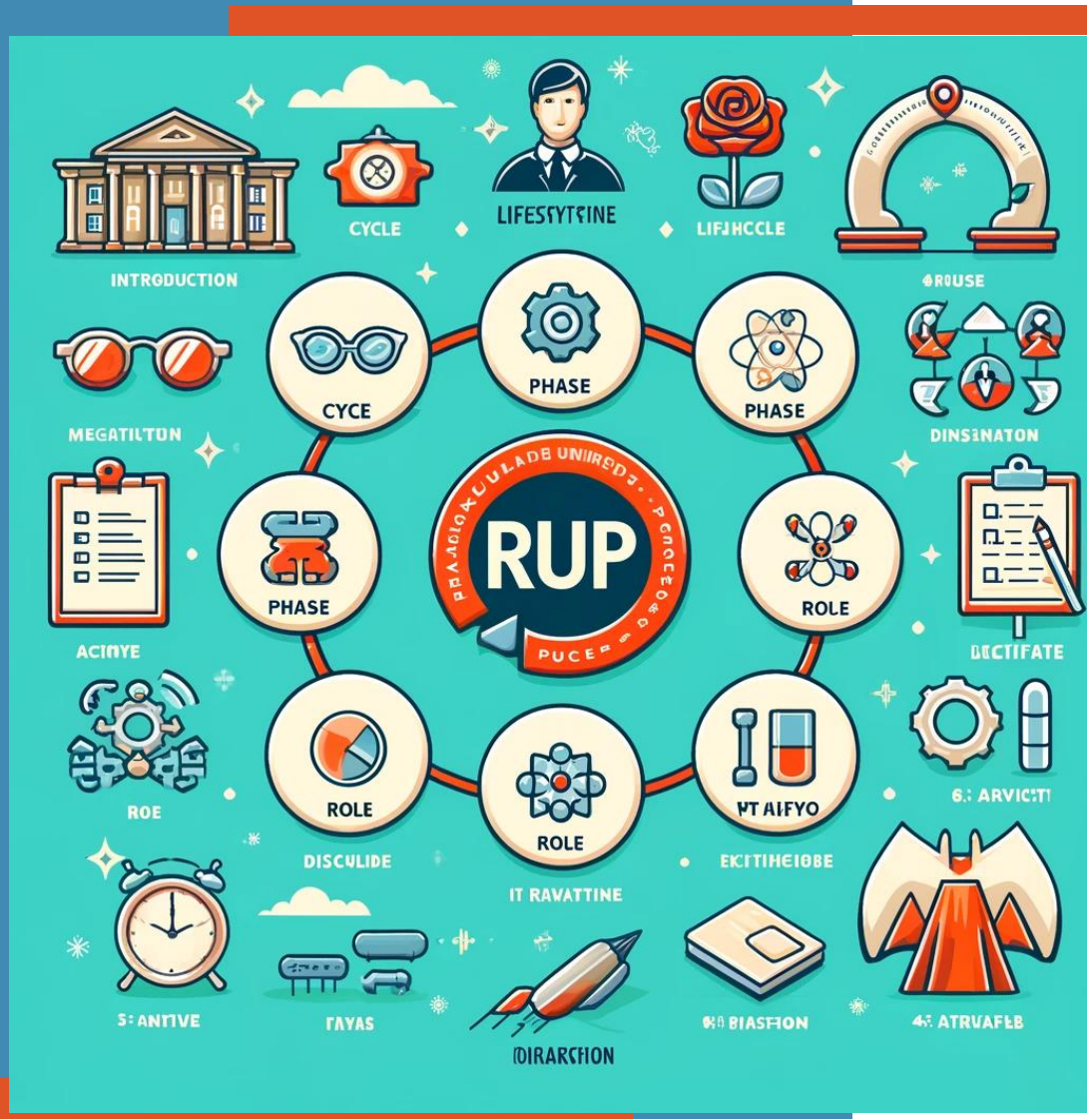
dr inż. Michał Malinowski
michal.malinowski@warszawa.merito.pl

https://www.linkedin.com/in/michał-malinowski-malkon/

Metodyka RUP i elementy projektu systemu informatycznego

Wykład 02

PROJEKTOWANIE SYSTEMÓW INFORMATYCZNYCH



Zagadnienia (2h)

- Klasyczny Model Wodospadowy
- Wprowadzenie do RUP
  - Cykl (Lifecycle)
  - Faza (Phase)
  - Iteracja (Iteration)
  - Dyscyplina (Discipline)
  - Rola (Role)
  - Artefakt (Artifact)
  - Aktywność (Activity)
  - Przepływ pracy (Workflow)
- Zalety RUP
- Przykład zastosowania RUP

Musimy myśleć o programowaniu jak o fundamentalnej umiejętności w XXI w. Porównałbym to do czytania, pisania, liczenia.

Września 2023

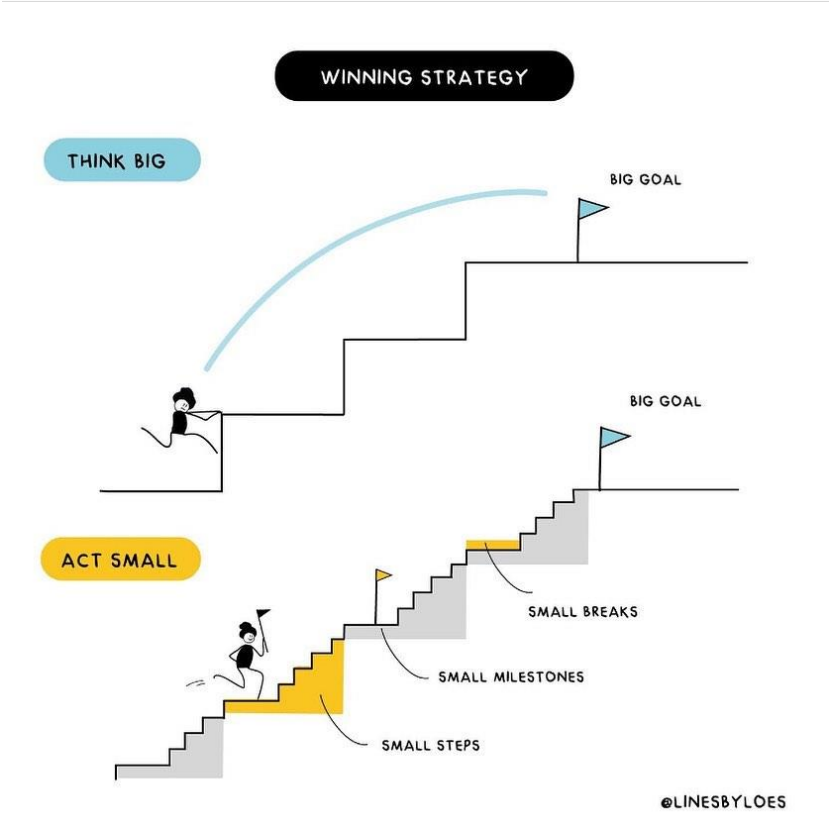
Miron Mironiuk, założyciel Cosmose.AI,
pomysłodawca programu "Programowanie = nasz drugi język"

Klasyczny Model Wodospadowy

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

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

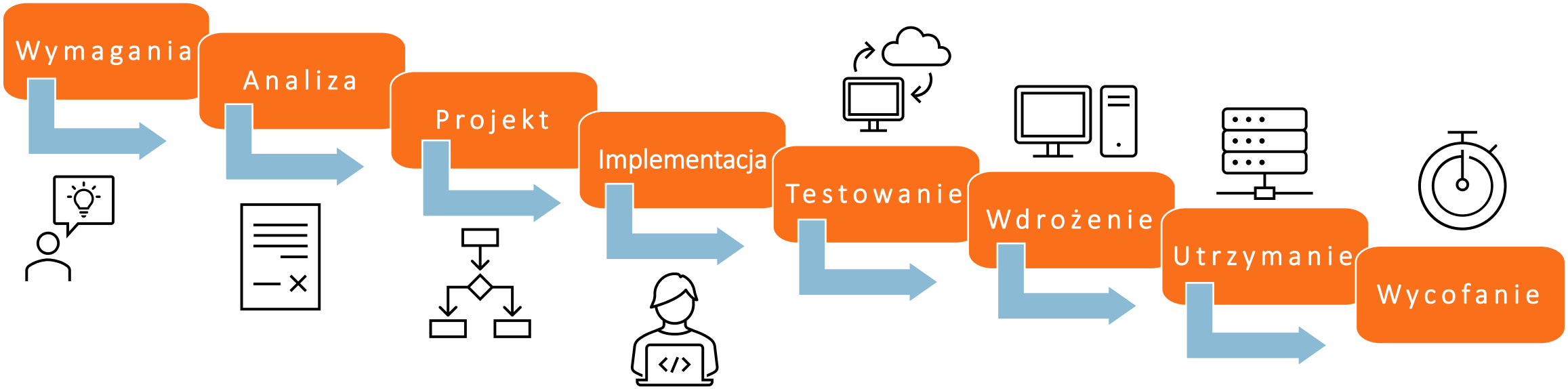
Z tego względu proces wytwarzania oprogramowania dzieli się na etapy.



