

## Inhaltsverzeichnis

1. [Herkunft Nexus](#)
2. [Aufbau des Nexus Framework](#)
3. [Das Nexus Integration Team \(NIT\)](#)
  - a. [Besetzung des NIT](#)
  - b. [Scrum Master des NIT](#)
  - c. [Arbeit im NIT sticht](#)
4. [Nexus Events](#)
  - a. [Nexus Sprint Planning](#)
  - b. [Nexus Daily Scrum](#)
  - c. [Nexus Sprint Review](#)
  - d. [Nexus Sprint Retrospective](#)
  - e. [Cross-Team Refinement](#)
5. [Nexus Artefakte](#)
  - a. [Nexus Sprint Backlog](#)
  - b. [Definition of Done](#)
  - c. [Integriertes Inkrement](#)
6. [Nexus+ - Nexus für mehr als 9 Teams](#)
7. [Fazit](#)

## Nexus - Ein einfacher Einstieg in die Skalierung von Scrum

Nexus ist ein agiles Vorgehensmodell für die Skalierung von Scrum. Dabei adressiert das Nexus Framework Organisationen, die bereits gute Erfahrungen mit Scrum gesammelt haben und ihr Vorgehen nun auf bis zu neun Teams ausweiten möchten.

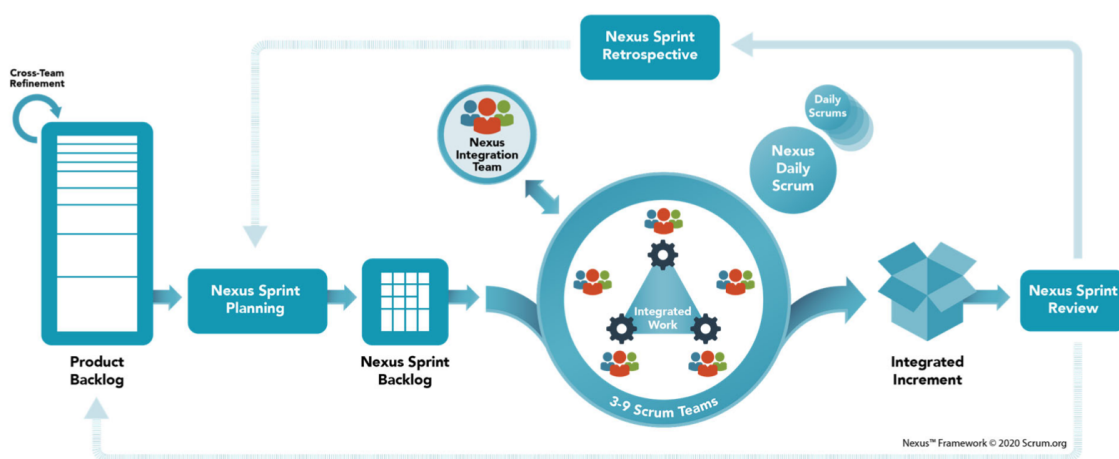
In diesem Artikel erkläre ich Dir den Aufbau und die grundlegenden Konzepte des Nexus Frameworks auf Basis des Nexus Guides 2021. Dabei gehe ich davon aus, dass Du mit den [Grundlagen von Scrum](#) vertraut bist.

### Herkunft Nexus

Nexus wurde vom Scrum-Gründer Ken Schwaber entworfen und 2015 auf [scrum.org](https://www.scrum.org) veröffentlicht. Das Framework ist im Nexus Guide beschrieben, die letzte Aktualisierung erfolgte im Januar 2021. Dabei ist Nexus eine direkte Alternative zu Scrum@Scale, das wiederum von Jeff Sutherland, dem zweiten Scrum-Gründer entwickelt wurde. Wie auch Scrum@Scale behält das Nexus Framework grundlegende Scrum Konzepte bei, ist dabei deutlich leichter jedoch nur begrenzt skalierbar. Außerdem steht Nexus in direkter Konkurrenz zu anderen skalierten Frameworks wie [LeSS](#) oder SaFE.

