Monday: Behaviour Driven Development and Test Driven Development

Behavior Driven Development- BDD

It is a process that facilitates the collaboration between the **product owner**, **tester** and **developer** when determining the system behavior.

In traditional development, the product owner comes up with behavior, tester and developer are left to translate the product owner's behavior into test cases and code.

This is known as the **three amigos** (product owner, tester and developer). The requirements are arrived at through **questions** and use of **examples** when describing **scenarios**.

BDD allows the development team to have a clear understanding of the uncertainties, which in turn helps them create a product that matters. It focuses on why the features are created rather than how.

Implementing BDD

We will use a scenario to explain the system behavior. Let's say we want to use BDD to explain how John is able to receive emails from Daisy.

We use the following keywords **Given**,**When** and **Then**.

So based on the above scenario:

Given : Daisy and John have email accounts.

When: John opens his email.

Then: He should see the mail from Daisy.

You can see that we have given a simple description of how the process behaves when Daisy sends an email to John. The main idea is to understand how we can break down key steps from the mentioned scenario.

Why Use BDD?

Clarity - everyone is able to understand how the software works.

Simplicity - using simple english to explain how the system works, this is key in helping the team to understand how everything works.

Transparency - everyone knows what to expect from the system.

Specific - we only write code for the features that we need.

Confidence - we are sure that the features are working since we are using tests.

Summary

We use BDD to facilitate good communication between the different parties that are involved when developing a product. We are able to better understand everything that is required when developing the system, through constant feedback and use of examples.

Test Driven Development -TDD

TDD stands for **test driven development** or **test driven design**.

It is a style of programming where 3 activities are involved

1. Coding
2. Testing
3. Refactoring

It can be summarized into the following steps:

- Write a single test describing a feature of the program.
- Run the test and to make sure it fails because your program lacks that feature.
- Write the simplest code possible to make test pass.
- Refactoring the code till it confirms to the simplest criteria.

Benefits of TDD

With TDD you get.

1. Fewer bugs in code -because you have tested out the main features
2. Focus - your focus is narrowed to making sure the test passes
3. Documentation - tests act like documentation in that if you do not know what a bit of code does you can read it and see what it does.