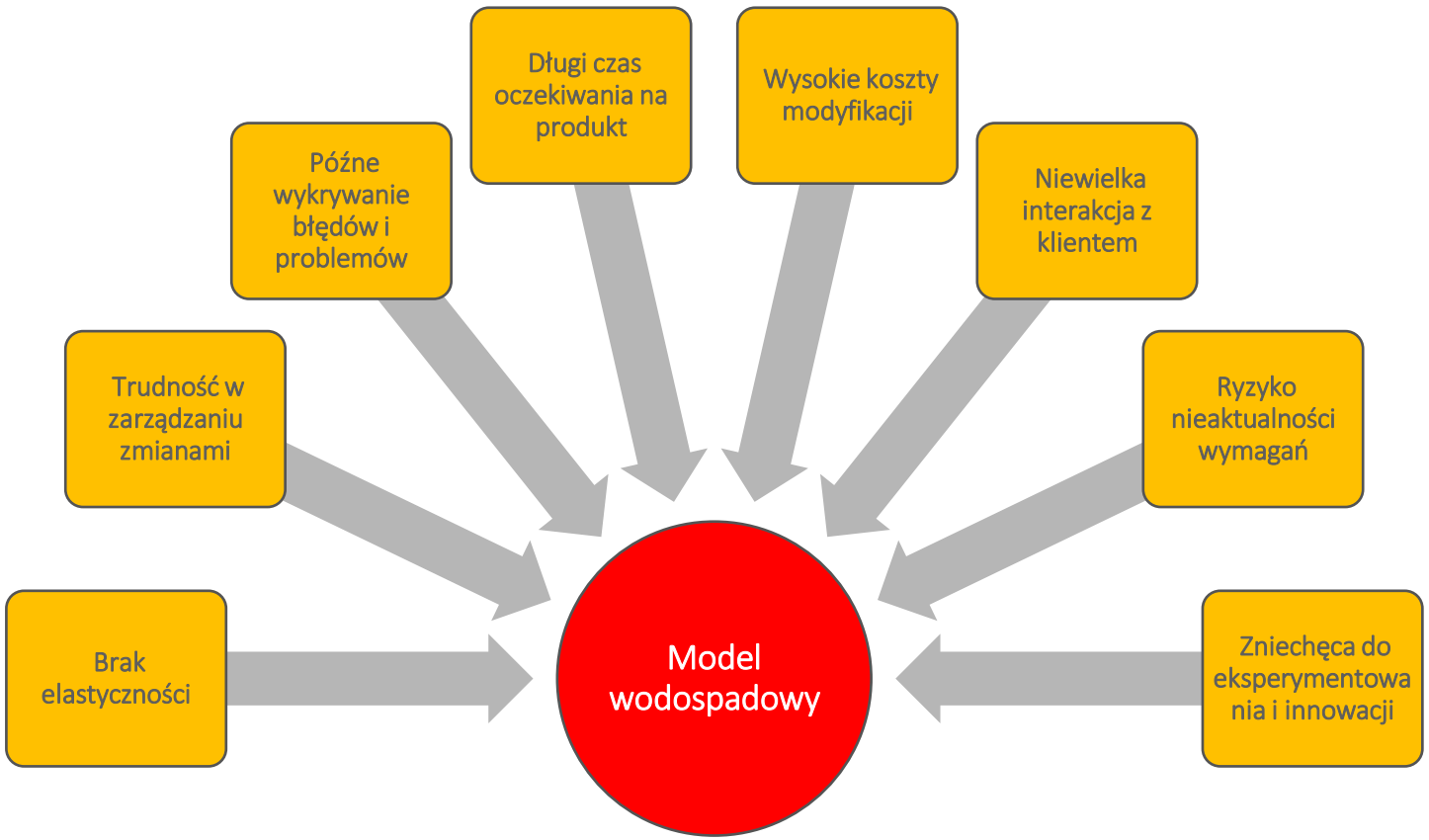
Klasyczny Model Wodospadowy

Model wodospadowy (kaskadowy)

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



Klasyczny Model Wodospadowy



Wprowadzenie do RUP

RUP (Rational Unified Process)

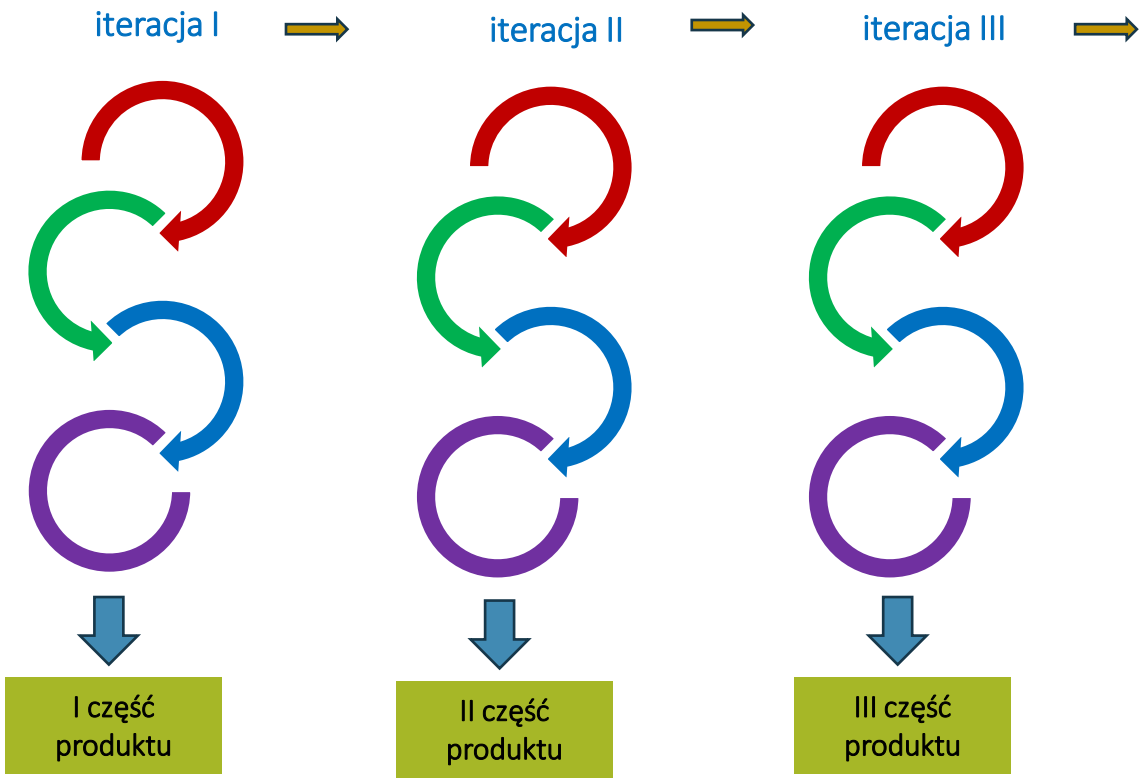
jest iteracyjną i przyrostową metodyką wytwarzania oprogramowania skoncentrowana na architekturze i ryzyku.

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 zrozumiał wymagania i oczekiwania klienta.

