

Cykle wytwarzania oprogramowania

dr hab. inż. Michał Śmiałek, prof. uczelni
dr inż. Kamil Rybiński



**Wydział
Elektryczny**
POLITECHNIKA WARSZAWSKA

Inżynieria oprogramowania



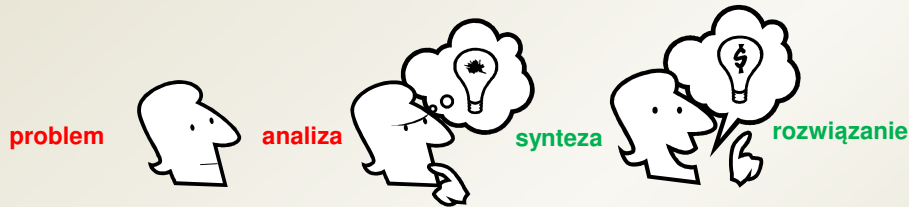
1

Podział inżynierii oprogramowania na dyscypliny

- Cykl wytwarzania oprogramowania powinien uwzględniać czynności z różnych dyscyplin
- Co to jest dyscyplina?
 - spójny zestaw czynności, które są powiązane z zasadniczym obszarem działania w ramach projektu konstrukcji oprogramowania
- Efektem czynności są produkty, np.:
 - Instrukcje użytkownika,
 - Specyfikacje wymagań,
 - Dokumentacja projektowa,
 - Kod źródłowy,
 - Plik wykonywalny, ...

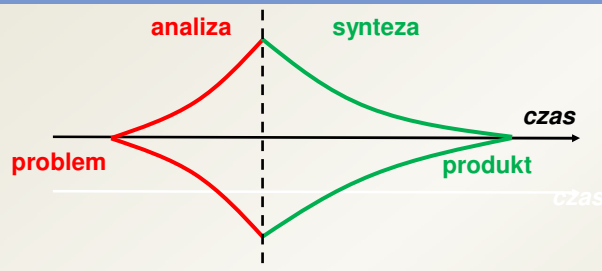
2

Inżynieria oprogramowania to działalność inżynierska



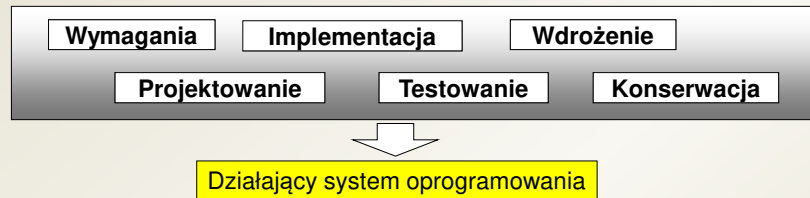
- Podział na dwie grupy czynności
 - Analiza - dokładne zidentyfikowanie i zrozumienie problemu
 - Synteza - rozwiązanie problemu poprzez realizację i scalanie mniejszych elementów w spójną całość
- Proces wytwarzania oprogramowania to ciąg czynności prowadzących od postawienia problemu do wytworzenia głównego produktu - systemu oprogramowania

Od analizy do syntezy ...



- Czynności analityczne: opisanie środowiska, specyfikowanie wymagań
- Czynności syntetyczne: projektowanie, implementacja, wdrożenie
- Uwaga: podział na dyscypliny nie oznacza podziału na etapy w ramach całego projektu

Dyscypliny inżynierii oprogramowania

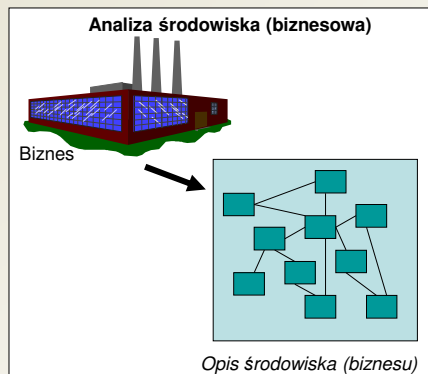


- Wymagania dostarczają informacji o kształcie systemu z punktu widzenia klienta
- Projektowanie wytwarza zestaw modeli projektowych („planów konstrukcyjnych”)
- Implementacja realizuje projekt i tworzy działający system
- Testowanie realizuje kontrolę jakości systemu
- Wdrożenie przekazuje klientowi system, który podlega utrzymaniu (konserwacji)

