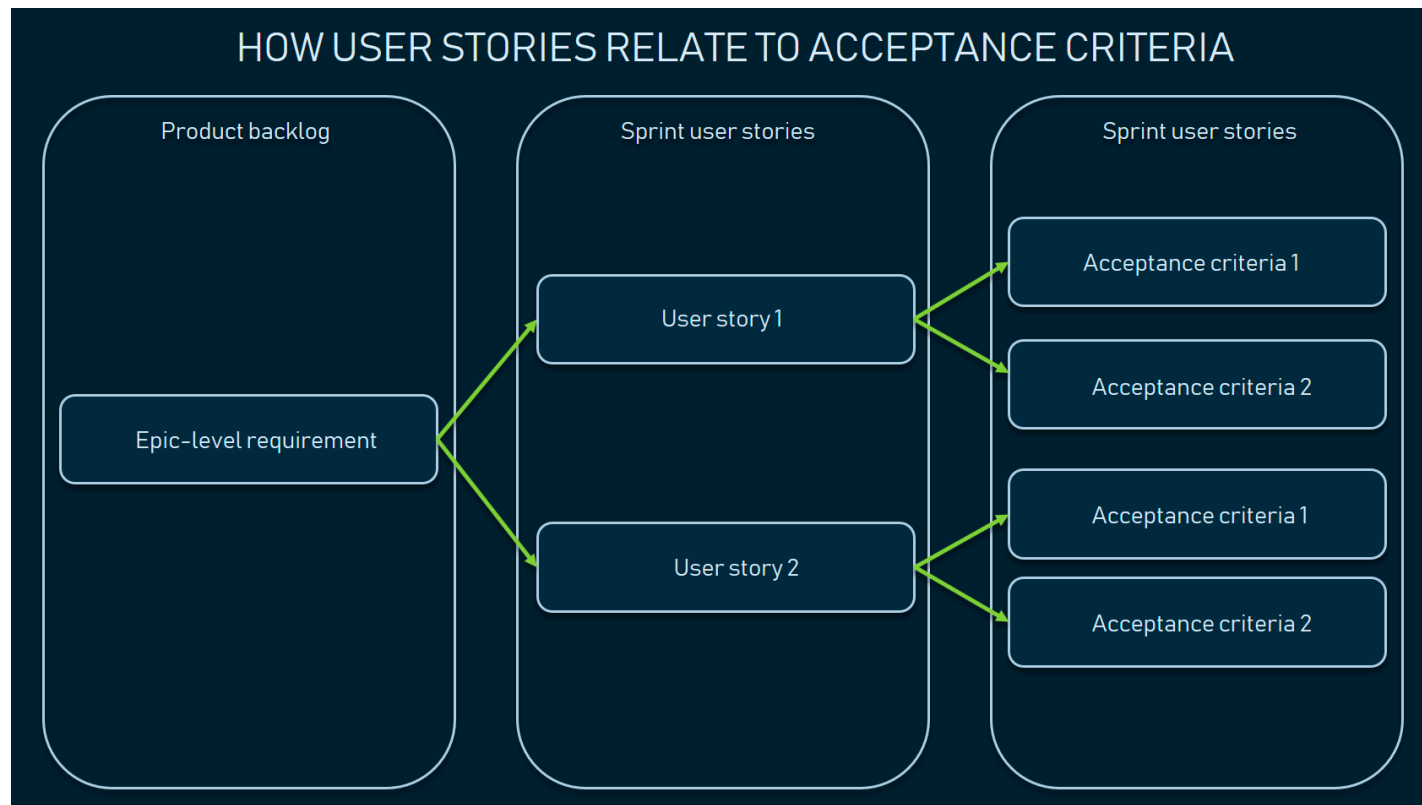


When is DONE – really done?



Acceptance Criteria

AC define the boundaries of a user story. Well-written acceptance criteria help avoid unexpected results in the end of a development stage and ensure that all stakeholders and users are satisfied with what they get.



User Story *without* Acceptance Criteria

Consider the following user story;

As a learner I want to pay for my course online so that I don't have to pay in person.

This user story doesn't cover all of the parameters we would need to know we are building the "right thing".

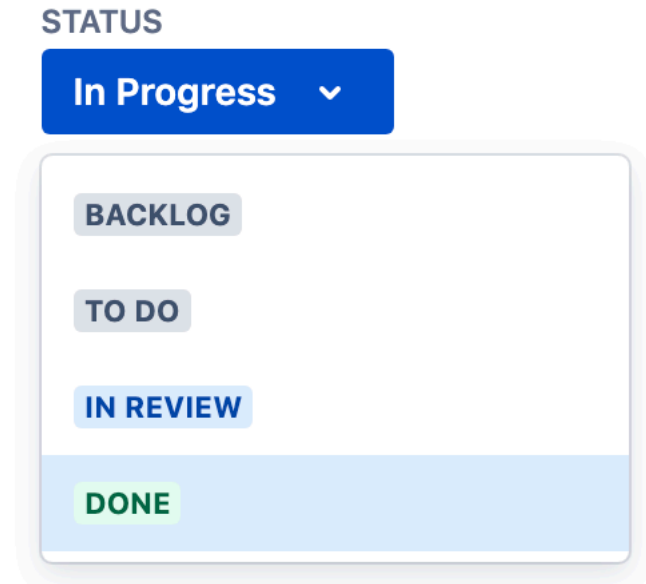
- What payment methods are to be supported?
- Which fields in the payment form are mandatory?
- Do I have to be registered first or can I make the payment as a guest?
- Do I get a receipt by email?
- How long do I have before my session expires?

Another example

User story: *As a user, I want to be able to register online, so that I can start shopping online.*

Acceptance criteria:

- ☐ User can only submit a form by filling in all required fields
- ☐ The email user provided must not be a free email
- ☐ Submission from same IP can only be made three times within 30 minutes
- ☐ User can only submit a form by filling in all required fields
- ☐ User will receive a notification email after successfully registration



A Definition of Done (DoD) drives the quality of work and is used to assess when a User Story has been completed

Definition of Done for a Feature (Story or Product Backlog Item)

- Source code documented
- No unintegrated work in progress has been left in any development repo
- Peer review completed
- Code has been **Refactoring** – look for “code smells”*
- Project builds without errors
- The Build is deployed on the test environment
- Check no log messages are printed to the console (Chrome debug tools)
- Acceptance criteria is met
- Unit tests are run and all pass
- Functional testing completed and documented
- Integration testing completed (App works with the APIs)
- Regression testing has been completed
- Tested on Cross-browsers/specified devices
- Functionality documented in necessary user user documentation
- Product Owner accepts the User Story

DoD - Code refactoring

Be on the look out for “code smells”!

- Duplicated code
- Usage of overcomplicated design patterns where simpler design would suffice.
- Shotgun_surgery: a single change needs to be applied to multiple classes at the same time.
- Excessive use of literals
- high coupling(global state)
- Long methods
- Too many parameters in method call
- Code executed multiple times without effect (e.g. render())
- Unhandled exceptions (e.g. missing catch in a Promise, try...catch)

We can also have a Definition of Done for a Sprint

In addition to DoD for features/stories;

- User Acceptance Testing (UAT) completed and results documented
- Product Backlog updated
- Features are ready to demo
- All features bundled and deployed to test environment
- Accessibility testing
- Performance testing