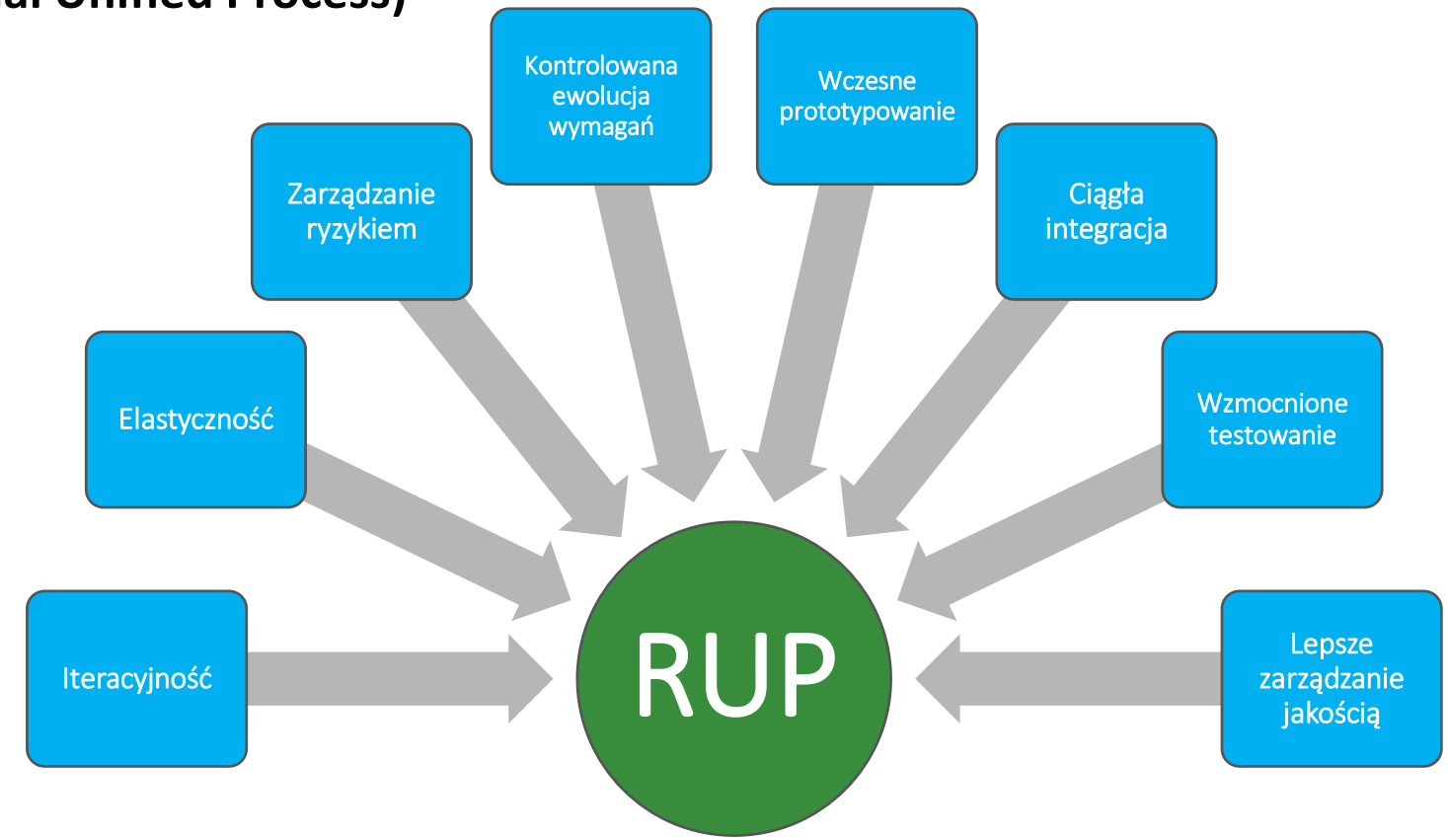
Szybkie wykrycie ewentualnych problemów lub nieporozumień, pozwala szybko wprowadzić odpowiednie modyfikacje.

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



Wprowadzenie do RUP

RUP (Rational Unified Process)



Wprowadzenie do RUP

Historia

Lata 80-te

Początki koncepcji iteracyjnego rozwoju oprogramowania, skontrastowane z modelem kaskadowym, który był dominujący w tamtym okresie.

Lata 90-te

Sformalizowanie RUP przez firmę Rational Software. Była to odpowiedź na potrzebę bardziej elastycznego i adaptacyjnego procesu tworzenia oprogramowania, który byłby w stanie lepiej radzić sobie ze zmianami w wymaganiach biznesowych i technologicznych.

2003 rok

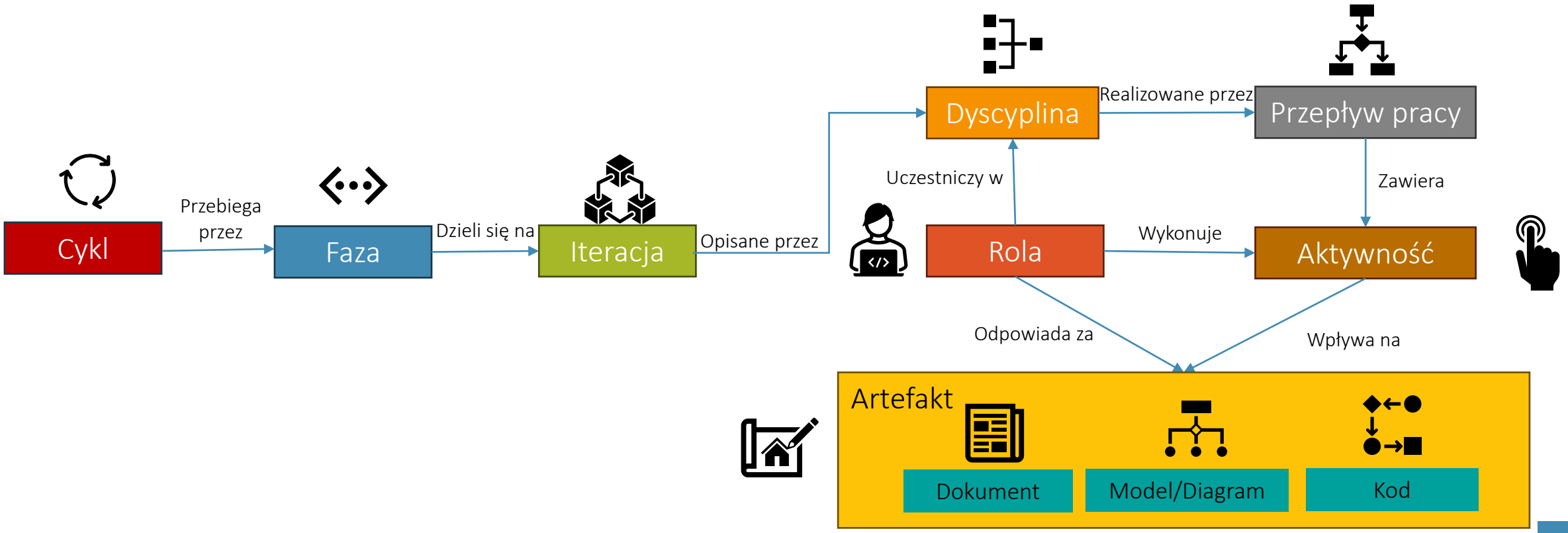
Przejęcie Rational Software przez IBM, co spowodowało dalszą integrację RUP z ofertą IBM i rozwój narzędzi wspierających metodykę, takich jak IBM Rational Rose i IBM Rational Software Architect.

Teraz

Współczesne zastosowania i adaptacje RUP, w tym jego użycie w metodykach zwinnych oraz jego rola w edukacji inżynierii oprogramowania.

Wprowadzenie do RUP

Architektura metodyki RUP



Cykl (Lifecycle)

Cykl (Lifecycle)

to całościowy okres czasu od rozpoczęcia projektu do jego zakończenia, zawierający w sobie sekwencję faz.

Cykl odzwierciedla kompletną ewolucję systemu od koncepcji, poprzez rozwój, aż po wdrożenie finalnego produktu.

Cykl kończy się dostarczeniem produktu (wynik projektu).

Cykl Poczatkpwy (Inception Cycle)

Ma na celu zdefiniowanie ogólnego zakresu projektu, głównych funkcji systemu, kluczowych wymagań oraz ocenę ryzyka, co pozwala na wstępną ocenę opłacalności i wykonalności projektu.

Cykl Rozwojowy (Development Cycle):

Prowadzi do stworzenia nowej wersji systemu z dodatkową funkcjonalnością lub znaczącymi ulepszeniami.

Cykl Utrzymaniowy (Maintenance Cycle):

Skupia się na bieżącym wsparciu, naprawie błędów, ulepszeniach wydajności i aktualizacji systemu.

Cykl Ewolucyjny (Evolutionary Cycle):

Przeznaczony do stopniowego rozwijania systemu przez dodawanie nowych funkcji i adaptację do zmieniających się wymagań rynku lub technologii.

