

Ein Scrum-Framework für ein Mehr an Teams und ein Mehr an Wert

Sieben Fragen zur Skalierung mit Nexus in der Praxis

Die Herausforderung bei Nexus beginnt, wenn die darin beschriebenen Events, Artefakte usw. in einem realen Projekt Gestalt annehmen sollen. Typische Fragen dazu beziehen sich beispielsweise auf die Rolle des Nexus Integration Teams (NIT), die Organisation des Reviews und vor allem auf den Umgang mit Abhängigkeiten. Dieser Artikel soll Scrum Mastern, Agile Coaches, Management und Product Ownern bei der konkreten Implementierung des Frameworks helfen.

Unternehmen skalieren Scrum in der Absicht, mit einem Mehr an Teams ein Mehr an Wert zu erzielen. Das klingt einfach. Nexus bietet ein leichtgewichtiges, klar strukturiertes Framework, um Scrum zu skalieren. Auch das klingt einfach. Was macht dann die Skalierung zu einem derart komplexen Thema, an dem einige Organisationen scheitern und andere zumindest hinter ihren Zielen zurückbleiben?

Unserer Meinung nach ist es der altbekannte Unterschied zwischen Theorie und Praxis. Ein Beispiel: Laut Nexus Guide [Sch18] dient das Nexus Sprint Review dazu, Feedback zum integrierten Inkrement zu erhalten. Diese Beschreibung zu lesen, ist eine Sache. Eine ganz andere ist es, das Review abzuhalten, sprich, 60 oder 80 Personen in einem Raum zu versammeln und zu wissen, dass hier gerade sehr viel Zeit und Geld investiert wird. Was sollte Unternehmen also bewusst sein, wenn sie sich fürs Skalieren entscheiden? Was ist bei Nexus anders als bei Scrum und vor allem, welche Fragen treten häufig auf, wenn Nexus in die Praxis umgesetzt wird?

Der Einstieg: Warum skalieren?

Eine Abgrenzung zu Beginn: Dieser Artikel befasst sich nicht mit der agilen Transformation, der Verbreitung von Scrum innerhalb einer Organisation. Wir verstehen unter Skalierung die Situation, in der für ein Produkt mehrere Entwicklerteams zusammenarbeiten, und das mindestens einen Sprint lang. Was dabei als derjenige Wert verstanden wird, der maximiert werden soll, legt der Product Owner fest. Letztendlich soll mit dem Produkt Geld verdient werden, dafür kann er oder sie unterschiedliche Faktoren als entschei-



dend werten. Häufig ist es eine verkürzte time-to-market, es kann beispielsweise auch eine weitreichende Risikominimierung sein.

Unabhängig davon, welchen Vorteil die Skalierung erzielt – sie bringt immer auch Nachteile. Am einfachsten erklären lässt sich das mit dem Beispiel einer Fahrt auf der Autobahn. Im ersten Fall ist entscheidend, möglichst früh anzukommen, also wird die Geschwindigkeit erhöht und mit ihr der Spritverbrauch, was die Kosten und die Umweltbelastung steigen lässt. Im zweiten Fall sind niedrige Kosten entscheidend, daher senkt der Fahrer die Geschwindigkeit. Die Fahrt wird billiger, dauert aber länger. Der Product Owner muss genau abwägen, ob der mit der Skalierung zu erzielende Vorteil alle Nachteile überkompensiert, denn zu skalieren, verursacht Kosten, das ist unver-

meidlich. Variabel ist lediglich die Höhe dieser Kosten.

Nexus – was ist daran anders als bei Scrum mit einem Team?

Scrum bezieht sich auf die Arbeit eines Scrum-Teams, bei Nexus können es viele sein. Ken Schwaber, Co-Schöpfer von Scrum, bezeichnet Nexus als „Exoskelett von Scrum“ und verdeutlicht damit, wofür es geht: um Scrum mit einigen wichtigen Erweiterungen (siehe **Abbildung 1**). Am Grundsätzlichen ändert sich wenig. Scrum ist ein empirischer, auf „inspect & adapt“ basierender Ansatz, in dem Teams fertige Inkremente liefern und damit Wert erzeugen. Bei Nexus tritt an die Stelle des fertigen Inkrements am Ende des Sprints das fertige integrierte Inkrement: Dieses Inkrement vereint die Arbeit aller Teams.

