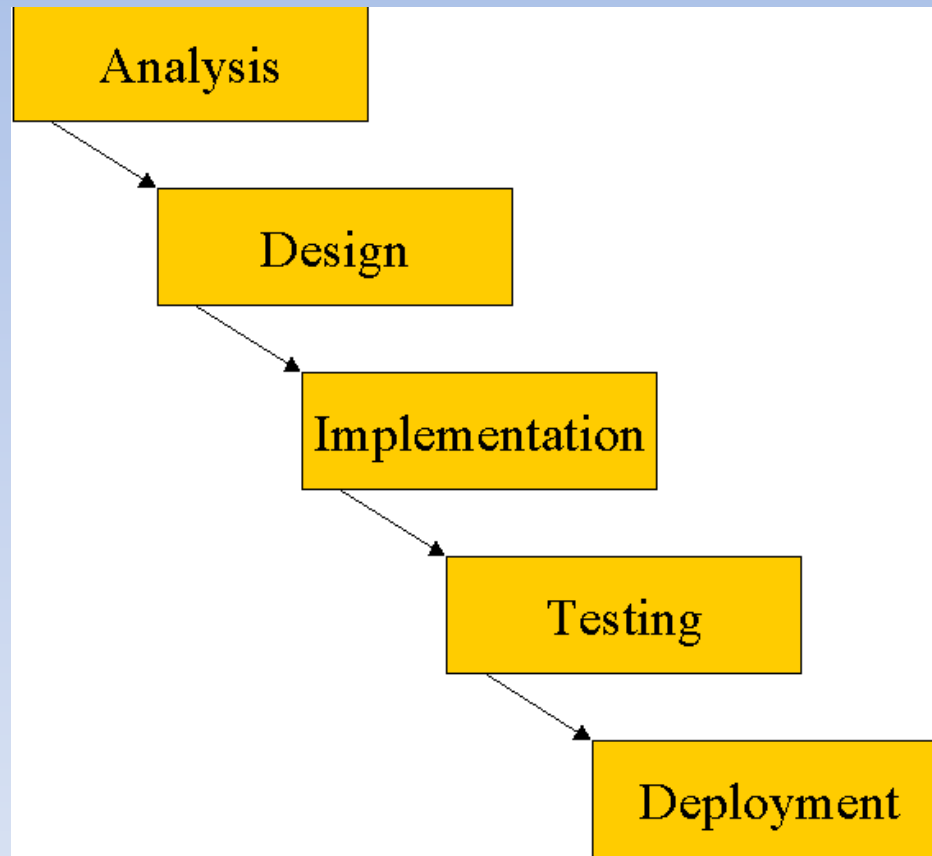


Metodologie wytwarzania oprogramowania

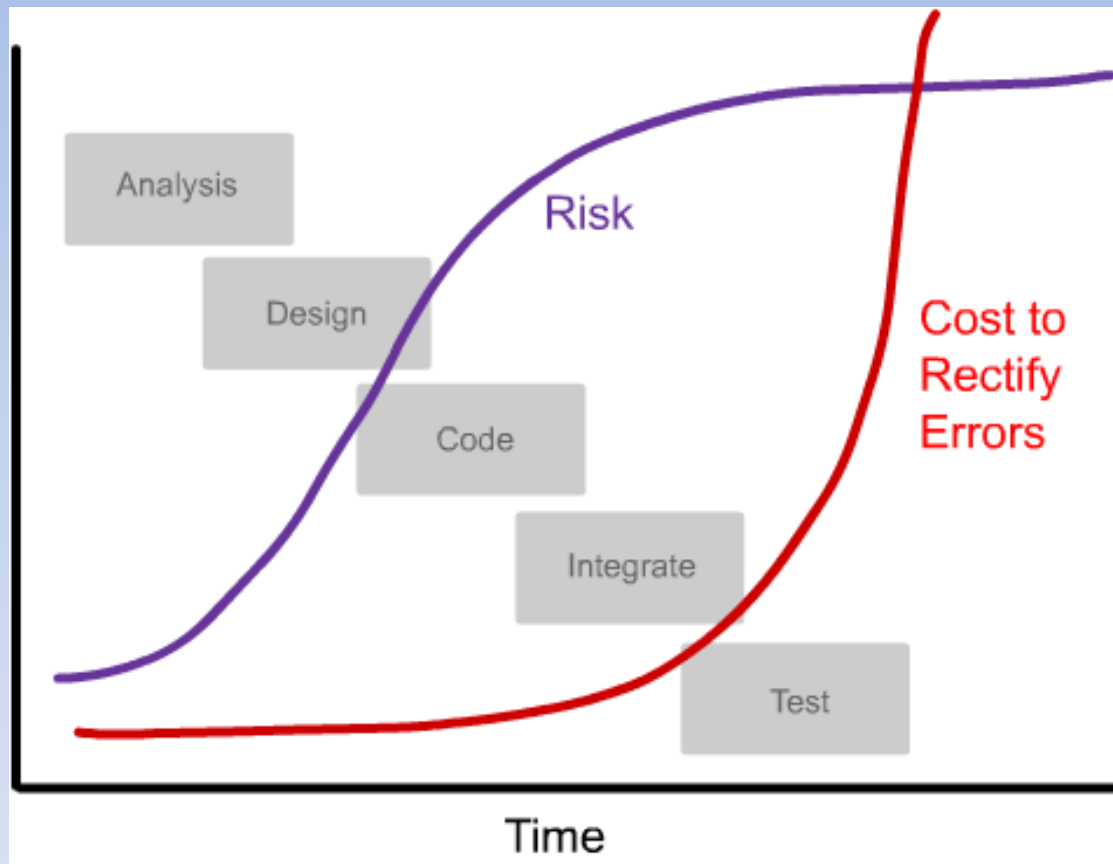
Metodologie

- Model kaskadowy
