

2 Einführung in Nexus

In diesem Kapitel beschreiben wir das Nexus-Framework in seiner Gesamtheit. Wie Sie sehen werden, handelt es sich bei Nexus um eine relativ kleine und einfache Erweiterung von Scrum. Wie gesagt: »Skaliertes Scrum ist immer noch Scrum.« Scrum selbst ist recht einfach, zumindest einfach zu verstehen. Bei der Skalierung ist diese Einfachheit ein großer Vorteil, denn Komplexität ist der Gegenspieler zur Skalierung. Die Einfachheit von Nexus macht das Framework auch sehr anpassungsfähig, wie wir in den folgenden Kapiteln sehen werden.

2.1 Was ist Nexus?

Nexus ist ein Framework, das es mehreren Scrum-Teams ermöglicht, ausgehend von einem einzigen Product Backlog, gemeinsam zu arbeiten und mindestens ein »fertiges« integriertes Inkrement pro Sprint zu liefern. »Mehrere« bedeutet typischerweise drei bis neun Scrum-Teams. Warum nicht zwei? Zwei Teams können sich in der Regel ohne zusätzliche Struktur untereinander abstimmen. Warum neun? So wie Scrum empfiehlt, Teams auf maximal neun Mitglieder zu beschränken, um den Zusammenhalt zu verbessern und die Komplexität zu reduzieren, empfiehlt Nexus dasselbe für die Anzahl der Teams. Genau wie bei Scrum ist diese Obergrenze jedoch nicht absolut und es kann, abhängig von den Rahmenbedingungen, auch noch mit einer größeren Anzahl funktionieren. Bei der Arbeit mit Nexus haben wir festgestellt, dass die Komplexität der Zusammenarbeit und die Koordination zwischen den Teams deutlich ansteigen, wenn die Zahl der Teams über neun hinausgeht. In solchen Fällen kommen verschiedene andere Techniken zum Einsatz.¹

1. George Millers oft zitierte Arbeit »The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information« beschreibt Einschränkungen in der Art und Weise, wie wir Informationen verarbeiten und Erinnerungen bilden, die die Ad-hoc-Erfahrung bei der Organisation von Teams verstärken: Wenn Teams über neun Personen hinauswachsen, verlieren sie den Zusammenhalt und ihre Arbeit wird schwieriger zu managen. Weitere Informationen zu Millers Artikel finden Sie unter https://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two.

Da Nexus auf Scrum aufbaut, werden seine Bestandteile denjenigen vertraut sein, die Scrum bereits verwendet haben. Der Unterschied besteht darin, dass den Abhängigkeiten und der Kommunikation zwischen den Scrum-Teams mehr Aufmerksamkeit geschenkt wird (siehe Abb. 2–1).

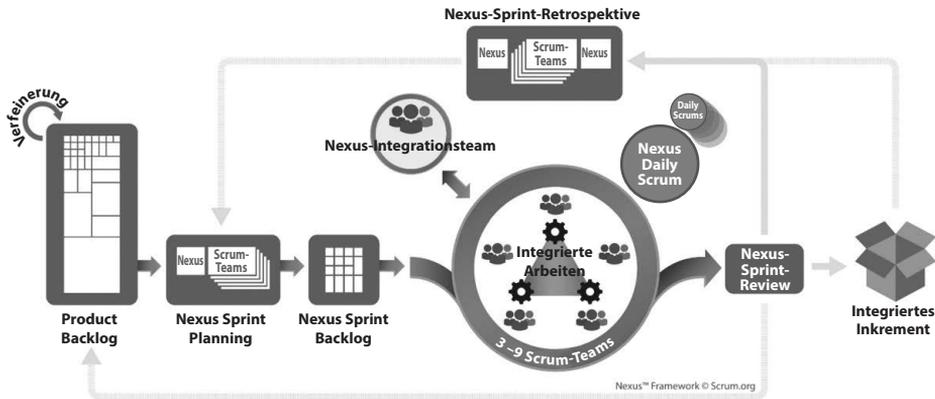


Abb. 2–1 Das Nexus-Framework zur Skalierung von Scrum (Nexus™ Framework © Scrum.org)

2.2 Nexus erweitert Scrum

Nexus ist Scrum mit einigen kleinen Ergänzungen (siehe Tab. 2–1).

