Test Driven Development (TDD)

By Miami Larry, Web Developer

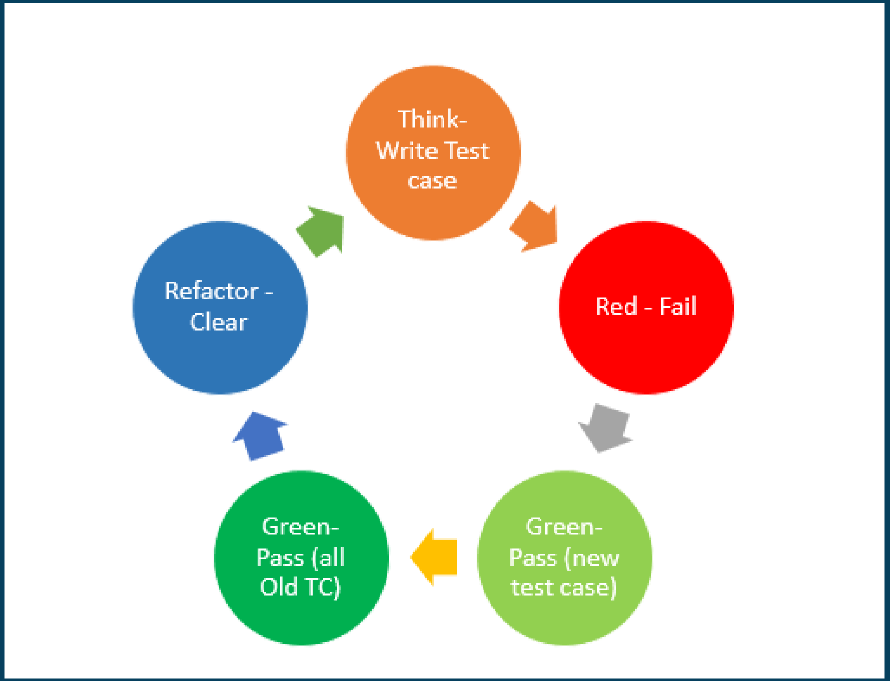
What is TDD?

Test-driven development is a programming methodology with which one can tackle the design, implementation, and testing of units of code, and to some extent the expected functionality of a program.

Why TDD?

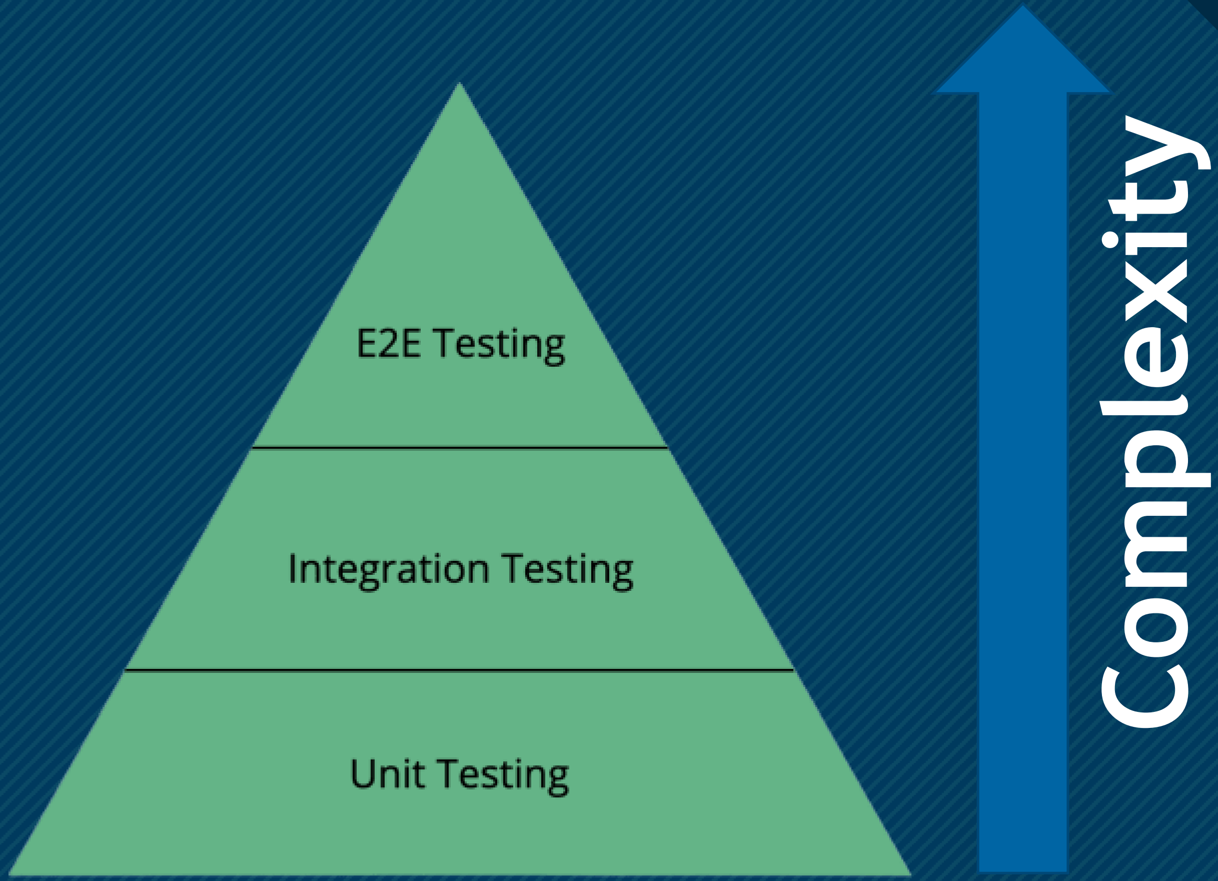
- Makes code easier to maintain and refactor code.
- It helps developers find mistakes that would waste everyone's time if they were found in QA.

