

IO - inżynieria oprogramowania

dr inż. M. Żabińska,
e-mail: zabinska@agh.edu.pl
<http://home.agh.edu.pl/~zabinska/>

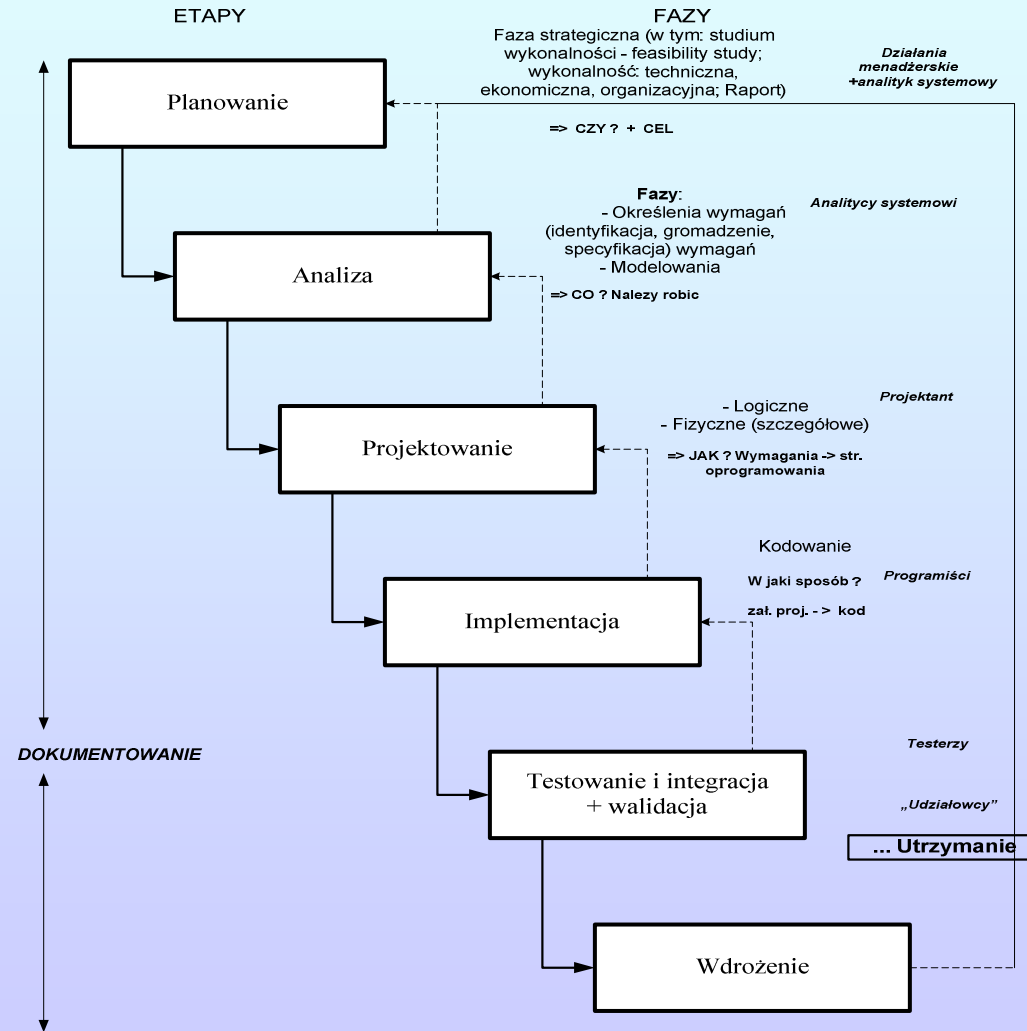
Faza określania wymagań (1)

- **Cel fazy określania wymagań** – dokładne ustalenie wymagań klienta wobec tworzonego systemu
- Zamiana celów klienta na konkretne wymagania zapewniające ich osiągnięcie
- Klient rzadko wie jakie dokładnie wymagania zapewnią osiągnięcie celów systemu

Faza określenia wymagań (2)

- Dwa przypadki:
 - system na konkretne zamówienie: ściśła współpraca z klientem; negocjacje, i/lub firma bierze udział w przetargu,
 - firma produkująca oprogramowanie sprzedawane rynkowo: planuje się produkcję systemu (nowej wersji); - uwagi użytkowników poprzednich wersji, - klientów potencjalnych, - badania marketingowe (również w fazie określania wymagań).

