

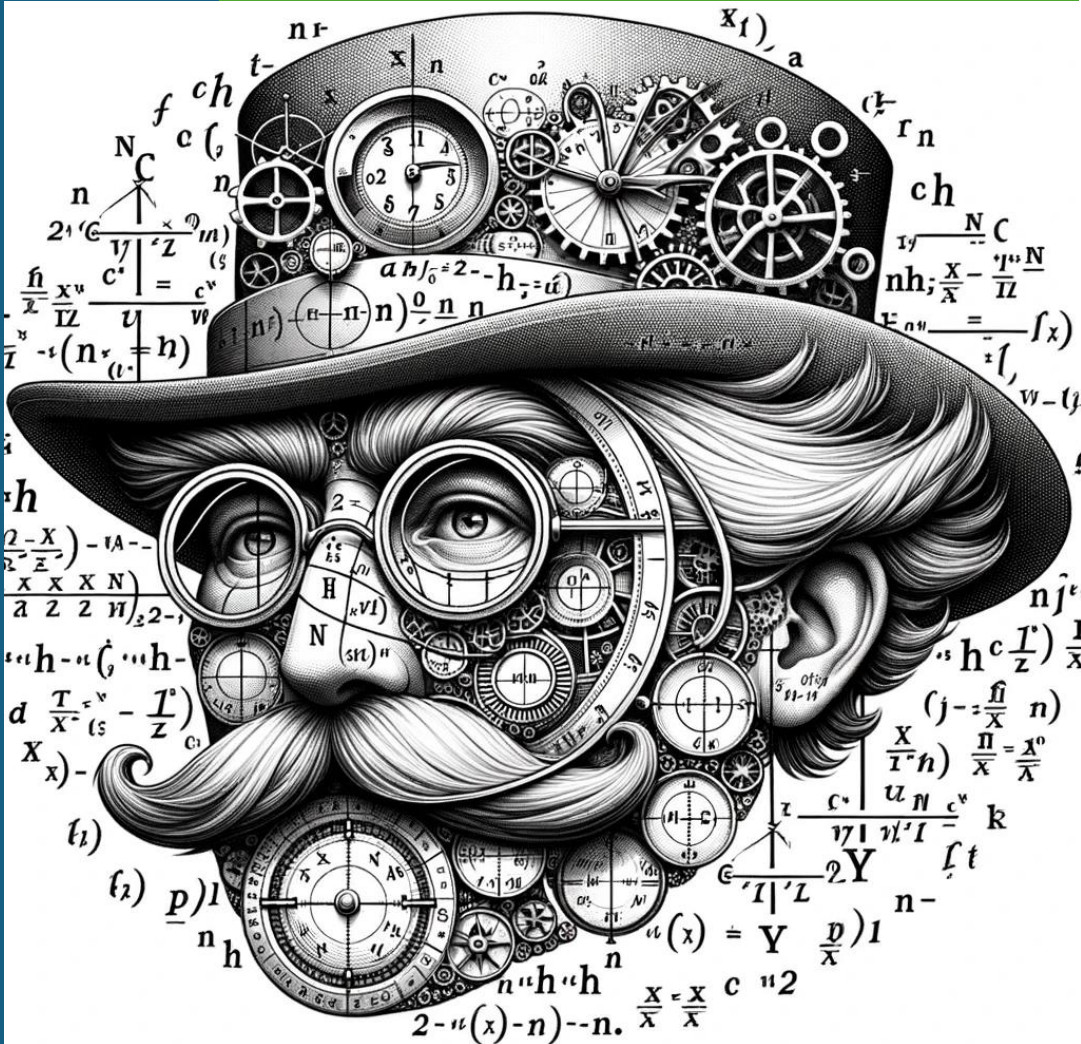
Temat 2 .

dr inż. Michał Malinowski

<https://www.drmalinowski.edu.pl>

Metodyki zarządzania projektami informatycznymi

ZARZĄDZANIE
PROJEKTAMI
INFORMATYCZNYMI



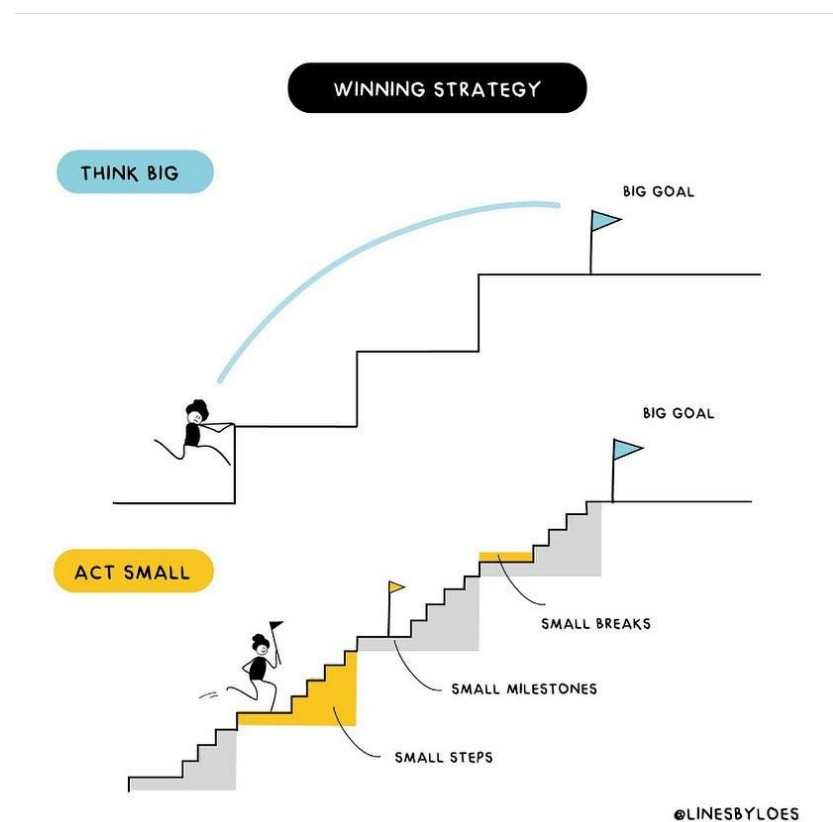
- Klasyczny Model Wodospadowy
- Metodyka Zwinna
- DevOps
- MLOps
- PRINCE2
- ITIL
- Manifest AGILE
- SCRUM
- Kanban
- JIRA
- RUP
- Rzeczywistość
- Narzędzia do zarządzania projektami

