

Scrum to jedno z najpopularniejszych ram postępowania (frameworków) zwinnych (Agile), pomagających zespołom i organizacjom w tworzeniu wartości poprzez adaptacyjne rozwiązywanie złożonych problemów. Zamiast planować cały projekt z góry, Scrum dzieli pracę na krótkie, powtarzalne cykle (Sprinty), co pozwala na szybkie reagowanie na zmiany i ciągłe doskonalenie produktu.

Opiera się na empiryzmie, co oznacza, że wiedza wynika z doświadczenia, a decyzje podejmowane są na podstawie tego, co jest znane i widoczne. Trzy filary empiryzmu w Scrumie to:

- **Przejrzystość:** Wszyscy zaangażowani w proces mają dostęp do tych samych informacji.
- **Inspekcja:** Regularne sprawdzanie postępów prac i artefaktów w celu wykrycia niepożądanych odchyleń.
- **Adaptacja:** Wprowadzanie zmian i poprawek, gdy inspekcja wykaże, że proces lub produkt wymykają się spod kontroli.

Zgodnie z oficjalnym *Scrum Guide*, Scrum składa się z trzech głównych kategorii elementów: **Odpowiedzialności**, **Wydarzeń** oraz **Artefaktów**.

1. Odpowiedzialności (Role) w Scrumie

W Scrumie nie ma tradycyjnej hierarchii. Zespół Scrumowy (Scrum Team) to spójna jednostka profesjonalistów, w której wyróżniamy trzy konkretne role:

- **Właściciel Produktu (Product Owner):** Reprezentuje głos klienta i biznesu. Jest jedną osobą, która odpowiada za maksymalizację wartości dostarczanego produktu oraz za zarządzanie Rejestrem Produktu (dbanie o to, by był zrozumiały i odpowiednio spriorytetyzowany).
 - **Scrum Master:** Dbą o to, aby zespół rozumiał i stosował zasady Scruma. Jest liderem służebnym (servant leader) – usuwa przeszkody (tzw. impedimenty) blokujące pracę zespołu, facylituje spotkania i pomaga organizacji w zrozumieniu zwinnego podejścia.
 - **Deweloperzy (Developers):** Specjaliści (nie tylko programiści, ale też testerzy, projektanci, analitycy itp.), którzy wykonują faktyczną pracę. Są samoorganizujący się i samodzielnie decydują o tym, w jaki sposób przekształcić zaplanowaną pracę w gotowy Przyrost.
-

2. Wydarzenia (Ceremonie) Scrumowe

Wszystkie wydarzenia w Scrumie mają z góry określony maksymalny czas trwania (timeboxing) i służą zapewnieniu regularności, eliminując potrzebę organizowania innych, niezaplanowanych spotkań.

- **Sprint:** "Serce" Scruma. To zamknięty przedział czasu (zazwyczaj od 1 do 4 tygodni), w którym powstaje gotowa do użycia i wartościowa część produktu. Wszystkie inne wydarzenia odbywają się w ramach Sprintu.
 - **Planowanie Sprintu (Sprint Planning):** Spotkanie rozpoczynające Sprint. Zespół ustala na nim odpowiedź na trzy pytania: *Dlaczego* ten Sprint jest wartościowy (Cel Sprintu)?, *Co* może zostać ukończone w tym Sprincie? oraz *Jak* ta praca zostanie wykonana?
 - **Codzienny Scrum (Daily Scrum):** Krótkie, codzienne spotkanie (maksymalnie 15 minut) przeznaczone wyłącznie dla Deweloperów. Służy do synchronizacji działań, inspekcji postępów w kierunku Celu Sprintu i ewentualnego dostosowania planu na najbliższe 24 godziny.
 - **Przegląd Sprintu (Sprint Review):** Przedostatnie wydarzenie w Sprincie. Zespół prezentuje interesariuszom (klientom, zarządowi, użytkownikom) to, co udało się zbudować. Zbierana jest informacja zwrotna, która wpływa na modyfikację Rejestru Produktu na przyszłość.
 - **Retrospektywa Sprintu (Sprint Retrospective):** Ostatnie wydarzenie, zamykające Sprint. Zespół spotyka się we własnym gronie, aby omówić, jak przebiegała praca pod kątem relacji międzyludzkich, procesów i narzędzi. Celem jest identyfikacja obszarów do poprawy i zaplanowanie konkretnych usprawnień na kolejny Sprint.
-

3. Artefakty Scrumowe

Artefakty to fizyczne lub wirtualne reprezentacje pracy, które zapewniają przejrzystość informacji. Każdy z nich zawiera powiązane **zobowiązanie (commitment)**, które nadaje cel i pomaga w mierzeniu postępów.

