Product Backlog Refinement Explained

By Stephan van Rooden

Product Backlog Refinement Explained

One of the most challenging activities in Scrum is Product Backlog Refinement. What do you do during Product Backlog refinement? How do you prevent discussions going off track or in too much detail? Who should be there? When do you estimate? In this whitepaper, you will get some good practices and guidelines for having better, more effective and more vivid Product Backlog refinement. This paper consists of three parts:

1. Before you bring an item into a meeting
2. What do you typically do during a meeting focusing on refinement?
3. Facilitating a meeting on Product Backlog refinement

Before you bring an item into a meeting

'Ready state'

The goal of Product Backlog refinement is to work with the Scrum Team and stakeholders (when relevant), to get Product Backlog items in a 'ready state'. What does this mean? This basically means that the development team has the idea that an item is:

1. Clear enough, so they understand what stakeholders are asking for and why they are asking for it.
2. Small enough, so the items should be small enough to get done within a sprint (typically a few days of work) to comply with the definition of done.

This activity is all about interaction between the Product Owner, Development Team and stakeholders. If you were expecting a blueprint for a 'ready' item you clearly need to do some homework on agility. When an item is ready depends on many different aspects like experience of the Scrum Team or knowledge about the product. It even differs per item when a Development Team considers it to be ready. This activity takes time and doing this right saves a lot of time in Sprint Planning.

Before you bring it to a meeting

The most important part of Product Backlog refinement actually is before you start refining. The most ineffective use of a Scrum Team's time would be to refine an item that doesn't contribute to the product vision. For a Product Owner, one of the first steps when a stakeholder has an idea is to find out what this person would like to have and why they need it. A common pitfall is that a stakeholder asks for a solution, the 'how', and a Product Owner in all it's enthusiasm fails to retrieve what they would like to have and why they need it. Numerous times I have seen stakeholders ask for an iPad app. This sounds incredibly valuable and any developer would like to spend time on this over working on some legacy application. However, when the reasons behind the solution are unclear it will most likely end up somewhere hidden in the app store.

So, knowing the reasons behind the idea makes it easy for a Product Owner to judge whether this idea contributes to achieving the long-term vision of the product. If not, a Product Owner might consider to work on it anyway since it opens up a new market opportunity but most often it would be better to focus on those items that contribute to the team's reason for existing. If these boxes are all checked then comes the most important decision. With the knowledge the Product Owner has, is it a valuable idea to create? Does this item have a fighting chance to be picked up by the team or will it end up somewhere at the bottom of the Product Backlog. I have seen Product Owners collecting every question ever being asked over a course of 8 years ending up with 12,000 items on the Product Backlog. A complete waste of time and effort. Even a product backlog with 100 items might be too much for a single team. As a Product Owner, you should ask yourself, how big is the chance of this item being picked up if it initially will be placed on the 40th position on the Product Backlog?
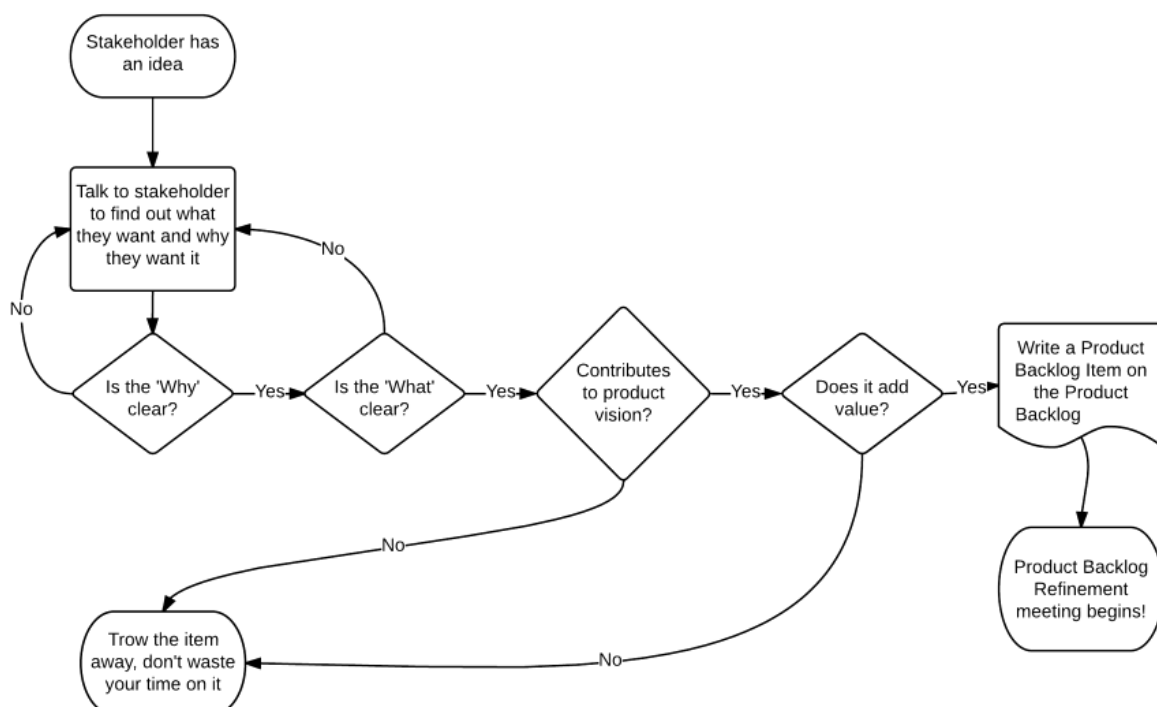
Wouldn't a stakeholder be happier in the long run, if you spare him the false hope of his idea ever to be picked up? As you can see in the image above, there will be a lot of conversation between a Product Owner and stakeholder before it ends up on a product backlog. If the Product Owner does not consider the idea to be valuable, the stakeholder has two options, provide a better business case for the idea or just accept it just wasn't a good idea to begin with. Our best advice for good Product Backlog refinement is to prevent everything to be discussed in Product Backlog refinement.

Stages of Readiness
One of the key pillars of the Scrum framework is 'transparency'. For managing the Product Backlog, this means that it should be visible for the Scrum Team and stakeholders what the order is and in what stage of readiness a particular item is. The image below shows an example of a Product Backlog Kanban board.

Typically a Product Backlog item goes through three refinement meetings before it is considered to be in a ready state. First, when a stakeholder comes with an idea or wish, the team would roughly estimate the size of the item. A very fast way to do this is by 't-shirt sizing'. Nobody knows the exact size of a small sized t-shirt but everybody has an idea about the relative difference in size between a small and medium sized shirt. It is the first input for a Product Owner to get an idea on the effort involved in realizing the item. The second part is to assign story points to the item but again in a quick and dirty way. The format often used is Magic estimation or Silent estimation. This is estimating effort without having long and in depth discussions on the item. The final stage before an item is considered to be ready by a Development Team is to do planning poker. This is a frequently used technique for estimating items. This technique is time consuming so preferably you would only apply this for items you actually want to be realized and based on previous estimation you have considered to be valuable enough to spend the effort.

## What do you typically do during a meeting focusing on refinement?

The refinement meeting
The Scrum guide states:

The Scrum Team decides how and when refinement is done. Refinement usually consumes no more than 10% of the capacity of the Development Team.

In practice, this means that most Scrum Teams plan three time slots of each one hour, throughout the sprint where they spend time with the Product Owner and stakeholders. This should be enough time for a Product Owner and Development Team to create a flow of items in a ready state. Ideally, a Scrum Team has 2 sprints of 'ready' work on the product backlog so if a product owner goes on a holiday or falls ill, the team can go ahead. We find teams struggling to gets enough work in a ready state for the upcoming sprint.

Vague items being brought into Product Backlog refinement and the Development Team getting caught up in discussing any possible solution are signs of refinement gone wrong. Such discussion are consuming the energy of everyone in the meeting, including those involved in the discussion. When facing this teams often set a 10 minute time box to discuss a Product Backlog item. If after 10 minutes there is still no direction of a solution, the discussion is stopped and the Scrum Team decides what to do next. For example, it might be that the Product Owner needs to verify with his stakeholders some assumptions or that the Development Team needs to do some homework on the possible solutions.
Sometimes it might be useful to involve stakeholder in product backlog refinement. When stakeholders and the Development Team are in direct contact with each other, it prevents both sides from making estimates based on assumptions. Like Al Sapienza said in the Steven Seagal movie 'Under Siege 2':
"Assumption is the mother of all fuck ups."

Product Backlog Kanban Board — Tshirt Sizing — Magic Estimation — Ready 4 Sprint
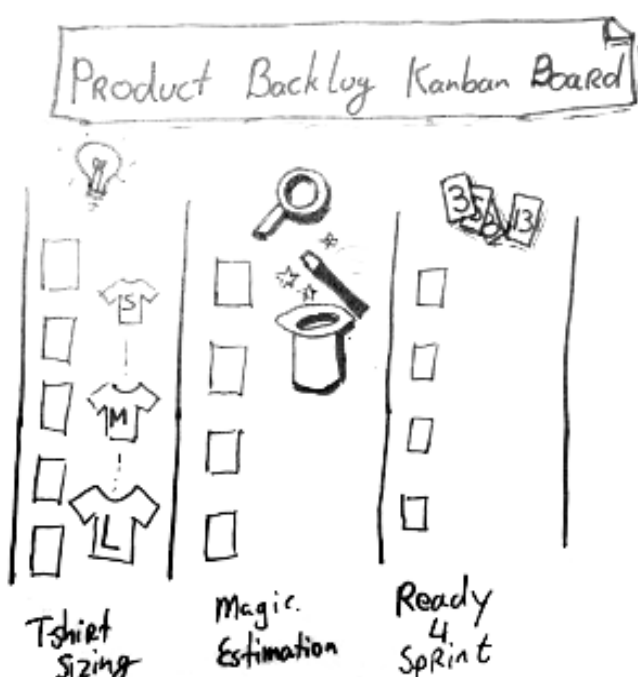
Spikes

Spikes finds their origin in Extreme Programming and are [described](#) as 'A story or task aimed at answering a question or gathering information, rather than at producing shippable product'.

So during Product Backlog refinement a Development Team can decide to create a spike. This spike will be added to the Sprint Backlog of the current Sprint. Preferably will bring back a result before the next meeting so the item can be brought a step closer to being ready.

Two characteristics of a [spike](#):

- **Have clear objectives and outcomes for the spike.** Just like any other sprint backlog item
- **Be timeboxed.** Start with the timebox of one hour and after that decide if you need more time.

The biggest risk when introducing spikes to a Scrum Team is that they embrace this type of task as a tool to create detailed plans and designs. A spike is an exception, not the rule!

Activities during Product Backlog Refinement

Performing the activity of Product Backlog Refinement is of primary interest to the Product Owner. It is up to this role to have an effective Product Backlog refinement. For each item they decide to bring forward during refinement they need to have a clear idea of what they would like to achieve for this item. In the movie by Henrik Kniberg, '[Agile Product Ownership in a Nutshell](#)', three things you typically do during Product Backlog Refinement are mentioned. Slicing, estimating and writing acceptance criteria for Product Backlog items in collaboration with stakeholders..

Slicing

As stated earlier ,the purpose of Product Backlog refinement is to get Product Backlog items in a ready state. This means that an item should be small enough to be picked up in a Sprint. This may sometimes take some creativity to achieve. The [story splitting cheat sheet](#) is a great tool to help teams with exploring possibilities for splitting items. To practice slicing there is a great game to have teams see the value of splitting stories, Alistair Cockburn created the [Elephant Carpaccio](#) exercise. A word of warning, some teams have figured out how to slice items, they tend to slice up beyond the point it being necessary to make an item smaller. This is the time where a Scrum Master of Agile Coach should intervene and explain the team the purpose of slicing.

Estimation

One of the most debated activities when teams apply the Scrum Framework is the point of estimation. Scrum simply states that items should be estimated, however how to estimate is left blank. Whatever work best in your situation.

First, let's be clear, you can always estimate! Even if something is unclear, you can come up with an estimate (most likely something big). Assigning an estimation is not the same as getting an item in a ready state. Product backlog refinement would be a lot less difficult and time-consuming, if everyone involved agrees that an estimation is by default incorrect. It doesn't matter which technique you apply. If you cannot get past that point, any technique will result in the same frustration. If you do get everyone to agree on this, then you might consider the following techniques.

- T-shirt sizing
  - A great technique for quickly estimating an item. It's fast and easy to use. Everyone has an idea of the concept of small, medium or large.
- Magic estimation, a technique for fast estimation of multiple items
  - I would recommend reading [this blog](#) on how to do this activity.
- Planning poker
  - Best known technique for facilitating team estimation. [Planning Poker](#) has its origin in the widepand delphy method and was made popular by Mike Cohn. A time-consuming technique but very effective.

Acceptance Criteria

Adding acceptance criteria to a new feature is not new, we have worked this way for decades. This activity is done together with the entire Scrum team, preferably during Product Backlog refinement. The level of detail when writing acceptance criteria depends on;

- The item at hand
- The level of knowledge and experience of the people involved