Dyscyplina wymagań

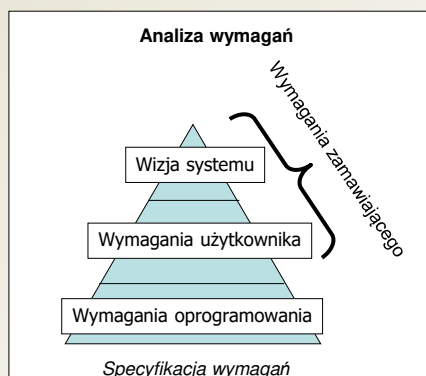
- Identyfikacja problemów klienta
- Podjęcie decyzji o budowie systemu
- Określenie kształtu i zakresu przyszłego systemu
- Dwie dyscypliny składowe
 - Analiza środowiska (biznesu) - opisanie aktualnego i docelowego stanu środowiska (np. procesów biznesowych)
 - Analiza wymagań - określenie potrzeb zamawiającego (np. w ramach umowy) oraz szczegółów wymaganego działania systemu

Analiza środowiska



- Produkt: opis środowiska
 - Opis fragmentu świata, który jest obszarem działań klienta
 - Precyzyjny i zrozumiały model struktury środowiska (biznesu) i sposobu jego działania
 - Zawiera m.in. słownik pojęć, opis procesów (biznesowych)

Analiza wymagań



- Produkt: specyfikacja wymagań
 - Piramida: coraz bardziej szczegółowe wymagania
 - Wizja: ogólne cechy systemu w odpowiedzi na potrzeby klienta
 - Wymagania użytkownika: zakres systemu
 - Wymagania oprogramowania: szczegóły działania, danych, wyglądu, ...

Dyscyplina projektowania

- Opracowanie „planów projektowych” na podstawie wymagań
- Produkt: fizyczny model architektoniczny
 - Projekt elementów wykonawczych (maszyn, sieci, ...)
- Produkt: architektura logiczna
 - Definicja komponentów oprogramowania i usług
 - Definicja danych wymienianych między komponentami
- Produkt: szczegółowy projekt modułu
 - Model struktury modułu (klasy, relacje)
 - Model działania modułu (sekwencje komunikatów)

Dyscyplina implementacji

- Wykonanie (zakodowanie) i uruchomienie zaprojektowanych modułów
- Produkt: kod (program)
 - Kod źródłowy oraz skompilowany kod wykonywalny
 - Stworzony zgodnie z regułami danego języka i technologii oraz projektem
- Produkt: środowisko implementacyjne
 - Fizyczne środowisko, w którym sprawdzane jest działanie systemu
- Produkt: środowisko produkcyjne
 - Środowisko, w którym system jest używany przez klienta
 - Zgodne z architekturą fizyczną systemu
- Produkt: testy jednostkowe i integracyjne
 - Plan i wyniki testów przeprowadzanych przez zespół implementacyjny (programistów)

Dyscyplina walidacji (testowania)

- Sprawdzenie, czy zbudowany system spełnia wymagania
- Testy wykonywane w ścisłej współpracy z klientem
- Produkt: test akceptacyjny
 - Sprawdzenie realizacji konkretnych potrzeb klienta (wymagań)
 - Zawiera scenariusz testu i wyniki jego wykonania
- Produkt: plan testów
 - Ułożenie testów w odpowiedniej kolejności
- Produkt: scenariusz testu
 - Instrukcja postępowania podczas wykonywania testu
 - Kryteria sukcesu

Dyscyplina wdrożenia

- Przeniesienie systemu do środowiska produkcyjnego i przekazanie klientowi
- Produkt: zainstalowany system
 - System gotowy działający w środowisku produkcyjnym
- Produkt: dokumentacja (użytkownika i techniczna)
- Produkt: szkolenia
 - Materiały pozwalające na nauczanie użytkowników i administratorów systemu
 - Nauczanie zasad operowania systemem
- Różne strategie wdrożeń
 - Bezpośrednie (od razu), równoległe (ze starym systemem), pilotażowe (kolejne jednostki), stopniowe (kolejne funkcje)

