preparadas y se pueden llamar durante el test para simular un comportamiento.

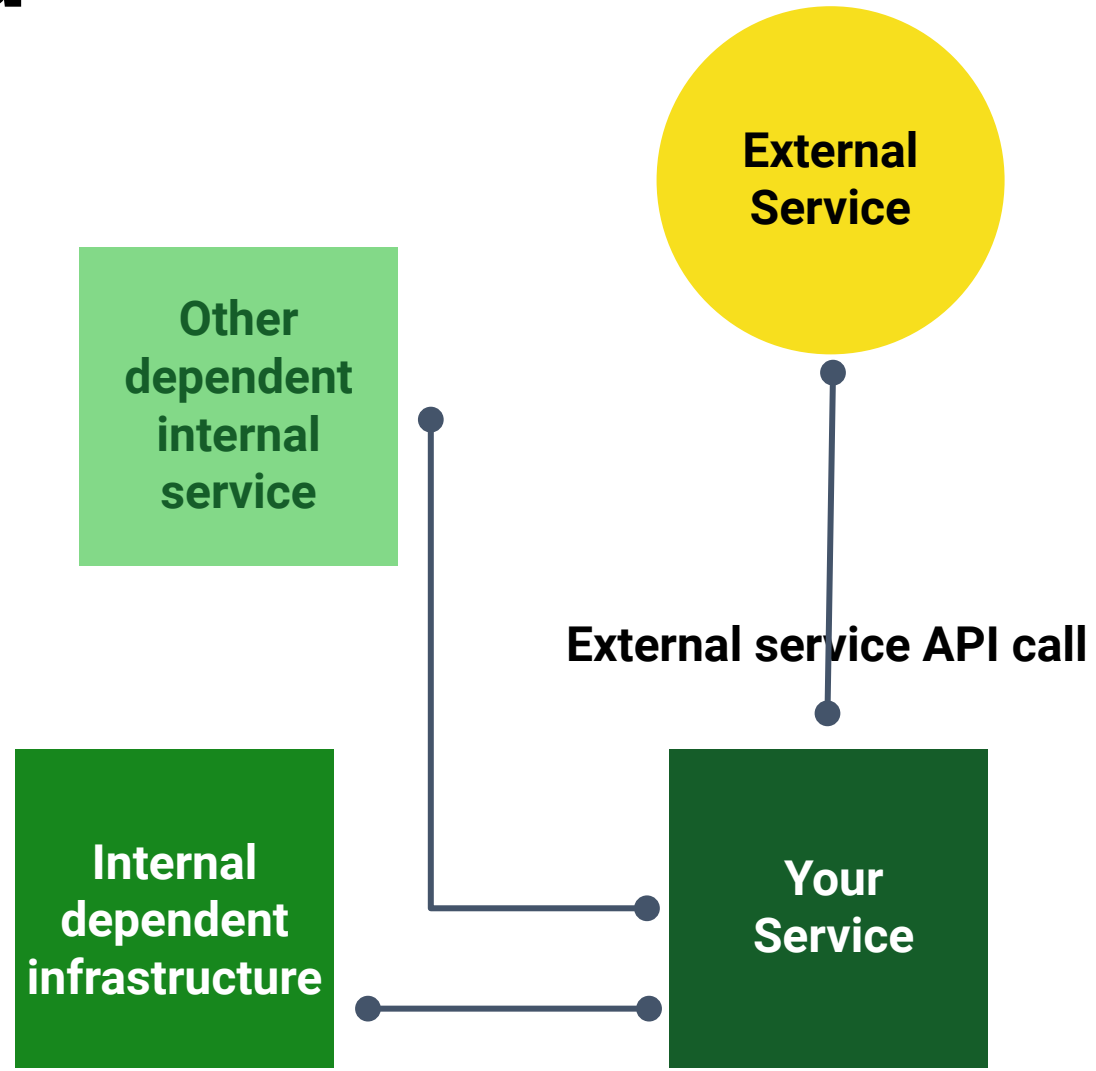
Spies

Pueden ser Stub pero puedo recolectar información de como fue llamado.

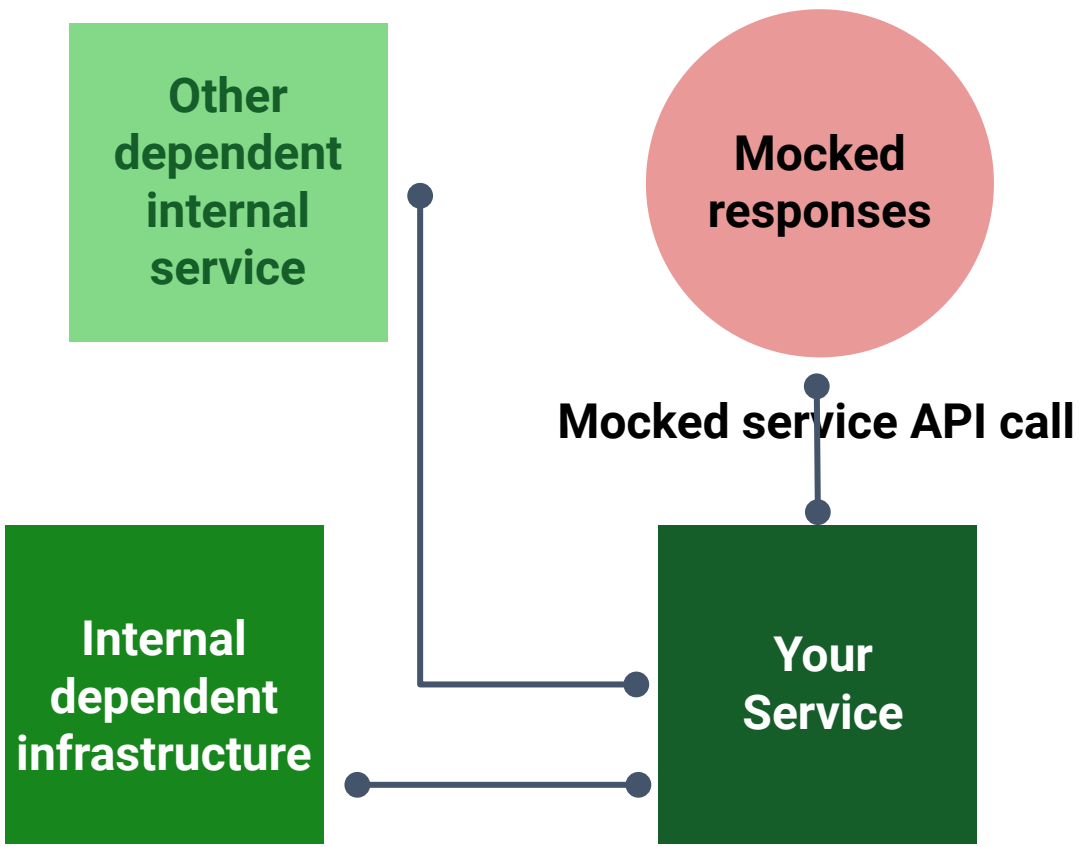
Mocks

Stub + Spies, a veces pueden ya estar pre-programados.

