

---

# WSTĘP

---

Ta książka jest doskonała. Zaczyna się od opisu prostego zastosowania Nexusa. Następnie przechodzimy do jego zastosowań w coraz bardziej złożonych sytuacjach. Autorzy wyjaśniają kolejne komplikacje, powodowane przez nie problemy oraz sposoby zastosowania Nexusa do ich rozwiązania. Osnowę pomysłów stanowi studium przypadku. I opiera się na dokumencie *The Nexus Guide* (Nexus – przewodnik), ostatecznym źródle wiedzy.

Jednak jaka jest przyczyna istnienia Nexusa?

Scrum jest środowiskiem pozwalającym zespołowi osób rozwiązywać złożone problemy w celu tworzenia przyrostów wartości w krótkim czasie. Przez 27 lat Scrum dowiódł swojej wartości w wielu zastosowaniach.

Jednak Scrum został zaprojektowany tylko dla pojedynczego zespołu. Typowe są sytuacje wymagające współpracy wielu zespołów o różnych kompetencjach w celu utworzenia wartości. Organizacje naturalnie dążą do rozbudowania początkowego środowiska Scrum.

Przez lata pracowałem z setkami organizacji, odwołując się do środowiska i wartości Scrum, skalując jego użycie do dziesiątek, setek, a nawet tysięcy osób pracujących razem, aby osiągnąć pożądany wynik. Wielu innych praktyków Scrum również tak postępowało. Stosując naszą wcześniejszą wiedzę, udało się nam w pewnym stopniu zachować produktywność i wartości Scruma.

Na podstawie moich doświadczeń oraz doświadczeń innych osób pracujących ze mną w Scrum.org zdefiniowałem środowisko zaprojektowane do stosowania przez wiele zespołów Scrum podczas pracy nad pojedynczym

produktem lub problemem. Wynikiem jest Nexus, egzoszkielet utrzymujący wiele zespołów scrumowych. Nexus dostarcza informacji i zarządzania informacjami w celu prowadzenia ich prac razem. Zachowywana jest możliwie największa produktywność, opisywane są metody zwiększania produktywności i dołączane są techniki korekcyjne do rozwiązywania niepowodzeń.

Czytaj i dowiedz się więcej. *Scrum on.*

–Ken Schwaber

---

# PRZEDMOWA

---

Nasz cel podczas pisania tej książki był prosty: pokazać osobom, które już znają Scrum, prosty, ale potężny sposób stosowania znanych koncepcji Scrum do produktów, które wymagają wysiłków więcej niż jednego zespołu. Ponad 12 milionów osób używa codziennie Scruma, a praca wielu z nich wymaga dużych wysiłków obejmujących wiele zespołów. Nexus ewoluował, aby spełnić potrzeby tych osób, a chociaż jest używany przez wiele organizacji, do tej pory nie było książki, która go opisywała. Mamy nadzieję, że czytelnicy tej książki będą w stanie zastosować Nexusa w celu skalowania, a przypuszczalnie nawet udoskonalania, ich praktyk Scrum. Zgodnie z powiedzeniem „Skalowalny Scrum to nadal Scrum”.

## DLA KOGO JEST TA KSIĄŻKA

Każdy, kto używa Scruma, skorzysta na czytaniu tej książki, ponieważ w pewnym momencie odkrywamy, że pojedynczy zespół scrumowy już nie wystarcza do dostarczenia produktu. Dodawanie zespołów brzmi łatwo, ale niezarządzone zależności między zespołami szybko przekraczają możliwości podejścia opartego tylko na intuicji. Ta książka pomoże każdemu członkowi zespołu w lepszym zrozumieniu Nexusa. Nie tylko zespoły scrumowe, ale także ich interesariusze uznają tę książkę za przydatną w zrozumieniu wyzwań stojących przed wysiłkami wielu zespołów, co pozwoli im lepiej wspierać zespoły, z którymi pracują.

## JAK ZORGANIZOWANA JEST TA KSIĄŻKA

W tej książce założyliśmy, że Czytelnicy znają już środowisko Scrum, i bazując na tej wiedzy wyjaśniamy, jak skalować Scrum, aby rozwijać duży produkt przy użyciu Nexusa.

Rozdział 1, „Wprowadzenie do skalowalnego programowania zwinnego”, zgodnie ze swoim tytułem wprowadza do programowania zwinnego w kontekstach, które wymagają więcej niż jednego zespołu scrumowego pracującego nad projektem.