Weil es genau ein integriertes Inkrement gibt, gibt es weiterhin auch nur einen Product Owner und ein Product Backlog. Wenn mehrere Teams zusammenarbeiten, dann entstehen Abhängigkeiten, sie können technisch sein, fachlich oder personell, teilweise beziehen sie sich auf die Software. Diese Abhängigkeiten frühzeitig und fokussiert zu lösen oder einzuplanen, das ist ein Erfolgskriterium der Skalierung. Gleichzeitig ist es in der Praxis eine der größten Herausforderungen. Nicht umsonst taucht der Begriff „Abhängigkeiten“ mehr als 30 Mal im Nexus Guide auf. Das Nexus-Framework soll ermöglichen, diese Abhängigkeiten zu managen. Im Vergleich zu Scrum enthält Nexus eine weitere Rolle, zwei zusätzliche Artefakte und fünf ergänzende Events. Die erste häufige Frage bezieht sich auf die neue Rolle: Das Nexus Integration Team.

Frage eins: Was macht unser Nexus Integration Team (NIT)?

„Das Nexus Integration Team ist dafür ergebnisverantwortlich (accountable), dass mindestens jeden Sprint ein integriertes Inkrement (die gemeinsame, fertigestellte Arbeit eines Nexus) produziert wird ...“ So steht es im Nexus Guide [Sch18]. Das NIT ist ein Scrum-Team, dazu gehören auf jeden Fall ein Scrum Master, der Product Owner und ein oder mehrere weitere

Teammitglieder. Die Kernaufgabe des NIT ist es, die Teams zu befähigen, das Inkrement zu liefern. Dazu unterstützt und fördert es die Integration und Qualität und macht deren Stand transparent. Im Sinne der Professionalität bietet das NIT auch Coaching in Engineering-Praktiken, beispielsweise mittels eines temporären Pair Programmings. Das NIT spielt auch eine Rolle bei den zusätzlichen Meetings, beispielsweise betreut es den Nexus Daily. Darüber hinaus setzt sich das NIT dafür ein, dass diese Extra-Meetings dazu beitragen, die Arbeit aller Teams sinnvoll zusammenzubringen. De facto bedeutet das, die Abhängigkeiten zwischen den Teams und, soweit vorhanden, von Dritten offenzulegen und zu lösen. Zudem kann das NIT, wie jedes andere Team auch, Verbesserungen durchführen, die in der Nexus Retrospektive beschlossen wurden.

Damit das NIT funktionieren kann, sind nach unserer Erfahrung vor allem drei Voraussetzungen wichtig: Erstens sollten die Teammitglieder ihre Mitgliedschaft in Vollzeit ausüben können. Wir halten ein Vorgehen im Sinne von „eigentlich entwickle ich Features in Team X, aber wenn ich Zeit habe, mache ich ein bisschen mit im NIT“ für unrealistisch.

Damit kommen wir zur zweiten Voraussetzung: NIT-Mitglieder sind nicht nur mit Fähigkeiten wie Soft Skills ausgestattet, sie sind auch exzellente Entwickler. Der letzte Punkt sorgt oft für Diskussi-

onen, denn der Product Owner hat vermutlich wenig Interesse daran, einige der besten Entwicklerinnen und Entwickler von der Feature-Entwicklung abzugeben. Das ist jedoch erforderlich, damit im NIT die nötigen Kompetenzen vertreten sind. Wichtiger dritter Punkt ist die Co-Lokation. Die Möglichkeit zur direkten, spontanen Kommunikation ist für das NIT essenziell.

Bei allen seinen Aufgaben und besonders, wenn es das Feedback der Entwickler im Review einholt, darf das NIT an einem Punkt keinen Zweifel lassen: Das NIT ist ein Team, das das Gesamtbild des Projekts im Blick hat. Es ist kein Kopfteam im Sinne einer übergeordneten, hierarchischen Instanz und keine Kontrollinstanz. NIT kommandiert die anderen Teams nicht, es befähigt und unterstützt sie.

Frage zwei: Wie lange dauern die veränderten Meetings? Sind es nicht zu viele?

