



Metodologie prowadzenia projektu



Dobór metodologii prowadzenia projektu

Wiele projektów jest realizowanych z powodzeniem, lecz niemal tyle samo kończy się porażką. Dlaczego tak się dzieje? Główne przyczyny są dwie: czynnik ludzki i pieniądze. Odpowiednie zarządzanie projektem to odpowiednie zarządzanie zespołem. Pieniądze pomagają w działaniach lub ograniczają je. Dlatego tak ważny jest dobór odpowiedniej metodologii prowadzenia projektu, czyli zbioru zasad i dobrych praktyk, według których zespół projektowy będzie realizował swoje zadania. Wiedza o stosowanym modelu jest niezbędna do właściwego planowania pracy zarówno dla testera, jak i programisty.

Rodzaje metodologii prowadzenia projektu



Rodzaje metodologii prowadzenia projektu

Metodologie wykorzystywane najczęściej w prowadzeniu projektów to:

- model kaskadowy
 - model prototypowy
- model spiralny
- PRINCE2
- metodyki zwinne
 - scrum
 - kanban
 - scrumban

Model kaskadowy



Model kaskadowy (waterfall)

Zarządzanie projektem w modelu kaskadowym to **sekwencyjna metoda zarządzania projektem**, w której projekt podzielony jest na odrębne fazy. Każda faza rozpoczyna się dopiero po ukończeniu poprzedniej. Ten sposób zarządzania projektem narodził się w branży produkcyjnej i budowlanej, gdzie proces produkcyjny nie może ruszyć dalej, dopóki nie zostanie osiągnięty poprzedzający kamień milowy. Na przykład nie można zbudować ścian domu, zanim nie wylano jego fundamentów. Pomimo tego, że zarządzanie projektem w modelu kaskadowym wywodzi się z branży produkcyjnej, metoda ta została od tamtego czasu dostosowana do potrzeb wielu branż, w tym do tworzenia oprogramowania.

Model kaskadowy (waterfall)

Model kaskadowy często przedstawiany jest w formie schematu blokowego lub wykresu Gantta. Model ten nazywany jest kaskadowym, ponieważ każde zadanie przechodzi kaskadą w kolejne. Na wykresie Gantta widać, jak poprzednia faza spada na kolejną.





Model kaskadowy (waterfall)

W modelu kaskadowym kolejność etapów jest ważna. Niedopuszczalne jest realizowanie kilku etapów równolegle czy też pomijanie niektórych z nich. Formalne zakończenie każdej fazy to przede wszystkim sprawdzenie i upewnienie się, że wszystkie zaplanowane prace zostały wykonane. Jeżeli w jednym z etapów wystąpiły błędy, których naprawa jest zbyt skomplikowana i pociąga za sobą zmiany w wymaganiach, to zawsze jest możliwy powrót do wcześniejszych faz. Takie cofanie niesie za sobą ryzyko opóźnienia realizacji projektu czy zwiększenia kosztów, ale jest dopuszczalne i w razie konieczności stosowane.



Fazy w modelu kaskadowym

Kaskadowa metoda zarządzania projektem obejmuje 6 następujących kroków:

- określenie wymagań
- projektowanie systemu
- implementacja
- testowanie
- wdrożenie
- konserwacja



Określenie wymagań

Jest to wstępny proces planowania, w którym zespół gromadzi tak dużo informacji, jak to możliwe, aby zapewnić powodzenie projektu. Ta faza wymaga wielu przemyśleń, ponieważ zadania w metodzie kaskadowej są zależne od poprzedzających kroków. Proces planowania ma kluczowe znaczenie dla modelu kaskadowego i z tego powodu zajmuje największą część osi czasu projektu.

Aby jak najlepiej skorzystać z tego modelu, opracuj szczegółowy [plan projektu](#), który będzie zawierać objaśnienie każdej fazy projektu, a także wykaz niezbędnych zasobów oraz opis zadań każdego z członków zespołu. Ten dokument jest powszechnie nazywany opisem wymagań projektowych.



Określenie wymagań

Faza określania wymagań powinna zakończyć się opracowaniem bardzo jasnego i zrozumiałego konspektu projektu od jego rozpoczęcia do zakończenia, który będzie opisywać:

- analizę potrzeb klienta
- każdy etap procesu
- kto pracuje nad poszczególnymi etapami
- kluczowe zależności
- wymagane zasoby
- oś czasu wskazującą, ile będzie trwać każdy etap projektu.



Projektowanie systemu

W procesie tworzenia oprogramowania faza projektowania obejmuje określenie przez zespół projektowy z jakiego sprzętu będzie korzystać oraz ustalenie innych szczegółowych kwestii, takich jak wybór języka programowania i interfejsu użytkownika.

Faza projektowania systemu obejmuje dwa etapy, a mianowicie etap projektowania ogólnego oraz etap projektowania szczegółowego. Na etapie projektowania ogólnego zespół buduje szkielet tego, w jaki sposób oprogramowanie będzie działać i w jaki sposób będzie odbywał się dostęp do informacji. Na etapie projektowania szczegółowego zespół buduje konkretne składniki oprogramowania.



Projektowanie systemu

Członkowie zespołu, którzy tworzą oprogramowanie, korzystając z modelu kaskadowego, powinni dokumentować każdy krok, aby w miarę postępu projektu zespół mógł prześledzić, co zostało wcześniej zrobione.



Implementacja

Jest to etap, na którym przystępuje się do działania. Na podstawie opisu wymagań z pierwszego etapu oraz procedury projektowania systemu z etapu drugiego, zespół rozpoczyna pełen proces tworzenia, aby zbudować oprogramowanie zgodnie z konspektem powstałym w ramach etapu określenia wymagań i etapu projektowania systemu.



Testowanie

Na tym etapie zespół deweloperski przekazuje projekt zespołowi ds. testowania i kontroli jakości. Testerzy wyszukują błędy, które muszą zostać naprawione przed wdrożeniem projektu.

Testerzy powinni w jasny sposób dokumentować wszystkie problemy, które napotykają podczas kontroli jakości. W przypadku gdy inny deweloper napotka podobny błąd, będzie mógł skorzystać z wcześniejszej dokumentacji, aby naprawić ten problem.



Wdrożenie i konserwacja

W przypadku projektów wdrożeniowych ten etap obejmuje dystrybucję oprogramowania do użytkowników końcowych. W przypadku innych branż jest to etap, kiedy produkt końcowy wprowadzany jest na rynek i trafia do klientów końcowych.