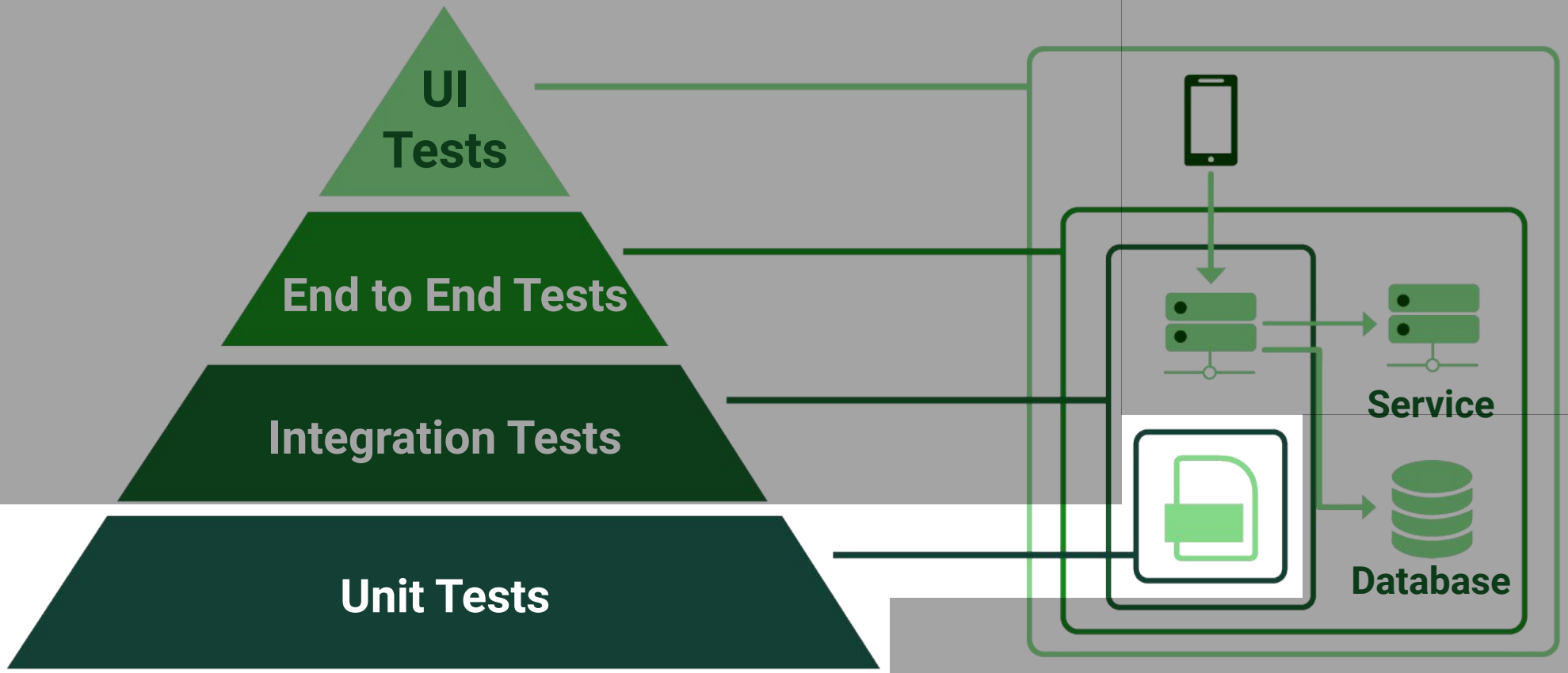
JS Prod


External Service
Other dependent internal service
External service API call
Internal dependent infrastructure
Your Service

