



dr inż. Michał Malinowski
michal.malinowski@uth.edu.pl



[https://www.linkedin.com/in/
michał-malinowski-malkon/](https://www.linkedin.com/in/michał-malinowski-malkon/)

Metodyki Tradycyjne,
Zwinne, DevOps
oraz zarządzania projektami

Wykład 03

INŻYNIERIA
OPROGRAMOWANIA



```
gmb{position:absolute;z-index:999;cop  
adows:0 1px 5px #ccc}.gbrtl .gbm{-moz-bo  
olor:#ccc;display:block;position:absolu  
lue=5);*opacity:1;*top:-2px;*left:-5px;  
mpty:1\0;/top:-4px\0;/left:-6px\0;/rig  
-moz-inline-box;display:inline-block;fo  
e, gbmoo{display:block;list-style:none;  
play:inline-block;line-height:27px;padd  
q{cursor:pointer;display:block;text-de  
sion:relative;z-index:1000}.gbts{*disp  
ad).gbts{padding-right:9px}#gbz .gbzt,  
h10).gbts  
background:url(//
```

Zagadnienia (2h)

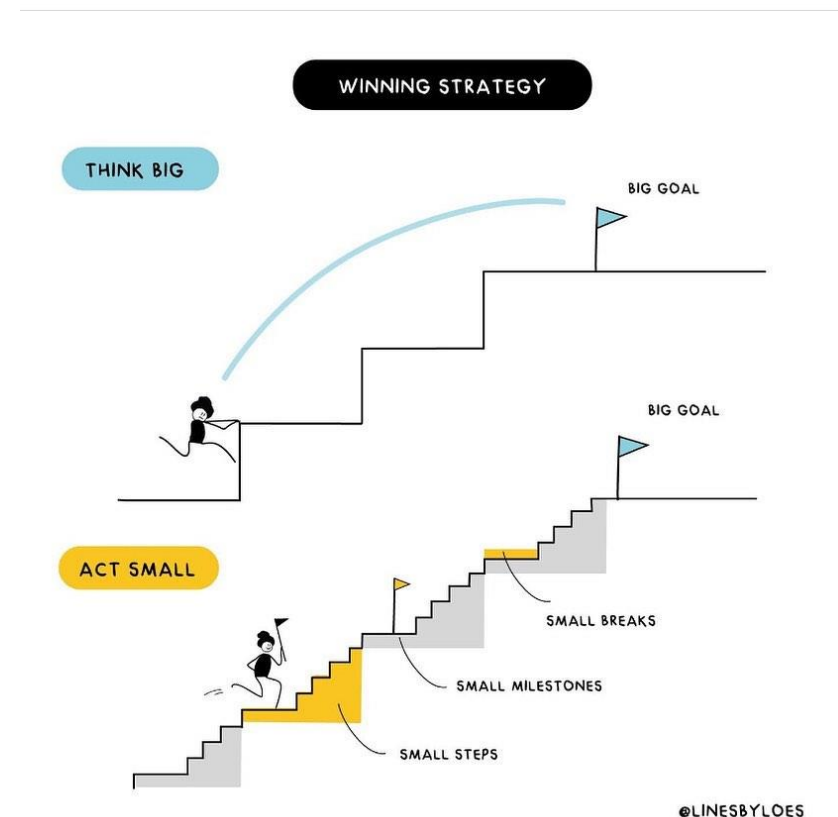
- [Klasyczny Model Wodospadowy](#)
- [Metodyka Zwinna](#)
- [DevOps](#)
- [Rzeczywistość](#)
- [PRINCE2](#)
- [ITIL](#)
- [PDCA](#)
- [Wybór odpowiedniej metodyki dla różnych organizacji i projektów](#)

Klasyczny Model Wodospadowy

Myśląc o czymś bardzo skomplikowanym, nie próbuj robić wszystkiego jednocześnie.

Podziel to na mniej złożone części i skoncentruj się kolejno na każdej z nich.

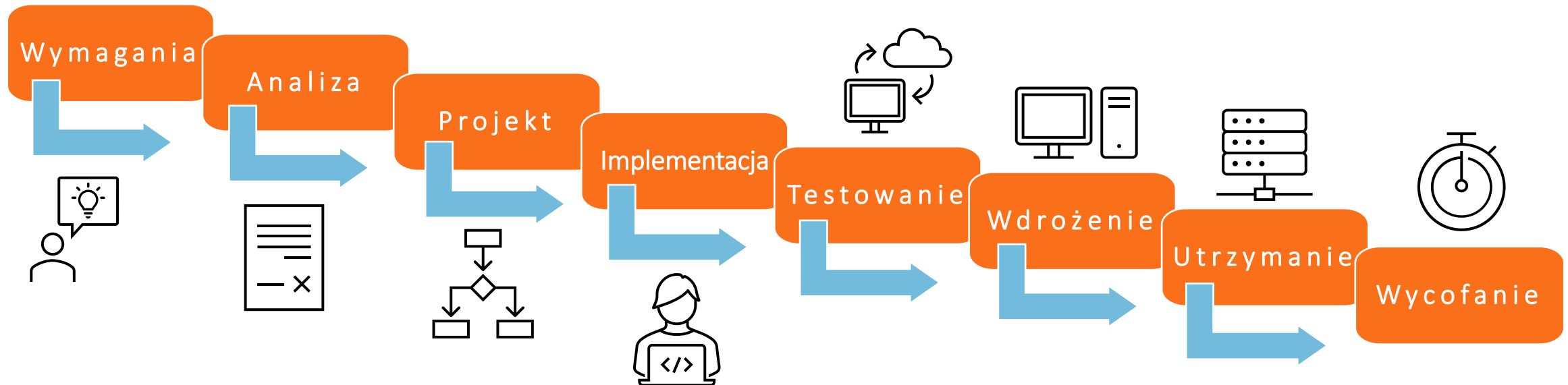
Z tego względu proces wytwarzania oprogramowania dzieli się na fazy.



