



# USER STORIES

A user story is a brief and simple description of a product feature, valuable to a customer or user of the product, and written from their perspective.

This technique helps encourage conversation and promotes balanced and timely communication between customer and development team. User stories represent customer requirements expressed in non-technical language but don't attempt to document them. As a consequence the team avoids spending time on upfront analysis and is freed up to concentrate on work in progress. This helps prevent two sources of waste: analysis going stale when the circumstances change; and analysis which is not used because a feature is never implemented.

A user story is composed of three aspects: Card, Conversation and Confirmation.

The **Card** holds a description of the user story, typically on one side of an index card. It should express what requirement it relates to and enable a focused discussion. It is therefore sometimes referred to as a 'placeholder for a conversation'. User story cards are commonly written using the following template:

## Template

<i>User Story</i>
<i>As a &lt;user role&gt;</i>
<i>I want &lt;product feature&gt;</i>
<i>So that &lt;value/benefit&gt;</i>

## Example

<i>User Story</i>
<i>As a potential customer</i>
<i>I want to read book reviews</i>
<i>So that I decide which one</i>
<i>to buy</i>

The requirement is communicated in detail through **Conversation** between customers and developers. This takes place over time as the story is being developed and when establishing further details becomes important. This aspect of the story puts emphasis on verbal communication but when required additional information can be captured in written documentation.

The development team will also require **Confirmation** that the user story has been completed and is working as intended. This is done through capturing 'acceptance criteria' (such as rules that must be met, non-functional requirements and so on), usually on the back of the card that the user story is written on in the form of a simple checklist.

<i>Acceptance Criteria</i>
<input type="checkbox"/> <i>Up to 10 most recent</i>
<i>reviews displayed</i>
<input type="checkbox"/> <i>Reviews ordered by most</i>
<i>favourable first</i>

When captured early on, acceptance criteria help the development team understand and estimate the story. They can also be added or refined later when the user story is revisited for further discussion. It is important that they get added before the user story is implemented, so that developers know when the story can be declared 'done'.

*Outcome*

*Function*

*Benefit*

*Who*

*Scaling Factors*

*Difficulty*



## Implementation

A customer team – which can include testers, a product manager, real users, interaction designers, etc. – should write the story cards.

1. While discussing requirements (whether in an ad hoc meeting or in a workshop), grab a pack of index cards (or Post-It notes).
2. Capture the user story itself on one side of a card, using the template ('As a ... I want ...') shown above.
3. If the customer has any acceptance criteria, then jot them down on the back of the card.
4. Read the story back to your customer and any team members present to ensure that you all understand what it relates to.

Once you have created a user story, make sure it's readily available to be revisited and adapted during your development process, as you continue to have more conversations around it. Change the user story as required to reflect new information or changed preferences. For example, this may include:

- adding some notes based on ad hoc conversation with the customer
- adding a size estimate to the card during an estimation meeting
- breaking the story down into several smaller stories if it proves too large, or merging smaller stories together
- ripping up the user story if it is no longer needed, or duplicated elsewhere.

## Potential pitfalls

A user story exists to invite and inform a future conversation. Don't attempt to perfect each story, or to expand the size of the card you are capturing it on. The imperfect nature of a story is its actual benefit – it forces people using it subsequently to ask more questions, thus enabling a conversation.

Don't state the solution in your user stories. Stick to describing the problem or opportunity in the language of your customers. Make sure that you focus on outcomes.

One of the benefits of using user stories is that, because the analysis takes place closer to the solution being built, more of that information can be conveyed verbally (through the conversation), rather than in a document. However, it is important to make sure your team has all the information needed to solve the problem before building a solution. Where it provides useful information and examples, add supplementing documentation.

If you want to learn more, consider reading:

*User Stories Applied: For Agile Software Development* by Mike Cohn