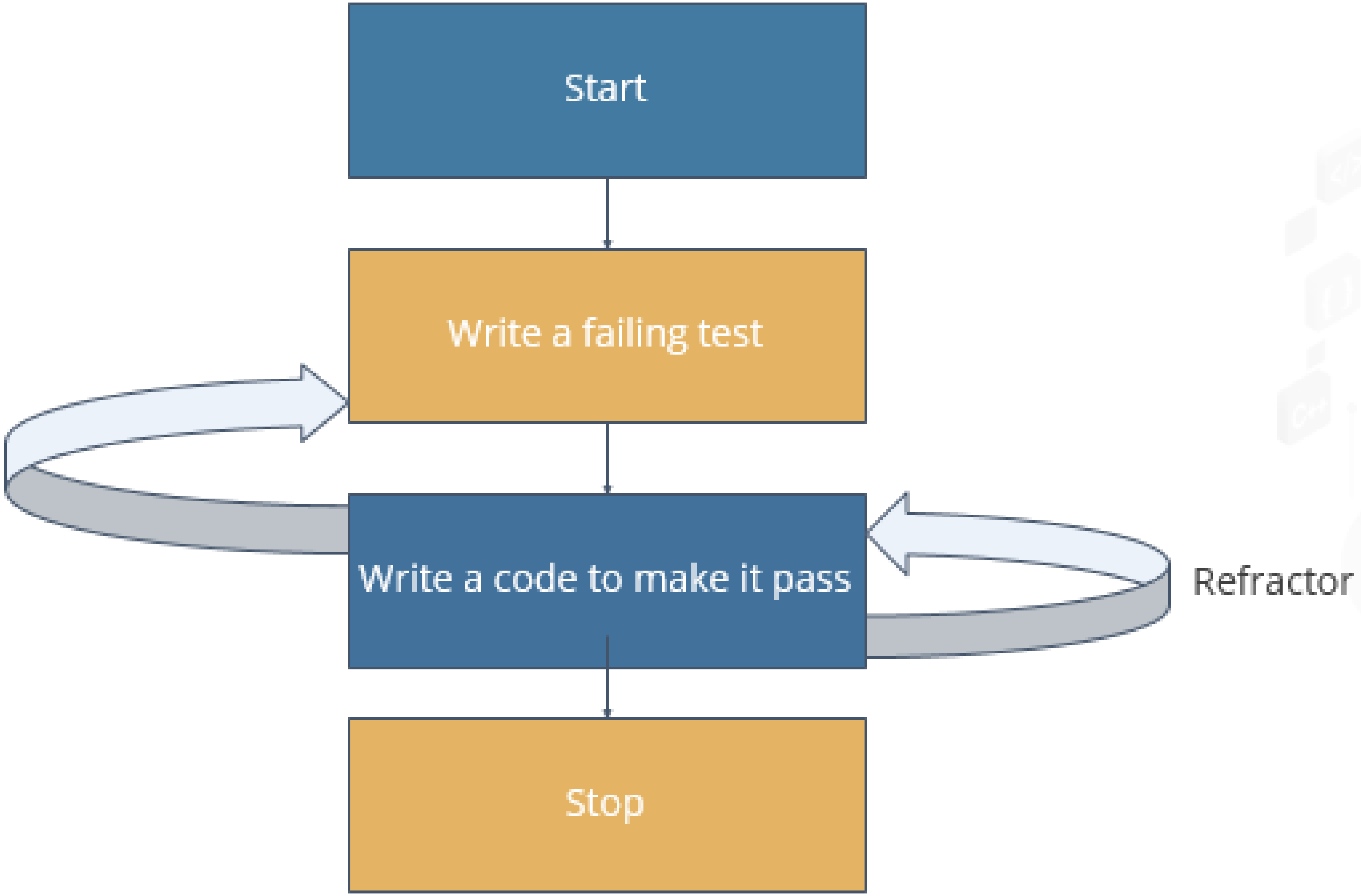
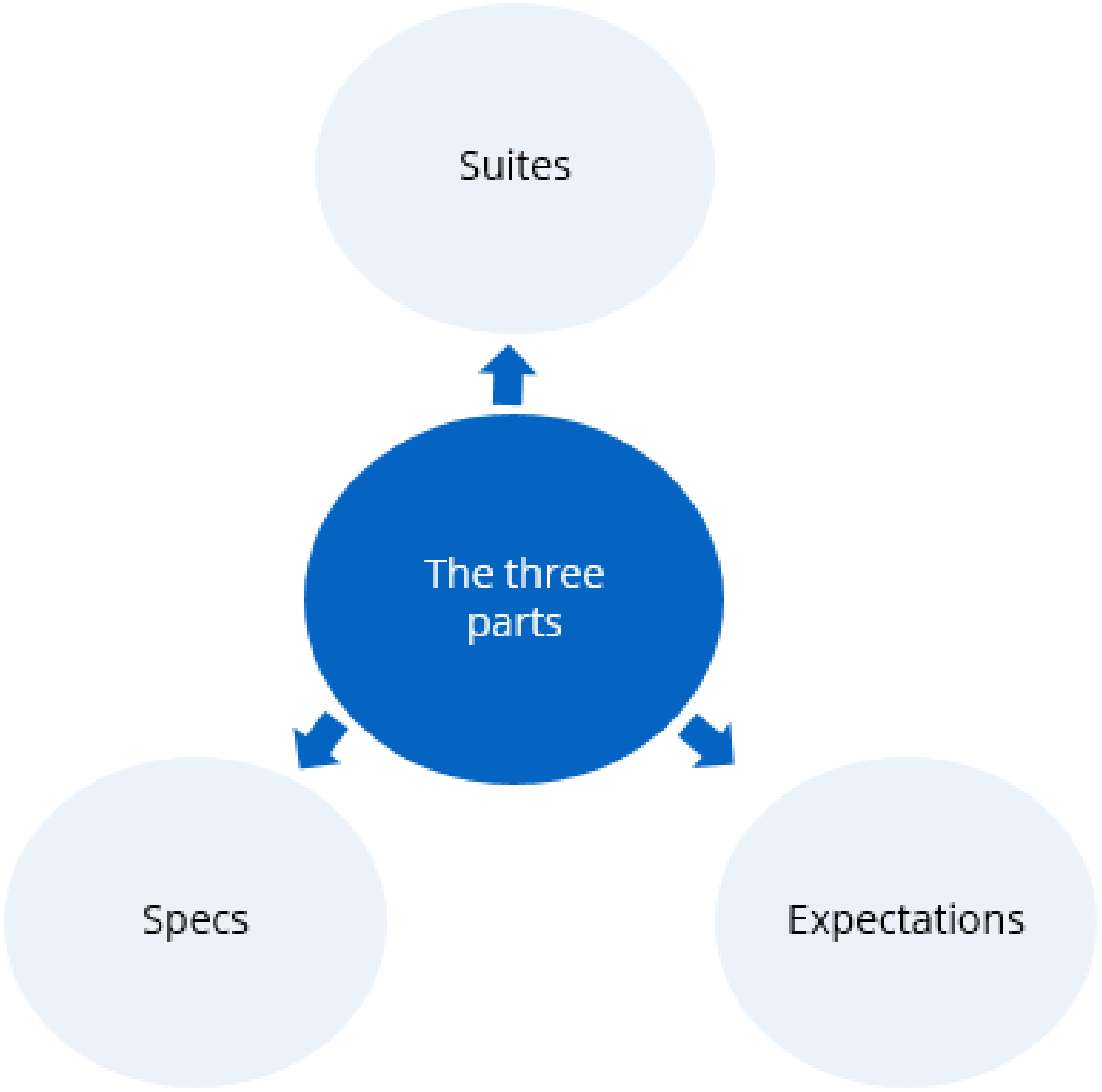
# Jasmin Unit Testing

# BDD Architecture

Jasmine follows the Behavioral Driven Development (BDD) framework.



# Jasmine Tests



The three parts

Suites

Specs

Expectations

# Suites, Specs, and Expectations

## Suites

A test suite begins with a call to the global Jasmine function *describe* with two parameters:

- String: A name or title for a spec suite
- Function: A block of code that implements suite

## Specs

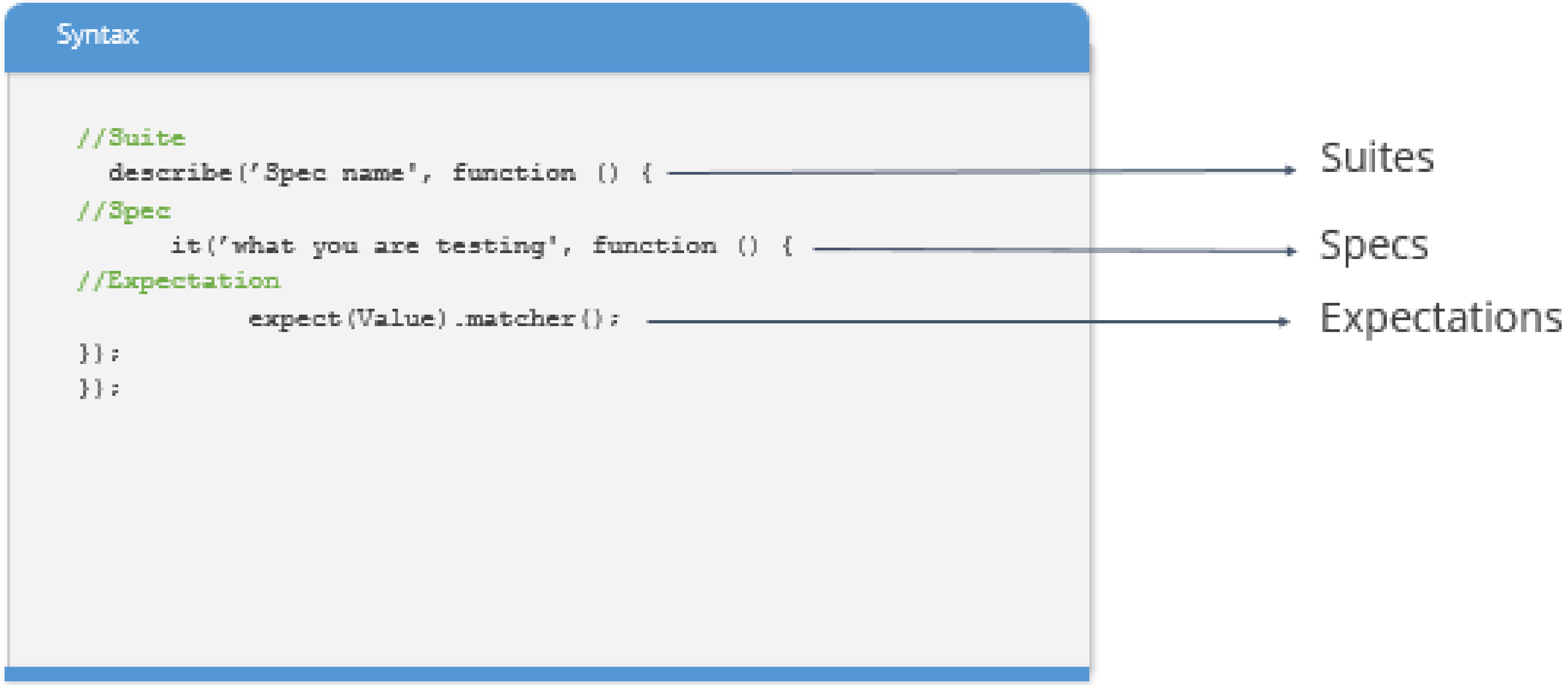
Defined by calling the global Jasmine function *it* with two parameters:

- String: A title for a spec
- Function: A spec or test

## Expectations

- Built with the *expect* function which takes a value.
- Chained with a *matcher* function, which takes an expected value.

# Suites, Specs, and Expectations



Syntax

```
//Suite
  describe('Spec name', function () {
//Spec
      it('what you are testing', function () {
//Expectation
           expect(Value).matcher();
});
});
```

# Matchers

- Implements a boolean comparison between the actual value and the expected value
- Reports to Jasmine and then passes or fails the spec
- Evaluates a negative assertion by chaining the call to *expect* with a *not* before calling the matcher

**Examples of matchers:**

- toBe(true)
- toEqual(10)
- toMatch(/bar/)
- toBeGreaterThan(6)
- toBeLessThan(4)
- not.toBe(true)
- toThrow()