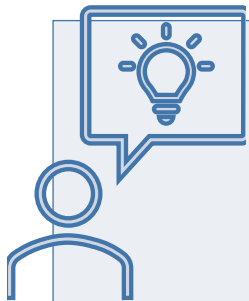
Klasyczny Model Wodospadowy

Model wodospadowy

to sekwencyjny proces rozwoju oprogramowania, gdzie każda faza musi być zakończona przed rozpoczęciem następnej.

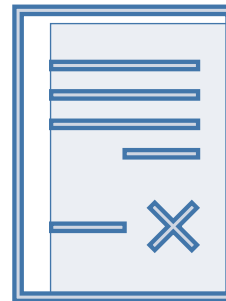


Klasyczny Model Wodospadowy



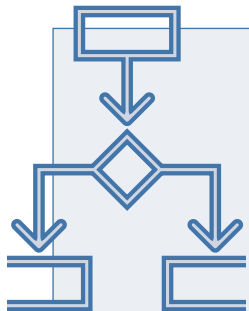
Wymagania (Requirements)

- Identyfikacji funkcjonalnych i нефункциональных wymagań, które muszą zostać spełnione przez projektowane oprogramowanie.



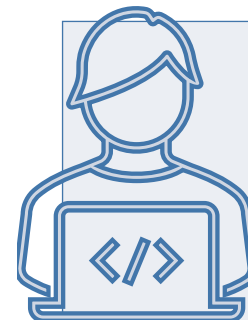
Analiza (Analysis)

- W tej fazie zbierane są wszystkie wymagania od klienta. Są one analizowane i dokumentowane, aby stały się podstawą do kolejnych faz cyklu.



Projektowanie (Design)

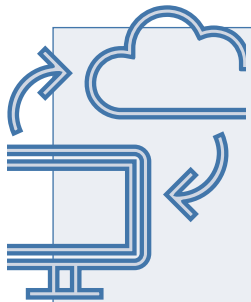
- Po zebraniu wymagań, projektanci i architekci oprogramowania tworzą projekt systemu, który spełni te wymagania. Mogą być tworzone różne artefakty, takie jak diagramy UML, modele danych czy dokumentacja techniczna.



Implementacja (Coding or Development)

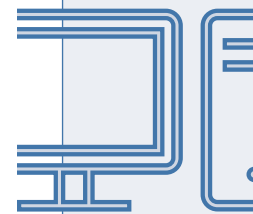
- Na podstawie zatwierdzonych projektów, programiści zaczynają pisać kod. W tej fazie używane są różne języki programowania i narzędzia.

Klasyczny Model Wodospadowy



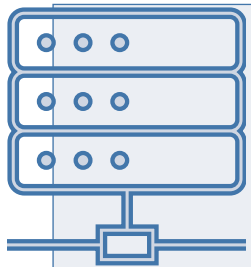
Testowanie (Testing)

- Kod jest testowany pod kątem jakości i zgodności z wymaganiami. Testy mogą być automatyczne lub manualne i mogą obejmować różne poziomy (jednostkowe, integracyjne, systemowe).



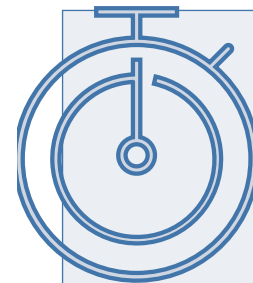
Wdrożenie (Deployment)

- Gotowe oprogramowanie jest instalowane i konfigurowane w środowisku docelowym. Użytkownicy są szkoleni i zaczynają korzystać z systemu.



Utrzymanie (Maintenance)

- Po wdrożeniu, oprogramowanie jest monitorowane, naprawiane i aktualizowane. W tej fazie mogą również być dodawane nowe funkcjonalności.



Wycofanie (Retirement)

- Ostateczna faza, w której oprogramowanie jest wycofywane z użycia. Może to być spowodowane różnymi czynnikami, takimi jak starzenie się technologii, brak wsparcia lub pojawienie się lepszych rozwiązań.

Klasyczny Model Wodospadowy

Zalety Modelu Wodospadowego

Prostota, łatwość zarządzania, jasno określone etapy i cele.