Rozdział 2, „Wprowadzenie do Nexusa”, skupia się nad zasadami i koncepcjami leżącymi u podstaw Nexusa, w tym określeniu sytuacji, kiedy Nexus jest potrzebny i co trzeba zrobić, aby rozpocząć.

Rozdział 3, „Formowanie Nexusa”, skupia się nad sposobem formowania Nexusa wokół produktu, nawet jeśli ten produkt jest nadal jedynie pomysłem bez zespołu. Ponadto opiszemy, jak dodawać zespoły podczas tworzenia Nexusa w przypadku istniejących produktów i zespołów. Opiszemy także, jak można organizować zespoły scrumowe w Nexusie oraz rozpoznawać (i minimalizować) zależności backlogu produktu.

Rozdział 4, „Planowanie w Nexusie”, skupia się na organizowaniu pracy w Nexusie: proponowaniu, doskonaleniu i sprawdzaniu dużego backlogu względem zagadnień biznesowych, stawianiu celów oraz planowaniu sprintu.

Rozdział 5, „Przebieg sprintu w Nexusie”, skupia się na pracy z Nexusem podczas sprintu: pracy z backlogiem sprintu Nexusa, przebiegu codziennego Scrumu Nexusa, prowadzenie przeglądów sprintów Nexusa i prowadzenia retrospektywy sprintu Nexusa.

Rozdział 6, „Ewolucja Nexusa”, skupia się na zarządzaniu Nexusem, w tym raportowaniu postępu, poprawie wydajności i przepustowości oraz usuwaniu wąskich gardeł.

Rozdział 7, „Nexus w trybie emergencji”, skupia się na tym, jak Nexus pomaga organizacjom przejść przez typowe wyzwania związane ze skalowaniem, ułatwiając współpracę rozproszonych zespołów i odpowiadając na wyzwania, które odciągają zespoły od skutecznej współpracy.

Rozdział 8, „Retrospektywa na drodze Nexusa”, odzwierciedla typową drogę, którą zespoły i organizacje podejmują, gdy skalują Scrum. Przyglądamy się, jak elementy Nexusa pomagają im w tej drodze, poznajemy typowe wyzwania oraz dowiadujemy się, jak im sprostać. Spoglądamy także w przyszłość, aby poznać, co możemy zrobić, aby kontynuować drogę poprawiania zdolności dostarczania złożonych aplikacji.

Zarejestruj swoją kopię książki *Nexus, czyli skalowalny Scrum* w witrynie InformIT, aby uzyskać wygodny dostęp do aktualizacji i/lub poprawek, kiedy się ukażą. W celu rozpoczęcia procesu rejestracji przejdź do [informit.com/register](http://informit.com/register) i zaloguj się lub utwórz konto. Wprowadź kod ISBN (9780134682662) książki i kliknij przycisk Submit (Prześlij). Spójrz na kartę Registered Products (Zarejestrowane produkty), aby znaleźć łącze Access Bonus Content (Dostęp do dodatkowej zawartości) obok tego produktu i skorzystaj z tego łącza, aby uzyskać dostęp do materiałów dodatkowych. Jeśli chcesz otrzymywać powiadomienia o wyjątkowych ofertach nowych wydań i aktualizacji, zaznacz pole wyboru, aby otrzymywać od nas wiadomości e-mail.

---

# PODZIĘKOWANIA

---

Otrzymaliśmy wiele pomocy i wsparcia podczas pisania tej książki. Po pierwsze chcemy podziękować Kenowi i Christinie Schwaberom za ich wsparcie, zachętę i wgląd w ewoluowanie Nexusa ze środowiska Scrum, a także podziękować Kenowi Schwaberowi i Jeffowi Sutherlandowi za utworzenie samego Scruma, na którym bazuje Nexus. Środowisko Nexus istnieje dzięki wspólnej pracy zespołu osób, które spotkały się, aby przekuć swoje doświadczenia w coś, co może być udostępniane wszystkim w formie dokumentu *The Nexus Guide*.

