



Podstawy inżynierii oprogramowania

DR INŻ. KRYSZTOF WOJTKIEWICZ

Agenda

Cykl życia oprogramowania



Modele cyklu życia

Sekwencyjne

- Waterfall

Ewolucyjne

- Model przyrostowy
- Modele zwinne
- Model spiralny

Komponentowy

Cykl życia oprogramowania



Planowanie

Ocena warunków projektu

Obliczanie kosztów

- Praca
- Zasoby

Tworzenie harmonogramu

Przekazanie ewentualnych informacji zwrotnych dla interesariuszy

Planowanie

Zakres

Cel

Kurs

Granice

Definiowanie wymagań



Czasami uznawane na element planowania

Określa co oprogramowanie ma robić

Pozwala skupić się na ważnych elementach oprogramowania

Pozwalają na określenie niezbędnych zasobów

Projektowanie i prototypowanie

Architektura



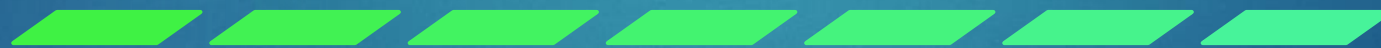
GUI



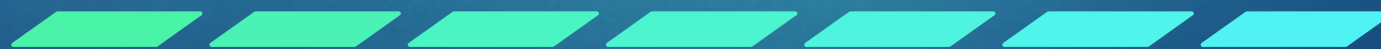
Platforma



Stack technologiczny



Komunikacja



Bezpieczeństwo



Rozwój oprogramowania

Właściwe pisanie oprogramowania

W zależności od projektu realizowane przez zespoły różnej wielkości

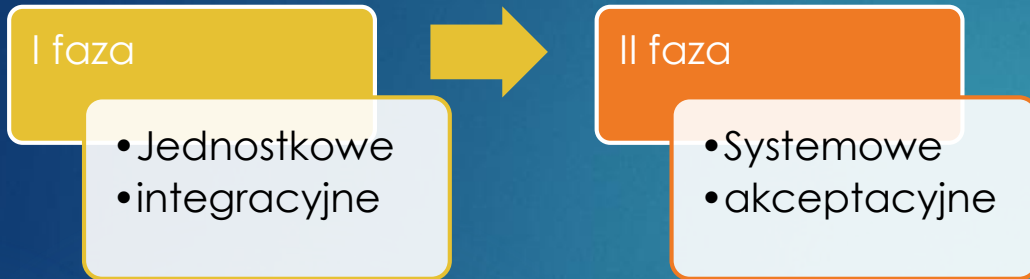
Zalecane użycie oprogramowania wspomagającego

- Kontrola dostępu
- Zarządzanie kodem

Doceniaj instrukcje i wyjaśnienia

- Podręcznik użytkownika
- Komentarze w kodzie

Testowanie



Niefunkcjonalne Funkcjonalne



- Wydajnościowe
- Obciążeniowe
- Użyteczności
- Niezawodności



Wdrożenie

Automatyczne

Dla prostych systemów

Dla podsystemów i modułów

Złożone

Zmiana systemu

Uruchomienie nowej, nieznannej usługi

Działanie i konserwacja

Naprawa błędów

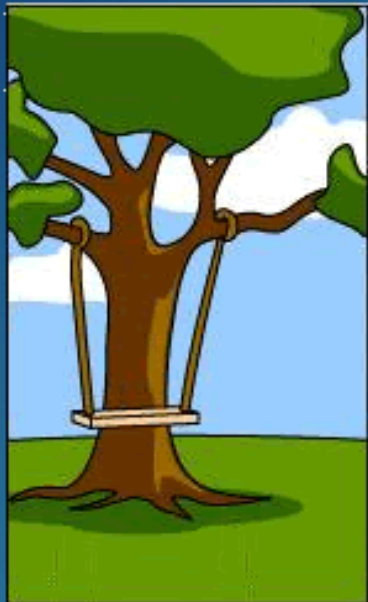
Uruchamianie cyklu od nowa

Uzupełnienie listy funkcjonalności

itp.