- RUP
- Scrum
- Kanban

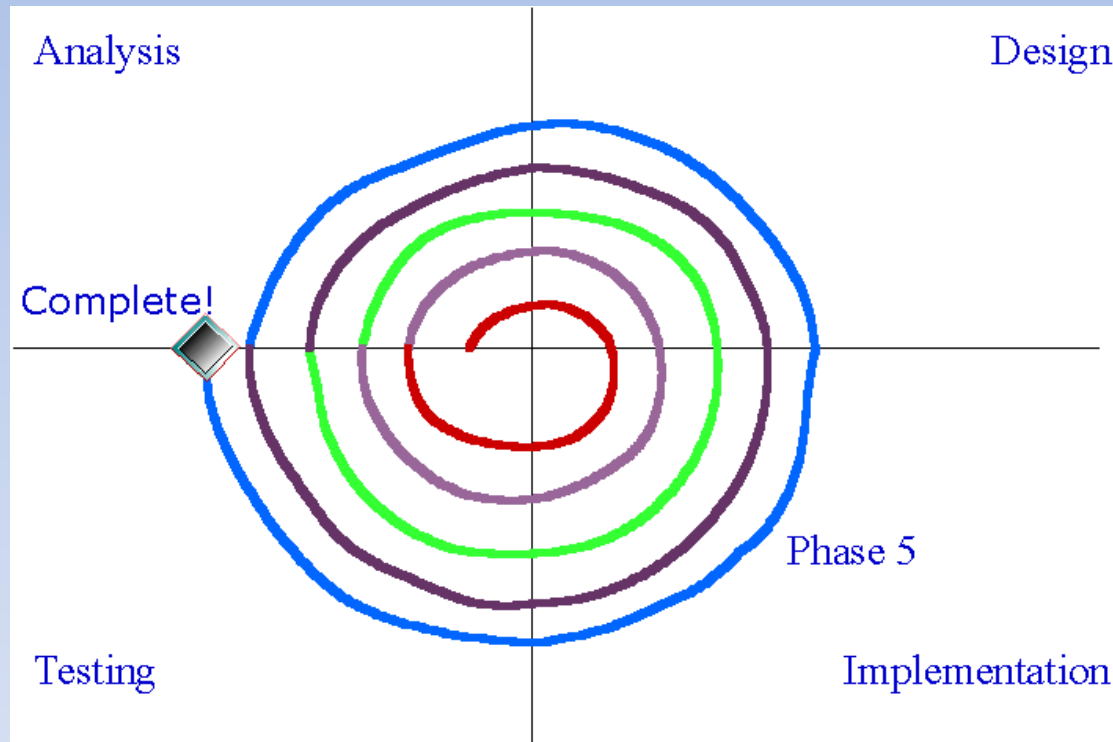
Model kaskadowy (Waterfall Model)



Model kaskadowy



Model spiralny (Spiral Model)

