User story

In software development and product management, a **user story** is an informal, natural language description of one or more features of a software system. User stories are often written from the perspective of an end user or user of a system. They are often recorded on index cards, on Post-it notes, or digitally in project management software[1] Depending on the project, user stories may be written by various stakeholders including clients, users, managers, or development team members.

User stories are a type of boundary object. They facilitate sensemaking and communication; that is, they help software teams organize their understanding of the system and its context.[2]

Contents

History

- 1998: Alistair Cockburn visited the Chrysler C3 project in Detroit and coined the phrase "A user story is a promise for a conversation."[3]
- 1999: Kent Beck published the first edition of the book *Extreme Programming Explained*, introducing Extreme Programming (XP),[4] and the usage of user stories in the planning game.
- 2001: Ron Jeffries proposed a "Three Cs" formula for user story creation:[5]
  - The *Card* (or often a post-it note) is a tangible physical token to hold the concepts;
  - The *Conversation* is between the stakeholders (customers, users, developers, testers, etc.). It is verbal and often supplemented by documentation;
  - The *Confirmation* ensures that the objectives of the conversation have been reached.
- 2004: Mike Cohn generalized the principles of user stories beyond the usage of cards in his book *User Stories Applied: For Agile Software Development*[6] that is now considered the standard reference for the topic according to Martin Fowler.[7]
- 2014: After a first article in 2005[8] and a blog post in 2008,[9] in 2014 Jeff Patton published the user-story mapping technique, which intends to improve with a systematic approach the identification of user stories and

to structure the stories to give better visibility to their interdependence.[10]

Principle

User stories are written by or for users or customers to influence the functionality of the system being developed. In some teams, the product manager (or product owner in Scrum), is primarily responsible for formulating user stories and organizing them into a product backlog. In other teams, anyone can write a user story. User stories can be developed through discussion with stakeholders, based on personas or simply made up.

Common templates

User stories may follow one of several formats or templates.

The most common is the *Connextra template*, stated below.[11][6][12] Mike Cohn suggested the "so that" clause is optional although still often helpful.[13]

```
As a <role> I can <capability>, so that <receive benefit>
```

Chris Matts suggested that "hunting the value" was the first step in successfully delivering software, and proposed this alternative:[14]

```
In order to <receive benefit> as a <role>, I can <goal/desire>
```

Another template based on the Five Ws specifies:[15]

```
As <who> <when> <where>, I <want> because <why>
```

Examples

Screening Quiz (Epic Story)

> As the HR manager, I want to create a screening quiz so that I can understand whether I want to send possible recruits to the functional manager.[16]

Quiz Recall

> As a manager, I want to browse my existing quizzes so I can recall what I have in place and figure out if I can just reuse or update an existing quiz for the position I need now.[16]

Limited Backup

> As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved.[17]

Usage

As a central part of many agile development methodologies, such as in XP's planning game, user stories describe what may be built in the software project. User stories are prioritized by the customer (or the product owner in Scrum) to indicate which are most important for the system and will be broken down into tasks and estimated by the developers. One way of estimating is via a Fibonacci scale.

When user stories are about to be implemented, the developers should have the possibility to talk to the customer about it. The short stories may be difficult to interpret, may require some background knowledge or the requirements may have changed since the story was written.

User stories can be expanded to add detail based on these conversations. This can include notes, attachments and acceptance criteria.

Acceptance criteria

Mike Cohn defines acceptance criteria as "notes about what the story must do in order for the product owner to accept it as complete."[18] They define the boundaries of a user story and are used to confirm when a story is completed and working as intended.

The appropriate amount of information to be included in the acceptance criteria varies by team/program/project. Some may include 'predecessor criteria', "The user has already logged in and has already edited his information once". Some may write the acceptance criteria in 'typical' Agile format, Given/When/Then. Others may simply use bullets points taken from original requirements gathered from customer/stakeholders.

In order for a story to be considered Done or Complete, all acceptance criteria must be met.

Benefits

There is no good evidence that using user stories increases software success or developer productivity. However, user stories facilitate sensemaking without undue problem structuring, which is linked to success.[19]

Limitations

Limitations of user stories include:

- **Scale-up problem**: User stories written on small physical cards are hard to maintain, difficult to scale to large projects and troublesome for geographically distributed teams.
- **Vague, informal and incomplete**: User story cards are regarded as conversation starters. Being informal, they are open to many interpretations. Being brief, they do not state all of the details necessary to implement a feature. Stories are therefore inappropriate for reaching formal agreements or writing legal contracts.[20]
- **Lack of non-functional requirements**: User stories rarely include performance or non-functional requirement details, so non-functional tests (e.g. response time) may be overlooked.
- **Don't necessarily represent how technology has to be built:** Since user stories are often written from the business perspective, once a technical team begins to implement it may find that technical constraints require effort which may be broader than the scope of an individual story. Sometimes splitting stories into smaller ones can help resolve this. Other times, 'technical-only' stories are most appropriate. These 'technical-only' stories may be challenged by the business stakeholders as not delivering value which can be demonstrated to a customer/stakeholders.