How the customer explained it



How the Project Leader understood it



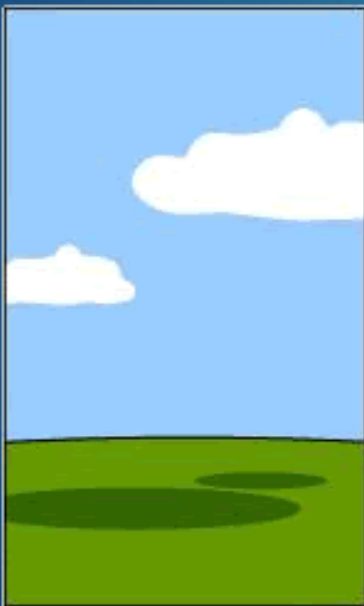
How the Analyst designed it



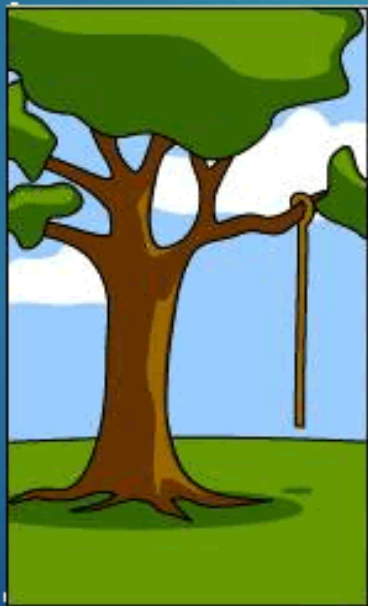
How the Programmer wrote it



How the Business Consultant described it



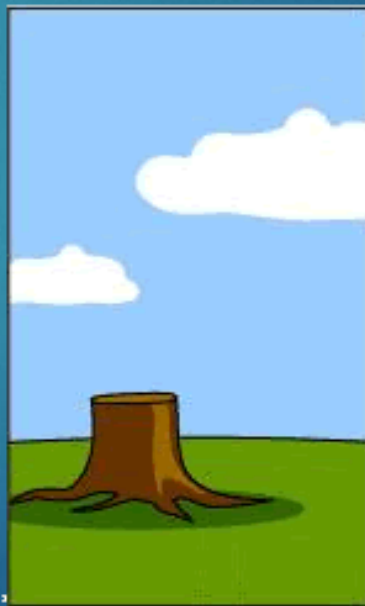
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