Metodologie projektowania i utrzymania

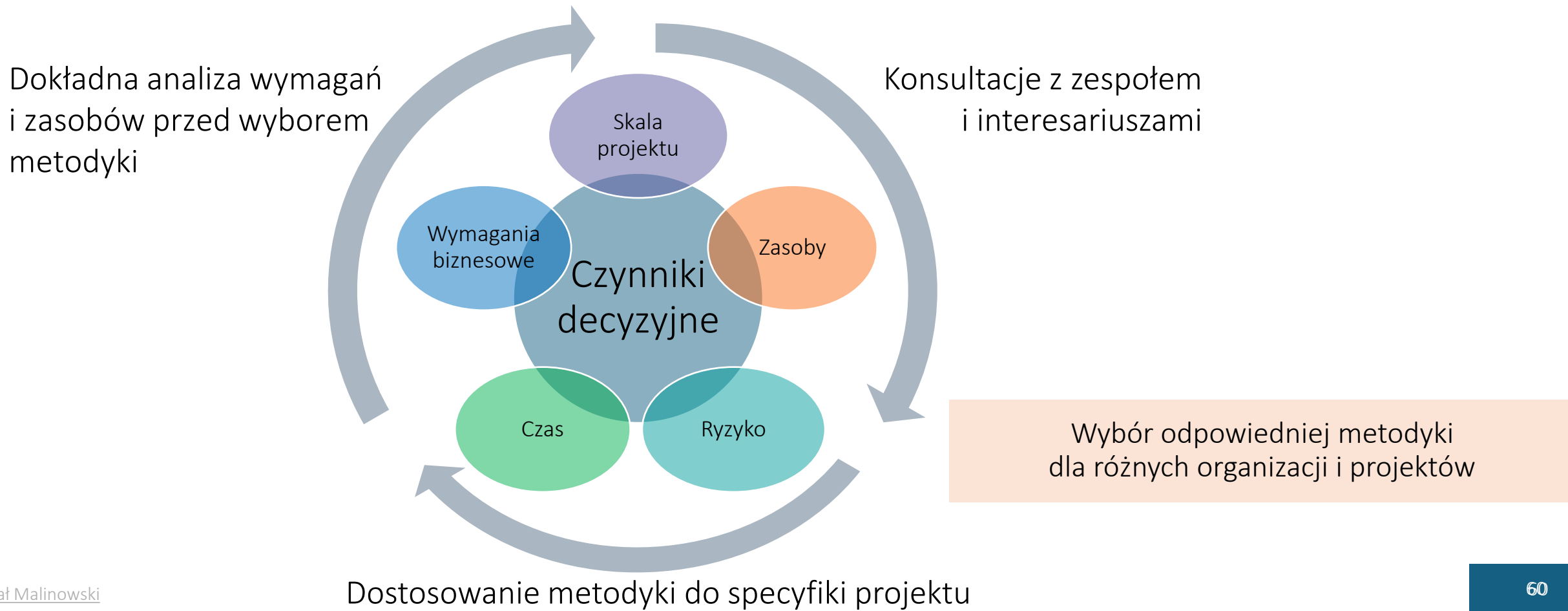
Myśląc o czymś bardzo skomplikowanym, nie próbuj robić wszystkiego jednocześnie.

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

Z tego względu proces wytwarzania systemu dzieli się na fazy.



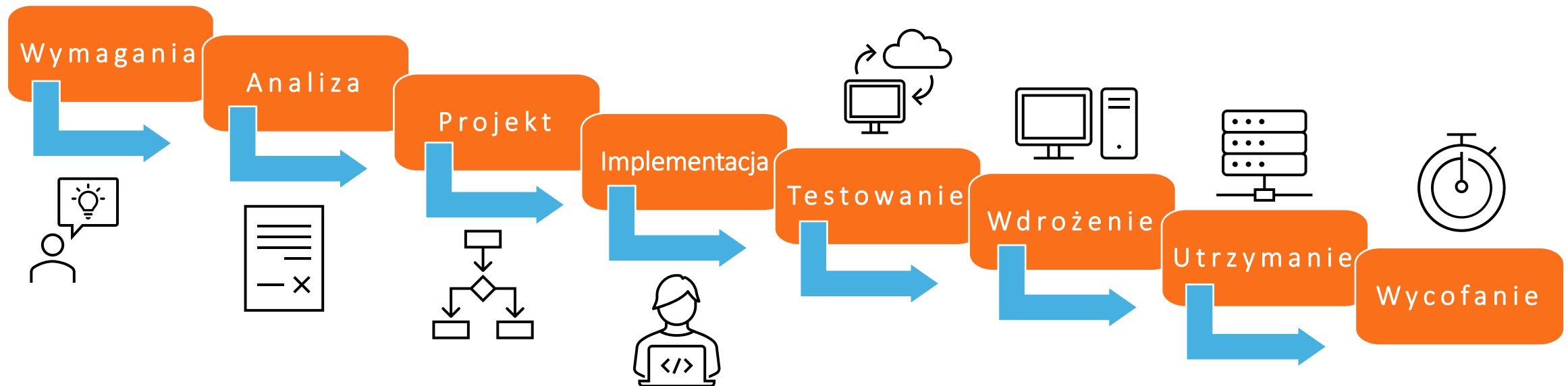
Metodologie projektowania i utrzymania



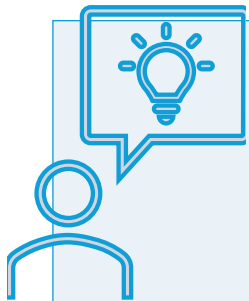
Klasyczny Model Wodospadowy

Model wodospadowy

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

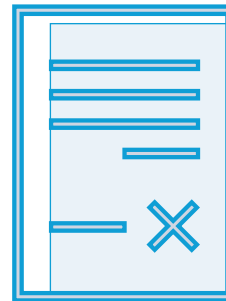


Klasyczny Model Wodospadowy



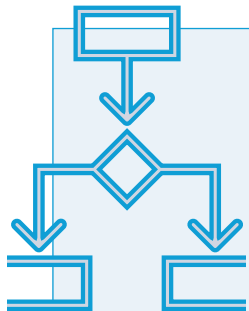
Wymagania (Requirements)

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



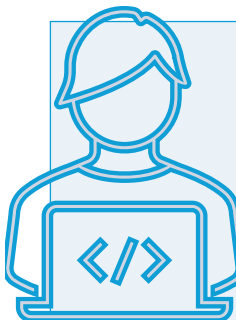
Analiza (Analysis)

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



Projektowanie (Design)

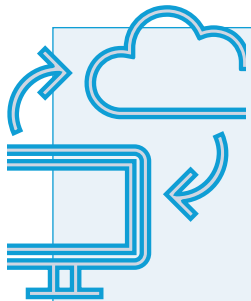
- Po zebraniu wymagań, projektanci i architekci oprogramowania tworzą projekt systemu, który spełni te wymagania. Mogą być tworzone różne artefakty, takie jak diagramy UML, modele danych czy dokumentacja techniczna.



Implementacja (Coding or Development)

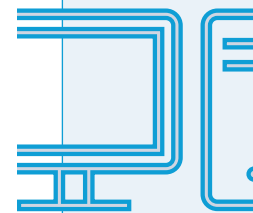
- Na podstawie zatwierdzonych projektów, programiści zaczynają pisać kod. W tej fazie używane są różne języki programowania i narzędzia.