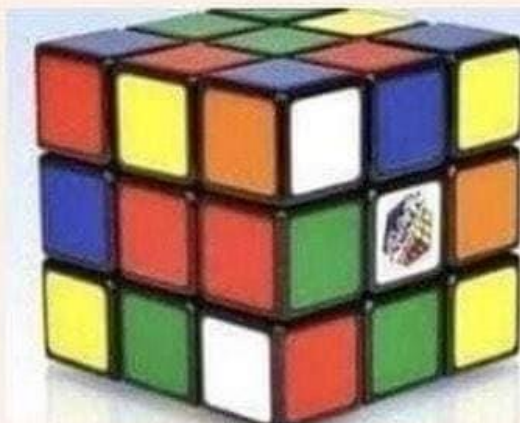
Wady Modelu Wodospadowego

Brak elastyczności, trudności w implementacji zmian, ryzyko opóźnień i przekroczenia budżetu.

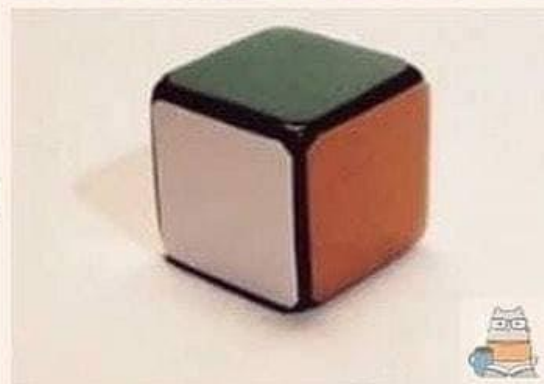
Gdzie się sprawdza?

W projektach o dobrze zdefiniowanych wymaganiach i niskim poziomie niepewności.

Co omówił
wykładowca



Co
zapamiętałem



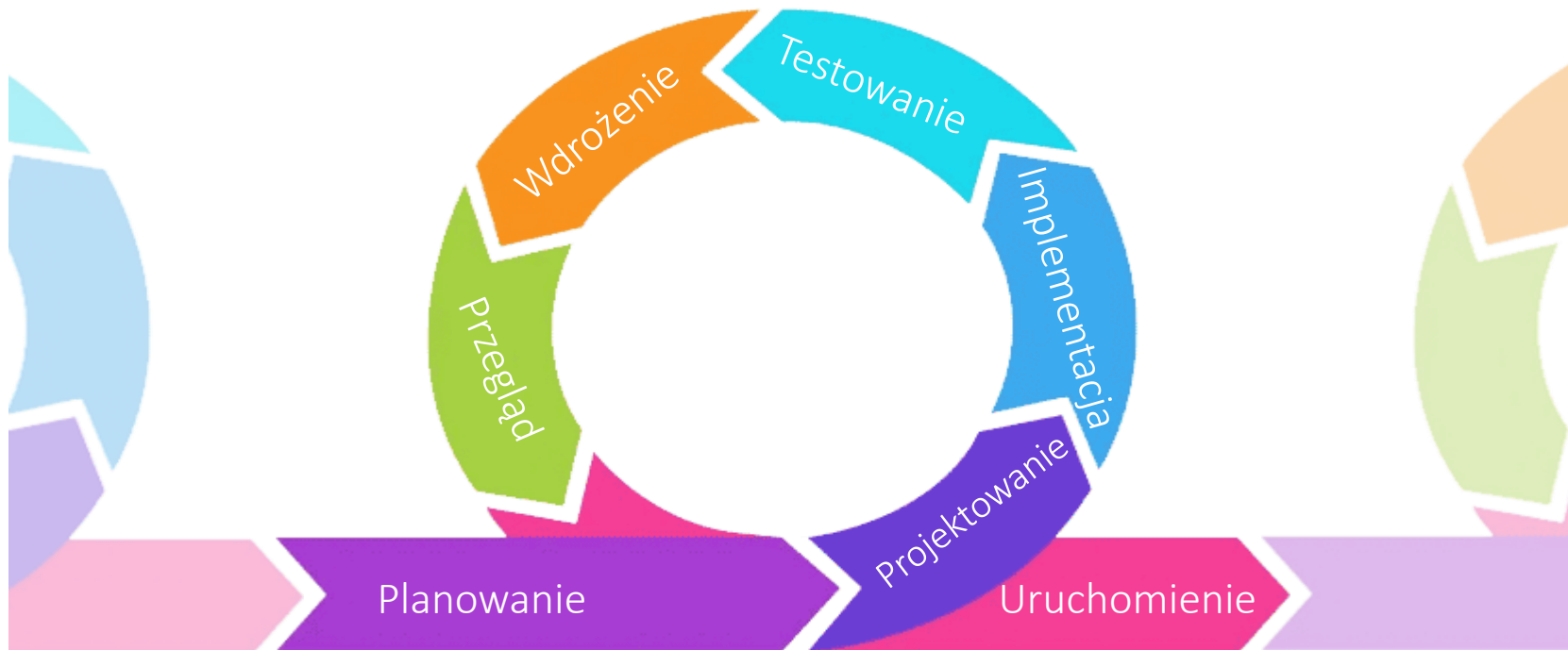
Co było
na zaliczeniu



Metodyka Zwinna

Metodyka Zwinna

to zbiór ramowych zasad i praktyk służących do efektywnego zarządzania i tworzenia programowania, charakteryzujący się elastycznością, adaptacją do zmian i iteracją.