- **Ein weiteres Artefakt kommt hinzu: das Nexus Sprint Backlog.**
Das Nexus Sprint Backlog ist der Plan des Nexus für den Sprint. Es hilft allen an der Produktentwicklung Beteiligten, zu verstehen, woran die jeweiligen Scrum-Teams arbeiten, und macht alle Abhängigkeiten transparent, die zwischen den Teams während des Sprints bestehen können.
- **Fünf zusätzliche Ereignisse kommen hinzu: Refinement, Nexus Sprint Planning, Nexus Daily Scrum, Nexus-Sprint-Review und die Nexus-Sprint-Retrospektive.**
Diese zusätzlichen Ereignisse erweitern Scrum, um sicherzustellen, dass die Arbeit möglichst effektiv zwischen den Scrum-Teams aufgeteilt und koordiniert wird. Zudem soll dadurch der Erfahrungsaustausch zwischen den Teams gefördert werden.
- **Auf die einzelnen Scrum-Team-Sprint-Reviews wird zugunsten des Nexus-Sprint-Reviews verzichtet.**
Da die Scrum-Teams in Nexus zusammenarbeiten, um ein einziges integriertes Inkrement zu erzeugen, sollte dieses integrierte Inkrement als Ganzes betrachtet werden.

■ **Es kommt eine neue Rolle hinzu: das Nexus-Integrationsteam.**

Das Nexus-Integrationsteam (NIT) wurde geschaffen, um eine transparente Verantwortlichkeit für die Integration in Nexus zu fördern und zu gewährleisten. Es coacht und begleitet die Anwendung von Nexus sowohl innerhalb der Scrum-Teams als auch innerhalb der Organisation. Das NIT besteht aus dem Product Owner des Produkts, einem Scrum Master und NIT-Mitgliedern, die normalerweise Mitglieder der Scrum-Teams im Nexus sind. Personen aus anderen Funktionsbereichen der Organisation wie Betrieb, Sicherheit, Architektur oder anderen Fachbereichen können ebenfalls Mitglieder sein, wenn sie dabei helfen können, ein integriertes Inkrement zu liefern. Diese »außenstehenden« Mitglieder können temporäre Mitglieder sein, die dem NIT nur so lange angehören, wie sie gebraucht werden.

Rollen	Ereignisse	Artefakte
Entwicklungsteams	Sprint	Product Backlog
Product Owner	<i>Nexus Sprint Planning *</i>	<i>Nexus Sprint Backlog *</i>
Scrum Master	Sprint Planning	Sprint Backlog
<i>Nexus-Integrationsteam *</i>	<i>Nexus Daily Scrum *</i>	Integriertes Inkrement
	Daily Scrum	
	<i>Nexus-Sprint-Review *</i>	
	<i>Nexus-Sprint-Retrospektive *</i>	
	Sprint Retrospektive	
	<i>Refinement *</i>	
* Nexus-spezifisch		

Tab. 2–1 Rollen, Ereignisse und Artefakte in Nexus (Scrum.org)

2.3 Das Nexus-Integrationsteam

Das NIT stellt sicher, dass der Nexus in jedem Sprint ein integriertes Inkrement erzeugt. Die dafür notwendigen Arbeiten werden von den Scrum-Teams erledigt. Letztendlich ist das NIT dafür verantwortlich, den Wert des integrierten Produkts zu maximieren (siehe Abb. 2–2). Zu seinen Aktivitäten kann die Entwicklung von Werkzeugen und Praktiken gehören, die bei der Integration helfen, ebenso, wie die Unterstützung bei der Kooperation als Coaches oder Berater.

NIT-Mitglieder müssen die Grundeinstellung eines Lehrers haben, um Scrum-Teams bei der Lösung ihrer Probleme zu unterstützen, wann immer dies möglich ist. Ihre Aufgabe ist es, Probleme aufzuzeigen, die gelöst werden müssen, und den Scrum-Teams bei der Lösung der Probleme zu helfen. Nur in Notfällen springt das NIT ein und löst Probleme direkt.