and most of all;
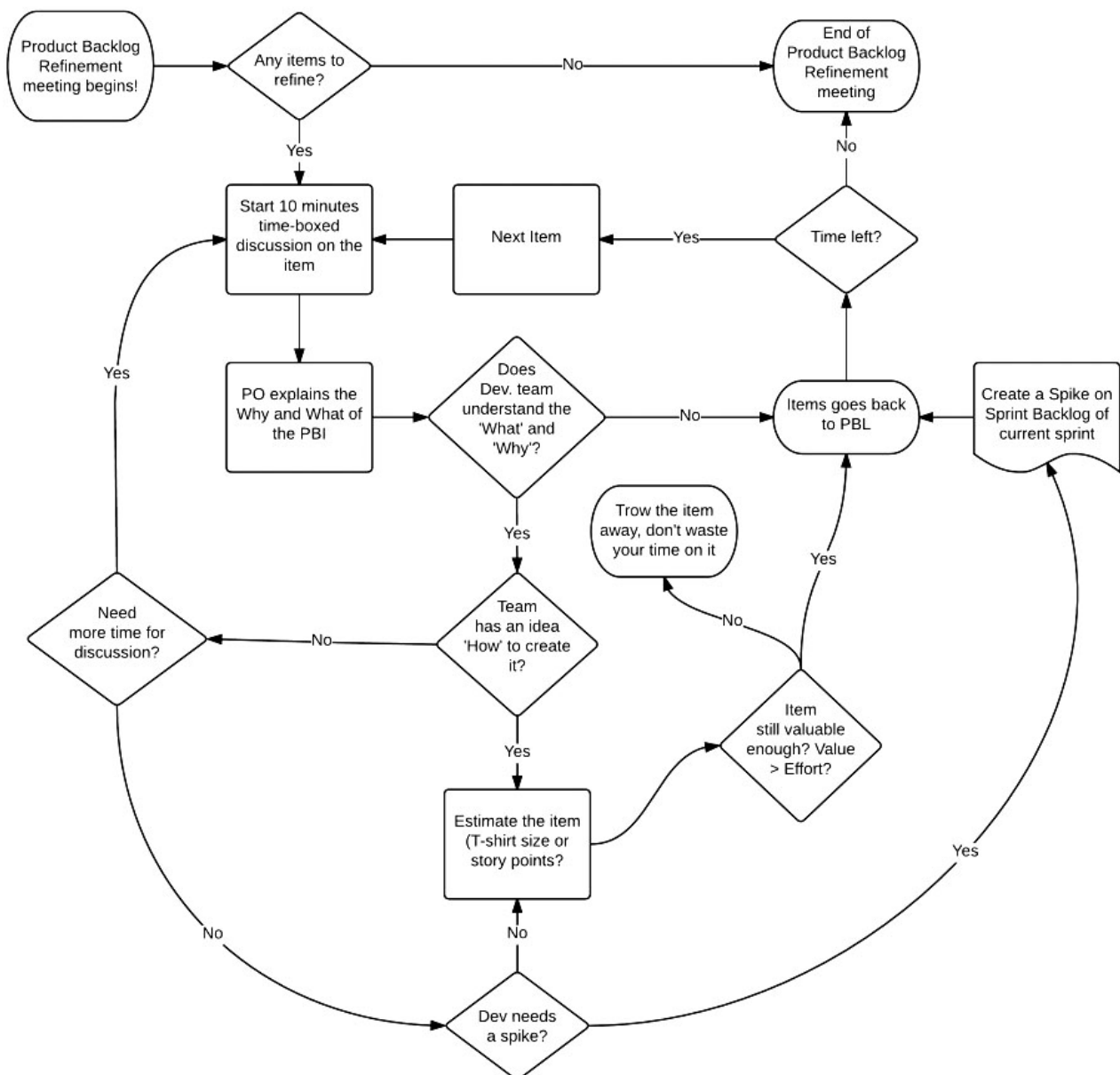- How good the Product Owner and the Development Team interact.

Facilitating a meeting on Product Backlog refinement

All the theory on Product Backlog Refinement is very nice and understanding the goal and the mechanisms at play during this activity might be very useful. Most questions from Scrum Masters is how to keep a meeting on track. Below you can find a graphical representation of what steps you take and which decisions you make during a refinement meeting.

Needles to say, is that there should be not meeting held if the Product Owner has nothing to be refined. Once gathered it never hurts to repeat why we have an activity called Product Backlog refinement so everyone involved is reminded of the value of this meeting.

First action to take when starting to refine a product backlog item is to have a time-box of 10 minutes to discuss the product backlog item. After those 10 minutes you are reminded to reflect on the next step to take. Making this a working agreement for Product Backlog refinement is a great way to embed this in the Scrum Team.In those ten minutes the Product Owner starts with explaining 'what' the product backlog items requires. What is the problem or wish a customer needs a solution for?

Followed by 'Why' this item has any value for a customer/user and for the Scrum Team's vision. If the Product Owner is unable to explain this to the Development Team the Product Owner has some homework to do. It does not make sense to continue refinement on this item and the time will move back to the product backlog. If there is time left for a new item to be discussed, the Product Owner decides which item to discuss next.

When this item under refinement is clear and the team has an idea how to create it. The team can estimate the item. It doesn't matter if the item is too big to complete in a single sprint. The estimation provides the Product Owner with input on the ordering of the Product Backlog. If the team has no idea how to create a Product Backlog item, there is a possibility to start another 10 minute time-box to explore this. Or the Development team identifies a spike to further investigate possible solutions.

As you can see in the image, there are a few decisions to be made during a Product Backlog refinement meeting. This overview will help a Scrum Team in losing focus or forgetting options to get an item in a ready state. This flow has been created by combining successful practices applied by several clients and is a useful 'cheat sheet' for Scrum Masters and Product Owners to get the focus back into Product Backlog refinement. Good luck!