SCRUM GUIDE

2020

Ken Schwaber
Jeff Sutherland

KOMENTARZE

Andy Brandt
Rafał Markowicz

**Przewodnik po Scrumie
z komentarzem doświadczonych trenerów
Professional Scrum Trainer**

CodeSprinters
AGILE EXPERTS

Informacje prawne

Scrum Guide: ©2020 Ken Schwaber and Jeff Sutherland. 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 Scrum 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.

Oryginalny Scrum Guide można pobrać tutaj: https://www.scrumguides.org/

Komentarz do Scrum Guide: © 2020 Code Sprinters sp. z o.o. 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/.

Scrum Guide z komentarzem można pobrać tutaj: https://www.scrumguide2020.pl/

Wersja komentarza 1.0.

Wstęp od komentatorów

Niniejsza książeczka to nasz autorski komentarz do Scrum Guide. Komentarze, jakie zamieszczamy poniżej, dotyczą wszystkich aspektów metody, a nie tylko zmian, które wprowadziła najnowsza wersja Scrum Guide, wydana w listopadzie 2020. Chcemy w ten sposób ułatwić Czytelnikom zrozumienie podstaw Scruma i odpowiedzieć na pytania, które zadawane są najczęściej.

Począwszy od następnej strony, tekst na białym tle to oryginalny, nie tłumaczony i nie modyfikowany Scrum Guide w języku angielskim. W niektórych miejscach w tekście tym są oznaczenia kolorem – pochodzą one również od nas, w ten sposób chcieliśmy zwrócić uwagę Czytelników na niektóre sformułowania.

Nasze komentarze umieszczone są w ramkach, zawsze poniżej tekstu, do którego się odnoszą.

Więcej o nas możesz przeczytać tutaj:

- https://codesprinters.pl/trainer/andy-brandt/
- https://codesprinters.pl/trainer/rafal-markowicz/

Możesz także spotkać się z nami na szkoleniach Code Sprinters. Więcej informacji znajdziesz tutaj: https://codesprinters.pl/classes/

Mamy nadzieję, że nasze komentarze pomogą Ci lepiej zrozumieć Scrum Guide i w efekcie lepiej korzystać ze Scruma. Jeśli chcesz podzielić się z nami Twoimi uwagami lub sugestiami, napisz do nas.

Chcesz podzielić się tą książeczką ze znajomymi i współpracownikami? Przekaż im link do strony, na której można ją pobrać: https://scrumguide2020.pl/

Andy i Rafał

Listopad 2020

The Scrum Guide

The Definitive Guide to Scrum: The Rules of the Game

Osoby zaznajomione ze Scrumem już wcześniej zauważą, że w najnowszej wersji Scrum Guide uległ uproszczeniu i uporządkowaniu, ale sam Scrum nie zmienił się. Wciąż jest frameworkiem, który stanowi szkielet procesu empirycznego, dzięki któremu możliwe jest rozwiązywanie złożonych problemów w zmiennym i nieprzewidywalnym środowisku w sposób przynoszący korzyści.

Zasadnicze zmiany, które wprowadza Scrum Guide 2020 to:

- *Rezygnacja z ról i zastąpienie ich definicjami odpowiedzialności osób, które w Zespole Scrumowym są Product Ownerem, Developerami i Scrum Masterem.*
- *Rezygnacja z konceptu Zespołu Developerskiego, dzięki czemu Product Owner, Scrum Master i Developerzy są teraz członkami jednego Zespołu.*
- *Jednoznaczne określenie odpowiedzialności Scrum Mastera za efektywność Zespołu Scrumowego.*
- *Doprecyzowanie definicji artefaktów tak, aby z każdym z nich wiązało się konkretne zobowiązanie Zespołu Scrumowego; dla Backlogu Produktu jest to Cel Produktu, dla Backlogu Sprintu jest to Cel Sprintu, a dla Przyrostu jest to Definition of Done.*
- *Uproszczenie opisów wszystkich elementów Scruma, w tym usunięcie sugestii jak powinny być realizowane poszczególne zdarzenia oraz bezpośrednich referencji do specyficznych praktyk związanych z wytwarzaniem oprogramowania.*

Zmiany te, choć istotne, nie są fundamentalne. W naszej ocenie ten Scrum Guide pozwala lepiej zauważyć różnicę między tym, co jest Scrumem, a co jego wypaczeniem. Przykładem może być jednoznaczny wymóg, by w Zespole Scrumowym był Scrum Master. Innym przykładem jest jasno określona odpowiedzialność członków Zespołu Scrum za wymaganie od innych, by wykonywali to, do czego się zobowiązali. Jeszcze innym jest wyklarowanie, iż Przyrost musi powstać przed Przeglądem Sprintu, a Product Owner odpowiada nie tylko za to, co jest w Backlogu Produktu, ale też za wartość każdego powstałego Przyrostu.

Uważna lektura Scrum Guide powinna skutkować zrozumieniem, że uzyskanie korzyści za pomocą Scruma jest możliwe tylko dzięki wysiłkowi, zaangażowaniu i przede wszystkim odpowiedzialności osób pracujących w Zespole Scrumowym. Inaczej rzecz ujmując, jeśli ktoś liczy, że niskim kosztem „wdroży Scrum" i będzie z tego czerpał zyski, srodze się zawiedzie. Z drugiej strony wysiłek włożony w to, by faktycznie użyć Scruma w sposób świadomy (a nie „robić Scruma" lub „wdrożyć" tę metodę), pozwoli nie tylko uzyskać wartościowy produkt, ale także udoskonali Zespół i organizację, która Zespół ten powołała.

Purpose of the Scrum Guide

We developed Scrum in the early 1990s. We wrote the first version of the Scrum Guide in 2010 to help people worldwide understand Scrum. We have evolved the Guide since then through small, functional updates. Together, we stand behind it.

The Scrum Guide contains the definition of Scrum. Each element of the framework serves a specific purpose that is essential to the overall value and results realized with Scrum. Changing the core design or ideas of Scrum, leaving out elements, or not following the rules of Scrum, covers up problems and limits the benefits of Scrum, potentially even rendering it useless.

Elementów Scruma nie jest wiele, ale są tak dobrane, że ciężko jest usunąć któryś z nich bez szkody dla całości. Co istotne, wszystkie służą zapewnieniu empirycznej kontroli procesu, więc niezwykle trudno usunąć którykolwiek z nich bez obniżenia (lub uniemożliwienia) możliwości empirycznego działania. W praktyce zamiast usuwać coś ze Scruma, uzupełnia się go różnymi praktykami, które stanowią implementację elementów Scruma.

We follow the growing use of Scrum within an ever-growing complex world. We are humbled to see Scrum being adopted in many domains holding essentially complex work, beyond software product development where Scrum has its roots. As Scrum's use spreads, developers, researchers, analysts, scientists, and other specialists do the work. We use the word "developers" in Scrum not to exclude, but to simplify. If you get value from Scrum, consider yourself included.

Choć Scrum powstał, aby lepiej zarządzać wytwarzaniem oprogramowania, to ostatnio trendem jest wychodzenie poza tę domenę. Scrum pojawia się więc w innych dziedzinach. W powyższym akapicie twórcy Scruma podkreślają, że widzą i rozumieją ten trend. W takiej sytuacji pod pojęciem „Developer" należy rozumieć wszystkich, którzy wkładają pracę w produkt.

As Scrum is being used, patterns, processes, and insights that fit the Scrum framework as described in this document, may be found, applied and devised. Their description is beyond the purpose of the Scrum Guide because they are context sensitive and differ widely between Scrum uses. Such tactics for using within the Scrum framework vary widely and are described elsewhere.

*Twórcy podkreślają tutaj, że Scrum jest tylko ramą wokół innych praktyk, wzorców i sposobów działania. Te są kontekstowe – to znaczy mogą działać w jednej firmie czy zespole, a w innych okolicznościach już nie. Z tego powodu ich opis nie jest zawarty w Scrum Guide tylko „gdzie indziej" – czyli w szerokiej już obecnie literaturze przedmiotu, książkach i blogach, a także na szkoleniach (takimi szkoleniami koncentrującymi się na tym właśnie temacie są nasze **Scrum Master Toolbox™** i **Product Owner Toolbox™**).*

Siła Scruma – i „tajemnica" jego popularności – leży m.in. w tym właśnie, że celowo Scrum nie próbuje obejmować wszystkiego, więc i Scrum Guide nie próbuje być księgą opisującą wszystko – począwszy od praktyk w zespole, po sposób zarządzania całą organizacją.

Ken Schwaber & Jeff Sutherland November 2020

Wstęp od komentatorów 1
Purpose of the Scrum Guide 2
Scrum Definition 5
Scrum Theory 6
Transparency 7
Inspection 8
Adaptation 8
Scrum Values 9
Scrum Team 9
Developers 11
Product Owner 12
Scrum Master 13
Scrum Events 15
The Sprint 16
Sprint Planning 19
Daily Scrum 22
Sprint Review 23
Sprint Retrospective 24
Scrum Artifacts 26
Product Backlog 27
Commitment: Product Goal 28
Sprint Backlog 30
Commitment: Sprint Goal 30
Increment 31
Commitment: Definition of Done 32
End Note 34
Acknowledgements 35
People 35
Scrum Guide History 35

Scrum Definition

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

> *Scrum jest skoncentrowany na tworzeniu („generowaniu") **wartości** dla organizacji poprzez produkty, a więc rozwiązanie złożonych problemów.*
>
> *Zdanie to podkreśla również fakt, że Scrum stosujemy tam, gdzie mamy do czynienia ze złożonymi problemami. Przy czym użyte słowo „complex" oznacza tutaj problemy złożone w sposób wymagający zastosowania empiryzmu – a więc takie, w których mierzymy się ze zmiennością i nieprzewidywalnością. Po polsku na taki rodzaj problemu mówimy „niestabilna złożoność".*

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

> *Scrum Guide nie pozostawia żadnych wątpliwości, że Scrum Master musi być. Innymi słowy, gdy czytamy Scrum Guide, już na pierwszej stronie poświęconej samej metodzie dowiadujemy się, że Scrum Master jest w Scrumie **wymagany**.*
>
> *Jednocześnie – o czym jest mowa szerzej w dalszej części – Scrum Master nie jest rolą, ale odpowiedzialnością w Scrum Team.*
>
> *Podsumowując: ktoś w Scrum Team musi być Scrum Masterem. Przy czym to nie sam Scrum Master ma stworzyć środowisko o cechach opisanych poniżej, ale odpowiada za to, by powstało. Skoro jest odpowiedzialność, to ktoś musi wziąć ją na siebie i być Scrum Masterem.*

1. A Product Owner orders the work for a complex problem into a Product Backlog.
2. The Scrum Team turns a selection of the work into an Increment of value during a Sprint.
3. The Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.
4. *Repeat*

> *Zwracamy uwagę na to, że to cały Zespół Scrumowy wytwarza Przyrost, który jest nośnikiem wartości. Product Owner nie może więc funkcjonować jak „klient" Developerów „zlecający" im realizację elementów Backlogu Produktu. Ponieważ Product Owner to jedna z odpowiedzialności w Scrum Team, zatem osoba, którą ją niesie, odpowiada także – wraz z zresztą zespołu – za wytworzenie w Sprincie wartościowego Przyrostu.*
>
> *Te cztery punkty stanowią zarazem proste podsumowanie metody, w którym od razu zwraca uwagę jej cykliczność. Scrum to seria Sprintów, w czasie których Zespół Scrum tworzy kolejne wartościowe Przyrosty, pracując nad rzeczami z Backlogu Produktu (te rzeczy stanowią definicję „pracy do wykonania").*

Scrum is simple. Try it as is and determine if its philosophy, theory, and structure help to achieve goals and create value. The Scrum framework is purposefully incomplete, only defining the parts required to implement Scrum theory. Scrum is built upon by the collective intelligence of the people using it. Rather than provide people with detailed instructions, the rules of Scrum guide their relationships and interactions.

Scrum – o czym już wspominaliśmy w komentarzu do wstępu – celowo nie jest metodyką totalną, opisującą wszystko, od sposobu pracy pojedynczego developera, do zarządzania organizacją. Scrum jest zestawem reguł dla jednego niewielkiego zespołu, pracującego nad jednym produktem, które ułatwiają mu współpracę; pomagają zarówno tworzyć lepszy produkt jak i udoskonalać swój sposób pracy. Stąd nazwa „framework" – rama nie decyduje, co będzie na obrazie, ale utrzymuje go na miejscu.

Zwracamy uwagę, że Ken i Jeff nie namawiają do stosowania Scruma. Przeciwnie: proponują, aby go spróbować („try"), by sprawdzić, czy pomoże on w tworzeniu wartości i osiąganiu celów. Ma to głęboki sens. Scrum nie jest „panaceum" na wszystko. Do niektórych rodzajów pracy i wykonujących ją zespołów po prostu nie pasuje. Czasami jest to oczywiste, czasami warto po prostu empirycznie się przekonać. Implikuje to oczywiście konieczność mierzenia wartości oraz stopnia osiągania celów – tylko wówczas możemy porównać i sprawdzić, czy wprowadzenie Scruma rzeczywiście przyniosło korzyści.

Various processes, techniques and methods can be employed within the framework. Scrum wraps around existing practices or renders them unnecessary. Scrum makes visible the relative efficacy of current management, environment, and work techniques, so that improvements can be made.

Istniejące praktyki albo mogą zostać „objęte" Scrumem, albo stają się zbędne, kiedy zaczynamy go stosować.

Jednocześnie Scrum czyni bardzo widocznym, czy obecny sposób pracy jest skuteczny. Dzieje się tak dlatego, że metoda wymaga przejrzystości w takiej choćby prostej kwestii jak to, czy zespół przynajmniej raz na Sprint, na jego koniec, jest w stanie wyprodukować nową wersję produktu (Przyrost). I to taki produkt, który jest ukończony (Done) i nadaje się do użytkowania bez żadnych dalszych poprawek czy testów. Jeśli tak nie jest z przyczyn „obiektywnych" (brak umiejętności, brak narzędzi, zła organizacja pracy – w tym nieprawidłowy jej podział np. przez zły podział na projekty), to próba pracy w Scrumie obnaży to, dając szansę na poprawę.