Test Environment



JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

MOCKING

Suplantando MongoLib

SPIES

Pruebas de caja blanca

Generando Fakes

Usando FakerJS

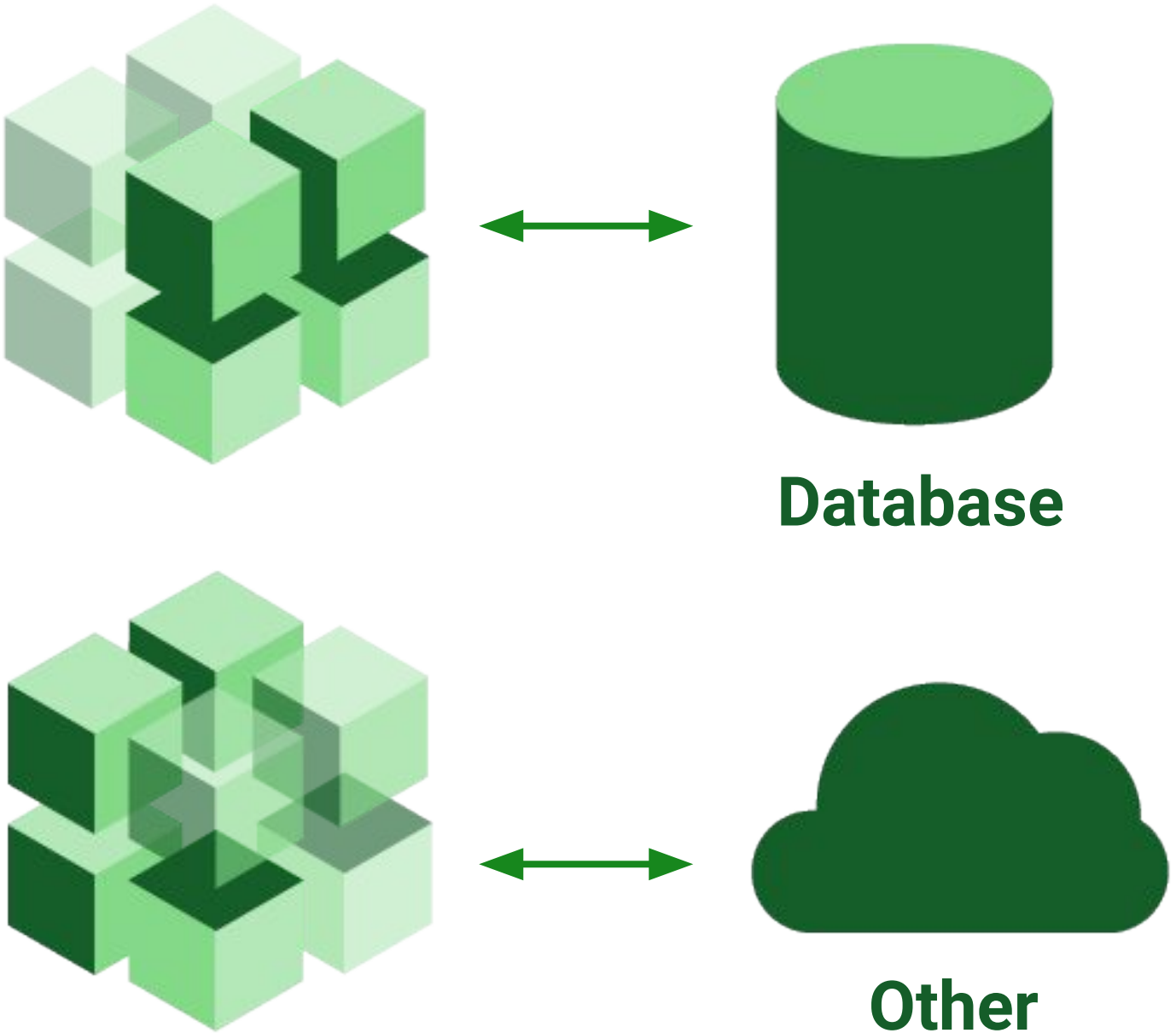
Fake

Simulan un objeto real y sirven para suplantar ciertos datos y comportamientos.

