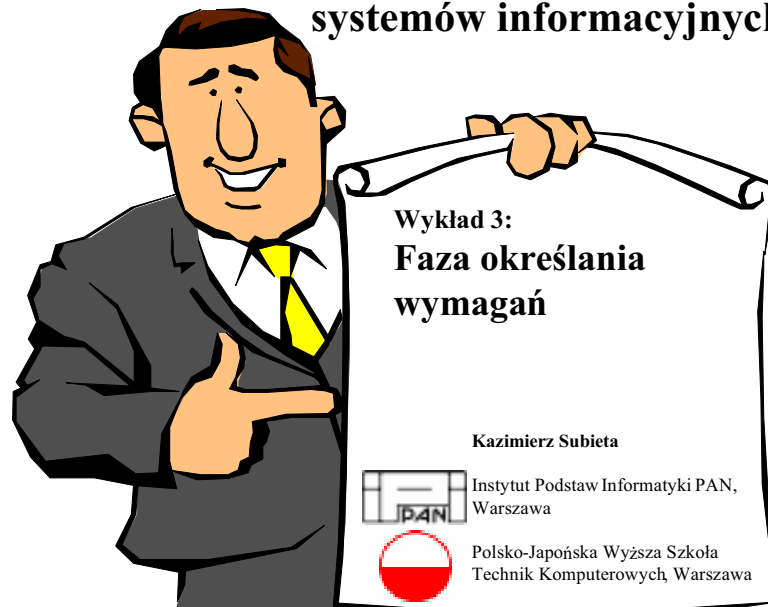


# Wytwarzanie, integracja i testowanie systemów informacyjnych



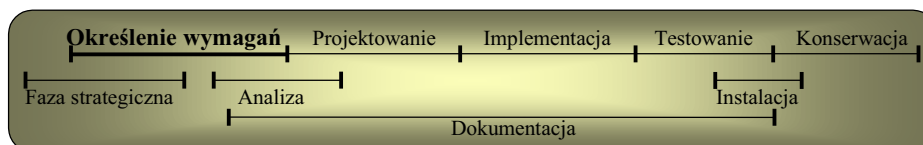
K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 1

## Określenie wymagań

Celem fazy określenia wymagań jest ustalenie wymagań klienta wobec tworzonego systemu. Dokonywana jest zamiana celów klienta na konkretne wymagania zapewniające osiągnięcie tych celów.

**Klient rzadko wie, jakie wymagania zapewnią osiągnięcie jego celów.**

Ta faza nie jest więc prostym zbieraniem wymagań, lecz procesem, w którym klient wspólnie z przedstawicielem producenta konstruuje zbiór wymagań zgodnie z postawionymi celami.



W przypadku systemu na zamówienie analitycy mają bezpośredni kontakt z przedstawicielami klienta. Faza ta wymaga dużego zaangażowania ze strony klienta, ze strony przyszłych użytkowników systemu i ekspertów w dziedzinie.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 2

## Trudność określenia wymagań

- ✦ Klient z reguły nie wie dokładnie w jaki sposób osiągnąć założone cele. Cele klienta mogą być osiągnięte na wiele sposobów.
- ✦ Duże systemy są wykorzystywane przez wielu użytkowników. Ich cele są często sprzeczne. Różni użytkownicy mogą posługiwać się inną terminologią mówiąc o tych samych problemach.
- ✦ Zleceniodawcy i użytkownicy to często inne osoby. Głos zleceniodawców może być w tej fazie decydujący, chociaż nie zawsze potrafią oni właściwie przewidzieć potrzeby przyszłych użytkowników.

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

## Poziomy ogólności opisu wymagań

- ✦ **Definicja wymagań** (*requirements definition*), to ogólny opis w języku naturalnym. Opis taki jest rezultatem wstępnych rozmów z klientem.
- ✦ **Specyfikacja wymagań** (*requirements specification*), to częściowo ustrukturalizowany zapis wykorzystujący zarówno język naturalny, jak i proste, częściowo przynajmniej sformalizowane notacje
- ✦ **Specyfikacja oprogramowania** (*software specification*), to w pełni formalny opis wymagań.

**“Formalny” nie oznacza “matematyczny”. Metody matematyczne bardzo rzadko mają zastosowanie w tej fazie (w zasadzie nigdy). Formalna specyfikacja oznacza bardzo dokładne zdekomponowanie wymagań (najlepiej w pewnym formularzu) na krótkie punkty, których interpretacja nie powinna nastręczać trudności lub prowadzić do niejednoznaczności. Formalna specyfikacja powinna stanowić podstawę dla fazy testowania.**

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 4

## Jakość opisu wymagań

**Dobry opis wymagań powinien:**

- ✦ Być kompletny oraz niesprzeczny.
- ✦ Opisywać zewnętrzne zachowanie się systemu a nie sposób jego realizacji.
- ✦ Obejmować ograniczenia przy jakich musi pracować system.
- ✦ Być łatwy w modyfikacji.
- ✦ Brać pod uwagę przyszłe możliwe zmiany wymagań wobec systemu.
- ✦ Opisywać zachowanie systemu w niepożądanych (skrajnych) sytuacjach.