Jesteśmy także wdzięczni społeczności profesjonalnych trenerów Scrum, której członkowie dzielili się swoim cennym czasem, pomagając poprawić tę książkę dzięki wnikliwym sugestiom i starannym recenzjom. Za obszerny wkład wyrażamy najgłębszą wdzięczność następującym osobom: Peter Götz, Jesse Houwing, Richard Hundhausen, Ralph Jocham, Mikkel Toudal Kristiansen, Rob Maher, Jeronimo Palacios i Steve Porter. Nasze podziękowanie kierujemy także do Erica Naiburga, którego dokładne oko pisarza pomogło nam wyrazić idee prościej i skuteczniej, oraz Sabriny Love, która zaprojektowała naszą okładkę.

W końcu, ta książka nie powstałaby bez wsparcia otrzymanego od wydawnictwa Pearson/Addison-Wesley, a w szczególności redaktora Chrisa Guzikowskiego, redaktora merytorycznego Chrisa Zahna, redaktora technicznego Julie Nahil i adiustatora Stephanie Geels, którzy pomogli nam dopracować i opublikować niniejszą książkę.

–Kurt, Patricia i Dave

---

# O AUTORACH

---

**Kurt Bittner** ma ponad 35 lat doświadczenia w pomaganiu zespołom w dostarczaniu oprogramowania w krótkich cyklach sterowanych informacjami zwrotnymi, jako programista, menedżer produktu i właściciel produktu, jako analityk branżowy i jako organizacyjny agent zmiany. Jest autorem trzech innych książek o inżynierii oprogramowania i wielu blogów oraz artykułów, a także często występuje na konferencjach.

**Patricia Kong** miała kluczowy wkład w środowisko Nexus i środowisko zarządzania Evidence-Based Management. Zajmowała się rozwojem produktu, zarządzaniem produktu i marketingiem, prowadząc wiele początkujących firm w Stanach Zjednoczonych i Europie i pracowała w rozwoju biznesowym i zarządzaniu zaangażowaniem dla Forrester Research. Płynnie posługuje się czterema językami.

**Dave West** jest CEO i właścicielem produktu w Scrum.org. Często prowadzi przewodnie prezentacje na większych konferencjach branżowych i jest powszechnie czytany autorem książek, blogów, artykułów i raportów z badań. Kierował wytwarzaniem produktów, a także był konsultantem organizacji wielonarodowościowych.





---

# WPROWADZENIE DO SKALOWALNEGO PROGRAMOWANIA ZWINNEGO

---

Zwinne (agile) wytwarzanie oprogramowania, używając terminu, który spopularyzował Geoffrey Moore, *przeskoczyło przepaść*<sup>1</sup>. Dzisiaj nikt na poważnie nie rozmawia o tym, czy programowanie zwinne jest właściwe. Dzisiejsze rozmowy skupiają się na tym, kiedy i gdzie je stosować. W dużych organizacjach te rozmowy często sprowadzają się do pytań o skalę: nikt nie kwestionuje, że podejścia zwinne działają dobrze w małym, lokalnym zespole, ale pojawiają się wątpliwości, czy to podejście może służyć do wytwarzania i dostarczania dużych produktów przez wiele zespołów.

Ta książka jest o skalowaniu środowiska Scrum przy użyciu podejścia o nazwie Nexus, rozwiniętego przez jednego z współtwórców Scruma. W miarę przebiegu książki będziemy dyskutować, dlaczego skalowanie jest trudne i jak pokonać jego wyzwania. W tym krótkim rozdziale nakreślimy tło, wyjaśniając, dlaczego programowanie zwinne jest ważne, dlaczego Scrum jest ważny i dlaczego Nexus jest najprostszym i naszym zdaniem najlepszym sposobem skalowania Scruma.

---

1. <https://www.forbes.com/sites/danschawbel/2013/12/17/geoffrey-moore-why-crossing-the-chasm-is-still-relevant/#123c4f95782d>

## DLACZEGO PROGRAMOWANIE ZWINNE?

Praktyki programowania zwinnego są niezbyt nowe. Scrum ma teraz ponad 20 lat<sup>2</sup>, a Manifest Agile został podpisany ponad 15 lat temu<sup>3</sup>.

Nowością jest natomiast to, że oprogramowanie stało się siłą dezorganizującą w każdej branży, a organizacje zwróciły się do podejść zwinnych, aby stosować je do dostarczania innowacyjnych rozwiązań opartych na oprogramowaniu<sup>4</sup>.