Modele cyklu życia

Sekwencyjne

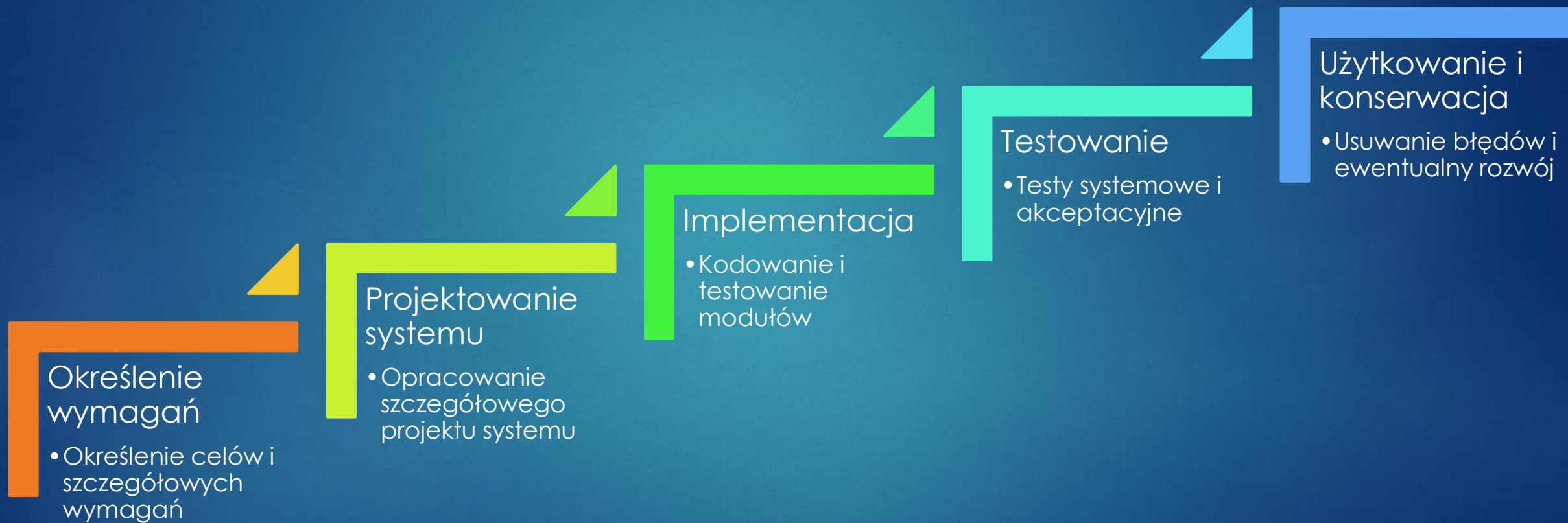
- Waterfall

Ewolucyjne

- Model przyrostowy
- Modele zwinne
- Model spiralny

Komponentowy

Model kaskadowy



Model kaskadowy

Wady

Rygorystyczne określenie następstwa faz

