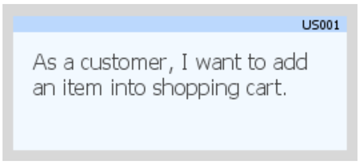
User Story

Written Date : August 15, 2016

User story is one of the most important tool for agile development. They are often used for identifying the features of a system under developed. User stories are well compatible with the other agile software development techniques and methods, such as scrum and extreme programming.

What is a User Story?

A user story is a note that captures what a **user** does or needs to do as part of her work. Each user story consists of a short description written from user's point of view, with natural language. Unlike the traditional requirement capturing, user story focuses on what the user need instead of what the system should deliver. This leaves room for further discussion of solutions and the result of a system that can really fit into the customers' business workflow, solving their operational problems and most importantly adding value to the organization.



US001

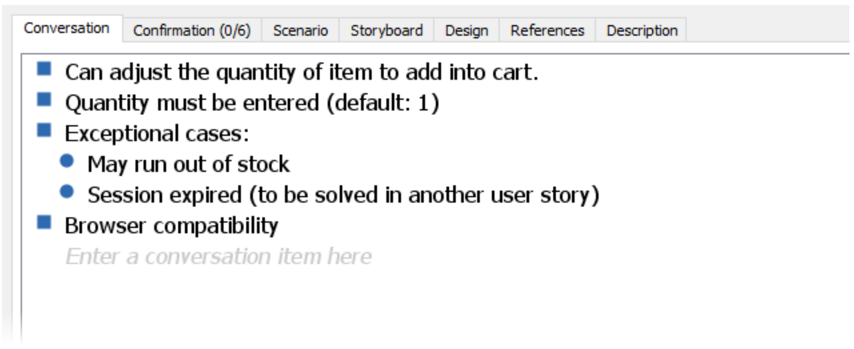
As a customer, I want to add an item into shopping cart.

Concept of 3C's

The 3C's refer to the three critical aspects of good user stories. The concept was suggested by Ron Jeffries, the co-inventor of the user stories practice. Nowadays, when we talk about user stories, we usually are referring to the kind of user stories that are composed of these three aspects.

1. **Card -** User stories are written as cards. Each user story card has a short sentence with just-enough text to remind everyone of what the story is about.

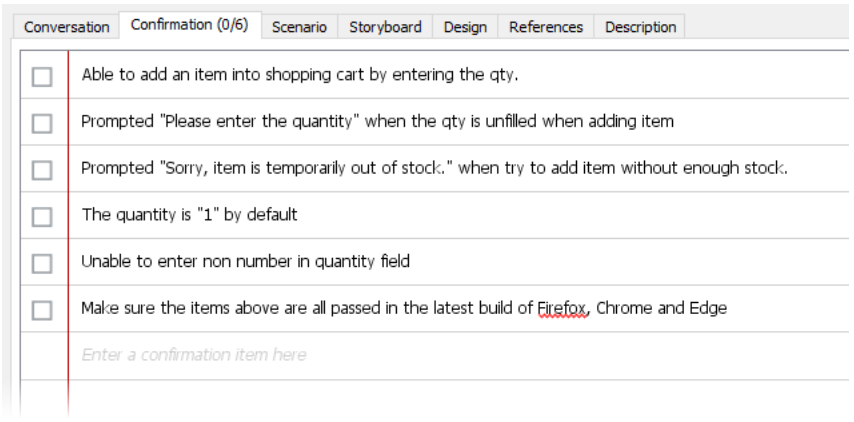
2. **Conversation** - Requirements are found and re-fined through a continuous conversations between customers and development team throughout the whole software project. Important ideas and decisions would be discovered and recorded during the stakeholder meetings.



- Can adjust the quantity of item to add into cart.
- Quantity must be entered (default: 1)
- Exceptional cases:
  - May run out of stock
  - Session expired (to be solved in another user story)
- Browser compatibility

Enter a conversation item here

3. **Confirmation** - or also known as Acceptance criteria of the user story. During the discussion of requirements, the customers tells the analyst not only what he/she wants, but also confirming under what conditions and criteria the working software would be accepted or rejected. The cases defined are written as confirmation. Note that confirmation focuses on verifying the correctness of work of the corresponding user story. It is not an integration testing.



☐	Able to add an item into shopping cart by entering the qty.
☐	Prompted "Please enter the quantity" when the qty is unfilled when adding item
☐	Prompted "Sorry, item is temporarily out of stock." when try to add item without enough stock.
☐	The quantity is "1" by default
☐	Unable to enter non number in quantity field
☐	Make sure the items above are all passed in the latest build of Firefox, Chrome and Edge
	Enter a confirmation item here