Już po wdrożeniu projektu mogą pojawić się sytuacje, w których zostanie wykryty nowy błąd lub wymagana będzie aktualizacja oprogramowania. Jest to tak zwana faza konserwacji. W obszarze tworzenia oprogramowania powszechne jest, że praca w tej fazie wykonywana jest bez przerwy.



Kiedy korzystać z modelu kaskadowego?

Model kaskadowy jest powszechną formą zarządzania projektami, ponieważ umożliwia on dokładne planowanie i szczegółowe dokumentowanie. Ta metoda nie jest jednak odpowiednia dla każdego projektu. Oto kilka przykładów, kiedy korzystać z tego sposobu zarządzania projektem :

- Projekt ma jasno zdefiniowany cel końcowy

Jedną z mocnych stron metody kaskadowej jest to, że pozwala ona na wyznaczenie jasnej ścieżki z punktu A do punktu B. Projekty o jasno określonych celach są odpowiednie dla metody kaskadowej, ponieważ kierownicy projektów mogą spojrzeć na projekt z perspektywy celu i dzięki temu stworzyć jasną oraz dobrze opisaną ścieżkę ze wskazaniem wszystkich niezbędnych wymagań.



Kiedy korzystać z modelu kaskadowego?

- Nie ma ograniczeń budżetowych i czasowych

Jeżeli Twój projekt nie ma żadnych ograniczeń budżetowych lub czasowych, członkowie zespołu mogą poświęcić maksymalnie dużo czasu na przepracowanie faz określania wymagań i projektowania systemu. Mogą dostosowywać wymagania projektowe tak długo, jak zechcą, aż do momentu, w którym będą mieli opracowany dobrze przemyślany i zdefiniowany plan projektu.



Kiedy korzystać z modelu kaskadowego?

- Tworzenie powtarzalnych procesów

W modelu kaskadowym konieczne jest dokumentowanie praktycznie każdego elementu tego procesu. Dzięki temu łatwo można wytłumaczyć przebieg projektu nowemu członkowi zespołu – każdy krok jest jasno opisany, co pozwala na odtworzenie procesu.

Tworzenie powtarzalnych procesów pomaga również w szkoleniu nowych członków zespołu odnośnie tego, co dokładnie należy robić w podobnych projektach. Sprawia to, że model kaskadowy to efektywny sposób zarządzania projektami obejmującymi standaryzację procesów.



Zalety modelu kaskadowego

- + Konsekwentna dokumentacja ułatwia cofnięcie się do wcześniejszych etapów projektu
- + Śledzenie postępów jest łatwe
- + Członkowie zespołu mogą efektywnie zarządzać czasem

Model prototypowy



Model prototypowy

Model prototypowy jest przeznaczony dla projektu, w którym klient i zespół chcą przygotować prostą, szybko implementację rozwiązania lub pewnej funkcjonalności. Ma to na celu weryfikację danego rozwiązania w praktyce oraz umożliwia klientowi sprawdzenie, czy zespół poprawnie rozumie jego wizję. Model ten składa się z następujących etapów:

1. Ogólne określenie wymagań
2. Budowa prototypu
3. Weryfikacja prototypu przez klienta / użytkownika końcowego
4. Wyczerpujące określenie wymagań
5. Realizacja pewnego systemu zgodnie z modelem kaskadowym



Model prototypowy

Głównym celem realizacji prototypu jest szczegółowe określenie wymagań, to jest wykrycie i uniknięcie:

- Nieporozumień pomiędzy klientem a twórcami systemu
- Brakujących funkcji
- Trudnych, wręcz niewykonalnych usług
- Braków w specyfikacji wymagań

Prototyp nie jest częścią przyszłego pełnego systemu. Jest raczej zapowiedzią tego, co pełny system będzie zawierać. W momencie akceptacji prototypu przez klienta zespół developerski przystępuje do pracy nad pewnym systemem.



Zalety modelu prototypowego

- + Możliwość szybkiej demonstracji pracującej wersji systemu
- + Możliwość przeprowadzenia szkoleń dla użytkowników, zanim zbudowane zostanie pełny system

