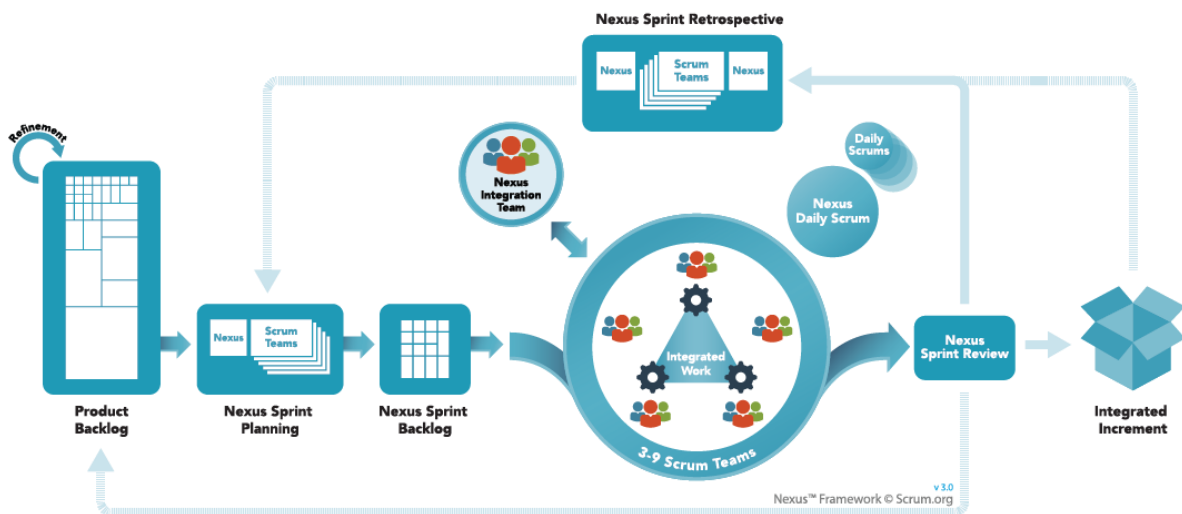


# Nexus™ Guide

## The Definitive Guide to scaling Scrum with Nexus: The Rules of the Game

January 2018



*Ken Schwaber*

Developed and sustained by Ken Schwaber and Scrum.org

# Table of Contents

---

- Nexus Overview ..... 2
  - Purpose of the Nexus Guide ..... 2
  - Definition of Nexus..... 2
  - Nexus Background..... 2
  - Nexus Framework ..... 3
  - Nexus Process Flow ..... 4
- Nexus ..... 5
  - Nexus Roles..... 5
    - Nexus Integration Team ..... 5
    - Product Owner in the Nexus Integration Team ..... 6
    - Scrum Master in the Nexus Integration Team ..... 6
    - Nexus Integration Team Members ..... 6
  - Nexus Events ..... 6
    - Refinement ..... 7
    - Nexus Sprint Planning ..... 7
    - Nexus Sprint Goal..... 8
    - Nexus Daily Scrum..... 8
    - Nexus Sprint Review..... 8
    - Nexus Sprint Retrospective ..... 8
- Nexus Artifacts ..... 9
  - Product Backlog ..... 10
  - Nexus Sprint Backlog..... 10
  - Integrated Increment..... 10
- Artifact Transparency ..... 10
  - Definition of “Done” ..... 10
- End Note ..... 11
- Acknowledgement ..... 11

# Nexus Overview

## Purpose of the Nexus Guide

Nexus is a framework for developing and sustaining scaled product and software delivery initiatives. It uses Scrum as its building block. This Guide contains the definition of Nexus. This definition consists of Nexus roles, events, artifacts, and the rules that bind them together. Ken Schwaber and Scrum.org developed Nexus. The Nexus Guide is written and provided by them.

## Definition of Nexus

Nexus (n): a relationship or connection between people or things

Nexus is a framework consisting of roles, events, artifacts, and rules that bind and weave together the work of approximately three to nine Scrum Teams working on a single Product Backlog to build an Integrated Increment that meets a goal.

## Nexus Background

Software delivery is complex, and the integration of that work into working software has many artifacts and activities that must be coordinated to create a “Done” outcome. The work must be organized and sequenced, the dependencies resolved, and the outcomes staged.

Many developers have used the Scrum framework to work collectively to develop and deliver an Increment of working software. However, if more than one Scrum Team is working off the same Product Backlog and in the same codebase for a product, difficulties often arise. If the developers are not on the same collocated team, how will they communicate when they are doing work that will affect each other? If they work on different teams, how will they integrate their work and test the Integrated Increment? These challenges appear when two Scrum Teams are integrating their work into a single increment, and become significantly more difficult when three or more Scrum Teams integrate their work into a single increment.