Praktyki programowania zwinnego pozwalają zespołom dostarczać więcej wartości biznesowych szybciej dzięki poprawieniu współpracy i użyciu procesu empirycznego do inspekcji, adaptacji i poprawiania wyników biznesowych. W dzisiejszym rywalizacyjnym środowisku biznesowym długoterminowe planowanie i wieloletnie projekty ustąpiły częstym wydanom. Podejście „inspekcja i adaptacja” programowania zwinnego spełnia te oczekiwania.

## DLACZEGO SCRUM?

Zgodnie z Forrester Research, 90% zespołów programowania zwinnego używa środowiska Scrum<sup>5</sup>. Kluczem do tej popularności jest fakt, że Scrum nie jest normatywny, ale jest środowiskiem opartym na zbiorze zasad i wartości, składającym się z trzech ról (właściciel produktu, Scrum master i zespół deweloperski), pięciu wydarzeń (sprint, planowanie sprintu, codzienny Scrum, przegląd sprintu i retrospektywa sprintu) oraz trzech artefaktów (baclog produktu, backlog sprintu i przyrost produktu), dzięki czemu dobrze dostosowuje się do różnych sytuacji<sup>6</sup>.

---

2. <https://kenschwaber.wordpress.com/2015/11/22/scrum-development-kit/>

3. <http://agilemanifesto.org>

4. Inwestor kapitału wysokiego ryzyka Marc Andreessen przedstawił sławną opinię, że „oprogramowanie pożera świat”, odwołanie do narastającej tendencji, że startupy programistyczne pokonują znacznych, bardziej stabilnych, lepiej finansowanych i ogólnie mocnych konkurentów. Zobacz cały artykuł pod adresem <http://www.usj.com/articles/SB10001424053111903480904576512250915629460>.

5. <https://www.forrester.com/How+Can+You+Scale+Your+Agile+Adoption/fulltext/-/E-res110444#AST9629982013>

6. Więcej informacji o Scrumie można przeczytać pod adresem <http://www.scrumguides.org/scrum-guide.html>.

Siła Scruma polega na jego prostocie. Skupia się na pojedynczym zespole wytwarzającym pojedynczy produkt. Ma tylko trzy role: właściciel produktu jest skupiony na celach biznesowych, zespół deweloperski wytwarza produkt, a Scrum master pomaga właścicielowi produktu i zespołowi deweloperskiemu osiągnąć te cele m.in. dzięki kształceniu, coachingowi i ułatwianiu. Choć Scrum jest łatwy do zrozumienia, osiągnięcie biegłości nadal wymaga zgody i poświęcenia, aby przełamać stare nawyki i nabrać nowych.

### **CO TO JEST PRODUKT?**

Wiele organizacji nadal zwyczajowo myśli o projektach.

Projekt jest inicjatywą o skończonych rozmiarach, dobrze zdefiniowanym początku i konkretnym zakresie do dostarczenia, która zwykle ma zdefiniowaną datę zakończenia.

Natomiast produkt jest długotrwały, często nie ma zdefiniowanego zakończenia. Używanie projektów do dostarczania i wspierania produktów skutkuje wieloma problemami, z których nie najmniejszym jest tendencja, aby interesariusze dokładali wymagania w niepewności, kiedy będzie „następne” wydanie. W wielu organizacjach projekty są widziane jako źródło kosztów, a produkty są widziane jako źródło wartości biznesowej. Przekształcanie od rozwoju projektu do rozwoju produktu będzie często zmianą postrzegania tego, co robią zespoły deweloperskie, z zaledwie wspierania biznesu na jego aktywne napędzanie.

Jeśli produkt jest warty rozwoju, musi być finansowany i zarządzany w inny sposób. Produkty wymagają regularnych wydań, aby spełniać stale zmieniające się potrzeby użytkowników produktu. Produkty wymagają dedykowanego zespołu do budowania i wspierania produktu w kolejnych wydaniach i wymagają, aby nie było różnicy między utrzymaniem, rozwijaniem a udoskonalaniem z perspektywy zespołu produktu.