Cykl Wydania (Release Cycle):

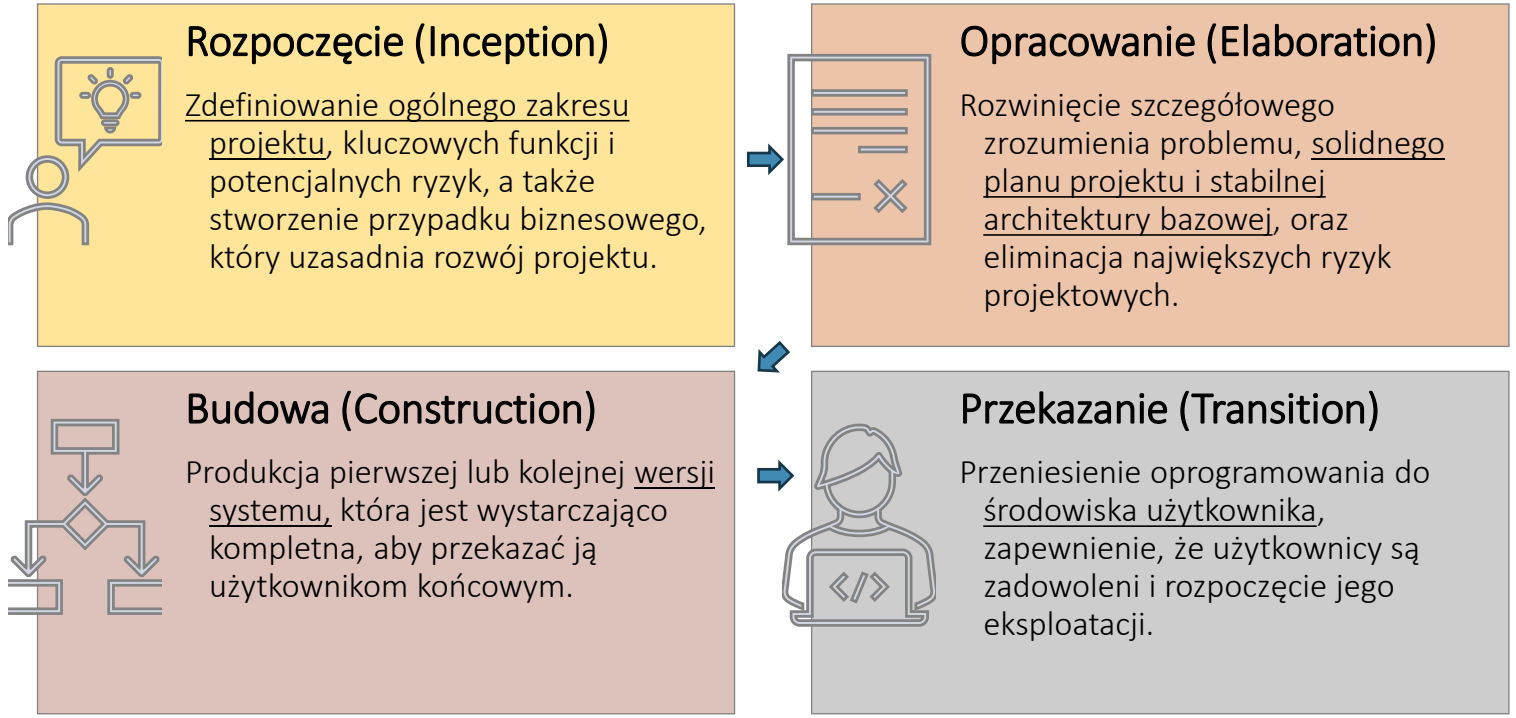
Obejmuje wydanie produktu, które dostarcza wartość dla użytkowników końcowych i jest gotowe do użytku produkcyjnego.

Faza (Phase)

Faza (Phase)

to wyraźnie określony okres w cyklu życia projektu, który ma swoje cele, zadania do wykonania oraz wyniki i rezultaty, które muszą zostać osiągnięte (kamienie milowe) przed przejściem do następnej fazy.

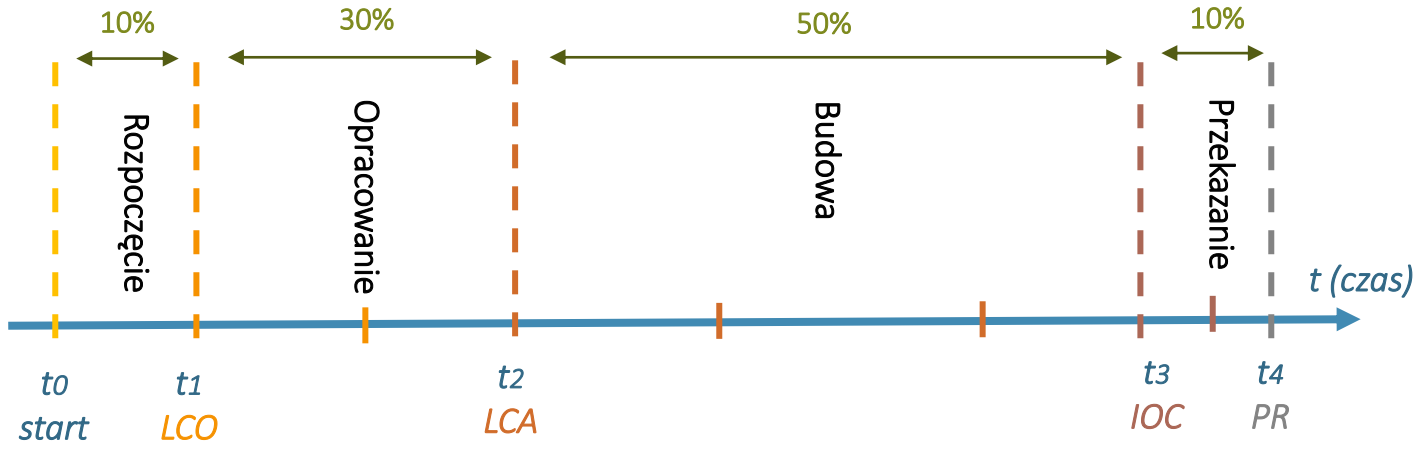
Faza dzieli się na iteracje.



Faza (Phase)

Kamień milowy (milestone)

to punkt kontrolny w projekcie, który oznacza osiągnięcie kluczowego etapu w procesie rozwoju projektu.



LCO - *Kamień Milowy Cyklu Życia* (Lifecycle Objective Milestone)

Potwierdzenie, że projekt jest opłacalny, cele są jasne, ryzyka zidentyfikowane, a wszyscy główne interesariusze zgadzają się z planowanym zakresem i harmonogramem.

LCA -*Kamień Milowy Architektury Cyklu Życia* (Lifecycle Architecture Milestone)

Zatwierdzenie stabilnej architektury systemu, która może sprostać wymaganiom i zmniejszenie kluczowych ryzyk do akceptowalnego poziomu.

IOC - *Kamień Milowy Wstępnej Operacyjności* (Initial Operational Capability Milestone)

Demonstracja, że system jest funkcjonalny i gotowy do pierwszych testów operacyjnych z ograniczoną grupą użytkowników końcowych.

PR - *Kamień Milowy Wydania Produktu* (Product Release Milestone)

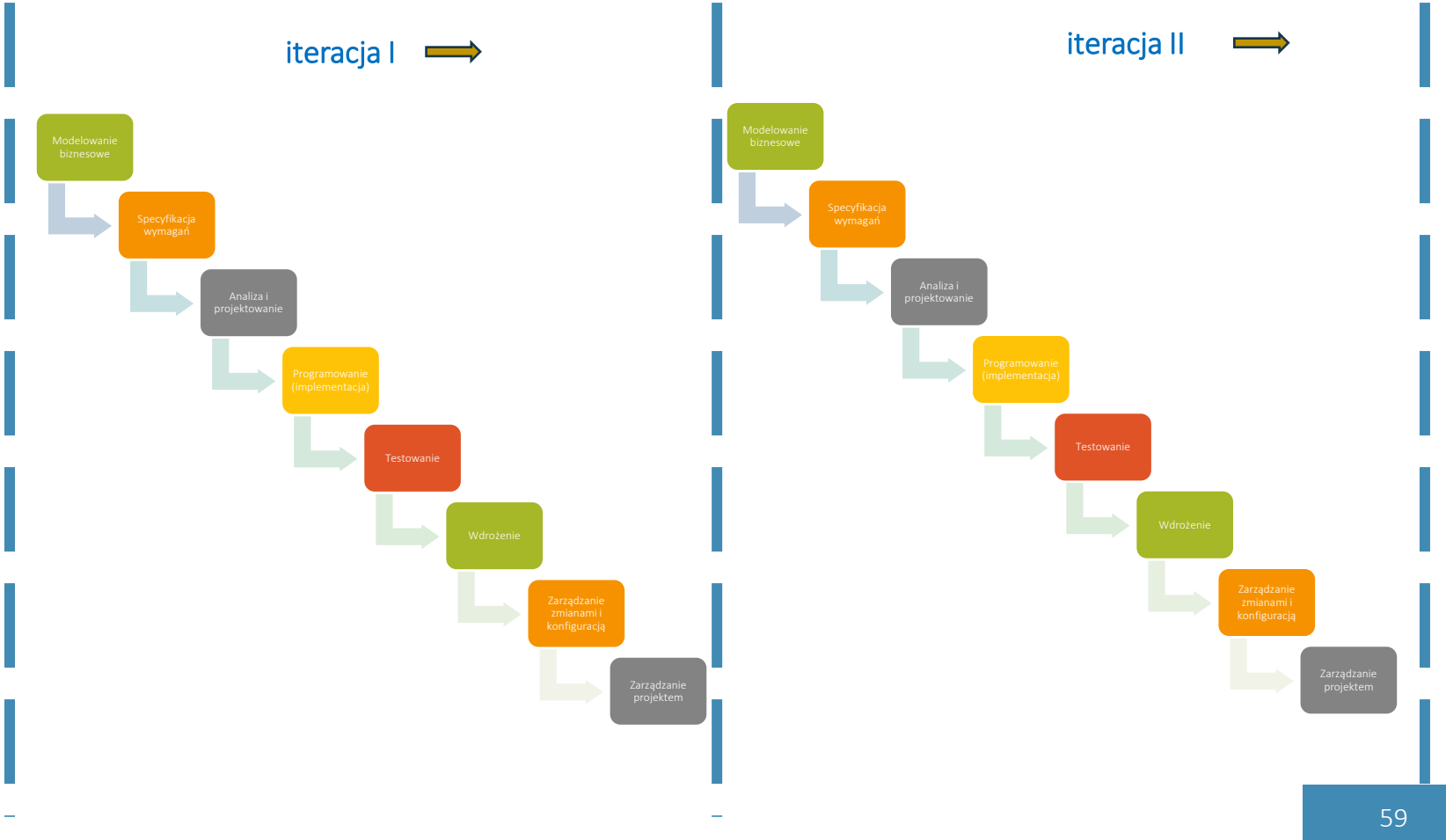
Potwierdzenie, że system jest gotowy do pełnego wdrożenia i użytkowania, spełnia wszystkie wymagania i jest dostępny dla wszystkich docelowych użytkowników.