Model spiralny



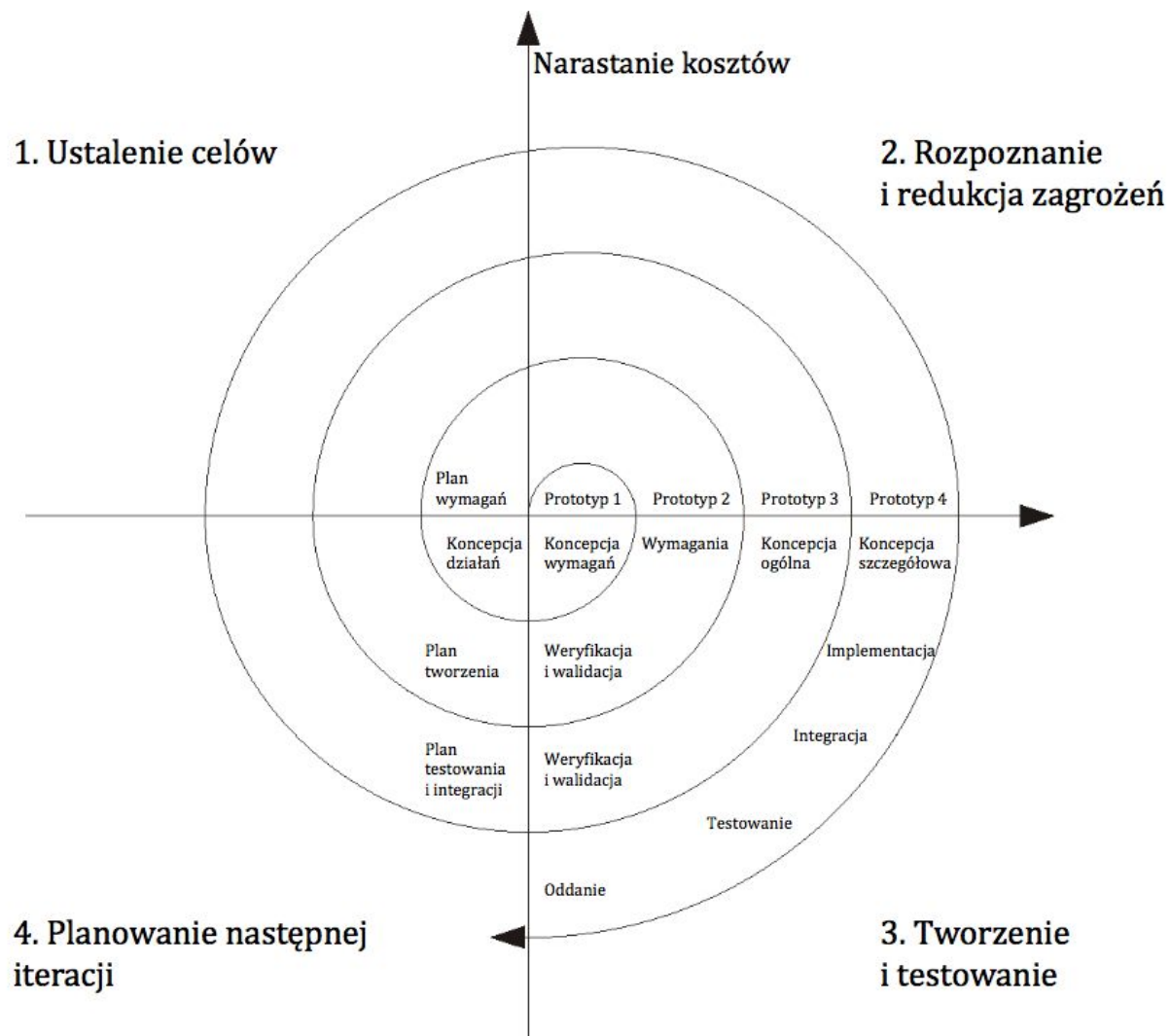
Model spiralny

Model spiralny wywodzi się z połączenia modelu kaskadowego oraz prototypowego. Głównym założeniem modelu jest stwierdzenie, że każda następna wersja powstającego systemu/produktu będzie stworzona na podstawie wyników wytworzenia wcześniejszych wersji (prototypów), dzięki czemu z każdym kolejnym przebiegiem procesu realizującego końcowy rezultat musi być coraz lepszy. Proces ten ma formę spirali, w której każda pętla odzwierciedla kolejne fazy procesu, które umożliwiają ciągłe doskonalenie produktu oraz zarządzanie ryzykiem.



Fazy modelu spiralnego

1. **Planowanie** — obejmuje oszacowanie kosztów, harmonogramu i zasobów iteracji. Obejmuje to również zrozumienie wymagań systemowych dotyczących ciągłej komunikacji pomiędzy analitykiem systemowym a klientem
2. **Analiza ryzyka** — identyfikacja potencjalnego ryzyka odbywa się w trakcie planowania i finalizowania strategii ograniczania ryzyka
3. **Inżynieria** — obejmuje testowanie, kodowanie i wdrażanie oprogramowania u klienta
4. **Ocena** — ocena oprogramowania przez klienta. Obejmuje także identyfikację i monitorowanie ryzyk, takich jak przesunięcie harmonogramu i przekroczenie kosztów





Kiedy korzystać z modelu spiralnego?

Metoda spiralna jest szczególnie przydatna w projektach:

- Z dużą niepewnością co do wymagań użytkownika.
- O wysokim ryzyku technologicznym, gdzie prototypowanie może pomóc zredukować ryzyko.
- Które są złożone i wymagają precyzyjnego zarządzania ryzykiem.

Najczęściej stosuje się ją przy tworzeniu dużych systemów informatycznych, gdzie ważne jest ciągłe dostosowywanie produktu.

PRINCE2



PRINCE2

PRINCE2 (Projects IN Controlled Environments 2) to rozbudowana i strukturalna metodologia zarządzania projektami, która została opracowana przez rząd brytyjski i jest stosowana na całym świecie. Charakteryzuje się formalnym podejściem, naciskiem na kontrolę procesów oraz szczegółową dokumentację na każdym etapie projektu.



Główne założenia PRINCE2

Podział projektu na etapy: Każdy projekt jest podzielony na etapy, co pozwala na lepsze planowanie, kontrolę i ocenę postępów. Po każdym etapie przeprowadzany jest przegląd, aby upewnić się, że projekt zmierza w dobrym kierunku.

Definicja ról i odpowiedzialności: PRINCE2 kładzie duży nacisk na precyzyjne określenie ról oraz ich odpowiedzialności, co minimalizuje ryzyko nieporozumień i umożliwia bardziej efektywne zarządzanie.

Dostosowywanie do projektu: Metodologia PRINCE2 nie jest tradycyjną strukturą; można ją dostosowywać do specyfiki projektu, co pozwala na jej elastyczne stosowanie zarówno w małych, jak i dużych projektach.



Główne założenia PRINCE2

Zarządzanie ryzykiem i zmianami: PRINCE2 wymaga systematycznego identyfikowania, monitorowania i zarządzania ryzykiem oraz kontrolowania zmian, aby utrzymać projekt na właściwym torze.

Koncentracja na produkcie: PRINCE2 zakłada, że każdy projekt ma jasno określone, mierzalne produkty, które muszą spełniać zdefiniowane wcześniej kryteria jakościowe.

Elementy PRINCE2

PRINCE2 składa się z 4 realizowanych elementów:

- zasad
- tematów
- procesów
- środowiska projektu





7 zasad PRINCE2

- Ciągła zasadność biznesowa

Zasada ta odpowiada na pytanie "Dlaczego realizujemy dany projekt?" Odpowiedź została przedstawiona w **uzasadnieniu biznesowym**, który jest jednym z najważniejszych produktów zarządzania w metodzie PRINCE2. Business Case to żywy dokument, który jest ciągle sprawdzany i poprawiany przez cały czas trwania projektu. Jeśli w jakimś momencie projektu uzasadnienie przestanie być opłacalne, projekt zostanie zamknięty. Metoda PRINCE2 wymaga, aby projekt był stale usprawniany, zapewniając, że zasoby nie zostaną nigdy zmarnowane.



7 zasad PRINCE2

- **Korzystanie z doświadczeń**

Wymaga analizy wniosków wyciągniętych z poprzednich projektów przed rozpoczęciem nowego. Celem jest uniknięcie tego, co nie działało w przeszłości, i wdrożenie tego, co się sprawdziło. Jest to możliwe dzięki uczeniu się przez cały czas trwania obecnego projektu i ciągłemu wprowadzaniu zmian w celu poprawy.



7 zasad PRINCE2

- **Zdefiniowane role i obowiązki**

Podstawowym celem tej zasady jest zapewnienie zaangażowania odpowiednich osób w projekt. Wielokrotnie projekty wymagają zespołów interdyscyplinarnych, co może prowadzić do konfliktów interesów i konkurujących celów. Może to utrudnić postęp w projekcie. Aby odnieść sukces, musi istnieć w pełni zdefiniowana struktura zespołu projektowego, a każda osoba powinna w pełni zrozumieć swoją rolę w projekcie i obowiązki.



