Human-readable, executable tests with Gherkin and Cucumber

19.3.2024

Armin Troy - Technical Agile Coach - www.technicalagile.coach

Fundamentals of Test-Driven Development

technical
agile coach

The TDD
Loop

add a failing unit test (the test is written before the corresponding production code!)

red

green

make minimal changes to the code in order to make all tests pass

refactor

clean up the code while preserving its behavior which is defined by the existing tests

Acceptance-Test-Driven Development

| Acceptance-Test-Driven Development (ATDD)	=	Behavior-Driven Development (BDD)	=	Specification by Example
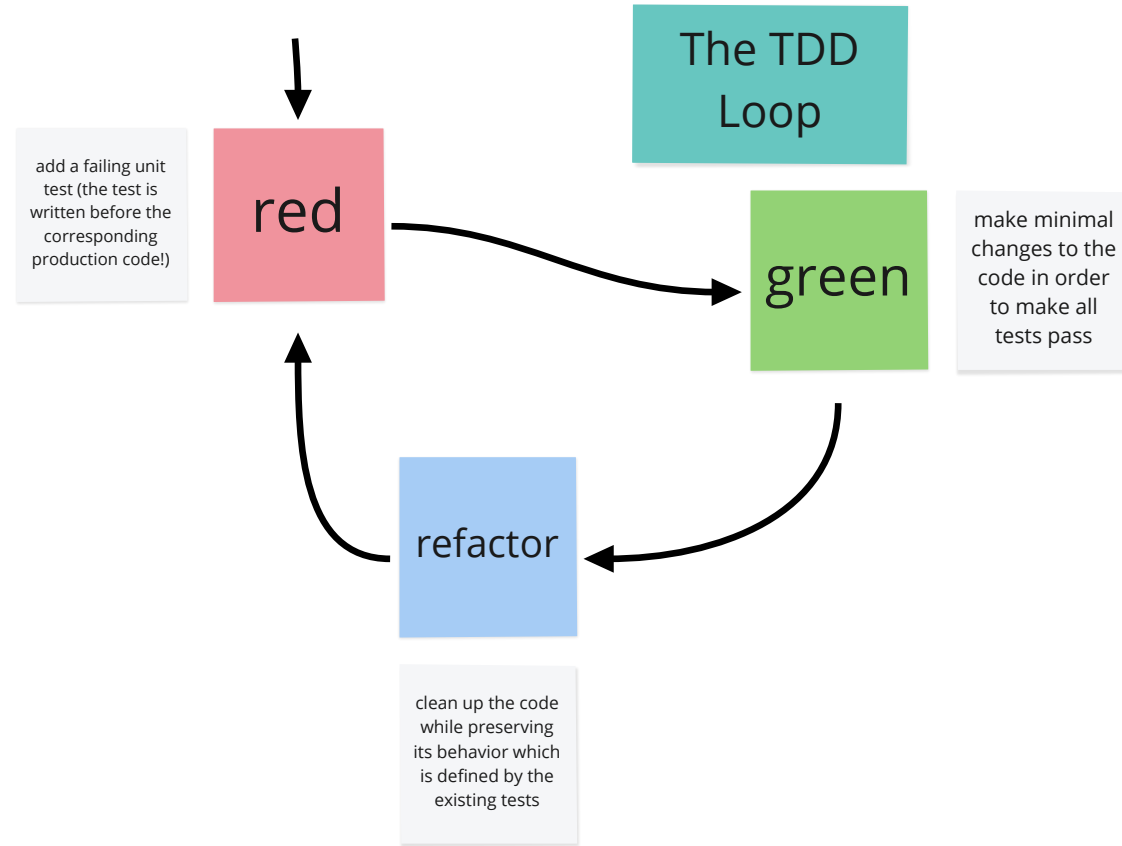
collaboration between developers, testers and business experts

tests written in human readable domain-specific language

formalization of requirements using code

Armin Troy - Technical Agile Coach - www.technicalagile.coach

Cucumber/Gherkin

Tool

syntax for the DSL
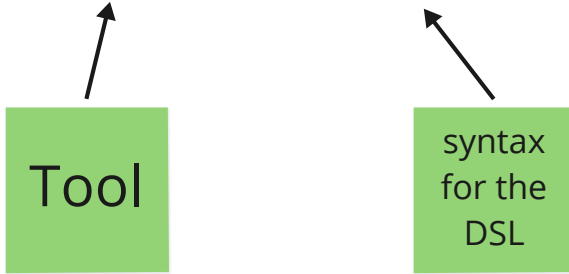
Step Definitions

```
@given(u'we have two numbers {x:d} and {y:d}')
def step_impl(context,x,y):
    context.x = x
    context.y = y


@when(u'they are added together')
def step_impl(context):
    context.result = add(context.x,context.y)


@then(u'the result should be {z:d}.')
def step_impl(context,z):
    assert context.result == z
```

translate steps into actual code

context is an implicitly created scope for each scenario

Feature Files

```
Feature: Addition

  Addition of integers

  Scenario: 1+3
    Given the numbers 1 and 3
    When they are added together
    Then should the result be 4

  Scenario: -1+3
    Given the numbers -1 and 3
    When they are added together
    Then should the result be 2
```

Arrange Act Assert

```
Feature: Multiplication

  Multiplication of integers

  Scenario Outline: <a> * <b>
      Given the numbers <a> and <b>
      When they are multiplied together
      Then should the result be <answer>

      Examples:
        | a  | b | answer |
        | 1  | 0 | 0      |
        | -1 | 3 | -3     |
        | 2  | 3 | 6      |
```

Cucumber integrations exist for most programming languages

cucumber.io

Gherkin Reference

Gherkin uses a set of special keywords to give structure and meaning to executable specifications. Each keyword is translated to many spoken languages; in this reference we'll use English. Most lines in a Gherkin document start with one of the keywords....

behave.readthe...

Welcome to behave! - behave 1.2.7.dev5 documentation

behave is behaviour-driven development, Python style. Behavior-driven development (or BDD) is an agile software development technique that encourages collaboration between developers, QA and non-technical or business participants in a software project. ...

behave.github.io

behave Examples and Tutorials - behave 1.2.6.1: Examples and Tutorials

behave is framework and a cucumber-clone for Python. This project provides tutorials and examples how to write tests by using behave. The examples are mostly based on the excellent behave documentation.

Ensemble Programming

technical agile coach

Roles

Ensemble Programming (Mob Programming)

>2 persons are programming together on one computer

Driver

does all the typing

Navigator

gives directions to the driver / decides what to code next

different patterns/rules when to switch roles, ...