Das NIT besteht aus folgenden Personen:

- Dem **Product Owner**, dem Eigentümer des Produkts, der letztendlich für den Produkterfolg verantwortlich ist. Im Rahmen des NIT sorgt der Product Owner dafür, dass von den Teams im Nexus bei jedem Sprint der maximale Wert geliefert wird. Die Rolle des Product Owners ändert sich nicht gegenüber Scrum, nur der Umfang der Arbeit ist komplexer.
- Einem **Scrum Master**, der die Gesamtverantwortung dafür trägt, dass das Nexus-Framework gelebt und verstanden wird. Dieser Scrum Master ist oft ein Scrum Master in einem oder mehreren der anderen Scrum-Teams im Nexus.
- Dem **Entwicklungsteam**, dessen Mitglieder in der Regel ebenfalls Mitglieder von Scrum-Teams im Nexus sind.

Auch wenn der Name es vermuten lässt, das NIT integriert nicht die Arbeit aller Scrum-Teams bei der Auslieferung. Vielmehr ist es dafür verantwortlich, dass die Teams in der Lage sind, die Integration selbst durchzuführen.

NIT-Mitglieder coachen die Scrum-Teams und helfen ihnen, Abhängigkeiten zu beseitigen. Wenn etwas die Scrum-Teams im Nexus daran hindert, ein integriertes Produkt herzustellen, muss das NIT dafür sorgen, dass diese Probleme gelöst werden.



Abb. 2-2 *Das NIT ist für die Maximierung des Werts des integrierten Produkts verantwortlich. (Scrum.org)*

Mitglieder des NIT können auch in den Scrum-Teams des Nexus arbeiten, sie sollten aber zunächst ihre Arbeit im NIT durchführen, da diese den größeren Nutzen für den gesamten Nexus bringt (siehe Abb. 2-3).

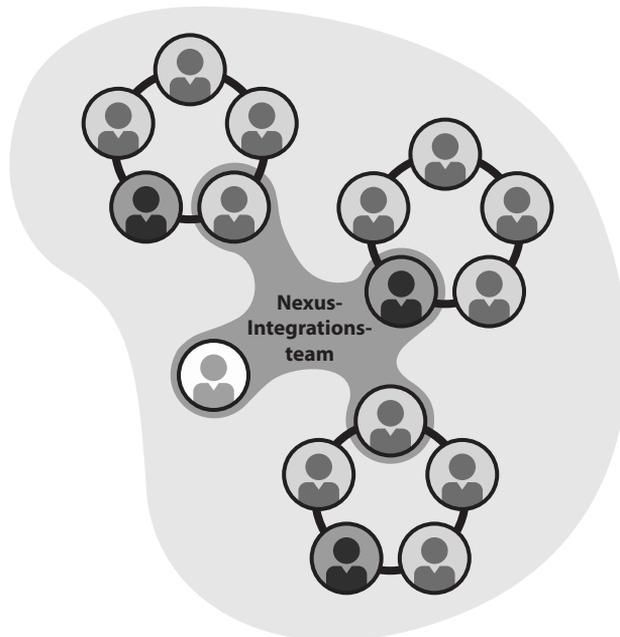


Abb. 2-3 Mitglieder des NIT kommen in der Regel aus Scrum-Teams. (Scrum.org)

NIT-Mitglieder können auch von außerhalb der Scrum-Teams kommen, d.h. aus anderen Teilen der Organisation. In diesem Fall geht es darum, den Scrum-Teams ein spezielles, ihnen fehlendes Fachwissen zur Verfügung zu stellen. Das können Bereiche wie Unternehmensarchitektur oder kontinuierliche Auslieferung (Continuous Delivery) sein oder ein Bereich mit spezialisiertem Domänenwissen. NIT-Mitglieder können auch unterstützen, ohne dass sie tatsächlich vollwertige Mitglieder des Nexus werden. Wenn jedoch eine umfangreiche Unterstützung benötigt wird, kann es sinnvoll sein, dass sie dem Nexus beitreten. In diesem Fall werden sie für die Produktlieferung mitverantwortlich, genau wie die anderen Mitglieder des NIT.

