

Wytwarzanie, integracja i testowanie systemów informacyjnych



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 1

Literatura

Podstawą części tego cyklu wykładów będzie książka:

Andrzej Jaskiewicz
Inżynieria Oprogramowania
Wydawnictwo HELION 1997

Oprócz tej pozycji, do wykorzystania są następujące pozycje:

M. Kliszewski. **Inżynieria Oprogramowania Obiektowego, część 1 i 2**
Wydawnictwo Książki Technicznej RESPEKT, 1994

M. Flasiński. **Wstęp do Analitycznych Metod Projektowania Systemów Informatycznych**, Wydawnictwo Naukowo-Techniczne 1997

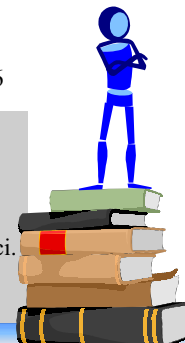
C. Mazza, et all. **Software Engineering Guides**. Prentice Hall Europe 1996

Wykłady będą udostępnione w formie slajdów pod:
<http://www.ipipan.waw.pl/~subieta>

Również do wykorzystania:

K.Subieta. Słownik często spotykanych terminów dotyczących obiektowości.
Prace IPI PAN Nr 839, wrzesień 1997.

Format elektroniczny poprzez: <http://www.ipipan.waw.pl/~subieta>



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 2

Przedmiot inżynierii oprogramowania

Inżynieria oprogramowania jest to wiedza techniczna dotycząca wszystkich faz cyklu życia oprogramowania. Traktuje oprogramowanie jako produkt.

Dobre oprogramowanie powinno być:

- ☞ zgodne z wymaganiami użytkownika,
- ☞ niezawodne,
- ☞ efektywne,
- ☞ łatwe w konserwacji,
- ☞ ergonomiczne.



Produkcja oprogramowania jest procesem składającym się z wielu faz. Kodowanie (pisanie programów) jest tylko jedną z nich, niekoniecznie najważniejszą.

Inżynieria oprogramowania jest praktyczną wiedzą techniczną, a nie pochodną "teorii".

Metody matematyczne w inżynierii oprogramowania są **dramatycznie nieskuteczne**. Są one ubocznym produktem "naukowych" karier i do niczego innego nie służą. *"Jeżeli ktoś będzie cię do nich przekonywał lub zmuszał **po prostu powiedz nie.**"*

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 3

Walka z "kryzysem oprogramowania"

- ✦ Stosowanie technik i narzędzi ułatwiających pracę nad złożonymi systemami;
- ✦ Korzystanie z metod wspomagających analizę nieznanych problemów oraz ułatwiających wykorzystanie wcześniejszych doświadczeń;
- ✦ Usystematyzowanie procesu wytwarzania oprogramowania, tak aby ułatwić jego planowanie i monitorowanie;
- ✦ Wytworzenie wśród producentów i nabywców przekonania, że budowa dużego systemu wysokiej jakości jest zadaniem wymagającym profesjonalnego podejścia.

Również:

- podnoszenie poziomu **abstrakcji**
- **dekompozycja** problemów na mniejsze komponenty
- **ponowne (wielokrotne) użycie** gotowych komponentów
- dostosowanie metod tworzenia oprogramowania do naturalnej **ludzkiej** percepcji, skłonności i ograniczeń.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 4

Zagadnienia inżynierii oprogramowania

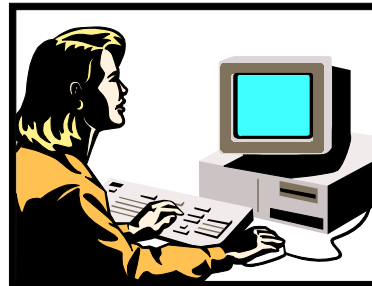
- ✦ Sposoby prowadzenia przedsięwzięć informatycznych.
- ✦ Techniki planowania, szacowania kosztów, harmonogramowania i monitorowania przedsięwzięć informatycznych.
- ✦ Metody analizy i projektowania systemów.
- ✦ Techniki zwiększania niezawodności oprogramowania.
- ✦ Sposoby testowania systemów i szacowania niezawodności.
- ✦ Sposoby przygotowania dokumentacji technicznej i użytkowej.
- ✦ Procedury kontroli jakości.
- ✦ Metody redukcji kosztów konserwacji (usuwania błędów, modyfikacji i rozszerzeń)
- ✦ Techniki pracy zespołowej i czynniki psychologiczne wpływające na efektywność pracy.