Koszt pomyłek jest wysoki

Mały kontakt z klientem

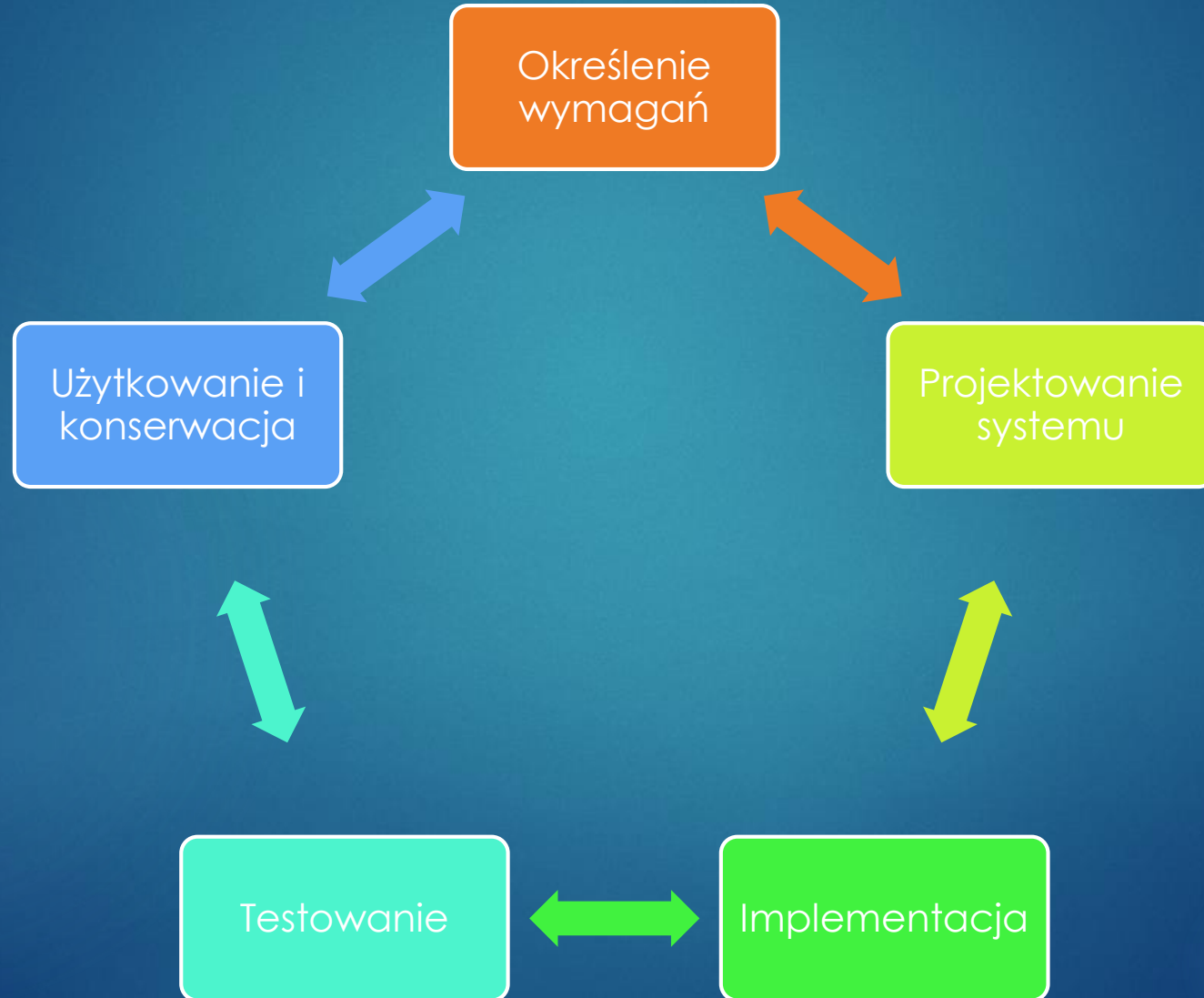
Zalety

Łatwość zarządzania

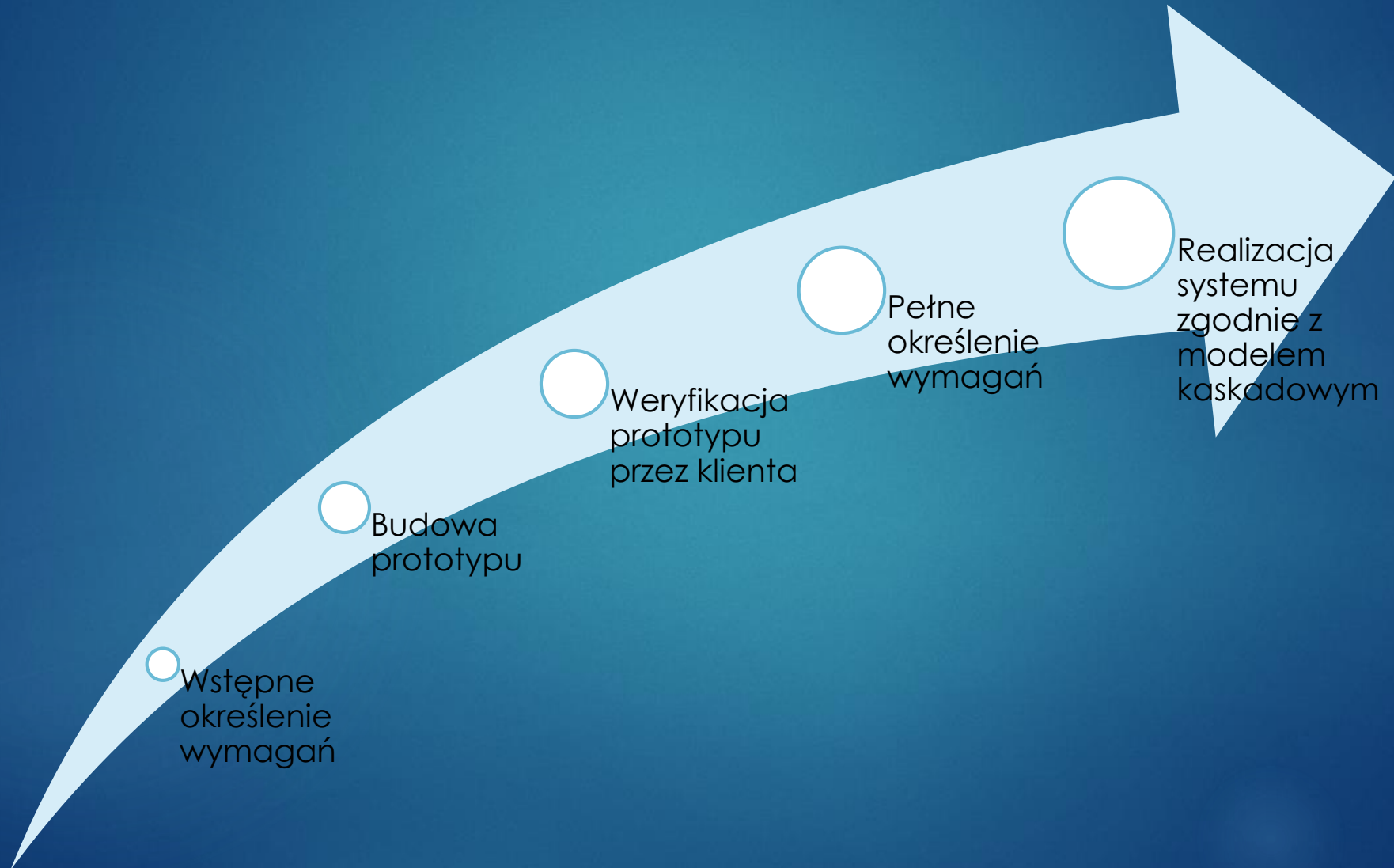
Przewidywalność

Ułatwia planowanie,
harmonogramowanie i
monitorowanie projektu