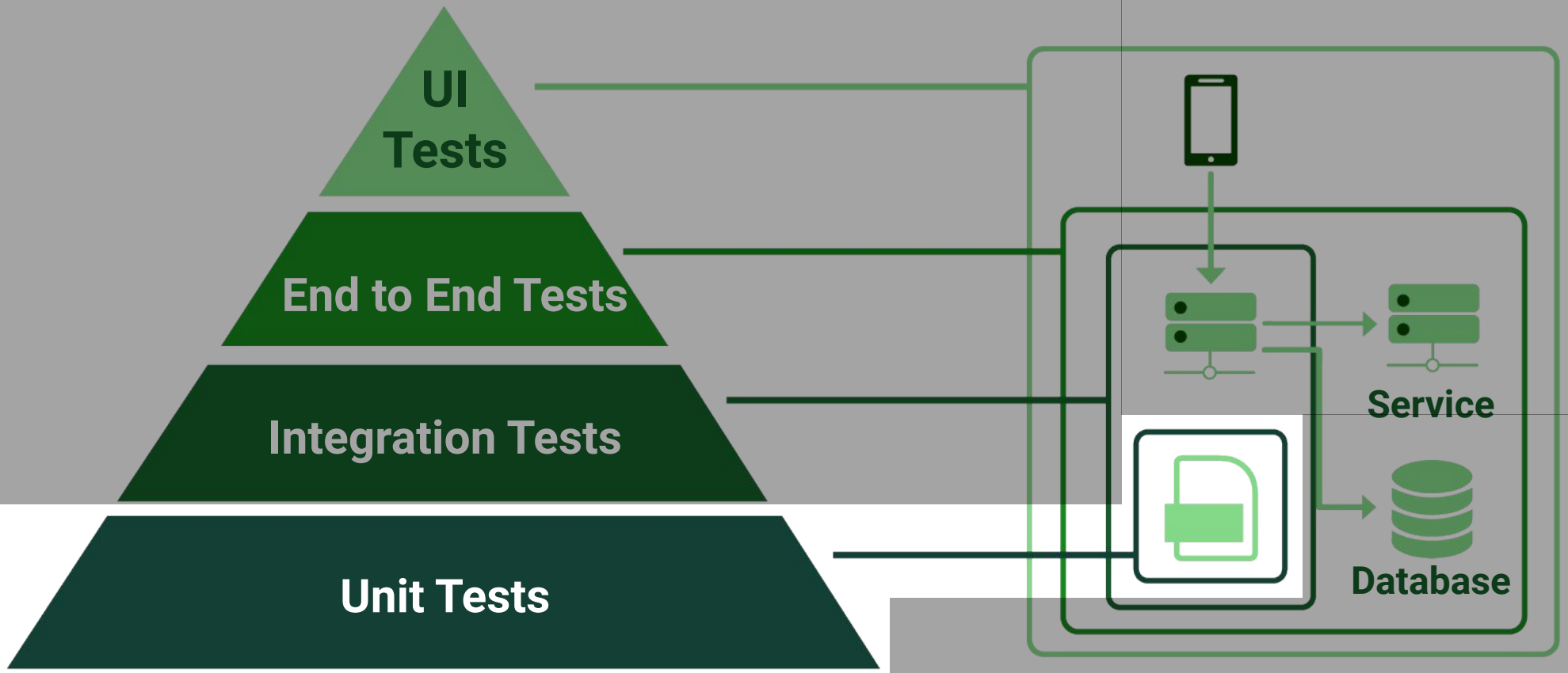
SUPERTEST

Integration test & e2e

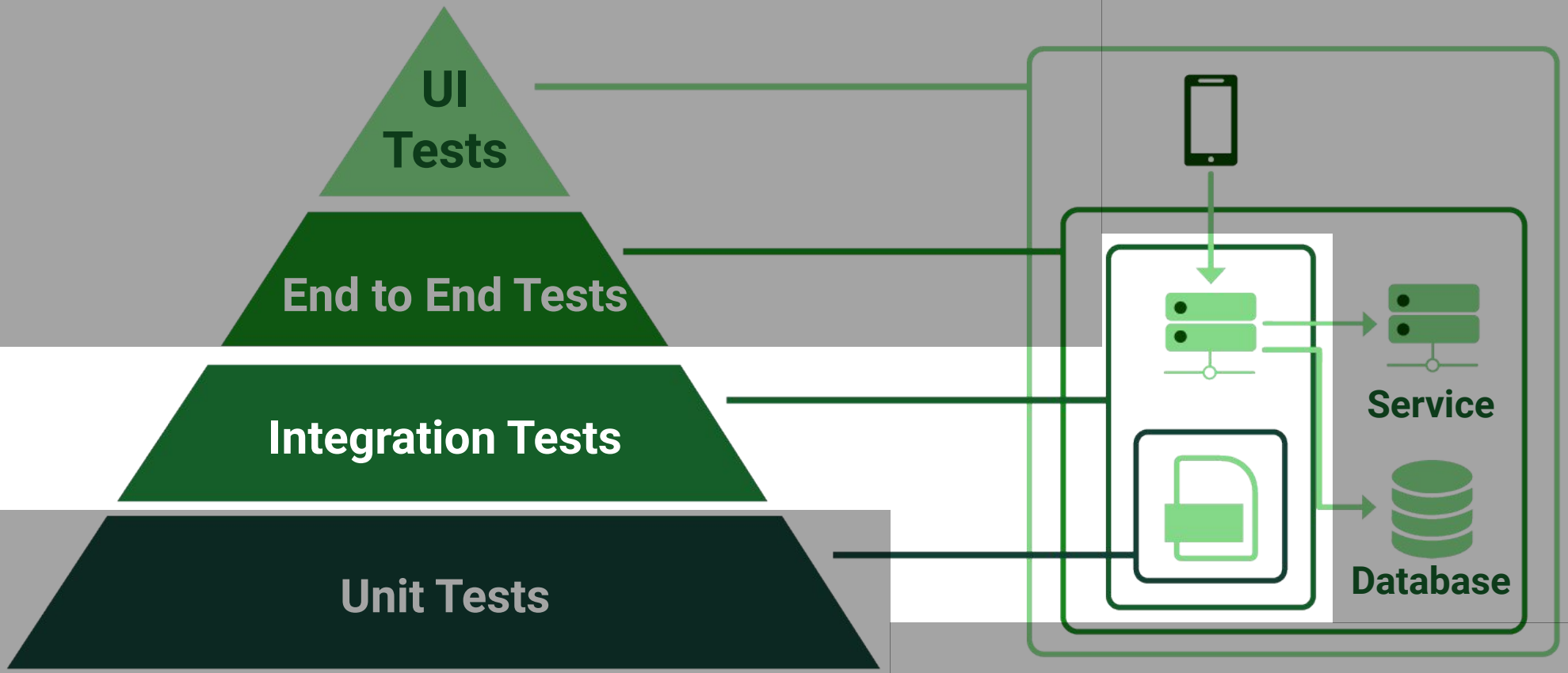
JS



JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