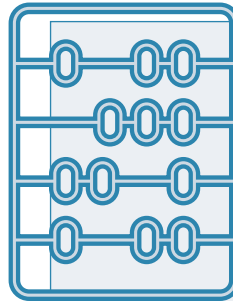


Metodyka Zwinna



Planowanie (Plan)

- Etap, na którym zespół ustala cele i zadania na nadchodzącą iterację.



Przegląd (Review)

- Faza oceny wykonanej pracy i wyników po zakończeniu iteracji.



Uruchomienie (Launch)

- Moment, w którym nowe funkcje lub cały produkt są udostępniane użytkownikom.

Metodyka Zwinna

Zalety Metodyki Zwinnej

Szybka adaptacja, większe zaangażowanie zespołu, lepsza jakość produktu oraz skrócenie czasu dostarczenia produktu.

Wady Metodyki Zwinnej

Trudne do zastosowania w dużych, złożonych projektach, wymaga zaawansowanej komunikacji i współpracy. Problemy z oszacowaniem czasu końca projektu.

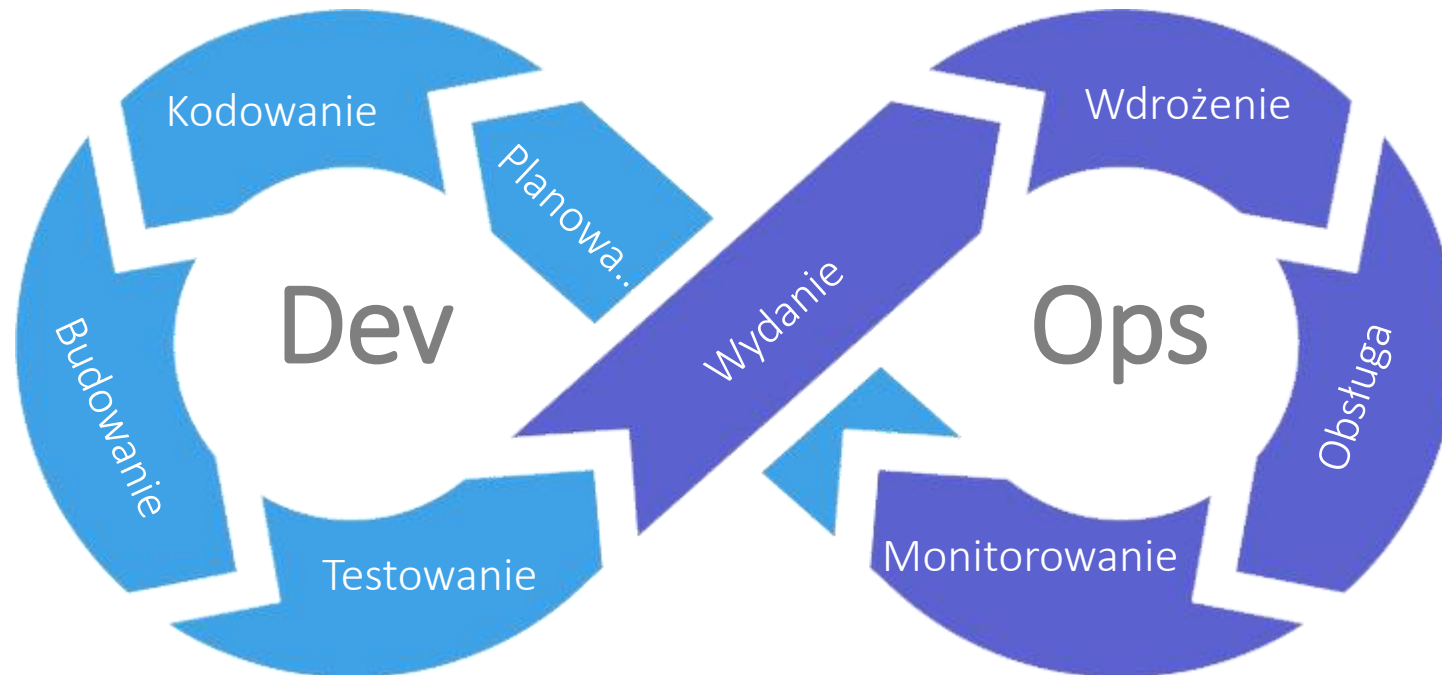
Gdzie się sprawdza?

W projektach z niejasnymi lub dynamicznie zmieniającymi się wymaganiami oraz w projektach, gdzie są częste zmiany lub rozbudowa.

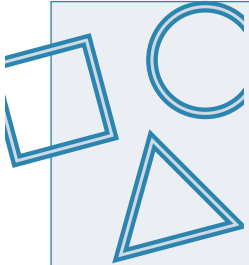
DevOps

DevOps

to zestaw praktyk integrujących rozwój oprogramowania (Dev - Development) i operacje IT (Ops - Operations), aby skrócić cykl tworzenia systemów i dostarczyć wysokiej jakości oprogramowanie.

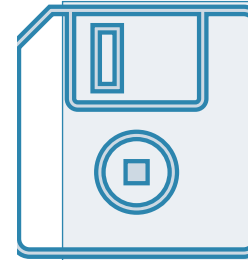


DevOps



Kodowanie (Code) i Budowanie (Build)

- Programowanie i tworzenie kodu źródłowego dla systemu. Który jest kompilowany i przekształcany w oprogramowanie gotowe do wdrożenia.