Iteracja (Iteration)

Iteracja (Iteration)

to powtarzalny cykl prac, który koncentruje się na progresywnym rozwijaniu i ulepszaniu systemu poprzez serię wydzielonych kroków.

Każda iteracja obejmuje wszystkie dyscypliny projektowe.



Iteracja (Iteration)

Faza	Iteracja 1	Iteracja 2 i kolejne
Rozpoczęcie (Inception)	Wstępna identyfikacja kluczowych interesariuszy, głównych wymagań użytkownika i ryzyk projektu.	-
Opracowanie (Elaboration)	Definiowanie architektury systemu, precyzyjne określenie wymagań, rozpoczęcie redukcji największych ryzyk projektowych.	Rozszerzenie architektury i adresowanie kluczowych ryzyk.
Budowa (Construction)	Implementacja i testowanie podstawowych funkcji systemu.	Rozwijanie systemu o dodatkowe funkcjonalności, intensywne testowanie i optymalizacja wydajności.
Przekazanie (Transition)	Wdrożenie, szkolenia użytkowników, zbieranie informacji zwrotnych.	Rozwiązywanie problemów wykrytych po wdrożeniu, dostosowanie systemu do warunków produkcyjnych.

Iteracja (Iteration)

Przykładowy plan projektu

Fazy / iteracje/ kamienie milowe	Dni
Plan Projektu	125
Rozpoczęcie	
Iteracja Rozpoczęcia I1 - Wstępna iteracja	15
Kamień Milowy Cyklu Życia	0
Opracowanie	
Iteracja Opracowania E1 - Architektoniczny Prototyp dla Aplikacji CICS	15
Iteracja Opracowania E2 - Architektoniczny Prototyp dla Usług Sieciowych i Łączności	15
Iteracja Opracowania E3 - Architektoniczny Prototyp dla Dostępu do Katalogów Usług Sieciowych	15
Kamień Milowy Architektury Cyklu Życia	0
Budowa	
Iteracja Budowy C1 - Rozwój Możliwości Zamawiania	14
Iteracja Budowy C2 - Rozwój Możliwości Uzupełnienia Zapasów i Wydanie Wersji Beta Systemu	15
Kamień Milowy Wstępnej Operacyjności	0
Przekazanie	
Iteracja Przekazania T1 – Wdrożenie Wersji Beta	30
Iteracja Przekazania T2 – Monitorowanie wydajności wersji Bata	15
Kamień Milowy Wydania Produktu	0

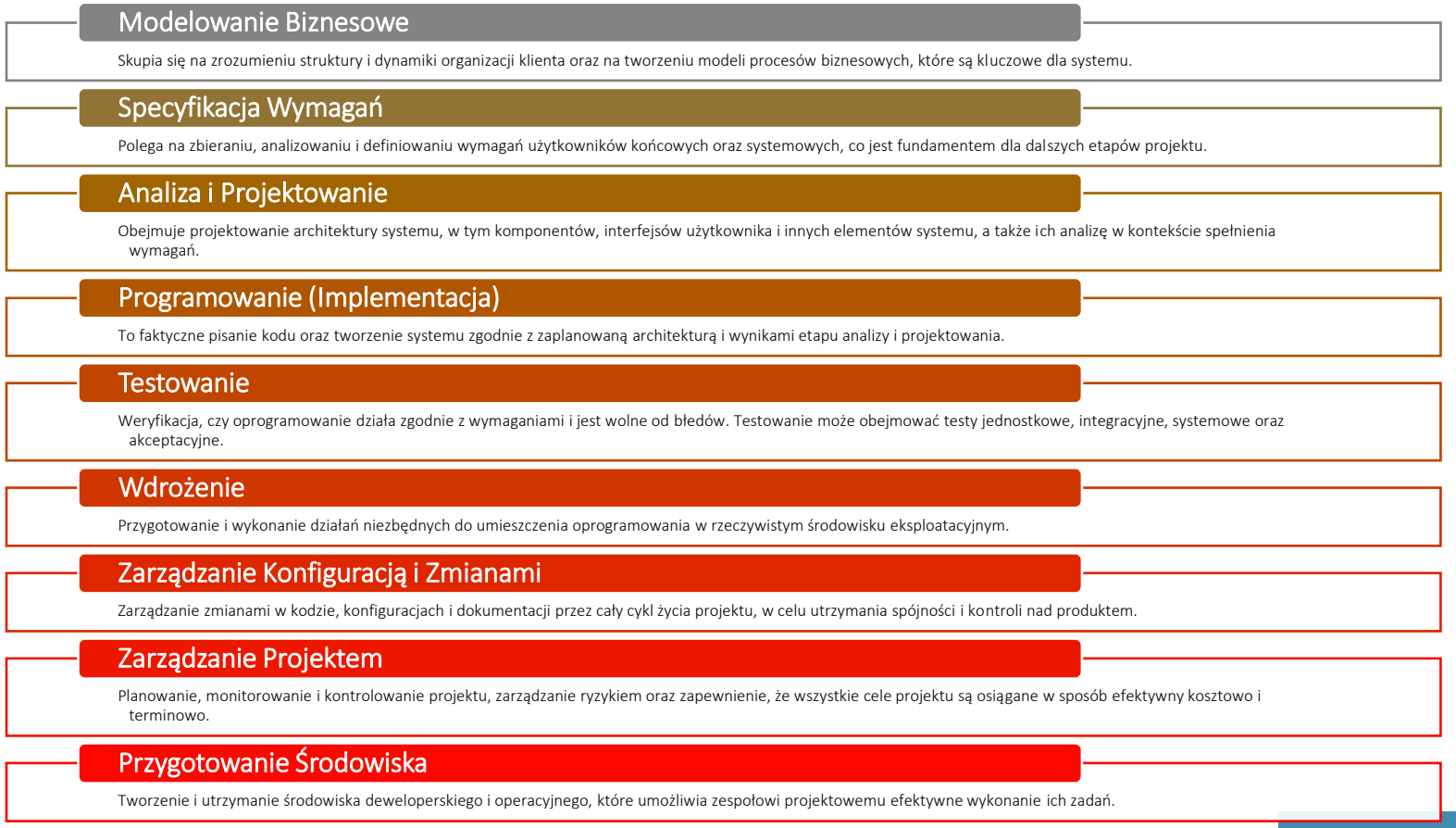
Dyscyplina (Discipline)

Dyscyplina (Discipline)

to specjalizowany obszary działań (zadań, technik i praktyk) skoncentrowany na określonej dziedzinie, który jest wykonywany przez cały czas trwania projektu.