## Aufbau des Nexus Framework

Ein **Nexus** (lat. Verknüpfung) ist eine Gruppe von 3 bis 9 Scrum-Teams, die gemeinsam an der Entwicklung eines Produkts arbeiten. Dabei arbeiten alle Teams des Nexus nach Scrum in einem gemeinsamen Sprint-Rhythmus an der Erstellung eines integrierten Inkrements. Zudem hat das Nexus nur einen Product Owner, der für alle Teams des Nexus zuständig ist. Damit die Zusammenarbeit der Teams und vor allem die Zusammenführung der Arbeitsergebnisse zu einem gemeinsamen Inkrement einwandfrei funktioniert, etabliert das Nexus Framework eine neue Verantwortlichkeit, das Nexus Integration Team (NIT). Zudem werden die Scrum Events auch in einer skalierten Nexus Version abgehalten und die bekannten Scrum Artefakte werden für die Arbeit im Nexus leicht angepasst.



Das Nexus Framework in der Übersicht - Quelle: [scrum.org](https://www.scrum.org)

## Die magische Zahl neun

Die Limitierung von neun Teams pro Nexus ist das Ergebnis von empirischer Beobachtung und der Erkenntnis, dass Zusammenarbeit bei mehr als neun Teams zunehmend ineffizient wird. Diese Limitierung lehnt sich an die [Millersche Zahl](#) an, die besagt, dass Du nur  $7 \pm 2$  Informationseinheiten parallel im Kurzzeitgedächtnis halten kannst. Deswegen haben Scrum-Teams auch maximal  $6 \pm 3$  Entwickler, obwohl es mittlerweile eine klare Tendenz zu einer optimalen Teamgröße von 5 Entwicklern pro Scrum-Team gibt. Die magische Zahl neun findest Du übrigens auch in Nexus+, der Erweiterung des Frameworks mit wiederum 9 Nexus. Dazu zum Abschluss des Beitrags mehr.

## Das Nexus Integration Team (NIT)

Das Nexus Integration Team (NIT) befähigt die Teams durch einheitliche Prozesse, geeignete Werkzeuge und Vorgehensmodelle ihre Arbeitsergebnisse mit denen der anderen Teams zu integrieren. Während also die Scrum-Teams für die tatsächliche Integration zuständig sind, kümmert sich das NIT um technische und nichttechnische Rahmenbedingungen für eine reibungslose Integration. Damit ist das NIT [Servant-Leader](#) des Nexus und befähigt die Teams ein integriertes Inkrement zu liefern. Darüber hinaus ist das NIT der Hüter der [Definition of Done](#).

Beispielhafte Aufgaben des NIT im Rahmen der Softwareentwicklung:

- Einführung, Schulung und der reibungslose Betrieb eines Versionierungs- und Releasetools.
- Formulierung der Definition of Done.
- Sicherstellen einer effizienten, gemeinsamen Backlog Umgebung.
- Coaching in Engineering-Fragen (z.B.: Peer Programming, Test Driven Development).
- Hilfestellung bei der Koordination der Teams
- Abhängigkeiten im Sprint sichtbar machen

## Besetzung des Nexus Integration Teams

Das NIT besteht aus dem Product Owner, dem NIT Scrum Master und weiteren Mitgliedern des Nexus, die über kritische Kompetenzen für die erfolgreiche Arbeit des NIT verfügen. Während der PO und der Scrum Master dauerhafte Mitglieder des NIT sind, können Teammitglieder im Laufe des Entwicklungszyklus variieren. Dann nämlich wenn neues Wissen und weitere Fähigkeiten benötigt werden, um die Arbeit des NIT und der angeschlossenen Teams bei ihrer Integration wirksam zu unterstützen.

## Scrum Master des NIT

Der Scrum Master des NIT ist für die reibungslose Umsetzung des Nexus verantwortlich und wird typischerweise aus den Scrum-Teams des Nexus rekrutiert. Während sich ein Scrum Master auf Teamebene also um die Zusammenarbeit und den Scrum-Prozess des Teams kümmert, ist der Scrum Master im NIT auf teamübergreifende Angelegenheiten fokussiert. Damit ist der Scrum Master des NIT in letzter Konsequenz verantwortlich dafür, dass das Nexus Integration Team (NIT) wirksam ist und dass das Nexus Framework konsequent umgesetzt wird.

## Arbeit im NIT sticht

Sofern Teammitglieder und Scrum Master eine Doppelrolle haben, gilt die sehr klare Regel, dass die Arbeit im NIT immer Vorrang vor den Verantwortlichkeiten auf Teamebene hat. Das heißt, die Arbeit im NIT trumps im Zweifel die Mitarbeit im Team. Auch wenn das die Teamarbeit potentiell einschränkt, haben Doppelbesetzungen den

Vorteil, dass das Nexus Integration Team (NIT) die Probleme der Teams aus erster Hand erfährt. Zudem bleibt das Nexus Framework an dieser Stelle sehr schlank und Du musst für die Gründung des Nexus keine zusätzlichen Positionen schaffen.