Die Zusammensetzung des NIT kann sich – je nach Bedarf – mit der Zeit ändern. Zu Beginn des Lebenszyklus eines Nexus kann das NIT sich darauf konzentrieren, die Scrum-Teams im Nexus bei Skalierungspraktiken zu coachen, oder es kann stärker an der Stabilisierung des gemeinsamen Build- und Testautomatisierungs-Frameworks beteiligt sein. Später im Lebenszyklus, wenn Nexus reibungslos läuft, kann das NIT dazu übergehen, das Bewusstsein für Probleme aus teamübergreifenden Abhängigkeiten zu schärfen. Kapitel 3 beschreibt die Entstehung, Zusammensetzung und Entwicklung des NIT im Detail.

2.4 Nexus-Ereignisse

Um die Scrum-Teams dabei zu unterstützen, die Arbeit untereinander so effektiv wie möglich zu verteilen und zu koordinieren, fügt Nexus vier Ereignisse zu Scrum hinzu. Zudem wird ein Scrum-Ereignis ersetzt². Die von Nexus definierten Ereignisse sind folgende:

- Das **Refinement** ist ein formelles Ereignis im Nexus-Framework, um gemeinsam an den Details der Product-Backlog-Einträge (Product Backlog Items – PBIs) zu arbeiten und sicherzustellen, dass diese ausreichend unabhängig voneinander sind, sodass die Teams diese ohne übermäßige Konflikte auswählen und daran arbeiten können. Im Zuge der Ermittlung der Abhängigkeiten legen die Teams auch fest, an welchen Backlog-Elementen sie voraussichtlich arbeiten werden. Das Refinement des Product Backlogs im Nexus-Framework erfolgt – je nach Bedarf – kontinuierlich. Eine spezifische zeitliche Obergrenze gibt es für dieses Ereignis nicht.
- Das **Nexus Sprint Planning** unterstützt die Teams dabei, sich gemeinsam auf das Nexus-Ziel und den jeweiligen Beitrag eines Teams zu einigen.
- Das **Nexus Daily Scrum** hilft, Integrationsprobleme transparent zu machen, sodass die Scrum-Teams entscheiden können, wer für das Beheben verantwortlich ist. Es bietet den Teams täglich die Gelegenheit, sich untereinander zu synchronisieren.
- Das **Nexus-Sprint-Review** ermöglicht es, Feedback zum integrierten Produktinkrement zu erhalten. Es ersetzt die Sprint-Reviews der einzelnen Scrum-Teams.
- Die **Nexus-Sprint-Retrospektive** unterstützt die Teams dabei, Erfahrungen auszutauschen und die Lösung gemeinsamer Herausforderungen zu koordinieren.

2.4.1 Refinement

In Scrum ist das Refinement des Product Backlogs kein Pflichtereignis, sondern eine dringend empfohlene Praktik. Im Nexus-Framework ist die Verfeinerung unerlässlich. Sie hilft den Scrum-Teams, gemeinsam zu bestimmen, welches Team spezifische PBIs liefert, und dabei gleichzeitig Querabhängigkeiten zwischen den Teams zu identifizieren. Das Refinement ist ein teamübergreifendes Ereignis, bei

2. Nexus-Ereignisse lehnen sich an die Dauer ihrer zugehörigen Scrum-Ereignisse an. Das bedeutet, dass sie in der Regel genauso viel Zeit brauchen. Praktischerweise braucht ein Nexus-Ereignis so viel Zeit wie der Nexus braucht und ist vorbei, wenn es vorbei ist. Falls der Nexus danach denkt, dass das Ereignis zu lange gebraucht hat, ist das eine gute Möglichkeit für Inspektion und Adaption, um das nächste Mal besser zu sein.

dem so viele Mitglieder der Scrum-Teams anwesend sind wie nötig, um die PBIs zu verstehen und falls erforderlich aufzuteilen.

Das Refinement führt zu einem Product Backlog, das feingranular genug ist, damit die Scrum-Teams ihre Arbeit ohne unüberschaubare Abhängigkeiten erledigen können.

Während des Refinements sollten sich die Scrum-Teams auf die folgenden Fragen konzentrieren:

- Welche Arbeit wird sich jedes Team nehmen?
- In welcher Reihenfolge muss diese Arbeit geleistet werden, um den größtmöglichen Geschäftswert zu erzielen und gleichzeitig Risiken und Komplexität zu minimieren?