There are many dependencies that arise between the work of multiple teams that collaborate to create a complete and “Done” Increment at least once every Sprint. These dependencies are related to:

1. **Requirements:** The scope of the requirements may overlap, and the manner in which they are implemented may also affect each other. That knowledge should be considered when ordering the Product Backlog and selecting Product Backlog items.

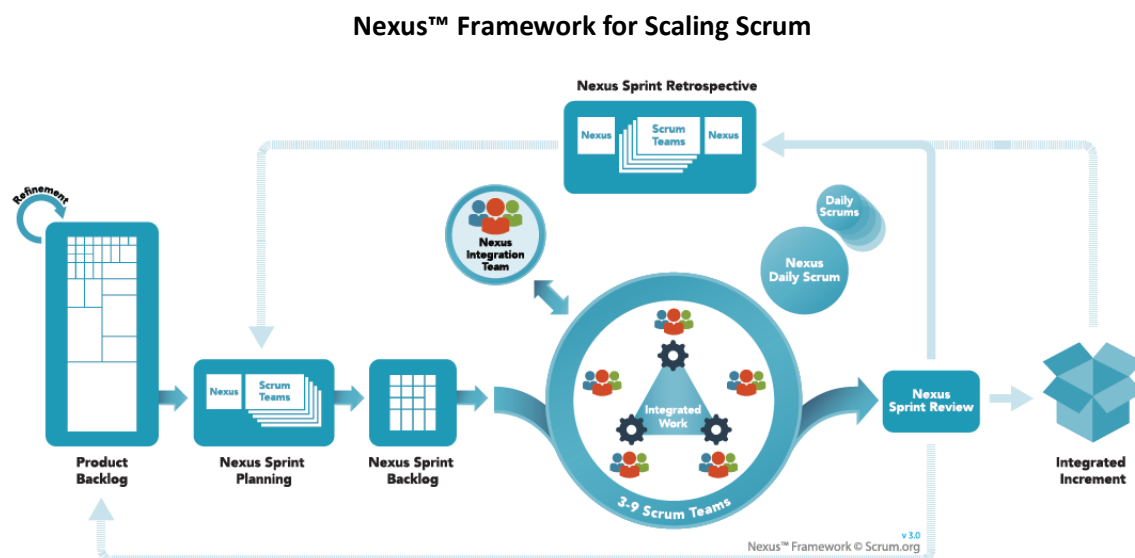
2. **Domain knowledge:** The people on the teams have knowledge of various business and computer systems. Their knowledge should be distributed across the Scrum Teams to ensure that the teams have the knowledge they need to do their work, to minimize interruptions between Scrum Teams during a Sprint.
3. **Software and test artifacts:** The requirements are, or will be, instantiated in software.

To the extent that requirements, team members’ knowledge, and software artifacts are mapped to the same Scrum Teams, teams can reduce the number of dependencies between them.

When software delivery using Scrum is scaled, these dependencies of requirements, domain knowledge, and software artifacts should drive the organization of the Development Teams. To the extent that it does, productivity will be optimized.

## Nexus Framework

Nexus is a process framework for multiple Scrum Teams working together to create an Integrated Increment. Nexus is consistent with Scrum and its parts will be familiar to those who have used Scrum. The difference is that more attention is paid to dependencies and interoperation between Scrum Teams, delivering at least one “Done” Integrated Increment every Sprint.



As displayed in the graphic, Nexus consists of:

- **Roles:** A new role, the Nexus Integration Team, exists to coordinate, coach, and supervise the application of Nexus and the operation of Scrum so the best outcomes are derived. The Nexus Integration Team consists of the Product Owner, a Scrum Master, and Nexus Integration Team Members.

©2018 Scrum.org. Offered for license under the Attribution Share Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Nexus Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

- **Artifacts:** All Scrum Teams use the same, single Product Backlog. As the Product Backlog items are refined and made ready, indicators of which team will do the work inside a Sprint are made transparent. A new artifact, the Nexus Sprint Backlog, exists to assist with transparency during the Sprint. All Scrum Teams maintain their individual Sprint Backlogs.
- **Events:** Events are appended to, placed around, or replace (in the case of the Sprint Review) regular Scrum events to augment them. As modified, they serve both the overall effort of all Scrum Teams in the Nexus, and each individual team.

## Nexus Process Flow

A Nexus consists of multiple cross-functional Scrum Teams working together to deliver a potentially releasable Integrated Increment at least by the end of each Sprint. Based on dependencies, the teams may self-organize and select the most appropriate members to do specific work.