Fünf weitere beziehungsweise erweiterte Meetings zusätzlich zu den vier Events in Scrum können durchaus die Frage aufwerfen, ob nicht ein übertrieben hoher Anteil der Arbeitszeit nur noch dafür verwendet wird, zusammensitzen und zu reden. Dazu kommt, dass der Nexus Guide keine expliziten Angaben zur Länge der einzelnen Events macht. Der Hinweis, es handele sich „um zusätzliche

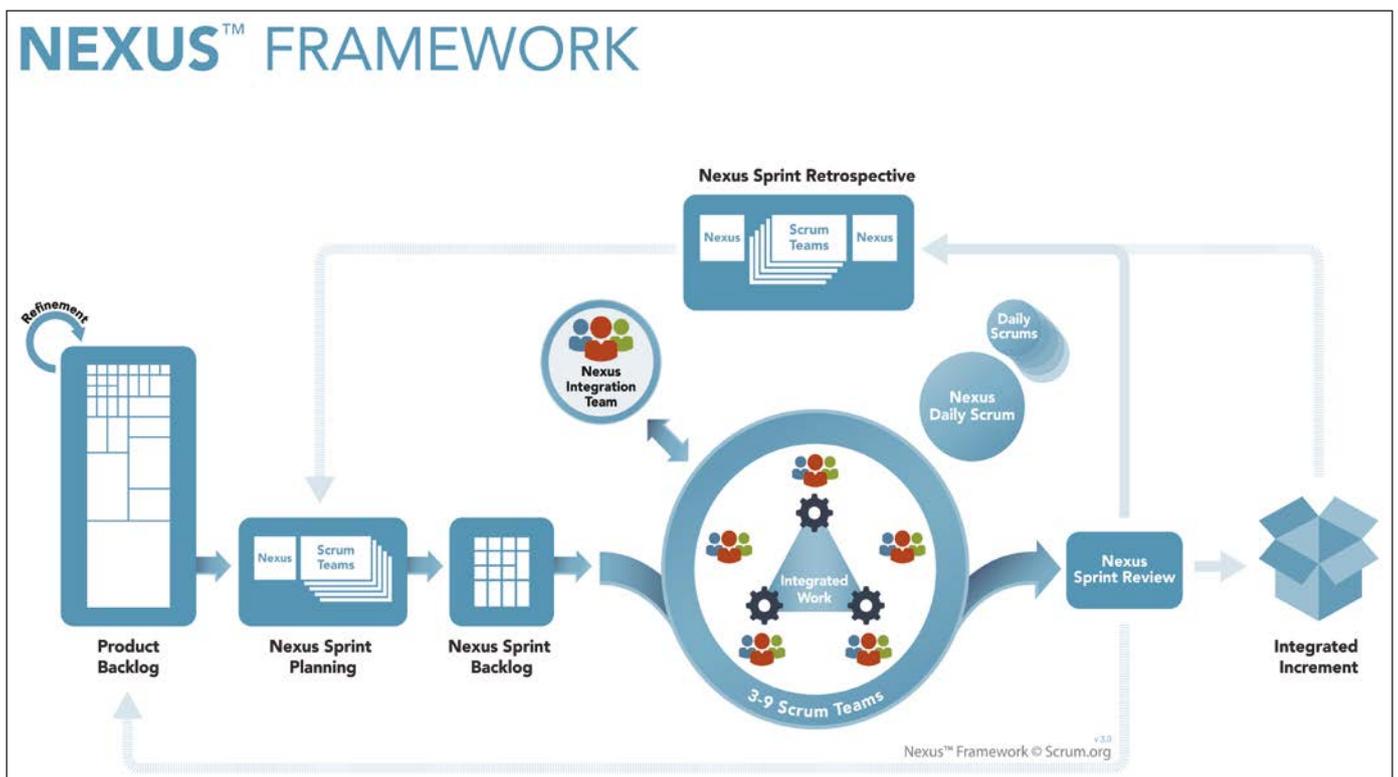


Abb. 1: Darstellung Nexus Framework, Nexus™ Framework © Scrum.org

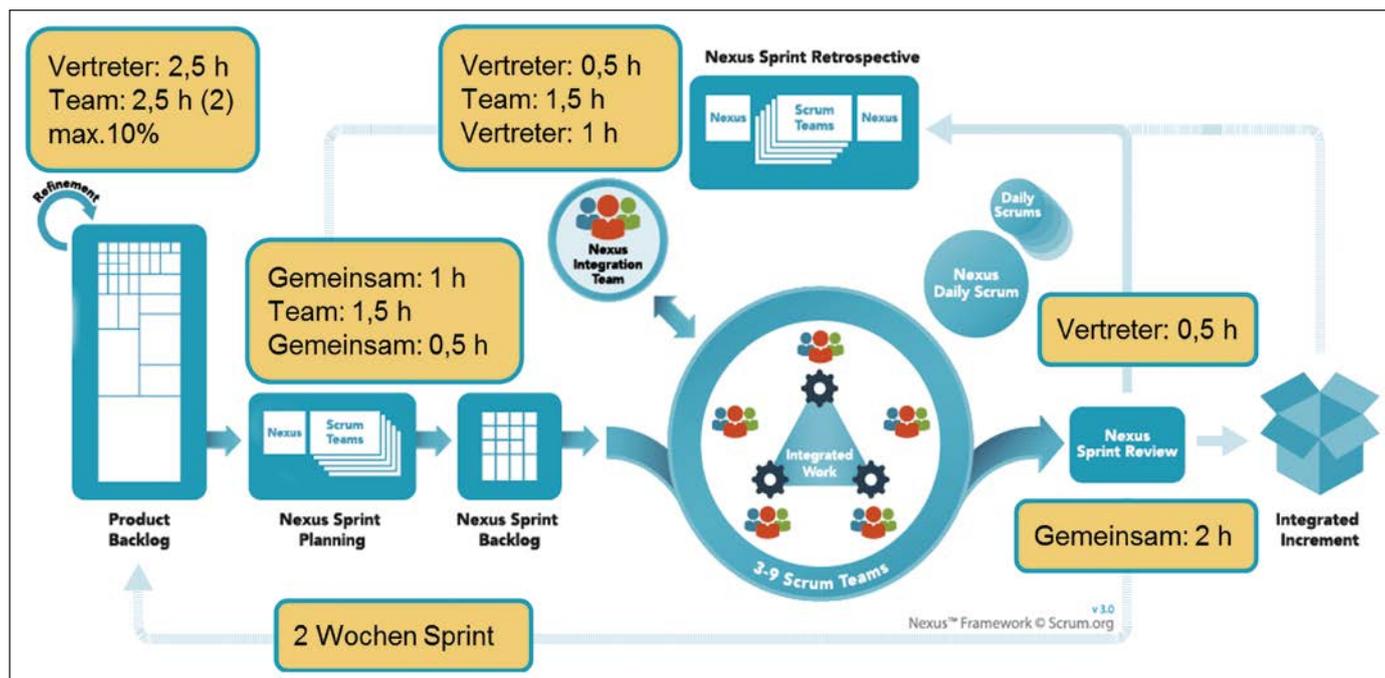


Abb. 2: Unsere Vorschläge für Timeboxen zu den einzelnen Nexus-Meetings

Timeboxen zu den entsprechenden Scrum Events“ [Sch18] lässt eine Menge Raum für Interpretationen.

Ist es nicht sinnvoller, wenigstens ein oder zwei der Meetings wegzulassen? Nach unseren Erfahrungen ist die Antwort ein klares Nein. Die Meetings repräsentieren bereits die minimale Struktur, um Abhängigkeiten gezielt aufzulösen. Wir empfehlen jedoch, diese „zusätzlichen Timeboxen“ klar festzulegen und einzuhalten. Bei dem, was sich bei uns bewährt hat, entstehen innerhalb eines 2-wöchigen Sprints die in **Abbildung 2** vorgeschlagenen zeitlichen Aufwände.

Die einzigen beiden Meetings, bei denen alle Projektbeteiligten persönlich anwesend sein sollten, sind nach unserer Erfahrung Nexus Planning und Nexus Review. Zu den anderen drei Eventtypen entsenden die einzelnen Entwicklerteams Vertreter. Wie sich die verschiedenen Meetings gestalten lassen, dieser Thematik widmen sich die nächsten Fragen.

Frage 3: Was passiert im Nexus Daily?

Im Grunde gibt es nicht das Nexus Daily, sondern es sind zwei Dailys: das klassische Daily jedes Teams und, laut Nexus Guide davor, das Nexus Daily.