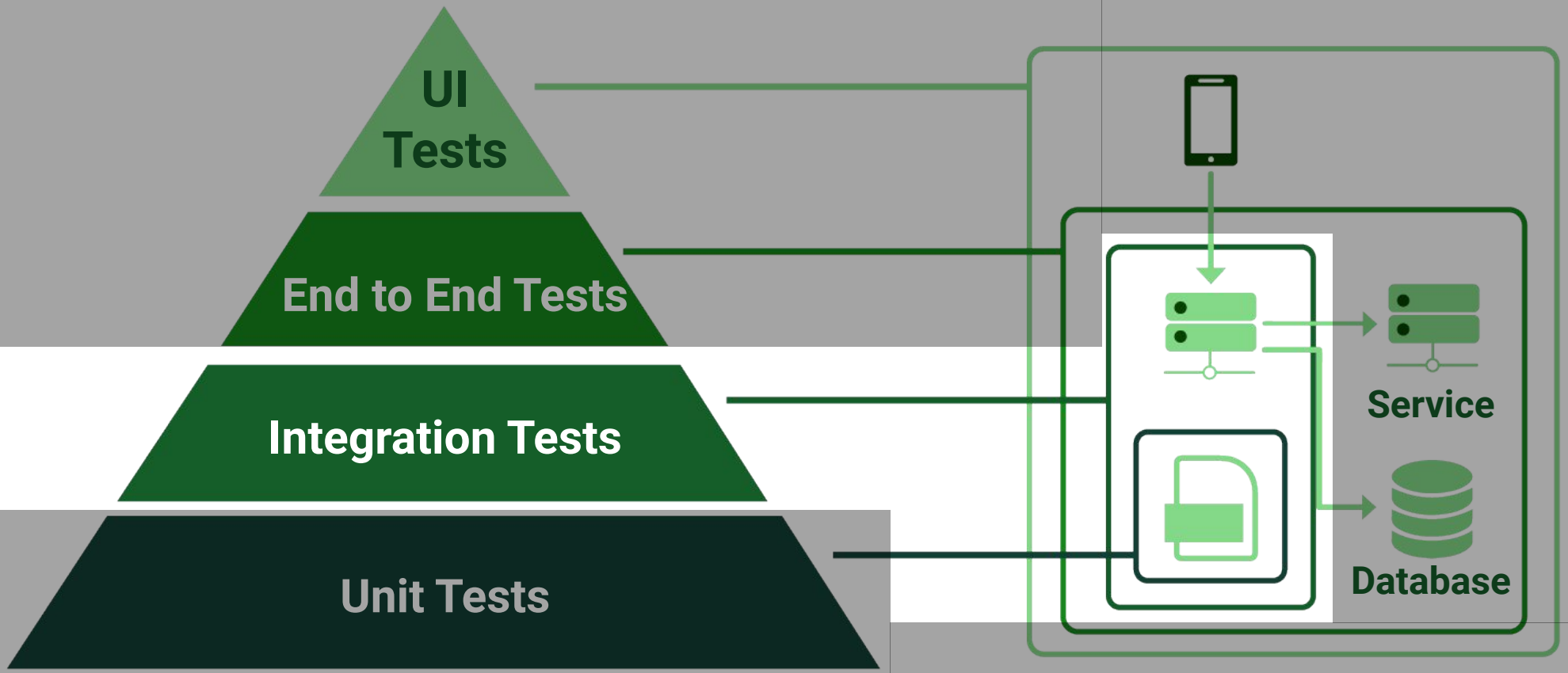
JS

API Testing

INTEGRATION

Integration test & e2e

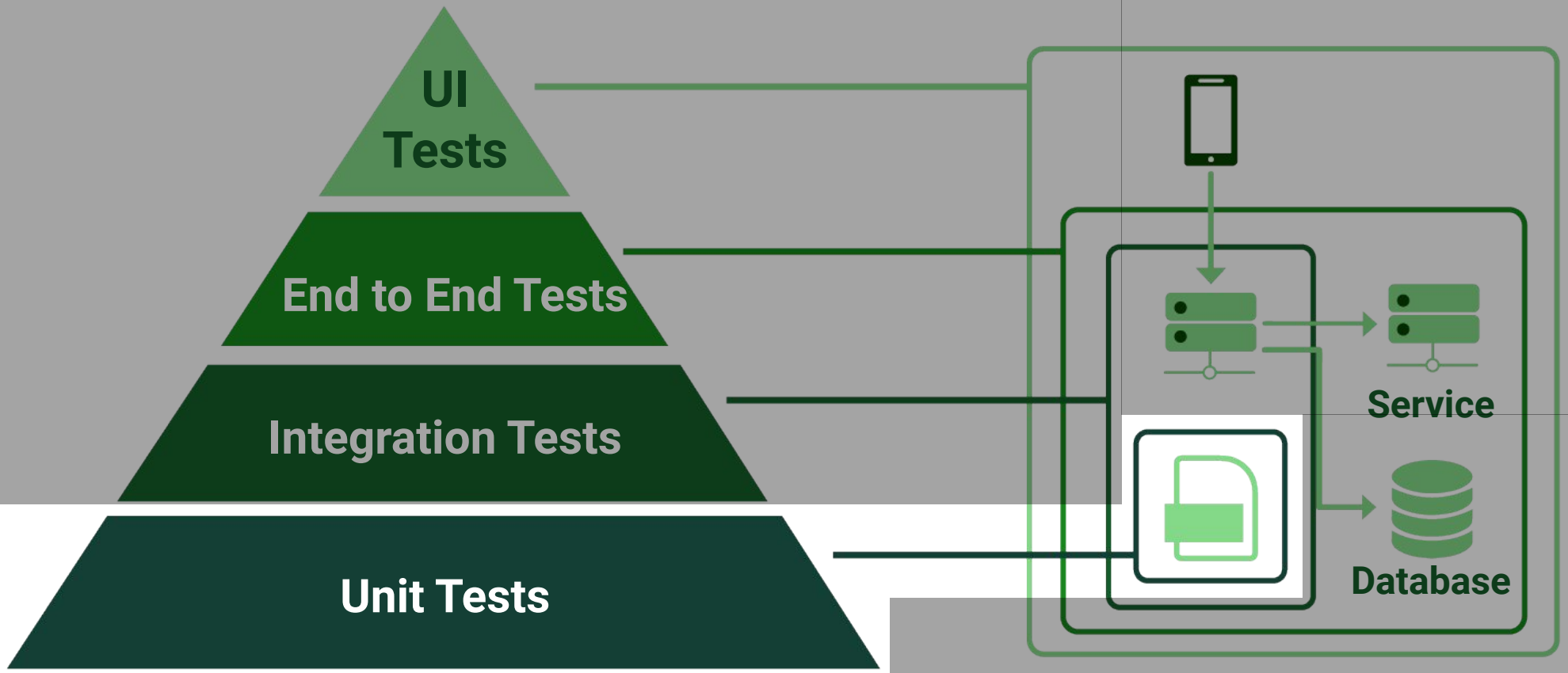
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

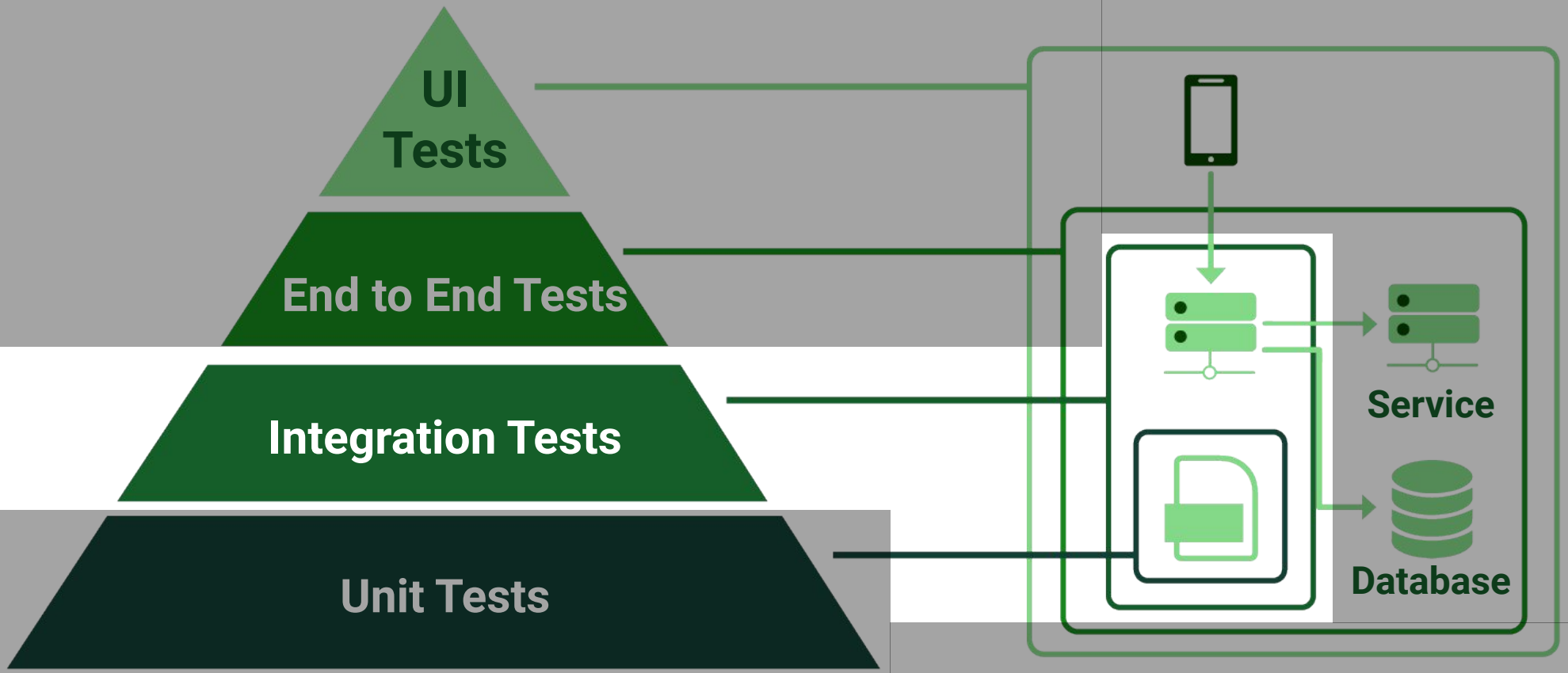
E2E

Integration test & e2e

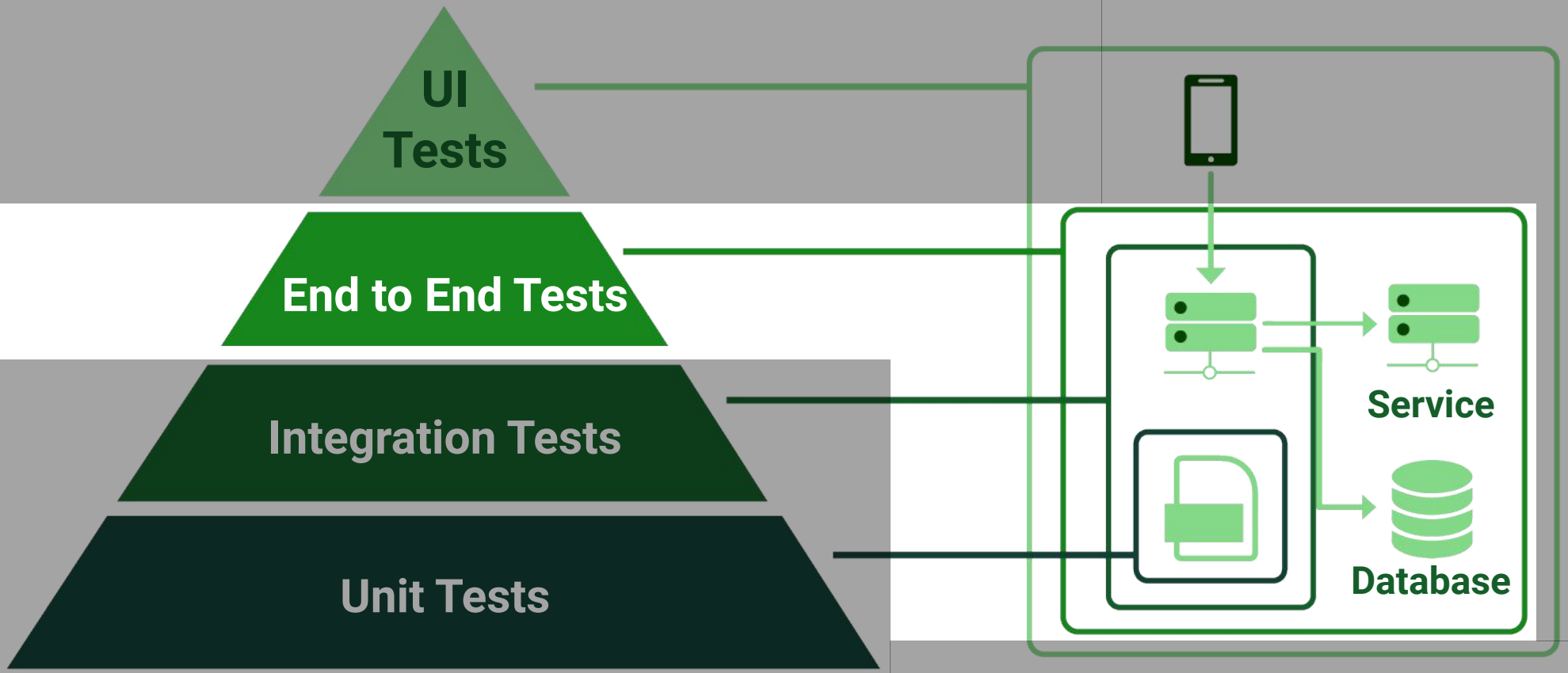
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

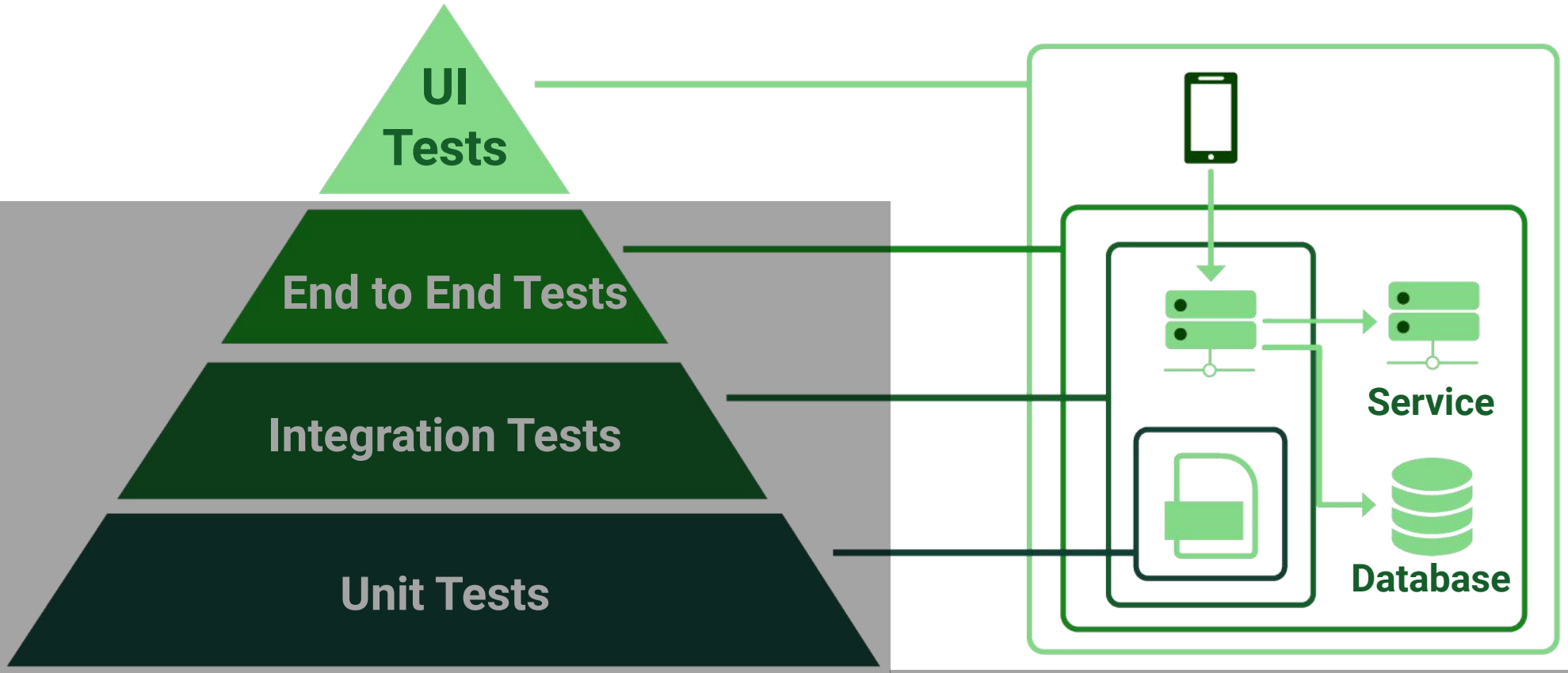
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

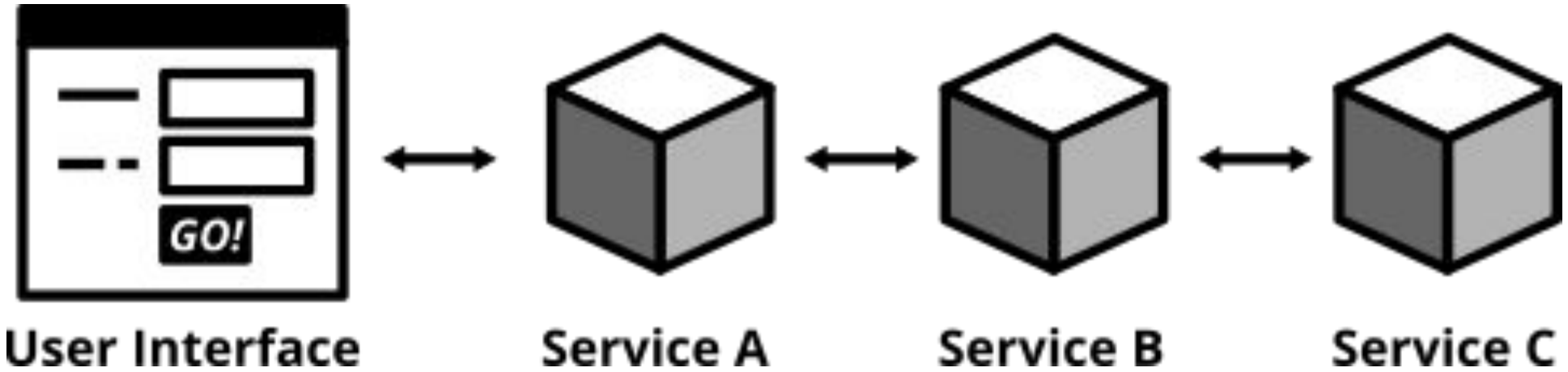
UI TESTS

Probando con devices

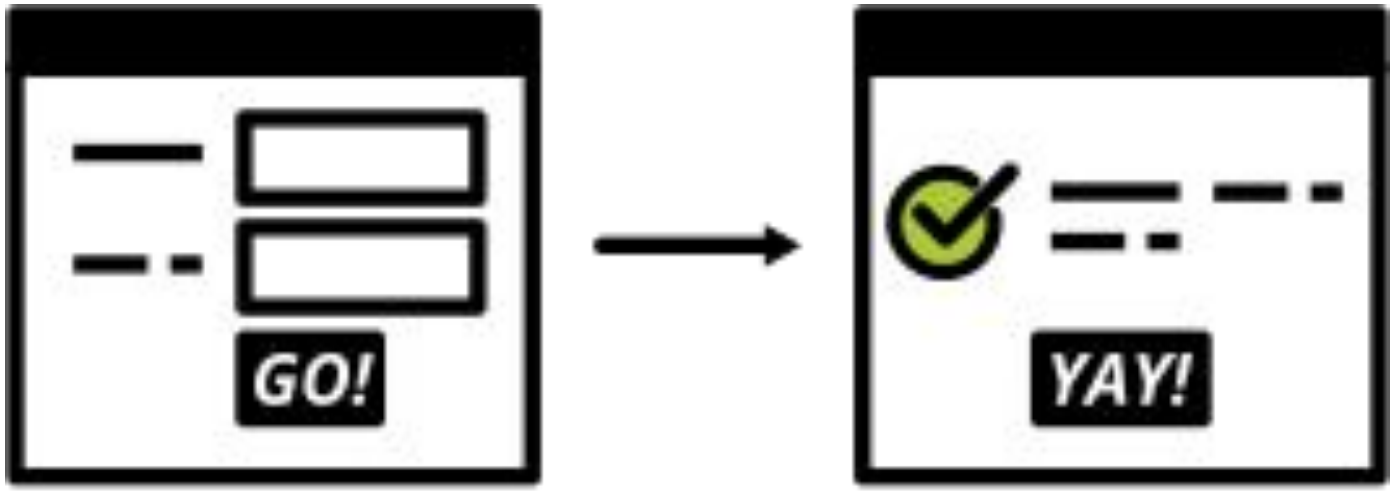
JS


UI
Tests
End to End Tests
Integration Tests
Unit Tests
Service
Database

JS



JS



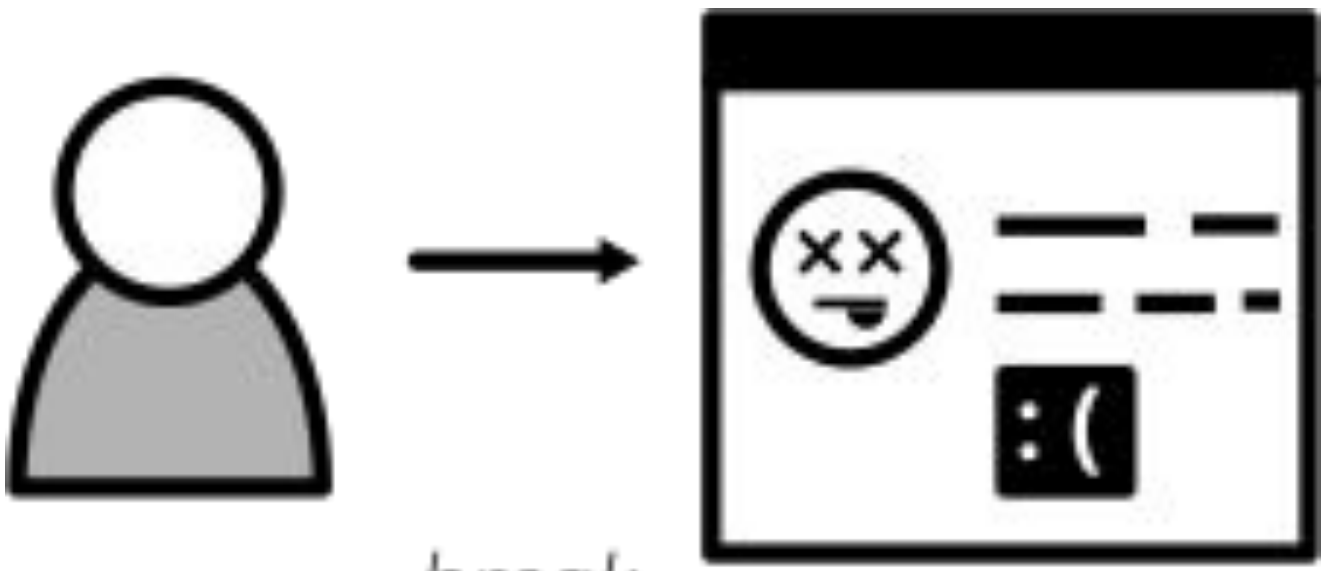
<tap, click>

check

TEST EXPLORATORIO

Una ayuda estratégica.

JS

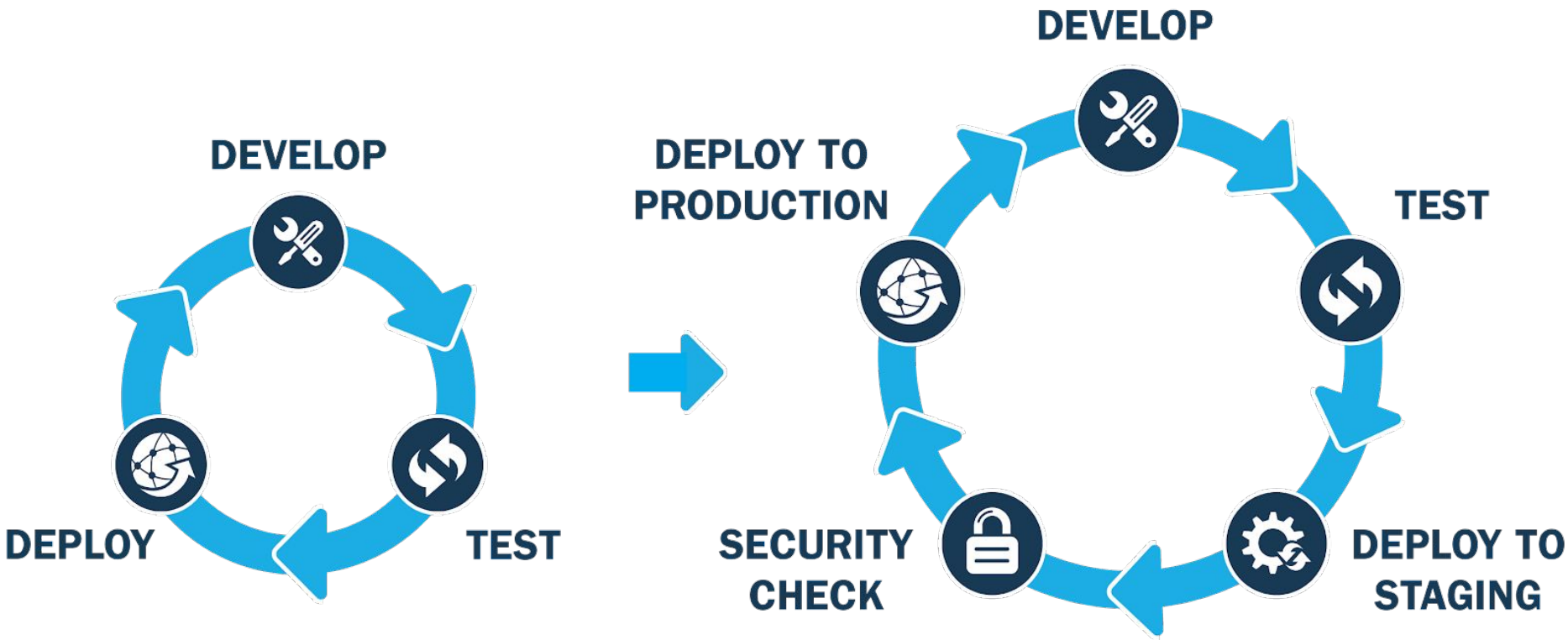


MOCKING

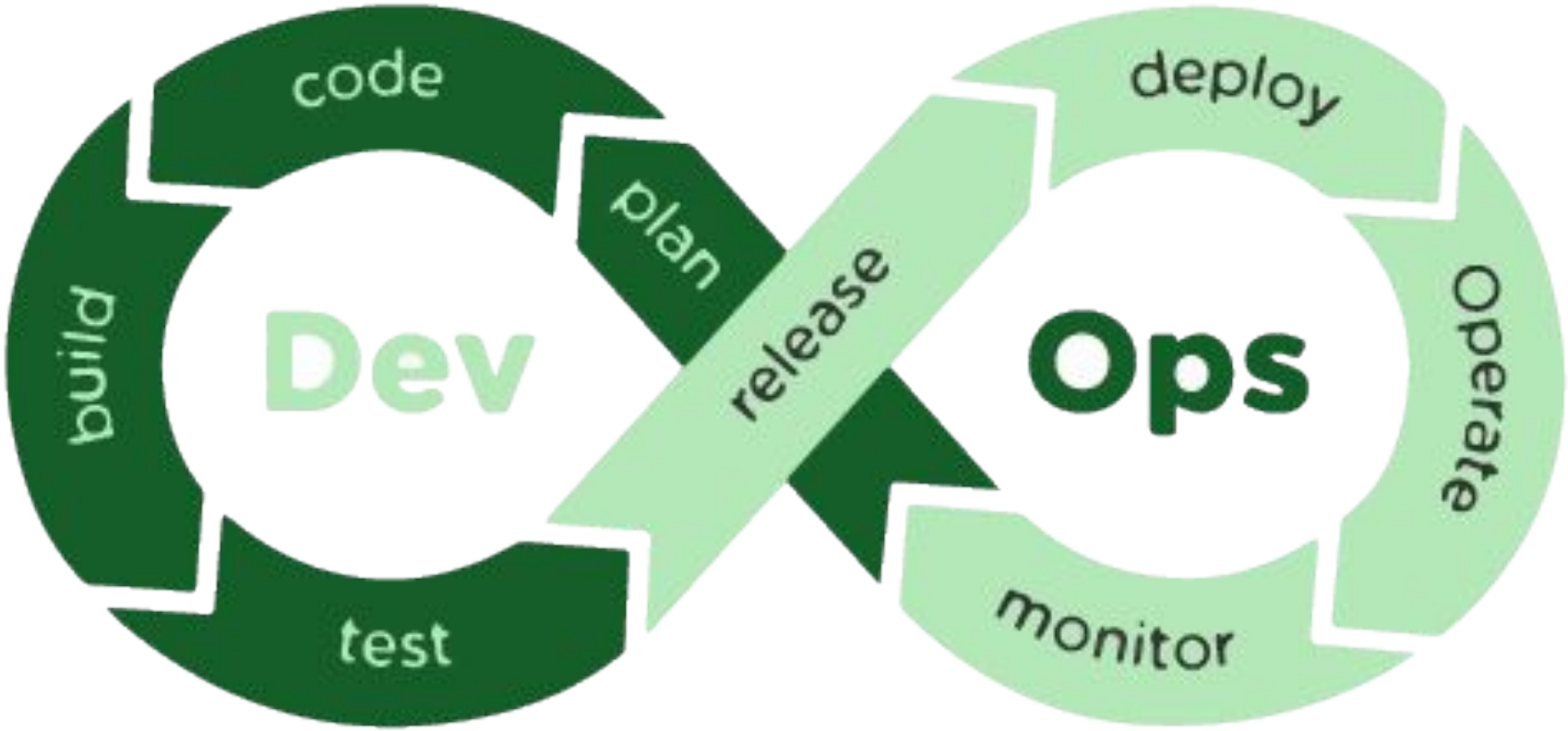
Una ayuda estratégica.

AUTOMATIZAR

con GitHub Actions


DEVELOP
DEPLOY
TEST
DEVELOP
DEPLOY TO
PRODUCTION
TEST
SECURITY
CHECK
DEPLOY TO
STAGING

JS


code
plan
build
Dev
test
release
deploy
Ops
operate
monitor