- **Refine the Product Backlog:** The Product Backlog needs to be decomposed so that dependencies are identified and removed or minimized. Product Backlog items are refined into thinly sliced pieces of functionality and the team likely to do the work should be identified.
- **Nexus Sprint Planning:** Appropriate representatives from each Scrum Team meet to discuss and review the refined Product Backlog. They select Product Backlog items for each team. Each Scrum Team then plans its own Sprint, interacting with other teams as appropriate. The outcome is a set of Sprint Goals that align with the overarching Nexus Sprint Goal, each Scrum Team's Sprint Backlog and a single Nexus Sprint Backlog. The Nexus Sprint Backlog makes the work of all Scrum Team's selected Product Backlog items and any dependencies transparent.
- **Development work:** All teams frequently integrate their work into a common environment that can be tested to ensure that the integration is done.
- **Nexus Daily Scrum:** Appropriate representatives from each Development Team meet daily to identify if any integration issues exist. If identified, this information is transferred back to each Scrum Team's Daily Scrum. Scrum Teams then use their Daily Scrum to create a plan for the day, being sure to address the integration issues raised during the Nexus Daily Scrum.
- **Nexus Sprint Review:** The Nexus Sprint Review is held at the end of the Sprint to provide feedback on the Integrated Increment that a Nexus has built over the Sprint. All individual Scrum Teams meet with stakeholders to review the Integrated Increment. Adjustments may be made to the Product Backlog.

- **Nexus Sprint Retrospective:** Appropriate representatives from each Scrum Team meet to identify shared challenges. Then, each Scrum Team holds individual Sprint Retrospectives. Appropriate representatives from each team meet again to discuss any actions needed based on shared challenges to provide bottom-up intelligence.

## Nexus

Nexus roles, events, and artifacts inherit the purpose and intent attributes of the corresponding Scrum roles, events, and artifacts, as documented in the Scrum Guide ([www.scrumguides.org](http://www.scrumguides.org)).

### Nexus Roles

A Nexus consists of a Nexus Integration Team and approximately three to nine Scrum Teams.

#### Nexus Integration Team

The Nexus Integration Team is accountable for ensuring that a “Done” Integrated Increment (the combined work completed by a Nexus) is produced at least once every Sprint. The Scrum Teams are responsible for delivering “Done” Increments of potentially releasable products, as prescribed in Scrum. All roles for members of the Scrum Teams are prescribed in the Scrum Guide.

The Nexus Integration Team consists of:

- The Product Owner
- A Scrum Master
- One or more Nexus Integration Team Members

Members of the Nexus Integration Team are often also members of the individual Scrum Teams in that Nexus. If this is the case, they must give priority to their work on the Nexus Integration Team; membership in the Nexus Integration Team takes precedence over individual Scrum Team membership. This preference helps ensure that the work to resolve issues affecting many teams has priority.

Composition of the Nexus Integration Team may change over time to reflect the current needs of a Nexus. Common activities the Nexus Integration Team might perform include coaching, consulting, and highlighting awareness of dependencies and cross-team issues. It might also perform work from the Product Backlog.

The Scrum Teams address integration issues within the Nexus. The Nexus Integration Team provides a focal point of integration for the Nexus. Integration includes resolving any technical and non-technical

cross-team constraints that may impede a Nexus' ability to deliver a constantly Integrated Increment. They should use bottom-up intelligence from the Nexus to achieve resolution.

### **Product Owner in the Nexus Integration Team**

A Nexus works off a single Product Backlog, and as described in the Scrum framework, a Product Backlog has a single Product Owner who has the final say on its contents. The Product Owner is responsible for maximizing the value of the product and the work performed and integrated by the Scrum Teams in a Nexus. The Product Owner is a member of the Nexus Integration Team.

The Product Owner is accountable for managing the Product Backlog so that maximum value is derived from the Integrated Increment created by a Nexus. How this is done may vary widely across organizations, Nexuses, Scrum Teams, and individuals.

### **Scrum Master in the Nexus Integration Team**

The Scrum Master in the Nexus Integration Team has overall responsibility for ensuring the Nexus framework is understood and enacted. This Scrum Master may also be a Scrum Master in one or more of the Scrum Teams in that Nexus.

### **Nexus Integration Team Members**

The Nexus Integration Team consists of professionals who are skilled in the use of tools, various practices, and the general field of systems engineering. Nexus Integration Team Members ensure the Scrum Teams within the Nexus understand and implement the practices and tools needed to detect dependencies, and frequently integrate all artifacts to the definition of "Done." Nexus Integration Team Members are responsible for coaching and guiding the Scrum Teams in a Nexus to acquire, implement, and learn these practices and tools.

