POWERING POTENTIAL

ACCEPTANCE CRITERIA

OBJECTIVES

To understand what Acceptance Criteria is

How it helps to move a Product Backlog Feature into DoR state ready for development.

ACCEPTANCE CRITERIA

Acceptance criteria dictates the conditions for software to be considered as done.

It is a set of statements that usually have a pass / fail result for all requirements.

Are attached to user stories to understand what a feature needs.

It is easier to understand the minimum viable product based on these criteria, and you can also derive tests from the criteria.

DEFINITION OF READY

Definition of Ready (DoR) defines what a product backlog item needs before it can go into the sprint backlog.

Once a product backlog feature has acceptance criteria so that a developer understands what a feature needs.

Understands the minimum viable product based on these criteria, and

can derive tests from the criteria.

Then the feature is DoR and is ready to be considered in the Sprint Planning Meeting for the next Sprint

EXAMPLE

User Story 1

As a subscriber

I want to be able to create an account on my technical website

So that I can learn the best technical indicators to help me improve my trading.

This user story requires further criteria so that a developer can start work

EXAMPLE

User Story 1

As a **subscriber**

I want to **be able to create an account on MyTechnicals website**

So that **I can learn the best technical indicators to help me improve my trading.**

Acceptance Criteria

1. **Able to create an account manually (filling out the sign-up form)**
2. **Able to create an account via Facebook**
3. **Able to create an account via Google**
4. **Able to create an account via LinkedIn**

The user story is now DoR

REVIEW

You now know

- what Acceptance Criteria is
- How it helps move a Product Backlog Feature into DoR state ready for development.

LAB

Complete Lab7.

Duration 30 minutes

THINGS TO THINK ABOUT

How long does it take to

1. Set the Priority
2. Read through the User story and understand it completely
3. Come up with a plan of action (use existing modules, how to test...)
4. Write the code
5. Write Unit tests for the code
6. Integrate (merge) it with the rest of the project code (GIT)

Deployment: locally, test server, on company server, cloud