Das Nexus Daily hat den Zweck, Transparenz zu schaffen über den aktuellen Stand der Integration und den Tagesplan des Nexus zu erarbeiten. Anwesend sind Repräsentanten der Entwicklerteams, häufig auch der Product Owner sowie

ein Scrum Master. Sobald Abhängigkeiten von Externen offensichtlich werden, etwa von Operations, sollten diese Teams ebenfalls im Nexus Daily vertreten sein. Sie alle besprechen im Nexus Daily die Qualität des Inkrements und inwieweit es „done“ ist, diskutieren Abhängigkeiten und den Verlauf der Integration des Vortages. Außerdem richten sie den Blick auf das Nexus Goal, das übergeordnete Ziel, und klären, ob und wie sie auf dieses Ziel hinarbeiten.

Ein Nexus Daily sollte unserer Meinung nach maximal 30 Minuten in Anspruch nehmen.

Frage 4: Wie machen wir Refinements mit so vielen Teams?

In Scrum ist das Refinement eine Empfehlung, in Nexus ein Muss. Es dient dazu, aus vagen Ideen oder Zielen Sprint-taugliche User-Stories zu machen und Abhängigkeiten frühzeitig zu identifizieren. Umgangssprachlich gesagt ist ein Refinement ein Meeting, in dem sich ein „Ich möchte X“ des Product Owners verwandelt in ein „mit dem gemeinsamen Verständnis, das wir jetzt haben, können wir anfangen, X umzusetzen“ des Entwicklerteams.

In der Größenordnung des Nexus gibt es viele Ebenen von Refinement. Nur wenn die Product Backlog Items in einem angemessenen Grad unabhängig voneinander sind, können sie ohne übermäßige Konflikte zwischen den Scrum-Teams im Nexus ausgewählt und bearbeitet werden. Nach unserer Erfahrung sind kontinuier-

liche Refinements vor allem im skalierten Umfeld zwingend notwendig. Einerseits dienen sie dazu, die ganze Strategie zu betrachten. Andererseits wird in ihnen die nahe Zukunft – beispielsweise die nächsten drei Sprints – detailliert geplant und nicht nur vage geschätzt.

Wir haben gute Erfahrungen damit gemacht, zwei Refinement-Typen zu etablieren: Das übergreifende „Cross Refinement“ und die „lokalen Refinements“ der einzelnen Teams beziehungsweise für konkrete Themen.

In das Cross Refinement entsendet jedes Entwicklerteam einen Repräsentanten, zwingend erforderlich ist die Teilnahme des Product Owners sowie weiterer benötigter Stakeholder. Das Cross Refinement verfolgt mehrere Ziele: Zum einen stellt der Product Owner die aktuelle Roadmap vor, wobei frühzeitig neue große Themen identifiziert und ein teamübergreifender Plan fürs Refinement erarbeitet wird. Zum anderen gibt es auch die Möglichkeit, aus Entwicklerteam-Sicht technische Fragestellungen einzuwerfen, die teamübergreifend erarbeitet werden sollen. Zusätzlich zu den strategischen Punkten wird auch der gemeinsame Plan der nächsten drei Sprints aktualisiert. Dabei sollen Abhängigkeiten identifiziert und ein Plan zur Lösung erarbeitet werden.

Die lokalen Refinements haben einen klaren Themen- oder Teamfokus. Sie finden während eines Sprints statt, in ihnen werden Features zerlegt in einzelne, Sprint-taugliche User-Stories. Teilnehmende sind das oder die umsetzenden Entwickler-