Klasyczny Model Wodospadowy



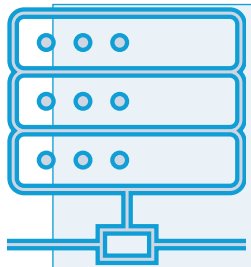
Testowanie (Testing)

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



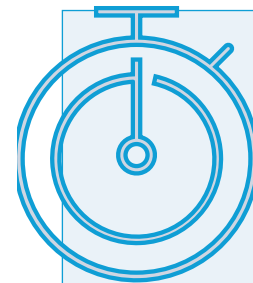
Wdrożenie (Deployment)

- 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.

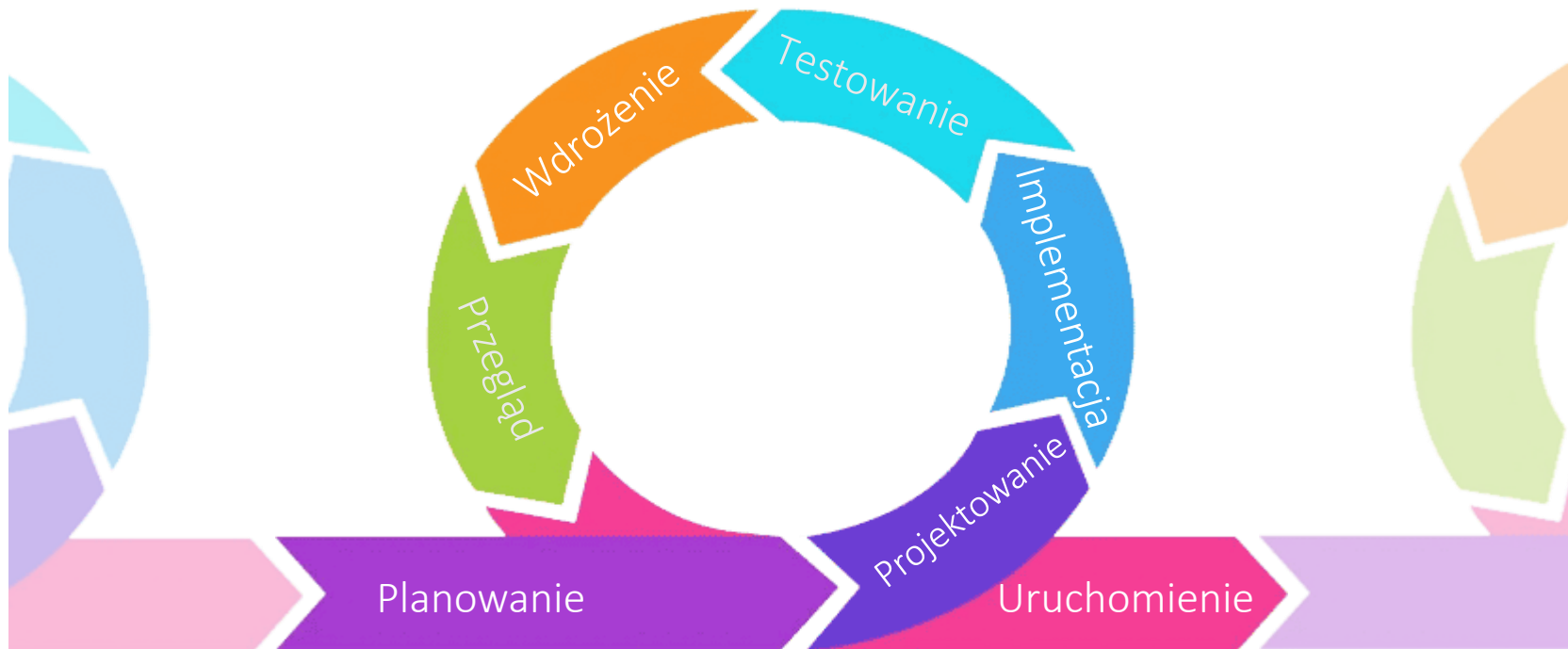
Gdzie się sprawdza?

W projektach o dobrze zdefiniowanych wymaganiach i niskim poziomie niepewności.

Metodyka Zwinna

Metodyka Zwinna

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

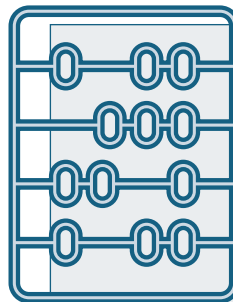


Metodyka Zwinna



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.

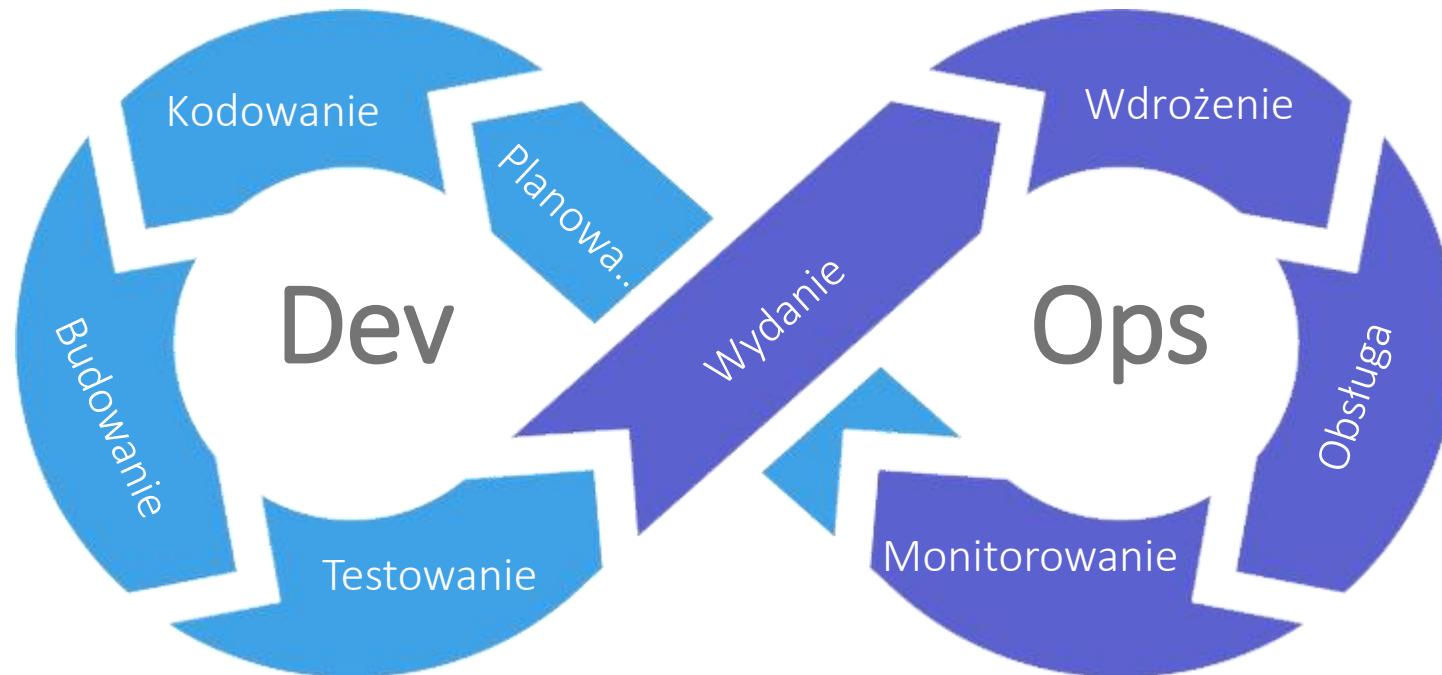
Gdzie się sprawdza?