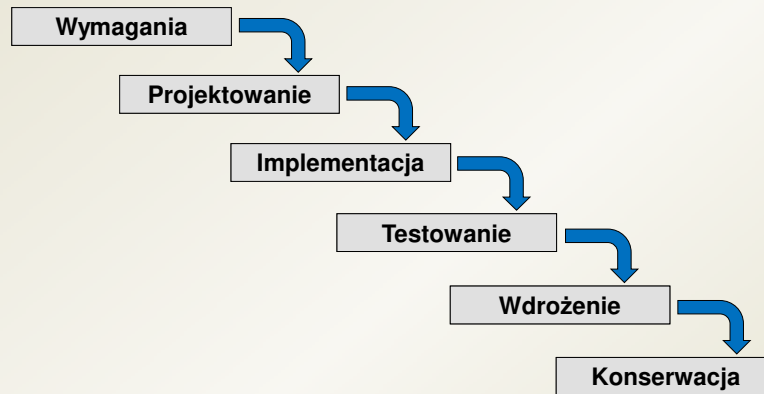
Dyscyplina konserwacji

- Zapewnienie poprawnego funkcjonowania systemu po jego wytworzeniu
 - Czynności naprawcze (usuwanie defektów),
 - Czynności adaptacyjne (dostosowanie do zmian),
 - Czynności ulepszające (rozszerzenie funkcjonalności),
 - Czynności prewencyjne (przygotowanie do zmian).
- Dyscyplina konserwacji może przerodzić się w kolejny projekt

Cykl wytwarzania oprogramowania

- Cykl wytwórczy definiuje rozłożenie czynności poszczególnych dyscyplin w czasie
- Cykl wodospadowy
 - Sekwencja dobrze wyodrębnionych kroków (faz, etapów) w ramach kolejnych dyscyplin
- Cykl iteracyjny
 - Wielokrotne wykonywanie czynności dyscyplin - powtarzanie czynności w ramach różnych iteracji
- Cykle posiadają warianty zależne od przyjętej metodyki
- W zależności od projektu wybrany cykl może ulec modyfikacji

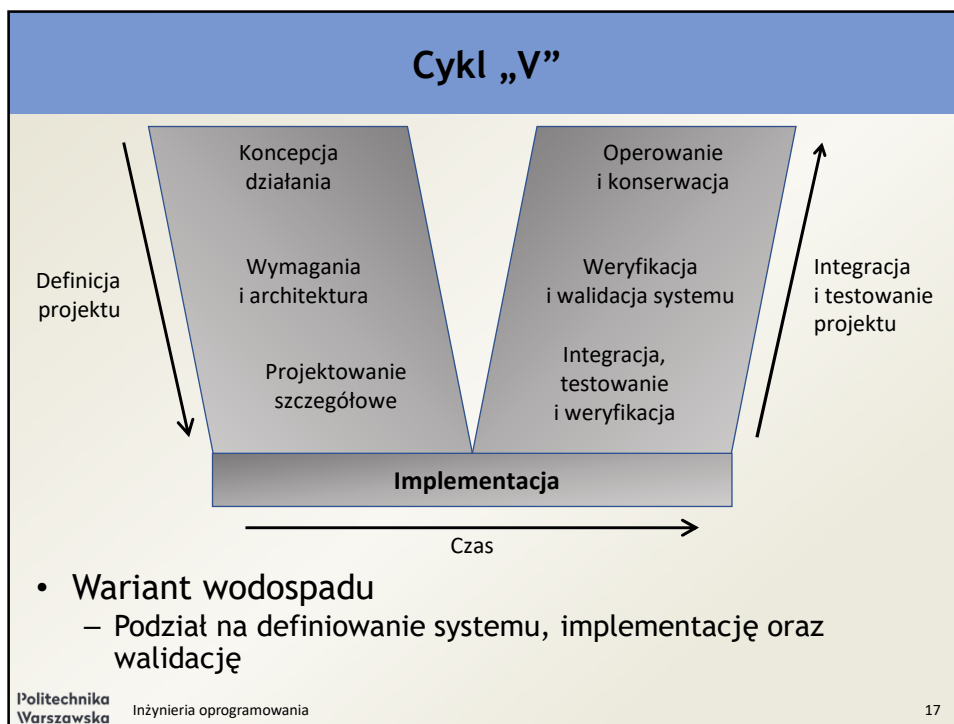
Cykl wodospadowy (kaskadowy)