7 zasad PRINCE2

- **Zarządzanie etapowe**

Ta zasada ma na celu podzielenie projektu na praktyczne części. Podstawową ideą jest posiadanie długoterminowej wizji projektu na wysokim poziomie i posiadanie dość szczegółowego planu krótkoterminowego. Pozwala to na dużą kontrolę nad projektem; pod koniec każdego etapu podejmuje się decyzje związane z zapewnieniem kontynuacji i uzasadnienia projektu.



7 zasad PRINCE2

- Zarządzanie z wykorzystaniem tolerancji

Wymaga ustalenia poziomu tolerancji względem sześciu zmiennych, w tym: czasu, kosztów, jakości, zakresu, ryzyka i korzyści. Jeśli przewiduje się, że projekt przekroczy którąkolwiek z sześciu, zasada ta zapewnia, że zostanie zwiększona kontrola. Pozwala to podejmować decyzje na właściwej warstwie w strukturze zarządzania.



7 zasad PRINCE2

- Skoncentrowanie na produktach

Metoda PRINCE2 wymaga utworzenia opisu produktu. Zasada ta zapewnia zwiększenie prawdopodobieństwa spełnienia oczekiwań interesariuszy wobec produktów, zwłaszcza pod względem jakości.



7 zasad PRINCE2

- **Dostosowanie do warunków projektu**

Metoda PRINCE2 ma charakter ogólny i uniwersalny, dlatego wymaga dostosowania jej nie tylko do organizacji, ale także do indywidualnego projektu. Pomaga to uniknąć dwóch wspólnych skrajności: sztywnej metodologii, która jest ślepo przestrzegana w każdym szczególe, oraz podejścia, w którym brakuje ustalonej struktury podczas zarządzania projektami. Celem jest uzyskanie zrównoważonego i zdroworozsądkowego podejścia dostosowanego do każdej organizacji i indywidualnego projektu.



7 tematów PRINCE2

Tematy są tworzone na początku projektu, a następnie monitorowane przez cały czas jego trwania.

1. Uzasadnienie biznesowe — wiąże się z zasadą „ciągłej zasadności biznesowej”. Ten temat dostarcza wiedzy na o tym, czy projekt jest opłacalny i możliwy do zrealizowania.

2. Organizacja — wiąże się z zasadą „określonych ról i obowiązków”. Temat organizacji wymaga od kierowników projektów posiadania dokumentacji w zakresie ról i obowiązków wszystkich osób.



7 tematów PRINCE2

3. Jakość — wiąże się z zasadą „koncentracji na produktach”. Jakość może być pojęciem dość abstrakcyjnym, dlatego określenie jej na początku projektu ma kluczowe znaczenie dla utrzymania go na właściwym kursie.

4. Plany — plan to sposób osiągnięcia celów. Koncentruje się na produktach, skali czasowej, kosztach, jakości i korzyściach.

5. Ryzyko — celem tego tematu jest identyfikacja, ocena i kontrola niepewnych zdarzeń w trakcie projektu. Zapisuje się je w rejestrze ryzyka. Ryzyka negatywne zwane są zagrożeniami, natomiast pozytywne szansami.



7 tematów PRINCE2

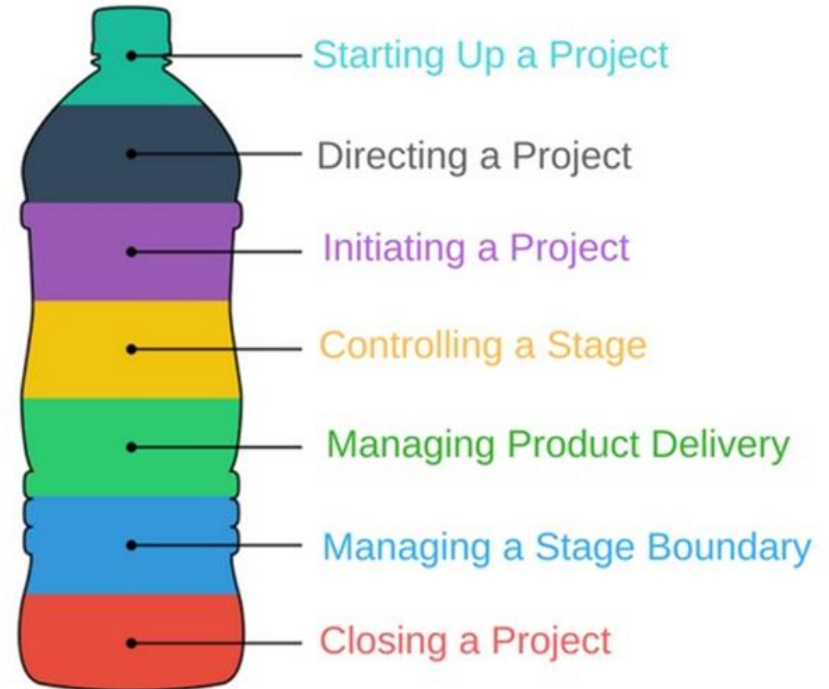
6. Zmiany — temat dotyczy postępowania z wnioskami dotyczącymi zmian i zagadnieniami pojawiającymi się podczas projektu. Nie chodzi o to, by zapobiegać zmianom, ale żeby uzgodnić je, zanim zostaną przeprowadzone.

7. Postępy — dotyczą śledzenia projektu. Śledzenie pozwala kierownikom projektów sprawdzać i kontrolować swoje położenie względem planu. Bez tego – lub innych tematów – projekty nie tylko mogą wymknąć się spod kontroli, ale nie śledząc ich, można w ogóle nie zdawać sobie sprawy, że tak się dzieje.

7 procesów PRINCE2

- przygotowanie projektu
- inicjowanie projektu
- zarządzanie strategiczne projektem
- sterowanie etapem
- zarządzanie wytwarzaniem produktów
- zarządzanie końcowym etapem
- zamykanie projektu

7 Processes of Prince2





PRINCE2 — role w zarządzaniu projektem

Kierownik projektu

- przejmuje odpowiedzialność za organizację i kontrolę
- wybiera osoby, które wykonają pracę nad projektem i upewnia się, czy jest to zrobione poprawnie i na czas
- opracowuje plany projektu, opisujące czynności zespołu projektu i planowaną datę realizacji.

Klient lub wykonawca finansuje projekt.



PRINCE2 — role w zarządzaniu projektem

Użytkownik to osoba, która zamierza korzystać z produktu lub wyniku projektu. W przypadku niektórych projektów klient i użytkownik może być tą samą osobą.