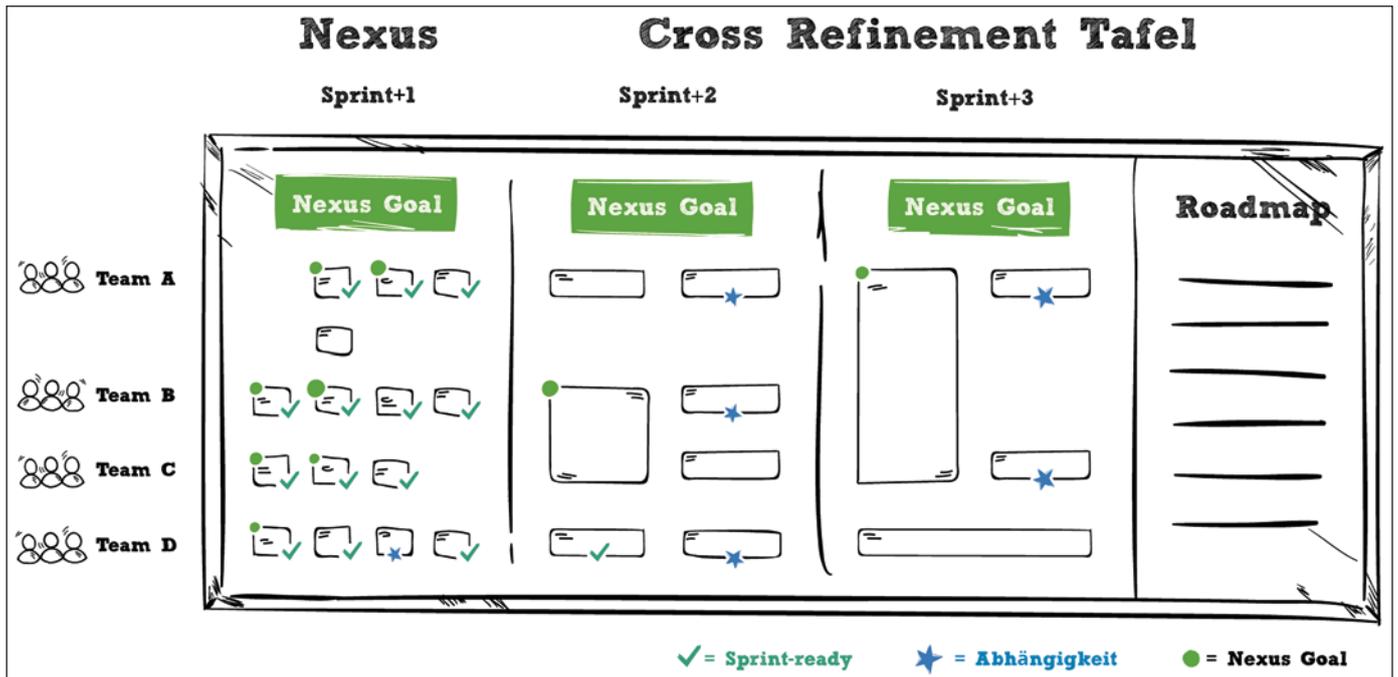


Abb. 3: Ein Beispiel für die Darstellung der Sprints usw. im Cross-Refinement

teams, der Product Owner und weitere benötigte Stakeholder, zum Beispiel Mitglieder des Fachbereichs. Die Refinements sollten maximal 10 Prozent der Teamkapazität beanspruchen. Entscheidend für die Refinements ist nach unserer Erfahrung der Aspekt der Visualisierung. Dazu bietet sich an, tragbare Whiteboards oder andere „Tafeln“ zu verwenden, auf denen die Sprints und die User-Stories gut sichtbar abgebildet sind. Sehr erhellend ist auch, Abhängigkeiten zu markieren, zum Beispiel mit Zusatz-

Kärtchen, die von den Teams an die User-Stories geklebt werden und die fachliche oder technische Abhängigkeiten aufzeigen (siehe Abbildung 3). Über allen Sprints steht, für alle offensichtlich, das Nexus Goal.

Es klingt simpel: Wir illustrieren Abhängigkeiten an der Wand und zeigen zusätzlich, welche Storys „sprint-ready“ sind. Tatsächlich trägt dieses Vorgehen deutlich dazu bei, Transparenz zu schaffen und sicherzustellen, dass alle Beteiligten im Wortsinn das gleiche Bild haben. Auch die Refinements leben vom direkten Kontakt und Austausch. Ergebnisse der lokalen Refinements fließen in das nächste Cross Refinement ein und umgekehrt. Bis ins Detail geklärte, gut verstandene Storys sind die Grundlage für das Nexus Sprint Planning.

aufwendig, jedoch aus zwei Gründen ein Vorteil: Erstens stärkt die gemeinsame Teilnahme den „Wir haben alle das gleiche Produkt“-Gedanken. Zweitens haben so alle Beteiligten die Möglichkeit, ihr Wissen einzubringen und Abhängigkeiten gemeinsam zu erkennen und zu lösen. Mithilfe der Moderation eines Scrum Masters entsteht das gemeinsame Planning Board, auf dem auch Sprint-Ziele und das Nexus Goal visualisiert werden. Um die empfehlenswerte Dauer von maximal 60 Minuten einzuhalten, ist eine gute Vorbereitung wichtig. Das heißt, alle Teams sollten ihre eigenen Storys bereits ausgedruckt dabei haben.