**Najbardziej typowy błąd w tej fazie: koncentrowanie się na sytuacjach typowych i pomijanie wyjątków oraz przypadków granicznych. Zarówno użytkownicy jak i analitycy mają tendencję do nie zauważania sytuacji nietypowych lub skrajnych.**

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 5

## Zalecenia dla fazy definicji wymagań użytkowników

*Wg Software  
Engineering Guides*

**Wymagania użytkowników powinny być wyjaśniane poprzez krytykę i porównania z istniejącym oprogramowaniem i prototypami.**

**Powinien być uzyskany stan porozumienia pomiędzy projektantami i użytkownikami dotyczący projektowanego systemu.**

**Wiedza i doświadczenia potencjalnej organizacji podejmującej się rozwoju systemu powinny wspomóc studia nad osiągalnością systemu.**

**W wielu przypadkach dużym wspomaganie jest budowa prototypów.**

**Wymagania użytkowników powinny być**

- jasne, jednoznaczne
- weryfikowalne
- kompletne
- dokładne
- realistyczne, osiągalne



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 6

## Wymagania funkcjonalne i нефункционалне

Dwa zasadnicze rodzaje wymagań:

- ✦ **Wymagania funkcjonalne.** Opisują one funkcje (czynności, operacje) wykonywane przez system.
  - ✦ **Wymagania нефункционалне.** Opisują ograniczenia, przy zachowaniu których system powinien realizować swoje funkcje
- ✦ Wymagania powinny być zebrane w dokumencie - opisie wymagań.
  - ✦ Dokument ten powinien być podstawą do szczegółowego kontraktu między klientem a producentem oprogramowania.
  - ✦ Powinien także pozwalać na weryfikację stwierdzającą, czy wykonany system rzeczywiście spełnia postawione wymagania.
  - ✦ Powinien to być dokument zrozumiały dla obydwu stron.
  - ✦ Często producenci nie są zainteresowani w przyzwyjnym formułowaniu wymagań, które pozwoliłyby na rzeczywistą weryfikację powstałego systemu.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 7

## Części dokumentu opisującego wymagania

- ✦ **Wprowadzenie** - cele, zakres i kontekst systemu. Ta część dokumentu zawiera także wyniki fazy strategicznej (być może zmodyfikowane).
- ✦ **Opis ewolucji systemu** - opis przewidywanych zmian w stosunku do systemu.
- ✦ **Opis wymagań funkcjonalnych.**
- ✦ **Opis wymagań нефункционалnych.**
- ✦ **Model systemu** (będzie objaśniony na dalszych wykładach).
- ✦ **Słownik.**

Opis wymagań może zawierać szereg terminów niezrozumiałych dla jednej ze stron. Mogą to być terminy informatyczne (niezrozumiałe dla klienta) lub terminy dotyczące dziedziny zastosowań (niezrozumiałe dla przedstawicieli producenta). Słownik powinien wyjaśniać te terminy. Powinien także precyzować terminy niejednoznaczne i określać ich znaczenie w kontekście tego dokumentu (być może nieco węższe niż w innych kontekstach).

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 8

## Dodatki do dokumentu wymagań

- ✦ Specyfikacja wymagań funkcjonalnych.
- ✦ Specyfikacja wymagań нефункциональных.
- ✦ Wymagania sprzętowe.
- ✦ Wymagania dotyczące bazy danych.
- ✦ Indeks pomocny w wyszukiwaniu w dokumencie konkretnych informacji.

### Zasady przy tworzeniu dokumentu wymagań:

- ✦ Wymagania funkcjonalne powinny być oddzielone od wymagań нефункциональных.
- ✦ Należy rozdzielać opisy poszczególnych wymagań (w punktach lub akapitach).
- ✦ Dla każdego wymagania powinien być podany powód jego wprowadzenia cele przedsięwzięcia, których osiągnięcie jest uwarunkowane danym wymaganiem

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 9

## Wymagania funkcjonalne

Opisują funkcje (czynności, operacje) wykonywane przez system.  
Funkcje te mogą być wykonywane przy użyciu systemów zewnętrznych.

Np. funkcja “Ewidencja podatników” realizowana przez system podatkowy jest wykonywana przy udziale użytkownika, który wprowadza, usuwa i modyfikuje informacje o podatkach.

**Język naturalny jest najczęściej stosowanym sposobem opisu wymagań.  
Ma on jednak wady:**

- ✦ Niejednoznaczność języka naturalnego, która utrudnia precyzyjny opis wymagań. Może także prowadzić do niejednoznaczności w interpretacji tego samego tekstu przez różne osoby.
- ✦ Elastyczność języka naturalnego, która pozwala wyrazić te same treści na wiele sposobów. Utrudnia to wykrycie powiązanych wymagań i może prowadzić do trudności w wykryciu sprzeczności.

