July 2016

Scrum.org Whitepapers

Cross-Team Refinement in Nexus™

Rob Maher, Patricia Kong

In Scrum, Product Backlog refinement is an ongoing activity for a single team; however, it is not a mandatory event. As explained in the Nexus Guide, due to the added complexity of many teams working together on a single product, Refinement is an official and required event in the Nexus Framework. Teams in a Nexus need to be involved in a shared Refinement event, because Refinement is focused on decomposing Product Backlog Items (PBIs) enough so that the teams can understand which work they could deliver and in what sequence over upcoming Sprints. The event also allows the teams to focus on minimizing and removing dependencies between teams.

In a Nexus, there are many Scrum Teams pulling work from a single Product Backlog; therefore, new Refinement questions need to be answered.

- Which teams pull what work?
- How can we best sequence the work, across Sprints and teams to balance early delivery of value against risk and complexity?

When scaling Scrum, we suggest that these questions are best answered by the Development Teams themselves through Cross-Team Refinement. In this paper, we describe how a Nexus can approach a Cross-Team Refinement event.

Cross-Team Refinement

Representatives from each team attend. These representatives should be selected and attend the workshop based on the work being refined instead of the role that they play inside their team. It may be common to have different people attend these workshops based on the skills required.

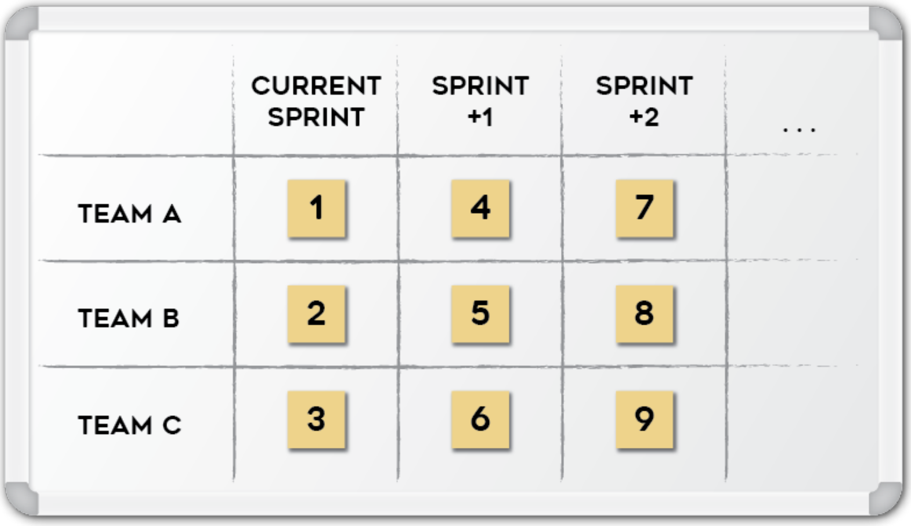
Which teams pull what work?

Decompose and sequence Product Backlog Items

The Product Owner should explain the PBIs to be refined. These are normally large items that have not yet been decomposed or sliced to a size that will fit inside a Sprint. Delaying this breakdown allows the opportunity to decompose the items based on skills and dependencies that emerge during Refinement.

Expect initial conversations to focus on the liquidity of skills. (e.g. which teams have the skills necessary to undertake what work.) Consider that choices may be constrained around existing specialist skills that are not available in each team. Once representatives from the Development Teams start to understand the PBIs, they can decompose them into smaller items, which can then be taken back to their teams for normal Product Backlog refinement.

Figure 1. Understanding the flow of work for the upcoming Sprints and visualizing it per team.



