

Testing in NodeJS



Objectives

- Get the definition and importance of unit testing
- Understand what TDD and BDD are. Know the difference between them
- Get familiar with common packages used in unit testing

Agenda

- Why do we need test?
- What is TDD?
- What is BDD?
- Common package use for testing

Why?

- Unit tests help you verify functionality as you add it. You do it at the level which makes sense, often testing in isolation

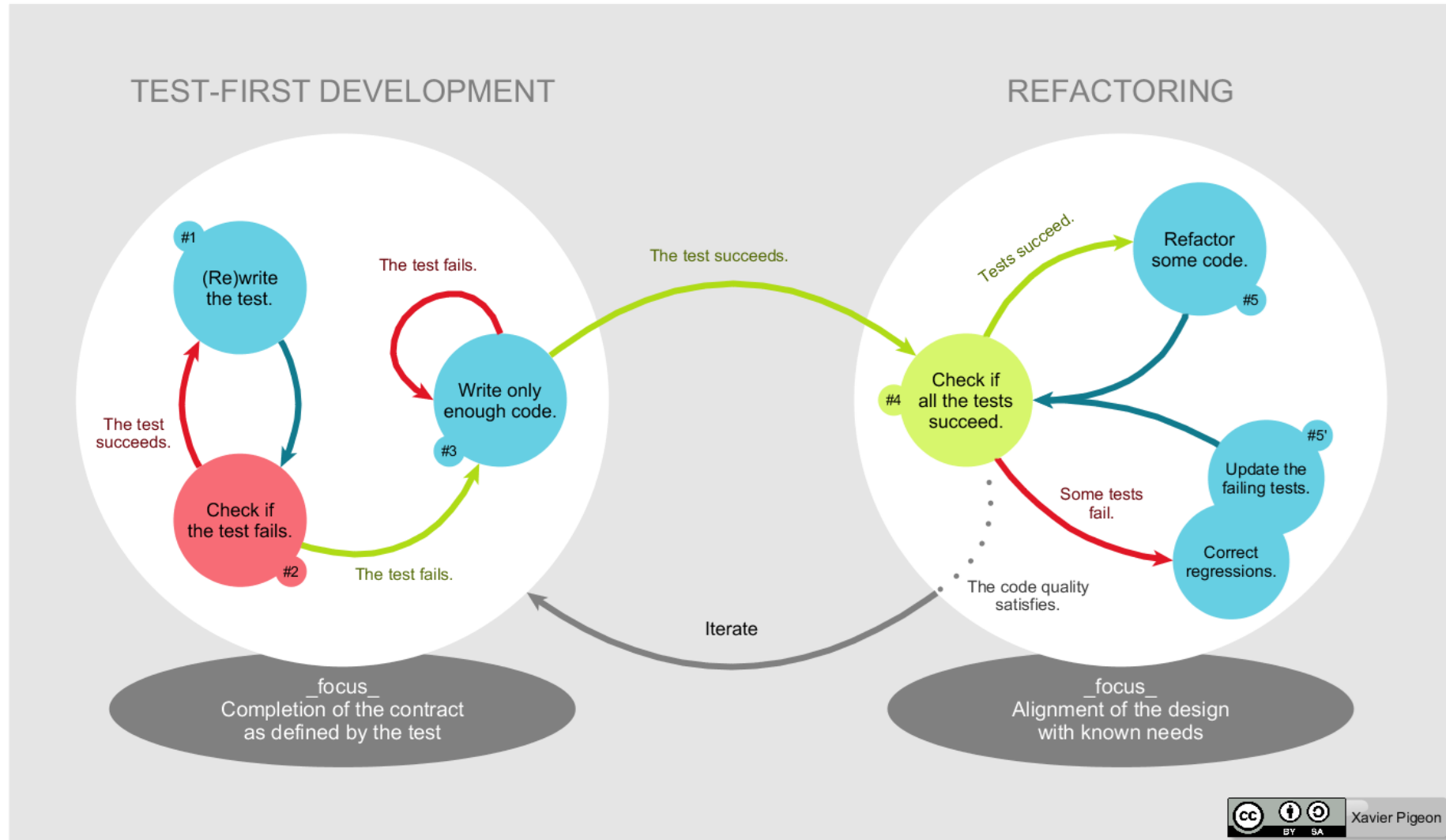
Why?

Unit tests:

- Let you validate your component design in isolation, making it easier to debug when there's a problem.
- Let you test functionality which is either not exposed or not easily accessible from the program's external interfaces, for example error handling and functionality which has no UX yet.
- Provides a cheap way to ensure you haven't broken any major existing functionality across the system when you make cross-component changes.
- Provides a way for people unfamiliar with your code to verify they haven't broken it when they are working with it. And, for them to see how it works.
- Automates what might otherwise be arduous manual testing. It helps keep you from getting lazy and not testing the impact of changes.