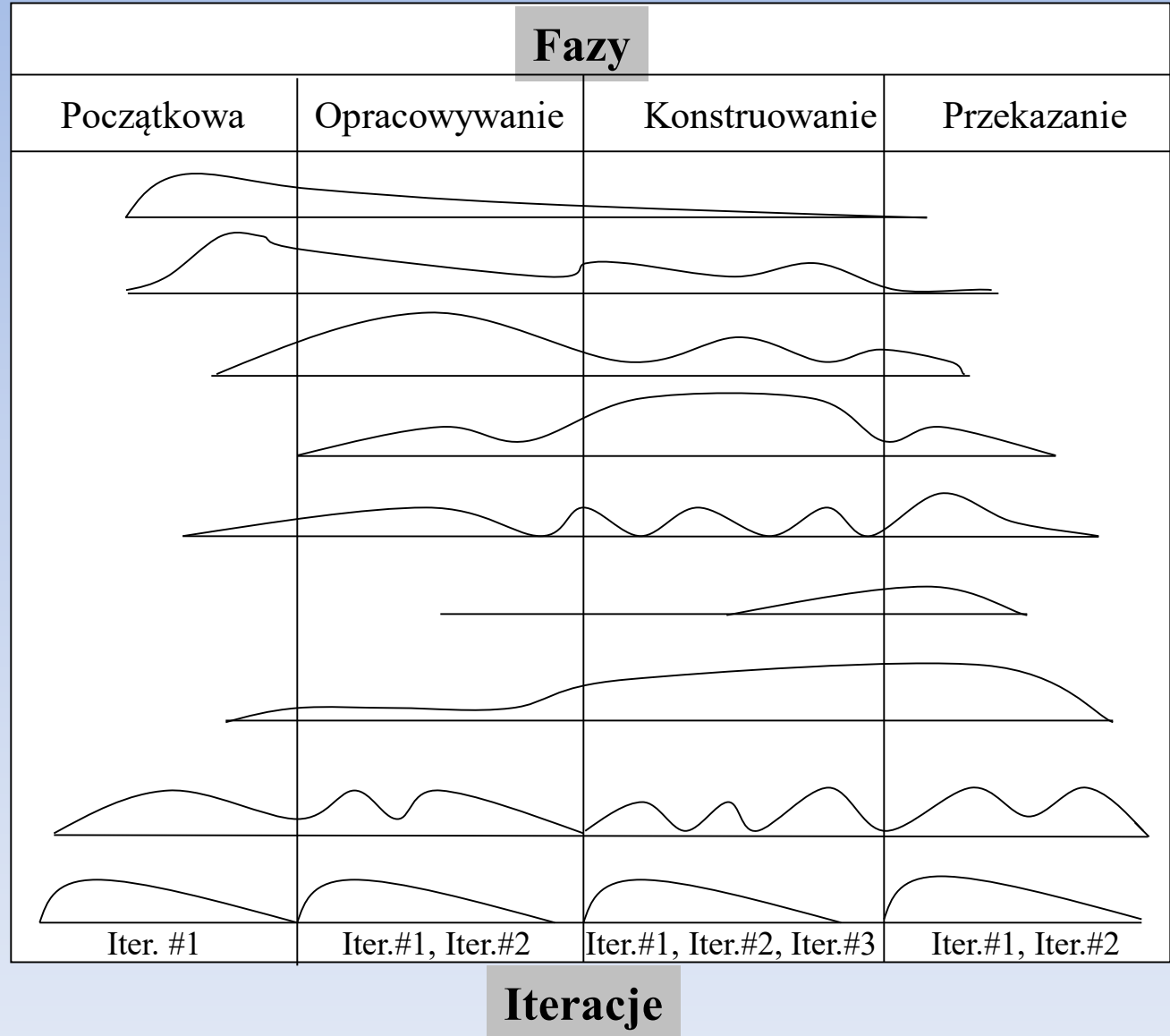


RUP wprowadzenie

- RUP (Rational Unified Process) jest :
 - Iteracyjną i przyrostową metodyka
 - W pełni konfigurowalną platformą do obsługi procesu tworzenia oprogramowania, z której można wybierać i wdrażać tylko te składniki procesu, które są potrzebne na danym etapie projektu.

RUP

Przebieg prac



RUP jako proces iteracyjny

- Tworzony system jest rozbity na miniprojekty
- Pozwala na : „trochę planowania”, „trochę projektowania”, „trochę implementacji”
- Nastawienie na komponenty a nie na całość
- Umożliwia ewaluację systemu, nie na wytworzenie całego systemu od razu
- Umożliwia poprawę błędów na wcześniejszym etapie i wyciąganie wniosków na podstawie poprzednich iteracji

Struktura RUP

- Struktura RUP składa się z czterech elementów składowych:
 - Pracownicy: kto?
 - Aktywności: jak?
 - Artefakty: co?
 - Przepływy prac: kiedy?

Struktura RUP

- Struktura RUP składa się z czterech elementów składowych:
 - **Pracownicy: kto?**
 - Aktywności: jak?
 - Artefakty: co?
 - Przepływy prac: kiedy?

Pracownicy

- **Pracownik** posiada pewną **role**. W RUP mówi się raczej o roli niż o pracowniku
- Każdy pracownik jest skojarzony z pewnym zbiorem wzajemnie powiązanych **aktywności**, które ma do wykonania i za które jest odpowiedzialny
- **Odpowiedzialność** pracownika jest zwykle wyrażana w terminach artefaktów, które tworzy, modyfikuje czy nadzoruje.

Aktywności

- **Aktywność** specyfikuje **jednostkową** pracę, którą pracownik ma do wykonania w projekcie
- Rozmiar aktywności jest określany za pomocą czasu potrzebnego na jej wykonanie: zazwyczaj kilka godzin do kilku dni.
- Aktywność z reguły związana jest z jednym pracownikiem i pracą nad jednym lub niewielką liczbą artefaktów.
- Aktywność związana z danym artefaktem może być wielokrotnie powtarzana, szczególnie gdy przechodzi się do kolejnych iteracji w procesie rozszerzania i udoskonalania produktu.