Wydanie (Release)

- Przygotowywana jest wersja oprogramowania gotowa do wdrożenia włącznie ze środowiskiem wdrożeniowym.



Obsługa (Operate)

- Zarządzanie i utrzymanie systemu w środowisku produkcyjnym. Włącznie ze skalowaniem i obsługą użytkowników.



Monitorowanie (Monitoring)

- Stałe obserwowanie i śledzenie zachowania systemu. Na tym etapie zbierane są dane o działaniu systemu, które mogą być użyte do optymalizacji i zapobiegania problemom oraz kolejnym modernizacją.

DevOps

Zalety DevOps

Szybsze wdrożenia i aktualizacje, które pozwalają organizacjom na błyskawiczne reagowanie na zmiany na rynku. Ponadto, zintegrowane zespoły DevOps ułatwiają komunikację i współpracę między działami programistycznymi i operacyjnymi, co jest kluczowe dla efektywności procesów.

Wady DevOps

Wiąże się z wysokimi kosztami początkowymi, zarówno pod względem zasobów, jak i czasu. Wymaga od organizacji posiadania zgranych zespołów Dev i Ops stanowiących logiczną jedność dzięki rozwiniętej platformie komunikacji.

Gdzie się sprawdza?

Najlepiej sprawdza w organizacjach, które są w szybkim tempie rozwoju i wymagają ciągłego wdrażania nowych funkcji czy poprawek. Jest także idealnym rozwiązaniem dla systemów, które muszą być dostępne 24/7 i które potrzebują elastycznego skalowania zasobów w zależności od obciążenia.

Rzeczywistość

Nurty przestrzeni teleinformatycznej Organizacji

Użytkowanie (działania realizowane przez użytkowników)

Administrowanie (działania realizowane przez administratorów)

HelpDesk (wsparcie użytkowników)