Warto przy tym pamiętać, że to nie Scrum powoduje problemy, które stają się widoczne, gdy zastosujemy tę metodę – problemy te musiały istnieć wcześniej, i dopiero dążenie do wytworzenia działającego i wartościowego produktu w relatywnie krótkich Sprintach czyni widocznym, na ile potrafimy to robić.

Scrum Theory

Scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Bardzo wiele osób stawia Scrum w kontrze do metod wywodzących się z Leana, nawet jeśli są to metody zwinne takie jak np. Kanban. Tymczasem Agile i Lean przenikają się i w wielu aspektach uzupełniają. W szczególności mechanizmy Scruma, które służą określeniu wartości i skupieniu na niej, są odzwierciedleniem leanowego ograniczenia marnotrawstwa, jakim niewątpliwie byłoby robienie rzeczy niepotrzebnych lub mało wartościowych.

Dlatego poszukując praktyk, którymi wspierać będziemy się w Scrumie, warto sięgnąć po narzędzia związane z Leanem, na przykład tak, jak proponuje to ***Kanban Guide for Scrum Teams****.*

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed.

Scrum nie czyni niestabilnej złożoności bardziej stabilną lub mniej złożoną, ale dzięki empiryzmowi pozwala lepiej sobie z nią radzić. Iteracyjność i przyrostowe wytwarzanie wartości mają w tym pomóc, dzielą bowiem złożoność na mniejsze fragmenty, nad którymi pracujemy w czasie na tyle krótkim, że mają one szansę pozostać stabilne.

Przy czym sam fakt pracy iteracyjnej lub budowania rozwiązań inkrementalnie niewiele zmieni, jeśli brak w tym empiryzmu. Empiryzm to sterowanie działaniami (tu procesem budowy kolejnych Przyrostów produktu) w oparciu o wiedzę na temat obecnego stanu faktycznego całego przedsięwzięcia. Każda taka iteracja i krok w rozwoju produktu musi zatem kończyć się sprawdzeniem, jak jest teraz - bo z tego wynika, co powinno być robione dalej (kolejna iteracja / krok). W skrócie: Inspect&Adapt.

Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation.

Każde z czterech zdarzeń w Scrum jest w nim po to, aby zapewnić empiryzm – każde jest okazją do inspekcji adaptacji. Przy czym mowa jest o zdarzeniach, a nie spotkaniach (choć zdarzenia te najczęściej mają formę spotkań właśnie) aby podkreślić, że są one częścią pracy, integralną częścią procesu, niezbędną do jego działania.

W trzech akapitach powyżej zawarta jest „esencja" Scrum. *Warto przeczytać je z uwagą i zastanowić się nad ich znaczeniem. Współcześnie niestety dość często spotykamy się z tym, że mówi się o empiryzmie albo Lean bez zrozumienia tych terminów i ich praktycznego przełożenia na rzeczywistość.*

Transparency

The emergent process and work must be visible to those performing the work as well as those receiving the work. With Scrum, important decisions are based on the perceived state of its three formal artifacts. Artifacts that have low transparency can lead to decisions that diminish value and increase risk.

Artefakty stanowią podstawę sprawdzenia bieżącego stanu – ustalenia, jak jest naprawdę. Przejrzystość tych artefaktów oznacza, że pozwalają one ustalić stan faktyczny wszystkiego, co ma wpływ na podejmowane decyzje. Przejrzystość ta nie powinna jednocześnie być interpretowana jako konieczność informowania wszystkich o wszystkim – nadmiar nieistotnych informacji utrudni wyłowienie tych, które mają znaczenie. Przejrzystość oznacza tu, że kiedy ktoś sięga po informację to otrzymuje informację prawdziwą, opisującą stan faktyczny a nie np. „rzeczywistość podmalowaną" tak, żeby wyglądała ładniej.

Scrum stosowany jest do rozwiązywania problemów interesariuszy, użytkowników produktu, organizacji. Aby możliwe było ustalenie, na ile to, co wytwarza Zespół Scrumowy, jest wartościowe, artefakty powinny być przejrzyste również dla osób, na rzecz których Zespół ten pracuje. Dotyczy to zwłaszcza Backlogu Produktu oraz Przyrostu.

Transparency enables inspection. Inspection without transparency is misleading and wasteful.

Sprawdzanie, którego wyniki są zafałszowane (właśnie „podmalowane" żeby było ładniej, żeby się interesariusze nie denerwowali itp.) prowadzi do podejmowania błędnych decyzji. Rzecz pozornie oczywista, ale jednak, mimo dużego wysiłku, by uczynić Scrum Guide maksymalnie zwięzłym, te dwa zdania są wciąż w nim obecne.

Inspection

The Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently to detect potentially undesirable variances or problems. To help with inspection, Scrum provides cadence in the form of its five events.

Cadence, czyli rytm. Scrum zapewnia rytm – rytm sprawdzeń (inspekcji), a co za tym idzie dostosowań (adaptacji). Stały rytm jest ważny zarówno dla zapewnienia szybkości reakcji, jak i dla uczynienia z tego stałego, nawykowego elementu pracy. Pozwala też dostrzegać i analizować trendy, wychwytywać powtarzające się wzorce (np. sekwencje zdarzeń prowadzących do problemów lub spadku przejrzystości).

Inspection enables adaptation. Inspection without adaptation is considered pointless. Scrum events are designed to provoke change.

Zdarzenia w Scrumie zdefiniowane są tak, by zapewnić przestrzeń do sprawdzenia stanu artefaktów / zadbania o ich przejrzystość. Celem nie jest „zaraportowanie stanu spraw", ale dostosowanie procesu / narzędzi / działań itd., jeśli sprawdzenie ujawni rozbieżność między stanem oczekiwanym, a stanem bieżącym. Jeśli którekolwiek ze zdarzeń odbywa się bez intencji takiego dostosowania, prawdopodobnie marnujemy na nim tylko czas.

Adaptation

If any aspects of a process deviate outside acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted. The adjustment must be made as soon as possible to minimize further deviation.

Adaptation becomes more difficult when the people involved are not empowered or self-managing. A Scrum Team is expected to adapt the moment it learns anything new through inspection.

Powyżej Scrum Guide podkreśla, że zespół powinien reagować natychmiast, gdy tylko w efekcie sprawdzenia okazuje się, że taka reakcja (zmiana) jest konieczna. Innymi słowy, adaptacji nie podlega moment, w którym się czegoś dowiadujemy (a tak można z rozpędu zrozumieć ten zapis) – chodzi o to, że adaptacja ma następować bez zbędnej zwłoki po tym, kiedy inspekcja pokaże, że jakaś reakcja jest konieczna.

Ta niezwłoczność podjęcia działań dostosowujących umożliwia kierowanie się przy ich wykonywaniu tym, co ujawniło sprawdzenie. Nadmierna zwłoka spowodować mogłaby natomiast, że stan spraw uległby zmianie, a zatem i działania oparte na nieaktualnych już danych mogłyby okazać się niewłaściwe.

Odkładanie dostosowania może w szczególności spowodować, że będzie ono nieskuteczne, bo sprawy całkowicie wymkną się spod kontroli, albo jej utrzymanie i przywrócenie spraw do pożądanego stanu będzie bardzo kosztowne.

Stąd płynie wniosek, że częstotliwość sprawdzenia musi być skorelowana z poziomem i skalą zmienności tego, co sprawdzamy. Im ta zmienność większa albo im większe zmiany mogą wystąpić, tym częściej trzeba dokonywać sprawdzenia, aby nadążyć z dostosowywaniem się.

Przykładem doboru częstotliwości sprawdzenia jest długość Sprintu. Im bardziej zmienne i nieprzewidywalne są potrzeby interesariuszy (wynikające ze zmienności i nieprzewidywalności ich otoczenia), tym krótszy musi być Sprint – żeby ograniczyć ryzyko budowania rzeczy już niepotrzebnych. Z drugiej strony Sprint nie może być zbyt krótki, bo wtedy braknie czasu na wytworzenie czegokolwiek, co można by poddać sprawdzeniu.

Scrum Values

Successful use of Scrum depends on people becoming more proficient in living five values:

Commitment, Focus, Openness, Respect, and Courage

The Scrum Team commits to achieving its goals and to supporting each other. Their primary focus is on the work of the Sprint to make the best possible progress toward these goals. The Scrum Team and its stakeholders are open about the work and the challenges. Scrum Team members respect each other to be capable, independent people, and are respected as such by the people with whom they work. The Scrum Team members have the courage to do the right thing, to work on tough problems.

> *Innymi słowy Scrum to narzędzie dla **zaangażowanych** profesjonalistów, którzy **szanując** zarówno siebie nawzajem jak i swoich interesariuszy **skupiają się** na celu i dążą do niego w sposób empiryczny, mając **otwartość** i **odwagę,** by opierać swoje działania i decyzje na rzeczywistym stanie produktu, zespołu – i całego przedsięwzięcia.*

These values give direction to the Scrum Team with regard to their work, actions, and behavior. The decisions that are made, the steps taken, and the way Scrum is used should reinforce these values, not diminish or undermine them. The Scrum Team members learn and explore the values as they work with the Scrum events and artifacts. When these values are embodied by the Scrum Team and the people they work with, the empirical Scrum pillars of transparency, inspection, and adaptation come to life building trust.

> *Scrum wymaga kultury **odwagi i zaufania**: działa najlepiej w środowisku, w którym jest dość odwagi, by zapewnić przejrzystość niezbędną do działania empiryzmu, czyli mówić sobie prawdę – i zaufania, że można to bezpiecznie zrobić. Jako minimum kultura ta powinna być obecna przynajmniej wewnątrz Zespołu Scrumowego. Jednak, żeby Scrum działał naprawdę dobrze i przynosił korzyści, kultura taka – i wartości Scrum – powinny być rzeczywiście obecne w organizacji, a przynajmniej w bezpośrednim otoczeniu Zespołu Scrum.*
>
> *Zaufanie nie powstaje w sposób deklaratywny i nie da się nakazać go komukolwiek. Wartości Scrum muszą być obecne przynajmniej w Zespole, a najlepiej i w jego otoczeniu właśnie po to, by zaufanie mogło tam się pojawić.*

Scrum Team

The fundamental unit of Scrum is a small team of people, a Scrum Team. The Scrum Team consists of one Scrum Master, one Product Owner, and Developers. Within a Scrum Team, there are no sub-teams or hierarchies. It is a cohesive unit of professionals focused on one objective at a time, the Product Goal.

> *Zwracamy uwagę, że w końcu od 2020 roku w Scrum Guide mowa jest nie o rolach (które można „pełnić") ale odpowiedzialnościach, które spoczywają na konkretnych osobach. A więc ktoś **jest** Product Ownerem (a nie pełni rolę Product Ownera), ktoś inny **jest** Scrum Masterem (a nie pełni rolę Scrum Mastera), zaś ci, którzy wytwarzają wartość, **są** Developerami (a nie pełnią taką rolę).*
>
> *Jednocześnie podkreśla się, że osoby mające te trzy odpowiedzialności tworzą jeden zespół. I ten zespół – Zespół Scrum – **wspólnie odpowiada** za wytworzenie co Sprint kolejnego wartościowego Przyrostu.*

Scrum Teams are cross-functional, meaning the members have all the skills necessary to create value each Sprint. They are also self-managing, meaning they internally decide who does what, when, and how.

> *We wcześniejszych wersjach Scrum Guide definiowany był Zespół Developerski, który organizował swoją pracę w ramach Sprintu, ale w ograniczonym stopniu decydował o tym, co będzie robił; określał, ile elementów Backlogu Produktu zrealizuje, ale o ich kolejności decydował Product Owner, nie będący częścią Zespołu Developerskiego.*
>
> *Scrum definiuje obecnie tylko jeden Zespół Scrum, którego częścią jest Product Owner. Zespół ten ma więc możliwości decydowania nie tylko o tym jak i ile elementów Backlogu Produktu zrealizuje, ale też o ich kolejności (oczywiście odpowiedzialność za zawartość i kolejność Backlogu Produktu spoczywa nadal na Product Ownerze).*
>
> *W ten sposób czytelniejsze staje się, że np. w czasie Planowania Sprintu cały Zespół może uzgodnić, że zmieniona zostanie kolejność czy zawartość Backlogu Produktu.*
>
> *Zmiana „samoorganizacji" na „samorządzanie" wydaje się czysto semantyczna, ale jest istotna. Termin „samoorganizacja" odnosi się w literaturze do zespołów, które decydują o swojej strukturze (czasami określa się je również jako „self-designing") – a więc to Zespoły, które np. decydują o tym, kto i na jakich warunkach będzie w nich pracował. Scrum nie wymaga aż takiego poziomu autonomii Zespołu.*
>
> *Twórcy Scruma definiują więc Zespół jako „samozarządzający", bo członkowie Zespołu zarządzają swoją pracą w ramach i granicach, które zdefiniowała organizacja powołująca Zespół do życia.*

The Scrum Team is small enough to remain nimble and large enough to complete significant work within a Sprint, typically 10 or fewer people. In general, we have found that smaller teams communicate better and are more productive. If Scrum Teams become too large, they should consider reorganizing into multiple cohesive Scrum Teams, each focused on the same product. Therefore, they should share the same Product Goal, Product Backlog, and Product Owner.

> *Wielkość zespołu nigdy nie była sztywno zdefiniowana, ale obecny zapis czyni to bardzo czytelnym. To, czy zespół jest za duży lub za mały, odkryć powinniśmy empirycznie – typowy rozmiar 10 lub mniej osób (uwaga: w tej liczbie mieści się również Scrum Master i Product Owner) jest jedynie wskazówką.*
>
> *Czytelniej niż do tej pory określono, że w przypadku, gdy produkt rozwijany jest przez wiele Zespołów Scrumowych, wszystkie one posługują się jednym Backlogiem Produktu (a więc mają jeden Cel Produktu), i we wszystkich nich Product Ownerem jest ta sama osoba.*

The Scrum Team is responsible for all product-related activities from stakeholder collaboration, verification, maintenance, operation, experimentation, research and development, and anything else that might be required. They are structured and empowered by the organization to manage their own work. Working in Sprints at a sustainable pace improves the Scrum Team's focus and consistency.

To bardzo ważny fragment, którzy należy czytać mając z tyłu głowy, że w Scrumie nie ma ról, ale są odpowiedzialności (w domyśle: konkretnych osób, które je niosą). Odpowiedzialności rozumiane nie jako „robienie czegoś osobiście", ale jako „zapewnienie, że zostało zrobione".