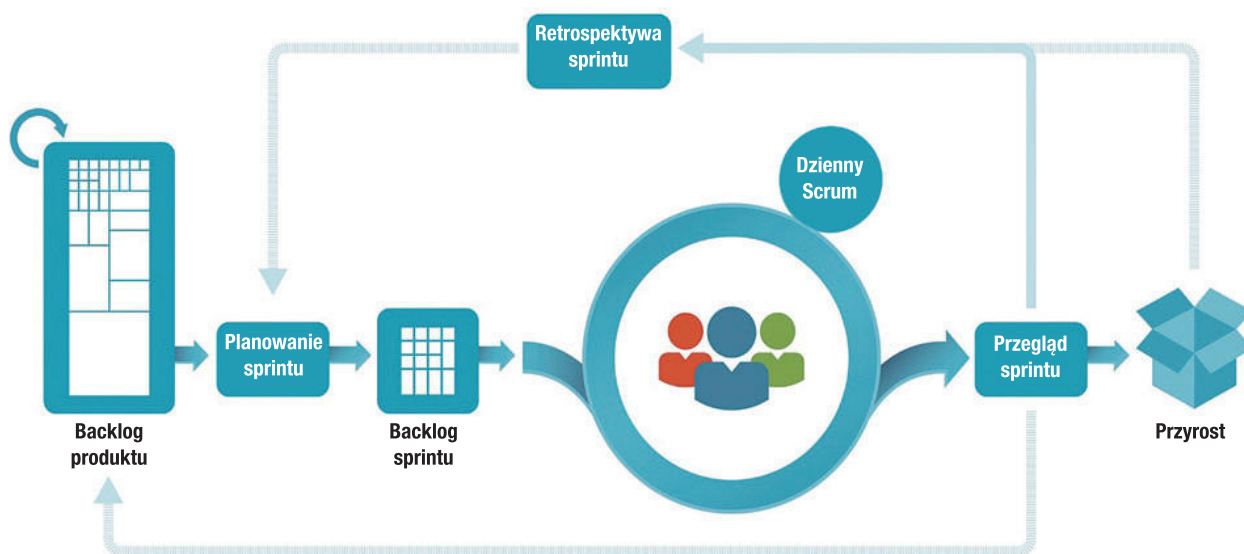
### **CO TO JEST SCRUM?**

Scrum to środowisko (framework), które pomaga zespołom pokonywać złożone problemy adaptacyjne w celu dostarczenia produktu o najwyższej

możliwej wartości. Wiele organizacji z powodzeniem używało Scruma, począwszy od wczesnych lat 1990.

Scrum jest oparty na teorii empirycznego sterowania procesami, czyli empiryzmie. Empiryzm zakłada, że wiedza pochodzi z doświadczenia i dokonywania decyzji na podstawie tego, co jest wiadome. Scrum korzysta z iteracyjnego, przyrostowego podejścia do optymalizacji przewidywalności i sterowania ryzykiem przez ciągłe uczenie się. Trzy filary podtrzymują każdą implementację empirycznego sterowania procesem: przejrzystość, inspekcja i adaptacja. Jest ponad 12 milionów praktyków Scrum i codziennie ta liczba się zwiększa<sup>7</sup>. Choć jest możliwe używanie niektórych technik Scrum w projektach, Scrum zasadniczo skupia się na wytwarzaniu produktu.

Podstawowe elementy środowiska Scrum są przedstawione na rysunku 1-1<sup>8</sup>.



Rysunek 1-1 Środowisko Scrum

Jednak istnieją realistyczne ograniczenia tego, co może osiągnąć pojedynczy zespół scrumowy. Organizacje mogą być skłonne do prostego zwiększenia liczby osób w zespole lub zwiększenia liczby zespołów

7. Scrum.org.

8. Środowisko Scrum jest udokumentowane w publikacji Scrum Guide (Przewodnik po Scrumie), która jest dostępna za darmo pod adresem <http://www.scrumguides.org/>.

zaangażowanych w produkt, aby osiągnąć większą prędkość, ale dekady praktycznego doświadczenia pokazały, że zmniejsza to produktywność<sup>9</sup>.

## DLACZEGO NEXUS?

Niektóre produkty – a niektórzy mogą argumentować, że większość produktów – są zbyt złożone, aby były dostarczane przez pojedynczy zespół scrumowy. Takie przykłady obejmują samochody lub inne fizyczne produkty, które łączą sprzęt i oprogramowanie, lub bardzo skomplikowane produkty oprogramowania, które wymagają skoordynowanych wysiłków wielu zespołów scrumowych do realizacji. Inne produkty są pod presją czasu wprowadzenia na rynek, które wymagają dostarczania większych możliwości w krótkim czasie niż może dostarczyć jeden zespół.