Monitorowanie bezpieczeństwa

Monitorowanie funkcjonowania

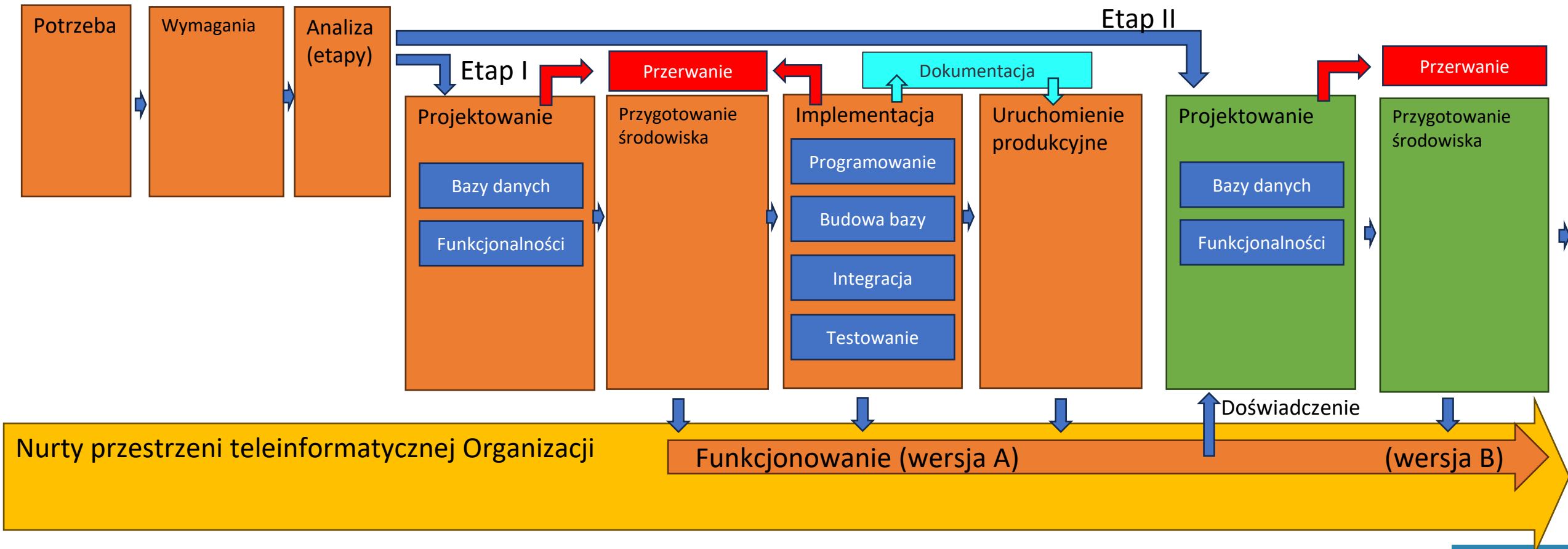
Środowisko aplikacyjne

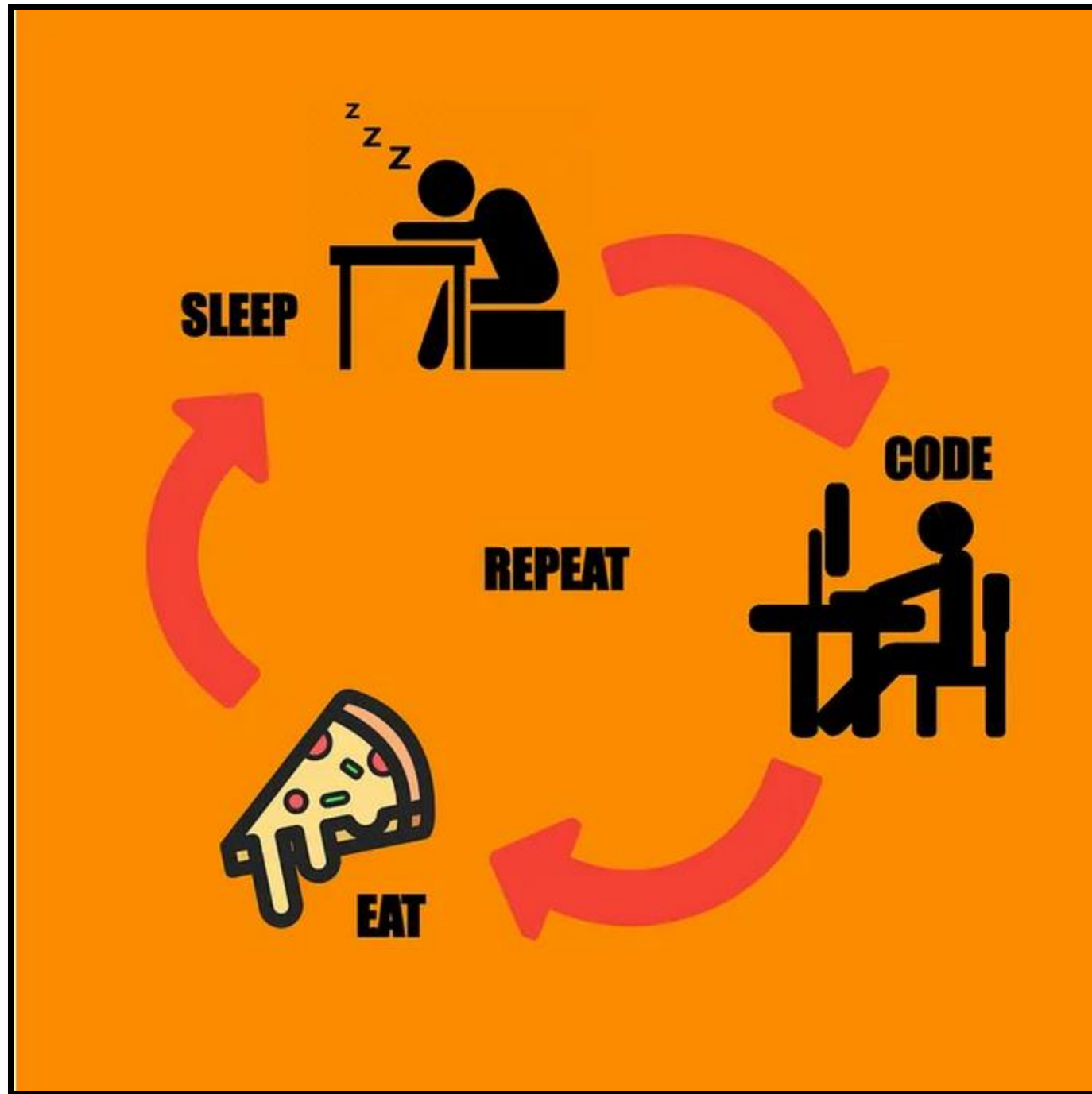
Środowisko bazodanowe

Środowisko pracy (stacje robocze, drukarki, ...)

Środowisko sieciowe

Rzeczywistość





PRINCE2

PRINCE2 (Projects IN Controlled Environments)

to podejście do zarządzania projektami oparte na procesach.

Procesem jest ustrukturyzowany zbiór działań zaprojektowany w celu osiągnięcia konkretnego celu.



PRINCE2

Siadem procesów PRINCE2

- Przygotowanie projektu
- Zarządzanie strategiczne projektem
- Inicjowanie projektu
- Zarządzanie końcem etapu
- Sterowanie etapem
- Zarządzanie dostarczaniem produktów
- Zamknięcie projektu

Siadem tematów PRINCE2

- Uzasadnienie biznesowe
- Organizacja
- Jakość
- Plany
- Ryzyko
- Zmiana
- Postępy

PRINCE2

Zalety PRINCE2

- Strukturalizacja i Standaryzacja: PRINCE2 oferuje dobrze zorganizowaną strukturę, która ułatwia zarządzanie projektami na każdym etapie.
- Skalowalność: Możliwość dostosowania metodyki do różnorodnych projektów sprawia, że PRINCE2 jest uniwersalnym narzędziem w zarządzaniu projektami.

Wady PRINCE2

- Złożoność: Metodyka może być zbyt skomplikowana dla małych, prostych projektów, co sprawia, że jej wdrożenie może być nadmierne.
- Koszt: Wdrożenie i utrzymanie metodyki PRINCE2 może generować znaczne koszty, co może być barierą dla mniejszych organizacji.