Przykłady aktywności

- *Planuj iterację*
 - Wykonywane przez pracownika *Kierownik Projektu*
- *Znajdź przypadki użycia*
 - Wykonywane przez pracownika *Analityk*
- *Zrób recenzję projektu*
 - Wykonywane przez pracownika *Recenzent projektu*
- *Przeprowadź test wydajnościowy*
 - Wykonywane przez pracownika *Tester wydajności.*

Artefakty

- Artefakt
 - Termin wprowadzony przez Rational Unified Process (inne nazwy: “produkt pracy”, “jednostka pracy”) oznaczający produkt wytwarzany, modyfikowany, nadzorowany, bądź używany w trakcie którejś z aktywności realizowanych w procesie wytwarzania produktu finalnego.
- Produkt finalny obejmuje pewien podzbiór artefaktów - tych, które zostaną dostarczone do rąk klienta.

Przykłady artefaktów

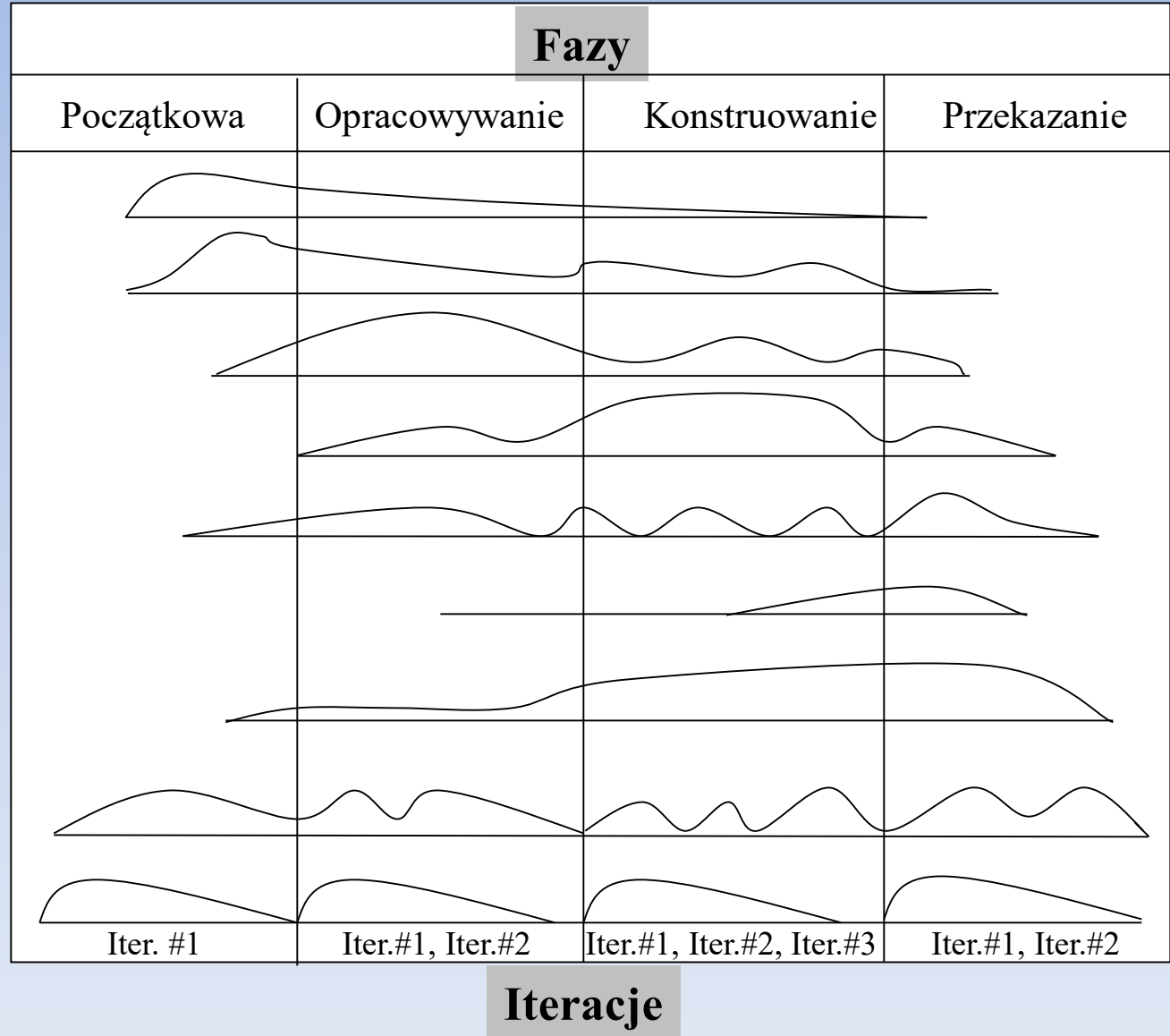
- Projekt implementacji (model logiczny)
- Plan projektu
- Dokument wizji projektu
- Dokument wymagań użytkownika
- Kod źródłowy
- Element modelu np.. Przypadek użycia lub klasa