Das Nexus Integration Team (NIT) - Quelle: Eigene Grafik

## Nexus Events

Damit die Synchronisation und die Zusammenarbeit der Teams innerhalb Nexus gelingt, werden die Scrum Events in einer skalierten Nexus Version abgehalten. Dabei halten Teams nach wie vor ihre eigenen Scrum Events, jedoch gibt es fast immer eine sehr enge Verzahnung mit den Nexus Events.

## Nexus Sprint Planning

Im Nexus Sprint Planning werden die Product Backlog Items und Ziele für den kommenden Sprint festgelegt. Dazu treffen sich im **ersten Schritt** der Product Owner und Repräsentanten der Teams, um die priorisierten Backlog Items zu besprechen und auf die Teams aufzuteilen. Im **zweiten Schritt** verfeinern die Teams in einem teaminternen Planning ihre zugeteilten Aufgaben. Sofern sie dabei Überschneidungen oder Abhängigkeiten zu anderen Teams und deren Aufgaben feststellen, stimmen sich die Teams untereinander ab. Erst wenn die Aufgaben auf Teamebene besprochen, die Teams sich auf ihre zugeteilten Aufgaben verpflichtet haben, alle offenen Fragen und potentiellen Abhängigkeiten geklärt sind, ist das Nexus Sprint Planning formal beendet.

Das Nexus Sprint Planning nimmt, wie auch ein reguläres Scrum Planning, einen Tag bei einem vier Wochen Sprint in Anspruch.

## Nexus Daily Scrum

Für das Nexus Daily Scrum treffen sich im **ersten Schritt** die Repräsentanten der Teams um ihren Status, Probleme bei der Integration ihrer Arbeitsergebnisse und teamübergreifende Abstimmungsfragen zu besprechen. Bei Bedarf nehmen der Product Owner und der Scrum Master des NIT ebenfalls teil. Im **zweiten Schritt** findet das Daily Scrum der einzelnen Teams statt. Dabei werden teaminterne Angelegenheiten besprochen und vor allem die Punkte aus dem Nexus Daily, die das jeweilige Team betreffen. Für das Daily solltest Du eine Timebox von 30 Minuten setzen, jeweils 15 Minuten für das Nexus Daily und 15 Minuten für das Daily der Teams.

## Nexus Sprint Review

In der Nexus Sprint Review wird das integrierte Inkrement Stakeholdern und Kunden präsentiert, um deren Feedback einzuholen. Dabei ist die Nexus Sprint Review das einzige Review Ereignis. Das heißt, es findet kein analoges Event mehr auf Teamebene statt. Schließlich geht es im Nexus um die Erschaffung eines gemeinsamen Inkrements, deswegen sind Reviews auf Teamebene unnötig.

## Nexus Sprint Retrospective

Ziel der Nexus Retrospective ist die Verbesserung des Arbeitsprozesses auf Ebene des gesamten Nexus. Dazu treffen sich im **ersten Schritt** das Nexus-Integration-Team (NIT) und Repräsentanten der Teams, um übergreifende Themen und Probleme zu identifizieren. Zu diesem Zusammentreffen sind grundsätzlich alle Mitglieder des Nexus herzlich eingeladen. Im **zweiten Schritt** gehen die Teams in ihre teaminterne Retrospektive. Zum einen, um eigene Verbesserungspotentiale, zum anderen, um Lösungen für die teamübergreifenden Probleme zu identifizieren. Im **dritten Schritt** der Nexus Sprint Retrospective kommen die Repräsentanten aus Schritt eins noch einmal zusammen, um die identifizierten Lösungen zusammenzutragen und eine gemeinsame Vorgehensweise festzulegen. Damit ist die Nexus Sprint Retrospective abgeschlossen.

