

Home¹

Answer

Notifications¹

Search Quora



Add Question or Link

Agile Operations Scrum (product development)

Who writes user stories in Agile with Scrum?

This question previously had details. They are now in a comment.



Answer



Follow · 34



Request



1



Ad by Udacity.com

Explore Udacity online courses.

Master in-demand skills, design amazing projects, and boost your hireability today!

Learn more at udacity.com



26 Answers



Andriy Zapisotskyi

Answered Nov 29, 2017

So, the answer here is a **Product Owner**. There are great answers below, but I will also add, that **writing agile user stories** is one of the most important and essential parts of the Agile process. You work very closely with a dev team to create good stories which will provide valuable input.

As a result, you will improve the current version of a product and maybe even **take it to the next level**.

There's more on Quora...

Pick new people and topics to follow and see the best answers on Quora.

[Update Your Interests](#)

Related Questions

[How do I define tasks vs milestones vs deliverables in an Agile Scrum methodology?](#)

[What type of projects are suitable for Scrum/Agile methodology?](#)

[In agile methodology, should developers under any circumstances be creating user stories?](#)

[What is your favorite online tool for agile \(scrum\) story estimation?](#)

[What's the difference between Agile, Scrum and SAFe?](#)

[How should I write my User Story to define a breadcrumb in a Scrum backlog?](#)

[How do you define a user story for a small bug fix in Scrum?](#)

[Software Engineering: Why do some people hate on Agile/Scrum?](#)

[Do we need to write test cases in Agile \(Scrum\) methodology?](#)

What are top 10 challenges faced during a transition to Scrum/Agile?

[+ Ask New Question](#)

More Related Questions

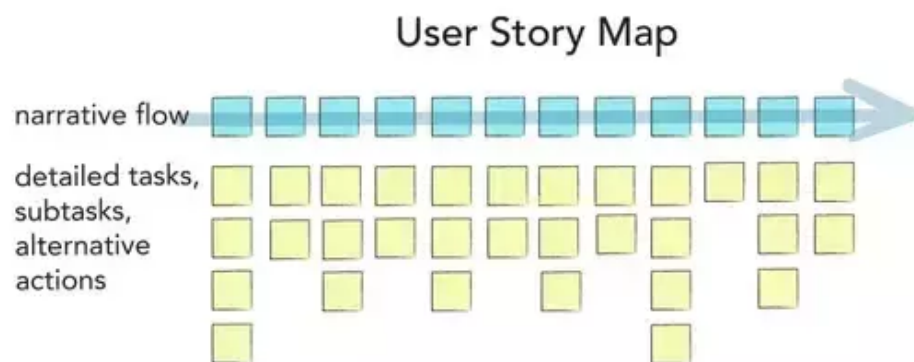
Question Stats

33 Public Followers

35,006 Views

Last Asked Jun 6, 2016

Edits



With user stories in Agile developers can get a precise and clear acceptance criteria, you can easily mitigate delayed feedback risks and this offers a much better way to define what to work on and how to work.

I would like to add **a template of a good user story** :

- User story ID – this is necessary in order to name the story and make it easier to track
- As a <type of user> – here you should integrate the user type and figure out what works great for your products and services.
- I want to <perform some task> – you can add any task you want, as long as it's relevant and connected to the user story
- So that I can <achieve some goal> – the task above should be interconnected with the goal, in fact, it should showcase the means of completing that goal adequately.
- Priority <Low, Normal, High> – prioritizing a task is a crucial thing to keep in mind, and it can provide you with a lot of information.

Hope it will help you! Read more here: [Why User Stories Are A Big Deal And How To Create One](#)

6.2k Views · View Upvoters

 You upvoted this

 Upvote · 5  Share



Recommended All

Promoted by Inflectra

SpiraTeam - The perfect ALM tool for teams.

SpiraTeam helps you manage project requirements, releases, iterations, tasks and bugs/issues. Free trial!

Free trial at inflectra.com



Matthieu Cornillon

Answered Jan 28, 2016 · Upvoted by Cliff Gilley, [10+ years of Agile product management experience](#).

Short version: The Product Owner is responsible for the backlog. If all else fails, they must write the stories. However, I believe this should be a last resort; instead, I strongly recommend that teams strive to create stories collaboratively.

