AGILECONNECTION™
A TECHWELL COMMUNITY

Published on *AgileConnection* (https://www.agileconnection.com)

18 Questions To Ask For Better Backlog Refinement

By Derk-Jan de Grood - January 29, 2020

During sprints, Scrum teams work on the items in their sprint backlog. Since the purpose of the team should be to deliver value, their focus should be on completion of the items they've committed to.

However, there's more to think about than only the current sprint. Teams are expected to prepare the product backlog items that they will work on in later sprints as well.

Refinement is time spent during the current sprint discussing and elaborating product backlog items so that they are ready for future sprints. The importance of refinement is underestimated by many teams, which is a pity since it has some nice advantages.

Good refinement:

- Leads to better solutions, since it's the product of many smart minds
- Reduces the number of surprises during the sprint, since the team thought about risks and dependencies in advance
- Provides accurate information about the technical complexity, which can be used to plan and prioritize the product backlog
- Leads to better estimations and thus better predictability
- Leads to better quality and efficiency, since the team can divide tasks in such a way that everyone contributes with their specialized knowledge
- Ensures that the team delivers the highest value first
- Gives stakeholders time to define and discuss their needs with their supporters, such as operational employees
- Increases flexibility, because changes in priority have no impact on team effectiveness
- Results in commitment by the whole team, since everyone is involved the definition of the solution

Unfortunately, many teams do not unlock the full potential of refinement.

Not only is the time spent on refinement often limited, but many of the refinement meetings I join are inefficient. I meet teams that spend half the meeting watching the product owner entering the new backlog items in the workflow system. Although they estimate the user stories afterward, little time is left to discuss the best solution and risks that need to be avoided.

Often, user stories are owned by one developer who is regarded as the expert, and other team members are not really involved and focus on their own user stories instead. Again, little time is spent on exchanging ideas in order to come up with a better solution.

I coached a distributed team where half of the team worked abroad. "Since our team members sometimes have problems with the language of the requirements, we read the user story aloud during the online refinement meeting," the product owner explained. "This way we all get involved and have the same understanding of the backlog item."

This seemed like a waste of valuable time. I explained to them how to increase their impact: by seeing refinement as a process, not just as a meeting. In a process, the backlog item is sliced, and a solution is proposed, reviewed, and discussed. Refinement becomes a series of activities like thinking, writing, reviewing, discussing, and preparing.

"Preparing?" Several team members in the room chuckled. "We don't prepare our refinement meeting."

It's crucial to allot enough time for preparing your refinement. You can do this as a team, but I think it's fine when one developer or business analyst takes the lead in defining a user story and proposes a solution.

If you see refinement as a process, this can be done way before or in between two refinement meetings. This way other team members and even stakeholders have enough time to get involved. When they have trouble with the language, they can read the user story at their own pace and give it a thought. These thoughts can be shared and discussed by email, in virtual whiteboard meetings, or during the next refinement meeting.

There's no better way to gain clarity than to ask questions, so I came up with 18 example questions for team members to consider that should trigger refinement discussions.

If you didn't prepare for your refinement meeting, you can select any question from the list and gain better insight into the desired solution. If a question seems irrelevant, skip it.

If you like to prepare, you can also use these questions to involve participants who are otherwise reluctant to speak their mind or who are less involved. Ask them to select a question and prepare the answer ahead of time. This way they come to the meeting prepared, and you can discuss their opinions and answers.

1. Do we understand what we can do with the system after the story user is completed (that we cannot do with the current system)?
2. How would you test this user story?
3. Where do we see dependencies with other user stories or teams?
4. What architectural challenges do you see?
5. What parts of the implementation looks like code that already exists elsewhere, and should we reuse it?
6. Are there security aspects we need to take into account?
7. Are there performance or UX aspects we need to take into account?
8. Is there any developer that thinks they cannot build this? Why?
9. What should we demo to show it works as planned?
10. Do we know any other implementation of a similar solution (in our own organization or elsewhere), and where does our solution differ?
11. Is there any other implementation that we can do that is clearly not what we want, but that satisfies the requirements anyway?
12. What other interpretation could a developer use to build the user story, and what test do we need to perform to detect this?
13. Where do we create technical debt if we implement this user story as planned?
14. What vague terms are used in the user story—e.g., *fast*, *soon*, *early*, *better*, *easy*?
15. What tasks do we need to complete for this user story, and who wants to do (or not do) what tasks?
16. What skills are needed to complete this user story, and do we have these skills in the team?
17. Is it clear how the user story contributes to the product vision of the product owner?
18. Can we split the user story into smaller ones and still add value?

Depending on context and the type of user stories you're working with, other questions can be more suitable than the ones listed here. If you have suggestions, please do share them by commenting below.

About the author

Derk-Jan De Grood

Derk-Jan de Grood works as agile transition coach for Squerist. He has worked for organizations like Nationale Nederlanden, ING Bank, DPD and Greenchoice and supported them in their Agile Transformation. He wrote several successful books and frequently publishes articles and columns for the major magazines. In 2014 he won the EuroSTAR testing excellence award. In 2016 he published his 7th book: "Agile in the real world, starting with Scrum". He is an experienced trainer, workshop host and a regular (keynote) speaker at conferences like Agile Summit Ibiza, Agile Amsterdam, Testing Uruguay, the Agile Testing Days and STAR conferences in Europe and America. On his own blog he shares his knowledge and experience for everyone to benefit.