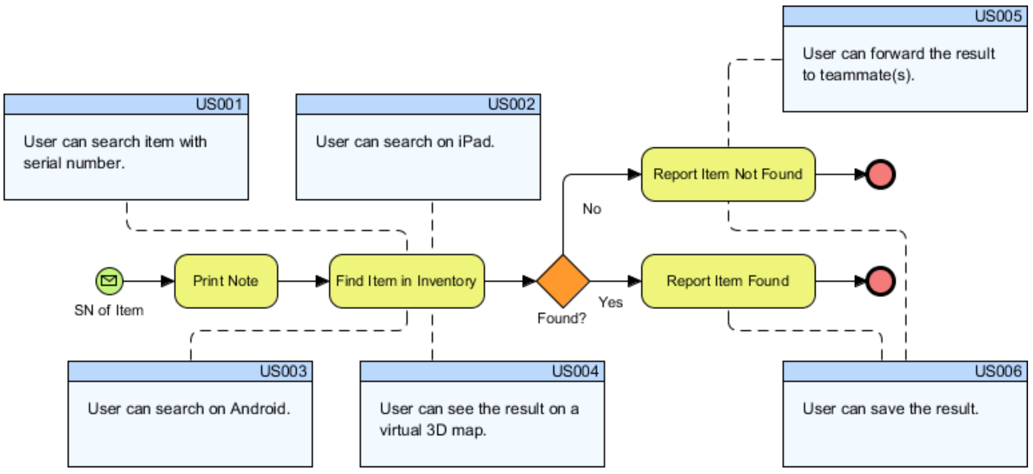
How to Identify User Story?

User stories should be identified together with the stakeholders, preferably through a face-to-face meeting. User story is a requirement discovery process instead of an upfront requirement analysis process. In the traditional requirements capturing approaches, system analyst tries to understand customers' needs and then prepare a requirement specification for the system in detail. This is not how the user story approach works. Instead of a documentation process, the identification of user story is more like a note taking process. Through the discussions with users, we listen to and understand their problems and needs, and then write down their needs as user stories at the same time. These user stories will become the source of requirements. The details could be subsequently filled just-in-time, providing the team with a "just-enough" requirement references throughout the project development process.

Mapping Business Process with User Stories

BPMN is one of the most powerful tool for business analysis and modeling. Not only we can use it to perform process improvement, but also we can identify user stories from those processes intended to be automated through the following steps:

1. Simply model the user's workflow with BPMN business process diagram.
2. Walk through the process model with users.
3. And, analyze the business activities of the problem, and then identify the user stories in related to the process required to be automated, which is also known as business process to user story mapping.


US005
User can forward the result to teammate(s).
US001
User can search item with serial number.
US002
User can search on iPad.
Report Item Not Found
No
SN of Item
Print Note
Find Item in Inventory
Found?
Yes
Report Item Found
US003
User can search on Android.
US004
User can see the result on a virtual 3D map.
US006
User can save the result.

How to Write User Story?

When writing a user story, try to write in the user's voice as the form below (or at least, make sure what you have written qualifies the following statement):

As a <role>, I want to <business objective> so that <business value/reason>.

E.g. As a , I want to so that I can .

where:

1. **<role>** represents the person, system, subsystem or any entity else who will interact with the system to be implemented to achieve a goal. He or she will gain values by interacting with the system.
2. **<business objective>** represents a user's expectation that can be accomplished through interacting with the system.
3. **<business value>** represents the value behind the interaction with the system.

It is not a rule but a guideline that helps you think about a user story by considering the followings:

1. The user story will bring value to someone or certain party (e.g. customers).
2. The user story is fulfilling a user's need (e.g. receive an SMS when the item is arrived)
3. There is a reason to support implementing this user story (e.g. customer can go pick up the item she purchased)

Each user story should bring value(s) to someone. But sometimes, the value is obvious enough just by reading the business objective. To write down the value as part of the statement is cumbersome. In such case, we will just skip it. Here are several examples:

As a customer, I want to settle payment by using credit card .

As a user, I want to perform searching by entering my friend's name .

No matter how you write a user story, there are two things that you must keep in mind. First, a user story must be written from user's point of view. Second, keep the description 'just enough'. Avoid adding too much details at the beginning of a software project. Later on you will have chance to refine and detail the user story to make it become something that can be used by the developers in design and implementation.