Stojąc przed tymi wyzwaniami, organizacje potrzebują więcej niż jednego zespołu scrumowego do budowania pojedynczego produktu. Wiele zespołów scrumowych pracujących razem, aby budować pojedynczy produkt, często ma problemy z utworzeniem „ukończonej” zintegrowanej pracy w każdym sprincie z powodu dodanej złożoności zależności w skali. Oznaki, że złożoność utrudnia skuteczne dostarczanie, są widoczne w różnych sytuacjach. Na przykład, kiedy poszczególne zespoły prezentują swoje indywidualne przyrosty zamiast przyrostu zintegrowanego produktu podczas przeglądu sprintu. Wtedy, gdy konieczna jest seria „wzmacniających” sprintów do poradzenia sobie ze zakumulowanym długiem technicznym. A także, kiedy potrzeba zespołu integracyjnego, aby połączyć ze sobą prace innych zespołów.

Środowisko Nexus pomaga rozwiązać to, pozwalając organizacjom planować, uruchamiać, skalować większe inicjatywy wytwarzania produktu i zarządzać nimi (szczególnie inicjatywy, które obejmują istotne wytwarzanie oprogramowania) przy użyciu Scruma. Nexus pozwala, aby wiele

---

9. Zostało to opisane w klasycznej pracy Freda Brooksa Mityczny osobomiesiąc: Eseje o inżynierii oprogramowania, w kontekście tradycyjnych projektów programistycznych. Podejście zwinne nie zmienia podstawowego problemu: zwiększenie rozmiaru zespołu powoduje wykładniczy wzrost złożoności komunikacji, który przewyższa zysk produktywności po przekroczeniu praktycznego maksimum 7;2 członków zespołu. Studium przypadku FBI Sentinel również stanowi instruktaż: <http://www.scrumcasestudies.com/fbi/>.

zespołów scrumowych pracowało nad pojedynczym produktem, łącząc się w większą jednostkę nazywaną Nexusem.

Nexus może być uważany jako rodzaj „egzoszkieletu”, który chroni i wzmacnia zespoły scrumowe, upraszczając połączenia i zależności między nimi oraz zarządzając tymi połączeniami i zależnościami, zapewniając przejrzysty wgląd od dołu do góry w to, jak zespoły współpracują ze sobą. Fundamentem Nexusa jest zachęcanie do przejrzystości i komunikacji, aby skalowane było możliwie jak najbardziej jednolite. W ramach Nexusa środowisko Scrum, skalowane do większych bardziej złożonych produktów, nadal pozostaje Scrumem.

## **PROSTOTA JEST KLUCZEM DO SKALOWANIA**

Kluczem do skalowania programowania zwinnego do pracy wśród wielu zespołów jest redukcja lub eliminacja zależności między tymi zespołami. Nexus zapewnia prosty zbiór rozszerzeń do Scruma, aby pomóc zespołom robić właśnie to. W kolejnych rozdziałach opiszemy te rozszerzenia, a także uzupełniające praktyki, które pomagają organizacjom dostarczać lepsze produkty w bardziej skuteczny sposób. Po krótkim opisie Nexusa w następnym rozdziale przejdziemy w pozostałej części książki do badania różnych aspektów Nexusa na podstawie studium przypadku. Zatem bez dalszego wprowadzenia zajmijmy się Nexusem.