Przeptywy prac

- Przeptywy prac
 - Sekwencja aktywności, której efektem jest wytworzenie “obserwowalnej (znaczącej) wartości”

RUP

Przebieg prac



Przepływy prac (workflows)

- Przepływy podstawowe związane z czynnościami inżynierskimi:
 - Modelowanie biznesowe
 - Wymagania
 - Analiza i projektowanie
 - Implementacja
 - Testowanie
 - Wdrażanie
- Przepływy wspierające
 - Konfiguracja i zarządzanie zmianami
 - Zarządzanie projektem
 - Określenie środowiska

Cykle

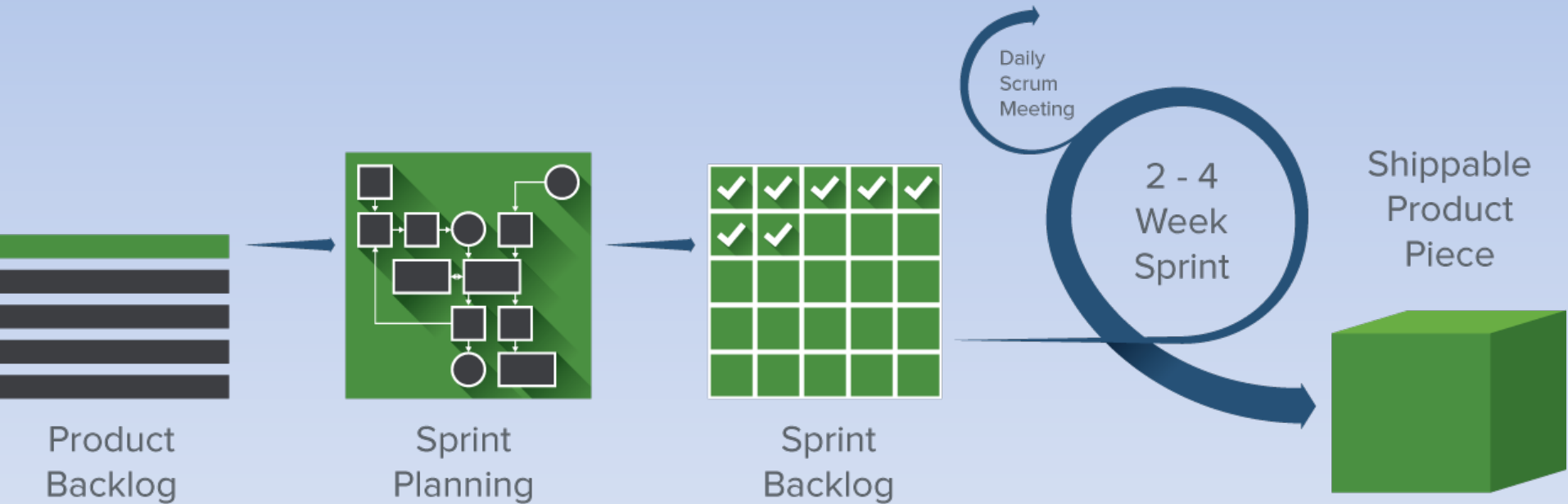
- Wytwarzanie oprogramowania następuje w cyklach:
 - Cykl początkowy
 - Cykle ewolucyjne
- Fazy wytwarzania oprogramowania:
 - Rozpoczęcie (Inception)
 - Opracowanie (Elaboration)
 - Konstruowanie (Construction)
 - Przekazanie (Transition)

Scrum

Cechy

- Iteracyjny
- Inkrementacyjny
- Posiada swoją terminologię i role

Poszczególne kroki w Scrum



Poszczególne kroki w Scrum

- Product backlog: zawiera listę funkcjonalności do zrealizowania
- Sprint planning: z backlog wybierane są takie elementy, które będą zrobione w najbliższym Sprincie

Role

- Właściciel produktu
- Szef scruma
- Członek zespołu

Właściciel produktu

- Jest odpowiedzialny za decyzje w sprawie cech funkcjonalnych produktu oraz zarządzanie priorytetami w ich implementacji.
- Reprezentuje on interesy wszystkich ludzi zaangażowanych w projekt i finalny produkt – klientów, użytkowników.