Model ewolucyjny



Prototypowanie



Prototypowanie

Wady

Dodatkowy koszt
wytworzenia prototypu

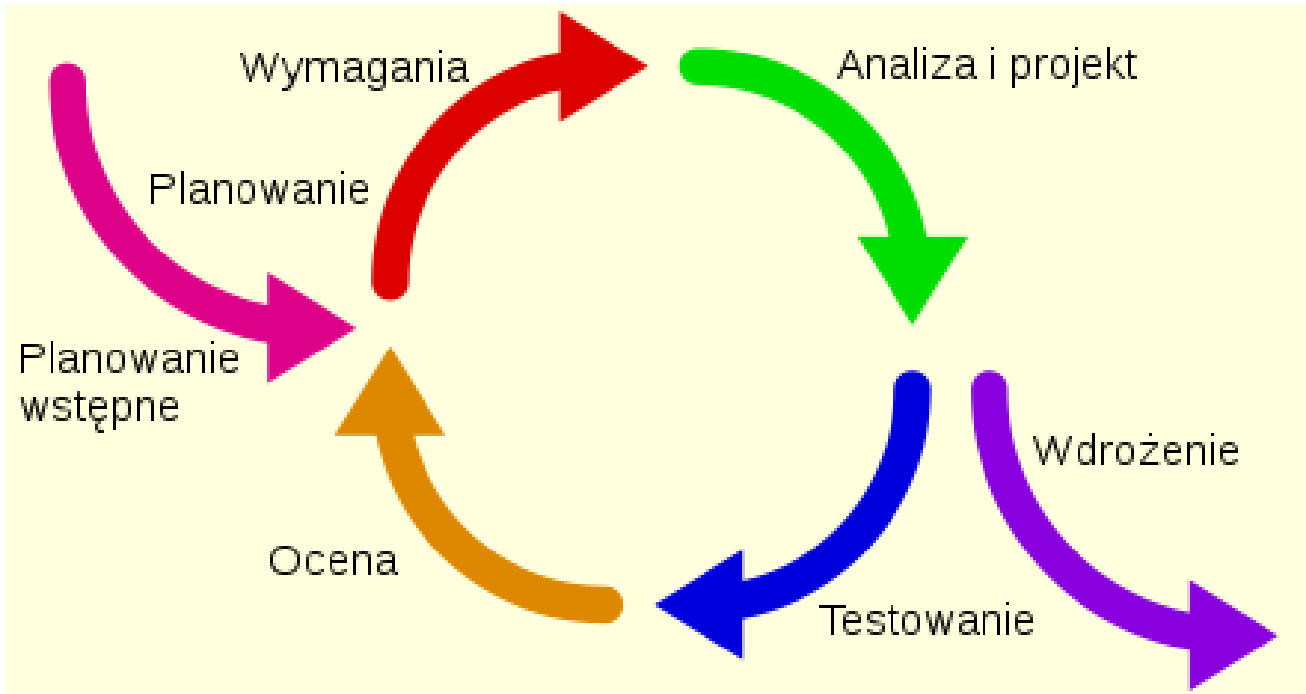
Potencjalne nie
porozumienie z klientem,
który ma nadzieje, że
prototyp to działający
system