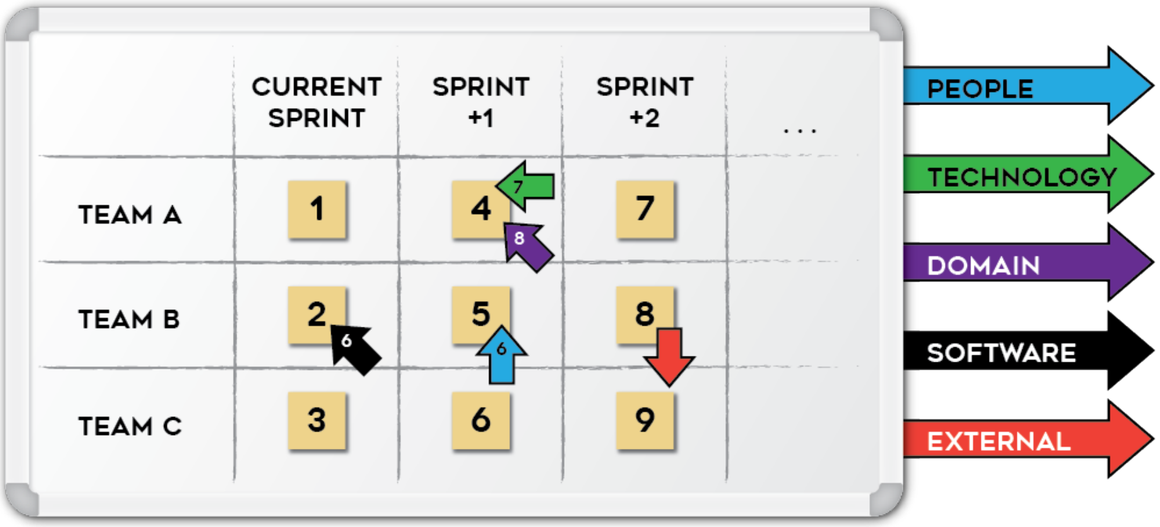
How can we best sequence the work, across Sprints and teams to balance early delivery of value against risk and complexity?

Visualize and Manage Dependencies

As teams are being decomposed, dependencies will emerge. We recommend categorizing dependencies. Categories for dependencies may include:

- Build Sequence – An item cannot be completed until its parent is complete (can include technology, domain, software…)
- People / Skills – Only certain people / teams can complete an item
- External – The parent item is being delivered outside the Nexus

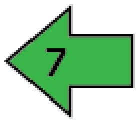
Figure 2. Dependencies can be visualized on a Cross-Team Refinement board.



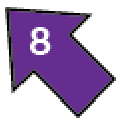
In the example in Figure 2, the teams are shown vertically while the Sprints are on the horizontal. Dependency types have been color-coded, which allows us to see dependency categories as well asa any recurring dependencies. At a minimum, consider identifying *external* dependencies with a color. You may want to add additional color codes to denote common causes of dependencies in your organization e.g. Operations, Legal, DBA Team etc.

Dependencies should be represented in the form of arrows (*dependency arrows*), because the direction of the arrows indicates parent to child relationships. (e.g. Item number 8 *depends on* Item number 4.) Commonly, teams will write a child ID on the parent card also. In the example above, card 7 could have the number 4 written on it also. It is also important to represent dependencies in arrows, because their direction informs delivery risk.

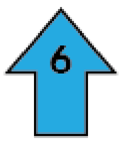
Dependency arrows on an item highlight relationships of work. More arrows indicate high risk due to the number of dependent items impacted. This visualization helps the teams within the Nexus identify the 'critical path' of work throughout the upcoming Sprints and provides the basis for conversations about ways to remove or minimize the impact of these dependencies.



A dependency arrow that is horizontal represents a dependency within a single team across time. It means that a single team is building an item in one Sprint that is needed by an item that will be delivered in a subsequent Sprint. This can be considered a low risk relationship.



A dependency arrow that is diagonal represents a dependency that is across teams and across time. A team is building an item in one Sprint that is needed by an item that will be delivered in a subsequent Sprint *by a different team*. Cross-team collaboration and communication will be vital to success. This is a medium risk relationship.



A dependency arrow that is across teams within a single Sprint is vertical. One team will build an item in a Sprint that is needed by an item that will be delivered *in the same Sprint by a different team.* This dependency gives little room for delay or unexpected complexity. This is a high risk relationship.

External Dependencies typically do not have an ID as they are delivered from a team outside the Nexus and can be represented as shown below.

↙ This downward facing diagonal dependency arrow is external across time. It represents that a team is relying on an item delivered by an external group in order to build a subsequent item.

↓ This downward facing vertical dependency arrow is external and represents another in Sprint dependency. A team is relying on an item delivered by an external group *in the same Sprint* in order to build a subsequent item. This is an extremely high risk item.

Once dependencies have been visualized and the complexities are understood, teams might move toward a model where a single team owns a feature from start to finish as this is a good way to minimize cross-team dependencies. However, keep in mind that this model might cause delayed delivery. Teams in a Nexus may still find occasions when an item can be scaled across multiple teams concurrently in order to bring forward delivery and release value as soon as possible.