2.4.2 Nexus Sprint Planning

Im Nexus-Framework wird das verfeinerte Product Backlog als Input für das Nexus Sprint Planning verwendet (siehe Abb. 2–4). Das Nexus Sprint Planning hilft, die Aktivitäten der Scrum-Teams für einen einzelnen Sprint zu synchronisieren.

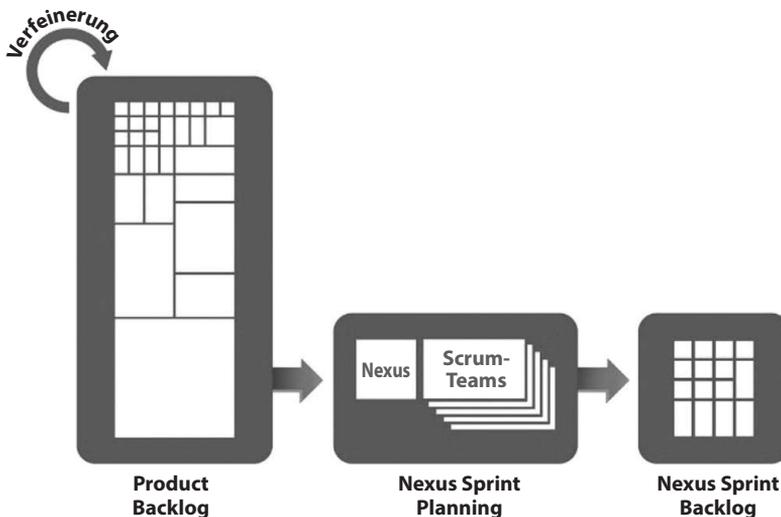


Abb. 2–4 Nexus Sprint Planning (Scrum.org)

Das Nexus Sprint Planning besteht aus folgenden Teilen:

■ Validierung des Product Backlogs

Die Scrum-Teams überprüfen die PBIs und nehmen alle notwendigen Anpassungen an der Arbeit aus dem Refinement vor. Alle Scrum-Teams sollten teilnehmen und dazu beitragen, Kommunikationsprobleme zu minimieren. Not-

wendig ist die Teilnahme allerdings nur für die Vertreter aus den Scrum-Teams, die glauben, dass sie einen Beitrag zur Verfeinerung der PBIs leisten können.

■ Formulierung des Nexus-Ziels

Das Nexus-Ziel ist ein Sprint-Ziel, das durch die Implementierung von PBIs durch mehrere Teams erreicht wird.

■ Sprint Planning für Scrum-Teams

Sobald das Nexus-Ziel für den Sprint verstanden ist, führt jedes Scrum-Team ein individuelles Sprint Planning durch, bei dem die Mitglieder jeweils ihr eigenes Sprint Backlog erstellen. Wenn sie Abhängigkeiten mit anderen Teams identifizieren, arbeiten sie mit diesen Teams zusammen, um die Abhängigkeiten zu minimieren oder zu beseitigen.

In manchen Fällen bedeutet dies, dass die Reihenfolge der Arbeiten zwischen den Teams angepasst werden muss, damit ein Team seine Arbeit beenden kann, bevor ein anderes beginnt.

Dies könnte dadurch erreicht werden, dass abhängige Arbeiten aufgeteilt werden, sodass einzelne Teile unabhängig voneinander bearbeitet werden können. Alternativ kann ein Team unabhängige PBIs zur Bearbeitung auswählen. Ziel ist jeweils, die Verschwendung durch ungelöste, teamübergreifende Abhängigkeiten zu vermeiden. Die Teams können auch zusammen überlegen, wie sie die Arbeit verlagern, um sie besser unter den Teams zu verteilen. Das NIT wird dabei helfen, sicherzustellen, dass Abhängigkeiten kommuniziert und im Nexus Sprint Backlog visualisiert werden.

Das Nexus Sprint Planning ist abgeschlossen, wenn jedes Scrum-Team des Nexus die eigene Planung beendet hat.