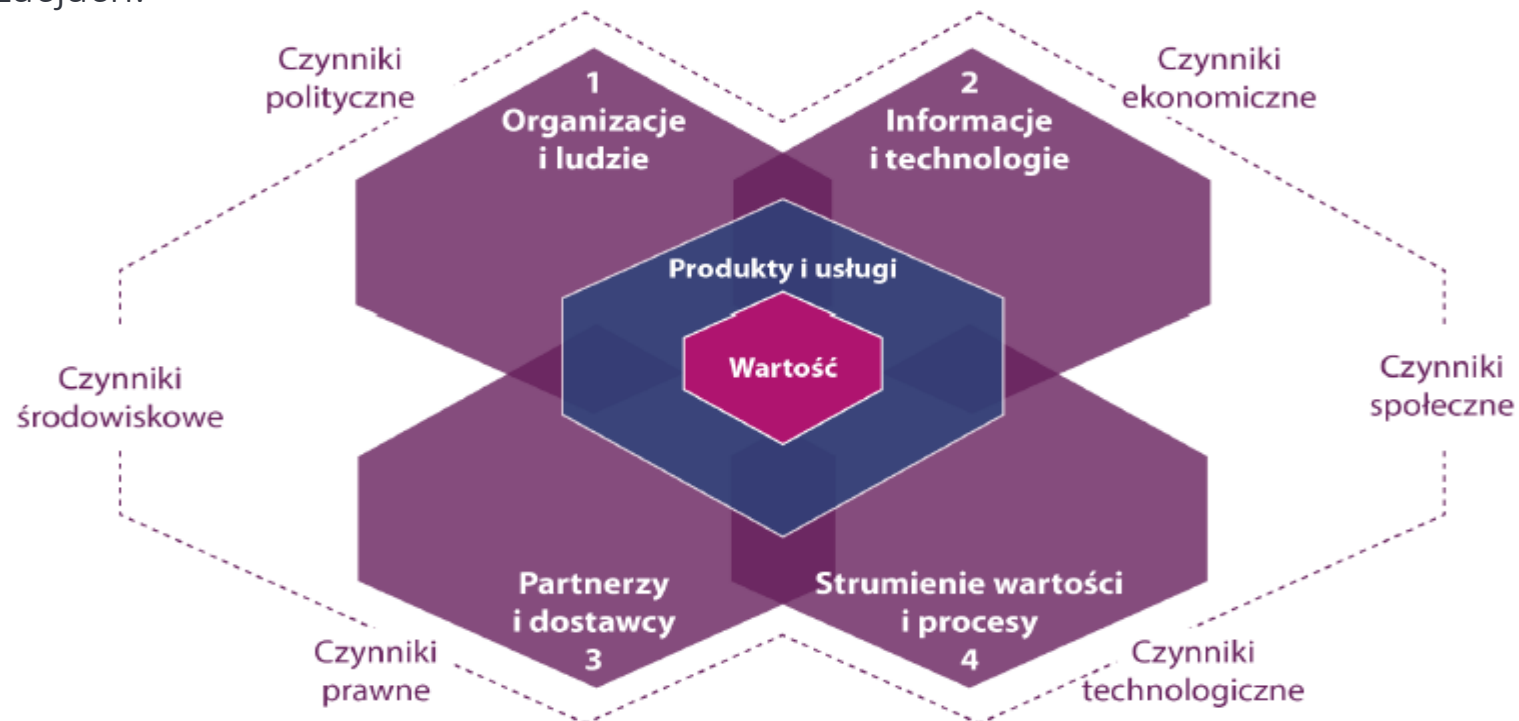
Gdzie się sprawdza?

- Duże projekty korporacyjne: Ze względu na możliwość skalowania i złożoną strukturę, PRINCE2 jest często wykorzystywana w dużych przedsiębiorstwach.
- Projekty publiczne: Metodyka jest popularna w sektorze publicznym, gdzie istnieje potrzeba szczegółowej dokumentacji i standaryzacji procesów.

ITIL

ITIL (Information Technology Infrastructure Library)

to zestaw najlepszych praktyk mających na celu poprawę zarządzania i dostarczania usług informatycznych w organizacjach.



Cztery wymiary zarządzania usługami

Wymiar 1:

- Organizacje i ludzie

Wymiar 2:

- Informacje i technologie

Wymiar 3:

- Partnerzy i dostawcy

Wymiar 4:

- Strumienie wartości i procesy

Czynniki zewnętrzne wpływające na wymiary

- Polityczne (Political)
- Ekonomiczne (Economic)
- Społeczne (Social)
- Technologiczne (Technological)
- Prawne (Legal)
- Środowiskowe (Environmental)

ITIL

Zalety ITIL

- Standaryzacja Procesów: ITIL wprowadza zestaw najlepszych praktyk, co umożliwia standaryzację procesów i procedur w zarządzaniu usługami IT.
- Zorientowanie na Klienta: Metodyka skupia się na dostarczaniu usług, które są zgodne z potrzebami i oczekiwaniami klientów i użytkowników.

Wady ITIL

- Złożoność: Implementacja ITIL może być złożona i czasochłonna, co może stanowić barierę, zwłaszcza dla mniejszych organizacji.
- Koszt: Oprócz kosztów licencyjnych, wdrożenie ITIL często wymaga szkolenia personelu i może generować dodatkowe koszty operacyjne.