- **Rejestr Produktu (Product Backlog):** Uporządkowana i ewoluująca lista wszystkiego, co jest potrzebne w produkcie (nowe funkcjonalności, poprawki błędów, wymagania techniczne). Jest to jedyne źródło pracy dla zespołu.
 - **Zobowiązanie: Cel Produktu** – długoterminowy cel biznesowy, do którego dąży zespół i z którym spójny musi być cały Rejestr Produktu.
- **Rejestr Sprintu (Sprint Backlog):** Zbiór elementów wybranych z Rejestru Produktu na bieżący Sprint, wraz z planem ich dostarczenia. Jest to plan stworzony przez Deweloperów i dla Deweloperów.
 - **Zobowiązanie: Cel Sprintu** – krótkoterminowy cel wyznaczający kierunek prac w danym Sprincie. Promuje współpracę i skupienie.
- **Przyrost (Increment):** Użyteczna, działająca wersja produktu. Jest to suma wszystkich elementów ukończonych podczas aktualnego Sprintu oraz wartości Przyrostów ze wszystkich poprzednich Sprintów.
 - **Zobowiązanie: Definicja Ukończenia (Definition of Done)** – formalny zbiór kryteriów (np. kod przetestowany, dokumentacja zaktualizowana, brak krytycznych błędów), które muszą zostać spełnione, aby pracę można było uznać za faktycznie zrealizowaną i gotową do wydania.

Epiki, User Story czy Taski nie są pojęciami opisanymi w oficjalnym Scrum Guide. Scrum mówi po prostu o "Elementach Rejestru Produktu" (Product Backlog Items - PBI).

Jednak w praktyce niemal **każdy** zespół Scrumowy zapożycza te pojęcia z innej metodyki (głównie z Extreme Programming - XP), aby łatwiej organizować i kategoryzować pracę. Tworzą one spójną hierarchię – od wielkich wizji po drobne zadania techniczne.

Oto jak dokładnie wygląda ta hierarchia w codziennej pracy:

1. Epik (Epic)

Epik to duży, złożony fragment pracy, który jest zbyt obszerny, aby można go było ukończyć w ramach jednego Sprintu. To zazwyczaj grubszy temat, nowa duża funkcjonalność lub moduł systemu.

- **Charakterystyka:** Epiki są rozbijane na mniejsze, łatwiejsze do zarządzania kawałki (najczęściej na Historyjki Użytkownika), zanim zespół weźmie się za ich realizację.
- **Przykład:** Jeśli budujesz sklep internetowy, Epikiem może być "Wdrożenie nowego systemu płatności" albo "Przebudowa panelu użytkownika".

2. Historyjka Użytkownika (User Story)

To najpopularniejszy format opisywania wymagań w Agile. Jest to krótki, prosty opis funkcjonalności napisany z perspektywy osoby, która będzie z niej korzystała (najczęściej klienta lub użytkownika końcowego). User Story musi być na tyle małe, aby **dało się je ukończyć w jednym Sprincie**.

- **Format:** Zazwyczaj pisze się je według sprawdzonego szablonu:

Jako [typ użytkownika], chcę [wykonać jakąś akcję], aby [osiągnąć konkretną korzyść/cel].

- **Przykład:** (W ramach Epiku "Wdrożenie nowego systemu płatności") -> "Jako klient sklepu, chcę móc zapłacić za pomocą BLIKa, aby szybciej i wygodniej sfinalizować moje zakupy."
- **Kryteria Akceptacji (Acceptance Criteria):** Każde Story powinno mieć listę warunków, które muszą zostać spełnione, aby uznać je za gotowe (np. "Przycisk BLIK jest widoczny w koszyku", "Po wpisaniu błędnego kodu pojawia się komunikat o błędzie").

3. Zadanie (Task / Sub-task)

O ile User Story skupia się na wartości biznesowej (CZYM jest funkcja i dla KOGO), o tyle Zadanie to już techniczny podział pracy wskazujący, **JAK** to zostanie zrobione.

- **Charakterystyka:** Zadania to konkretne, mniejsze kroki (często zajmujące od kilku godzin do maksymalnie 1-2 dni), które Deweloperzy muszą wykonać, aby dostarczyć całe User Story.
- **Przykłady:** (Dla Historyjki o płatności BLIKiem) -> "Zaprojektować wygląd przycisku BLIK", "Napisać zapytanie do bazy danych", "Skonfigurować połączenie z API operatora płatności", "Przeprowadzić testy bezpieczeństwa".

4. Błąd (Bug / Defect)

To po prostu usterka lub nieprawidłowe działanie systemu, które zostało wykryte (przez testerów, klientów lub sam zespół) i wymaga naprawy. Często traktuje się je na równi z User Story pod kątem planowania pracy w Sprincie.

Jak to wszystko układa się w Scrumie?

1. **W Rejestrze Produktu (Product Backlog)** Właściciel Produktu gromadzi Epiki oraz Historyjki Użytkownika. Te na samej górze (najważniejsze) to precyzyjnie opisane i wycenione Historyjki. Te na samym dole to często jeszcze duże, niejasne Epiki, które czekają na doprecyzowanie w przyszłości.
2. Na **Planowaniu Sprintu (Sprint Planning)**, zespół wybiera z góry Rejestru Produktu tyle Historyjek (User Stories), ile jest w stanie zrealizować w nadchodzącym cyklu.
3. Następnie Deweloperzy rozbijają te Historyjki na mniejsze **Zadania (Taski)**, tworząc w ten sposób **Rejestr Sprintu (Sprint Backlog)**.