Szef scrum

- Jest odpowiedzialny za egzekwowanie praktyk i zasad Scrum
- Dbanie o zespół

Członek zespołu

- Każdy, kto należy do zespołu wykonującego czynności związane bezpośrednio z wytwarzaniem oprogramowania
- Członkowie zespołu pełnią różnorodne funkcje w zależności od projektu

Czynności procesowe i narzędzia

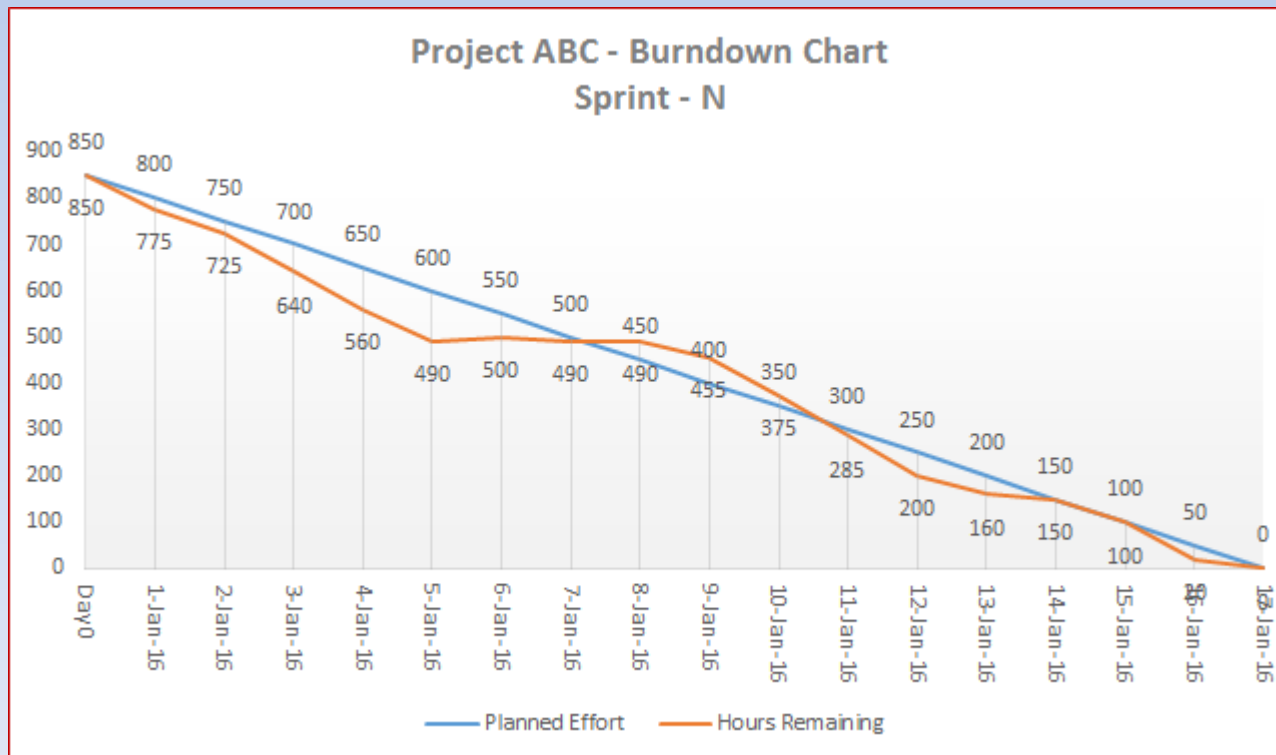
- **Sprint – iteracja:** czas trwania 13 dni. Sugerowany jest podobny czas trwania sprintu.
- Na początku sprintu całodniowe spotkanie składające się z dwóch części:
 - **Pierwsza 4h: planowanie - wykaz prac sprintu.** Następnie na tej podstawie określany jest cel sprintu, czyli wartość biznesowa.
 - **Druga część: podział wykazu prac na zadania.** Lista zadań niekoniecznie kompletna, zadania mogą się pojawić potem.
- Codziennie odbywa się spotkanie (**stand-up meeting**)

Stand-up meeting

- Spotkanie codzienne
- Czas trwania co najwyżej 15min
- Odpowiedź na pytania:
 - Co robiłeś wczoraj?
 - Co będziesz robić dzisiaj?
 - Co ci stoi na przeszkodzie?

Burn-down chart

- wykres przedstawiający ilość pracy do wykonania w bieżącym sprincie



Scrum board

Story	To Do	In Process	To Verify	Done
<p>As a user, I... 8 points</p>	<p>Code the... 9</p> <p>Code the... 2</p> <p>Test the... 8</p> <p>Test the... 4</p>	<p>Code the... DC 4</p> <p>Test the... SC 8</p>	<p>Test the... SC 6</p>	<p>Code the... D</p> <p>Test the... SC 8</p> <p>Test the... SC</p> <p>Test the... SC</p> <p>Test the... SC 6</p>
<p>As a user, I... 5 points</p>	<p>Code the... 8</p> <p>Code the... 4</p> <p>Test the... 8</p> <p>Code the... 6</p>	<p>Code the... DC 8</p>		<p>Test the... SC</p> <p>Test the... SC</p> <p>Test the... SC 6</p>

Zakończenie sprintu

- **sprint review meeting:** zespół przedstawia właścicielowi produktu, co osiągnięto w ostatniej iteracji (4h)
- **sprint retrospective meeting:** spotkanie retrospektywne sprintu trwające trzy godziny, na którym zespół i szef scruma rozmawiają o tym, które zadania i czynności zostały wykonane należycie podczas ostatniego sprintu i jak można usprawnić następny
- **Sprint można zakończyć wcześniej:** jeśli są uciążliwe problemy z realizacją bądź cel sprintu okazał się nieistotny. Może to zrobić szef scruma bądź właściciel produktu. Po tym organizuje się nowy sprint.