Teil zwei: Team Planning

Das gemeinsame Planning Board aus Teil eins zeigt die Aufgaben, die für jedes einzelne Team anstehen. Rund 90 Minuten lang gehen die Teams jetzt daran, daraus ihre individuelle Sprint-Planung zu erstellen.

Teil drei: Nexus Sprint Planning, der Abschluss

Nach Abschluss der Team Plannings kommen alle Teams erneut zusammen und besprechen neu erkannte Abhängigkeiten, evaluieren, inwieweit ihr Planning auf das Nexus Goal hinarbeitet, und finalisieren die gemeinsame Planung. Das Nexus Sprint Backlog zeigt dann den gemeinsamen letzten Stand (siehe Abbildung 4). Dieser letzte Teil sollte maximal 30 Minuten in Anspruch nehmen. Zugegeben, die beiden großen Meetings – Nexus Sprint

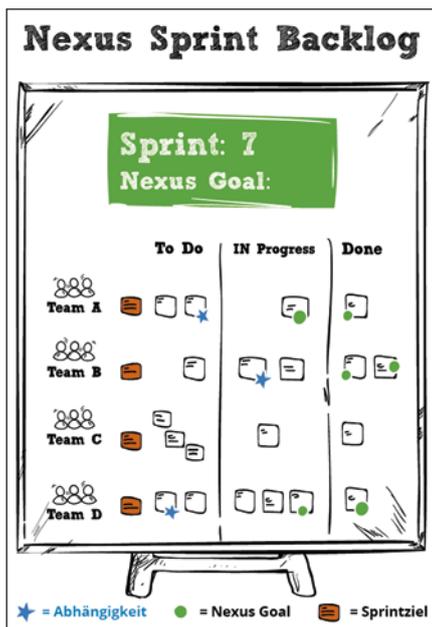


Abb. 4: Ein Beispiel für ein Nexus Sprint Backlog

Frage 5: Wie machen wir eine zielgerichtete Sprintplanung mit so vielen Teams?

Das Nexus Sprint Planning ist ein Planning mit bis zu neun Teams und damit eine Herausforderung. Wir sind ihr in der Praxis begegnet, indem wir das Planning in drei Teile gegliedert und die jeweiligen Ziele ganz klar voneinander abgrenzt haben.

Teil eins: Nexus Sprint Planning, der Anfang

Dazu laden wir alle Teams ein, die an ihrem jeweiligen Teamtisch – alle im gleichen Raum – Platz nehmen. Das ist

Literatur & Links

[Sch18] Ken Schwaber und Scrum.org, Nexus™ Guide, Januar 2018, siehe: <https://www.scrum.org/resources/nexus-guide>, abgerufen wurde die deutsche Version vom Januar 2018

Planning und das im nächsten Absatz beschriebene Review – sind allein logistisch ein Aufwand. Trotzdem bleiben sie, ungeachtet der großen Personenanzahlen, gut steuerbar, wenn die drei Voraussetzungen eingehalten werden:

- gute Vorbereitung (fertige User-Stories usw.),
- Einhalten des Zeitrahmens (eine Aufgabe des Moderators),
- das Nexus Goal bleibt als übergeordnetes Ziel immer im Fokus.

Frage 6: Wie machen wir ein effektives Review mit so vielen Teams und Stakeholdern?

Das Nexus Sprint Review ersetzt individuelle Scrum Team Sprint Reviews, da das komplette integrierte Inkrement im Fokus steht. Der Nexus Guide meint dazu, es sei „nicht möglich, die komplette Arbeit im Detail zu zeigen. Bestimmte Techniken können nötig sein, um das Feedback der Stakeholder zu maximieren.“

Dieser Satz beschreibt die Ambivalenz auf einer abstrakten Ebene: Einerseits sollen im Review alle Teams ihre Arbeit zeigen können und wertvolles Feedback erhalten, andererseits droht der Review dann zu einer Art Marathon-Event zu werden. Wir haben gute Erfahrungen damit gemacht, den Review als Marktplatz zu konzipieren: Alle Teams bekommen einen „Stand“, aus Gründen der Lautstärke idealerweise in einem eigenen Raum, ebenso das NIT und das Scrum Master Team. Im Stil eines Speed-Datings, in Form von 10- bis 15-minütigen Runden, gehen dann der Product Owner, die Teammitglieder und andere Stakeholder von Stand zu Stand. An jedem erhalten sie ein Blitzlicht auf die Sprintziele und deren Bezug zum Nexus Goal. Feedback kommt via selbstbeschrifteter Zettel an die Teams zurück. Der Schwerpunkt der Kurzdarstellung des NIT liegt auf dem Stand der Integration und der erreichten Qualität.