Zespół musi wykonywać mnóstwo działań, które pozwolą mu zbudować wartościowy produkt (Przyrost) i przygotować się do kolejnego Sprintu. To wymaga różnych kompetencji, a o tym co, kto i jak zrobi, decydować powinien Zespół (dlatego mówimy, że jest samozarządzający).

A skoro tak, niekoniecznie tylko Product Owner będzie rozmawiał z interesariuszami i niekoniecznie tylko Scrum Master będzie dbał o proces.

*Warto też zwrócić uwagę na koncept **sustainable pace** – jest to najwyższe tempo działań, jakie Zespół może utrzymywać w sposób długotrwały. Scrum wprowadza Sprinty (iteracje) i stały rytm także po to, by uniknąć pracy „od zrywu do zrywu", i by dać możliwość równomiernego obciążenia pracą w sposób, który nie jest niszczący ani dla ludzi, ani dla produktu, ani dla organizacji.*

The entire Scrum Team is accountable for creating a valuable, useful Increment every Sprint. Scrum defines three specific accountabilities within the Scrum Team: the Developers, the Product Owner, and the Scrum Master.

Scruma używa się po to, by uzyskać wartość przewyższającą poniesione koszty (to narzędzie, a nie cel sam w sobie). Odpowiedzialność za to spoczywa na całym Zespole Scrum. Product Owner, Scrum Master i Developerzy mają wspólnie znaleźć sposób na to, by skutecznie wybierać, co jest wartościowe i wytworzyć tę wartość. Innymi słowy, to cały Zespół Scrum odpowiada za to, by co Sprint powstał wartościowy i używalny (czyli ukończony, technicznie kompletny) Przyrost. Nie ma więc miejsca w Scrumie na postawę „to nie mój problem" niezależnie od tego, jaka odpowiedzialność spoczywa na konkretnej osobie.

Ważną odpowiedzialność ma tu Scrum Master, który w szczególności musi nauczyć Zespół jak używać Scruma (uwaga: nie nauczyć teorii, ale praktycznego użycia tej metody).

Raz jeszcze podkreślamy, że w Scrumie już nie ma ról – są odpowiedzialności i konkretne osoby, na których one spoczywają (konkretne osoby, które na siebie te odpowiedzialności przyjęły).

Developers

Developers are the people in the Scrum Team that are committed to creating any aspect of a usable Increment each Sprint.

Innymi słowy – Developerzy to osoby, które wkładają konkretną pracę w produkt (Przyrost), kreując jego dowolny aspekt (np. dokumentację, testy, kod itp.).

Jeśli Product Owner lub Scrum Master wykonują pracę związaną z wytworzeniem Przyrostu, to są również Developerami (te konkretne osoby mają jednocześnie więcej niż jedną odpowiedzialność w Zespole).

The specific skills needed by the Developers are often broad and will vary with the domain of work. However, the Developers are always accountable for:

- Creating a plan for the Sprint, the Sprint Backlog;
- Instilling quality by adhering to a Definition of Done;
- Adapting their plan each day toward the Sprint Goal; and,
- Holding each other accountable as professionals.

Bardzo ważne stwierdzenie: Developerzy mają obowiązek wymagać od innych profesjonalizmu i wywiązywania się z odpowiedzialności. To oznacza, że widząc, iż kolega nie robi tego, co należy, albo robi coś byle jak, nie mogę czekać, aż „ktoś" z tym „coś" zrobi.

A ponieważ Scrum Master i Product Owner również są częścią Zespołu, Developer ma prawo domagać się również i od nich, by wywiązywali się z obowiązków i robili to należycie.

Product Owner

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Scrum Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

Często pojawia się pokusa, by na odpowiedzialność Product Ownera patrzyć wyłącznie przez pryzmat Backlogu Produktu – jeśli „dobrze wybrał wartość", ale ona z jakiegoś powodu nie powstała, to „nie jest wina Product Ownera". To niewłaściwie rozumienie odpowiedzialności Product Ownera.

Osoba taka nie ma być „sekretarzem" zajętym wyłącznie obsługą Backlogu Produktu – ma ona zdecydować, co warto zrobić i współpracować z Developerami tak, by mieli oni szansę wartość tę wytworzyć. Ostatecznie to cały Zespół razem wytwarza wartościowy Przyrost co Sprint.

W praktyce oznacza to, że Product Owner niekoniecznie musi sam wszystkie elementy w Backlogu Produktu tworzyć i edytować, ale zachowuje odpowiedzialność za to, co tam jest. Nie musi sam rozmawiać z interesariuszami, ale odpowiada za to, co ostatecznie znajdzie się w produkcie. Nie buduje sam produktu (chyba, że jest też jednocześnie Developerem), ale musi być dostępny w czasie Sprintu i podejmować decyzje, gdy to potrzebne (np. odnośnie zmiany zakresu zmian w produkcie).

The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal;
- Creating and clearly communicating Product Backlog items;

Pamiętajmy, że mowa tu o odpowiedzialności, a nie robieniu tego osobiście w każdym przypadku. Product Owner przede wszystkim odpowiada za zapewnienie, że Zespół skupia się na robieniu tego, co ma wartość. Innymi słowy, jeśli Product Ownerowi brak czasu na zrozumienie, co warto robić, bo zajmuje się doskonaleniem opisów elementów w Backlogu Produktu, to nie wywiązuje się ze swojej odpowiedzialności.

- Ordering Product Backlog items; and,
- Ensuring that the Product Backlog is transparent, visible and understood.

*Przejrzysty, widoczny i zrozumiały – albo prościej: dostępny i zrozumiały. Taki właśnie musi być Backlog Produktu. Oznacza to, że wszyscy, którzy powinni, mają do niego dostęp, widzą taką samą zawartość oraz kolejność elementów i **rozumieją** na co patrzą. Ten ostatni postulat ciężko spełnić bez rozmowy i potwierdzenia – stymulowaniu tej dyskusji służą różne techniki (np. User Stories).*

Bardzo ważne jest również, aby każdy widział Backlog Produktu w tej samej kolejności – tej mianowicie, jaką ustalił Product Owner. Chodzi o to, żeby nie mogło powstać jakieś nieporozumienie co do tego, nad czym Zespół będzie pracował teraz lub w najbliższym czasie, wynikłe z tego, że Backlog Produktu ktoś widział uszeregowany tak, a ktoś siak (bo np. włączył filtry i sortowania, które niestety niektóre „narzędzia do Scruma" udostępniają).

The Product Owner may do the above work or may delegate the responsibility to others. Regardless, the Product Owner remains accountable.

Istotą roli Product Ownera jest decydowanie o tym, co Zespół Scrum robi i w jakiej kolejności. Wyrazem tej decyzji jest zawartość Backlogu Produktu (w tym Cel Produktu). By decyzje te podejmować, trzeba wykonać wiele innej pracy niż samo „edytowanie elementów Backlogu". Product Owner może ją wykonać sam lub zlecić innym. W każdym jednak przypadku Product Owner odpowiada za skutki swoich decyzji – za to, czy rzeczywiście wybrał i umieścił na Backlogu Produktu rzeczy najważniejsze.

For Product Owners to succeed, the entire organization must respect their decisions. These decisions are visible in the content and ordering of the Product Backlog, and through the inspectable Increment at the Sprint Review.

Product Owner jest częścią zespołu odpowiedzialnego za dostarczenie wartości, w czym uczestniczy realizując swoją odpowiedzialność – przede wszystkim określając, co jest tą wartością.

W stosunku do poprzednich wersji Scrum Guide pojawia się bardzo ważne stwierdzenie, że decyzje Product Ownera nie ograniczają się wyłącznie do Backlogu Produktu, ale ich „nośnikiem" jest ***również Przyrost****. Dlatego Product Owner nie może przyjść na Przegląd Sprintu, aby zobaczyć, co też Developerzy dostarczą; Product Owner* ***razem z Developerami i Scrum Masterem*** *demonstrują dostarczoną wartość interesariuszom. Oczywiście demonstracja ta jest jedynie narzędziem stymulującym dyskusję z interesariuszami o kolejnych krokach w rozwoju produktu, a efektem tej dyskusji mogą być zmiany w Backlogu Produktu.*

The Product Owner is one person, not a committee. The Product Owner may represent the needs of many stakeholders in the Product Backlog. Those wanting to change the Product Backlog can do so by trying to convince the Product Owner.

Istotna rzecz: odpowiedzialność Product Ownera jest jednoosobowa. Nie można jej rozmyć na „komitet sterujący" lub inne ciało, które pozwoliłoby się od niej uchylić. Nie może być też tak, że Product Owner jest przechodnią rolą wewnątrz Zespołu Scrum, którą okazjonalnie podejmują różne osoby, w zależności od sytuacji. Interesariusze muszą przecież wiedzieć, kto decyduje o zawartości i kolejności Backlogu Produktu.

Oznacza to zatem, że Product Owner musi mieć odpowiednią władzę: władzę ostatecznego decydowania co będzie w produkcie (a więc przede wszystkim władzę nad Backlogiem Produktu). Bez tej władzy nie mógłby ponosić odpowiedzialność za wartość produktu. Dlatego jest tak istotne, by decyzja Product Ownera mogła być zmieniona tylko przez samego Product Ownera.

Scrum Master

The Scrum Master is accountable for establishing Scrum as defined in the Scrum Guide. They do this by helping everyone understand Scrum theory and practice, both within the Scrum Team and the organization.

Scrum ma być taki, jak opisany w tym Scrum Guide. Jeśli jest jakiś inny, to nie jest to już Scrum. Oczywiście, nie oznacza to, że nie może istnieć jakiś inny pożyteczny i sensowny sposób pracy niż Scrum. Chodzi jednak o to, żeby rzeczy prawidłowo nazywać, nie oszukiwać samego siebie. Zadanie Scrum Mastera to doprowadzić do tego, żeby tak właśnie było. Osiągać je ma przede wszystkim poprzez budowanie ***zrozumienia*** *Scruma w zespole i w organizacji. Chodzi o to, żeby ludzie chcieli działać zgodnie ze Scrumem nie dlatego, że ktoś tak kazał, ale dlatego, że rozumieją, w jaki sposób pomoże im to lepiej pracować.*

The Scrum Master is accountable for the Scrum Team's effectiveness. They do this by enabling the Scrum Team to improve its practices, within the Scrum framework.

To kluczowe zdanie, którego bardzo brakowało w poprzednich wydaniach Scrum Guide! Scrum nie jest celem, ale narzędziem do osiągania celów, natomiast Scrum Master ma doprowadzić do tego, by ci, którzy chcą użyć tego narzędzia, robili to skutecznie. Przy czym nie chodzi tu jedynie o „zgodność ze Scrum Guide", ale o użycie metody w sposób dający oczekiwane korzyści. Jedną z nich jest właśnie efektywny zespół.

Scrum Master, którego zespół nie potrafi dostarczać wartościowych, nadających się do użycia rozwiązań, nie wywiązuje się zatem ze swoich obowiązków. Inaczej mówiąc, jeśli Product Owner nie potrafi skutecznie określić, co warto robić, lub Developerzy nie potrafią tej wartości wytworzyć – to Scrum Master również ponosi za to odpowiedzialność.

Scrum Masters are true leaders who serve the Scrum Team and the larger organization.

Poniżej następuje lista sposobów w jaki sposób Scrum Master służy i wspiera zespół, Product Owner i organizację jako całość. Warto się w nią wczytać, bo z tego, iż twórcy Scruma włożyli dużo wysiłku, by uczynić Scrum Guide jak najkrótszym można wnioskować, że te punkty poniżej są właśnie tymi najważniejszymi.

Z drugiej strony nie wolno zapominać, że są to jedynie wybrane działania, a Scrum Master nie może się ograniczyć wyłącznie do nich, jeśli jego Zespół lub organizacja wymaga czegokolwiek innego, nie ujętego na liście poniżej.

The Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality;
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;

Niepozorne, ale ważne stwierdzenie: by skutecznie dostarczyć tej „usługi", Scrum Master musi posiadać dość wiedzy o domenie produktu i technologii jego wytwarzania, aby przynajmniej potrafić zadać właściwe pytania.

W połączeniu z odpowiedzialnością za efektywność zespołu zapis ten jasno pokazuje ograniczone możliwości wywiązania się z obowiązków przez „nietechnicznego" Scrum Mastera w zespole, który buduje bardzo „technologiczne" produkty.

Równie nieskutecznie „nietechniczny" Scrum Master będzie uczył taki „technologiczny" zespół wszechstronności w działaniu (cross-functionality), a jeśli nie zechce nabyć podstaw wiedzy technicznej, sam stanie się anty-przykładem postawy, której powinien uczyć, trzymając się kurczowo swej „nietechniczności".

- Causing the removal of impediments to the Scrum Team's progress; and,

Ważne, choć niepozorne doprecyzowanie: Scrum Master ma spowodować, że usunięte zostaną utrudnienia, ale niekoniecznie ma je usunąć osobiście. Często będzie to zatem oznaczać, że nie tyle Scrum Master będzie coś robił, co będzie przypominał o konieczności zrobienia czegoś, sprawdzenia postępu prac itd.

Decydując się na bezpośrednie działanie, Scrum Master musi uwzględnić konsekwencje: na ile negatywnie wpłynie to na samozarządzanie Zespołu, ale też na ile powstrzymanie się od działania ograniczy efektywność.

- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

Warto zauważyć, że nie chodzi tu o dążenie to tego, by było „miło za wszelką cenę", bo to sprzeczne będzie z postulatem produktywności. „Pozytywność", o której mowa, związana jest z nastawieniem zdarzeń na rozwiązywanie problemów i otwieranie nowych możliwości, zamiast na rozliczenia i szukanie wymówek albo winnych. W szczególności Scrum Master ma spowodować, by Zespół nie powtarzał „nie da się", tylko poszukał możliwości (tego, co „da się" zrobić).

The Scrum Master serves the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog management;
- Helping the Scrum Team understand the need for clear and concise Product Backlog items;
- Helping establish empirical product planning for a complex environment; and,
- Facilitating stakeholder collaboration as requested or needed.

Zakres wsparcia, na jakie może liczyć Product Owner ze strony Scrum Mastera, wskazuje wyraźnie na to, że Scrum Master musi mieć szeroką wiedzę, znacznie wykraczającą poza samego Scruma.