Zalety

możliwość szybkiej demonstracji
pracującej wersji systemu

możliwość szkoleń zanim
zbudowany zostanie pełen system

Zmniejszenie wpływu ewentualnych
błędów w początkowych fazach
cyklu życia na projekt



Model przyrostowy

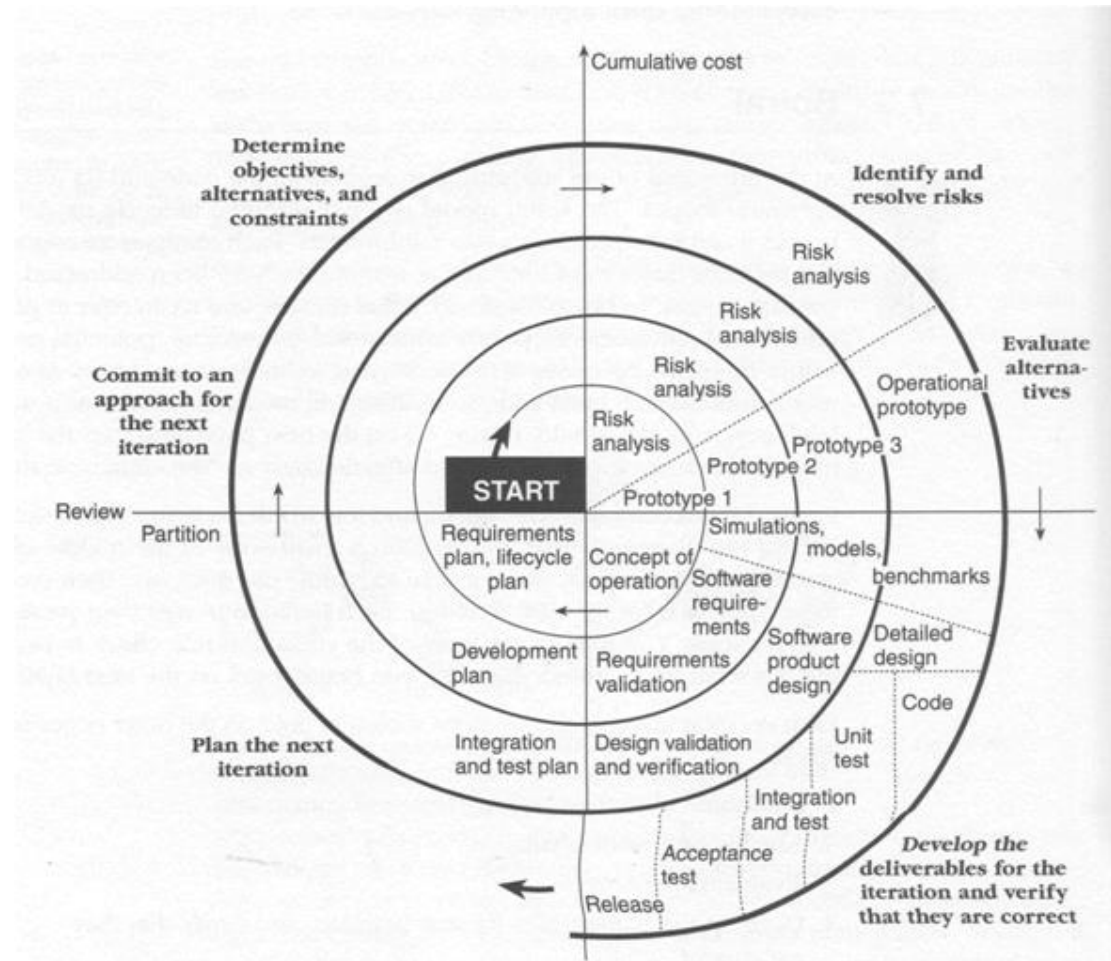
Model spiralny

Analiza

Konstrukcja

Testowanie

Planowanie



Model spiralny

Wady

Model nigdy do końca nie jest dopracowany

Wymaga doświadczenia i wiedzy (w tym ekonomicznej)