Wir haben festgestellt, dass diese Marktplatz-Variante es erlaubt, in relativ kurzer Zeit das Feedback aller Beteiligten zu sammeln, was uns mit der ursprünglichen Variante nicht gelungen war. Dabei hatten wir versucht, die einzelnen Informationen jeweils frontal vor dem Gesamtpublikum präsentieren zu lassen. Dieses Vorgehen blockiert jedoch eher den direkten Aus-

tausch und das konstruktive Brainstorming, wie sie an den einzelnen Ständen auftreten.

Frage 7: Wie gehen wir mit den ganzen Abhängigkeiten um?

Innerhalb von Nexus nutzen wir jedes Event, aber auch die ständige Zusammenarbeit dazu, Abhängigkeiten zu identifizieren, transparent zu machen und zu lösen. Einige Beispiele dafür, wie wir vorgehen:

- Mit der in Frage 4 vorgestellten Visualisierung am Board gelingt es in den Cross Refinements und den lokalen Refinements, Abhängigkeiten frühzeitig zu identifizieren.
- Das Nexus Daily wird genutzt, um über aktuelle Abhängigkeiten zu sprechen und den Tagesplan auf deren Lösung auszurichten.
- In den Retrospektiven erarbeiten wir Maßnahmen, um Abhängigkeiten noch besser zu lösen oder besser damit umgehen zu können.
- Ideal ist es, wenn es gelingt, alle beteiligten Teams am gleichen Ort zu haben. Denn die Co-Lokation ermöglicht, spontane Fragen direkt und ohne Umwege zu klären. Räumliche Trennungen der Teams dagegen erhöhen den Komplexitätsgrad und reduzieren den notwendigen Wissensaustausch.

Fazit: Nexus, unkomplizierter Rahmen für effektive und effiziente Produktentwicklung im komplexen Umfeld

Im Vergleich zu Scrum eine zusätzliche Rolle, zwei weitere Artefakte und sogar fünf neue oder erweiterte Events, das klingt zugegebenermaßen nicht nach Einfachheit. Dennoch repräsentiert Nexus damit unserer Meinung nach tatsächlich die minimale Struktur, die nötig ist, um die große Herausforderung bei der Skalierung zu managen: Die Abhängigkeiten. Nexus liefert den Rahmen, um unser Ziel zu erreichen. Mit einem Mehr an Teams ein Mehr an Wert zu erzeugen. Es geht nicht darum, mehrere Teams zu organisieren, es geht um echte Skalierung im Sinne von Wertmaximierung. Ziel bleibt ein integrated done Inkrement.

Entscheidend ist, auch innerhalb der Skalierung die agilen Prinzipien zu bewahren, beispielsweise die Selbstorganisation der Teams sowie Bottom up Intelligence zu nutzen. Das gilt auch oder gerade dann, wenn der äußere Druck auf den Nexus wächst. Und da Nexus einen Rahmen vorgibt, keine 1-zu-1-Bauanleitung ist, bleibt „inspect & adapt“ so wichtig. Skalierung ist harte Arbeit. Aber, wie wir meinen, eine Arbeit, die mit Nexus unkomplizierter wird. ||

Die Autoren



Marion Gakstatter

(marion.gakstatter@andrena.de)
ist Agile Coach bei andrena objects und verfügt über mehrjährige Erfahrung als Product Owner. Sie ist Scrum Master und unterstützt Unternehmen beim Etablieren und Optimieren agiler Prozesse. Erfahrungen mit Nexus hat sie bereits in verschiedenen Projekten gesammelt.



Felix Schad

(felix.schad@andrena.de)
ist Agile Coach bei andrena objects und unterstützt Unternehmen dabei, agile Strukturen in der Produktentwicklung auf- und auszubauen. Er ist ein erfahrener Softwareentwickler und teilt sein Wissen sehr gerne in Trainings und als Referent auf Konferenzen.