Pomagając Product Ownerowi Scrum Master powinien pamiętać, by nie przejąć od niego odpowiedzialności. Scrum Master ma nauczyć jak robić te wszystkie rzeczy, a jeśli trzeba, operacyjnie pomóc w ich robieniu. Nie ma natomiast stać się Product Ownerem ani zastępować Product Ownera.

The Scrum Master serves the organization in several ways, including:

- Leading, training, and coaching the organization in its Scrum adoption;
- Planning and advising Scrum implementations within the organization;
- Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
- Removing barriers between stakeholders and Scrum Teams.

Postrzeganie Scrum Mastera jako pośrednika między zespołem a otoczeniem – w szczególności interesariuszami lub managementem – świadczy o niezrozumieniu Scruma. Zespół Scrum ma wykreować wartość dla tego „otoczenia", więc nie może się od niego izolować. Przeciwnie, musi budować jak najlepsze relacje i utrzymywać stałe interakcje z interesariuszami, innymi zespołami i managementem.

W niektórych Zespołach pokutuje błędne przekonanie, że praca w Sprintach ma odbywać się w całkowitej izolacji od otoczenia. Oczekują one tym samym od Scrum Mastera, że będzie „bronił ich" od interesariuszy, managementu i każdego, kto zechciałby „zakłócić Sprint". W ten sposób Zespół uzyskuje jedynie pozory stabilizacji, a w praktyce przestaje działać empirycznie — ignoruje bowiem wszelkie sygnały, które wymagają reakcji (dostosowania planów działania).

Scrum Master musi nauczyć zespół jak efektywnie współpracować z otoczeniem by szybko i skutecznie reagować na zmiany. Opisana powyżej postawa Zespołu i oczekiwania względem Scrum Mastera stanowić mogą jedną z barier, jakie Scrum Master musi usunąć.

Innym przykładem bariery, jaką Scrum Master powinien wyeliminować, jest postawa Product Ownera, który izoluje interesariuszy od Developerów w przekonaniu, że wyłącznie Product Ownerowi wolno utrzymywać relacje z interesariuszami. W praktyce wielu interesariuszy może wesprzeć Zespół wiedzą ekspercką związaną z produktem i sposobem jego użytkowania, a bezpośrednia komunikacja z Zespołem, bez pośrednictwa Product Ownera, pozwala najskuteczniej z takiego eksperckiego wsparcia skorzystać.

Scrum Events

The Sprint is a container for all other events. Each event in Scrum is a formal opportunity to inspect and adapt Scrum artifacts. These events are specifically designed to enable the transparency

required. Failure to operate any events as prescribed results in lost opportunities to inspect and adapt. Events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. Optimally, all events are held at the same time and place to reduce complexity.

Każde spotkanie w Scrum (bo poza Sprintem wszystkie „Events" to spotkania) ma wspierać empiryzm. Każde jest zaprojektowane tak, by stan przedsięwzięcia (produktu, backlogu itp.) na moment, w którym spotkanie się odbywa, uczynić transparentnym w sposób, który umożliwia dokonanie skutecznego sprawdzenia i następującej po nim sensownej adaptacji. Scrum Guide wyraźnie ostrzega, że rezygnacja z któregoś ze spotkań, albo przeprowadzenie go w sposób, który nie umożliwi osiągnięcia celu, dla którego zostało ustanowione, będzie oznaczać obniżenie możliwości działania empirycznego. Czyli będzie skutkować obniżeniem korzyści ze stosowania Scruma.

Planowanie spotkań tak, by nie było konieczne zastanawianie się czy, kiedy i gdzie się odbędą, ogranicza złożoność w pracy Zespołu, ale też ułatwia interakcję z otoczeniem. Zwracamy przy tym uwagę, że Scrum Guide nie określa, kiedy dokładnie odbywać się mają zdarzenia, więc np. to wybór Developerów, czy Daily Scrum robią rano, czy w połowie dnia. Natomiast sekwencja zdarzeń w Sprincie jest sztywno zdefiniowana. Ciężko bowiem sobie wyobrazić sobie, że zrobimy Planowanie w połowie Sprintu, albo wartościową Retrospekcję przed Przeglądem.

Jeśli wszystkie zdarzenia realizujemy dobrze to najczęściej nie ma już potrzeby organizowania innych spotkań, poza być może jakimiś sesjami pielęgnacji Backlogu Produktu („refinement"). Wystarcza wówczas całkowicie normalna, codzienna interakcja w Zespole Scrum oraz między poszczególnymi członkami Zespołu Scrum i innymi osobami.

The Sprint

Sprints are the heartbeat of Scrum, where ideas are turned into value.

To krótkie zdanie świetnie podsumowuje istotę Scruma – zmiana pomysłów (idei) na wartość (działający Przyrost produktu) w krótkich iteracjach.

Uwaga: nie jest to więc metoda zamieniania z góry zdefiniowanej listy wymagań na listę funkcjonalności produktu – wykorzystywanie Scruma w ten sposób niekoniecznie skutkuje uzyskaniem wymiernych korzyści, zwłaszcza jeśli produkowane funkcjonalności są nikomu niepotrzebne.

They are fixed length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous Sprint.

Pamiętajmy, że mowa tu jest o „czasie pracy Zespołu nad produktem" a nie o czasie w rozumieniu fizycznym. Jest całkowicie poprawną sytuacja, w której koniec Retrospekcji poprzedniego Sprintu od Planowania Sprintu następnego dzieli np. dwa dni weekendu – o ile oczywiście w tym czasie nie wykonuje się żadnej pracy związanej z produktem, Backlogiem Produktu i tak dalej.

All the work necessary to achieve the Product Goal, including Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective, happen within Sprints.

During the Sprint:

- No changes are made that would endanger the Sprint Goal;
- Quality does not decrease;

Planowanie Sprintu odbywa się w kontekście obowiązującego w Zespole Definition of Done, które w trakcie Sprintu nie powinno być zmieniane (a precyzyjniej mówiąc, nie powinno się obniżać wyrażonych w nim wymogów jakościowych). Co oczywiste, jak najbardziej dopuszczalne jest wytworzenie rozwiązania o jakości wyższej, niżby wynikało to z Definition of Done – bo wtedy wciąż będzie ono spełnione.

- The Product Backlog is refined as needed; and,

Scrum Guide nie daje już wskazówek, ile czasu poświęcić należy na pielęgnację Backlogu Produktu („refinement"). Zespół Scrum musi sam uzgodnić, jak zamierza prowadzić proces pielęgnacji, pamiętając przy tym o spoczywającej na nim odpowiedzialności za efektywne dostarczanie wartości.

Co istotne, rezygnacja z konceptu Zespołu Developerskiego na rzecz jednego Zespołu Scrumowego czyni jasnym, że za pielęgnację odpowiadają w równym stopniu Developerzy, co Product Owner i Scrum Master. Bezprzedmiotowe staje się narzekanie, że Product Owner „marnie opisał wymagania" lub czekanie, aż zorganizuje sesję pielęgnacji Backlogu Produktu – skoro jesteśmy z nim w jednym Zespole i mamy doprowadzić do tego, by Backlog Produktu pozwalał skutecznie dostarczać wartość - nie narzekajmy tylko ustalmy, jak chcemy działać.

- Scope may be clarified and renegotiated with the Product Owner as more is learned.

Jeden z mitów o Scrumie mówi, że „zakres prac nie może zmieniać się w trakcie Sprintu" – co byłoby zaprzeczeniem działania empirycznego. Zakres prac jest wypadkową bieżącego stanu prac i Celu Sprintu, który zamierzamy osiągnąć. To oznacza, że ujawnione już w trakcie Sprintu czynniki (możliwości, problemy, pomysły itd.) wymagają dostosowywania zakresu prac. Zdarzeniem, które napędza ten proces empirycznego sprawdzenia i adaptacji, jest Daily Scrum.

Oczywiście to nie oznacza, że Product Owner może kazać Developerom coś zrobić bez porozumienia z nimi. To nie oznacza również, że Developerzy mogą zacząć robić coś innego, niż ustalili z Product Ownerem w czasie Planowania Sprintu. Skoro są w jednym Zespole, wspólnie decydują o ewentualnych zmianach, przy czym spoczywa na nich specyficzna odpowiedzialność: Product Owner musi zapewnić, że robione są wartościowe rzeczy, a Developerzy, że wartość ta powstanie. To czy zmiany zakresu prac nastąpią i jakie to będą zmiany, będzie więc wypadkową realizacji tych dwóch odpowiedzialności („czego potrzebujemy" vs. „co jest możliwe do zrobienia").

Sprints enable predictability by ensuring inspection and adaptation of progress toward a Product Goal at least every calendar month. When a Sprint's horizon is too long the Sprint Goal may become invalid, complexity may rise, and risk may increase. Shorter Sprints can be employed to generate more learning cycles and limit risk of cost and effort to a smaller time frame. Each Sprint may be considered a short project.

Określona jest maksymalna długość Sprintu, nie jest określona długość minimalna. Co ważne, określony jest meta-cel każdego Sprintu: zbliżyć się do realizacji ustalonego Celu Produktu.

Z tego wynika, że nie mają wiele wspólnego ze Scrumem „Sprinty", w których nie buduje się produktu (np. „Sprint wydaniowy", „Sprint testowy", „Sprint integracyjny", „Sprint Zero" itd.). Działania takie nie wytwarzają nowej wartości (nie podnoszą wartości produktu), a są jedynie etapem jej wytwarzania. Praca ta powinna odbywać się w ramach Sprintów, w których rozwijany jest produkt, a nie stanowić osobne fazy, być może nazywane dla zmyłki „sprintami".

Równie mało ze Scrumem będzie miał wspólnego „Sprint", po którym nie da się ustalić czy i jaki postęp w osiąganiu Celu Produktu nastąpił. Postęp rozumiany oczywiście nie jako „procent zrealizowanego planu" (bo to w praktyce oznacza najczęściej „nie mamy na razie nic dającego się użyć"), ale jako przyrost wartości produktu, który może zostać udostępniony użytkownikom.

Każdy Sprint można traktować jako mini-projekt, ponieważ jest inicjatywą ograniczoną czasowo, mającą na celu osiągnięcie konkretnego Celu Sprintu, a koszt (wynikający z tego jakich narzędzi korzystamy i jaki jest skład Zespołu) również jest ograniczony.

Natomiast tu analogia do projektów w zasadzie się kończy. W Sprincie działa empiryczna kontrola procesu, co oznacza, że w każdym momencie może okazać się konieczne dostosowanie planów działania czy zakresu prac – niezmienny pozostaje za to sam Cel Sprintu. Nie będzie zatem ze Scrumem miało nic wspólnego realizowanie „Sprintów" za pomocą klasycznych metod projektowych, gdzie na początku podczas Planowania zbudowany zostanie sztywny plan działań, wykonywanych następnie przez Developerów, bez dostosowywania tego planu w trakcie. Przeciwnie, takie podejście byłoby zarówno nieoptymalne (brak szans na dostosowanie w miarę postępów prac w Sprincie) jak i ryzykowne (na skutek braku empirycznej kontroli podczas Sprintu dużo większe ryzyko, że nie zostanie osiągnięty Celu Sprintu).

Various practices exist to forecast progress, like burn-downs, burn-ups, or cumulative flows. While proven useful, these do not replace the importance of empiricism. In complex environments, what will happen is unknown. Only what has already happened may be used for forward-looking decision making.

Warto zwrócić uwagę na powyższe podkreślenie znaczenia empiryzmu. W istocie różne praktyki, szacowania, prognozy są próbą okiełznania przez nas niepewności, którą zawiera w sobie przyszłość. Dopóki praca jest złożona, ale w miarę stabilna i również wykonywana w stabilnym środowisku, dopóty praktyki takie są pożyteczne, bo umożliwiają jakieś zaplanowanie np. innych działań, które zależą od tego, co w danej chwili „umie" rozwijany w Scrumie produkt.

Nie należy jednak nigdy zapominać, że działamy w środowisku niestabilnej złożoności, co skłoniło nas do wykorzystania Scruma. A to oznacza, że w każdej chwili nasze plany i prognozy mogą się okazać nic nie warte i będzie mogli oprzeć się wyłącznie na empirycznym reagowaniu na zmianę, póki – być może – sytuacja się ponownie nie ustabilizuje. Rok 2020 dostarczył nam wystarczająco wiele przykładów takich sytuacji.

A Sprint could be cancelled if the Sprint Goal becomes obsolete. Only the Product Owner has the authority to cancel the Sprint.

Anulowanie Sprintu oznacza oczywiście jego wcześniejsze zakończenie – wypada wtedy przeprowadzić, jak w przypadku każdego innego Sprintu, Retrospekcję. Co więcej, kolejny Sprint (o ile się odbędzie) rozpocznie się wcześniej – od razu po zakończeniu bieżącego, który został anulowany.

Zwracamy też uwagę, że anulowanie Sprintu związane jest z unieważnieniem Celu Sprintu, a nie uznaniem podjętych do Sprintu elementów Backlogu Produktu za zbędne. Inaczej mówiąc, jeśli Cel Sprintu nadal jest wartościowy i chcemy go osiągnąć, ale odkryjemy, że realizowane w tym Sprincie elementy Backlogu Produktu nie pozwolą nam go osiągnąć, to nie jest to powód do anulowania Sprintu. Zamiast tego należy dostosować zakres prac w Sprincie tak, by osiągnięcie Celu Sprintu było możliwe.

Z drugiej strony, jeśli Cel Sprintu stracił sens, nie implikuje to, że nie są potrzebne realizowane w tym Sprincie elementy Backlogu Produktu. Jeśli anulowany zostanie Sprint, niezrealizowane elementy wrócić powinny do Backlogu Produktu, oczywiście stosownie uaktualnione tak, by odzwierciedlić stan, w jakim teraz się znajdują. Dalszy ich los zależy rzecz jasna od decyzji Product Ownera.

Czy Product Owner może anulować Sprint tylko dlatego, że chce natychmiast rozpocząć realizację nowych elementów Backlogu Produktu – ważniejszych, niż ustalony Cel Sprintu bieżącego? W teorii tak (bo z punktu widzenia Product Ownera ten Cel jest już nieaktualny - „is obsolete"), ale konsekwencje anulacji są poważne: zaburzony rytm Sprintów spowoduje zamieszanie i utrudni pracę Zespołu, część pracy już wykonanej może zostać zmarnowana, spadnie efektywność działania całego Zespołu. Taka decyzja zazwyczaj źle świadczy o kunszcie Product Ownera, a jej podjęcie powinno być poprzedzone dyskusją całego Zespołu Scrum.