Relationship to epics, themes and initiatives

In many contexts user stories are used and also summarized in groups for semantic and organizational reasons. The different usages depend on the point-of-view, e.g. either looking from a user perspective as product owner in relation to features or a company perspective in relation to task organization.

While some suggest to use 'epic' and 'theme' as labels for any thinkable kind of grouping of user stories, organization management tends to use it for strong structuring and uniting work loads. For instance, Jira seems to use a hierarchically organized to-do-list, in which they named the first level of to-do-tasks 'user-story', the second level 'epics' (grouping of user stories) and the third level 'initiatives' (grouping of epics). However, initiatives are not always present in product management development and just add another level of granularity. In Jira, 'themes' exist (for tracking purposes) that allow to cross-relate and group items of *different parts of the fixed hierarchy*. [21] [22] In this usage, Jira, shifts the meaning of themes in an organization perspective: e.g how much time did we spent on developing theme "xyz". But another definition of themes is: a set of stories, epics, features etc for a user that forms a *common semantic unit or goal*. There is probably not a common definition because different approaches exist for different styles of product design and development. In this sense, some also suggest to not use any kind of hard groups and hierarchies. [23] [24] [25] [26] [27] [28]

Epic

Large stories or multiple user stories that are very closely related are summarized as epics. A common explanation of epics is also: a user story that is too big for a sprint.

Initiative

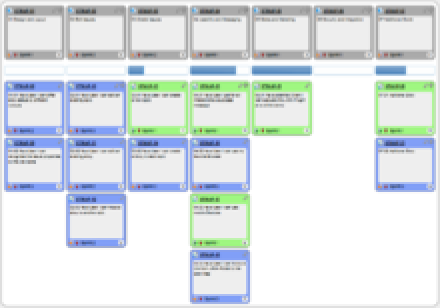
Multiple epics or stories grouped together hierarchically, mostly known from Jira.[29]

Theme

Multiple epics or stories grouped together by a common theme or semantic relationship.

Story map

A story map[30] is a graphical, two-dimensional visualization of the Product Backlog. At the top of the map are the headings under which stories are grouped, usually referred to as 'epics' (big coarse-grained user stories), 'themes' (collections of related user stories[31]) or 'activities'. These are identified by orienting at the user's workflow or "the order you'd explain the behavior of the system". Vertically, below the epics, the actual story cards are allocated and ordered by priority. The first horizontal row is a "walking skeleton"[32] and below that represents increasing sophistication.[33]



A story map in action

In this way it becomes possible to describe even large systems without losing the big picture.

Comparing with use cases

A use case has been described as "a generalized description of a set of interactions between the system and one or more actors, where an actor is either a user or another system."[34] While user stories and use cases have some similarities, there are several differences between them.

	User Stories	Use Cases
Similarities	▪ Generally formulated in users' everyday language. They should help the reader understand what the software should accomplish.	▪ Written in users' everyday business language, to facilitate stakeholder communications.
Differences	▪ Provide a small-scale and easy-to-use presentation of information, with little detail, thus remaining open to interpretation, through conversations with on-site customers.	▪ Use cases organize requirements to form a narrative of how users relate to and use a system. Hence they focus on user goals and how interacting with a system satisfies the goals.[35] ▪ Use case flows describe sequences of interactions, and may be worded in terms of a formal model. A use case is intended to provide sufficient detail for it to be understood on its own.
Template	*As a <type of user>, I can <some goal> so that <some reason>.*[17]	▪ Title: "goal the use case is trying to satisfy" ▪ Main Success Scenario: numbered list of steps ▪ Step: "a simple statement of the interaction between the actor and a system" ▪ Extensions: separately numbered lists, one per Extension ▪ Extension: "a condition that results in different interactions from .. the main success scenario". An extension from main step 3 is numbered 3a, etc.

Kent Beck, Alistair Cockburn, Martin Fowler and others discussed this topic further on the c2.com wiki (the home of extreme programming).[36]

See also

- Kanban board
- Persona (user experience)
- Scenario (computing)
- Use case

References

1. Dimitrijević, Sonja; Jovanović, Jelena; Devedžić, Vladan (2015). "A comparative study of software tools for user story management". *Information and Software Technology*. **57**: 352–368. doi:10.1016/j.infsof.2014.05.012. "a great number of software tools that provide, inter alia, support for practices based on user stories have emerged in recent years."
2. Ralph, Paul (2015). "The Sensemaking-coevolution-implementation theory of software design". *Science of Computer Programming*. **101**: 21–41. arXiv:1302.4061. doi:10.1016/j.scico.2014.11.007.
3. "Origin of story card is a promise for a conversation : Alistair.Cockburn.us". *alistair.cockburn.us*. Retrieved 16 August 2017.
4. Beck, Kent (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley. ISBN 9780201616415. OCLC 41834882.
5. Jeffries, Ron (30 August 2001). "Essential XP: Card, Conversation, Confirmation".
6. Cohn, Mike (2004). *User Stories Applied: For Agile Software Development*. Addison-Wesley. ISBN 0321205685. OCLC 54365622.
7. Fowler, Martin (22 April 2013). "User Story". *martinfowler.com*. Retrieved 14 July 2019.
8. Patton, Jeff (January 2005). "It's All In How You Slice It". *Better Software Magazine*: 16–22, 40.