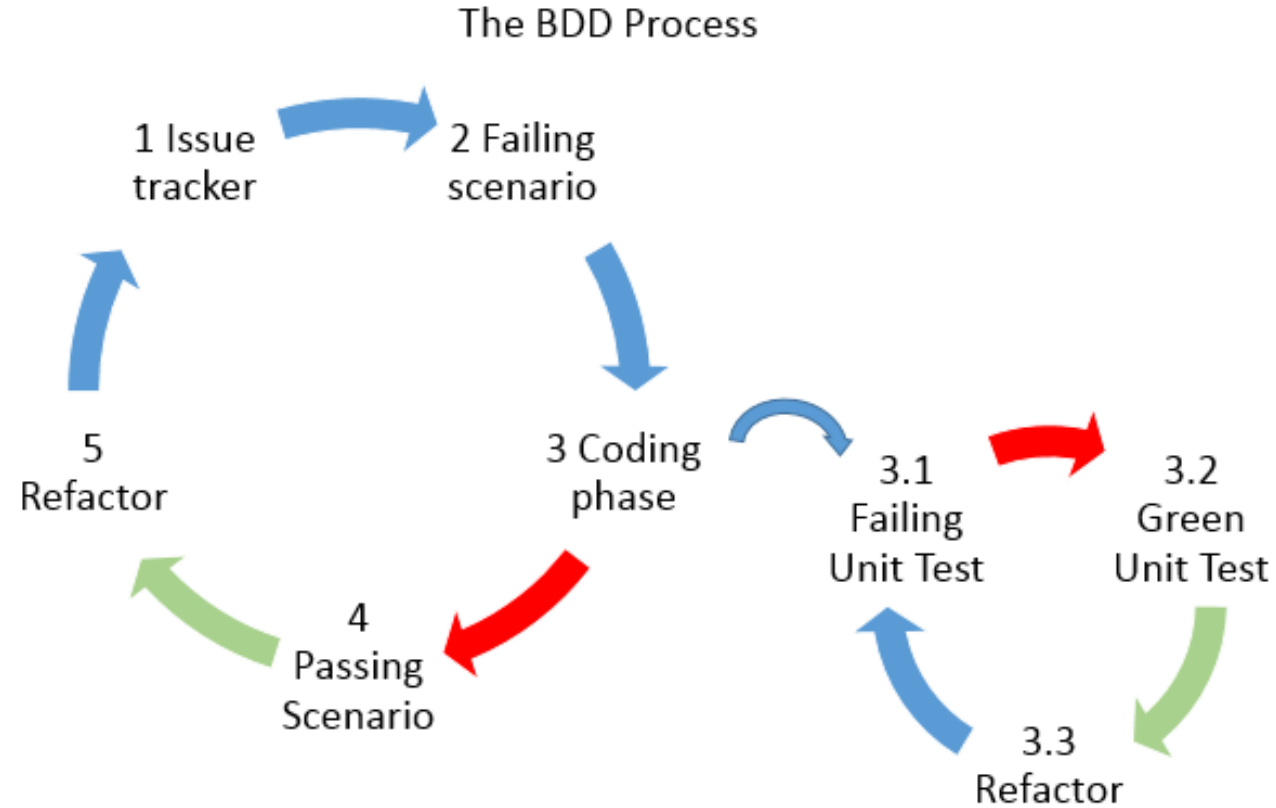
- **Test-driven development (TDD)** is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the software is improved to pass the new tests, only. This is opposed to software development that allows software to be added that is not proven to meet requirements.

TDD Process



In software engineering, **behavior-driven development (BDD)** is a software development process that emerged from test-driven development (TDD). Behavior-driven development combines the general techniques and principles of TDD with ideas from domain-driven design and object-oriented analysis and design to provide software development and management teams with shared tools and a shared process to collaborate on software development.

BDD Process



Common modules for testing

- Mocha is a feature-rich JavaScript test framework running on [Node.js](https://node.js.org/) and in the browser, making asynchronous testing *simple* and *fun*. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. <https://mochajs.org/>
- Chai is a BDD / TDD assertion library for [node](https://node.js.org/) and the browser that can be delightfully paired with any javascript testing framework. <http://chaijs.com/>
- Sinon Standalone test spies, stubs and mocks for JavaScript. Works with any unit testing framework. <http://sinonjs.org/>
- Nock
- ...