Sprint Planning

Sprint Planning initiates the Sprint by laying out the work to be performed for the Sprint. This resulting plan is created by the collaborative work of the entire Scrum Team.

Planowanie Sprintu inicjuje, czyli rozpoczyna Sprint. Dopóki nie rozpocznie się Planowanie, kolejny Sprint jeszcze się nie rozpoczął.

The Product Owner ensures that attendees are prepared to discuss the most important Product Backlog items and how they map to the Product Goal. The Scrum Team may also invite other people to attend Sprint Planning to provide advice.

Pierwsze zdanie nakłada na Product Ownera obowiązek upewnienia się, że uczestnicy są przygotowani do tego spotkania, a w szczególności, że rozumieją rzeczy, które znajdują się na szczycie Backlogu Produktu i wiedzą jak one się mają do aktualnego Celu Produktu. Product Owner może to osiągnąć tylko w jeden sposób: wcześniejszą dyskusją z planowanymi uczestnikami Planowania, do której dobrą okazją mogą być sesje / spotkania związane z pielęgnacją Backlogu Produktu.

„Uczestnicy" oznacza wszystkich uczestników Planowania, a więc także tych spoza Zespołu Scrum, których ów Zespół na to spotkanie zaprosił. Oznacza to, że opisany wyżej obowiązek Product Ownera rozciągać się może poza samych członków Zespół Scrum. Takim uczestnikiem spoza Zespołu może być np. ekspert, który dostarczy wiedzy niezbędnej do efektywnego planowania.

Drugie zdanie powyżej powinno być czytane w kontekście akapitu i rozdziału, w którym się znajduje. Chodzi o ekspertów, którzy pomogą / doradzą / dostarczą wiedzy tak, by Zespół mógł zaplanować Sprint. Nie wynika z tego np. prawo do tego, by Scrum Master zaprosił „dla siebie" eksperta, który będzie mu (i tylko mu) doradzał, jak wywiązywać się z obowiązków w czasie Planowania Sprintu. Oczywiście nic nie stoi na przeszkodzie, by Scrum Master pojawił się na Planowaniu np. ze swoim mentorem, który pomaga mu w doskonaleniu się – pod warunkiem, że Zespół nie ma nic przeciwko, cel pojawienia się mentora jest jasny, a obecność (i ewentualne dyskusje) Scrum Mastera z mentorem nie zaburzą przebiegu zdarzenia.

Specyficzna odpowiedzialność spoczywa tu zresztą na Scrum Masterze, który musi upewnić się, że to nie zaproszeni eksperci „planują Zespołowi Sprint", ale że to Zespół Scrum planuje swój Sprint.

Ponieważ mamy do czynienia z jednym Zespołem, nie ma już zastrzeżenia, że to Zespół Developerski zaprasza ekspertów (tak było w poprzedniej wersji Scrum Guide). Tym niemniej to najczęściej Developerzy potrzebować będą wsparcia (np. eksperta technicznego, osoby znającej dobrze jakiś standard, narzędzie lub infrastrukturę itd.).

Sprint Planning addresses the following topics:

Topic One: Why is this Sprint valuable?

The Product Owner proposes how the product could increase its value and utility in the current Sprint. The whole Scrum Team then collaborates to define a Sprint Goal that communicates why the Sprint is valuable to stakeholders. The Sprint Goal must be finalized prior to the end of Sprint Planning.

Planowanie Sprintu nie jest liniową / kaskadową, ale iteracyjną aktywnością. To oznacza, że w ramach jednego Planowania może pojawić się kilka koncepcji Celu Sprintu, zmieniać się może również Backlog Produktu i Backlog Sprintu. Zespół ma wszystkie kompetencje i odpowiedzialności niezbędne, by tego dokonać.

Dlatego Product Owner proponuje (za pomocą Backlogu Produktu) co może przynieść maksimum wartości w Sprincie, po czym w dyskusji z Developerami i Scrum Masterem wykuwa się Cel Sprintu i plan jego realizacji (Backlog Sprintu).

Scrum Guide wskazuje też, czym powinien być Cel Sprintu: zrozumiałym dla interesariuszy wyjaśnieniem wartości, jaką przyniesie Sprint. Tym samym często spotykane „składkowe Cele Sprintu", wymieniające de facto po kolei wymagania, jakie zostaną zrealizowane, nie są poprawnym Celem, bo odpowiadają na pytanie „co zrobimy?", a nie „co nam to da?". Jeszcze mniej sensu ma Cel Sprintu „zrobić to wszystko".

Tu niejako przypominamy, że trudność w określeniu Celu Sprintu prawie zawsze jest sygnałem marnej kondycji Backlogu Produktu. Skoro Zespół ma trudność z odpowiedzią na pytanie, dlaczego na szczycie Backlogu Produktu znalazły się takie a nie inne elementy, niewykluczone, że są one niezrozumiałe albo wręcz zbędne. Czasem trudność w określeniu Celów Sprintu wskazuje na głębszy problem – złą definicję produktu, a być może nawet zły dobór metody – czyli Scrum – do charakteru pracy danego zespołu.

Mała, ale istotna uwaga: Cel Sprintu nie musi zatem według Scrum Guide być określany na samym końcu, może być ustalony w dowolnym momencie Planowania. Ważne, aby był ustalony, zanim się ono zakończy.

Topic Two: What can be Done this Sprint?

Through discussion with the Product Owner, the Developers select items from the Product Backlog to include in the current Sprint. The Scrum Team may refine these items during this process, which increases understanding and confidence.

> *Developerzy wybierają z Backlogu Produktu elementy, które zrealizują – ale nie oznacza to, że traktują ten Backlog jak swoiste „menu", skacząc po nim dowolnie. Kolejność Backlogu Produktu odzwierciedla wiedzę Product Ownera o tym, co może przynieść największą wartość w tym Sprincie. Wiedza ta powinna być znana i zrozumiała dla Developerów – temu służy proces pielęgnacji Backlogu Produktu („refinement"). Product Owner ma nawet obowiązek upewnić się, że tak jest, a więc, że Developerzy są przygotowani do Planowania, o czym była mowa wcześniej.*
>
> *Jeśli więc celem Developerów nie jest sabotowanie produktu, który rozwijają, będą trzymać się kolejności elementów Backlogu Produktu, bo będą rozumieć, z czego ona wynika, a być może sami na nią wpłyną – dostarczając Product Ownerowi wiedzy o zależnościach.*
>
> *Może też zdarzyć się, że Developerzy zechcą „przeskoczyć" jakiś element w Backlogu Produktu – bo pozwoli to osiągnąć więcej w tym Sprincie albo Cel Sprintu zostanie w ten sposób wzmocniony. Pamiętajmy, że na Planowaniu obecny jest Product Owner i może w trakcie zmieniać kolejność elementów, dostosowując Backlog Produktu.*
>
> *Zwracamy uwagę, że choć mamy jeden Zespół, tylko Developerzy decydują, **ile elementów Backlogu Produktu podejmą do realizacji** w Sprincie.*

Selecting how much can be completed within a Sprint may be challenging. However, the more the Developers know about their past performance, their upcoming capacity, and their Definition of Done, the more confident they will be in their Sprint forecasts.

> *Pojawia się tu słówko „forecast" – prognoza. Przypomina ono, że mimo najlepszych intencji, wiedzy i doświadczenia, na etapie Planowania zarówno Backlog Sprintu jak i Cel Sprintu to tylko prognoza, zamiar. Co się naprawdę wydarzy w Sprincie - dopiero się okaże.*

Topic Three: How will the chosen work get done?

For each selected Product Backlog item, the Developers plan the work necessary to create an Increment that meets the Definition of Done. This is often done by decomposing Product Backlog items into smaller work items of one day or less. How this is done is at the sole discretion of the Developers. No one else tells them how to turn Product Backlog items into Increments of value.

Pierwsze zdanie powyższego akapitu może sugerować, że dla każdego zrealizowanego elementu Backlogu Produktu powstaje osobny Przyrost. I rzeczywiście tak będzie, jeśli Zespół serio potraktuje ideę pracy zespołowej oraz wartość Scrum jaką jest skupienie. Realizując elementy Backlogu Produktu sekwencyjnie jeden po drugim, po ukończeniu prac nad każdym z nich uzyskają nowy Przyrost. Taki sposób pracy w Scrumie jest optymalny, bo maksymalizuje wartość uzyskiwaną w kolejnych Sprintach i pozwala uzyskać pierwszy Przyrost na długo przed zakończeniem Sprintu.

*Przy czym pamiętać należy, że chodzi o Przyrost **wartości**, a nie „wykonanie pracy" – z faktu wykonania jakichś prac nie wynika automatycznie, że efekty tej pracy przynoszą komukolwiek wartość.*

Jeśli jednak Zespół zdecyduje się pracować nad wieloma elementami Backlogu Produktu równolegle, pojawi się konieczność integrowania efektów w spójną całość. To oznacza, że planując pracę w taki sposób Developerzy muszą nie tylko uwzględnić wysiłek związany z budowaniem funkcjonalności czy wprowadzaniem w nich zmian, ale też zaplanować integrację rozwijanych równolegle funkcjonalności tak, by powstał Przyrost. W praktyce takie działanie często powoduje, że pierwszy Przyrost pojawia się dopiero pod koniec Sprintu, a nierzadko w Sprincie wytworzyć udaje się tylko jeden Przyrost, tuż przed końcem iteracji. Ryzyko, że Sprint nie przyniesie żadnej wartości, jest wtedy spore.

„Work items", o których mowa, mogą mieć formę zadań, mogą to być listy kontrolne, może też nastąpić dekompozycja elementów Backlogu Produktu na mniejsze elementy. Żadna z tych praktyk nie jest w Scrumie wymagana. Pamiętając wszakże o wartości Scrum, jaką jest skupienie, warto poszukać takiego sposobu organizacji pracy i wizualizacji postępu, który zapobiegnie „rozpadowi" wymagań na liczne mniejsze elementy, zmniejszając przejrzystość Backlogu Sprintu.

W każdym przypadku sposób prowadzenia prac w Sprincie – a co za tym idzie także sposób ich wizualizacji – pozostaje wyłączną domeną Developerów.

The Sprint Goal, the Product Backlog items selected for the Sprint, plus the plan for delivering them are together referred to as the Sprint Backlog.

*Cel Sprintu jest **częścią** Backlogu Sprintu, a nie czymś oddzielnym. Więcej informacji na ten temat znajduje się w dalszej części Scrum Guide w sekcji poświęconej artefaktom.*

Sprint Planning is timeboxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Od dawna już nie obowiązuje zasada proporcjonalności wyznaczania czasu trwania Planowania Sprintu w zależności od długości samego Sprintu – jeśli jest to uzasadnione i Zespół Scrumowy uzna takie działanie za efektywne, można nawet bardzo krótkie Sprinty planować przez osiem godzin (oczywiście nie sugerujemy, by to robić).

Z drugiej strony, jeśli Planowanie np. dwutygodniowego Sprintu trwa kilka minut, choć teoretycznie jest to dopuszczalne, powinno skłonić do refleksji w ramach Retrospekcji. Bo, w praktyce, oznacza to, że działania, które powinny odbywać się w czasie Planowania, zachodzą wcześniej, a tym samym wiele decyzji podejmowanych zostaje przedwcześnie (bez wiedzy o aktualnym stanie spraw).

W każdym przypadku maksymalna długość Planowania powinna być ustalona z góry przez Zespół Scrum i przestrzegana. Nie powinna być „płynna" i ustalana ad hoc, a już zdecydowanie niezgodne ze Scrumem będzie kontynuowanie spotkania „tak długo jak trzeba", niezależnie od ustalonego wcześniej timeboxu.

Daily Scrum

The purpose of the Daily Scrum is to inspect progress toward the Sprint Goal and adapt the Sprint Backlog as necessary, adjusting the upcoming planned work.

Daily Scrum pozwala na synchronizację Developerów, którzy – co oczywiste – rozmawiają (albo powinni rozmawiać) ze sobą również poza tym zdarzeniem. Różnica między taką bieżącą komunikacją a Daily Scrumem polega na tym, że w trakcie Daily Scruma ***wszyscy*** *Developerzy patrzą na* ***całość prac*** *realizowanych w Sprincie, a nie tylko na ten ich wyrywek, którym aktualnie są zajęci.*

To odłożenie pracy na moment i sprawdzenie jaki jest globalny stan spraw w Sprincie, pozwala odkryć konieczność zmiany planów poczynionych wcześniej, przegrupowania się itd. Zmiany tej dokonuje się czasami na bieżąco w czasie Daily Scruma, ale często wymaga to szerszej dyskusji. Wtedy w trakcie Daily Scruma ustalane jest, kiedy taka dyskusja się odbędzie. Tym samym następuje rozdzielenie synchronizacji / identyfikacji problemów i decyzji o konieczności dokonania zmian od implementacji takich zmian.

The Daily Scrum is a 15-minute event for the Developers of the Scrum Team. To reduce complexity, it is held at the same time and place every working day of the Sprint. If the Product Owner or Scrum Master are actively working on items in the Sprint Backlog, they participate as Developers.

Jeśli Product Owner albo Scrum Master są jednocześnie Developerami w Zespole, uczestniczą w Daily Scrumie. Poza tym zastrzeżeniem jest to spotkanie ***dla*** *Developerów. Nie dla Scrum Mastera, nie dla Product Ownera, tym bardziej nie dla innych osób.*

The Developers can select whatever structure and techniques they want, as long as their Daily Scrum focuses on progress toward the Sprint Goal and produces an actionable plan for the next day of work. This creates focus and improves self-management.

Developerzy mogą wybrać dowolną strukturę czy techniki, czyli sami decydują w jaki sposób przebiegać będzie ich spotkanie. Muszą jednak pamiętać, że celem jest sprawdzenie postępu ku osiągnięciu Celu Sprintu i dopasowanie – a więc być może zmiana – planów na kolejny dzień, co może mieć odzwierciedlenie w zmianach na Backlogu Sprintu.

Odpowiedzialnością Scrum Mastera jest zadbanie, by Developerzy rozumieli zarówno cel i sens tego spotkania jaki swoją relację do niego (przede wszystkim odpowiedzialność za skuteczną synchronizację działań i adaptację planów).