Dostawca lub specjalista posiada umiejętności i wiedzę niezbędną do wykonania prac projektowych.



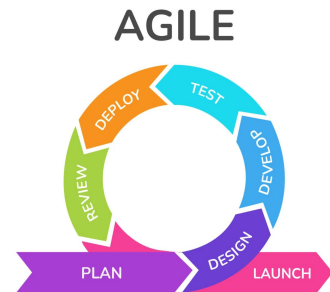
PRINCE2 — role w zarządzaniu projektem

Komitety Sterujące

Każdy projekt PRINCE2 ma swój Komitet, w skład którego wchodzi klient (lub organ wykonawczy), reprezentant strony użytkownika i reprezentant dostawcy lub specjalisty. Szef projektu regularnie składa raport Komitetowi o postępach i wszelkich przewidywanych problemach. Komitet projektu ma za zadanie podejmować decyzje niezbędne dla kierownika projektu, by rozwiązywać problemy i kontynuować pracę nad projektem. Współpraca wszystkich osób wymaga organizacji i koordynacji, aby projekt był zrealizowany na czas, wysokiej jakości i w ramach budżetu.

Metodyki zwinne (agile)

Metodyki zwinne (agile)



Metodyki zwinne są alternatywnym rozwiązaniem dla tych, którzy nie odnajdują się w uporządkowanym, mocno formalnym modelu kaskadowym. Wymagają one jednak dłużej samoorganizacji zespołu i koncentracji wielu kompetencji.

Metodologia zwinna to struktura zarządzania projektami, która dzieli projekty na kilka dynamicznych faz, powszechnie znanych jako **sprinty**. Agile jest **metodologią iteracyjną**. Po każdym sprincie zespoły dokonują refleksji i patrzą wstecz, aby sprawdzić, czy jest coś, co można ulepszyć, aby móc dostosować strategię na następny sprint.



Manifest Agile

Manifest Agile to dokument skupiający się na czterech wartościach i 12 zasadach tworzenia oprogramowania Agile. Został opublikowany w lutym 2001 roku przez 17 twórców oprogramowania, którzy potrzebowali alternatywy dla bardziej liniowego procesu rozwoju produktu .



4 filary Agile

Jak wskazano w Manifeście Agile, istnieją cztery główne wartości zarządzania projektami Agile:

- Jednostki ponad procesy i narzędzia: Zwinne zespoły cenią współpracę zespołową i pracę zespołową ponad niezależną pracę i robienie rzeczy „według zasad”.
- Działające oprogramowanie ponad obszerną dokumentację: Oprogramowanie tworzone przez zespoły Agile powinno działać. Dodatkowa praca, np. dokumentacja, nie jest tak ważna, jak tworzenie dobrego oprogramowania.



4 filary Agile

- Współpraca z klientami w zakresie negocjacji umów: Klienci są niezwykle ważni w metodologii Agile. Zwinne zespoły pozwalają klientom wskazywać, dokąd oprogramowanie powinno trafić. Dlatego współpraca z klientem jest ważniejsza niż najdrobniejsze szczegóły negocjacji umowy.
- Reagowanie na zmiany zamiast podążania za planem: Jedną z głównych korzyści zwinnego zarządzania projektami jest to, że pozwala zespołom na elastyczność. Ta struktura pozwala zespołom szybko zmieniać strategie i przepływy pracy bez zakłócania całego projektu.



12 zasad Agile

Cztery wartości Agile są filarami metodologii Agile. Na podstawie tych wartości zespół opracował 12 zasad.

1. Najwyższy priorytet ma zadowolenie klienta dzięki wczesnemu i ciągłemu wdrażania wartościowego oprogramowania.
2. Należy być przygotowanym na zmianę wymagań nawet na późnym etapie rozwoju projektu. Proces zwinne wykorzystują zmiany dla zapewnienia klientowi konkurencyjności.
3. Należy dostarczać funkcjonujące oprogramowanie często, w kilkutygodniowych lub kilku miesięcznych odstępach. Im częściej, tym lepiej.



12 zasad Agile

4. Zespoły biznesowe i deweloperskie muszą ściśle ze sobą współpracować w codziennej pracy przez cały czas trwania projektu.
5. Należy tworzyć projekty wśród zmotywowanych ludzi. Zapewnienie im potrzebnego środowiska oraz wsparcia i zaufania skutkuje wykonaniem powierzonych zadań.
6. Najbardziej efektywnym i wydajnym sposobem przekazywania informacji zespołowi the deweloperskie mu i wewnątrz niego jest rozmowa twarzą w twarz.
7. Działająca oprogramowanie jest podstawową miarą postępu.



12 zasad Agile

8. Procesy zwinne umożliwiają zrównoważony rozwój. Sponsorzy, deweloperzy oraz użytkownicy powinni być w stanie utrzymywać równe tempo pracy.
9. Ciągła skupienie na technicznej doskonałości i dobrym projektowaniu zwiększa zwinność.
10. Prostota – sztuka minimalizowania ilości koniecznej pracy – jest kluczowa.
11. Najlepsze rozwiązania architektoniczne, wymagania i projekty pochodzą od samoorganizujących się zespołów.
12. W regularnych odstępach czasu zespół analizuje możliwości poprawy swojej wydajności, a następnie dostraja i dostosowuje swoje działania do wyciągniętych wniosków.



Zalety metodyk zwinnych

- + Elastyczne podejście do zakresu zmian i ich akceptacji
- + Koncentracja na dostarczonej wartości zamiast na zakresie prac
- + Stosowanie krótkich iteracji oraz weryfikacji po każdej z nich
- + Wzmacnianie samodzielności i odpowiedzialności zespołu