Faza określenia wymagań (3)



Kaskadowy (klasyczny, konwencjonalny) cykl tworzenia systemu informatycznego
(----- z iteracjami)

Faza określenia wymagań (4)

- Proces określania wymagań – złożony
- konstruowanie zbioru wymagań wobec przyszłego systemu zgodnie z postawionymi celami
- wykonywany we współdziałaniu przedstawicieli producenta z udziałowcami (2 przypadki)
- Miejsce tej fazy w cyklu tworzenia/życia systemu/oprogramowania (el. w F. Strat.); Etap analizy: F. Okr. Wymagań, F. Modelowania):
- duże znaczenie dla sukcesu całego przedsięwzięcia
- duży koszt błędów popełnionych w tej fazie (rysunki: propagacja błędów, wzrost kosztu)

Faza określenia wymagań - trudności (1)

- **Trudności w określaniu wymagań:**
 - klient nie wie jak można osiągnąć założone cele; a można na wiele sposobów,
 - duże systemy – wykorzystanie przez wielu użytkowników; cele często sprzeczne; różna terminologia użytkowników w opisie problemów,
 - zleceniodawcy i użytkownicy to inne osoby; decydujący głos zleceniodawców; nie zawsze potrafią przewidzieć potrzeby rzeczywistych użytkowników (udziałowcy).

Faza określenia wymagań – czynności (1)

- rozmowy (dyskusje, wywiady) oraz prezentacja wyników udziałowcom i korygowanie wyników na podstawie uzyskanych uwag (jak w F. Strat.);
- Definicja wymagań (*requirements definition*) – ogólny opis w języku naturalnym (wstępny wynik rozmów z przedstawicielami klienta,
- Specyfikacja wymagań (*requirements specification*) – częściowo ustrukturyzowany zapis: częściowo w j. naturalnym, częściowo sformalizowana notacja,
- (wymagania klienta opisywane na różnych poziomach abstrakcji – ogólności)

Faza określenia wymagań – czynności (2)

Opis:

- ogólny opis w języku naturalnym częściowo ustrukturyzowany zapis
- częściowo sformalizowana notacja – pseudokod
- formularze i szablony
- notacje graficzne (Use Case – UML, Jacobson i in.)
- **Najbardziej podstawowy błąd:**
- koncentrowanie się na sytuacjach typowych i pomijanie wyjątków (udziałowcy jak i analitycy)

Faza określenia wymagań – czynności (3)

- **Cechy dobrego opisu wymagań:**
 - kompletny oraz niesprzeczny,
 - ma opisywać zewnętrzne zachowanie systemu, a nie sposób jego realizacji,
 - ma obejmować ograniczenia, przy jakich będzie pracować system,
 - ma być łatwy w modyfikacji,
 - ma brać pod uwagę przyszłe potencjalne zmiany wymagań wobec systemu,
 - ma opisywać zachowanie systemu w niepożądanych sytuacjach.

Faza określenia wymagań – rodzaje wymagań ⁽¹⁾

- Wymagania wobec systemu:
- Wymagania funkcjonalne – opisują funkcje (czynności, operacje, usługi) wykonywane przez system,
- Wymagania niefunkcjonalne – opisują ograniczenia, przy zachowaniu których system powinien realizować swe funkcje.

