WORKSHOP

Writing User Stories
# User Stories

1. **what is a user story?** 4
2. **user story template** 5
3. **examples: user stories** 6
4. **user story checklist** 7
5. **why not tasks?** 8

# Writing Acceptance Criteria

2. **what is acceptance criteria?** 10
3. **example: acceptance criteria** 11
4. **acceptance criteria checklist** 13
What Is A User Story?
A user story is a tool used in agile software development to capture the description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

A user story often follows the following ‘equation’:

As a <type of user>, I want <some feature> so that <reason>

A simple example of this could be:

As an online shopper, I want to add an item to my cart, so that I can purchase it
### user story template

<table>
<thead>
<tr>
<th>WHO</th>
<th>WHAT</th>
<th>WHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>are we building it for? Who is the user?</td>
<td>are we building? What is the intention?</td>
<td>are we building it? What is the value for the customer?</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As a &lt;type of user&gt;</td>
<td>I want &lt;some goal or objective&gt;</td>
<td>So that &lt;benefit/value&gt;</td>
</tr>
</tbody>
</table>
examples: user stories

As an internet banking customer
I want to see a rolling balance for my everyday accounts
So that I know the balance of my account after each transaction is applied

As an administrator
I want create other administrators
So that I can delegate tasks

As a marketer
I want create automated email campaigns
So that I can keep evaluators engaged
user story checklist

- Keep them short
- Keep them simple
- Write from the perspective of the user
- Make the value/benefit of the story clear - what is the reason for the story?
- Describe one piece of functionality. If you have to write and break it into 2 stories
- Write stories as a team
- Use acceptance criteria to show a MVP
why not just use ‘tasks’?

<table>
<thead>
<tr>
<th><strong>user stories</strong></th>
<th><strong>tasks</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>a user story = the WHAT</td>
<td>the task = the HOW</td>
</tr>
<tr>
<td>user stories describe a piece of functionality from the point of view of the user</td>
<td>“what are the activities we need to perform in order to deliver outcomes (user stories)”</td>
</tr>
<tr>
<td>divided features into business processes</td>
<td>tasks are individual pieces of work</td>
</tr>
</tbody>
</table>
Writing Acceptance Criteria
Acceptance criteria or ‘conditions of satisfaction’ provide a detailed scope of a user’s requirements. They help the team to understand the value of the story and set expectations as to when a team should consider something done.

Acceptance Criteria Goals:

- to clarify what the team should build before they start work
- To ensure everyone has a common understanding of the problem
- To help the team members know when the story is complete
- To help verify the story via automated tests
As an online banking customer, I want a strong password, so that my credit card information is secure

Acceptance Criteria:

- The password must be at least 8 characters
- The password must contain at least 1 character from each of the following groups: lower case alphabet, upper case alphabet, numeric, special characters (!, @, #, $, %, ^, &, *)
As a conference attendee, I want to be able to register online, so that registration is simple and paperless

Acceptance Criteria:

- A user can not submit a form without filling out all of the mandatory fields
- Information from the form is stored in the registrations database
- Protection against spam is working
- Payment can be made via Paypal, Debit and Credit Card
- An acknowledgment email is sent to the attendee after submitting the form
acceptance criteria should include

- Negative scenarios of the functionality
- Functional and non-functional use cases
- Performance concerns and guidelines
- What system/feature intends to do
- End-to-user flow
- The impact of a user story to other features
- UX concerns
acceptance criteria should NOT include

- Code review was done
- Non-blocker or major issues
- Performance testing performed
- Acceptance and functional testing done

why?

Your acceptance criteria should not include any of the above, because your team should already have a clear understanding of what your Definition of Done (DoD) means. This could mean:

- unit/integrated tested
- ready for acceptance test
- deployed on demo server
- releasable