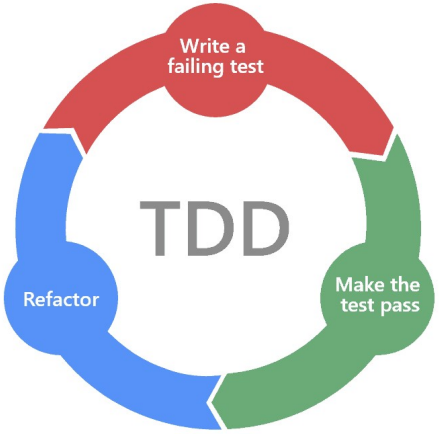
Test Driven Development

Using Python



What is Test Driven Development (TDD)?

Focuses

- On creating unit test cases before coding

Iterative methodology

Prioritizes

- Creation of and checking against test cases at every stage of software development

Converts

- Each component of the application into a test case

Examples in Python

- # Test case
- def test_reverse_list(self):
- 	expected = [6, 5, 4, 3, 2, 1]
- 	actual = reverse_list([1, 2, 3, 4, 5, 6])
- 	self.assertEqual(expected, actual)
- ----------

- # Actual code
- def reverse_list(input_list):
  - return input_list[::-1]
- ----------

TDD Vs. Traditional Testing

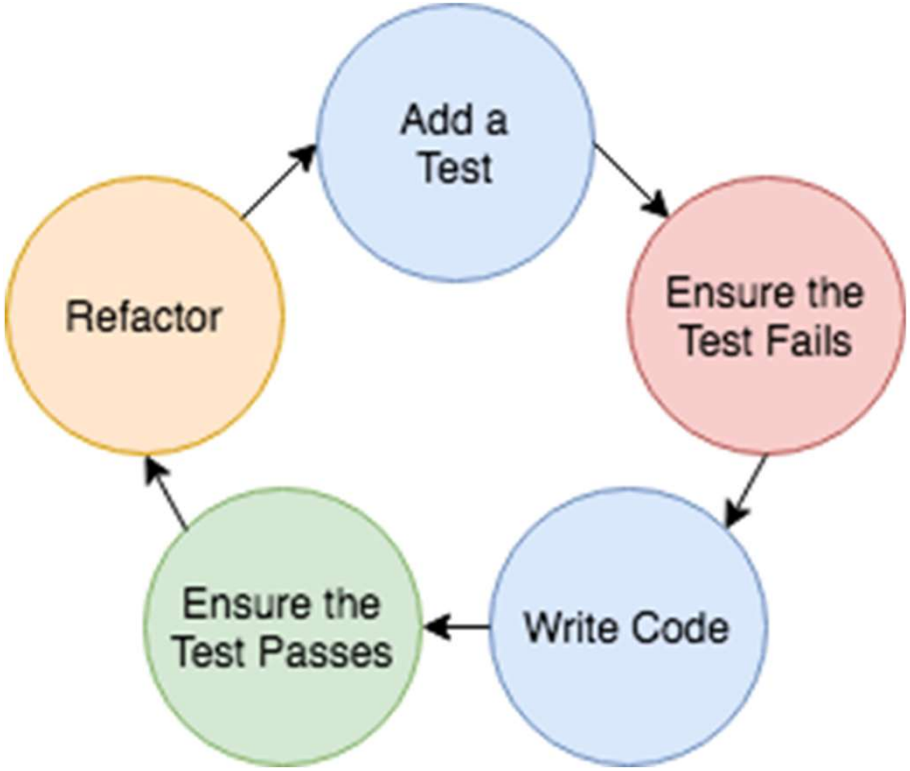
	TDD	Traditional
Approach	Tests are written before the code is developed	Testing is performed after the code is written.
Testing Scope	Focuses on testing small code units at a time	Covers testing the system as a whole, including integration, functional, and acceptance testing.
Iterative	Small chunks of code are developed, tested, and refined until they pass all tests	The code is usually tested once and then refined based on the results
Debugging	Aims to catch errors as early as possible in the development process, making debugging and fixing them easier	Require more effort to debug errors that are discovered later in the development process.

Best Practices for Test Driven Development (TDD)

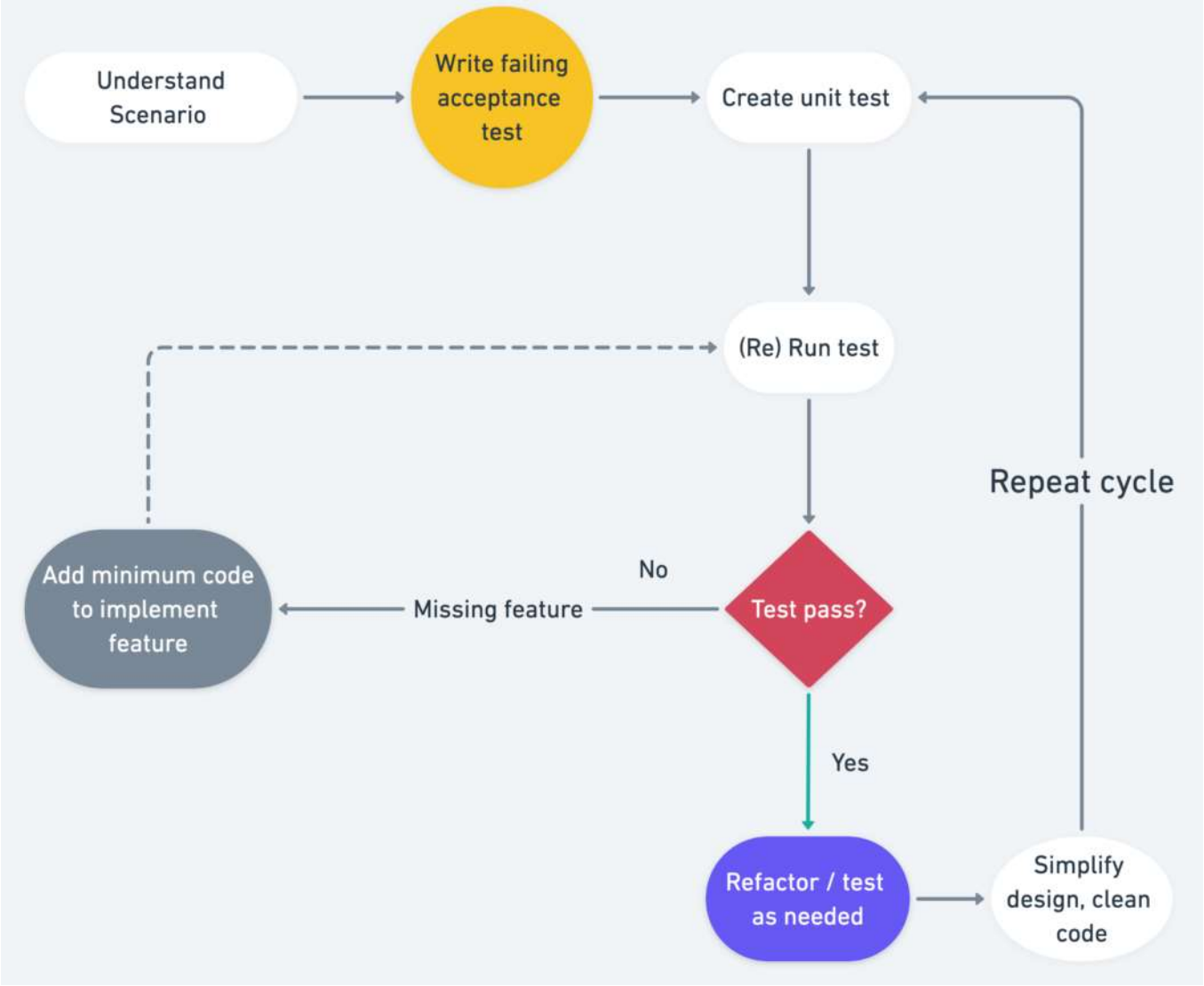
- Start with a clear understanding of requirements
- Write atomic tests
- Write the simplest test case first
- Refactor regularly
- Maintain a fast feedback loop
- Automate your tests
- Continuously run tests
- Test failures should guide development

Red-Green-Refactor cycle

1. Add a test to the test suite
2. (**Red**) Run all the tests to ensure the new test fails
3. (**Green**) Write just enough code to get that single test to pass
4. Run all tests
5. (**Refactor**) Improve the initial code while keeping the tests green
6. Repeat



How TDD Works?



Myths and Misconceptions

- You create a 100% regression test suite
- The unit tests form 100% of your design specification
- You only need to unit test
- TDD is sufficient for testing

TDD vs BDD

TDD (Test Driven Development)	BDD (Behavior Driven Development)
Focuses on the developer’s opinion on how functions of the software should work. It is basically a programmer’s view.	Focuses on the user’s opinion on how they want the application to behave. It is basically a customer’s view.
A Low-level approach	As a user approach
Verifies whether the implementation of the functionalities are correct	Verifies whether the application behaves the way the user wants it to behave

CONS OF TEST-DRIVEN DEVELOPMENT?

- Slows down the development process
- Difficult to learn
- Challenging to support and maintain

Python testing frameworks

unittest

- built-in Python framework for unit testing

pytest

- The most widely used Python testing framework
- Supports unit testing, functional testing, and API testing

doctest

- Ensures that all software programs are thoroughly documented and tested to ensure they run as they should

nose2

- Incredibly easy to adopt for those familiar with unittest

Testify

- For unit, integration, and system testing

Hypothesis

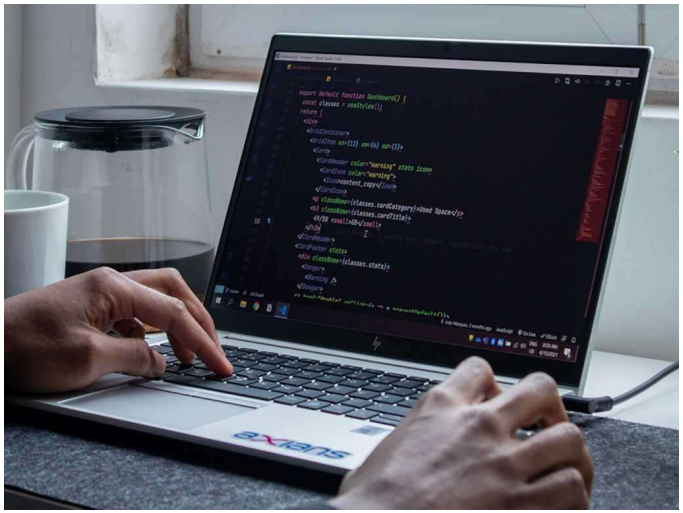
- Built to support data science projects

Hands-on

- Code:
  - https://github.com/atingupta2005/python-tdd

- Lab:
  - http://vmpython-tdd.eastus.cloudapp.azure.com:8888/tree?

  - Password: jupyter123



Thanks