---

# WPROWADZENIE DO NEXUSA

---

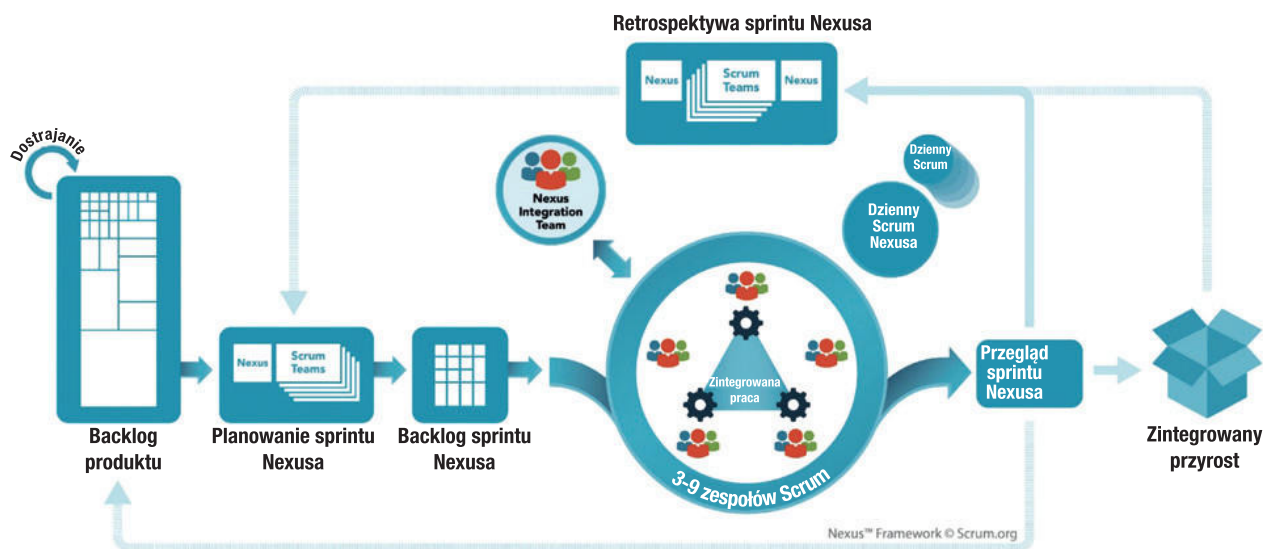
W tym rozdziale opiszemy całościowo środowisko Nexus. Jak pokażemy, Nexus jest względnie małym i prostym rozszerzeniem Scruma. Mamy powiedzenie „Skalowalny Scrum to nadal Scrum”. Sam Scrum jest dość prosty, przynajmniej do zrozumienia. Podczas skalowania jego prostota jest dużą zaletą, ponieważ złożoność jest wrogiem skalowania. Prostota Nexusa także sprawia, że można go dobrze adaptować, co zobaczymy w kolejnych rozdziałach.

## CO TO JEST NEXUS?

Nexus to środowisko (framework), które pozwala wielu zespołom scrumowym wspólnie pracować nad pojedynczym backlogiem produktu, aby dostarczać co najmniej jeden „ukończony” zintegrowany przyrost w każdym sprincie. „Wiele” oznacza typowo od trzech do dziewięciu zespołów scrumowych. Dlaczego nie dwa? Ponieważ dwa zespoły mogą zasadniczo koordynować swoje prace bez dodatkowej struktury. Dlaczego dziewięć? Scrum zaleca ograniczenie zespołów do maksymalnie dziewięciu członków, aby poprawić spójność i zredukować złożoność, podobnie Nexus zaleca taką samą liczbę zespołów. Zupełnie jak w Scrumie ta górna granica również nie jest bezwzględna i lekko większe liczby mogą nadal się sprawdzać, zależnie od okoliczności. W Nexusie odkryliśmy, że złożoność współpracy i koordynacji między zespołami wzrasta znacznie

po przekroczeniu dziewięciu zespołów, wymagając zastosowania trochę innych technik<sup>1</sup>.

Ponieważ Nexus jest oparty na Scrumie, jego części będą znajome dla tych, którzy używali Scruma. Różnica polega na większej uwadze przykładanej do zależności i komunikacji między zespołami scrumowymi (patrz rysunek 2-1).



Rysunek 2-1 Środowisko Nexus, czyli skalowalny Scrum

## NEXUS ROZSZERZA SCRUM

Nexus to Scrum z pewnymi małymi dodatkami (patrz tabela 2-1).

Tabela 2-1 Role, wydarzenia i artefakty Nexusa

Role	Wydarzenia	Artefakty
Zespoły deweloperskie	Sprint	Baclog produktu
Właściciel produktu	Planowanie sprintu Nexusa*	Backlog sprintu Nexusa*

1. Często cytowany dokument George’a Millera, „Magiczna liczba siedem, plus minus dwa”, opisuje ograniczenie w sposobie przetwarzania informacji i formowania wspomnień, które wzmacniają doświadczenie ad hoc w organizacji zespołów: kiedy zespoły rozrastają się powyżej dziewięciu osób, zaczynają tracić spójność, a ich praca staje się trudniejsza do zarządzania. Więcej informacji o Liczbie Millera można znaleźć w Wikipedii [https://en.wikipedia.org/wiki/The\\_Magical\\_Number\\_Seven,\\_Plus\\_or\\_Minus\\_Two](https://en.wikipedia.org/wiki/The_Magical_Number_Seven,_Plus_or_Minus_Two).