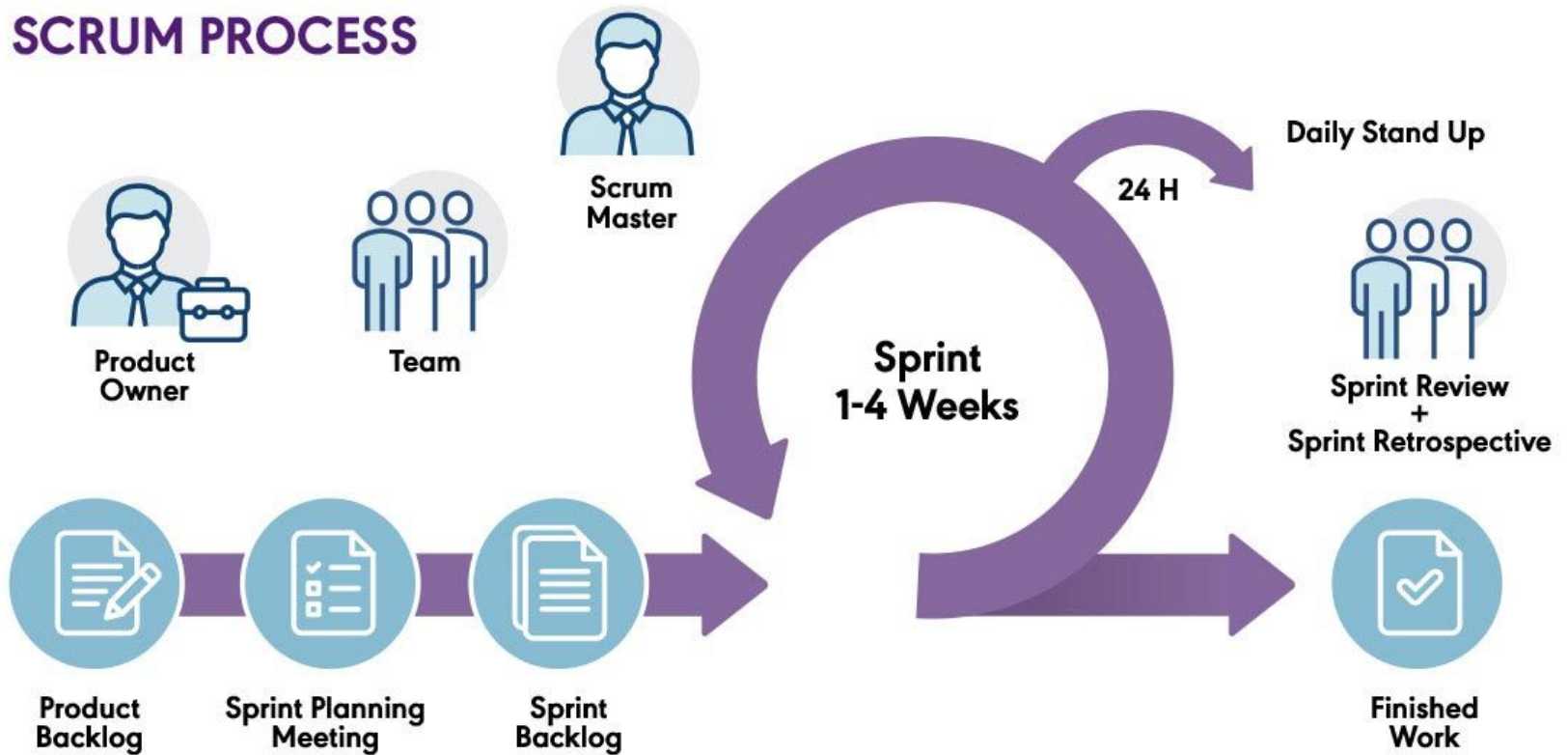
Scrum



Scrum

Scrum to metodyka o charakterze **iteracyjnym** oraz **przyrostowym**, zaliczana do zwinnych metod pracy zgodnych z manifestem Agile. Jest to taki sposób prowadzenia projektów, którego celem jest osiągnięcie zamierzonego rezultatu i panowanie nad zmianami. Idealnie wpasowuje się w młode, małe organizacje, które rozpoczynają swoją działalność.

SCRUM PROCESS





Scrum

Codzienny (ang. daily) Scrum – jak sama nazwa wskazuje, jest to codzienna spotkanie, trwającą maksymalnie do 15 min, na którym uczestnicy zespołu mówią, co udało im się zrobić wczoraj, co chcą zrobić dzisiaj i jakie widzą przeszkody.

Sprint – Scrum dzieli projekt na iteracje, czyli wydania cykliczne o stałej długości. W praktyce Iteracja trwa od dwóch do czterech tygodni. Długość iteracji nie ulega zmianie. Jeżeli przyjęto, że Sprint trwa cztery tygodnie, to ten zakres czasu jest ostatecznie. Jeżeli zadanie nie może być ukończona w czasie sprintu, odpowiednie cechy produktu są dzielone i zadanie wraca do backlogu produktu. Technika ta nazywa się ograniczeniem czasowym (ang. timeboxing).



Scrum

Przyrost produktu – wynikiem każdego sprintu jest produkt, który potencjalnie można wdrożyć.

Rejestr (ang. backlog) produktu – właściciel produktu zarządza spriorytetyzowaną listą planowanych pozycji do zrealizowania, czyli backlogiem produktu. Zadania produktu zmieniają się od sprintu do sprintu.

Rejestr (ang. backlog) sprintu – na początku każdego sprintu zespół scrumowy wybiera z backlogu produktu zbiór pozycji o najwyższym priorytecie. Ponieważ to zespół scrumowy, a nie właściciel produktu wybiera pozycje do realizacji podczas sprintu, wybór nawiązuje raczej do zasady pobierania (według której członek zespołu przydziela sobie zadania sam), niż do zasady wpychania pracy, czyli zadania przydzielane są odgórnie i realizowane według nadanego priorytetu.



Scrum

Definicja ukończenia (ang. Definition of Done) – by upewnić się, że na końcu sprintu uzyskany zostanie produkt nadający się do wdrożenia, zespół scrumowy przedyskutowuje i określa odpowiednie kryteria ukończenia sprintu. Dyskusja pogłębia zrozumienie przez zespół poszczególnych pozycji w backlogu i wymagań dotyczących produktu.

Raportowanie – status sprintu jest raportowany i uaktualniany codziennie w czasie spotkań nazwanych codziennym scrumem, dzięki czemu zawartość i postęp bieżącego sprintu – a także wyniki testów – są dostępne dla zespołu scrumowego, osób zarządzających i innych zainteresowanych.



Role w metodyce Scrum

Zgodnie z metodyką Scrum i mniej ról, tym lepiej. Łatwiej wtedy od utrzymanie wysokiego poziomu komunikacji, łatwiej o szybkie podejmowanie decyzji i reagowanie na zmiany.

Scrum wyróżnia trzy role, które wchodzi w skład zespołu scrumowego:

- Scrum Master
- Development Team
- Product Owner



Role w metodyce Scrum

Scrum Master – dba o odpowiednie stosowanie i implementację reguł scrumowych. Jeśli wystąpią naruszenia reguł, to master sprawdza, co jest przyczyną takich działań, i stara się je usuwać z procesu. Eliminuje wszelkiego rodzaju przeszkody, zarówno te związane z zasobami, jak i tak, które umożliwiają zespołowi stosowanie właściwych reguł i praktyk. Scrum Master nie jest liderem projektu, a trenerem. Daje tylko wskazówki i doradza zespołowi, aby wybrać właściwą drogę, technologię czy zastosować właściwą metodę. Nie dzieje guje pracy, nie planuje projektów, a raczej dba o to, aby zespół spełnił wszystkie kryteria, by osiągnąć określone przez siebie cele.