Sukces inżynierii oprogramowania: rozwój metodyk i narzędzi + edukacja.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 5

Modele cyklu życia oprogramowania

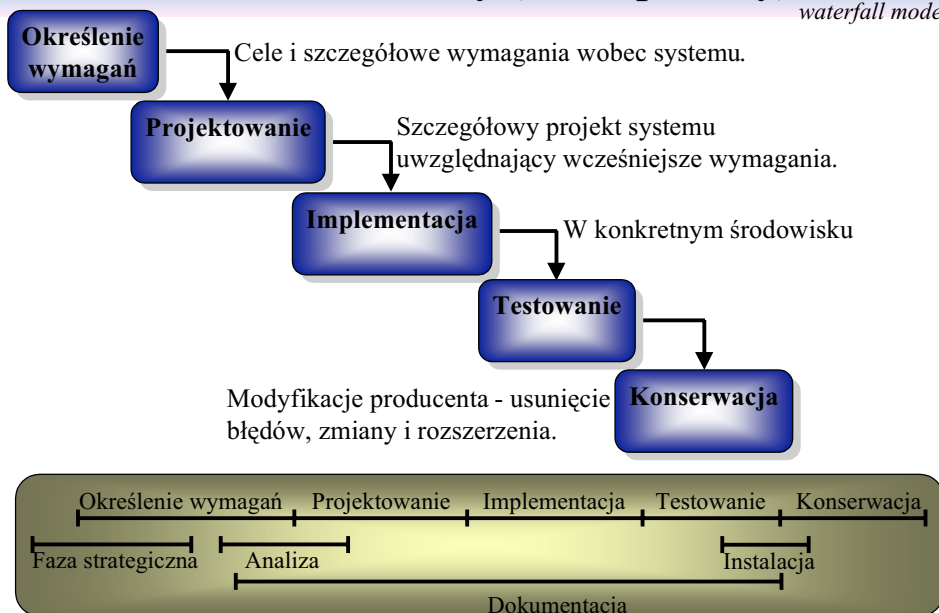
- ✦ Model kaskadowy (wodospadowy)
- ✦ Realizacja kierowana dokumentami (mutacja modelu kaskadowego)
- ✦ Prototypowanie
- ✦ Programowanie odkrywcze
- ✦ Realizacja przyrostowa
- ✦ Montaż z gotowych elementów
- ✦ Model spiralny
- ✦ Formalne transformacje



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 6

Model kaskadowy (wodospadowy)

waterfall model



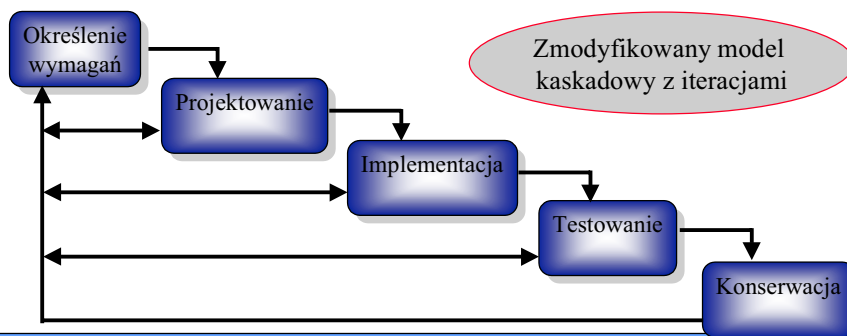
K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 7

Ocena modelu kaskadowego

Istnieją zróżnicowane poglądy co do przydatności praktycznej modelu kaskadowego. Podkreślane są następujące wady:

- ☞ Narzucenie twórcom oprogramowania ścisłej kolejności wykonywania prac
- ☞ Wysoki koszt błędów popełnionych we wczesnych fazach
- ☞ Długa przerwa w kontaktach z klientem

Z drugiej strony, jest on do pewnego stopnia niezbędny dla planowania, harmonogramowania, monitorowania i rozliczeń finansowych.