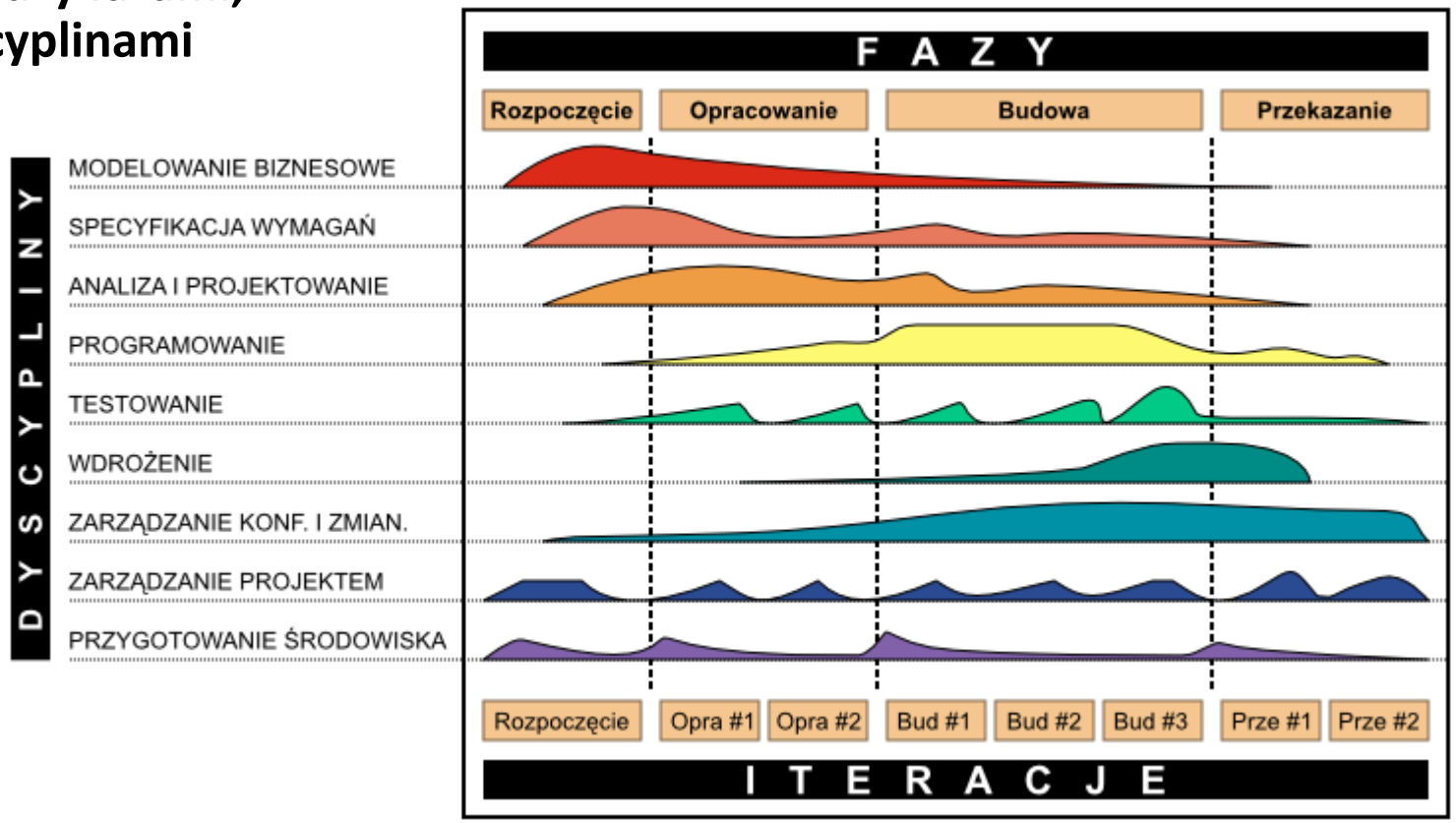
Dyscypliny dzielimy na:

- inżynieryjne (Modelowanie Biznesowe, Specyfikacja Wymagań, Analiza i Projektowanie, Programowanie (Implementacja), Testowanie, Wdrożenie)
- wspomagające (Zarządzanie Konfiguracją i Zmianami, Zarządzanie Projektem, Przygotowanie Środowiska



Dyscyplina (Discipline)

Zależność pomiędzy fazami, iteracjami a dyscyplinami



Rola (Role)

Zależność pomiędzy rolami, artefaktami, aktywnościami a przepływami pracy

Rola (Role)

Rola (Role)

definiuje zestaw obowiązków i odpowiedzialności przypisanych do członka zespołu projektowego, określając, jakie aktywności ma on wykonać i jakie artefakty dostarczyć.

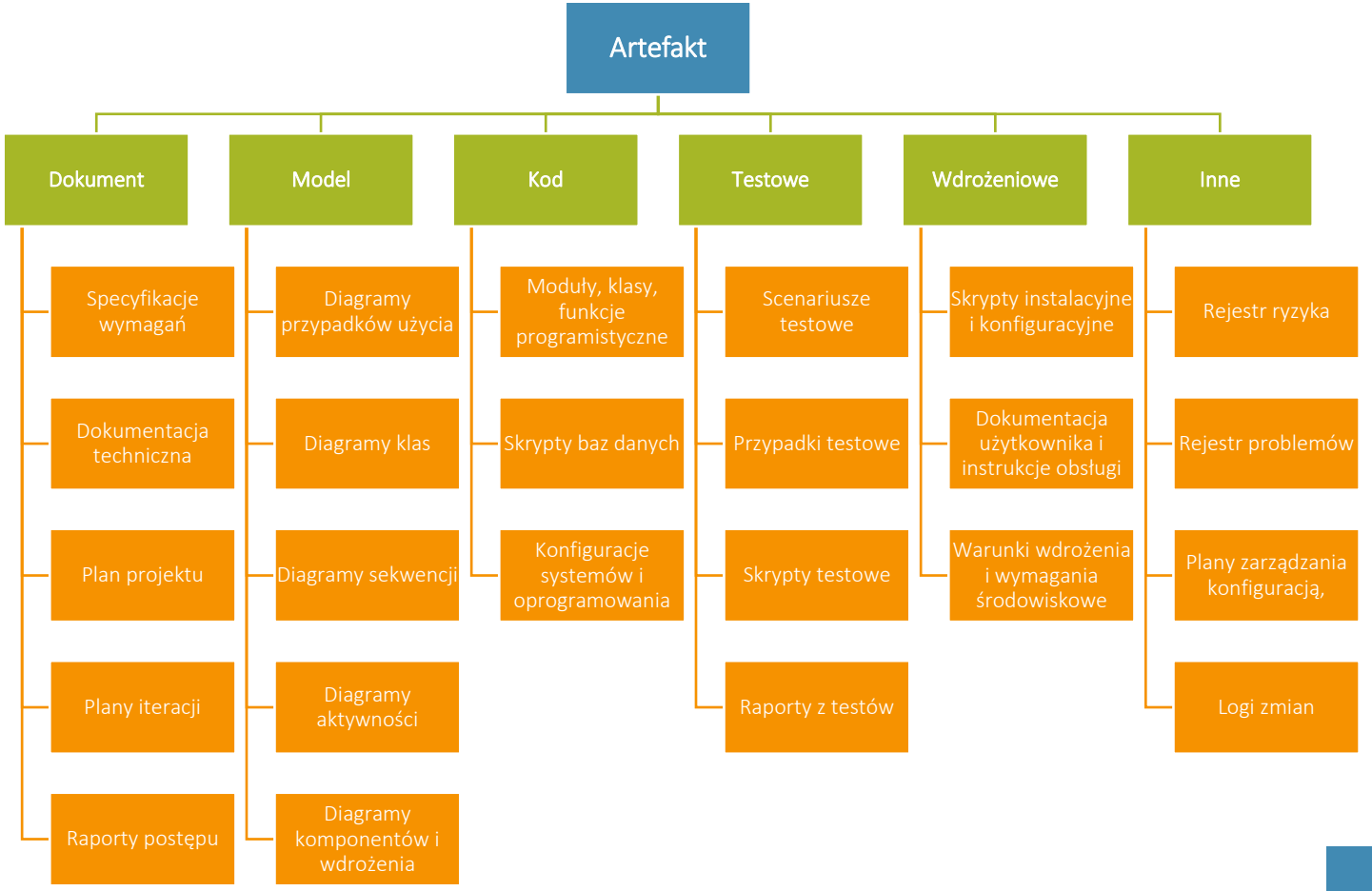
Role są zorganizowane wokół konkretnych dyscyplin projektowych i adaptowane do faz projektu.

Rola	Rozpoczęcie (Inception)	Opracowanie (Elaboration)	Budowa (Construction)	Przekazanie (Transition)
Analityk Biznesowy	■	■		
Architekt Systemu	■	■	■	
Projektant Oprogramowania		■	■	
Programista		■	■	■
Tester		■	■	■
Kierownik Projektu	■	■	■	■
Kierownik ds. Konfiguracji	■	■	■	■
Inżynier ds. Wymagań	■	■		
Użytkownik Końcowy				■
Specjalista ds. Wdrożenia				■

Artefakt (Artifact)

Artefakt (Artifact)

To konkretny produkt aktywności, który jest generowany podczas realizacji projektu przez rolę.