Zalety scruma

- **Większa transparentność projektu:** dzięki spotkaniom codziennym wiadomo co się robi w projekcie i niwelowane są ewentualne nieporozumienia;
- **Wzmocnienie odpowiedzialności zespołu:** nie ma project managera, zespół decyduje nad czym pracować w danym sprincie;
- **Łatwiejsze zmiany:** dzięki krótkim iteracjom i feedbacku od klienta;
- **Redukcja kosztów:** dzięki komunikacji problemy są znane wcześniej co wpływa na obniżenie kosztów i poprawę jakości.

Wady scruma

- **Ryzyko zwiększenia zakresu:** brak wyspecyfikowanej daty końca projektu może nakłonić klientów do żądania zwiększenia funkcjonalności;
- **Wymaga doświadczonego i zaangażowanego zespołu:** wymagana znajomość zasad scrumu oraz duża wiedza techniczna członków (brak ról);
- **Zły scrum master może zruinować projekt;**
- **Błędnie zdefiniowane zadania mogą prowadzić do nieścistości:** jeśli początkowe założenia są nieściste, planowanie staje się trudne i sprinty mogą się przeciągnąć w czasie.

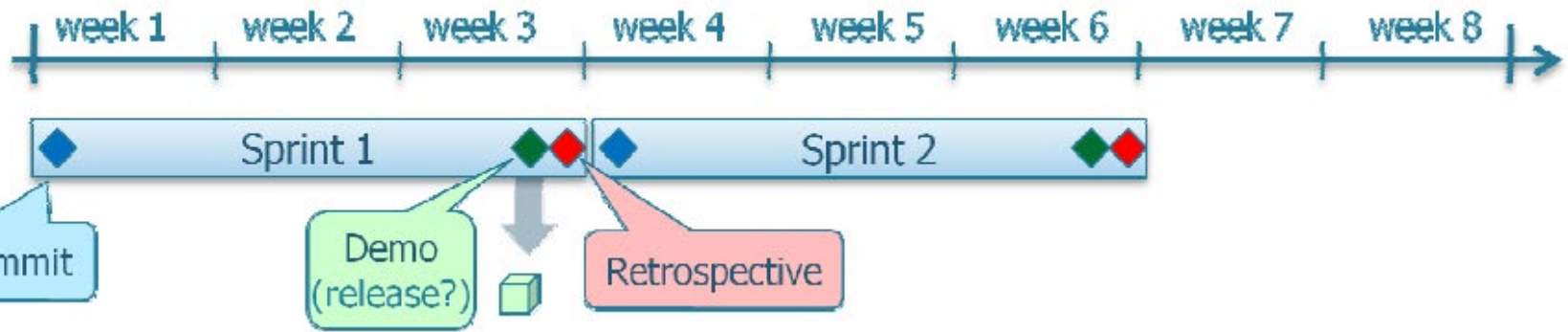
Kanban w skrócie

- Wizualizacja workflow:
 - Podział pracy na kawałki i wystawienie na tablicę
 - Używanie kolumn na tablicy do zilustrowania gdzie jest dany element w workflow
- Limit WIP(work in progress) – przypisanie limitu elementów w poszczególnych kolumnach workflow
- Mierzenie czasu wykonania poszczególnych elementów –należy umieć przewidzieć czas trwania poszczególnych elementów