Additionally, the Nexus Integration Team coaches the individual Scrum Teams on the necessary development, infrastructural, or architectural standards required by the organization to ensure the development of quality Integrated Increments.

If their primary responsibility is satisfied, Nexus Integration Team Members may also work as Development Team members in one or more Scrum Teams.

## **Nexus Events**

The duration of Nexus events is guided by the length of the corresponding events in the Scrum Guide. They are time-boxes in addition to their corresponding Scrum events.

## Refinement

Refinement of the Product Backlog at scale serves a dual purpose. It helps the Scrum Teams forecast which team will deliver which Product Backlog items, and it identifies dependencies across those teams. This transparency allows the teams to monitor and minimize dependencies.

Refinement of Product Backlog Items by the Nexus continues until the Product Backlog Items are sufficiently independent to be worked on by a single Scrum Team without excessive conflict.

The number, frequency, duration and attendance of Refinement is based on the dependencies and uncertainty inherent in the Product Backlog. Product Backlog items pass through different levels of decomposition from very large and vague requests to actionable work that a single Scrum Team could deliver inside a Sprint.

Refinement is continuous throughout the Sprint as necessary and appropriate. Product Backlog refinement will continue within each Scrum Team in order for the Product Backlog items to be ready for selection in a Nexus Sprint Planning event.

## Nexus Sprint Planning

The purpose of Nexus Sprint Planning is to coordinate the activities of all Scrum Teams in a Nexus for a single Sprint. The Product Owner provides domain knowledge and guides selection and priority decisions. The Product Backlog should be adequately refined with dependencies identified and removed or minimized prior to Nexus Sprint Planning.

During Nexus Sprint Planning, appropriate representatives from each Scrum Team validate and make adjustments to the ordering of the work as created during Refinement events. All members of the Scrum Teams should participate to minimize communication issues.

The Product Owner discusses the Nexus Sprint Goal during Nexus Sprint Planning. The Nexus Sprint Goal describes the purpose that will be achieved by the Scrum Teams during the Sprint. Once the overall work for the Nexus is understood, Nexus Sprint Planning continues with each Scrum Team performing their own separate Sprint Planning. The Scrum Teams should continue to share newly found dependencies with other Scrum Teams in the Nexus. Nexus Sprint Planning is complete when each Scrum Team has finished their individual Sprint Planning events.

New dependencies may emerge during Nexus Sprint Planning. They should be made transparent and minimized. The sequence of work across teams may also be adjusted. An adequately refined Product Backlog will minimize the emergence of new dependencies during Nexus Sprint Planning. All Product Backlog items selected for the Sprint and their dependencies should be made transparent on the Nexus Sprint Backlog.



## Nexus Sprint Goal

The Nexus Sprint Goal is an objective set for the Sprint. It is the sum of all the work and Sprint Goals of the Scrum Teams within the Nexus. The Nexus should demonstrate the functionality that it has “Done” developed to achieve the Nexus Sprint Goal at the Nexus Sprint Review in order to receive stakeholder feedback.

## Nexus Daily Scrum

The Nexus Daily Scrum is an event for appropriate representatives from individual Development Teams to inspect the current state of the Integrated Increment and to identify integration issues or newly discovered cross-team dependencies or cross-team impacts.

During the Nexus Daily Scrum, attendees should focus on each team’s impact on the Integrated Increment and discuss:

- Was the previous day’s work successfully integrated? If not, why not?
- What new dependencies or impacts have been identified?
- What information needs to be shared across teams in the Nexus?

The Development Teams uses the Nexus Daily Scrum to inspect progress toward the Nexus Sprint Goal. At least every Nexus Daily Scrum, the Nexus Sprint Backlog should be adjusted to reflect the current understanding of the work of the Scrum Teams within the Nexus.

The individual Scrum Teams then take back issues and work that were identified during the Nexus Daily Scrum to their individual Scrum Teams for planning inside their individual Daily Scrum events.

## Nexus Sprint Review

The Nexus Sprint Review is held at the end of the Sprint to provide feedback on the Integrated Increment that the Nexus has built over the Sprint and to adapt the Product Backlog if needed.

A Nexus Sprint Review replaces individual Scrum Team Sprint Reviews, because the entire Integrated Increment is the focus for capturing feedback from stakeholders. It may not be possible to show all completed work in detail. Techniques may be necessary to maximize stakeholder feedback. The result of the Nexus Sprint Review is a revised Product Backlog.