Faza określenia wymagań – wyniki⁽¹⁾

- **Raport:** dokument, w którym zbiera się i opisuje wymagania dotyczące przyszłego systemu (powinien być podstawą ustaleń (kontraktu) między klientem a producentem:
 - **Wprowadzenie** – cele, zakres i kontekst systemu – wyniki fazy strategicznej (być może częściowo zmodyfikowane),
 - Opis **ewolucji systemu** – opis przewidywanych zmian wymagań wobec systemu,
 - Opis **wymagań funkcjonalnych**,
 - Opis **wymagań niefunkcjonalnych**,
 - **Model systemu** (zwykle w formie graficznej),
 - **Słownik pojęć** – definicje i opis terminów niezrozumiałych dla jednej ze stron (terminy dotyczące dziedziny i informatyczne), a także terminy niejednoznaczne, zależne od kontekstu.

Faza określenia wymagań – wyniki⁽²⁾

- Może być uzupełniony o dodatki:
 - Specyfikacja wymagań funkcjonalnych,
 - Specyfikacja wymagań нефункциональных,
 - Wymagania sprzętowe,
 - Wymagania dotyczące bazy danych,
 - Indeks – pomoc w wyszukiwaniu w dokumencie konkretnych informacji.

Faza określenia wymagań – wyniki⁽³⁾

- Zalecenia przy tworzeniu Raportu
- wymagania funkcjonalne powinny być wyraźnie oddzielone od wymagań niefunkcjonalnych,
- rozdzielać opisy poszczególnych wymagań, np. umieszczając opis każdego w osobnym akapicie,
- dla każdego wymagania powinno być podane uzasadnienie – cel/e przedsięwzięcia, których osiągnięcie wspomaga dane wymaganie.
- *Dodatkowo: wymagania – podstawą testowania systemu; plan testów(użytkowych – funkcjonalnych)*

Wymagania funkcjonalne (1)

- **Wymagania funkcjonalne** opisują funkcje (czynności, operacje, usługi) wykonywane przez system
- Często stosowany sposób opisu wymagań – język naturalny
- Wady:
 - Niejednoznaczność języka naturalnego; utrudnia precyzyjny zapis wymagań, może prowadzić do różnej interpretacji tego samego tekstu przez różne osoby
 - Elastyczność języka naturalnego – pozwala te same treści wyrazić na różne sposoby, co utrudnia wykrycie powiązanych wymagań; może prowadzić do przeoczenia sprzeczności wymagań sformułowanych w różny sposób lecz dotyczących tych samych funkcji.

Wymagania funkcjonalne (2)

- Notacje formalne pozbawione są powyższych wad ale w zdecydowanej większości przypadków – niezrozumiałe dla klienta. Mogą one więc być jedynie uzupełnieniem zapisu słownego.
- Swego rodzaju kompromis pomiędzy powyższymi metodami – stosowanie formularzy do zapisu wymagań funkcjonalnych:
 - wykorzystują język naturalny,
 - zapis podzielony na konkretne pola, co pozwala uniknąć szeregu błędów w określeniu wymagań,
 - definiowanie wymagań funkcjonalnych – z reguły podaje się jedynie wymagany efekt realizacji funkcji a nie algorytm (deklaratywny opis systemu).

Wymagania funkcjonalne (3)

- Przykład formularza opisu wymagania:
 - ***Nazwa funkcji***
 - Opis
 - Dane wejściowe
 - Źródło danych wejściowych
 - Dane wyjściowe
 - Przeznaczenie
 - Wymaga
 - Warunek początkowy
 - Warunek końcowy
 - Efekty uboczne
 - Uwagi: (Uzasadnienie)

Wymagania funkcjonalne (4)

Przykład: [*Ian Sommerville, Inżynieria oprogramowania, Specyfikacja wymagań systemu z użyciem standardowego formularza*]

Nazwa funkcji: Dodaj węzeł

Opis: Dodaje węzeł do istniejącego projektu. Użytkownik wybiera typ, położenie węzła. Po dodaniu do projektu węzeł jest zaznaczony. Uż. wybiera miejsce węzła przesuwając wskaźnik na obszar, w którym dodano węzeł.