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.

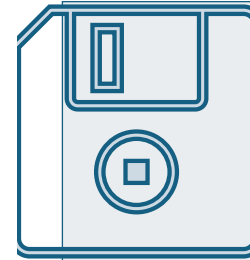


DevOps



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.

MLOps

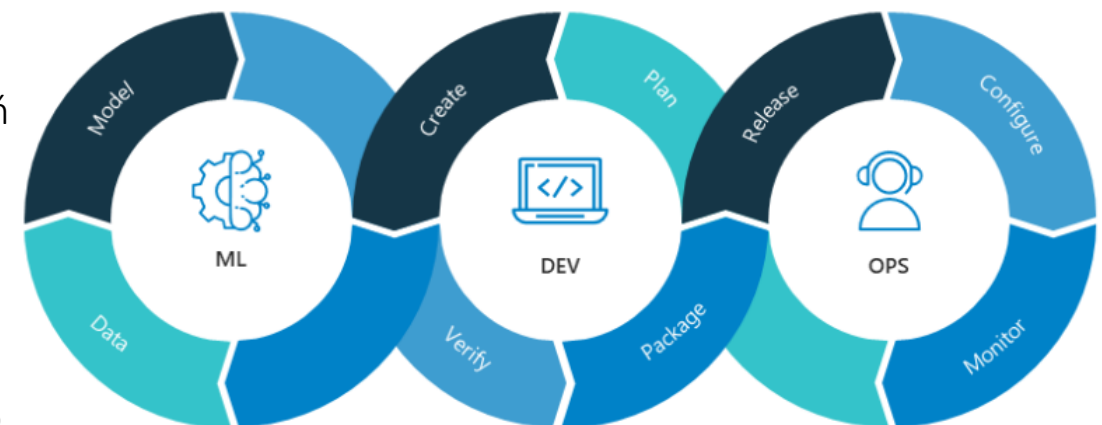
Zarządzanie cyklem życia modeli uczenia maszynowego - MLOps (Machine Learning Operations)

to zbiór praktyk i narzędzi, które łączą uczenie maszynowe (ML) i operacje IT (Operations) w celu zarządzania i wdrażania modeli ML w sposób skalowalny, niezawodny i zautomatyzowany.

MLOps obejmuje cały cykl życia modelu uczenia maszynowego, od tworzenia i trenowania, przez wdrażanie, monitorowanie, utrzymanie, aż po aktualizacje i zarządzanie danymi.

◆ Korzyści wynikające z MLOps

- ✓ **Automatyzacja procesów** – redukcja błędów i przyspieszenie wdrożeń modeli.
- ✓ **Reprodukowalność** – możliwość odtwarzania wyników modeli w różnych środowiskach.
- ✓ **Monitorowanie i zarządzanie modelami** – zapewnienie stabilności i efektywności w dłuższym okresie.
- ✓ **Skalowalność** – łatwe wdrażanie modeli na dużą skalę w chmurze lub on-premise.



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

Siedem procesów PRINCE2

1. Przygotowanie projektu
2. Zarządzanie strategiczne projektem
3. Inicjowanie projektu
4. Zarządzanie końcem etapu
5. Sterowanie etapem
6. Zarządzanie dostarczaniem produktów
7. Zamknięcie projektu

Siedem tematów PRINCE2

1. Uzasadnienie biznesowe
2. Organizacja
3. Jakość
4. Plany
5. Ryzyko
6. Zmiana
7. 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.

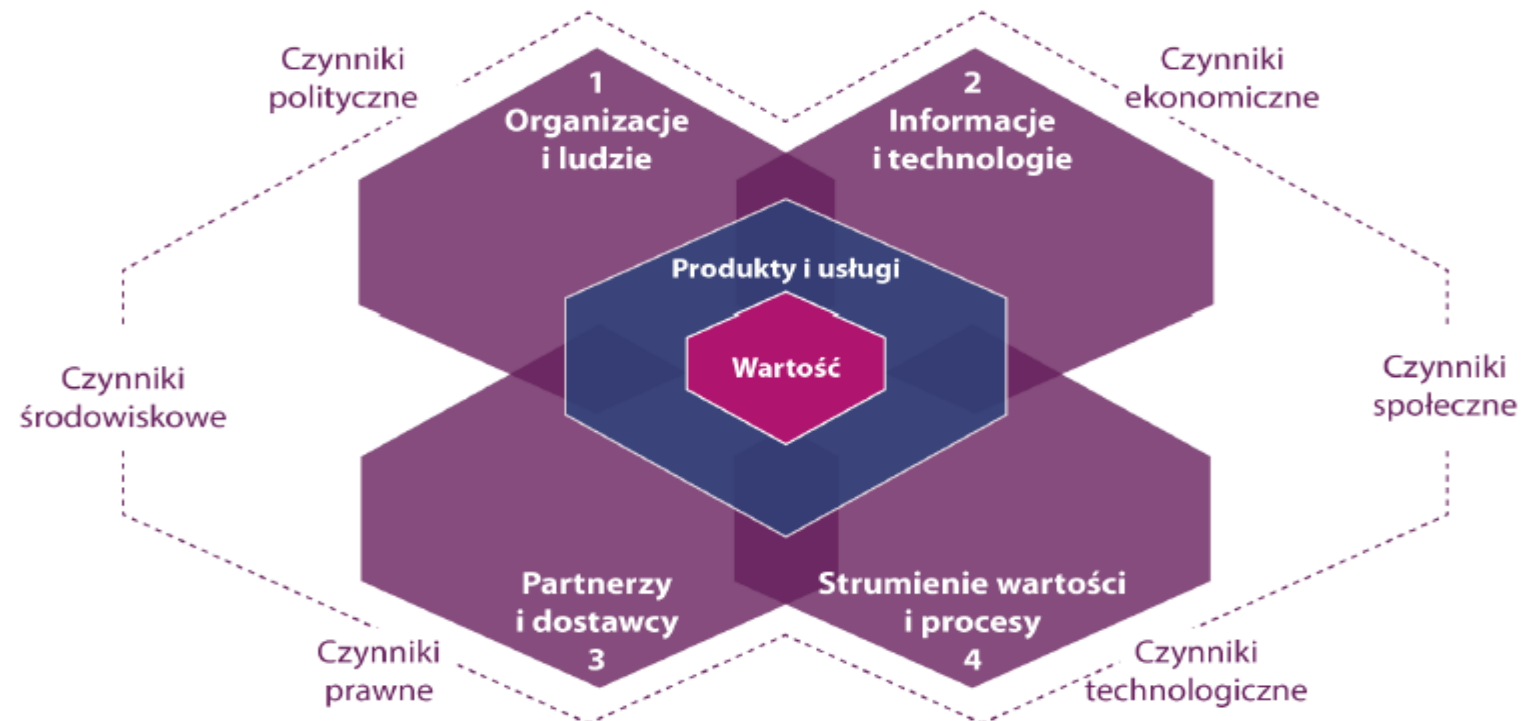
Gdzie się sprawdza?

