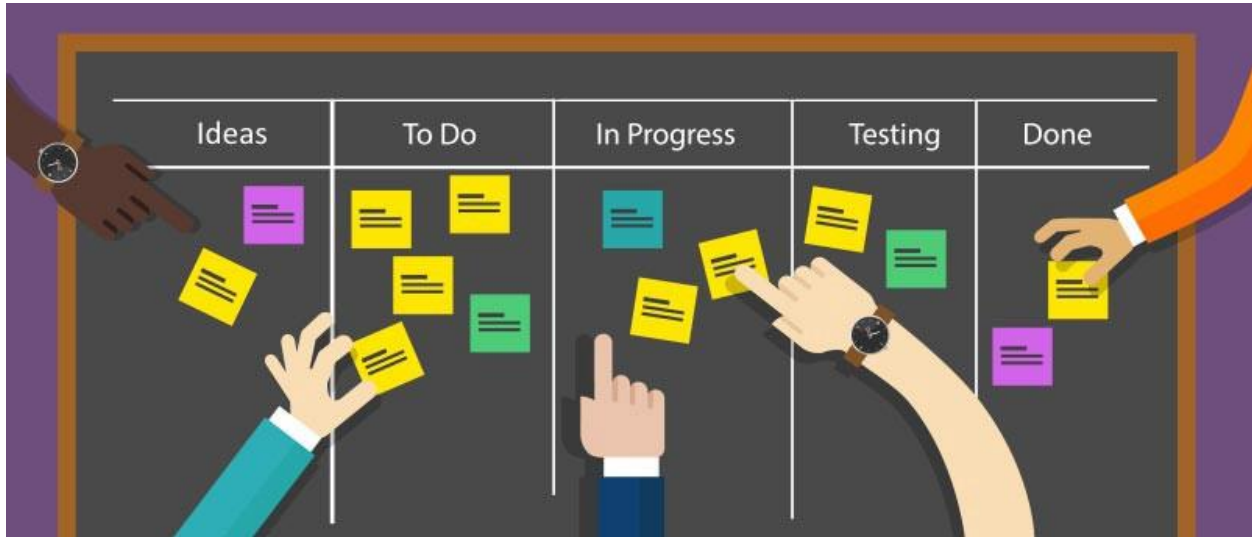


Agile Development: User Stories Are The New Requirements Document

By Lazaro Ibanez



User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them. All agile user stories include a written sentence or two and, more importantly, a series of conversations about the desired **functionality**.

What are user stories?

They are narrative texts that describe an interaction of the user and the system, focusing on the value a user gains from the system. A true user story is a metaphor for the work being done. It is not a highly documented requirement but rather a reminder to collaborate about the topic of the user story, the documentation is secondary to the collaboration. User stories aren't agile in and of themselves. Instead, their underlying agile values, collaboration and just-in-time definition, make user stories a good agile tool. Formality is specifically removed from the mix and specification of the user story is pushed as late as possible.

A definition is given by Martin Fowler, pioneer of agile methodologies:

User stories are a chunk of functionality that is of value to the customer.

Functionality, it's the key word here. User stories should be written using business language. They must be functional and state clearly what it is expected, not necessarily in detail but in purpose.

User stories usually look like:

- Parking passes can be paid via PayPal.
- Professors can input student marks.
- Transcripts will be available online via a standard browser.

An example of what is not a user story is:

- Redevelop the interface X so that it improves the performance of the whole system.

In this case the task is not written in a clear way and has no direct value to business and shouldn't be included into the product backlog, which is a prioritized list of the functionality to be developed in a product or service.

A good way to think about a user story is that it is a reminder to have a conversation with your customer, which is another way to say it's a reminder to do some just-in-time analysis. In short, user stories are very slim and high-level requirements artifacts.

Comparing User Stories, Use Cases, and Requirements

The basic difference between user stories and other forms of requirements specification has to do with perspective and intent, both of which affect the level of detail.

Traditional requirements are criteria to which the system or business must adhere. They are usually created before the coding begins and are nearly always written as text. They often are thought of as constraints, conditions, or capabilities to which the system must conform.

A use case is a series of interactions by the user with the system and the response of the system. Use cases consist of two main components: use case diagrams and the text of the use case itself. Use cases and use case diagrams focus on the user, with a use case text itself written in a call-and-response format that shows an action by the user, followed by the system's response.

Both traditional requirements and use cases are meant to be successively refined into greater detail through detailed analysis and design into. They are both written as the primary communication media for the system capabilities.

User stories focus on customer value. They don't just assume a looseness of specification; they actually encourage it in order to foster a higher level of collaboration between the stakeholders and the team. A user story is a metaphor for the work being done, not a full description of the work. The actual work being done is fleshed out via collaboration revolving around the user story as system development progresses. As such the level of detail of a user story is ultimately higher than a use case but lower than a traditional requirement. In order to limit scope, user stories have collaboratively developed acceptance criteria which define when the user story meets the stakeholder's expectations.

Do user stories replace a requirements document?

Agile projects, especially Scrum ones, use a product backlog. Although product backlog items can be whatever the team desires, user stories have emerged as the best and most popular form of product backlog items.

While a product backlog can be thought of as a replacement for the requirements document of a traditional project, it is important to remember that the written part of an agile user story ("As a user, I want ...") is incomplete until the discussions about that story occur.



It's often best to think of the written part as a pointer to the real requirement. However, we focus too much on *who* and *what*, but *why* is critical to understand their relevance. User stories could point to a diagram depicting a workflow, a spreadsheet showing how to perform a calculation, or any other artifact the product owner or team desires.