2.4.3 Das Nexus Daily Scrum

Im Nexus Daily Scrum kommen die relevanten Vertreter der einzelnen Teams zusammen, um den aktuellen Stand des integrierten Inkrements zu bewerten und Integrationsprobleme oder neu entdeckte Abhängigkeiten zwischen den Teams zu identifizieren. Typischerweise werden die folgenden Themen diskutiert:

- Wurde die Arbeit des vorherigen Tages erfolgreich integriert?
Und wenn nicht, warum?
- Wurden neue Abhängigkeiten identifiziert?
- Welche Informationen müssen zwischen den Teams im Nexus ausgetauscht werden?

Im Nexus Daily Scrum und über den gesamten Tag wird das Nexus Sprint Backlog von den Scrum-Teams aktualisiert, um die Abhängigkeiten zwischen den Teams zu visualisieren und zu managen. Dabei ist das Nexus Sprint Backlog keine einfache Aggregation der jeweiligen Team Sprint Backlogs, da jedes Team neben den für das Product Backlog notwendigen Tätigkeiten auch eigene Tätigkeiten auszuführen hat. Die während des Nexus Daily Scrum identifizierten Aufgaben werden in die Teams mitgenommen, um sie dort im Rahmen ihres nächsten Daily-Scrum einzuplanen.

2.4.4 Das Nexus-Sprint-Review

Das Nexus-Sprint-Review ersetzt die Sprint-Reviews der einzelnen Teams und findet am Ende des Sprints statt. Ziel ist es, Feedback von Stakeholdern zum gesamten integrierten Produktinkrement zu erhalten. Das Nexus-Sprint-Review ersetzt die einzelnen Scrum-Team-Sprint-Reviews, da einzelne Scrum-Teams bei der Verwendung von Nexus möglicherweise kein sinnvolles integriertes Inkrement erzeugen.

Ein einziges Nexus-Sprint-Review bietet viele Vorteile, einschließlich der folgenden:

- Teams sind logischerweise die Stakeholder der anderen, sodass sie sich gegenseitig Feedback geben können, das allen Beteiligten im Nexus hilft, sich zu verbessern.
- Wenn einzelne Scrum-Team-Sprint-Reviews durchgeführt werden, können Stakeholder möglicherweise nicht an allen teilnehmen, und selbst wenn, haben sie nicht das gesamte integrierte Produkt vor sich.
- Einige Probleme treten erst auf, wenn das integrierte Produkt als Ganzes betrachtet wird, insbesondere wenn jedes Team eine oder mehrere Komponenten entwickelt. Jede Komponente mag für sich funktionieren, aber es kann passieren, dass sie nicht zusammen funktionieren, um ein integriertes Produkt herzustellen.
- Die Überprüfung des integrierten Produktinkrements als Ganzes bringt alle Teams im Nexus zusammen und erinnert sie daran, dass ihr Ziel eine einzige integrierte Lösung ist.

Selbst wenn einige Teams tatsächlich logisch getrennte Teilprodukte liefern, die unabhängig geprüft, ausgeliefert und verwendet werden können, ist es sinnvoll, sie im Kontext des gesamten integrierten Produktinkrements zu betrachten.

Alle Nexus-Mitglieder nehmen am Nexus-Sprint-Review teil.

2.4.5 Die Nexus-Sprint-Retrospektive

Die Nexus-Sprint-Retrospektive stellt innerhalb des Nexus-Frameworks die für das Überprüfen und Anpassen notwendigen Instrumente zur Verfügung. Folgende Aktivitäten werden im Rahmen einer Nexus-Sprint-Retrospektive durchgeführt:

1. Vertreter aus dem gesamten Nexus treffen sich und identifizieren Probleme, die mehr als ein einziges Team daran gehindert haben, gemeinsame Probleme für alle Scrum-Teams transparent zu machen. Die Vertreter bestehen aus den NIT-Mitgliedern sowie aus allen, die daran interessiert sind, ihre Ansichten zu teamübergreifenden Themen auszutauschen.
2. Jedes Scrum-Team veranstaltet seine eigene Sprint-Retrospektive, genau wie in Scrum. Die Themen aus dem ersten Teil der Retrospektive werden zusätzlich als Input für die Teamdiskussion verwendet. Aktionen werden für Team- und teamübergreifende Themen festgelegt.
3. Vertreter der Scrum-Teams treffen sich erneut, um gemeinsame Themen der Scrum-Team-Retrospektiven zu diskutieren. Sie einigen sich darauf, wie die identifizierten Aktionen, die allen Beteiligten im Nexus ermöglichen, als Ganzes zu lernen und Veränderungen einzuleiten, visualisiert und verfolgt werden können.