- 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

1. Polityczne (Political)
2. Ekonomiczne (Economic)
3. Społeczne (Social)
4. Technologiczne (Technological)
5. Prawne (Legal)
6. 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.

Agile

Agile

to iteracyjne podejście do zarządzania projektami, które kładzie nacisk na elastyczność, szybkie dostosowywanie się do zmian oraz bliską współpracę z klientem.

◆ Kluczowe cechy Agile:

- ✓ Iteracyjne dostarczanie wartości w krótkich cyklach (sprintach)
- ✓ Ścisła współpraca między zespołem a klientem
- ✓ Elastyczność w reagowaniu na zmieniające się wymagania
- ✓ Samoorganizujące się zespoły i minimalizacja biurokracji
- ✓ Ciągła poprawa i doskonalenie procesów

◆ Najpopularniejsze metodyki Agile:

- ✓ Scrum – podział pracy na sprinty, role Scrum Mastera, Product Ownera i Zespołu Developerskiego
- ✓ Kanban – wizualizacja przepływu pracy na tablicach, ograniczanie ilości równocześnie realizowanych zadań
- ✓ Extreme Programming (XP) – nacisk na jakość kodu, testowanie, refaktoryzację

Manifest AGILE



Ludzie i interakcje



Działające oprogramowanie



Współpraca z klientem



Reagowanie na zmiany

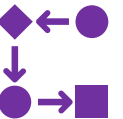


procesy i narzędzia

szczegółową dokumentację

negocjacje umów

realizację planu



Agile

12 zasad Manifestu AGILE

1. Zadowolenie klienta dzięki szybkiemu dostarczaniu wartościowych rozwiązań.
2. Elastyczność w przyjmowaniu zmian nawet na późnych etapach projektu.
3. Regularne dostarczanie działającego oprogramowania (lub innych produktów).
4. Bliska współpraca pomiędzy zespołem a klientem.
5. Tworzenie projektów wokół zmotywowanych i kompetentnych osób.
6. Wspieranie bezpośredniej komunikacji (najlepiej twarzą w twarz).
7. Działające oprogramowanie jako podstawowy miernik postępu.
8. Utrzymywanie stałego tempa pracy, które pozwala na długotrwałe zaangażowanie zespołu.
9. Dążenie do doskonałości technicznej i dobrego projektowania.
10. Prostota jako kluczowy element procesu (minimalizacja niepotrzebnej pracy).
11. Samoorganizujące się zespoły, które podejmują decyzje i dostarczają najlepsze rozwiązania.
12. Regularna refleksja zespołu nad swoimi działaniami i poszukiwanie sposobów na ich udoskonalenie.

SCRUM

SCRUM

to jedna z najpopularniejszych metodyk Agile, stosowana głównie w zarządzaniu projektami IT.

Opiera się na podziale pracy na krótkie cykle zwane **sprintami** i samoorganizujących się zespołach.

Cechy SCRUM

- ✓ Iteracyjne podejście – sprinty trwają zwykle 1–4 tygodnie
- ✓ Transparentność – każdy wie, co robi zespół
- ✓ Ciągłe doskonalenie – retrospektywy pomagają ulepszać procesy

Role w SCRUM

- ✓ Product Owner (Właściciel Produktu) – zarządza backlogiem produktu, definiuje wymagania i priorytety
- ✓ Scrum Master (Mistrz SCRUM) – facylitator zespołu, usuwa przeszkody, dba o przestrzeganie zasad SCRUM
- ✓ Zespół Developerski – programiści, testerzy, UX/UI designerzy – realizują sprinty i dostarczają wartość

SCRUM

Artefakty SCRUM

to kluczowe elementy dokumentujące postęp i zakres prac w metodyce SCRUM. Służą do organizacji zadań i zapewnienia przejrzystości pracy zespołu.

Artefakt		Przykładowy scenariusz
Backlog Produktu (Product Backlog)	Lista funkcjonalności do wykonania	W projekcie Product Owner dodaje do backlogu wymagania biznesowe, np.: - Możliwość filtrowania produktów po cenie i kategorii - Integracja z płatnościami online - Powiadomienia e-mail o statusie zamówienia
Backlog Sprintu (Sprint Backlog)	Konkretne zadania na dany sprint	Z backlogu produktu zespół wybiera zadania na bieżący sprint, np.: - Stworzenie API do obsługi płatności (Backend Developer) - Projekt strony koszyka zakupowego (Frontend Developer, UX Designer) - Testy jednostkowe systemu płatności (QA Engineer)
Przyrost (Increment)	Gotowa, działająca część produktu	Na koniec sprintu gotowy do wdrożenia fragment aplikacji: - Działająca funkcja filtrowania produktów - Możliwość dodawania przedmiotów do koszyka - Pierwsza wersja integracji z systemem płatności (PayU, Stripe)

SCRUM

Proces tworzenia backlogu produktu

- ✓ Zbieranie wymagań – identyfikacja potrzeb użytkowników i interesariuszy.
- ✓ Tworzenie elementów backlogu – opisanie funkcjonalności w postaci **User Stories** (*Jako [użytkownik], chcę [funkcja], aby [wartość]*).
- ✓ Priorytetyzacja – ustalanie kolejności realizacji zadań zgodnie z wartością biznesową.
- ✓ Szacowanie pracochłonności – przypisanie wartości **Story Points** lub innych metryk wysiłku.
- ✓ Ciągła aktualizacja – backlog jest na bieżąco modyfikowany i dostosowywany do zmian w projekcie.

Przykładowe elementy backlogu produktu

ID	Opis (User Story)	Priorytet	Szacowany wysiłek (Story Points)
#1	Jako użytkownik chcę zarejestrować konto za pomocą e-maila i hasła.	Wysoki	5
#2	Jako użytkownik chcę filtrować produkty według ceny i kategorii.	Średni	8
#3	Jako administrator chcę mieć dostęp do panelu zarządzania użytkownikami.	Wysoki	13

SCRUM

Planowanie Sprintu (Sprint Planning)

to pierwsze wydarzenie w sprincie, podczas którego zespół wybiera zadania do realizacji na podstawie backlogu produktu. Kluczową częścią tego procesu jest **szacowanie pracochłonności zadań**.