Dane wejściowe: Typ węzła, Położenie węzła, Identyfikator projektu.

Źródło danych wejściowych: Typ węzła i położenie węzła pochodzą od użytkownika, a identyfikator projektu z bazy danych.

Dane wyjściowe: Identyfikator projektu

Przeznaczenie: Baza danych projektów. Projekt jest utrwalany w bazie danych po zakończeniu operacji.

Wymaga: Korzeniem grafu projektu musi być dany identyfikator projektu.

Warunek początkowy: Projekt otwarty, wyświetlony przez użytkownika.

Warunek końcowy: Projekt nie uległ zmianie z wyjątkiem dodania węzła danego typu o danym położeniu.

Efekty uboczne: Nie ma

Uwagi: (np. uzasadnienie)

Definicja: Eclipse/Workstation/Tools/DE/RD/3.5.1

Wymagania funkcjonalne (5)

- Liczba wymagań funkcjonalnych może być bardzo duża; konieczne jest pewnego rodzaju uporządkowanie tych wymagań, które ułatwi pracę nad nimi (złożoność !)
- Dwie metody umożliwiające zapanowanie nad dużą liczbą wymagań:
 - **hierarchiczny** zapis wymagań,
 - diagramy **przypadków użycia** (*Use Cases*).

Wymagania niefunkcjonalne (1)

- **Wymagania funkcjonalne:** jakie usługi ma oferować system, jak reagować na określone dane wejściowe, jak się zachowywać w sytuacjach.
- Często – określają czego system ma nie robić.
- Wynikają z wymagań użytkownika – definiują konkretne udogodnienia oczekiwane od systemu.
- Specyfikacja wymagań funkcjonalnych powinna być kompletna i spójna (bez sprzeczności).
- Po wykryciu sprzeczności, należy je wyeliminować z dokumentacji wymagań.
- Por. **wymagania niefunkcjonalne**

Wymagania niefunkcjonalne (2)

- **Wymagania niefunkcjonalne:** nie dotyczą bezpośrednio konkretnych funkcji systemu: ograniczenia usług i funkcji systemu (np. czasowe, dotyczące procesu tworzenia, standardy, itd.).
- Mogą być związane z właściwościami systemu, jak: czas reakcji, niezawodność, zajętość pamięci.
- Mogą definiować ograniczenia systemu (możliwości urządzeń wejścia-wyjścia, reprezentacje danych używane przez interfejsy).

Wymagania niefunkcjonalne (3)

- **Wymagania niefunkcjonalne:**
 - Zwykle dotyczą systemu jako całości a nie poszczególnych cech systemu (=> bardziej istotne niż wymagania funkcjonalne (dla nich niespełnienie = pogorszenie cech systemu ale: niespełnienie pewnego wymagania niefunkcjonalnego – system może być bezużyteczny).
 - Mogą dotyczyć: nie tworzonego systemu lecz ograniczać proces tworzenia systemu (np. specyfikacja standardów, których należy użyć w procesie, metodyka, narzędzia, sposób opisu, itd.).

Wymagania niefunkcjonalne (4)

- **Wymagania niefunkcjonalne:**
 - Wynikają z: potrzeb użytkownika, ograniczeń budżetowych, strategii firmy, konieczności współpracy z innymi systemami (sprzętu lub oprogramowania), czynników zewnętrznych jak przepisy o bezpieczeństwie, ustawy chroniące prywatność, itd.

Wymagania нефunkcjonalne (5)

- **Klasyfikacja:**
 - ***Wymagania produktowe*** – zachowanie produktu (wymagania efektywnościowe – szybkość działania systemu i potrzeb pamięci, w. niezawodności – akceptowalna częstość awarii), w. przenośności, wymagania użyteczności.
 - ***Wymagania organizacyjne*** – wynikają ze strategii i procedur w firmie-kliencie i w firmie-wytwórcy (standardy procesu, w. implementacyjne: język, metoda programowania, wymagania dostawy – kiedy dostarczyć produkt i jego dokumentację).
 - ***Wymagania zewnętrzne*** – szeroka kategoria: wszystkie wymagania wynikające z czynników zewnętrznych dla systemu i procesu jego tworzenia (wymagania współpracy: interakcje z systemami innych firm, prawne, etyczne: akceptacja).

