

HENRY

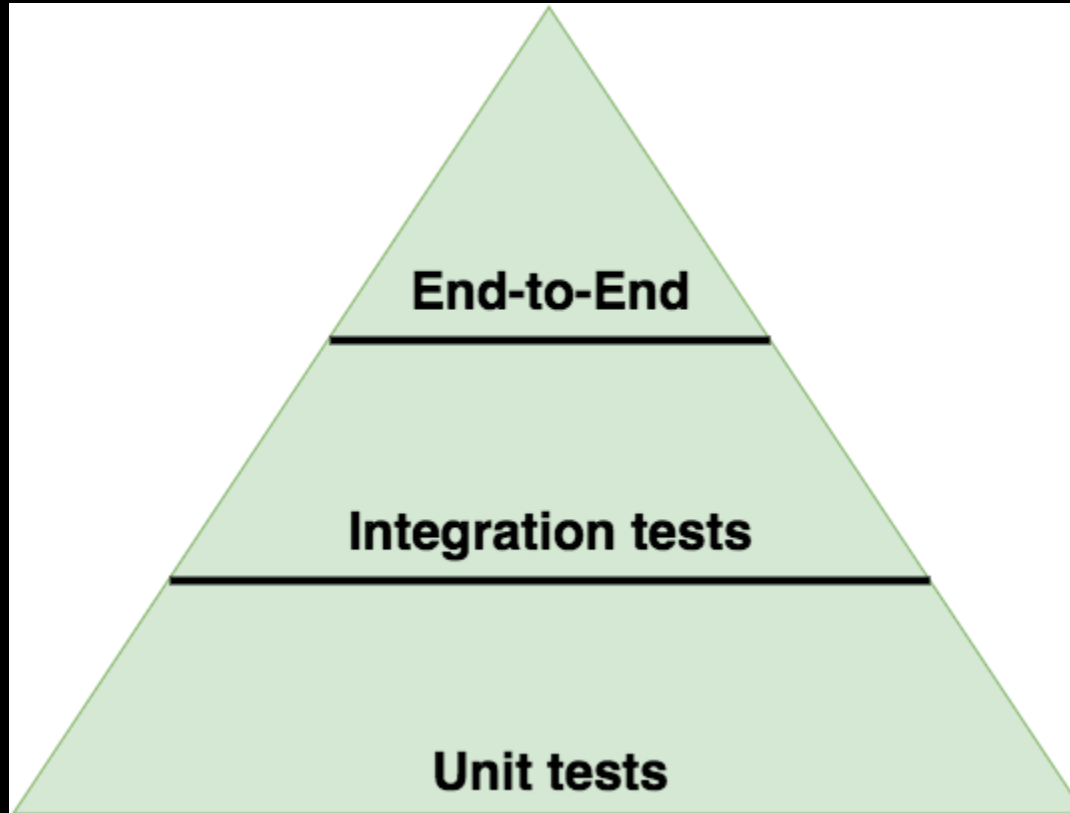
A bright yellow beam of light originates from the left edge of the frame and points towards the letter 'R' in the word 'HENRY'. The beam is wider on the left and tapers as it moves to the right. The word 'HENRY' is written in a bold, black, sans-serif font. The beam appears to be shining on the 'R', which has a small white highlight on its upper curve.



Testing



Qué cosas testear? Cuantos tests hacer?





Unit Testing

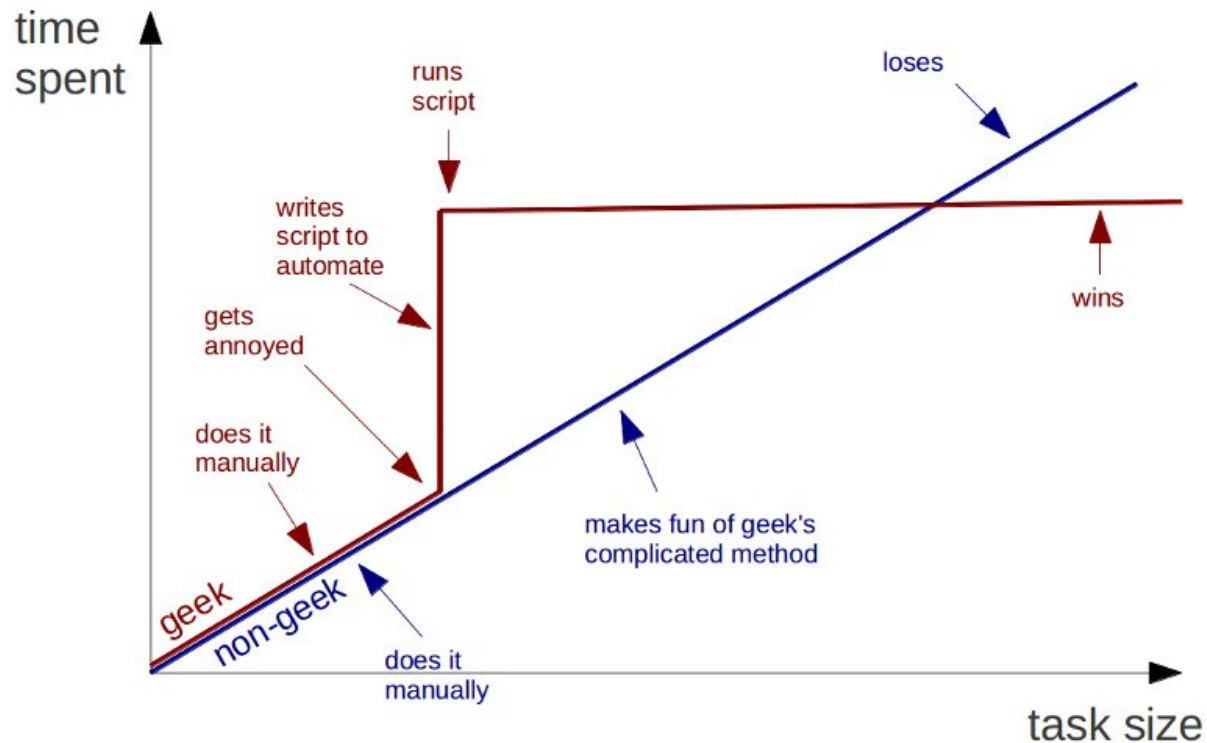
Un buen test unitario debería ser:

- 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

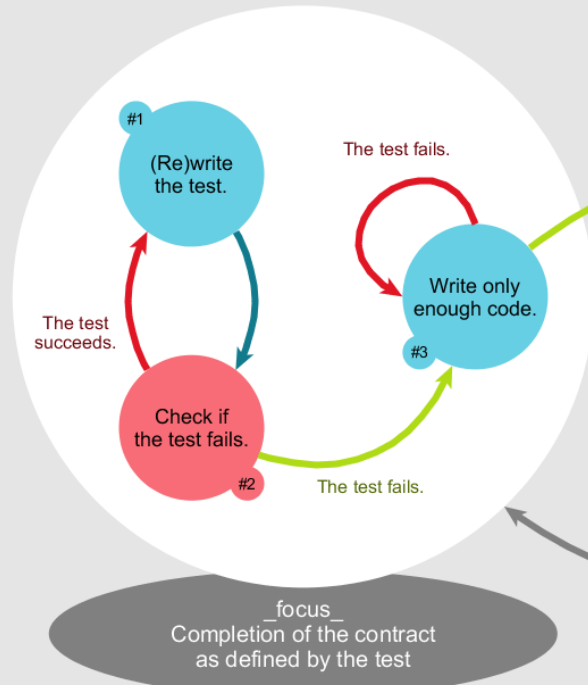
Geeks and repetitive tasks



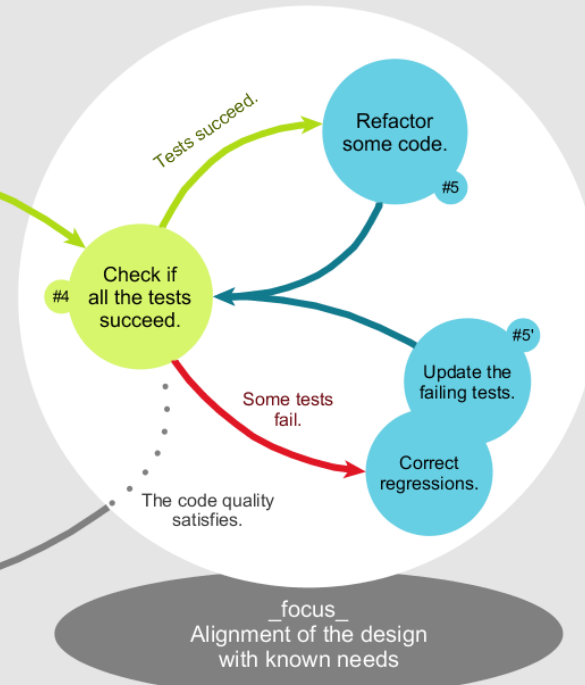


TDD

TEST-FIRST DEVELOPMENT



REFACTORING



Iterate



Frameworks Test Runner

```
1 // Mocha
2
3 var assert = require('assert');
4 describe('Array', function() {
5   describe('#indexOf()', function() {
6     it('should return -1 when the value is not present', function() {
7       assert.equal(-1, [1,2,3].indexOf(4));
8     });
9   });
10  describe('#slice()', function() {
11    it('should return [2,3] when args are (1,3)', function() {
12      assert.equal([2,3], [1,2,3].slice(1,3));
13    });
14  });
15  describe('#slice()', function() {
16    it('should return [2,3] when args are (1,3)', function() {
17      assert.deepEqual([2,3], [1,2,3].slice(1,3));
18    });
19  });
20 });
```



Frameworks

Test Runner - Hooks

```
1 // Mocha
2 //
3 var chai = require('chai');
4 describe('Array', function() {
5
6   before(function() {
7     console.log('Hola, probando hook');
8   });
9   afterEach(function() {
10     console.log('Cuantas veces aparezco!');
11   });
12   describe('#indexOf()', function() {
13     it('should return -1 when the value is not present', function() {
14       chai.assert.equal(-1, [1,2,3].indexOf(4));
15     });
16   });
17   describe('#slice()', function() {
18     it('should return [2,3] when args are (1,3)', function() {
19       chai.assert.deepEqual([2,3], [1,2,3].slice(1,3));
20     });
21   });
22 });
```




Frameworks Assertion Library

```
1 // Chai
2 var chai = require('chai');
3
4 describe('Array', function() {
5   describe('#indexOf()', function() {
6     it('should return -1 when the value is not present', function() {
7       chai.assert.equal(-1, [1,2,3].indexOf(4));
8     });
9   });
10  describe('#slice()', function() {
11    it('should return [2,3] when args are (1,3)', function() {
12      chai.assert.deepEqual([2,3], [1,2,3].slice(1,3));
13    });
14  });
15 });
```



< DEMO />