Wysoki koszt usuwania błędów w finalnych etapach (iteracjach)

Zalety

Można wykorzystać gotowe komponenty

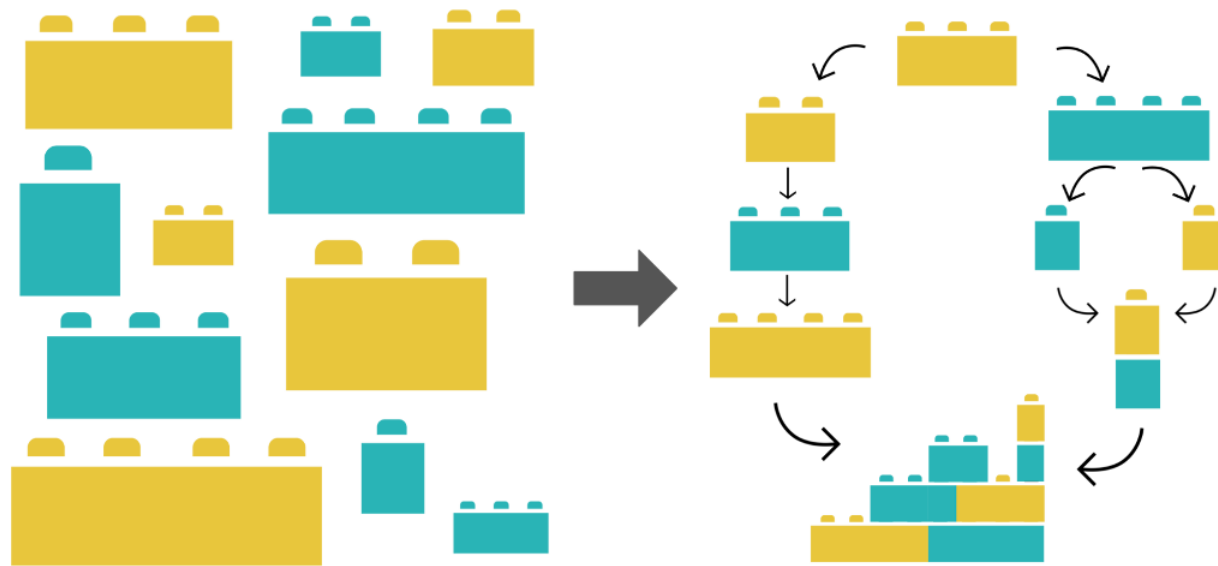
Faza oceny w każdym cyklu pozwala uniknąć błędów lub wcześniej je wykryć

Cały czas istnieje możliwość rozwijania oprogramowania

Częste kontrole jakości w kolejnych cyklach spirali

Nastawienie na wykrywanie błędów i działania kontrolne, a nie na zapobieganie

Orientacja na zarządzanie, czas i budżet.



Model komponentowy

WHAT IF I TOLD YOU

ITS NOT OVER YET

Modele cyklu życia

- ▶ https://fizyka.ujk.edu.pl/pl/files/lectures/Inzynieria_oprogramowania/UJK-IO-ModeleEtapy.pdf
- ▶ https://www.is.umk.pl/~grochu/wiki/doku.php?id=zajecia:ppz:cykl_zycia_oprogramowania
- ▶ <http://cacin.prv.pl/2.htm>
- ▶ <https://www.informatique-mania.com/pl/informatique/cycle-de-vie-du-logiciel/>
- ▶ https://edux.pjwstk.edu.pl/mat/200/lec/wyklady/2_2.html
- ▶ <https://ichi.pro/pl/przeglad-cyklu-zycia-oprogramowania-sdlc-215394495583694>
- ▶ http://zofia.kruczkiewicz.staff.iia.pwr.wroc.pl/wyklady/Kierowanie_projektem/W_2_INKU011.pdf