## Nexus Sprint Retrospective

The Nexus Sprint Retrospective is a formal opportunity for a Nexus to inspect and adapt itself and create a plan for improvements to be enacted during the next Sprint to ensure continuous improvement. The

Nexus Sprint Retrospective occurs after the Nexus Sprint Review and prior to the next Nexus Sprint Planning.

It consists of three parts:

1. The first part is an opportunity for appropriate representatives from across a Nexus to meet and identify issues that have impacted more than a single team. The purpose is to make shared issues transparent to all Scrum Teams.
2. The second part consists of each Scrum Team holding their own Sprint Retrospective as described in the Scrum framework. They can use issues raised from the first part of the Nexus Retrospective as input to their team discussions. The individual Scrum Teams should form actions to address these issues during their individual Scrum Team Sprint Retrospectives.
3. The final, third part is an opportunity for appropriate representatives from the Scrum Teams to meet again and agree on how to visualize and track the identified actions. This allows the Nexus as a whole to adapt.

Because they are common scaling dysfunctions, every Retrospective should address the following subjects:

- Was any work left undone? Did the Nexus generate technical debt?
- Were all artifacts, particularly code, frequently (as often as every day) successfully integrated?
- Was the software successfully built, tested, and deployed often enough to prevent the overwhelming accumulation of unresolved dependencies?

For the questions above, address if necessary:

- Why did this happen?
- How can technical debt be undone?
- How can the recurrence be prevented?

## Nexus Artifacts

Artifacts represent work or value to provide transparency and opportunities for inspection and adaptation, as described in the Scrum Guide.

©2018 Scrum.org. Offered for license under the Attribution Share Alike license of Creative Commons, accessible at <http://creativecommons.org/licenses/by-sa/4.0/legalcode> and also described in summary form at <http://creativecommons.org/licenses/by-sa/4.0/>. By utilizing this Nexus Guide, you acknowledge and agree that you have read and agree to be bound by the terms of the Attribution Share-Alike license of Creative Commons.

## Product Backlog

There is a single Product Backlog for the entire Nexus and all of its Scrum Teams. The Product Owner is accountable for the Product Backlog, including its content, availability, and ordering.

At scale, the Product Backlog must be understood at a level where dependencies can be detected and minimized. To support resolution, Product Backlog items are often resolved to a granularity called “thinly sliced” functionality. Product Backlog items are deemed “Ready” for the Nexus Sprint Planning meeting when the Scrum Teams can select items to be done with no or minimal dependencies with other Scrum Teams.

## Nexus Sprint Backlog

A Nexus Sprint Backlog is the composite of Product Backlog items from the Sprint Backlogs of the individual Scrum Teams. It is used to highlight dependencies and the flow of work during the Sprint. It is updated at least daily, often as part of the Nexus Daily Scrum.

## Integrated Increment

The Integrated Increment represents the current sum of all integrated work completed by a Nexus. The Integrated Increment must be usable and potentially releasable which means it must meet the definition of “Done”. The Integrated Increment is inspected at the Nexus Sprint Review.

## Artifact Transparency

Just like its building block, Scrum, Nexus is based on transparency. The Nexus Integration Team works with the Scrum Teams within a Nexus and the organization to ensure that transparency is apparent across all artifacts and that the integrated state of the Integrated Increment is widely understood.

Decisions made based on the state of Nexus artifacts are only as effective as the level of artifact transparency. Incomplete or partial information will lead to incorrect or flawed decisions. The impact of those decisions can be magnified at the scale of Nexus. Software must be developed so that dependencies are detected and resolved before technical debt becomes unacceptable to the Nexus. A lack of complete transparency will make it impossible to guide a Nexus effectively to minimize risk and maximize value.

## Definition of “Done”

The Nexus Integration Team is responsible for a definition of “Done” that can be applied to the Integrated Increment developed each Sprint. All Scrum Teams of a Nexus adhere to this definition of “Done”. The Increment is “Done” only when integrated, usable and potentially releasable by the Product Owner.

Individual Scrum Teams may choose to apply a more stringent definition of “Done” within their own teams, but cannot apply less rigorous criteria than agreed for the Increment.

## End Note

Nexus is free and offered in this Guide. As with the Scrum framework, the Nexus roles, artifacts, events, and rules are immutable. Although implementing only parts of Nexus is possible, the result is not Nexus.

## Acknowledgement

Nexus and Scaled Professional Scrum were collaboratively developed by Ken Schwaber, David Dame, Richard Hundhausen, Patricia Kong, Rob Maher, Steve Porter, Christina Schwaber, and Gunther Verheyen.