Proces planowania sprintu

- 1 Określenie celu sprintu – zespół i Product Owner ustalają, co ma zostać dostarczone.
- 2 Wybór elementów backlogu sprintu – wybierane są zadania o najwyższym priorytecie.
- 3 Szacowanie pracochłonności zadań – określenie, ile wysiłku wymaga każde zadanie (np. w Story Points).
- 4 Zadania i ich podział – dekompozycja dużych elementów backlogu na mniejsze zadania.
- 5 Zdefiniowanie kryteriów akceptacji – jasne warunki, które muszą być spełnione, aby zadanie zostało ukończone.

SCRUM

Ceremonie SCRUM

to regularne spotkania organizowane w ramach metodyki SCRUM, które pomagają w planowaniu, monitorowaniu i doskonaleniu procesu pracy zespołu.

Nazwa wydarzenia	Cel	Uczestnicy	Częstotliwość
Planowanie Sprintu (Sprint Planning)	Określenie zakresu pracy i celów sprintu.	Scrum Master, Product Owner, Zespół Developerski	Na początku każdego sprintu
Codzienny Stand-up (Daily Scrum)	Krótkie omówienie postępów, blokad i planów na dzień.	Zespół Developerski, Scrum Master	Codziennie (max 15 min)
Przegląd Sprintu (Sprint Review)	Prezentacja wyników sprintu, omówienie dostarczonych funkcjonalności.	Scrum Master, Product Owner, Zespół, interesariusze	Na końcu każdego sprintu
Retrospektywa Sprintu (Sprint Retrospective)	Analiza tego, co poszło dobrze i co można poprawić.	Scrum Master, Zespół Developerski	Na końcu każdego sprintu

Kanban

Kanban

to metoda zarządzania pracą, która koncentruje się na **wizualizacji procesu**, **ograniczaniu ilości zadań w toku (Work In Progress – WIP)** oraz **ciągłym doskonaleniu**.

Jest stosowana w zespołach IT, produkcji oraz zarządzaniu projektami.

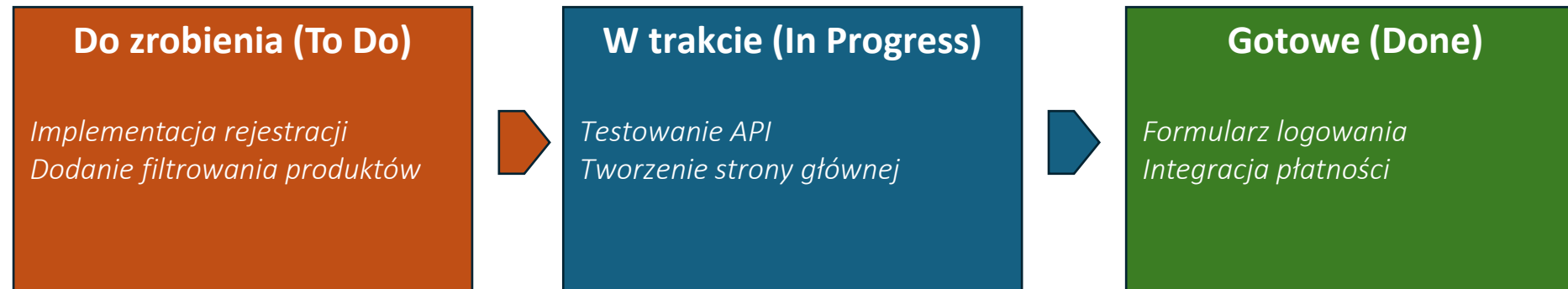
Kluczowe zasady Kanban

- ✓ **Wizualizacja pracy** – stosowanie tablicy Kanban do śledzenia postępu zadań.
- ✓ **Limit WIP (Work In Progress)** – ograniczenie liczby jednocześnie wykonywanych zadań.
- ✓ **Ciągły przepływ pracy** – zadania przechodzą przez kolejne etapy, unikając blokad.
- ✓ **Optymalizacja procesu** – analiza efektywności i dostosowanie zasad.

Kanban

Tablica Kanban

Tablica Kanban dzieli zadania na etapy, np.:



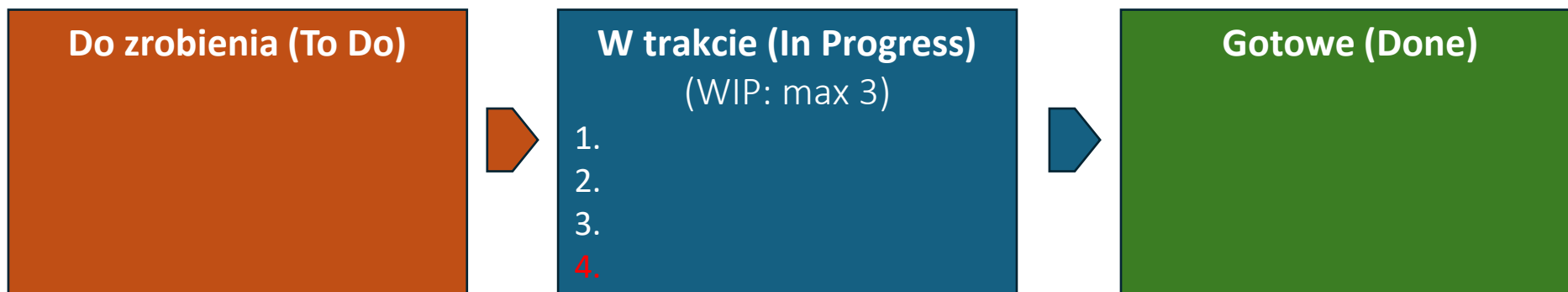
 **Zasada:** Nowe zadanie może zostać podjęte tylko wtedy, gdy zwolni się miejsce w kolumnie „W trakcie” (ograniczenie WIP).

Kanban

Limity WIP (Work In Progress)

to ograniczenie liczby zadań, które mogą być jednocześnie realizowane w danej kolumnie tablicy Kanban.

- ✓ Dlaczego stosujemy limity WIP?
- ✓ Zapobiegają przeciążeniu zespołu.
- ✓ Zwiększają koncentrację na bieżących zadaniach.
- ✓ Ułatwiają identyfikację wąskich gardeł w procesie.



Kanban

Kanban vs. SCRUM

Cecha	Kanban	SCRUM
Struktura	Elastyczna, ciągły przepływ zadań	Podzielony na sprinty
Planowanie	Brak określonych iteracji	Sprinty o stałej długości
Wykorzystanie	Świetny do wsparcia IT, DevOps, utrzymania	Lepszy dla projektów z nowymi funkcjami
Role	Brak formalnych ról	Scrum Master, Product Owner, Zespół

Kiedy wybrać Kanban?