9. Patton, Jeff (8 October 2008). "The New User Story Backlog is a Map". *Jeff Patton & Associates*. Retrieved 16 July 2019.
10. Patton, Jeff (2014). *User story mapping*. Economy, Peter, Fowler, Martin, Cooper, Alan, Cagan, Marty (First ed.). Beijing. ISBN 9781491904909. OCLC 880566740.
11. Lucassen, Garm; Dalpiaz, Fabiano; Werf, Jan Martijn E. M. van der; Brinkkemper, Sjaak (2016), Daneva, Maya; Pastor, Oscar (eds.), "The Use and Effectiveness of User Stories in Practice", *Requirements Engineering: Foundation for Software Quality*, Springer International Publishing, **9619**, pp. 205–222, doi:10.1007/978-3-319-30282-9_14, ISBN 978-3-319-30281-2, "The most prevalent user story template is the 'original' one proposed by Connextra"
12. "Glossary: User Story Template". *agilealliance.org*. Agile Alliance. 17 December 2015. Retrieved 3 February 2020. "Another name is the "Connextra format", in recognition of its origins"
13. Cohn, Mike (25 April 2008). "Advantages of the "As a user, I want" user story template". *Mountaingoatsoftware.com*. Archived from the original on 18 December 2016. Retrieved 18 December 2016. "While I consider the so-that clause optional, I really like this template."
14. Marcano, Antony (24 March 2011). "Old Favourite: Feature Injection User Stories on a Business Value Theme". *Antonymarcano.com*. Retrieved 23 February2017.
15. "User Story". *t2informatik GmbH*. Retrieved 3 February2020. ""As (who) (when) (where), I (want) because (why)." – this phrase is based on typical W questions: who, when, where, what and why."
16. Cowan, Alexander. "Your Best Agile User Story". *Cowan+*. Retrieved 29 April 2016.
17. Cohn, Mike. "User Stories". *Mountain Goat Software*. Retrieved 27 April 2016.
18. Cohn, Mike. "The Two Ways to Add Detail to User Stories". *Mountain Goat Software blog*. Retrieved 8 April 2019.
19. Ralph, Paul; Mohanani, Rahul (2015). "Is Requirements Engineering Inherently Counterproductive?". *2015 IEEE/ACM 5th International Workshop on the Twin Peaks of Requirements and Architecture*. IEEE. pp. 20–23. doi:10.1109/TwinPeaks.2015.12. ISBN 978-1-4673-7100-1.
20. "Limitations of user stories". Ferolen.com. 15 April 2008.
21. "Epics, Stories, Themes, and Initiatives". *Atlassian*. Retrieved 8 February 2019.
22. "User Stories". *Atlassian*. Retrieved 8 February 2019.
23. Britsch, Marcel (5 September 2017). "The Basics: Epics, Stories, Themes & Features". *The Digital Business Analyst*. Retrieved 8 February 2019.
24. Cohn, Mike. "User Stories, Epics and Themes". *Mountain Goat Software*. Retrieved 8 February 2019.
25. "Scrum Alliance Member-Submitted Informational Articles".
26. Guay, Constantin (26 January 2018). "Scrum tips: Differences between epics, stories, themes and features". Retrieved 8 February 2019.
27. "User Stories, Epics & Themes". 10 May 2016. Retrieved 8 February 2019.
28. Cohn, Mike. "You Don't Need a Complicated Story Hierarchy". *Mountain Goat Software*. Retrieved 8 February 2019.
29. "Configuring initiatives and other hierarchy levels - Atlassian Documentation". *confluence.atlassian.com*. Retrieved 5 February 2020. "An 'initiative' is a very large body of work, which spans multiple epics and sometimes, multiple teams. [...] An initiative is also an issue type in Jira."
30. Patton, Jeff (8 October 2008). "The new user story backlog is a map". Retrieved 17 May 2017.
31. Cohn, Mike. "User Stories, Epics and Themes". *Mountaingoatsoftware.com*. Retrieved 26 September 2017.
32. Cockburn, Alistair. "Walking Skeleton". Retrieved 4 March2013.
33. "Story Mapping". Agile Alliance. 17 December 2015. Retrieved 1 May 2016.
34. Cohn, Mike. "Project Advantages of User Stories as Requirements". *Mountaingoatsoftware.com*. Retrieved 26 September 2017.
35. Fowler, Martin (18 August 2003). "UseCasesAndStories". Retrieved 26 September 2017.
36. "User Story And Use Case Comparison". *C2.com*. Retrieved 26 September 2017.

Further reading

- Daniel H. Steinberg, Daniel W. Palmer, *Extreme Software Engineering*, Pearson Education, Inc., ISBN 0-13-047381-2.
- Mike Cohn, *User Stories Applied*, 2004, Addison Wesley, ISBN 0-321-20568-5.
- Mike Cohn, *Agile Estimating and Planning*, 2006, Prentice Hall, ISBN 0-13-147941-5.
- Business Analyst Time
- Payton Consulting 'How user stories are different from IEEE requirements