Once dependencies have been visualized and sequenced, conversations should focus on opportunities to minimize and remove them. Some solutions include:

- Moving work between teams so that there are less cross-team dependencies.
- Moving people between teams so that there are less cross-team dependencies. You may significantly reduce delivery risk if certain skills are rebalanced across teams for a Sprint or 2 in order to minimize dependencies.
- Reshaping the work. By splitting items in different ways it may be possible to eliminate dependencies.
- Using different risk-based strategies. Some groups might try to entirely remove an 'in-Sprint' cross-team dependency. Other groups may opt to front load all the risk as early as possible and take many cross-team in-Sprint dependencies in earlier Sprints in order to learn and respond.

Continuous Cross-Team Refinement

Once your Cross-Team Refinement workshop is complete, the Cross-Team Refinement board should remain as an up-to-date visualization of current plans for approximately the next 3 Sprints. The board should be a focal point for risk-based conversations with appropriate stakeholders.

Refinement is a continuous event, so Cross-Team Refinement should be scheduled regularly for teams to discuss new items as well as for time to adjust the board as needed with any new information. For instance, using the Cross-Team Refinement board as a focal point (*Figure 3*), we can see that if Item 2 is not finished in the current Sprint, then it will move into the next Sprint. This slip will create an additional *in-Sprint* dependency between items 2 and 6 (*Figure 4*). We can see that there is already an in-Sprint dependency in the next Sprint between items 5 and 6.

Figure 3.

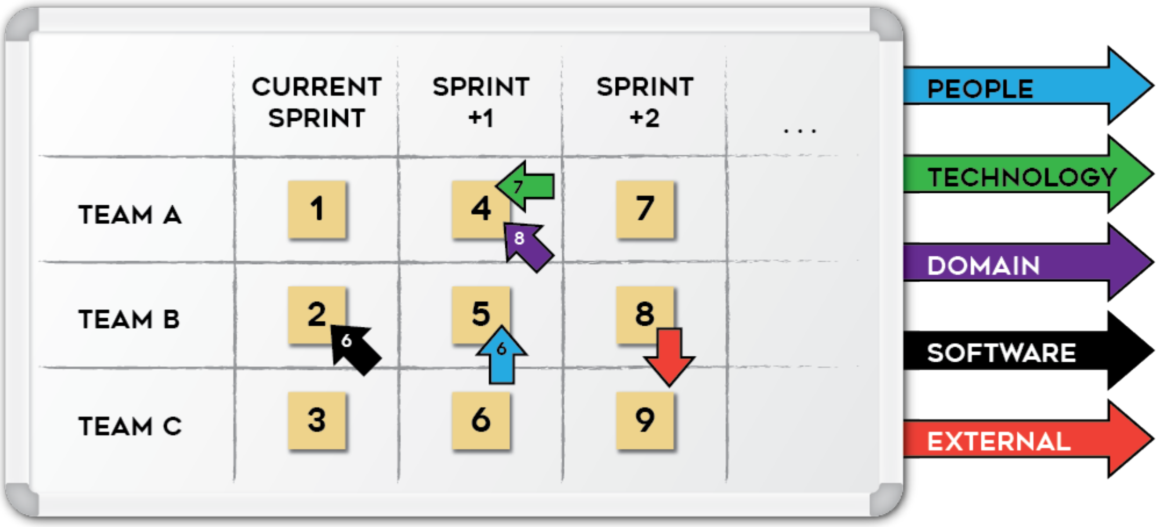
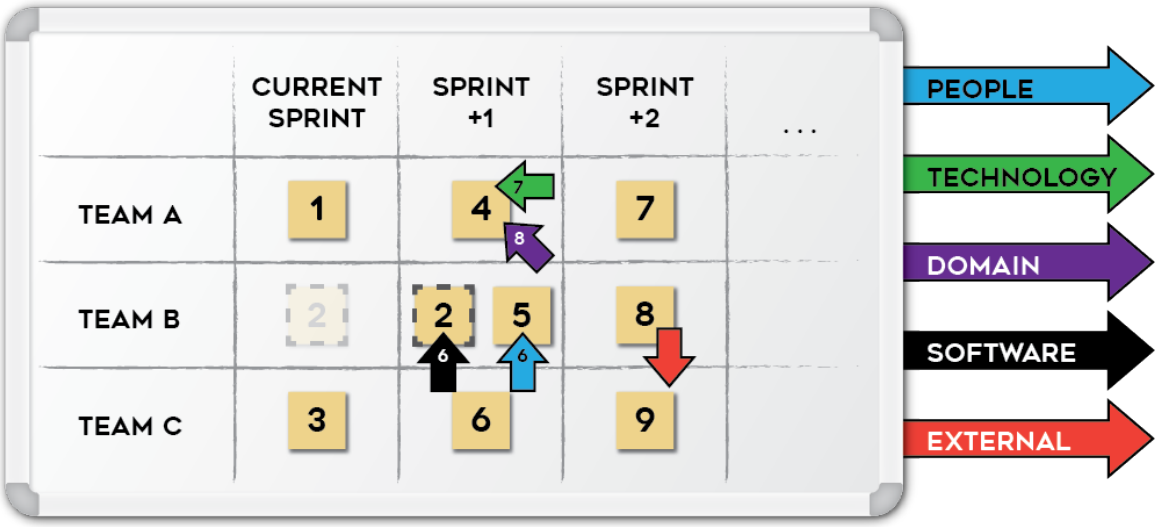