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 8

Realizacja kierowana dokumentami

- ☞ Przyjęty przez armię amerykańską dla realizacji projektów w języku Ada.
- ☞ Jest to odmiana modelu kaskadowego.
- ☞ Każda faza kończy się sporządzeniem szeregu dokumentów, w których opisuje się wyniki danej fazy.
- ☞ Łatwe planowanie, harmonogramowanie oraz monitorowanie przedsięwzięcia.
Dodatkowa zaleta: (teoretyczna) możliwość realizacji dalszych faz przez inną firmę.

Wady

- ✦ Duży nakład pracy na opracowanie dokumentów zgodnych ze standardem (DOD STD 2167) - ponad 50% całkowitych nakładów.
- ✦ Przerwy w realizacji niezbędne dla weryfikacji dokumentów przez klienta.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 9

Prototypowanie

Sposób na uniknięcie zbyt wysokich kosztów błędów popełnionych w fazie określania wymagań. Zalecany w przypadku, gdy określenie początkowych wymagań jest stosunkowo łatwe.

Fazy

- ☞ ogólne określenie wymagań
- ☞ budowa prototypu
- ☞ weryfikacja prototypu przez klienta
- ☞ pełne określenie wymagań
- ☞ realizacja pełnego systemu zgodnie z modelem kaskadowym

Cele

- ☞ wykrycie nieporozumień pomiędzy klientem a twórcami systemu
- ☞ wykrycie brakujących funkcji
- ☞ wykrycie trudnych usług
- ☞ wykrycie braków w specyfikacji wymagań

Zalety

- ☞ możliwość demonstracji pracującej wersji systemu
- ☞ możliwość szkoleń zanim zbudowany zostanie pełny system

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 10

Metody prototypowania

- ✦ **Niepełna realizacja:** objęcie tylko części funkcji
- ✦ **Języki wysokiego poziomu:** Smalltalk, Lisp, Prolog, 4GL, ...
- ✦ **Wykorzystanie gotowych komponentów**
- ✦ **Generatory interfejsu użytkownika:** wykonywany jest wyłącznie interfejs, wewnątrz systemu jest "podróbka".
- ✦ **Szybkie programowanie (quick-and-dirty):** normalne programowanie, ale bez zwracania uwagi na niektóre jego elementy, np. zaniechanie testowania

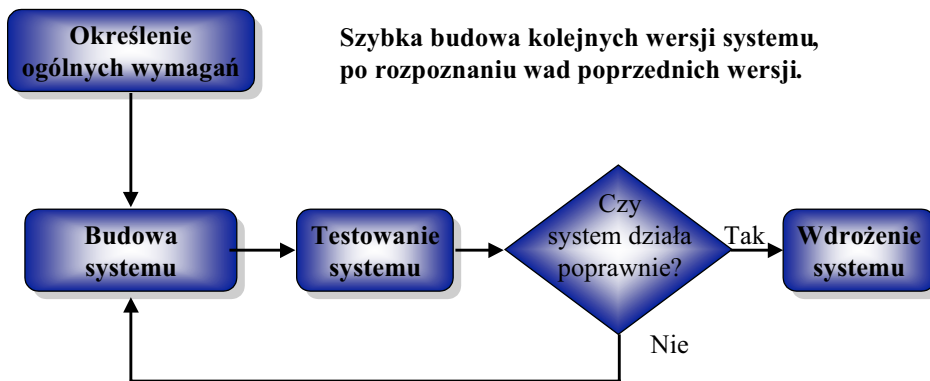
Pamiętać należy, że w niektórych krajach i systemach
...najtrwalsze są prowizorki...
(patrz osiedle Jelonki lub "siedziba" IPI PAN)