2.4.6 Fragen, die in jeder Nexus-Sprint-Retrospektive gestellt werden sollen

Nahezu jede Nexus-Produktentwicklung stößt auf ähnliche Herausforderungen bei der Skalierung. Zu den Fragen, die den Teams helfen, Herausforderungen zu erkennen, gehören die folgenden:

- Gibt es unerledigte Arbeiten?
- Wurden im Rahmen der Entwicklung technische Schulden generiert?
- Wurden alle Artefakte, insbesondere der Code, häufig (möglichst jeden Tag) erfolgreich integriert?
- Wurde die Software erfolgreich erstellt, getestet und oft genug eingesetzt, um einem Berg ungelöster Abhängigkeiten vorzubeugen?

Wenn Herausforderungen erkannt werden, fragen Sie Folgendes:

- Warum ist das passiert?
- Wie können technische Schulden rückgängig gemacht werden?
- Wie kann die Wiederholung verhindert werden?

Die Nexus-Ereignisse werden in Kapitel 5 noch detaillierter beschrieben.

2.5 Nexus-Artefakte

Artefakte erfassen die Ergebnisse der durchgeführten Arbeiten. Sie bieten zudem Transparenz und bilden damit die Grundlage für das Überprüfen und Anpassen.

2.5.1 Product Backlog

Es gibt ein einziges Product Backlog für den gesamten Nexus und alle darin arbeitenden Scrum-Teams. Da mit dem Nexus-Framework ein einzelnes Produkt organisiert wird, gibt es nur einen einzigen Product Owner, und dieser einzelne Product Owner verwaltet ein einziges Product Backlog. Alle Teams ziehen die Arbeit aus diesem einzelnen Artefakt.

2.5.2 Nexus-Ziel

Während des Nexus Sprint Planning bespricht der Product Owner ein Ziel für den Sprint, das sogenannte Nexus-Ziel. Es ist die Summe aller Arbeiten und Sprint-Ziele der einzelnen Scrum-Teams innerhalb des Nexus. Dabei sollte die Funktionalität, die zur Erreichung des Nexus-Ziels entwickelt wurde, beim Nexus-Sprint-Review demonstriert werden.

2.5.3 Nexus Sprint Backlog

Im Nexus Sprint Backlog werden die PBIs erfasst, die teamübergreifende Abhängigkeiten oder mögliche Integrationsprobleme enthalten. PBIs, die keine Abhängigkeiten aufweisen, und Aufgaben aus den einzelnen Sprint Backlogs der Scrum-Teams werden nicht eingetragen. Das Nexus Sprint Backlog dient dazu, Abhängigkeiten und den Arbeitsfluss während des Sprints hervorzuheben. Die Aktualisierung erfolgt mindestens täglich, oft im Rahmen des Nexus Daily Scrum.

2.5.4 Integriertes Inkrement

Das integrierte Inkrement ist die integrierte Zusammenfassung aller Arbeiten *aller* Scrum-Teams in einer Nexus-Implementierung. Das integrierte Inkrement muss nutzbar und potenziell releasefähig sein, d.h., es muss die vom Entwicklungsteam vereinbarte »Definition of Done« erfüllen. Der Product Owner ist ein wichtiger Stakeholder und definiert die Qualitätskriterien, denen das Produktinkrement entsprechen muss. Das integrierte Inkrement wird beim Nexus-Sprint-Review betrachtet.

2.5.5 Transparenz der Artefakte

Genau wie das dahinterliegende Scrum-Framework basiert Nexus auf Transparenz. Das NIT arbeitet mit den Scrum-Teams im Nexus und der weiteren Organisation zusammen, um sicherzustellen, dass alle Scrum- und Nexus-Artefakte sichtbar sind und der Zustand des integrierten Inkrements leicht nachvollzogen werden kann.