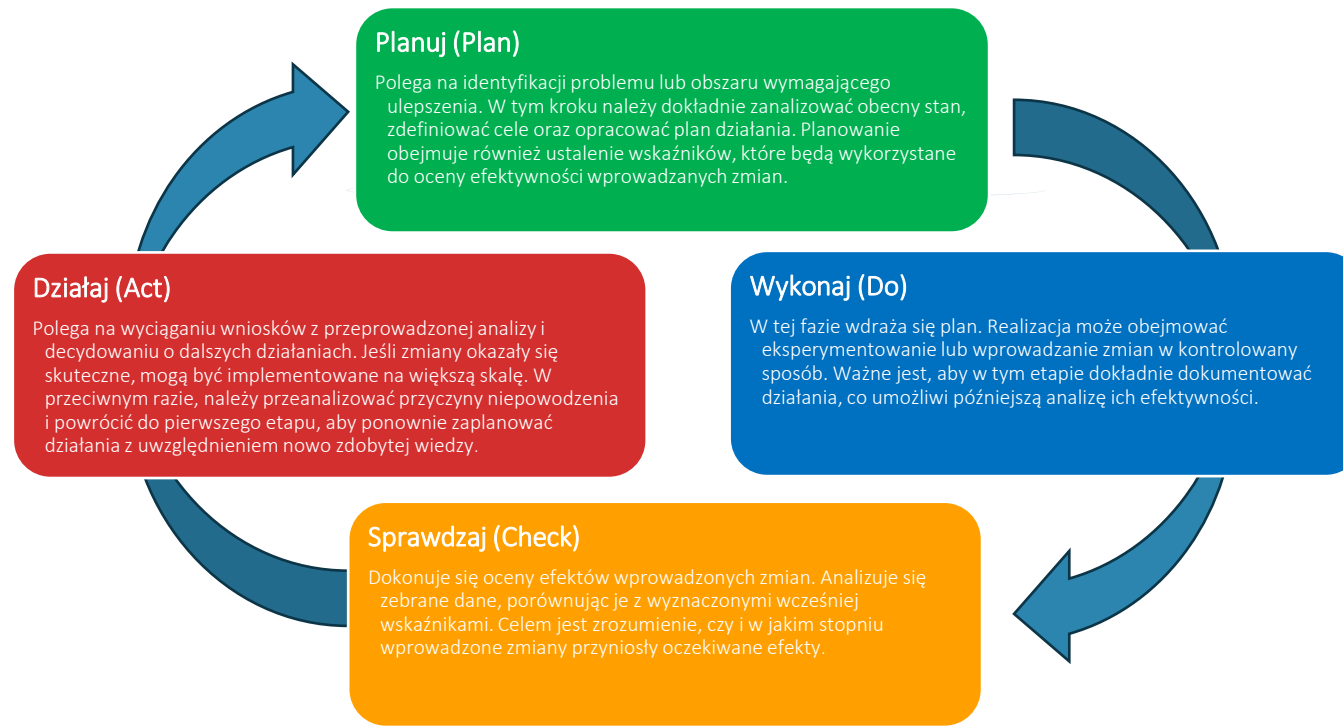
Gdzie się sprawdza?

- Duże Korporacje: Ze względu na skalowalność i złożoność, ITIL jest często stosowany w dużych organizacjach z rozbudowaną infrastrukturą IT.
- Sektor Publiczny: Metodyka jest również popularna w instytucjach publicznych, gdzie istnieje duże zapotrzebowanie na standaryzowane i dobrze zarządzane usługi IT.

PDCA

Model Planuj-Wykonaj-Sprawdzaj-Działaj (PDCA) znany również jako cykl Deminga

jest metodologią zarządzania ciągłym doskonaleniem procesów w organizacjach. Jego głównym celem jest systematyczne wprowadzanie zmian w celu poprawy efektywności i efektywności procesów.



PDCA

Zalety PDCA

- Systematyczne podejście do rozwiązywania problemów: Zapewnia zorganizowaną metodologię do identyfikowania, testowania i wdrażania rozwiązań, co jest szczególnie pomocne w skomplikowanych systemach
- Umożliwia ciągłe doskonalenie: Cykliczna natura modelu sprzyja ciągłemu monitorowaniu i doskonaleniu procesów, co może prowadzić do stopniowych, ale istotnych ulepszeń w działalności organizacji.

Wady PDCA

- Może być czasochłonny: Szczegółowa analiza i iteracyjny charakter modelu mogą wymagać znaczących nakładów czasu, co nie zawsze jest efektywne w szybko zmieniających się warunkach rynkowych.
- Ryzyko nadmiernej biurokratyzacji: Sztywna struktura PDCA może prowadzić do nadmiernej biurokratyzacji i formalizmu, co ogranicza kreatywność i elastyczność w podejmowaniu decyzji.

Gdzie się sprawdza?

- W zarządzaniu jakością i bezpieczeństwem: Jest szczególnie przydatny w kontekście zarządzania jakością i bezpieczeństwem, gdzie systematyczne podejście do identyfikacji i rozwiązywania problemów jest kluczowe.
- W ciągłym doskonaleniu procesów: Idealnie nadaje się do organizacji, które dążą do ciągłego doskonalenia swoich procesów operacyjnych, produkcyjnych lub administracyjnych.

Wybór odpowiedniej metodyki dla różnych organizacji i projektów