Role w metodyce Scrum

Zespół wytwórczy/deweloperski (ang. Development Team) – zespół taki może liczyć od trzech do dziewięciu osób, które razem wytwarzają i testują produkt. Nie ma lidera zespołu, więc wszystkie decyzje podejmowane są wspólnie. Scrum nie dostarcza wytycznych, jak powinno być wykonywane testowanie w tego typu projekcie.

Właściciel produktu (ang. Product Owner) – właściciel produktu reprezentuje użytkownika (przygotowuje historyjki użytkowników) oraz tworzy priorytety w backlogu produktu, zarządza nimi i określa ich wartość. Kontroluje też finansową stronę projektu (ROI). Właściciel produktu nie jest liderem zespołu.



ROI

Zwrot kosztów inwestycji (ROI) – podstawowy wskaźnik rentowności, wykorzystywane do mierzenia efektywności danego działania. Wzór na obliczenie:

$$\text{ROI \%} = (\text{przychód} - \text{koszt inwestycji}) / \text{koszt inwestycji} * 100$$

Wskaźnik ten daje odpowiedź na pytanie, czy dany projekt przyniesie firmie wymierną korzyść w określonym czasie. Im wyższy wskaźnik, tym lepiej.



Historyjki użytkownika (ang. User Stories)

Słabej jakości specyfikacja – główny powód niepowodzenia projektu – może powstawać wyniku braku zrozumienia przez klienta własnych potrzeb, braku globalnej wizji systemu, nadmiernych lub sprzecznych właściwości oraz innych błędów prze powin formacji.

W środowisku zwinnym historyjki użytkownika pisane są po to, by uchwycić wymagania z perspektywy programistów testerów i przedstawicieli biznesu. W przeciwieństwie do modelu sekwencyjnego, gdzie wspólna wizja właściwości jest osiągnięta przez przeglądy formalne po utworzeniu wymagań, w zwinnym rozwoju oprogramowania wizja ta jest osiągnięta przez częste przeglądy nieformalne podczas pisania wymagań.



Historyjki użytkownika (ang. User Stories)

Dobrze stworzona historyjka użytkownika spełnia warunki modelu **INVEST**, to oznacza że jest:

- Independent – niezależna
- Negotiable – Negocjowana
- Valuable – Wartościowa
- Estimable – Dająca się oszacować, czyli można określić niezbędny czas i zasoby
- Size Appropriately – Odpowiedniej wielkości
- Testable – testowalna

Historyjki użytkownika mają na celu skupienie się na realnym użytkowników systemu i na zaspokojeniu jego potrzeb. Zmuszają autora do zastanowienia się nad tym, jakie rozwiązanie będzie najlepsze dla użytkownika.



Historyjki użytkownika (ang. User Stories)

Każda historyjka powinna zawierać kryteria akceptacji dla właściwości wymienionych w schemacie INVEST. Te kryteria powinny być definiowane we współpracy pomiędzy przedstawicielami biznesu, programistami i testerami. Dają one programistom i testerom lepszy wygląd wizję właściwości, którą reprezentanci biznesu będą sprawdzać. Zespół zwinny uważa zadanie za ukończone, gdy zbiór kryteriów akceptacyjnych jest spełniony.



Historyjki użytkownika (ang. User Stories)

Koncepcja 3C mówi, z jakich części powinna składać się historyjka. Są to:

- Karta (ang. Card) lub fiszka, na której zapisano historyjkę. Stanowią one fizyczny odnośnik do wymagania
- Dyskusja (ang. Conversation) na temat szczegółów
- Potwierdzenie (ang. Confirmation) dzięki testom akceptacyjnym, które potwierdzają implementację z punktu widzenia biznesu



Zalety historyjek użytkownika

- + Mniej dokumentacji – więcej spotkań i rozmów
- + Najważniejsze jest cel użytkownika, a nie sam system
- + Nie narzucają ustalania szczegółów z góry

Historyjki użytkownika muszą obejmować zarówno właściwości funkcjonalny, jak i нефункционалне.

Przykład Właściciel produktu zdefiniował wymagania dotyczące aplikacji sklepu internetowego. Poniżej przykład jednej z historii użytkownika. Użytkownik z rolą sprzedawca chce, żeby po wejściu w zakładkę zamówienia stan zamówień ostatnio się nie opłacone wyświetla się na czerwonym tle. Użytkownik niezarejestrowany chce mieć możliwość złożenia zamówienia w sklepie internetowym.



Zalety metodyki Scrum

- + Komunikacja ponad dokumentację
- + Przejrzystość działań, czyli krótkie sprinty z ustalonym zakresem prac
- + Szybkie reagowanie na zmiany wymagań

Kanban



Kanban

Kanban jest metodyka wywodząca się od Toyoty, czyli metodą zarządzania produkcją stosowaną przez wiele firm w Japonii oraz na całym świecie. Nazwa Kanban w języku japońskim oznacza tablicę informacyjną. Jej celem jest wizualizacja i optymalizacja przepływu pracy w łańcuchu wartości dodanej. Kanban składa się z pięciu zasad:

- Zobrazuj przepływ pracy
- Ogranicz WIP (ang. Work In Progress)
- Zarządzanie przepływem
- Ustal wyraźne zasady
- Wspólnie ulepszaj

Prezentowane podejście nie przewiduje iteracji czyli powtarzania, a jedynie ciągły proces.



Kanban

Kanban wykorzystuje 3 narzędzia, którymi są:

1. Tablica kanbanowa

Łańcuch wartości, którym zarządzamy, jest wizualizowany na tablicy kanbanowej. Każda kolumna pokazuje etap procesu – zbiór powiązanych czynności, które powinny zostać zrealizowane, lub zadania, które będą przetwarzane, są symbolizowane przez kolorowe kartki. Kartki są przesuwane od lewej do prawej strony przez kolejne kolumny tablicy, odpowiadające następującym po sobie etapom procesu.



Kanban

2. Limit pracę w toku (ang. **Work in Progress**)

Liczba równolegle wykonywanych zadań jest ściśle ograniczona. Jest to kontrolowane przez ustalenie maksymalnej dozwolonej liczby kartek na danym etapie lub globalnie na tablicę. Gdy na danym etapie na tablicy pojawia się wolne miejsce, pracownik przesuwa kartkę z wcześniejszego etapu.