## Cross-Team Refinement

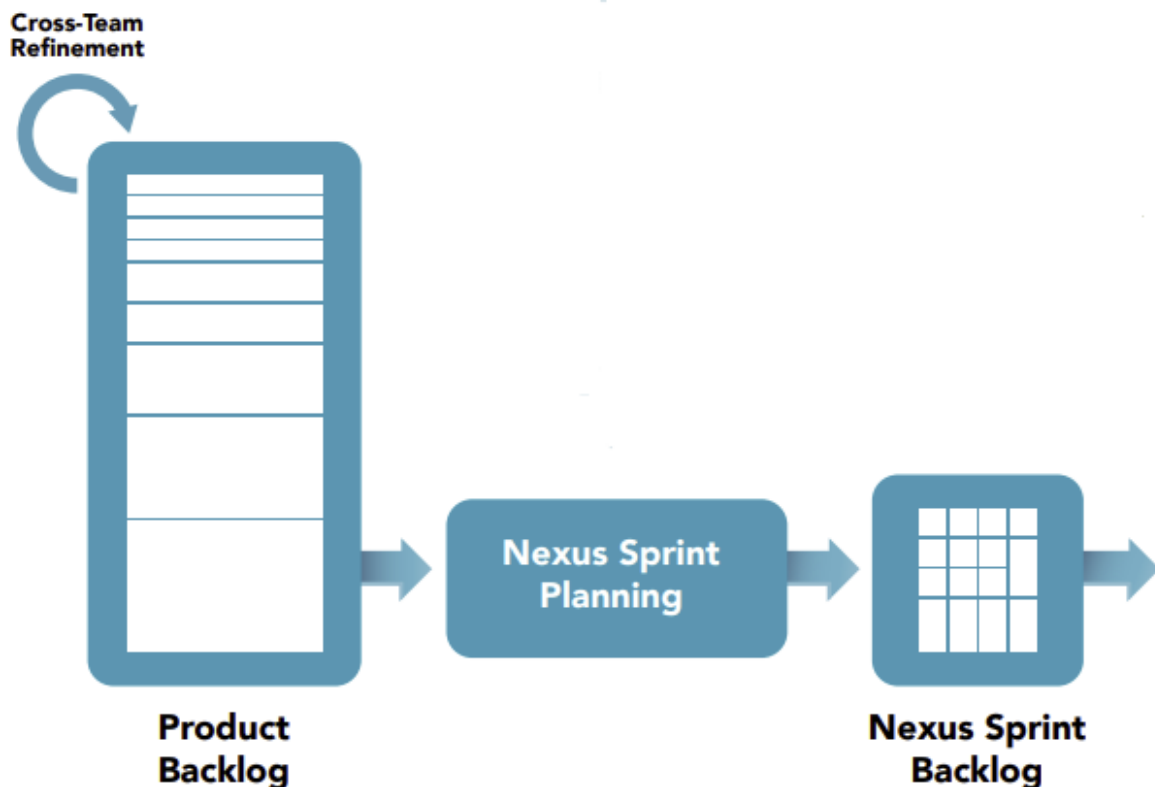
Das Cross-Team Refinement ist kein fest terminiertes Ereignis, sondern wie in Scrum auch eine andauernde Tätigkeit. Dazu treffen sich im **ersten Schritt** der PO und ausgewählte Teams des Nexus, um anstehende Aufgaben so zu zerlegen, dass sie im Idealfall in einem Sprint von einem Team autark erledigt werden können. Im **zweiten Schritt** verfeinern Teams in einem teaminternen Refinement ihre Aufgaben weiter. Für das Refinement sollten in Summe analog zu Scrum nicht mehr als 15% der Arbeitszeit aufgewendet werden.

## Nexus Artefakte

Für die Arbeit mit dem Nexus Framework gelten die bekannten Scrum Artefakte. Dabei werden das Backlog, die Definition of Done und das Inkrement erweitert bzw. präzisiert.

### Nexus Sprint Backlog

Das Nexus Sprint Backlog enthält alle priorisierten und übernommenen Aufgaben für den aktuellen Sprint des gesamten Nexus. Das heißt, Teams pflegen kein entkoppeltes Sprint Backlog, vielmehr ist das Sprint Backlog eines Teams nur ein Ausschnitt aus dem Nexus Sprint Backlog. Durch dieses gemeinsame Nexus Sprint Backlog sind der Arbeitsfortschritt, Bottlenecks und bestehende Abhängigkeiten über alle Teams jederzeit transparent. So hast Du jederzeit eine Rückmeldung über den Workflow im gesamten Nexus und kannst ihn bei Bedarf optimieren.