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 11

Programowanie odkrywcze

exploratory programming

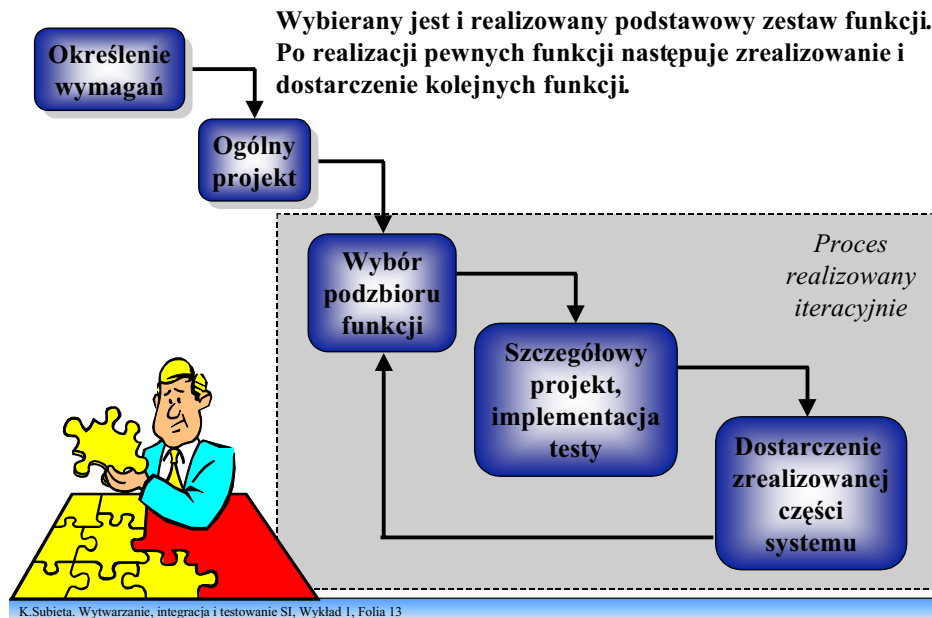


- Wady**
- ✦ Prawie niemożliwe jest zachowanie sensownej struktury systemu. Struktura początkowa zaginie w kolejnych iteracjach. W pewnym momencie usuwanie błędów powoduje dodawanie nowych błędów.
 - ✦ Testowanie systemu może odbywać się tylko przy pełnym udziale klienta.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 12

Realizacja przyrostowa

incremental development



Montaż z gotowych elementów

composition of reusable components

Kładzie nacisk na możliwość redukcji nakładów poprzez maksymalne wykorzystanie podobieństwa tworzonego oprogramowania do wcześniej tworzonych systemów.

Mogą być wykorzystane:

- ☞ biblioteki
- ☞ języki czwartej generacji
- ☞ pełne aplikacje, np. wykorzystanie przeglądarki plików w MS Windows.

Temat jest znany pod nazwą ponowne użycie (*reuse*)

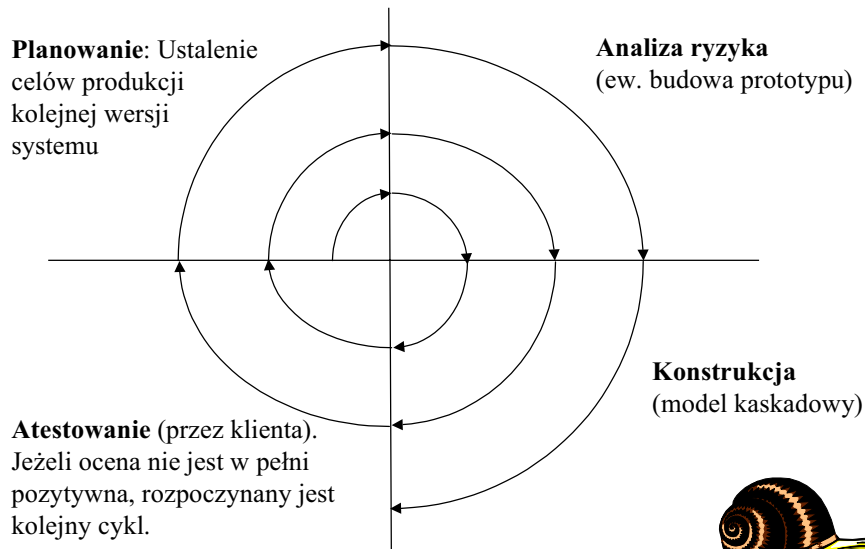
- Metody**
- ☞ zakup elementów ponownego użycia od dostawców
 - ☞ przygotowanie elementów poprzednich przedsięwzięć do ponownego użycia

- Zalety**
- ☞ wysoka niezawodność
 - ☞ zmniejszenie ryzyka
 - ☞ efektywne wykorzystanie specjalistów
 - ☞ narzucenie standardów

- Wady**
- ☞ dodatkowy koszt przygotowania elementów ponownego użycia
 - ☞ ryzyko uzależnienia się od dostawcy elementów
 - ☞ niedostatki narzędzi wspomagających ten rodzaj pracy.

Model spiralny

Istnieje wiele wariantów tego modelu.