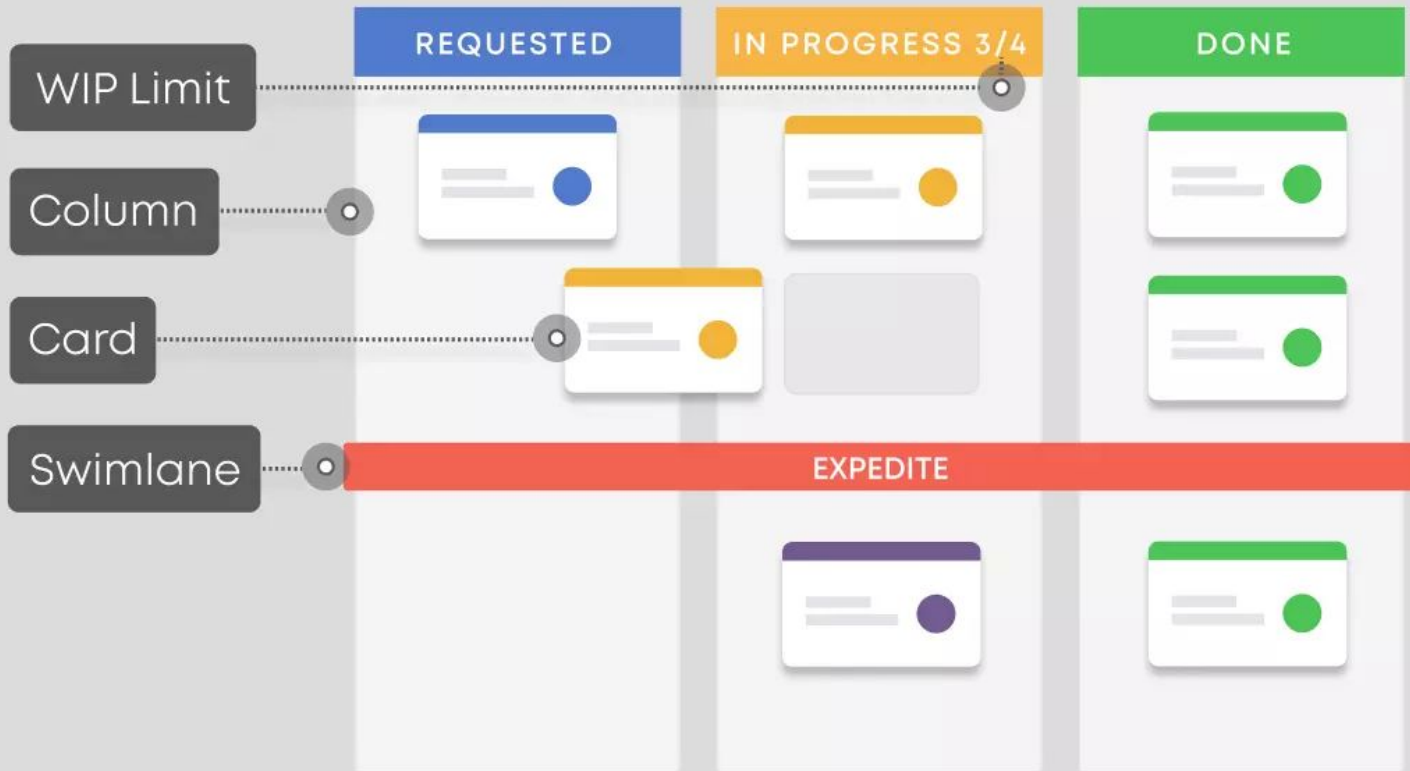
Kanban

3. Czas realizacji (ang. Lead Time)

Metodyki Kanban używa się do optymalizacji ciągłego przepływu zadań przez minimalizację średniego czasu realizacji dla całego łańcucha wartości. Kanban wykazuje pewne podobieństwo do scruma. W obu podejściach wizualizacja aktywnych zadań np. na dostępnej dla wszystkich tablicy, zapewnia przejrzystość zawartości i postępu zadań. Zadania jeszcze nieprzydzielone czekają w zbiorze zadań do wykonania. W momencie, gdy zwalnia się miejsce na kolejnym etapie, są one odpowiednio przesuwane na tablicę kanbanowej.

Zarówno iteracje, jak i sprinty są opcjonalne w kanbanie.

KANBAN BOARD





Zalety metodyki Kanban

Proces w kanbanie pozwala na wydawanie produktów raczej jeden po drugim niż jako część wydania. Ograniczenia czasowe, jako mechanizm synchronizujący, stają się tym samym opcjonalne, inaczej niż w scrumie, gdzie synchronizuje się wszystkie zadania w ramach sprintu. Zespoły stosujące metodykę Kanban mierzą czas realizacji, czyli średni czas od momentu zgłoszenia żądania pracy do momentu jej zakończenia, i optymalizują swoje procesy, aby skrócić czas wykonywania zadań.

Zalety metodyki Kanban:

- + Wizualizacja przebiegu prac
- + Łatwa weryfikacja optymalizacji procesów

Scrumban



Scrumban

Scrumban jest hybrydą uzyskaną z połączenia struktury Scrum z metodami Kanban. Jest rekomendowane metodyka dla zespołów, które chcą przejść od formy pracy właściwej dla Scruma do Kanbana. Obecnie jest to najpopularniejsza metodyka zwinna.

Elementy Scruma, które można tu odnaleźć:

- Planowanie – iteracje w regularnych odstępach czasu
- Przejrzystość – proces musi być zrozumiały dla wszystkich zaangażowanych
- Inspekcja – scrumowe artefakty muszą być często przeglądane w celu wykrycia niepożądanych rozbieżności
- Adaptacja – w momencie zauważenia rozbieżności proces musi zostać natychmiast skorygowany



Scrumban

Elementy kanbana, które można tu odnaleźć:

- Ciągły system i przepływ pracy
- Nowe wymagania mogą być dodane zawsze, o ile kolejce zadanie istnieje wolne miejsce
- Wyraźne limity liczby elementów w toku w dowolnym momencie
- Poszczególne role nie są jasno określone
- Krótkie terminy realizacji – nacisk na analizę i planowanie
- Korzystanie z buforów procesów i diagramów przepływu, aby ujawnić słabości procesu i zidentyfikować możliwości poprawy

Elementy zaczerpnięte z kanbana wpływają na ulepszenie procesu, dodają do metody Scrumban wizualizację i więcej wskaźników wartości.