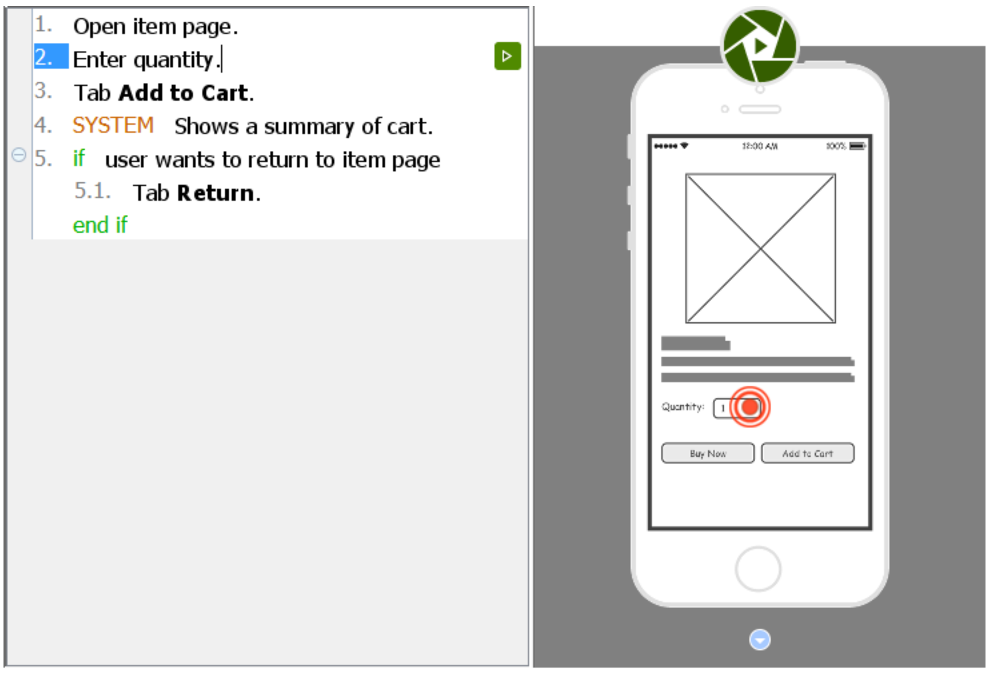
Lifecycle of a User Story

In a broad sense, there are six main states for each user story throughout a software project:

1. **Pending** - Through the communication between user and project team, user stories are found. At this state, the user stories have nothing more than a short description of user's need. There is no detailed discussion of requirements, no system logic and no screen design yet. In fact, the only purpose of user story, for now, is just for reminding all parties for a future discussion of user's request written in this user story (card). It is possible that the user story will be discarded in the future.

2. **Todo** - Through a discussion between different stakeholders, the user stories to be addressed in the next few weeks are decided, and are put into a time-box called a sprint. Such user stories are said to be in the Todo state. No detailed discussion has yet been carried out in this state.

3. **Discussing** - When a user story is in the Discussing state, the end user will communicate to the development team in confirming the requirements as well as to define the acceptance criteria. Development team will write down the requirements or any decisions as conversation notes. UX specialist may create wireframes or storyboards to let user preview the proposed features in visual mock-ups, and to feel it. This process is known as user experience design (UX design).



```
1. Open item page.
2. Enter quantity.
3. Tab **Add to Cart**.
4. SYSTEM Shows a summary of cart.
5. if user wants to return to item page
    5.1. Tab **Return**.
   end if
```

4. **Developing** - After the requirements are clarified, the development team will design and implement the features to fulfill user's requests.

5. **Confirming** - Upon the development team has implemented a user story, the user story will be confirmed by the end user. He/she will be given access to the testing environment or a semi-complete software product (sometimes known as an alpha version) for confirming the feature. Confirmation will be performed based on the confirmation items written when detailing the user story. Until the confirmation is done, the user story is said to be in the Confirming state.

6. **Finished** - Finally, the feature is confirmed to be done, the user story is considered in the Finished state. Typically, this is the end of the user story. If user has a new requirement, either it is about a new feature, or it is an enhancement of the finished user story, the team would create a new user story for the next iteration.

Detailing User Story - When and Why?

A key thing that makes user story different from traditional requirements capturing approaches is that user story approach split the identification of problem and solution into two steps, performed at different stages of a software project. While the problems and a brief understanding of user requests are found at the beginning of the software project, the details of system requirements are found only before the commencement of design and implementation. Here are some benefits of this arrangement:

1. Able to respond to latest user needs because requirements are detailed right before implementation, instead of having everything concluded at the beginning.

2. Able to identify the right requirements easier because both of the problems and solutions will be undergone detailed discussions. In the traditional approaches, since the details of all requirements are required to be found upfront at the beginning of a project, the "upfront requirements" could have been changed overtime, resulting a lot of wastage of analysis affords.
3. - On the other hands, user stories that are found invalid can be discarded easily. You do not lose much time on prior studying and documentation

How to Use User Story Effectively?

Just like many other software development methodologies, if you apply user story properly in your software project you will be able to produce a quality software system plus to win the trust and satisfaction from customers. Here are some points that you need to keep in mind when using user story:

1. Keep the description of user story short.
2. Think from end user's point of view when writing a user story.
3. A (UML) use case represents a business goal. The capability to group user stories under use cases allows you to ensure a user story is satisfying a business goal, in other words, they sharing the same system goal. It serve as a placeholder for you to manage, schedule and prioritize user stories in a more manageable manner.
4. Confirmation items must be defined before you start the development
5. Estimate the user story before implementation to ensure the workload of your team is under control.
6. Requirements are found with the end users, not by the end user or just by the development team. Keeping a good relationship with the end users will be a win-win situation for both parties.
7. Communication is always important in understanding what the end user wants.

Visual Paradigm provides all the tools you need in **agile software development**, which includes **UML Use Case Diagram tool**, **(agile) user story**, **sprint**, **storyboard** and **wireframes** for UX design, **task management tool**, etc.

Try now for FREE

References

1. Learn about Visual Paradigm's agile development tools
2. Tutorial - Business Process to User Story Mapping
3. Essential XP: Card, Conversation, Confirmation

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm handbooks
(https://www.visual-paradigm.com/learning/handbooks/)