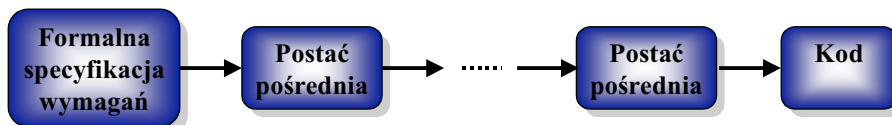


K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 15

Formalne transformacje

Jest on postulowany w ramach tzw. nurtu formalnego w inżynierii oprogramowania.

Wymagania na system są formułowane w pewnym formalnym języku, następnie poddawane są kolejnym transformacjom, aż do uzyskania działającego kodu.



Transformacje są wykonywane bez udziału ludzi (czyli w istocie, język specyfikacji wymagań byłby **nowym “cudownym” językiem programowania**).

Niestety, takie pomysły dowodzą braku wyobraźni, kompetencji i pomysłu.

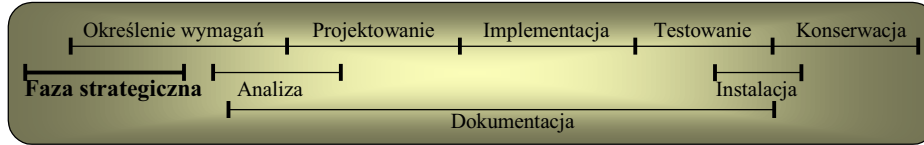
W ponad 20-letniej praktyce nie spotkałem się z przypadkiem, aby tego rodzaju koncepcje projektanci traktowali poważnie.

Metody matematyczne **nigdy** nie utworzą pełnej metodyki projektowania, mogą tylko wspomagać pewne szczegółowe tematy (tak jak w biologii, medycynie, ekonomii,...).

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 16

Faza strategiczna

strategy phase, feasibility study



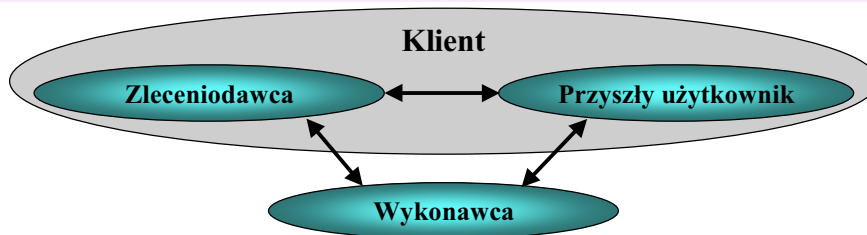
Faza strategiczna jest wykonywana zanim podejmowana jest decyzja o realizacji przedsięwzięcia. Nazywana także **strategicznym planem rozwoju informatyzacji** (SPRI) lub (niewłaściwie) **studium osiągalności**.

Czynności

- ☛ dokonanie serii rozmów (wywiadów) z przedstawicielami klienta
- ☛ określenie celów przedsięwzięcia z punktu widzenia klienta
- ☛ określenie zakresu oraz kontekstu przedsięwzięcia
- ☛ ogólne określenie wymagań, wykonanie zgrubnej analizy i projektu systemu
- ☛ propozycja kilku możliwych rozwiązań (sposobów realizacji systemu)
- ☛ oszacowanie kosztów oprogramowania
- ☛ analiza rozwiązań
- ☛ prezentacja wyników fazy strategicznej przedstawicielom klienta oraz korekta wyników
- ☛ określenie wstępnego harmonogramu przedsięwzięcia oraz struktury zespołu realizatorów
- ☛ określenie standardów, zgodnie z którymi realizowane będzie przedsięwzięcie

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 17

Faza strategiczna - współpraca z klientem



Po stronie klienta warto wyróżnić zleceniodawcę i przyszłych użytkowników. Starać się uwzględnić kryteria obydwu stron, ale należy pamiętać, że system będzie głównie oceniany przez przyszłych użytkowników.

- ✦ Ważnym elementem fazy strategicznej jest jasne określenie **celów** przedsięwzięcia z punktu widzenia klienta. Nie zawsze są one oczywiste, co często powoduje nieporozumienia pomiędzy klientem i wykonawcą.
- ✦ Równie ważne jest określenie **ograniczeń klienta** (np. finansowych, infrastruktury, zasobów ludzkich, czasu wdrożenia, itd.)

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 1, Folia 18