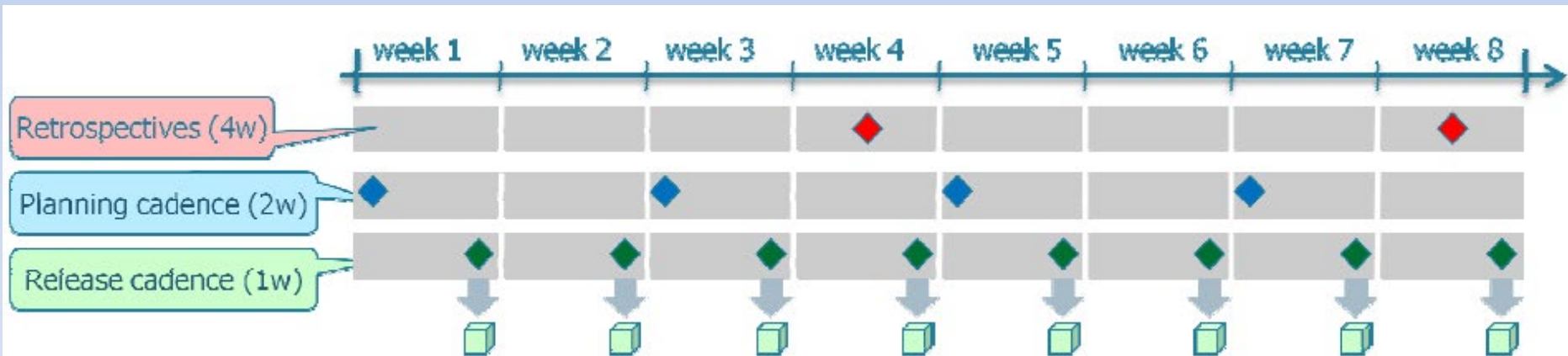
Scrum vs Kanban - różnice

- W Scrum są role, w Kanbanie nie ma
- W Scrum są dobrze zdefiniowane fazy iteracji: planowanie, implementacja i release. Kanban nie nakazuje iteracji. Można robić kiedy się chce. Np. release kiedy jest coś do pokazania.
- W Scrum mamy limit WIP na iterację, w Kanban limit na stan w workflow.
 - Dodanie nowego elementu do Scruma spowoduje dodanie do Product Backlog
 - Dodanie nowego elementu do Kanbana (kolumna TO DO) spowoduje usunięcie innego elementu.

Jedna kadencja vs trzy kadencje

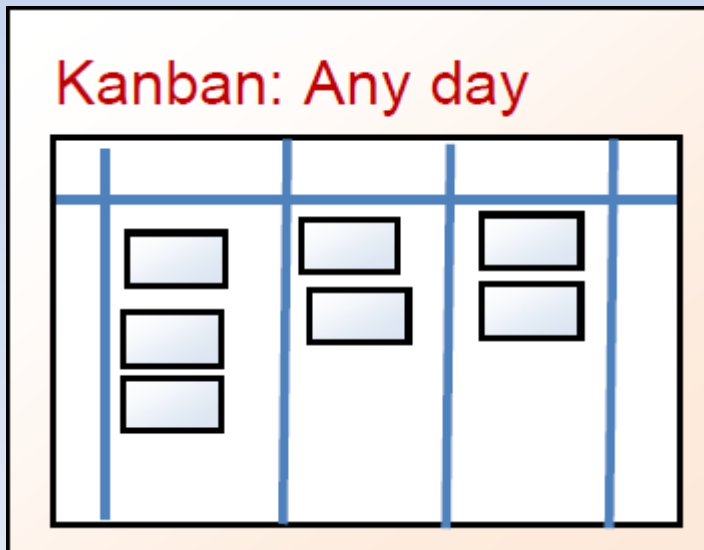
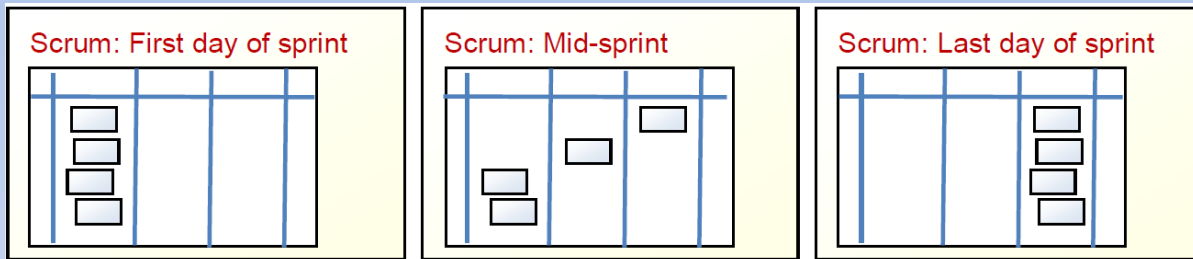


Scrum



Kanban

Tablica Scruma i Kanbana



Zalety Kanbana

- **Podnosi elastyczność pracy:** nie ma faz, priorytety są na nowo określane gdy przychodzą nowe zadania
- **Ogranicza marnotrawienie czasu:** zapewnia że zespół nie zajmuje się rzeczami niepotrzebnymi w danej chwili
- **Łatwy do zrozumienia** (poprzez wizualizacje)
- **Ułatwia łatwe udostępnianie produktu**

Wady Kanbana

- **Niezaktualizowana tablica:** w przeciwnym razie nie będzie aktualnych zadań do wykonania
- **Zbyt skomplikowana tablica:** może być nieczytelna
- **Brak synchronizacji:** nie ma wyszczególnionych ram czasowych wykonywania zadań

Manifest Agile

- Osiągnięcie satysfakcji klienta poprzez szybkość wytwarzania oprogramowania,
- Działające oprogramowanie jest dostarczane okresowo (raczej tygodniowo niż miesięcznie),
- Podstawową miarą postępu jest działające oprogramowanie,
- Późne zmiany w specyfikacji nie mają destrukcyjnego wpływu na proces wytwarzania oprogramowania,
- Bliska, codzienna współpraca pomiędzy biznesem a deweloperem,
- Bezpośredni kontakt jako najlepsza forma komunikacji w zespole i poza nim,
- Ciągła uwaga nastawiona na aspekty techniczne oraz dobry projekt (design),
- Prostota,
- Samozarządzalność zespołów,
- Regularna adaptacja do zmieniających się wymagań.