Artefakt (Artifact)

Artefakty będące modelami/diagramami

Faza	Artefakt	Diagram/Model	Opis
Rozpoczęcie (Inception)	Analiza możliwości	Diagram przypadków użycia	Identyfikuje funkcjonalności systemu z perspektywy użytkownika.
	Wymagania funkcjonalne	Diagram dekompozycji funkcji	Prezentuje hierarchię funkcjonalności systemu.
	Plan projektu	Diagram Gantta	Wizualizuje harmonogram projektu i przedstawia zależności między zadaniami.
	Zarządzanie ryzykiem	Mind Maps	Identyfikuje potencjalne ryzyka i ich źródła.
	Model biznesowy	Model procesu biznesowego BPMN	Prezentuje procesy biznesowe i modele przepływu danych.
	Specyfikacje wymagań	Diagram dekompozycji funkcji	Prezentuje hierarchię funkcjonalności systemu w kontekście wymagań.
Opracowanie (Elaboration)	Model architektury	Diagram komponentów	Prezentuje strukturę systemu z punktu widzenia komponentów.
		SysML	Modeluje zachowania i interakcje systemów inżynieryjnych.
	Model bazy danych	Diagram ERD	Prezentuje strukturę i relacje w bazie danych.
	Projekty Interfejsu Użytkownika	Diagram aktywności/czynności	Prezentuje przepływ interakcji użytkownika z interfejsem.
	Plan testów	Flowcharts	Prezentuje proces testowania i określa kroki testów.
Budowa (Construction)	Dokumentacja techniczna	Diagram klas	Prezentuje strukturę systemu z punktu widzenia klas.
		Diagram sekwencji	Prezentuje interakcje między obiektami systemu.
	Testy jednostkowe	Sequence Charts	Weryfikuje poprawność sekwencji działań w testach jednostkowych.
Przekazanie (Transition)	Wdrożenie	Diagram wdrożenia	Prezentuje infrastrukturę systemową i sposób wdrożenia systemu.
	Instrukcja użytkownika	Flowcharts	Prezentuje instrukcje krok po kroku obsługi systemu.
	Raport z projektu	Diagram czasowy	Prezentuje harmonogram projektu i przedstawia postępy w jego realizacji.

Aktywność (Activity)

Aktywność (Activity)

to podstawowa jednostka pracy, która ma być wykonana. Jest to zadanie lub zestaw zadań prowadzonych przez określone role w projekcie, mających na celu wytworzenie lub zaktualizowanie artefaktów.

Aktywność	Rola
Definiowanie wymagań systemowych	Inżynier ds. Wymagań
Projektowanie architektury systemu	Architekt Systemu
Implementacja komponentów oprogramowania	Programista
Przeprowadzanie testów jednostkowych	Tester
Tworzenie i utrzymanie modelu przypadków użycia	Analityk Biznesowy, Inżynier ds. Wymagań
Zarządzanie projektem	Kierownik Projektu
Zarządzanie konfiguracją i zmianami	Kierownik ds. Konfiguracji
Analiza i projektowanie	Projektant Oprogramowania, Architekt Systemu
Przeprowadzanie testów akceptacyjnych	Tester, Użytkownik Końcowy
Przygotowywanie środowiska deweloperskiego	Kierownik ds. Konfiguracji
Wdrożenie systemu	Specjalista ds. Wdrożenia
Monitorowanie i kontrola postępu projektu	Kierownik Projektu
Przeprowadzanie przeglądu kodu	Programista, Architekt Systemu
Przygotowywanie dokumentacji użytkownika i systemowej	Inżynier ds. Wymagań, Analityk Biznesowy
Zarządzanie ryzykiem projektowym	Kierownik Projektu, Architekt Systemu

Przepływ pracy (Workflow)

Przepływ pracy (Workflow)

to sekwencja aktywności, które mają być wykonane w określonym porządku w celu osiągnięcia konkretnego rezultatu (powstania artefaktu).

Reprezentuje proces, przez który artefakty są rozwijane, przeglądane i doskonalone przez różne role w projekcie.

Przepływy pracy w RUP pomagają w koordynacji, strukturyzacji i śledzeniu postępów prac w ramach projektu informatycznego.

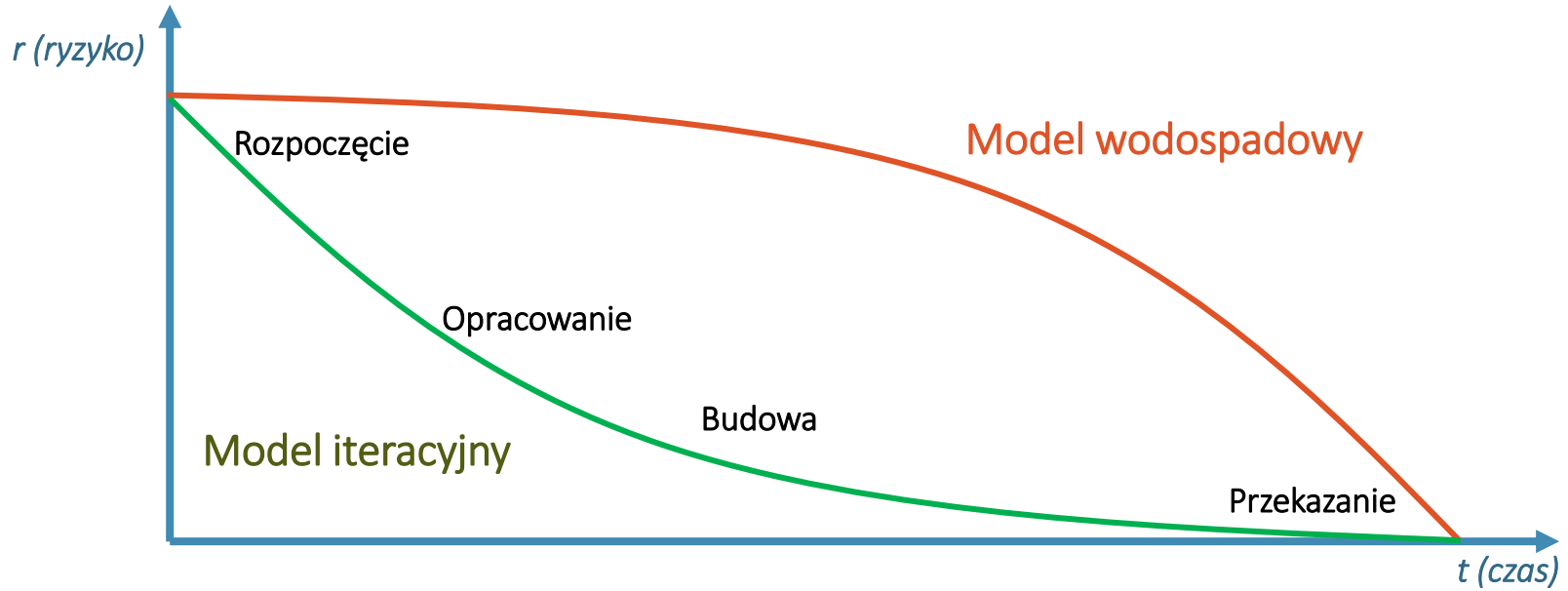
Przepływ pracy (Workflow)

Przykład - aktywności w przepływie pracy

Faza	Opracowania	
Dyscyplina	Analiza i projektowanie	
Przepływ pracy	Analiza wymagań	
Aktywność	Rola	Artefakt
Definiowanie architektury systemu	Architekt systemu	Model architektury
Modelowanie przypadków użycia	Analityk biznesowy	Diagramy przypadków użycia
Projektowanie komponentów	Projektant	Specyfikacje komponentów
Modelowanie danych	Specjalista od baz danych	Schematy baz danych
Projektowanie interfejsów użytkownika	Projektant interfejsu użytkownika	Projekty interfejsu użytkownika
Integracja systemu	Architekt systemu / Projektant	Plan integracji systemu
Ocena wydajności	Architekt systemu	Analiza wydajności