Entscheidungen, die aufgrund des Zustands der Nexus-Artefakte getroffen werden, sind nur so effektiv wie die Transparenz der Artefakte. Unvollständige Informationen führen zu falschen oder fehlerhaften Entscheidungen. Das macht es schwierig oder unmöglich, einen Nexus effektiv so zu managen, dass das Risiko minimiert und der Wert maximiert wird.

Die größte Herausforderung bei der Verwendung des Nexus-Frameworks besteht darin, Abhängigkeiten zu erkennen und aufzulösen, bevor sich technische Schulden in unzumutbarer Höhe ansammeln. Die Integration der Arbeit aus den Scrum-Teams ist der beste Test für inakzeptable technische Schulden. Wenn diese Integration fehlschlägt, bleiben die ungelösten Abhängigkeiten unentdeckt im Code und in der Testumgebung bestehen, wodurch der Wert der Software gemindert wird oder sogar ganz verfällt.

2.5.6 »Definition of Done« im Nexus-Framework

Das NIT ist für die »Definition of Done« (DoD) verantwortlich, die auf das bei jedem Sprint entwickelte integrierte Inkrement angewendet werden kann. Alle Scrum-Teams einer Nexus-Umgebung halten sich an diese »DoD«.

Das Inkrement ist erst fertig, wenn es vom Product Owner als brauchbar und potenziell auslieferbar eingestuft wurde. Ein PBI kann als »fertig« betrachtet werden, wenn dessen Funktionalität erfolgreich in das Produkt integriert wurde.

Alle Scrum-Teams sind verantwortlich für die Entwicklung und Integration ihrer Arbeit in ein Inkrement, das diese Attribute erfüllt. Einzelne Scrum-Teams können sich für eine strengere »Definition of Done« innerhalb ihres eigenen Teams entscheiden, aber sie können nicht weniger strenge Kriterien anwenden als diejenige, die für das gesamte Inkrement vereinbart wurden.

2.6 Was brauchen Sie, um mit Nexus zu beginnen?

Nexus basiert auf dem Scrum-Framework und fügt ein paar wenige Ereignisse, Rollen und Artefakte hinzu, um die Transparenz, Kommunikation und Zusammenarbeit zwischen den Teams zu erhöhen. Die neuen Nexus-Ereignisse, -Rollen und -Artefakte tragen dazu bei, dass ein erfolgreiches integriertes Inkrement entwickelt wird. Genau wie bei Scrum brauchen Sie nicht viel, um mit Nexus anzufangen. Und genau wie bei Scrum ist Nexus einfach zu erlernen, aber schwer zu beherrschen. Nachfolgend sind die minimalen, aber notwendigen Voraussetzungen aufgeführt, um Nexus zu implementieren.

Diese Punkte sollten gegeben sein:

- ✓ Erfahrung in der Arbeit mit Scrum
- ✓ Ein Product Backlog und ein Produkt Owner für jedes einzelne Produkt
- ✓ Festlegung der Teams, die zusammen als Nexus arbeiten werden. Diese sollten einen Überblick über das Nexus-Framework haben.
- ✓ Klarheit über die Mitglieder, die das NIT des Nexus bilden werden.
- ✓ Eine Definition of Done.
- ✓ Eine festgelegte Sprint-Länge.

2.7 Fazit

Nexus ist einfach zu verstehen, aber es erfordert Übung und Feedback. Wie bei Scrum sind die Grundkonzepte einfach. Und wie Scrum ist auch Nexus nicht präskriptiv. Es besagt z.B., dass Sie Nexus bei der Planung des Sprints einsetzen müssen, aber es sagt Ihnen nicht, wie Sie das tun sollen. Für die Planung gibt es viele verschiedene Techniken, die für Sie hilfreich sein könnten. Die Auswahl dürfen Sie selbst treffen. In den folgenden Kapiteln werden wir die Anwendung von Nexus in einer Fallstudie untersuchen. Dabei werden wir oft veranschaulichen, wie Nexus funktioniert, indem wir spezifische ergänzende Praktiken anwenden. Diese sind nicht speziell Teil von Nexus, aber sie werden Ihnen helfen, die Funktionsweise von Nexus besser zu verstehen. Ab Kapitel 3 beschreiben wir, was es braucht, um einen Nexus zu bilden.