Long version: What is a story, anyway? As noted above, it is not requirements. A popular definition I've heard and used is "a placeholder for a conversation". This recognizes that the story cannot capture it all, and leaves it to someone (I'll get to who) to have a richer conversation to flesh out the details. One limitation of this definition, though, is that it suggests only a future-facing perspective. It seems to say "a story is a reminder to have a conversation in the future," but it doesn't say much about what happens after that conversation.

This occurred to me after a colleague told me about a point Jeff Patton made at [Agile Day '15](#) . (I'll do my own version and apologize to Mr. Patton in case I butcher his point!) Imagine I show you a photo of a woman and a toddler on a bed with an open book in front of them. To you, the picture communicates only what you can see from the camera's perspective in that single moment. You might describe what you see: the woman and girl's appearance, the pattern on the bedspread, a few words in the large-print book. You might describe the action of the scene: a woman and girl talking about a book they are reading. You might even infer a mood from the two faces. Contrast that with what the photo does for me. It is my sister reading to my two-year-old daughter. You can't see much from the background, but I can see right away that it was taken in my grandparents' house, on their bed. I am reminded that this was the first time that my sister and daughter had spent time together in a while, and it warms me to think of this moment of their bonding. I am reminded that the book they are reading was handed down to me by my father. I remember that it was summer, and I can almost feel the humid heat that comes in that time of the year. I also remember it was a great relief for my sister to watch my daughter, as it gave my wife and I time to do some much-needed work on the house. What is striking here is not just the difference between my and your experience of the photo, but the fact that I am *reminded* of these things by the photo. If you asked me about the visit, I might have told you some of these things, but this photo sparks my memory of many specific things *tied to the moment of the photo* that I otherwise *would not have remembered*.

Now, imagine a user story. You have two options. In the first, a product owner writes the story, perhaps with a technical lead. She shows it to the team. They can only derive from it whatever is communicated by the text. Stripped of context, even slightly unclear wording can yield wildly varied interpretations. Contrast this with a second option, where the whole team has a discussion, at

the end of which they write down a story. In this second option, the user story can be a touchstone for the team, reminding them of the rich nuance of the conversation, just as that picture reanimates my memory of a moment more than three years ago for me. The words on the card carry less of the burden, as the memory of the context will fill in the blanks and correct most confusion. People will be reminded not only of what the team concluded, but why they did so. They will remember not only what they are planning to build, but why this is the right thing for the user, who they will remember talking about in their conversation.

This is where I would embellish the definition (and tarnish it through verbosity, I'm afraid): "a story is both a placeholder for a conversation and a memento of that conversation's rich detail". Both the future- and past-facing aspects of that definition suggest involvement by the whole team.

There's another reason that suggests full-team involvement. Pete Behrens introduced to me the distinction between *negotiable stories* and *solution stories*. A solution story tells you what to build. It might give you some context, but it pretty much just says "do this". A negotiable story, on the other hand, leaves things much more loosely defined. It focuses on the problem you are trying to solve, the reason why you are doing anything at all, rather than the specific solution you are being asked to implement. Eventually, you will want to nail down some specifics, and you might even convert your negotiable story to a solution story. However, if you start by discussing a negotiable story with the whole team, you get several benefits. First is deeper understanding of context, the problem at hand, the reason to do anything at all. No matter what happens next, the team has that informing the many decisions they will have to make as they move forward. Second, it leaves the door open for new ideas to emerge from the team. Instead of betting that the Product Owner can come up with the

best solution, you engage the team's creativity, and give new ideas the chance to emerge. (Also, remember that this isn't just about polling more individuals for ideas; it's about letting ideas evolve through the discussion, taking a raw kernel and developing it through what it sparks for others.) Third, it gives the team a feeling of involvement in what they are doing, which is always good. You just don't get these benefits if you skip the part where the team considers a negotiable story.

If it's such a good idea to have teams involved, why doesn't everyone do it this way? Well, first of all, I could be full of baloney with all of this. But even if you think what I say makes sense, you will see people hesitant to spend this much time. People are worried that it is just another meeting instead of "real work", and the enormous benefits of full-team involvement are counter-intuitive and somewhat intangible. It will take some convincing to get people to give it a try. But in my mind, it's an easy call because it's a sure thing. The team builds better software, they enjoy work more, and they do it faster (yes, even counting the meeting, because the team familiarity with the story greases the wheels at countless points down the line).