- ✓ Gdy zadania pojawiają się dynamicznie i mają różny priorytet.
 - ✓ W zespołach zajmujących się utrzymaniem i wsparciem IT (np. DevOps, Helpdesk).
 - ✓ Gdy potrzebna jest szybka reakcja na zmiany i optymalizacja przepływu pracy.
-  **Podsumowanie:** Kanban to elastyczna metoda, która pozwala na płynne zarządzanie zadaniami, bez sztywnych iteracji jak w SCRUM. 

JIRA

JIRA

to jedno z najpopularniejszych narzędzi do zarządzania projektami i śledzenia zadań, stworzone przez firmę **Atlassian**. Jest szeroko stosowane w zespołach programistycznych, zwłaszcza w metodykach **Agile**, **SCRUM** i **Kanban**.



<https://www.atlassian.com/pl/software/jira>

Kluczowe funkcje JIRA

- ✓ Zarządzanie backlogiem – tworzenie i priorytetyzowanie zadań.
- ✓ Tablice SCRUM i Kanban – wizualizacja pracy w sprintach lub w modelu ciągłego przepływu.
- ✓ Śledzenie postępu zadań – statusy, priorytety, przypisane osoby.
- ✓ Automatyzacja procesów – reguły automatyczne dla workflow (np. automatyczne przejście zadania do „Do przeglądu”).
- ✓ Raportowanie i analizy – wykresy spalania (burndown charts), velocity, analiza cyklu życia zadania.
- ✓ Integracje – współpraca z narzędziami takimi jak **Confluence**, **Bitbucket**, **GitHub**, **Slack**.

JIRA

JIRA w SCRUM i Kanban

◆ JIRA w SCRUM

- ✓ Planowanie sprintów
- ✓ Backlog produktu i backlog sprintu
- ✓ Raporty spalania i velocity

◆ JIRA w Kanban

- ✓ Wizualizacja przepływu pracy
- ✓ Limity WIP
- ✓ Analiza czasu realizacji

Przykładowe typy zgłoszeń w JIRA

Typ zgłoszenia	Opis
Story	Opis funkcjonalności z perspektywy użytkownika (np. „Jako użytkownik chcę zresetować hasło”).
Task	Zadanie do wykonania (np. „Zaimplementować formularz logowania”).
Bug	Błąd w systemie wymagający naprawy.
Epic	Duże wymaganie składające się z kilku Story (np. „Moduł płatności online”).

RUP

RUP (Rational Unified Process)

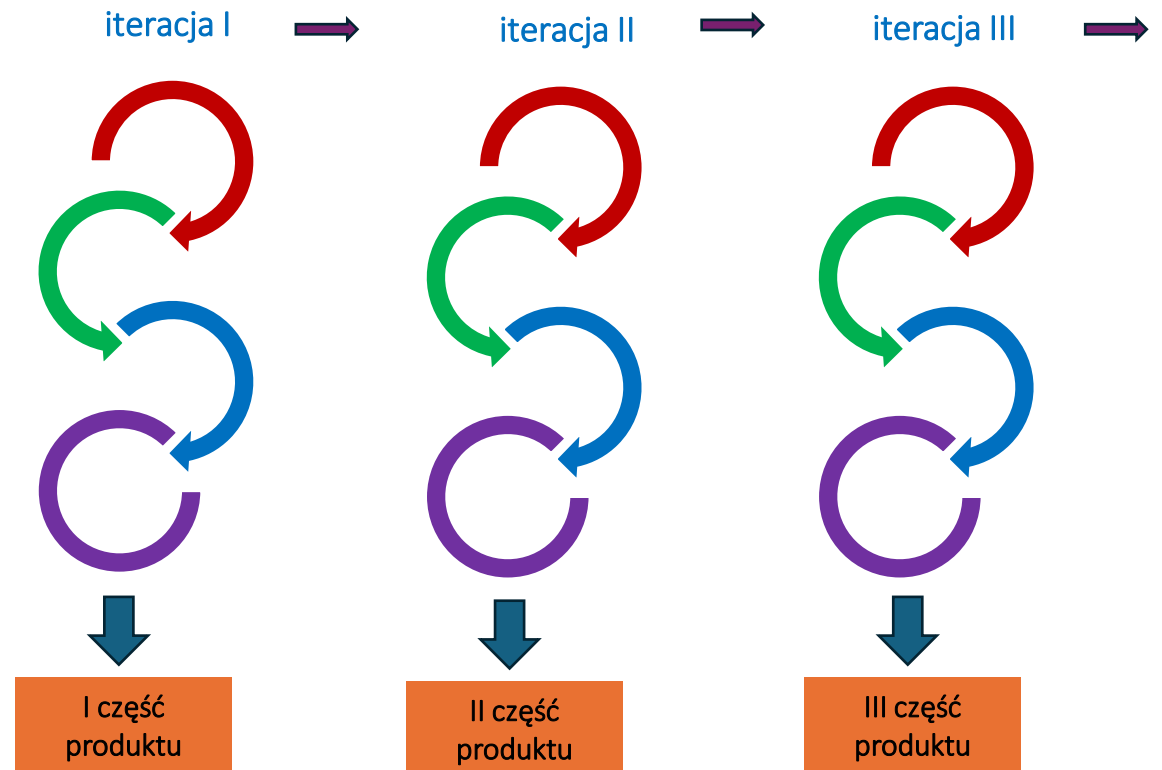
jest iteracyjną i przyrostową metodyką wytwarzania oprogramowania skoncentrowana na architekturze i ryzyku.

W każdej iteracji wytwarzany jest fragment systemu, który jest udostępniany klientowi.

Pozwala to na uzyskanie szybkiej informacji zwrotnej i upewnieniu się, że zespół realizujący projekt dobrze rozumiał wymagania i oczekiwania klienta.

