Acceptance Criteria and Definition of Done. Examples.

In the previous lesson we spoke about the definition of done and acceptance criteria. In this reading, we will look at examples of what they may look like.

First, let’s look at the definition of done. Definition of done is typically applied on product increment level (or higher) and describes criteria to consider a piece of work done, including all the relevant quality checks and non-development activities.

A typical definition of done will combine the following types of criteria:

- Criteria telling that the work is done in full
- Criteria telling that the work has passed needed tests and levels of acceptance
- Criteria telling that non-development activities required to accept the work are done (e.g. training, regulatory compliance checks, etc.)

Let’s say your project is building a new increment targeted at implementing accounts management for users. What might a definition of done look like?

EXAMPLE

Product Increment:

User accounts management

Definition of Done:

1. **Code Completed:** All code changes are implemented, following coding standards and best practices, to deliver all functionality outlined in acceptance criteria.
2. **Code Reviewed:** Code has been peer-reviewed and approved with no critical issues.
3. **Unit Tests Passed:** Automated unit tests cover key functionality.
4. **Integration Tested:** Functionality has been tested end-to-end in a test environment.
5. **Security Verified:** As needed, penetration test and vulnerability scanning are performed; security standards are followed.
6. **User Interface Finalized:** The UI pages are responsive, user-friendly, and meet accessibility guidelines.
7. **No Critical Bugs:** All high-severity defects are resolved before release.
8. **Documentation Updated:** Relevant user guides and API documentation are updated.
9. **Feature Demoed:** The feature is demonstrated to stakeholders and meets business expectations.

Note, that item 1 refers to “work done in full”, items 2-7 refer to “work has passed needed tests”, and items 8 and 9 refer to “non-development activities”.

Now, equipped with this definition of done we can start defining our backlog items - that is defining the scope of actual smaller pieces of functionality that will be built as part of this increment.

We will capture them as user stories (read this article if you are not familiar with user stories), and one of the user stories may read as following:

Backlog Item: EXAMPLE

As a user
I want to reset my password
So that I can regain access to my account when I forget the password

The acceptance criteria for this backlog item should cover everything that the developer must deliver for the backlog item to be accepted and closed as done. Typically, acceptance criteria cover all the different successful functional scenarios, additional non-functional considerations, and error handling. Acceptance criteria may also cover some acceptable alternatives for the delivery of the item.

Acceptance Criteria: EXAMPLE

- **The user must be able to request a password reset by entering their registered email.**
- **A password reset email is sent only if the email is associated with an existing account.**
- **The email contains a secure, time-limited (15 minutes) reset link.**
- **Clicking the reset link takes the user to a secure page to enter a new password.**
- **The new password must meet security requirements .**
- **The old password no longer works after a successful reset.**

Meeting acceptance criteria like this will contribute to compliance with item 1 in the definition of done for all the backlog items. Acceptance criteria for backlog items only capture the criteria that are unique to this backlog item; because the rest of criteria that are broadly applied to all the work items in the backlog are already captured in the definition of done.