The testing cycle



- Think and write test cases - this step ensures that you understand the functionality that is required
- Red – In this step, you try to run your test. You have no implementation code, so your test should fail.
- Green – Write the minimum code required to get the test to pass
- Green – Ensure that no old tests fail.
- Refactor - Refactor to ensure functionality is intact and the code is refined.
- Repeat this cycle - Steps 1 - 5 are repeated multiple times so that all the features are covered in TDD cycles

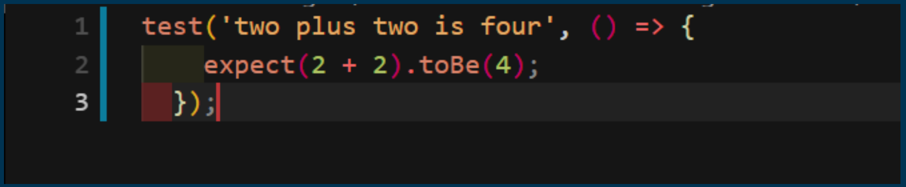
Type of tests



Unit Testing (Jest)

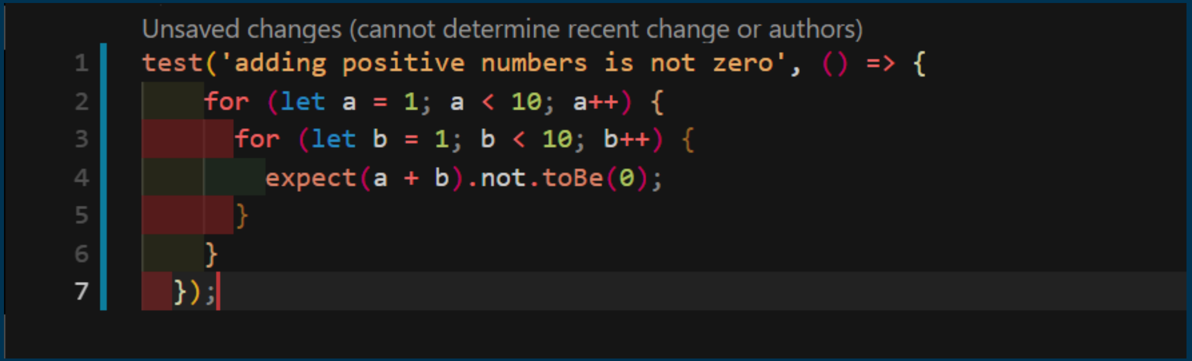
Common matchers

.toBe



```
test('two plus two is four', () => {
  expect(2 + 2).toBe(4);
});
```

.not.toBe



Unsaved changes (cannot determine recent change or authors)

```
test('adding positive numbers is not zero', () => {
  for (let a = 1; a < 10; a++) {
    for (let b = 1; b < 10; b++) {
      expect(a + b).not.toBe(0);
    }
  }
});
```

Truthiness

- toBeNull matches only null
- toBeUndefined matches only undefined
- toBeDefined is the opposite of toBeUndefined
- toBeTruthy matches anything that an if statement treats as true
- toBeFalsy matches anything that an if statement treats as false

"If something is hard to test, it's probably not your test's fault."
- Me

Thank You 2