Daily Scrums improve communications, identify impediments, promote quick decision-making, and consequently eliminate the need for other meetings.

The Daily Scrum is not the only time Developers are allowed to adjust their plan. They often meet throughout the day for more detailed discussions about adapting or re-planning the rest of the Sprint's work.

Tak, jak pisaliśmy wcześniej, Daily Scrum służy synchronizacji i decyzji o konieczności adaptacji planów. Natomiast adaptacja tych planów i komunikacja z tym związana odbywa się na bieżąco. Daily Scrum służy wyrównaniu stanu wiedzy między Developerami tak, by wszyscy rozumieli tak samo, jaki jest postęp prac w Sprincie i plan działania w pozostałej części Sprintu.

Sprint Review

The purpose of the Sprint Review is to inspect the outcome of the Sprint and determine future adaptations. The Scrum Team presents the results of their work to key stakeholders and progress toward the Product Goal is discussed.

*Ta pozornie niewielka zmiana jest bardzo ważna: Przegląd Sprintu służy sprawdzeniu tego, **jaką wartość ma produkt w momencie zakończenia Sprintu**. Nie chodzi więc o to, czy zostały zrealizowane podjęte do Sprintu elementy Backlogu Produktu, ani nawet o sam Przyrost. Chodzi o zderzenie wartości Przyrostu z bieżącymi potrzebami i sprawdzenie, na ile i czy w ogóle zbliżamy się do osiągnięcia aktualnego Celu Produktu.*

Stąd wniosek, że Przegląd Sprintu nie może być samą demonstracją, i że nie może się odbyć bez uwzględnienia wiedzy o aktualnych potrzebach interesariuszy (w tym opinii użytkowników, trendów rynkowych itd.).

During the event, the Scrum Team and stakeholders review what was accomplished in the Sprint and what has changed in their environment. Based on this information, attendees collaborate on what to do next. The Product Backlog may also be adjusted to meet new opportunities. The Sprint Review is a working session and the Scrum Team should avoid limiting it to a presentation.

W tym jednym akapicie zawartych jest wiele wskazówek jak powinny wyglądać Przeglądy Sprintu.

*Zespół Scrum i Interesariusze **współpracują** przeglądając nie tylko wytworzony Przyrost, ale także wszystko co się zmieniło w **ich otoczeniu** (a więc obejmuje to potrzeby interesariuszy i inne zmiany mające wpływ na produkt, jego wartość itp.) i w oparciu oto wypracowują **co robić dalej**. W efekcie może zostać zmieniony – od razu, w trakcie Przeglądu – Backlog Produktu. Ponieważ jest to **sesja robocza** (czyli dyskusja, współpraca) „odbębnienie" tego przez „demo" albo prezentację nie jest wartościowym wykorzystaniem tego zdarzenia Scrum.*

Zatem, wbrew nazwie, to nie jest spotkanie, na którym skupiamy się na tym co było (w Sprincie) – przeciwnie, przyglądamy się temu co było głównie po to, żeby ustalić, jak jest obecnie – i na tej podstawie zdecydować co robić dalej. W skrócie: Inspect&Adapt.

The Sprint Review is the second to last event of the Sprint and is timeboxed to a maximum of four hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Podobnie jak w przypadku Planowania Sprintu, czas trwania Przeglądu Sprintu zależy od decyzji Zespołu, maksymalny timebox to cztery godziny. Nieaktualny już od dawna jest wymóg, by czas ten uzależnić proporcjonalnie od długości samego Sprintu. Również jak w przypadku Planowania, czas tych spotkań powinien być ustalony z góry.

Sprint Retrospective

The purpose of the Sprint Retrospective is to plan ways to increase quality and effectiveness.

Dla wielu Scrum Masterów może być zaskoczeniem, iż Retrospekcja Sprintu nie może ograniczyć się do „powodowania, żeby nam się miło pracowało", a jeszcze większym, że to nie jest celem tego zdarzenia. Dobra atmosfera w pracy jest ważna, ale to jedynie narzędzie a nie cel sam w sobie – jeśli Zespół będzie w miłej atmosferze tracił czas na robienie zbędnych rzeczy, ciężko mówić o jakiejkolwiek efektywności.

Wielu Scrum Masterów błędnie utrzymuje, że Retrospekcja nie może służyć do dyskusji o tym jak zrefaktoryzować produkt, jakie standardy techniczne powinny obowiązywać, jak implementować nowe rozwiązania w produkcie itd. Twierdzą, że Retrospekcja ma „służyć rozwojowi Zespołu, a nie produktu czy technicznym dyskusjom".

Tymczasem sensem istnienia Zespołu jest przecież wytwarzanie produktu (a ściślej dostarczanie wartości poprzez działający produkt) i Retrospekcja jest jak najbardziej miejscem, gdzie należy takie tematy poruszać. Poczynione na Retrospekcji ustalenia przełożą się na wyższą jakość i większą efektywność działania Developerów. W szczególności to w trakcie Retrospekcji Sprintu przeglądowi i dostosowaniu poddawane bywa Definition of Done, co ma bezpośredni związek z tym, jak budowany jest produkt.

The Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done. Inspected elements often vary with the domain of work.

Assumptions that led them astray are identified and their origins explored. The Scrum Team discusses what went well during the Sprint, what problems it encountered, and how those problems were (or were not) solved.

> *Podczas Retrospekcji warto poszukać założeń, jakie poczyniliśmy, a które nie potwierdziły się. Nie tylko po to, by w przyszłości „robić lepsze założenia", ale aby wyeliminować albo ograniczyć ilość założeń, jakie czynimy, i oprzeć się na faktycznych danych tam, gdzie to możliwe (to wzmocni empiryzm w naszym procesie).*
>
> *Warto zauważyć, że na pierwszym miejscu wśród tematów omawianych jest to, co w Sprincie poszło dobrze. Jak wiemy częstą tendencją na Retrospekcjach jest koncentracja na problemach i wpadkach. Choć należy o nich oczywiście dyskutować, celem tego spotkania jest uzyskanie usprawnień, robienie rzeczy lepiej. Usprawnienia mogą wynikać także z przeanalizowania tego, co się udało – rzeczy dobre, które się zdarzyły, można uczynić stałym elementem pracy zespołu lub wyciągnąć z nich inne wnioski na przyszłość.*

The Scrum Team identifies the most helpful changes to improve its effectiveness. The most impactful improvements are addressed as soon as possible. They may even be added to the Sprint Backlog for the next Sprint.

> *Najważniejsze usprawnienia powinny być wykonane bez zbędnej zwłoki. To istotne – i logiczne. Jeśli podczas Retrospekcji jako zespół ustaliliśmy co należałoby zrobić, aby było lepiej, to czemu zwlekać z wprowadzeniem tego w życie? Jeśli zaś zwlekamy, bo czegoś nie ma lub czegoś brakuje, to być może pierwszą akcją, do wykonania natychmiast, powinno być podjęcie działania, które pozwoli nam ów brakujący element uzyskać.*
>
> *Jeśli usprawnienie implikuje wykonanie konkretnej pracy, powinna znaleźć ona odzwierciedlenie w Backlogu Sprintu kolejnego Sprintu. Jeśli jest to zmiana dotycząca procesu (czyli umawiamy się na jakiś sposób postępowania, wprowadzamy jakąś zasadę), to samo „wprowadzenie zmiany" nie implikuje pracy – ciężko więc będzie stworzyć reprezentujący ją element Backlogu Sprintu. Natomiast wciąż warto zadbać, by zmiana ta była widoczna. Czyli jeśli to jest jakaś zasada, standard, którego mamy przestrzegać jako Zespół, to warto mieć to gdzieś zapisane – niewykluczone, że nawet w Definition of Done.*
>
> *Może też zdarzyć się, że sprawdzenie czy zmiana procesowa daje oczekiwane rezultaty, wymaga określonych działań. Wtedy jak najbardziej można zdefiniować ją wprost w Backlogu Sprintu kolejnego Sprintu.*

The Sprint Retrospective concludes the Sprint. It is timeboxed to a maximum of three hours for a one-month Sprint. For shorter Sprints, the event is usually shorter.

Retrospekcja Sprintu może trwać maksymalnie trzy godziny i również dla tego zdarzenia już od dawna nie ma wymogu, by wyznaczać timebox proporcjonalnie do długości całego Sprintu. Warto zapewnić dość czasu na dyskusję, by uniknąć sytuacji, w której Retrospekcje Sprintu nie dają realnych usprawnień – dlatego choć teoretycznie zgodne ze Scrumem byłyby np. pięciominutowe Retrospekcje, w praktyce mogłyby one nie dać żadnej wartości. Oczywiście, czas trwania tego spotkania powinien być określony z góry i raczej stały ze Sprintu na Sprint.

Retrospekcja Sprintu jest ostatnim zdarzeniem w Sprincie. Innymi słowy jej zakończenie jest także końcem Sprintu.

Pojawia się czasami wątpliwość, co robić, gdy Retrospekcja Sprintu odbywa się w połowie dnia, ale zbyt późno, by zrobić od razu po niej Planowanie Sprintu kolejnego. Aby tego uniknąć, niektóre Zespoły celowo odkładają Retrospekcję na koniec dnia tak, by zaraz po niej wyjść do domu. Ale nie warto czynić takich zabiegów.

Scrum nie dopuszcza istnienia między Sprintami pracy nad rozwojem produktu – inaczej mówiąc, nie może być pracy związanej z wytwarzaniem wartości, która odbywa się poza Sprintem (a tym samym poza procesem, w sposób nieprzejrzysty). Ale Scrum nie wymusza „przyklejenia" Planowania Sprintu kolejnego do zakończenia Retrospekcji Sprintu poprzedniego. W tym czasie ludzie idą do domu, spędzić czas z rodziną, czasami Sprinty rozdziela weekend itd.

Jeśli „między sprintami" pojawi się klika godzin, w których można wykonać jakąś pracę, póki nie budujemy produktu „awansem" na przyszły, jeszcze niezaplanowany Sprint, każde działanie jest dopuszczalne. Można usunąć jakieś znane błędy z produktu, można dokonać refaktoryzacji istniejącego rozwiązania, nauczyć się korzystania z jakiegoś narzędzia, zmodyfikować infrastrukturę używaną do pracy developerskiej itd. Można też poświęcić ten czas np. na pielęgnację Backlogu Produktu albo szybki spike developerski. Nie zmieni to przecież stanu produktu w sposób, który unieważniłby ustalenia poczynione na Przeglądzie Sprintu.

To oznacza, że jak najbardziej można zakończyć Retrospekcję Sprintu o rozsądnej godzinie, zająć się takimi pracami rozwojowo-utrzymaniowo-serwisowymi do końca dnia i być może jeszcze rankiem dnia kolejnego. Po czym można rozpocząć kolejny Sprint od Planowania Sprintu w południe, gdy wszyscy członkowie Zespołu będą dostępni, by wziąć w nim udział.

Scrum Artifacts

Scrum's artifacts represent work or value. They are designed to maximize transparency of key information. Thus, everyone inspecting them has the same basis for adaptation.

„Artefakty reprezentują pracę albo wartość" – przyjrzyjmy się temu zdaniu.

Praca – czyli opis pracy, jednostki pracy – to coś, co znajduje się na Backlogach. W szczególności na Backlogu Produktu znajdują się elementy opisujące wartościową zmianę w produkcie (opisujące samą zmianę lub – lepiej – opisujące oczekiwane skutki dla użytkowników), a na Backlogu Sprintu jednostki pracy dające przejrzystość postępów pracy w Sprincie.

Z kolei Przyrost reprezentuje wartość. Wartość ta wynika z tego, że dzięki pracy Developerów jest ten Przyrost ukończony (Done) i używalny oraz z tego, że dzięki działalności Product Ownera znalazły się w nim zmiany powodujące, że produkt w tej nowej wersji jest bardziej wartościowy.

Sam fakt istnienia artefaktów zdefiniowanych w Scrumie nie determinuje ich przejrzystości. Odpowiedzialność za przejrzystość Backlogu Produktu spoczywa na Product Ownerze. Za przejrzystość Backlogu Sprintu i Przyrostu odpowiadają Developerzy.

A za przejrzystość jakich artefaktów odpowiada Scrum Master? Odpowiedź jest prosta: wszystkich, ponieważ odpowiada za uświadomienie Product Ownerowi i Developerom ich odpowiedzialności oraz nauczenie ich jak się z niej wywiązać. Ma też obowiązek ujawnić brak przejrzystości któregoś z artefaktów – jeśli faktycznie tej przejrzystości brakuje.

Each artifact contains a commitment to ensure it provides information that enhances transparency and focus against which progress can be measured:

> *„Commitment" jest w tym miejscu użyte nie w znaczeniu polskiego słowa „zaangażowanie", ale „zobowiązanie" – to umowa, swoisty kontrakt, jaki zawiera sam ze sobą Zespół Scrumowy.*
>
> *To, czy Definition of Done zostało spełnione, czy osiągnięty został Cel Sprintu i Cel Produktu, musi być binarne – odpowiedź powinna być „tak" lub „nie". Jako Zespół Scrum nie umawiamy się, że „spróbujemy z całych sił", tylko umawiamy się, że „zrobimy".*
>
> *Czasami dopadną nas realia i okaże się, że jednak nie potrafimy osiągnąć tego, na co się umówiliśmy. Z tego faktu płyną dwa konkretne działania: urealnienie zobowiązań („commitment") żeby były w ogóle w zasięgu naszych możliwość, a w oparciu o różnice między nimi a zobowiązaniami, jakie chcielibyśmy spełniać, stworzenie przejrzystej listy rzeczy, których trzeba się nauczyć lub które trzeba zrobić. Konsekwencją będzie plan usprawniania, który pozwoli rozszerzyć „comitmenty" do oczekiwanego przez Zespół poziomu.*
>
> *Zwracamy też uwagę, że te zobowiązania odnoszą się do artefaktów przez cały czas ich istnienia – z tego wynika obowiązek utrzymywania ich w należytym stanie odpowiadającym zdefiniowanym zobowiązaniom („commitmentom"). Nie może więc być tak, że Backlog Produktu tylko czasami odzwierciedla Cel Produktu, dla którego Backlog ten powstał. Nie możemy też mówić o Przyroście, jeśli nie istnieje produkt nadający się do użycia i spełniający kryteria zawarte w Definition of Done.*

- For the Product Backlog it is the Product Goal.
- For the Sprint Backlog it is the Sprint Goal.
- For the Increment it is the Definition of Done.