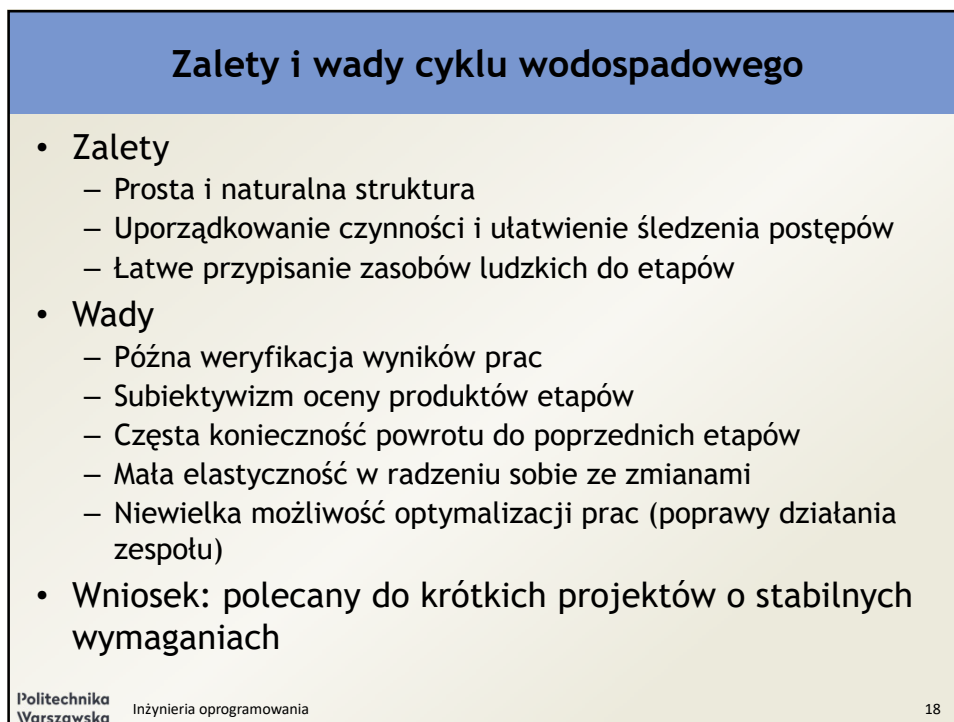
- Etapy cyklu odpowiadają dyscyplinom
- Przejście do następnego etapu po zakończeniu poprzedniej

Cechy cyklu wodospadowego

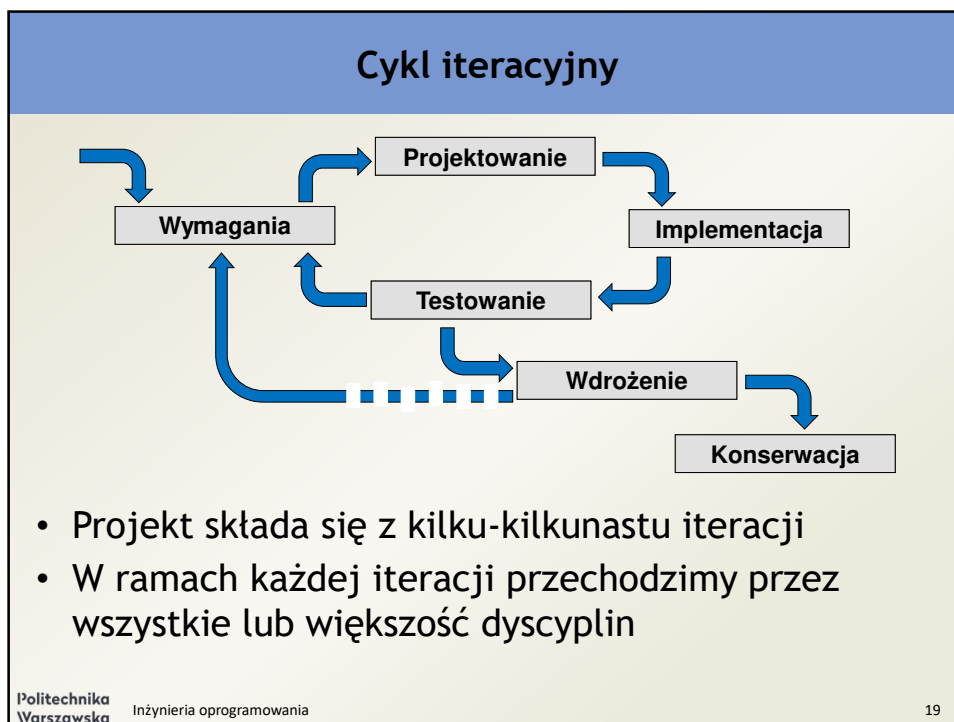
- Każdy etap kończy się wytworzeniem określonego produktu
 - Produktu poprzedniego etapu są punktem wyjścia dla następnego
- W praktyce etapy się zazębiają
 - „Nawroty” wodospadu - powrót do poprzedniego etapu
 - Sąsiadujące etapy działają częściowo równolegle - przekazują sobie informacje