Figure 4.



Keep in mind that Cross-Team Refinement does not replace individual Scrum Team Product Backlog refinement. Once the teams have agreed on the likely sequence of their work over the next few Sprints, each team still should perform detailed Refinement inside their team.

Tracking Dependencies Over Time

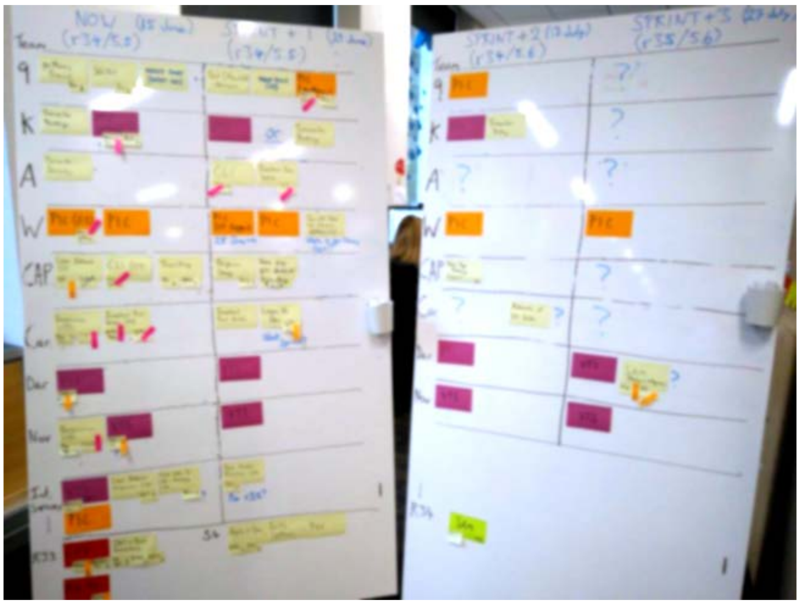
Some teams use the number and type of dependencies as an improvement measure. By noting the trend of dependencies over time, it can help track the impact of improvements to underlying architecture and working practices.

Some teams may also try to limit the number of dependencies they will accept, by limiting the number of arrows allowed. (e.g. We only have 10 dependency arrows to use when building the Cross-Team Refinement board. Once that limit is met, we cannot pull any more items with dependencies into the next 3 Sprints.)

Cross-Team Refinement and Nexus Sprint Planning

Cross-Team Refinement information should be taken into Nexus Sprint Planning for the teams to plan for the current Sprint and its dependencies. This information can also be used as a focal point for daily cross-team synchronization about risk and progress during the Nexus Daily Scrum.

Figure 5. Physical Cross-Team Refinement board for a Nexus.



Conclusion

The Cross-Team Refinement workshop is a technique and practice that supports scaling Scrum and Nexus. Nexus is a framework that consists of roles, events, and artifacts that bind together the work of multiple Scrum Teams working together on a single product.

Join a Scaled Professional Scrum training course to learn more about how to apply this and many other practices when scaling Scrum. Find a course at:
https://www.scrum.org/Courses/Scaled-Professional-Scrum