At the very least, give it a try for a few sprints. If you are struggling with it, make sure you understand why before discarding the method. Is your Product Owner really letting go of the solution and creating space for innovation to emerge? Is the meeting facilitated such that everyone has a voice? Are people using that voice, or sitting quietly through the meetings head down in laptops or phones? Are you really working to understand the user? To understand what problem you are trying to solve?

Good luck!

9.5k Views · View Upvoters

Your feedback is private.

Is this answer still relevant and up to date?

Yes

No

 Upvote · 12

 Share



Add a comment...

Recommended All

Promoted by Atlassian

Finally, a place for posting Atlassian product focused jobs.

Have an opening? Looking to network? Come to the Atlassian Community's Jobs & Careers group!

Learn more at community.atlassian.com

...



Lukasz Nowacki, Scrum Master @ Neoteric with over 10 successful projects

Answered Aug 29, 2017

In our software development company **Neoteric** - we write user stories or scenarios together with team. But here is really one important thing - **quality**. The **Product Owner** is responsible for this quality. He also has a final opinion whether the story is correct or not.

By the way, the scenarios are quite good for us - you can read more about it here:

[Speak Gherkin And Learn How To Collect Requirements For Your Project](#)

7.2k Views · View Upvoters

 Upvote · 18

 Share



Add a comment...

Recommended All



Viktor Grgic, Agile Coaching, Software Development and Architecture

Answered Apr 30, 2014

In most cases I would advise to have team members write User Stories. I have good experience with this. This does not mean they get to decide what is on the cards. It is very much a collaborative process where stakeholders and users present information, and team extract this information and gives feedback in a very proactive way. The act of writing a User Story forces the team to keep listening, understanding, exploring, asking, instead of becoming passive and only receive the information from PO. The role of Product Owner in this process, among other things, is much more to make decisions about whether to spend money on a User Story now, later, or never.

3.2k Views · View Upvoters

 Upvote · 2

 Share



Add a comment...

Recommended All



Mia McCroskey, Certified Scrum Master, Practitioner, Agile PM, a decade of experience.

Answered Feb 18, 2016

There's a lot of good info here so I will strive not to duplicate. The (hopefully) new thought I want to add contradicts some.

First, look at the standard structure of the user story:

"As a <persona> I want <business need> to that <business benefit>."

In my experience members of the development team rarely get that last part right. They usually express it, and the second part too, as functions of the software. After all, that's their business. For them, making the software work

better IS adding business value. They easily lose sight of whether the software is actually solving a real-world problem, or the right real-world problem.

If you can afford to send your scrum team to observe real end users and see how they use the software, then they'll be able to write some good stories. But it doesn't stick. Soon enough you'll be getting more "as a developer I want all the menus to be hierarchical so that I don't have to click so much" type stories. Because not clicking so much is inherently good and therefore provides business value, right? (Yes, I know, menu structure should be a GUI standard. Forgive my poor example.)

The people who are out there talking to and observing your client end users -- your support team, or people in a client services role -- are the ones who should, if not write, then at least directly inform the user stories. They know the real world pain points. The role most likely to be able to work with them to write user stories that meet INVEST* is the Product Owner. To back up the PO you can use a business analyst. I like having a BA as part of the scrum team. A good BA continually reinforces the business need as the team hashes out the technical detail. A strong BA prevents the team from delivering beautiful code that just misses the mark.

*INVEST is an acronym that expresses the test of a good requirement. Is the user story Independent, Negotiable, Valuable, Estimable, Small, Testable?

3.9k Views · View Upvoters

 Upvote · 3

 Share



Recommended [All](#)

Jayadeep Subhashis, Senior Content Writer/Editor at VMEdU.com (2015-



present)