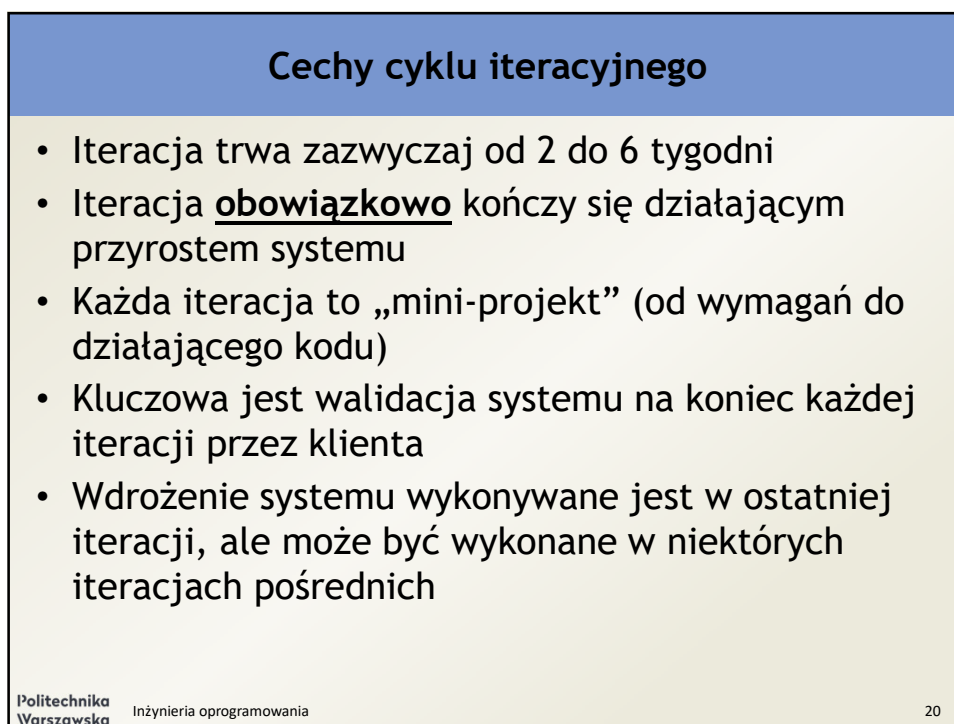
17



18



19



20

Zalety i wady cyklu iteracyjnego

- **Zalety**
 - Lepsze włączenie klienta w działania w projekcie
 - Szybka walidacja zgodności systemu z oczekiwaniami klienta - zmniejszenie ryzyka niepowodzenia
 - Szybkie odkrywanie problemów technicznych
 - Możliwość ustalenia priorytetów (kolejność realizacji w iteracjach)
 - Łatwiejsze zarządzanie zmianami
- **Wady**
 - Bardziej skomplikowane zarządzanie personelem
 - Większa złożoność i koszt zarządzania projektem
- **Wniosek:** polecany dla większości typowych projektów systemów interaktywnych, wykonywanych na zamówienie