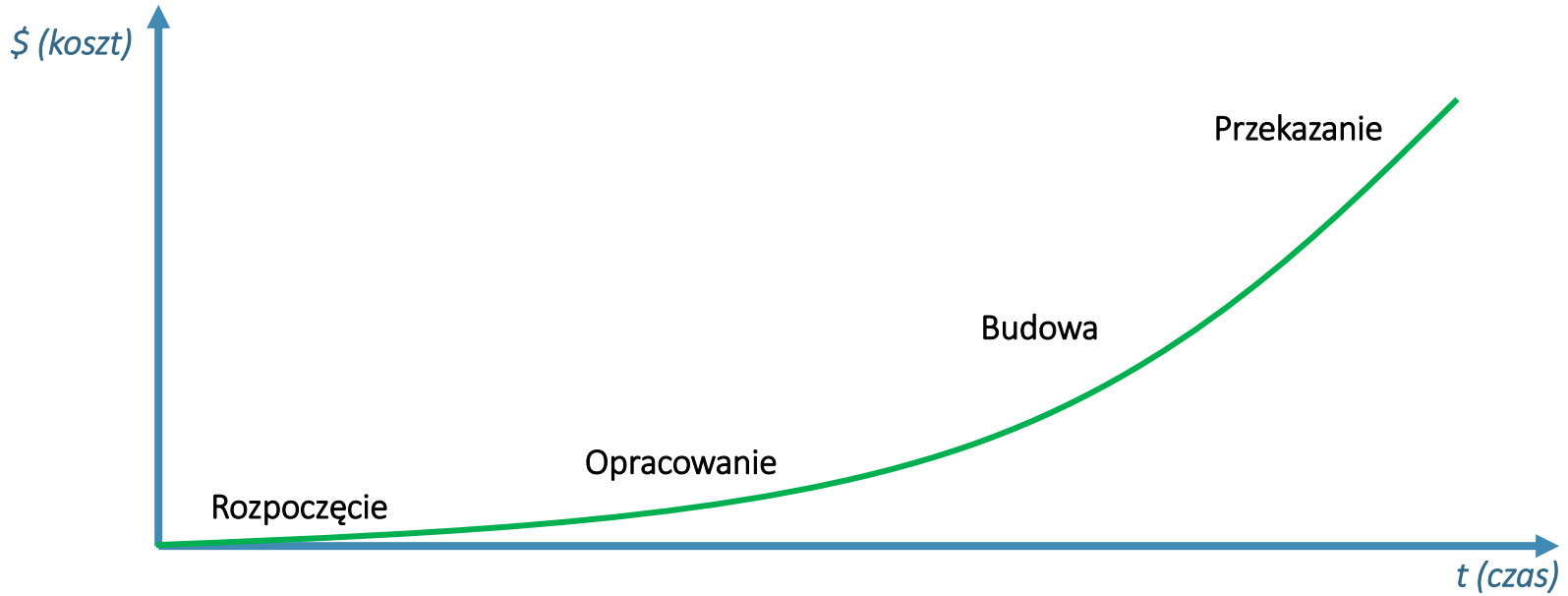
Zalety RUP

Ryzyko niepowodzenia



Zalety RUP

Koszty związane z poprawą błędów



Przykład zastosowania RUP

1. Faza Rozpoczęcia (Inception)

Identyfikacja głównych cech aplikacji i zrozumienie wymagań użytkowników.

Iteracja: Wykonywanie wstępnych iteracji dla określenia zakresu i celów aplikacji.

Dyscyplina: Specyfikacja wymagań i modelowanie biznesowe.

Rola: Analityk Biznesowy i Analityk Systemowy - osoby te definiują zakres projektu i gromadzą wymagania.

Aktywność: Tworzenie Wizji Projektu i Inicjalna Specyfikacja Wymagań.

Artefakt: Dokument Wizji oraz Wstępne Diagramy Przypadków Użycia (Use Case Diagrams), które są początkowym modelem systemu.

Przykład zastosowania RUP

2. Faza Opracowania (Elaboration)

Szczegółowe modelowanie i weryfikacja architektury aplikacji.

Iteracja: Iteracje koncentrujące się na rozwoju architektury i planowaniu fazy budowy.

Dyscyplina: Analiza i Projektowanie Systemu.

Rola: Architekt Oprogramowania - kluczowa osoba w procesie tworzenia architektury systemowej.

Aktywność: Modelowanie architektury systemu oraz interfejsów użytkownika.

Artefakt: Dokumentacja Architektury (Model/Diagram) oraz Prototypy UI/UX (Model/Diagram), które są szczegółowymi projektami architektury i użytkowania.

Przykład zastosowania RUP

3. Faza Budowy (Construction)

Realizacja zaprojektowanej funkcjonalności i przygotowanie produktu do wydania.

Iteracja: Cykliczne iteracje skupione na rozwoju i testowaniu funkcjonalności.

Dyscyplina: Implementacja i Testowanie.

Rola: Programista i Inżynier Testów - tworzą i weryfikują działanie kodu.

Aktywność: Kodowanie, testowanie jednostkowe i integracyjne.

Artefakt: Kod Źródłowy (Kod) i Plan Testów (Dokument).

Przykład zastosowania RUP

4. Faza Przekazania (Transition)

Działania zmierzające do umożliwienia użytkownikom skorzystania z produktu.

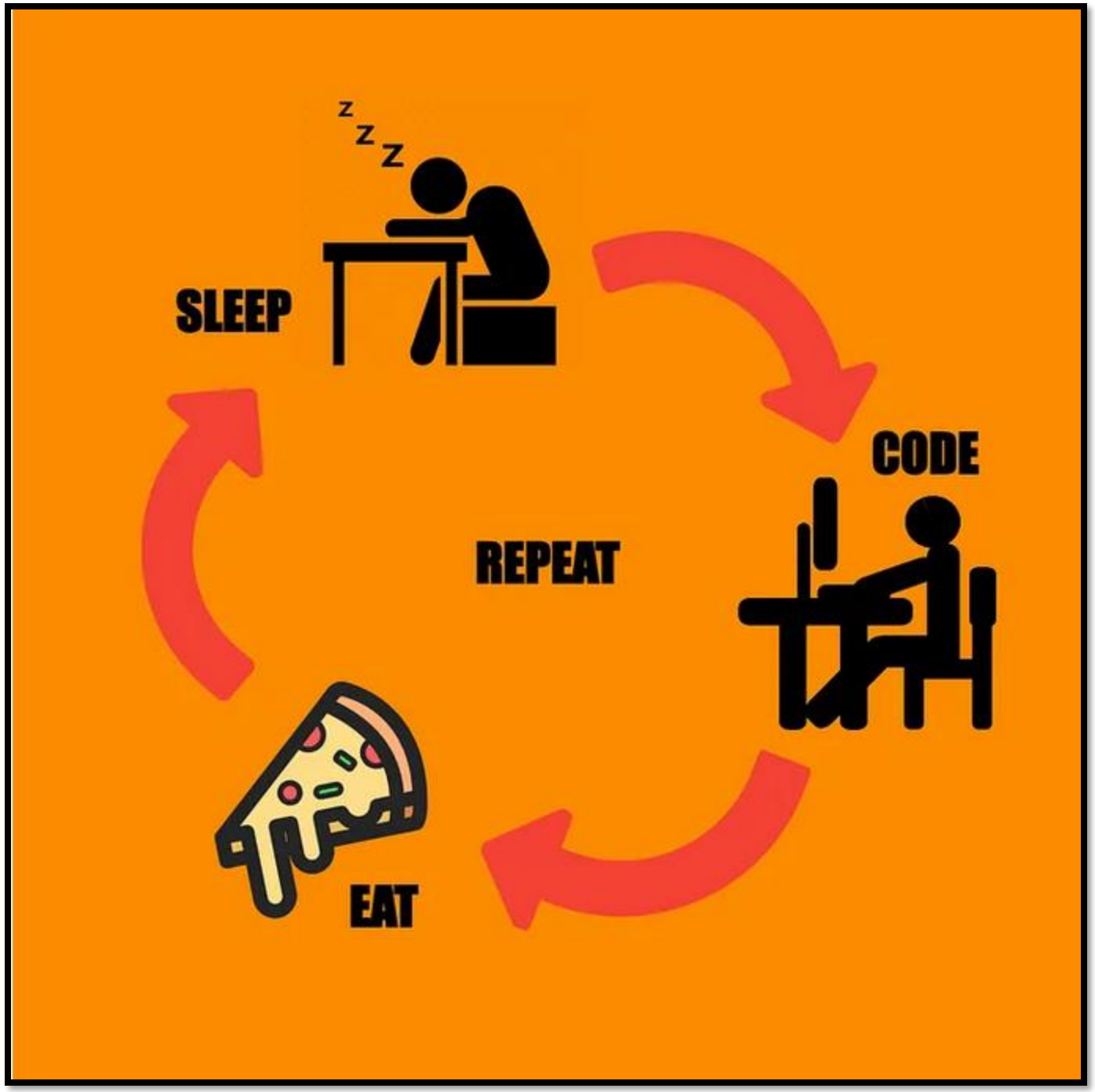
Iteracja: Iteracje związane z wdrożeniem i optymalizacją produktu po feedbacku od użytkowników.

Dyscyplina: Wdrożenie i Szkolenia.

Rola: Menedżer Wdrożeń i Trener Użytkowników - odpowiedzialni za pomyślne wdrożenie aplikacji i edukację użytkowników.

Aktywność: Przygotowanie środowiska produkcyjnego, szkolenia użytkowników.

Artefakt: Instrukcje Użytkownika (Dokument), Raport Wdrożenia (Dokument).


z z z
SLEEP
CODE
REPEAT
EAT