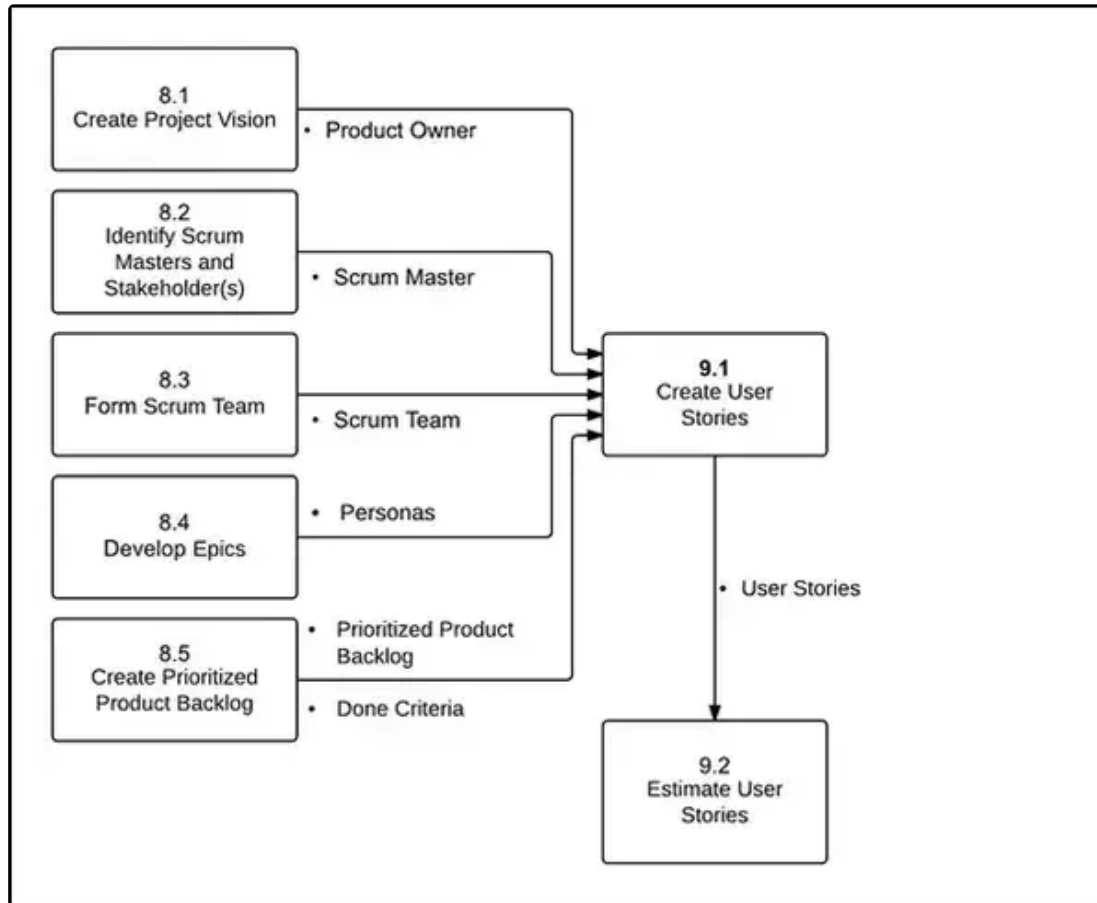
Answered Jun 26, 2017

Keywords: Product Owner, User Story, Prioritized Product Backlog

The Product Owner, based on his or her interaction with the stakeholders, business knowledge and expertise, and inputs from the team, **develops User Stories** that will form the initial Prioritized Product Backlog for the project.

The Prioritized Product Backlog represents the total sum of what must be completed for the project. **The objective of this exercise is to create elaborated and refined User Stories that can be estimated, and committed to by the Scrum Team.** At times, the Product Owner may bring a Business Analyst to assist with writing User Stories.

Now for better understanding, let us have a look on the data flow diagram which depicts how to create User Stories.



Although the Product Owner has the primary responsibility for writing User Stories and often carries out this exercise on his or her own, a **User Story Writing Workshop** can be held if desired.

For more about User Story Writing Workshop, Please Follow: [Download Free](#) or [Buy the SBOK™ Guide](#)

4.7k Views · View Upvoters

 Upvote · 2  Share



Add a comment...

Recommended All



Peter Trudelle, 10 years scrumming for fun & profit

Answered Apr 26

Nobody, ideally, because user stories are meant to be told and listened to together in real-time, not written to be read individually later.

And everybody, in that all involved can use this narrative form to guide the team's work.

It sounds like you see user stories as a direct replacement for requirements or 'specs', but that would be misusing them; they are intended as a completely different process. Think of the canonical user story formats as a ticket to a conversation. They don't contain requirements, they refer to their existence. The actual requirements emerge during the conversations that iteratively refine the story. The balance comes from having the right people at those discussions.

521 Views · View Upvoters

 Upvote · 2

 Share



Add a comment...

Recommended All

Top Stories from Your Feed