Das Nexus Sprint Backlog ist ein direktes Ergebnis des Nexus Sprint Planning. - Quelle: [scrum.org](https://www.scrum.org)

### Definition of Done

Die Definition of Done (DoD) definiert die Anforderungen an das integrierte Inkrement zum Ende des Sprints für das gesamte Nexus. Für die Ausformulierung und Entwicklung dieser gemeinsamen DoD ist das Nexus Integration Team (NIT)

verantwortlich. Dabei sind die Teams herzlich eingeladen ihren Beitrag zur Erstellung und Entwicklung der Nexus DoD beizutragen. In letzter Konsequenz ist jedoch das NIT dafür verantwortlich, dass die DoD des Nexus von allen Teams berücksichtigt wird. Dabei gilt die DoD des Nexus sozusagen als ein Mindeststandard, Teams können durchaus ihre eigene höherwertige Definition of Done pflegen.

## **Integriertes Inkrement**

Das integrierte Inkrement ist das gemeinsame Arbeitsergebnis aller Teams des Nexus zum Ende des Sprints. Dabei gibt es zwei grundsätzliche Anforderungen an dieses Nexus Artefakt. Erstens erfüllt das integrierte Inkrement die Definition of Done, zweitens sind die Arbeitsergebnisse aller Teams bereits zusammengeführt bzw. integriert. Das heißt, Teillösungen oder nicht vollständig integrierte Arbeitsergebnisse zählen nicht als integriertes Inkrement. Mit dieser strengen Definition sorgt das Nexus dafür, dass alle möglichen Integrationsprobleme bereits im Sprint gelöst wurden und daraus resultierende Arbeit nicht in den nächsten Sprint genommen wird und damit die Handlungsfähigkeit des Nexus einschränkt.

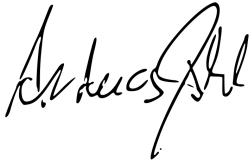
## **Nexus+ - Nexus für mehr als 9 Teams**

Nexus plus (Nexus+) ist die konzeptionelle Erweiterung des Nexus Frameworks auf bis zu neun Nexus, also maximal 81 Scrum Teams. Dabei funktioniert jedes weitere Nexus nach den hier vorgestellten grundlegenden Spielregeln und Konzepten. Wie genau die Zusammenarbeit der jeweiligen Nexus gelingen kann, gibt das Nexus Framework keine weitergehenden Anregungen und Konzepte. Tatsächlich scheint es bisher auch wenige praktische Anwendungen von Nexus in dieser Organisationsgröße zu geben. Sofern deine Organisation also mehr als 10 Scrum Teams hat, ist es ratsamer, wenn Du dir [LeSS](#) oder Scrum@Scale als alternative Konzepte zu Nexus anschaust.

## **Fazit - Ein einfacher erster Schritt für die Skalierung von Scrum**

Nexus ist ein interessantes Framework für Organisationen, die ihre erfolgreiche Arbeit mit Scrum mit weniger als zehn Teams auf das nächste Level heben wollen. Denn genau wie Scrum definiert Nexus nur ein minimales Set von festen Regeln. Dabei sind die Grundkonzepte klar und konsequent durchdacht, so dass Du als erfahrener Scrum-Praktiker keinerlei Probleme haben wirst deine Organisation erfolgreich auf die nächste Stufe zu heben. Für größere Organisationen mit mehr als neun Teams fehlt es jedoch an Vorgaben und Anregungen durch das Nexus Framework. Für diese größeren Implementierungen sind Scrum@Scale oder LeSS deutlich erfolgversprechende Startpunkte für die agile Skalierung. Aber ohnehin solltest Du Nexus und auch die anderen Frameworks nur als einen Startpunkt sehen, um in einem konsequenten agilen Prozess selbst zu lernen, was für deine Organisation ein zielführendes Setup ist.

Viel Erfolg dabei,



**Artikel zum Download:**

- Nexus Guide zum [Download](#)
- Cheatsheet - Nexus vs andere skalierte Frameworks

**Weiterführende Artikel:**

- [Agile Methoden - Teams und Organisation nach agilen Prinzipien strukturieren](#)
- [Was ist Scrum? - Eine kompakte Einführung in die Scrum Methode](#)
- [Design Sprint - 5 Tage, 1 Team, 1 Challenge - Design Thinking powered by Google Ventures](#)
- [Scrum einführen - Wie Du deine ersten Projekte mit Scrum umsetzt](#)
- [Product Owner \(PO\) - Verantwortung, Aufgaben und Kompetenzen](#)

**Mit Andreas arbeiten:**

- [Deine Organisation agil skalieren](#)