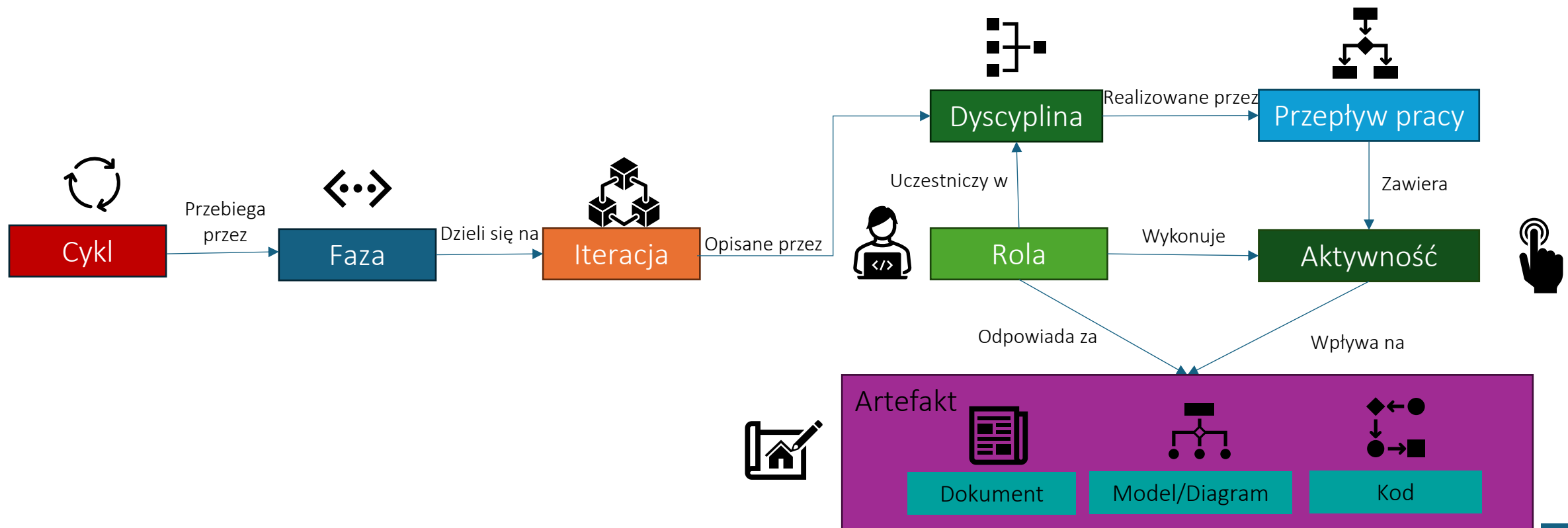
Szybkie wykrycie ewentualnych problemów lub nieporozumień, pozwala szybko wprowadzić odpowiednie modyfikacje.

Celem tej metodyki jest produkcja oprogramowania wysokiej jakości w przewidywanym dla klienta czasie i budżecie oraz akceptowalnym ryzykiem



RUP

Architektura metodyki RUP



Rzeczywistość

Nurty przestrzeni teleinformatycznej Organizacji

Użytkowanie (działania realizowane przez użytkowników)

Administrowanie (działania realizowane przez administratorów)

HelpDesk (wsparcie użytkowników)

Monitorowanie bezpieczeństwa

Monitorowanie funkcjonowania

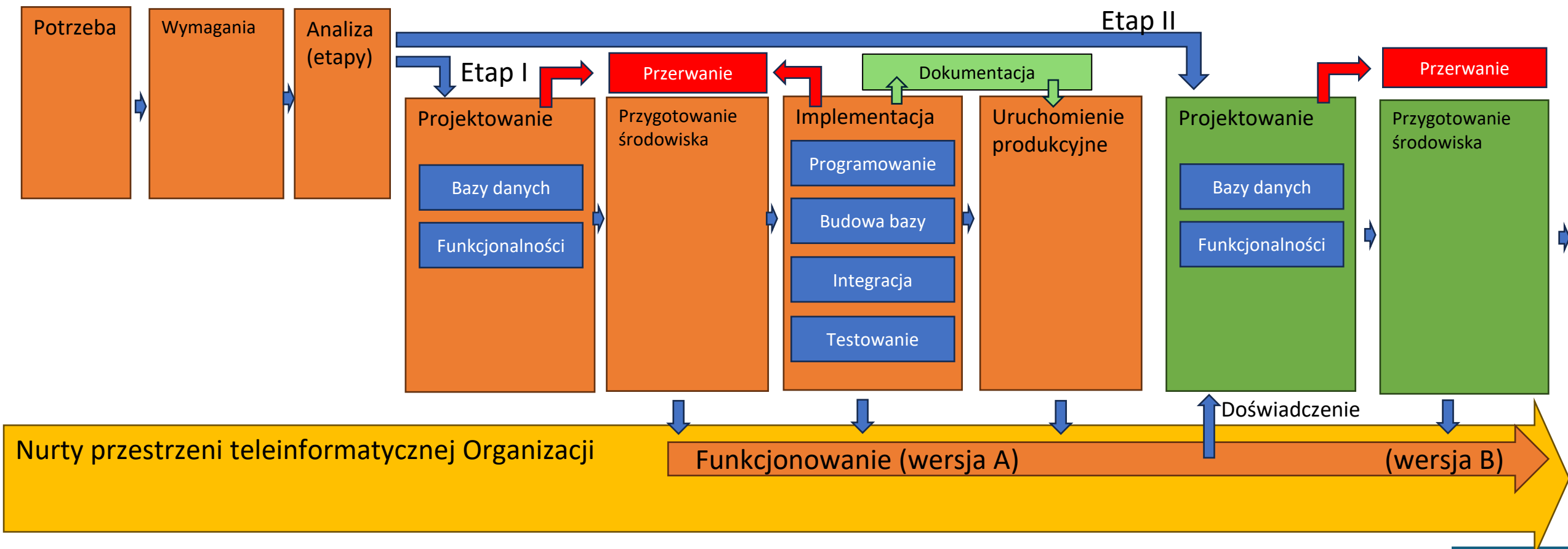
Środowisko aplikacyjne

Środowisko bazodanowe

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

Środowisko sieciowe

Rzeczywistość



Porównanie narzędzi do zarządzania projektami

Cecha	JIRA 	MS Project 	ClickUp 	Monday.com 
Główne zastosowanie	Zarządzanie projektami IT, Agile (SCRUM, Kanban)	Zaawansowane zarządzanie projektami, planowanie zasobów	Uniwersalne zarządzanie zadaniami i projektami	Wizualne zarządzanie pracą, dla zespołów marketingowych i HR
Metodyki Agile (SCRUM, Kanban)	✅ Tak – pełne wsparcie	⚠️ Ograniczone	✅ Tak	✅ Tak
Planowanie Gantta	❌ Nie	✅ Tak	✅ Tak	✅ Tak
Zarządzanie backlogiem	✅ Tak (Sprinty, Backlog Produktu)	❌ Nie	✅ Tak	✅ Tak
Śledzenie czasu pracy	✅ Tak (wtyczki, raporty)	✅ Tak	✅ Tak	✅ Tak
Raportowanie i analizy	✅ Bardzo zaawansowane	✅ Zaawansowane	✅ Dobre	✅ Dobre
Integracje	GitHub, Bitbucket, Confluence, Slack	Office 365, Power BI	Slack, Google Drive, GitHub	Trello, Google Drive, Slack
Automatyzacje	✅ Tak (JQL, reguły automatyczne)	✅ Tak	✅ Tak (rozbudowane)	✅ Tak (łatwe do skonfigurowania)
Łatwość obsługi	❌ Średnia – wymaga konfiguracji	❌ Trudniejsza – skomplikowane interfejsy	✅ Intuicyjna	✅ Bardzo intuicyjna
Cena	💰💰 (średnia)	💰💰💰 (droga)	💰 (darmowy plan, tanie płatne opcje)	💰💰 (średnia)