These commitments exist to reinforce empiricism and the Scrum values for the Scrum Team and their stakeholders.

Product Backlog

The Product Backlog is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the Scrum Team.

> *Ta wersja Scrum Guide (w końcu!) jasno określa Backlog Produktu jako artefakt „wyłaniający się" w miarę, jak rośnie wiedza o tym, jak podnieść wartość produktu. To oznacza, że nie jest nigdy kompletny i może zmienić się w dowolnym momencie. Ba! Powinien się zmieniać wraz ze zmianą świadomości Product Ownera co do aktualnych potrzeb interesariuszy. Pasuje to świetnie do modelu ciągłego rozwoju produktu – nie pasuje do realizacji z użyciem Scrum z góry zaplanowanych projektów.*
>
> *Scrum Guide nie używa celowo słowa „wymagania" („requirements"). Mowa jest o rzeczach, których zrobienie jest potrzebne, aby ulepszyć produkt.*
>
> *Drugie zdanie w tym akapicie należy bezwzględnie czytać w kontekście zdania pierwszego. Praca, o której tam mowa, odnosi się do wartości produktu. Nie chodzi tu w szczególności o definicję konkretnych działań w Sprincie podejmowanych przez członków Zespołu (np. zadań dla Scrum Mastera).*

Product Backlog items that can be Done by the Scrum Team within one Sprint are deemed ready for selection in a Sprint Planning event. They usually acquire this degree of transparency after refining activities. Product Backlog refinement is the act of breaking down and further defining Product Backlog items into smaller more precise items. This is an ongoing activity to add details, such as a description, order, and size. Attributes often vary with the domain of work.

W tym akapicie znajdujemy definicję pielęgnacji Backlogu Produktu („refinement"). Jest to rozbijanie, dekompozycja elementów Product Backlogu na mniejsze. Warto podkreślić, że chodzi tu ciągle o opisy potrzeb lub zmian w produkcie (rzeczy potrzebnych, by produkt był lepszy), a nie o planowanie czynności jakie Developerzy wykonają w ramach pracy nad danym elementem Backlogu. Choć można o tym podyskutować podczas pielęgnacji, to jednak miejsce takiego planowania jest na Planowaniu Sprintu.

Scrum nie wprowadza konceptu „Definition of Ready", a wręcz użycie tej praktyki wynika często z marnej komunikacji w Zespole. Może też popsuć tę komunikację, jeśli współpraca oparta na merytorycznej dyskusji zamieni się w udowadnianie sobie wzajemnie, przy pomocy listy kontrolnej, że coś jest „ready", albo że absolutnie takie nie jest.

„Ready" w tym zapisie odnosi się do subiektywnej oceny przez członków Zespołu Scrum czy konkretny element Backlogu Produktu może zostać zrealizowany w czasie jednego Sprintu. Nie implikuje to żadnych dodatkowych warunków, np. jak najbardziej mogą pozostać otwarte pytania, na które poszukiwać odpowiedzi trzeba będzie już w ramach realizacji – o ile Zespół uzna, że nie uniemożliwia to wytworzenia rozwiązania w czasie Sprintu, element jest „ready".

Zastosowanie wspomnianej wcześniej „Definition of Ready" może potencjalnie ułatwić podejmowanie takich decyzji, o ile praktyka stosowana jest jako zbiór sugestii o co zadbać. Tym niemniej szybko może pojawić się formalizm w procesie (następować będzie rytualne sprawdzanie czy element „spełnia Definition of Ready"), poza tym takie „definicje" albo są zbyt rozbudowane (więc trudne w użyciu), albo zbyt generyczne (więc nic nie dają), albo zbyt specyficzne (nie pasują do wielu elementów Backlogu Produktu). W efekcie zamiast pomagać, „Definition of Ready" najczęściej szkodzi Zespołom, prowokując do mechanistycznego sprawdzania listy kontrolnej (bez merytorycznej dyskusji), albo do traktowania „Definition of Ready" jak formalnej „bramki kontrolnej" w procesie.

The Developers who will be doing the work are responsible for the sizing. The Product Owner may influence the Developers by helping them understand and select trade-offs.

W poprzednich wersjach Scrum Guide była mowa o szacowaniu, co prowokowało do błędnych interpretacji. Szacuje się przecież różne rzeczy, np. wartość lub ryzyko – a tego zdecydowanie nie robią Developerzy (a już na pewno nie robią tego sami).

Teraz jest jasne, że chodzi o ocenę wielkości elementów Backlogu Produktu – mają tego dokonać Developerzy, którzy będą elementy te realizować. Przy czym nie jest określone, w jaki sposób ta wielkość ma być oszacowana – to pozostaje w gestii Zespołu (niekoniecznie tylko Developerów, albowiem Product Owner musi być w stanie zrozumieć i użyć wiedzy o wielkości elementów przy podejmowaniu decyzji, czy warto je realizować, czy też będą zbyt kosztowne).

Tu uwaga: czytając pierwsze zdanie tego akapitu należy pamiętać, że Scrum zakłada istnienie wspólnoty pracy (łączenie kompetencji różnych osób). Praca, której wielkość jest szacowana, związana jest z produktem jako całością – a nie konkretnymi zadaniami konkretnych osób. Tym samym niewłaściwe byłoby interpretowanie tego zapisu w ten sposób: „ustalmy, który Developer będzie realizował wymaganie, a wtedy określi on jego wielkość". Developerzy mają takiego oszacowania dokonać wspólnie, bo też będą łączyć swe kompetencje (dzieląc się pracą), by je zrealizować.

Commitment: Product Goal

The Product Goal describes a future state of the product which can serve as a target for the Scrum Team to plan against. The Product Goal is in the Product Backlog. The rest of the Product Backlog emerges to define "what" will fulfill the Product Goal.

Cel Produktu powinien być zwizualizowany w tym Backlogu tak, by każdy, kto wyświetla listę elementów, widział jaki jest obecny Cel Produktu, którego osiągnięcie ma umożliwić realizacja elementów Backlogu. To, w jakiej formie Cel Produktu umieszczony zostanie w Backlogu Produktu, pozostaje w gestii Product Ownera, podobnie jak forma i sposób prezentacji Backlogu w ogóle.

Z Celu Produktu wynika nie tylko zawartość i kolejność Backlogu Produktu, ale też Cel Sprintu każdej kolejnej iteracji aż do osiągnięcia Celu Produktu. Cel Sprintu wyjaśnia, jaką wartość (w kontekście Celu Produktu) przynosi Sprint, więc nie może dojść do rozbieżności między Celem Produktu i Celem Sprintu, nie powinno też być sytuacji, że Cel Sprintu nie ma związku z Celem Produktu.

Uwaga: narzędzia formatują nasze myślenie i sposób postępowania, więc może pojawić się pokusa, by Cel Produktu potraktować jako „epic" w narzędziu, którego używamy do zarządzania Backlogiem Produktu. To może prowadzić do fałszywej intuicji, że to „jeden z wielu Celów", lub że Cel Produktu dotyczy tylko niektórych elementów Backlogu Produktu. Tymczasem w danej chwili Cel Produktu może być tylko jeden.

Dlatego rozważając sposób wizualizacji Celu Sprintu nie należy czynić z niego jakiejkolwiek formy „super-elementu" na liście. Twórcom „narzędzi do Scruma" zapewne zajmie chwilę, zanim wymyślą w jaki sposób obsługiwać Cel Produktu w produkowanych przez siebie narzędziach.

A product is a vehicle to deliver value. It has a clear boundary, known stakeholders, well-defined users or customers. A product could be a service, a physical product, or something more abstract.

Produkt i Przyrost nie są tożsame. Produkt jest jeden, choć jego stan się zmienia w sposób ciągły w miarę, jak kolejne Sprinty podnoszą jego wartość. Natomiast Przyrostów jest wiele – powstają za każdym razem, gdy uda się wytworzyć nową wersję produktu spełniającą Definition of Done. Każdy taki Przyrost jest niezmienny – jeśli wprowadzimy do niego zmiany, powstanie kolejny, nowy Przyrost. Produkt jest zatem sumą wszystkich Przyrostów do tej pory.

The Product Goal is the long-term objective for the Scrum Team. They must fulfill (or abandon) one objective before taking on the next.

Nie może w jednym momencie być wielu Celów Produktu, choć oczywiście w cyklu życia produktu pojawiać się będą kolejne Cele Produktu. Z faktu, że należy osiągnąć ten Cel albo porzucić go jawnie przed podjęciem kolejnego Celu wynika wprost, że nie można potraktować Celu Produktu jako listy kolejnych kamieni milowych w projekcie biznesowym. Inaczej mówiąc, nie możemy zdefiniować „kolejki Celów", które mają być realizowane w jakiejś sekwencji – w danym momencie produkt ma jeden Cel.

Z faktu, że tylko jeden Cel Produktu jest związany z Backlogiem Produktu wynika natomiast, że nie da się „zlać" do jednego Backlogu wymagań dla różnych produktów (chyba, że istotnie będą one miały jeden, wspólny Cel, ale wtedy rodzi się pytanie, czy w praktyce to nie jest jeden produkt złożony z wielu mniejszych komponentów).

Co więcej, nie powinno się przechowywać na Backlogu Produktu elementów, które nie mają związku z obecnym Celem Produktu. Po osiągnięciu go (lub jawnym porzuceniu) zaczynamy najpierw od ustalenia kolejnego Celu Produktu, następnie nowy Backlog Produktu wyłania się w wyniku zastanowienia się – pracy nad tym co trzeba w produkcie zmienić, by ów nowy cel osiągnąć.

Wyłania się z tego i kolejny wniosek: w Backlogu Produktu powinny znajdować się tylko elementy opisujące to, co naprawdę chcemy zrealizować, a dokładniej to, co Product Owner uznaje za niezbędne do osiągnięcia aktualnego Celu Produktu. Niekoniecznie jest w Backlogu Produktu miejsce na elementy, które opisują coś, co „może kiedyś zrobimy", albo „przyda się, ale nie teraz" – bo nie będą one najprawdopodobniej nijak związane z bieżącym Celem Produktu, albo nie będą niezbędne, by Cel ten osiągnąć.

Sprint Backlog

The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment (how).

> *Podobnie jak to miało miejsce w przypadku Backlogu Produktu i Celu Produktu, Cel Sprintu ma być zwizualizowany w Backlogu Sprintu, wyjaśniając czemu służy zawarty w elementach Backlogu Sprintu plan działania. Oczywiście nie powinien to być „super-element" opisujący kolejną rzecz do zrobienia – forma i sposób wizualizacji Celu Sprintu w Backlogu Sprintu pozostaje w gestii Zespołu.*
>
> *Przy okazji zwracamy uwagę, że Backlog Sprintu to nie sama lista elementów Backlogu Produktu podjętych do realizacji w Sprincie, ale również plan ich realizacji (w jakiejkolwiek formie przyjętej przez Developerów). Sama taka lista albo plan działań niezwiązany z Celem Sprintu, nie może być traktowany jako Backlog Sprintu.*
>
> *W takim przypadku należy też uznać, że Planowanie Sprintu, w którym nie powstał Backlog Sprintu (a dokładniej jego pierwsza wersja w bieżącym Sprincie), nie przebiegło tak, jak powinno.*

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.

> *„Plan dla Developerów, przez Developerów" – Backlog Sprintu, choć jest widoczny, nie jest przeznaczony dla innych. Jest przeznaczony dla Developerów, ma im pomagać w orientacji podczas Sprintu. Sprint Backlog jest robiony i utrzymywany przez Developerów. Scrum Master może im w tym pomóc przede wszystkim pokazując różne możliwe narzędzia i sposoby prowadzenia Backlogu Sprintu, ale nie powinien ich w tym wyręczać.*
>
> *Zwracamy uwagę na to, że na Planowaniu Sprintu powstaje pierwsza wersja Backlogu Sprintu, którzy w ciągu Sprintu zmienia się wielokrotnie – za każdym razem, gdy sprawdzenie stanu spraw (np. w czasie Daily Scruma) ujawni taką konieczność.*
>
> *To oznacza, że nie ma potrzeby stworzenia podczas Planowania Sprintu „kompletnego i detalicznego" planu działań dla każdego Developera, zupełnie wystarczające może okazać się określenie strategii realizacji każdego z elementów podjętych do Sprintu oraz szczegółowych kroków na pierwsze dwa-trzy dni Sprintu. W miarę, jak Sprint trwa, plan ten będzie uzupełniany / dostosowywany – to pozwoli uwzględnić na bieżąco rosnącą wiedzę o tym, co jest do zrobienia.*
>
> *Taka strategia planowania oczywiście nie zadziała, jeśli Developerzy zdecydują o realizacji wielu elementów Backlogu Produktu równocześnie, rozdzielając je między sobą. W oczywisty sposób ich pierwszy Backlog Sprintu będzie o wiele bardziej detaliczny i dodatkowo musi opisywać pracę związaną z integracją pracy wykonywanej oddzielnie w jeden Przyrost. Jak już pisaliśmy powyżej, takie postępowanie nie jest efektywne.*

Commitment: Sprint Goal

The Sprint Goal is the single objective for the Sprint. Although the Sprint Goal is a commitment by the Developers, it provides flexibility in terms of the exact work needed to achieve it. The Sprint Goal also creates coherence and focus, encouraging the Scrum Team to work together rather than on separate initiatives.

Powyższy akapit jasno potwierdza to, o czym pisaliśmy wcześniej: Cel Sprintu ma być jeden, nie może być to „cel składkowy". Jeśli Zespół ma trudność z ustaleniem sensownego Celu Sprintu, być może nie jest jasne, dlaczego Backlog Produktu ma taką, a nie inną zawartość i kolejność. To bowiem z kolejności elementów Backlogu Produktu, a precyzyjniej mówiąc z powodów, dla których akurat te, a nie inne elementy znalazły się w momencie Planowania Sprintu na szczycie, wynikać powinien Cel Sprintu.

Czasami Zespoły formalnie mają jeden Cel Sprintu, ale powstaje on w wyniku zabiegu semantycznego (dodanie spójników „i", „oraz" itp., aby połączyć w jedno zdanie kilka rozbieżnych celów). W praktyce taki Cel jest bardzo słaby (nic nie znaczy). W takim przypadku Scrum Master ma obowiązek nauczyć Zespół czym jest Cel Sprintu i pomóc w osiągnięciu stanu, w którym poprawne Cele Sprintu mogą być definiowane.

Dobry Cel Sprintu zapewnia elastyczność – określa, co chcemy osiągnąć, ale nie określa bardzo precyzyjnie jak, ani co powinno być zrobione. To daje przestrzeń do zmiany zakresu prac w Sprincie, jeśli ujawni się taka potrzeba. To zaś pozwala na użycie pełni możliwości intelektualnych przez Developerów – mogą oni współpracując wypracować najlepszy sposób, najlepsze techniczne rozwiązanie, które zapewni osiągnięcie Celu Sprintu.

Zwracamy uwagę, że Cel Sprintu ustalany jest przez cały Zespół, ale jego osiągnięcie będzie możliwe wyłącznie dzięki pracy Developerów. Stąd to wyłącznie Developerzy mają prawo decydować, ile elementów Backlogu Produktu mogą podjąć do Sprintu, a ich możliwości często determinują, jaki Cel Sprintu w ogóle będzie możliwy do ustanowienia.

The Sprint Goal is created during the Sprint Planning event and then added to the Sprint Backlog. As the Developers work during the Sprint, they keep the Sprint Goal in mind. If the work turns out to be different than they expected, they collaborate with the Product Owner to negotiate the scope of the Sprint Backlog within the Sprint without affecting the Sprint Goal.

Cel Sprintu jest na Backlogu Sprintu. Scrum Guide wymaga, aby był bardzo widoczny. Oczywiście, nie określa jak konkretnie, technicznie to osiągnąć – to pozostawione jest Zespołom. W każdym razie Cel Sprintu musi być gdzieś zapisany wraz z innymi elementami Backlogu Sprintu. W przypadku „tradycyjnych" Backlogów Sprintu na ścianie jest to łatwe. W przypadku „narzędzi do Scruma" może wymagać chwili, zanim ich twórcy dostosują je odpowiednio.

Increment

An Increment is a concrete stepping stone toward the Product Goal. Each Increment is additive to all prior Increments and thoroughly verified, ensuring that all Increments work together. In order to provide value, the Increment must be usable.

Multiple Increments may be created within a Sprint. The sum of the Increments is presented at the Sprint Review thus supporting empiricism. However, an Increment may be delivered to stakeholders prior to the end of the Sprint. The Sprint Review should never be considered a gate to releasing value.

W każdym Sprincie może powstać wiele Przyrostów (które są nośnikami wartości dodanej do produktu). Co oczywiste, one nie istnieją równolegle (czyli w danym momencie nie mamy kolekcji Przyrostów, z których możemy sobie wybrać kilka), ale są tworzone sekwencyjnie – każdy nowy Przyrost obejmuje wszystkie Przyrosty wytworzone wcześniej.

To oznacza, że Przyrost stanowi de facto nową wersję produktu, której można użyć. Jeśli ta nowa wersja nie nadaje się do użycia (a to, czy tak jest, determinuje Definition of Done), nie jest Przyrostem.

Skoro zaś Przyrost jest wersją produktu, która nadaje się do użycia, można jej użyć jeszcze przed zakończeniem Sprintu.

Przegląd Sprintu nie powinien być uważany za jedyny moment, w którym można podjąć decyzję o dostarczeniu Przyrostu (aktualnej wersji produktu) użytkownikom. Powtarzające się co Sprint Przeglądy służą rytmowi empirycznej pracy nad rozwojem produktu, zapewniając moment regularnej inspekcji stanu przedsięwzięcia. Wydania produkcyjne (udostępnienie produktu użytkownikom) mogą następować częściej lub rzadziej, według ustalonego kalendarza lub ad hoc.

Tu warto przypomnieć, że jedynym sposobem na potwierdzenie wartości produktu i uzyskanie wymiernych korzyści jest wydanie (użycie) produktu, czyli udostępnienie go tym, którzy będą z niego korzystali. Wszystko inne jest jedynie promesą wartości, jej oszacowaniem – bez gwarancji, że faktycznie zostanie uzyskana.

Dlatego niewiele wspólnego z działaniem prawdziwie empirycznym będzie miała praca w krótkich Sprintach, jeśli efekty tej pracy zostaną użyte np. tylko raz na rok. Bo tylko raz na rok będzie dało się potwierdzić, czy wysiłki ponoszone przez 12 miesięcy przyniosły wartość. Oczywiście w niektórych wyjątkowych środowiskach to może być wystarczające, ale dla większości złożonych produktów, im częściej udostępniane są nowe wersje użytkownikom, tym większe prawdopodobieństwo, że budowany jest właściwy produkt.

Work cannot be considered part of an Increment unless it meets the Definition of Done.

Commitment: Definition of Done

The Definition of Done is a formal description of the state of the Increment when it meets the quality measures required for the product.

Wymóg „formalnego opisu" wydaje się sprzeczny z duchem Scruma, ale w tym przypadku chodzi o zapewnienie, że Definition of Done zostało uzgodnione i każdy może zapoznać się z zawartymi w nim warunkami. Nie może być w szczególności tak, że „Developerzy mają Definition of Done w głowie". Być może oni skutecznie będą mogli z niego korzystać (choć trudno to sprawdzić), ale takie Definition of Done nie pozwalałoby nikomu, kto nie jest Developerem (w tym Product Ownerowi i Scrum Masterowi) zrozumieć, co to znaczy, gdy Developerzy mówią, że coś ukończyli.

Definition of Done nie jest samo w sobie artefaktem, ale w praktyce powstaje jako fizyczny (albo wirtualny) byt w postaci listy, która jest widoczna dla każdego.

Warto zwrócić uwagę, że mowa jest o „miarach jakości". Sugeruje to, że elementami Definition of Done powinny być rzeczy, które można zmierzyć lub binarnie (tak/nie) ustalić. Nie ma tu miejsca na rozmyte i dyskusyjne stwierdzenia w rodzaju „jest ładne" albo „jest szybkie".

The moment a Product Backlog item meets the Definition of Done, an Increment is born.

Należy pamiętać, że istnieje jeden produkt (a nie równolegle wiele różnych produktów, które rozwijamy niezależnie od siebie). To oznacza, że ukończenie prac nad realizacją elementu Backlogu Produktu wymaga wprowadzenia poczynionych zmian do produktu. I dopiero, gdy efekt takich prac spełnia kryteria Definition of Done, można uznać go za Przyrost.

Jeśli więc Developerzy równolegle implementują dwa różne elementy Backlogu Produktu, to Przyrost nie powstanie, dopóki nie będą dysponować jedną zintegrowaną wersją, w której oba te elementy są zrealizowane – i działają.

The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. Instead, it returns to the Product Backlog for future consideration.

Od tej wersji Scrum Guide w końcu jest jasne, że nie ma sensu omawiać efektów nieukończonej pracy w czasie Przeglądu Sprintu. Oczywiście zasadne może być omówienie powodu, dla którego nie udało się pracy ukończyć, jeśli ma to wpływ na decyzje co do dalszego rozwoju produktu.

Przy okazji: skoro w czasie Przeglądu Sprintu rozmawiamy o tym, co zostało ukończone, to musimy o tym wiedzieć wcześniej. Przegląd Sprintu to nie miejsce, na którym Developerzy tłumaczą Product Ownerowi, co udało się wytworzyć. Są jednym Zespołem, więc Product Owner powinien wiedzieć co znajduje się w najnowszym Przyroście jeszcze zanim Przegląd się zacznie.

To, co nieukończone, powraca do Backlogu Produktu – w jakiej formie i na jaką pozycję, zdecyduje Product Owner. Może nawet usunąć taki element z Backlogu, jeśli już nie warto kontynuować prac nad nim. Taka nieukończona i porzucona praca może okazać się marnotrawstwem (bo poniesione zostały wysiłki na jej wykonanie, a nie przyniosła żadnej wartości) – dlatego kluczowe dla efektywności Zespołu Scrum jest nie tylko to, by Product Owner skutecznie wybierał, co warto zrobić, ale również by Developerzy potrafili tę wartość wytworzyć. Za to, by Zespół Scrum osiągnął tę efektywność i potrafił ją utrzymać (i podnosić, jeśli to możliwe), odpowiada Scrum Master.

If the Definition of Done for an increment is part of the standards of the organization, all Scrum Teams must follow it as a minimum. If it is not an organizational standard, the Scrum Team must create a Definition of Done appropriate for the product.

Czytając ostatnie zdanie w tym akapicie nie można zapominać o różnych odpowiedzialnościach osób w Zespole Scrum. Definition of Done obowiązuje dla całego zespołu, ale tylko Developerzy muszą mieć wiedzę na temat tego, jakie standardy i w jaki sposób mogą być spełnione, a także jakie mają realne możliwości ich spełnienia.

Product Owner uczestniczy w określaniu Definition of Done przynosząc wiedzę o tym, czym jest produkt, jak jest używany, jaki jest kontekst, w którym będzie funkcjonował. Np. może określić przepisy prawa, jakie muszą być spełnione, zdefiniować wymogi wydajnościowe, te związane z bezpieczeństwem itd. Produkt, który takich wymogów nie będzie spełniał, nie może zostać użyty – a zatem Developerzy muszą określić jak te kryteria zostaną zaspokojone.

Scrum Master również uczestniczy w określaniu Definition of Done dbając o to, by umożliwiało ono działanie empiryczne. Z jednej strony jest to dążenie do „binarności" Definition of Done (które musi pozwalać na jednoznaczne ustalenie czy kryteria są spełnione i powstał Przyrost, czy nie), z drugiej strony ważna jest przejrzystość samego Definition of Done – musi ono być tak samo zrozumiane przez wszystkich Developerów. Ale Scrum Master musi też zadbać o to, by Definition of Done nie prowokowało do działań sprzecznych z empiryzmem (np. aby niemożliwe było uznanie za ukończonej pracy, jeśli zrobiona została jedynie analiza, a reszta działań będzie podejmowana w kolejnych Sprintach).

Może zdarzyć się, że Definition of Done produktu nadającego się do użycia jest nieosiągalne dla Zespołu (np. ze względu na brak kompetencji wśród Developerów lub narzędzi). W takim przypadku Definition of Done powinno odzwierciedlać to, co Zespół faktycznie jest w stanie zrobić, zaś wszystkie pozostałe klauzule determinować będą to, czego Zespół musi się nauczyć oraz co trzeba zrobić, by produktu użyć. To zapewnia przejrzystość stanu produktu i praktyk Zespołu.

Oczywiście taki stan jest skrajnie niepożądany, ponieważ w praktyce tak działający Zespół nie wytwarza wartości – by móc użyć tego, co wytwarza, wciąż konieczne będzie wykonanie jakiejś dalszej pracy. Jeśli istnieje możliwość rozszerzenia kompetencji Zespołu tak, by Definition of Done opisywało używalny produkt (np. poprzez zmianę składu Zespołu), należy z niej skorzystać. Długoterminowe budowanie rozwiązań, których nie da się użyć, nie ma nic wspólnego z empiryzmem – bo jedynie użycie produktu pozwala ostatecznie potwierdzić jego wartość.

The Developers are required to conform to the Definition of Done. If there are multiple Scrum Teams working together on a product, they must mutually define and comply with the same Definition of Done.

End Note

Scrum is free and offered in this Guide. The Scrum framework, as outlined herein, is immutable. While implementing only parts of Scrum is possible, the result is not Scrum. Scrum exists only in its entirety and functions well as a container for other techniques, methodologies, and practices.

„Niezmienność" Scruma nie odnosi się do tego, że został on zdefiniowany raz i nie będzie się zmieniał. Z czasem być może pojawi się kolejna wersja Scrum Guide.

„Niezmienność", o której mowa w tym akapicie, oznacza, że nie można jednocześnie zmienić czy usunąć elementów Scruma i nadal mówić, że stosuje się tę metodę. Jeśli dokonamy takich zmian (które być może będą dawały bardzo dobre rezultaty w naszej firmie czy zespole), posługiwać się zaczniemy naszą własną metodą. To świetnie! Nie nazywajmy jej jednak Scrumem, bo to, czym jest Scrum, definiuje Scrum Guide.

To ważne szczególnie dla Scrum Masterów, którzy często muszą borykać się z niewłaściwym rozumieniem Scruma, przyniesionym do Zespołu z organizacji, gdzie stosowano coś, co Scrumem nie jest, niestety wciąż posługując się tą nazwą.

Acknowledgements

People

Of the thousands of people who have contributed to Scrum, we should single out those who were instrumental at the start: Jeff Sutherland worked with Jeff McKenna and John Scumniotales, and Ken Schwaber worked with Mike Smith and Chris Martin, and all of them worked together. Many others contributed in the ensuing years and without their help Scrum would not be refined as it is today.

Scrum Guide History

Ken Schwaber and Jeff Sutherland first co-presented Scrum at the OOPSLA Conference in 1995. It essentially documented the learning that Ken and Jeff gained over the previous few years and made public the first formal definition of Scrum.

The Scrum Guide documents Scrum as developed, evolved, and sustained for 30-plus years by Jeff Sutherland and Ken Schwaber. Other sources provide patterns, processes, and insights that complement the Scrum framework. These may increase productivity, value, creativity, and satisfaction with the results.

The complete history of Scrum is described elsewhere. To honor the first places where it was tried and proven, we recognize Individual Inc., Newspage, Fidelity Investments, and IDX (now GE Medical).

Historia Scruma jest bardziej obszerna niż to, co napisano powyżej. Jednak ważne jest, aby zauważyć, że Scrum nie jest wynikiem akademickich rozważań teoretyków. Scrum powstał w praktyce, w pracy z zespołami nad oprogramowaniem jako narzędzie dla zaangażowanych ludzi, którzy przy jego pomocy rozwiązywali problemy, z jakimi się zmagali, i którym Scrum pozwalał działać efektywniej.

Kiedy Scrum stał się bardziej popularny, również inni dzielili się swoimi spostrzeżeniami z autorami Scrum Guide, o czym piszą oni powyżej. Obecny kształt Scruma jest więc zebraniem, „destylatem" doświadczeń bardzo wielu zespołów. Właśnie dlatego odpowiednio zastosowany Scrum jest taki skuteczny.