Wymagania niefunkcjonalne (6)

- **Wymagania niefunkcjonalne:**
 - Wymagania niefunkcjonalne – trudne do weryfikacji (zapisywane jako ogólne dążenia klienta: np. łatwość użycia, zdolność do naprawy po awarii, szybka reakcja na działania użytkownika) – jakość !
 - Kłopot dla wytwórców systemu (duży margines dla interpretacji i konfliktów po dostarczeniu systemu)
 - Np.: Cel systemu: łatwy dla doświadczonych, sposób jego organizacji powinien zmniejszać liczbę błędów użytkownika...

Wymagania niefunkcjonalne (7)

- **Wymagania niefunkcjonalne:**
 - Sposób sformułowania – np.: Cel systemu: łatwy dla doświadczonych, sposób jego organizacji powinien zmniejszać liczbę błędów użytkownika...
 - Próba weryfikacji: np.: możliwość używania wszystkich funkcji systemu po szkoleniu ca 2h. Średnia liczba błędów nie powinna przekraczać 3/dzień => podanie miar

Wymagania niefunkcjonalne (8)

- Weryfikowalność: niektóre właściwości – wyrażone przez miary, możliwość testowania obiektywnego, np.:
 - szybkość (liczba transakcji/sek, czas reakcji, czas odświeżania),
 - rozmiar (KB, MB),
 - łatwość użycia (czas szkolenia, liczba okien pomocy),
 - niezawodność (średni czas bez awarii, częstość błędów),
 - solidność (czas uruchomienia po awarii, prawdopodobieństwo utraty danych, procent zdarzeń powodujących awarie),
 - przenośność (procent poleceń zależnych od platformy, liczba docelowych systemów).

Wymagania niefunkcjonalne (9)

- **Wymagania niefunkcjonalne:**
 - wymagania produktowe,
 - wymagania organizacyjne,
 - wymagania zewnętrzne.

Wymagania produktowe (1)

- **Wymagania produktowe:**
 - użyteczności,
 - sprawnościowe:
 - efektywności,
 - pamięci.
 - niezawodności,
 - przenośności.

Wymagania organizacyjne (1)

- **Wymagania organizacyjne:**
 - dostawy,
 - implementacyjne,
 - standardów.

Wymagania zewnętrzne (1)

- **Wymagania zewnętrzne:**
 - współpracy,
 - etyczne,
 - prawne:
 - **ochrony prywatności,**
 - **wymagania zabezpieczeń.:**

Wymagania niefunkcjonalne – weryfikowalność (1)

- Dobrze sformułowane wymagania niefunkcjonalne:
 - powinny być weryfikowalne – możliwość sprawdzenia czy system je spełnia.
 - wymagania sformułowane jako stwierdzenia, np.: łatwy w obsłudze, niezawodny, wydajny - nie mogą być obiektywnie zweryfikowane;
 - konieczność wyrażenia wymagań przy pomocy wielkości mierzalnych

Wymagania niefunkcjonalne – weryfikowalność (2)

- Przykład formularza opisu wymagań niefunkcjonalnych, np. w aspekcie rozważanych rozwiązań:

Cecha:

Miary:

- Wydajność: Liczba trans. obsłużonych/s.
- Rozmiar: Wymagana pamięć RAM;
Wymagana pamięć dyskowa
- Łatwość użytkowania: Czas dla przeszkolenia
Liczba stron dokumentacji

Wymagania niefunkcjonalne – weryfikowalność (3)

- **(C.d.)** Przykład formularza opisu wymagań niefunkcjonalnych, np. w aspekcie rozważanych rozwiązań:

Cecha:

Niezawodność:

–

–

–

–

Miary:

Prawdopodobieństwo błędnego wykonania podczas realizacji transakcji

Częstotliwość błędnych wykonań

Średni czas między błędnymi wykonaniami

Dostępność (procent czasu)

Wymagania niefunkcjonalne – weryfikowalność (4)

- **(C.d.)** Przykład formularza opisu wymagań niefunkcjonalnych, np. w aspekcie rozważanych rozwiązań:

Cecha:

Odporność:

Miary:

Czas restartu po awarii systemu

Prawdopodobieństwo zniszczenia danych w przypadku awarii

Wymagania niefunkcjonalne – weryfikowalność (5)

- **(C.d.)** Przykład formularza opisu wymagań niefunkcjonalnych, np. w aspekcie rozważanych rozwiązań:

Cecha:

Przenośność:

Miary:

- Procent kodu zależnego od platformy docelowej
- Liczba platform docelowych
- Koszt przeniesienia na nową platformę

Wymagania niefunkcjonalne – a jakość (1)

- W NF można stosować jako miary jakości systemu (por. ocena rozwiązań w fazie strategicznej).
- Często nazywane „pożądanymi cechami systemu informatycznego”.
- Wymagania F. i WNF – tzw. wymagania dziedzinowe – dotyczą dziedziny zastosowania systemu i odzwierciedlają charakterystykę.

Faza określania wymagań – podsumowanie ⁽¹⁾

- **Kluczowe czynniki sukcesu**
 - Zaangażowanie właściwych osób ze strony klienta,
 - Pełne rozpoznanie wymagań, wykrycie sytuacji szczególnych i nietypowych (typowy błąd: koncentrowanie się na sytuacjach typowych),
 - Sprawdzenie kompletności i spójności wymagań (przed przystąpieniem do dalszych prac),
 - Określenie wymagań нефunkcjonalnych w sposób możliwy do weryfikacji.

Faza określania wymagań – podsumowanie (2)

- Podstawowe wyniki fazy określania wymagań:
 - dokument opisujący wymagania (część główna + opcjonalne dodatki),
 - ew. plan testów.

Faza określania wymagań – podsumowanie (3)

- **Narzędzia (CASE):**
 - Wyniki: duże objętości danych (informacje n.t. wymagań)
 - Specjalna baza danych (data dictionary, repository):
 - opisy wymagań funkcjonalnych i niefunkcjonalnych,
 - opisy systemów zewnętrznych,
 - plany testów.
 - Generatory raportów: tworzenie różnych raportów na podstawie słownika danych; definiowanie własnych raportów
 - Narzędzia graficzne (np. diagramy)
 - Moduły przygotowywania dokumentacji: raporty, diagramy graficzne, informacje tekstowe
 - Budowa prototypu – elementem fazy określania wymagań; szybkie przygotowywanie prototypów (element CASE)



Koniec

Podsumowanie



How the customer explained it



How the Project Leader understood it



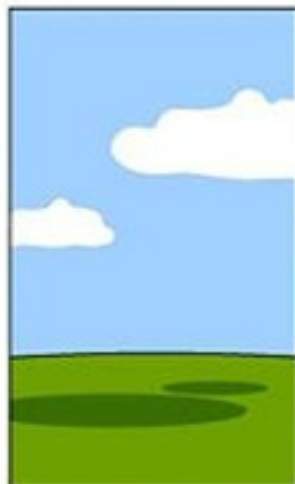
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



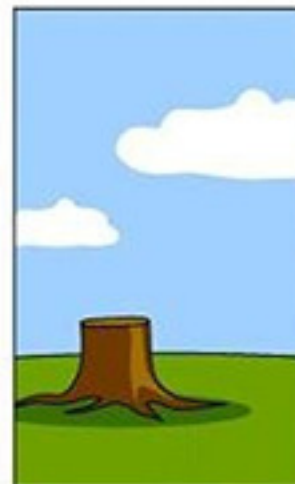
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed