

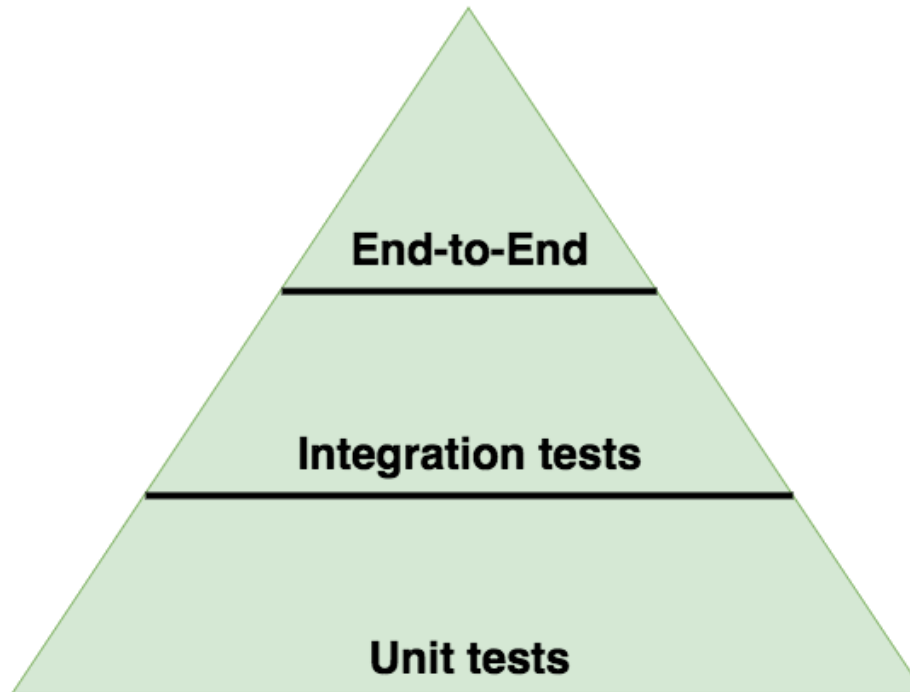
HENRY

A bright yellow beam of light originates from the left edge of the frame and points towards the letter 'R' in the word 'HENRY'. The beam is wider on the left and tapers as it moves to the right. The word 'HENRY' is written in a bold, black, sans-serif font.



Testing

¿Qué cosas testear? ¿Cuántos tests hacer?



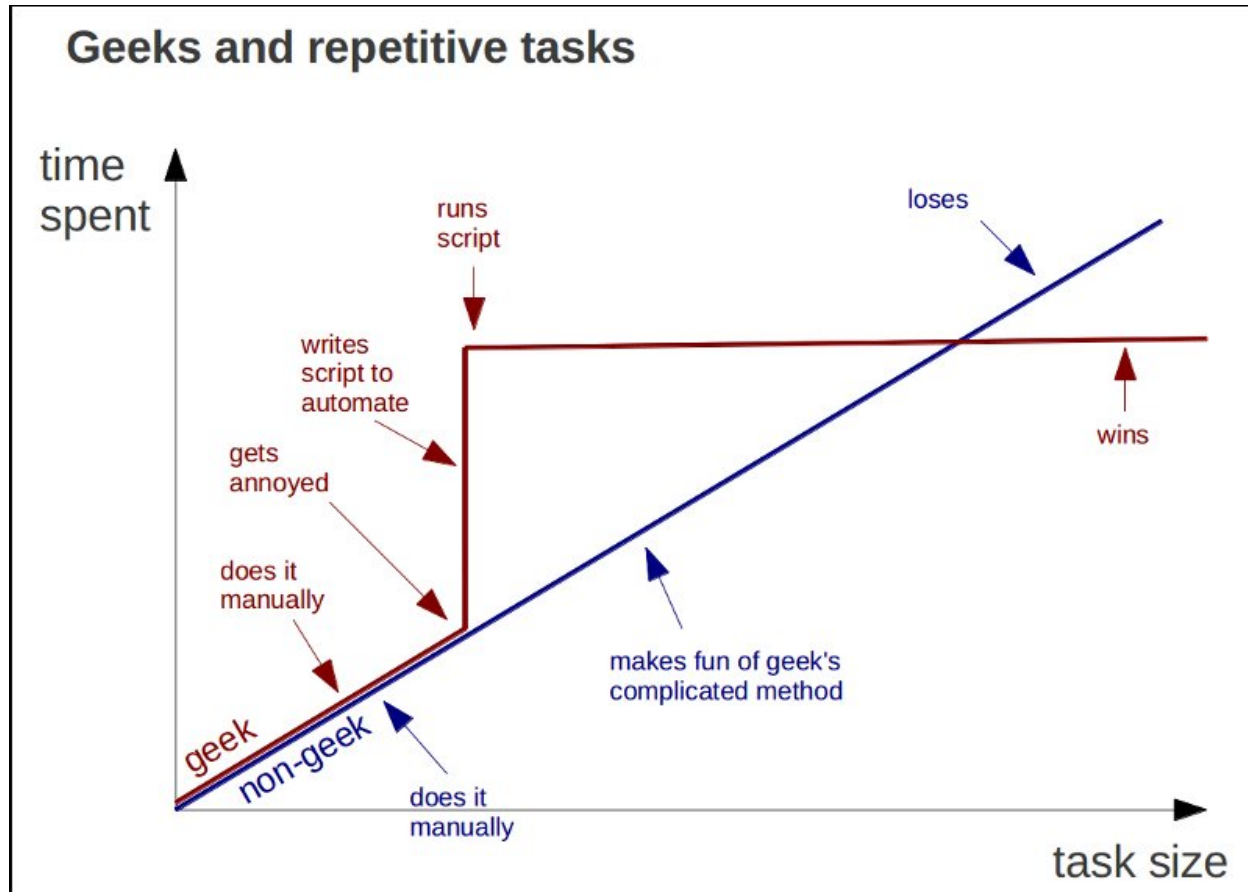


Unit Testing

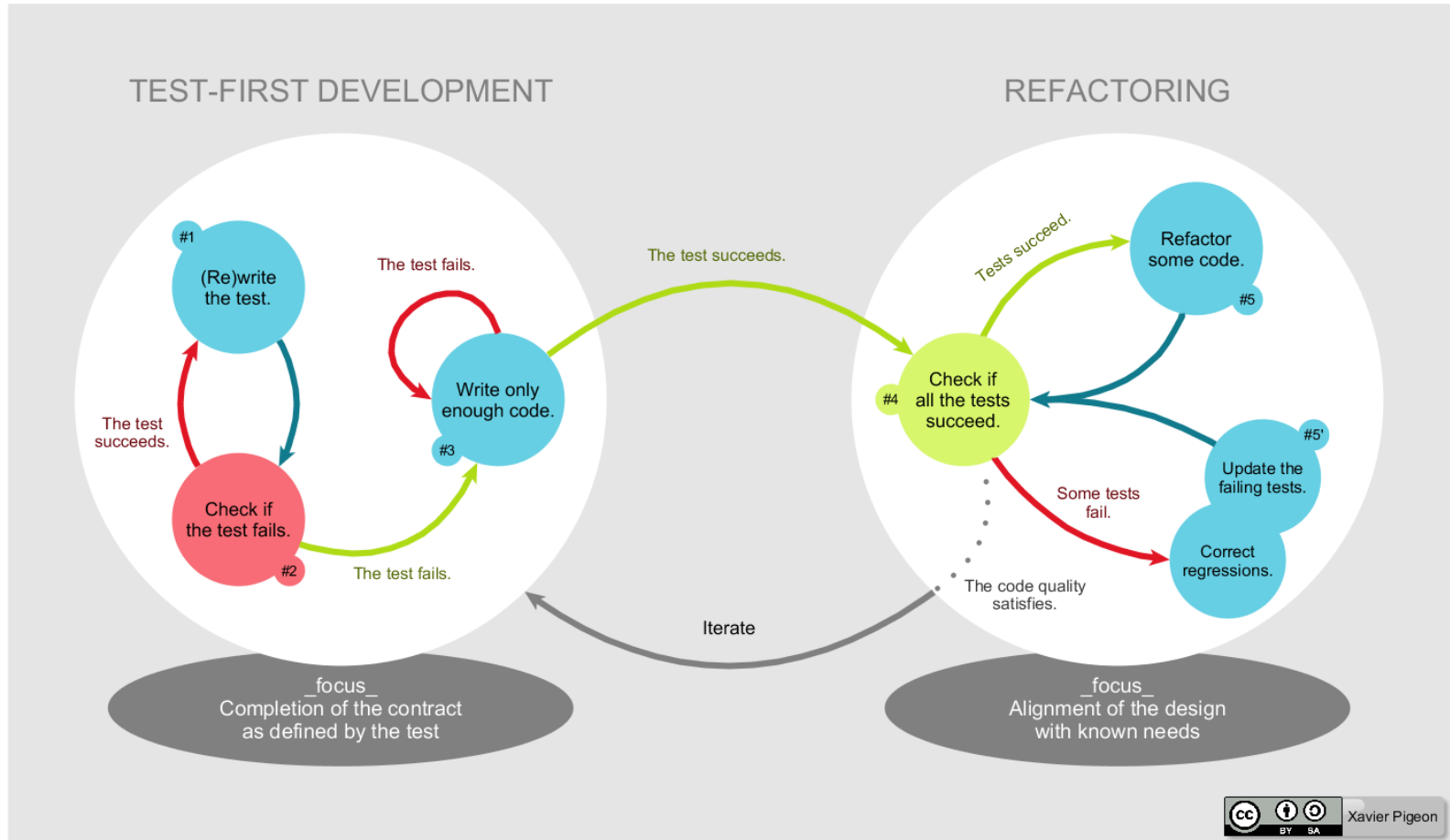
Un buen test unitario debería ser:

- Completamente automatizable
- Poder ejecutarse en cualquier orden en conjunto con otros tests.
- **Siempre** retorna el mismo resultado, no importa cuantas veces lo corra.
- Es rápido
- Testea un solo concepto lógico del sistema
- Es fácil de entender al leerlo
- Es fácil de mantener

Unit Testing



TDD



Testing Frameworks




Jest



```
1 // Instalation
2 npm install --save-dev jest
3
4 // package.json configuration
5 ...
6
7   "scripts": {
8     "test": "jest"
9   }
10
11 ...
```



Jest



```
1 // Run only tests matching
2 jest test-pattern // pattern
3 jest path/to/test.js // filename
4 jest -t name-spec // describe or test name
5
6 // Run in watch mode
7 jest --watch // by default only changed files
8 jest --watchAll
9
10 // Show summary of each test file
11 // Running one file only --> automatically verbose
12 jest --verbose
```

Jest

```
1 // sum.js
2
3 function sum(a, b) {
4   return a + b;
5 }
6
7 module.exports = sum;
```

```
1 // sum.test.js
2
3 const sum = require('./sum');
4
5 // it === test
6 it('should return 8 if adding 3 and 5', () => {
7   expect(sum(3, 5)).toBe(8);
8 });
```





Jest



expect

toBe

toEqual

toBeNull toBeUndefined toBeDefined

toBeTruthy

true

toBeFalsy

false

Jest

toBeGreaterThan toBeGreaterThanOrEqual
toBeLessThan toBeLessThanOrEqual

toBeCloseTo

toMatch

toContain

toThrow

Y más ...

<matchers.test.js />



Jest



`xit`

`describe`

`xdescribe`

`it.only`

Asynchronous Code



```
1 it('should resolve to Henry Promise', () => {  
2   promisifiedFunction(false).then(data => {  
3     expect(data).toBe('asd');  
4   });  
5 });
```



PASS ./promise.test.js
✓ should resolve to Henry Promise (1 ms)



```
1 it('should resolve to Henry Promise', () => {  
2   return promisifiedFunction(true).then(data => {  
3     expect(data).toBe('Henry Promise');  
4   });  
5 });
```



return

Asynchronous Code



```
1 it('should reject to Rejected Promise', () => {  
2   return promisifiedFunction(true).catch(e => {  
3     expect(e).toMatch('Rejected Promise')  
4   });  
5 });
```



```
PASS ./promise.test.js  
✓ should reject to Rejected Promise (2004 ms)
```



```
1 it('should reject to Rejected Promise', () => {  
2   expect.assertions(1);  
3   return promisifiedFunction(false).catch(e => {  
4     expect(e).toMatch('Rejected Promise')  
5   });  
6 });
```



+ sobre
assertions



Dispositivos assertions

Hooks

```
1 beforeAll(() => {  
2   ...  
3 });  
4  
5 beforeEach(() => {  
6   ...  
7 });  
8  
9 afterEach(() => {  
10  ...  
11 });  
12  
13 afterAll(() => {  
14  ...  
15 });
```



Mock Functions

```
1 const mockFunction = jest.fn(person => person.age > 18);
```

.mock.calls

.mock.results

Pss, quieres
saber **más...** ?



<mock.test.js />

Supertest

statusCode

response

type

.listen