Tabela 2-1 Role, wydarzenia i artefakty Nexusa

Role	Wydarzenia	Artefakty
Scrum master	Planowanie sprintu	Backlog sprintu
Zespół integracyjny Nexusa*	Codzienny Scrum Nexusa* Codzienny Scrum Przegląd sprintu Nexusa* Retrospektywa sprintu Nexusa* Retrospektywa sprintu Doskonalenie*	Zintegrowany przyrost

\*specyficzne dla Nexusa

- **Dodatkowy artefakt: backlog sprintu Nexusa.** Backlog sprintu Nexusa to plan Nexusa dla sprintu. Pomaga Nexusowi zrozumieć, nad czym pracują zespoły scrumowe, i zapewnia przejrzystość zależności, które mogą istnieć między zespołami w sprincie.
- **Pięć dodatkowych wydarzeń: doskonalenie, planowanie sprintu Nexusa, codzienny Scrum Nexusa, przegląd sprintu Nexusa oraz retrospektywa sprintu Nexusa.** Te dodatkowe wydarzenia rozszerzają Scrum, aby zapewnić, że praca jest dzielona i koordynowana między zespołami scrumowymi w możliwie najbardziej skuteczny sposób, oraz aby dzielić się doświadczeniami między zespołami w Nexusie.
- **Usunięcie indywidualnego przeglądu sprintu zespołu scrumowego na rzecz przeglądu sprintu Nexusa.** Ponieważ zespoły scrumowe w Nexusie działają razem, aby wytwarzać pojedynczy zintegrowany przyrost, ten zintegrowany przyrost powinien być przeglądany w całości.
- **Nowa rola: zespół integracyjny Nexusa.** Zespół integracyjny Nexusa (NIT) istnieje, aby promować i zapewniać przejrzystą odpowiedzialność za integrację w Nexusie. Prowadzi coaching i udziela wskazówek dotyczących zastosowania Nexusa w zespołach scrumowych, a także wewnątrz organizacji. Zespół NIT składa się z właściciela produktu, Scrum mastera i członków zespołu integracyjnego Nexusa, którzy są zwykle członkami zespołów scrumowych w Nexusie, ale mogą pochodzić z innych obszarów funkcjonalności w organizacji,

takich jak dział operacyjny, zabezpieczenia, architektura lub innych specjalistycznych obszarów, które mogą pomóc Nexusowi dostarczyć zintegrowany przyrost. Ci „zewnątrzni” członkowie mogą tymczasowo dołączać do zespołu NIT na czas, kiedy są potrzebni.

## ZESPÓŁ INTEGRACYJNY NEXUSA

Zespół NIT zapewnia, że zintegrowany przyrost jest wytwarzany przynajmniej w każdym sprincie dla Nexusa. Zespoły scrumowe wykonują pracę. Ostatecznie zespół NIT jest odpowiedzialny za maksymalizację wartości zintegrowanego produktu (patrz rysunek 2-2). Ich aktywności mogą obejmować rozwój narzędzi i praktyk ułatwiających integrację lub coaching i konsultacje ułatwiające koordynację.

Członkowie zespołu NIT powinni mieć podejście nauczycielskie, aby pomóc zespołom scrumowym rozwiązywać ich problemy, kiedy jest to potrzebne. Ich rolą jest pomóc wyróżnić problemy, które mogą wymagać rozwiązania, oraz pomóc zespołom scrumowym rozwiązać te problemy. Tylko w razie emergencji zespół NIT wkracza i rozwiązuje problemy bezpośrednio.

Skład zespołu integracyjnego Nexusa:

- **Właściciel produktu** ostatecznie odpowiada za sukces produktu, którego jest właścicielem. W kontekście zespołu NIT właściciel produktu ponosi odpowiedzialność za to, aby Nexus dostarczał maksymalną wartość podczas każdego sprintu. Rola właściciela produktu nie zmienia się w porównaniu ze Scrumem. Po prostu zakres prac jest bardziej złożony.
- **Scrum master** jest ogólnie zobowiązany do zapewniania, że środowisko Nexus jest przyjęte i zrozumiane. Ten Scrum master jest często Scrum masterem w jednym lub wielu innych zespołach scrumowych w Nexusie.
- **Zespół deweloperski** jest zwykle złożony z członków zespołów scrumowych w Nexusie.