**Teoretycznie wady te usuwają notacje formalne, ale ich użyteczność jest bardzo niska.**

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 10

## Metody zdefiniowania wymagań użytkowników(1)

Wg Software  
Engineering Guides

### Metody rozpoznania wymagań

**Wywiady i przeglądy.** Wywiady powinny być przygotowane (pytania) i podzielone na odrębne zagadnienia. Podział powinien przykrywać całość tematu i powinny być przeprowadzone na reprezentatywnej grupie użytkowników. Wywiady powinny doprowadzić do szerokiej zgody i akceptacji projektu.

**Studia na istniejącym oprogramowaniu.** Dość często nowe oprogramowanie zastępuje stare. Studia powinny ustalić wszystkie dobre i złe strony starego oprogramowania

**Studia wymagań systemowych.** Dotyczy sytuacji, kiedy nowy system ma być częścią większego systemu.

**Studia osiągalności.** Określenie realistycznych celów systemu i metod ich osiągnięcia.

**Prototypowanie.** Zbudowanie prototypu systemu działającego w zmniejszonej skali, z uproszczonymi interfejsami.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 11

## Metody zdefiniowania wymagań użytkowników(2)

Wg Software  
Engineering Guides

### Metody specyfikacji wymagań

**Język naturalny.** Forma oczywista, najczęściej stosowana. Wadą jest niejednoznaczność wielu wyrazów i sformułowań oraz trudność szybkiego znalezienia wymaganej informacji

**Formalizm matematyczny.** Stosuje się rzadko, dla dobrze zdefiniowanych problemów raczej o małej skali. Mało przydatny na wczesnych etapach projektu

**Język naturalny strukturalny.** Język naturalny z ograniczonym słownictwem i składnią. Poszczególne tematy i zagadnienia wyspecyfikowane w punktach i podpunktach.

**Tablice, formularze.** Wyspecyfikowanie wymagań w postaci (zwykle dwuwymiarowych) tablic, kojarzących różne aspekty (np. tablica ustalająca zależność pomiędzy typem użytkownika i rodzajem usługi).

**Diagramy blokowe** tradycyjna forma graficzna pokazująca wymagany cykl przetwarzania.

**Diagramy kontekstowe** ukazują system w postaci jednego bloku oraz jego powiązania z otoczeniem, wejściem i wyjściem.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 12

## Formularze do zapisu wymagań funkcjonalnych

W formularzach zapis jest podzielony na konkretne pola, co w wielu przypadkach pozwala na łatwe stwierdzenie kompletności opisu oraz na jednoznaczną interpretację.

### Przykład jednej wypełnionej tabeli wg przyjętego formularza:

<b>Nazwa funkcji</b>	<i>Edycja dochodów pracownika</i>
<b>Opis</b>	Funkcja pozwala edytować łączne dochody podatnika uzyskane w danym roku.
<b>Dane wejściowe</b>	Informacje o dochodach pracowników uzyskane uzyskanych z różnych źródeł: kwoty przychodów, koszty uzyskania przychodów oraz zapłaconych zaliczek na poczet podatku dochodowego. Informacje o dokumentach opisujących dochody z poszczególnych źródeł.
<b>Źródło danych wejściowych</b>	Dokumenty oraz informacje dostarczone przez podatnika. Dane wpisywane przez pracownika firmy podatkowej.
<b>Wynik</b>	
<b>Warunek wstępny</b>	Kwota dochodu = kwota przychodu - kwota kosztów (zarówno dla konkretnych dochodów, jak i dla łącznych dochodów podatnika). Łączne kwoty przychodów, kosztów uzyskania dochodów oraz zapłaconych zaliczek są sumarycznymi kwotami dla dochodów z poszczególnych źródeł.
<b>Warunek końcowy</b>	Jak wyżej.
<b>Efekty uboczne</b>	Uaktualnienie podstawy opodatkowania.
<b>Powód</b>	Funkcja pomaga przyspieszyć obsługę klientów oraz zmniejszyć ryzyko popełnienia błędów.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 13

## Porządkowanie wymagań

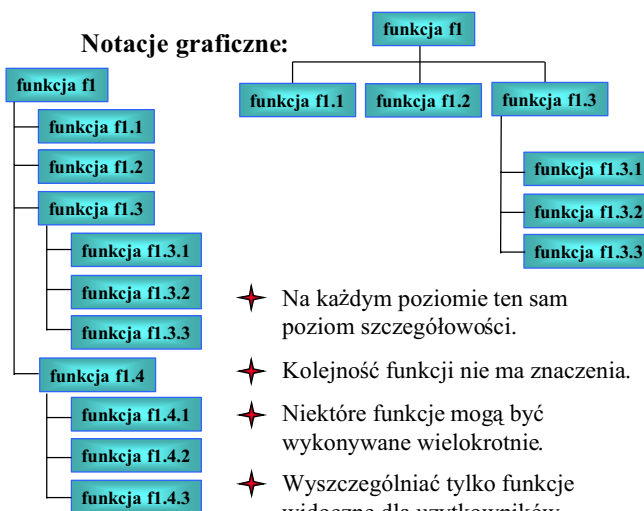
### Hierarchia wymagań funkcjonalnych:

Z reguły funkcje można rozbić na podfunkcje.

#### Format tekstowy:

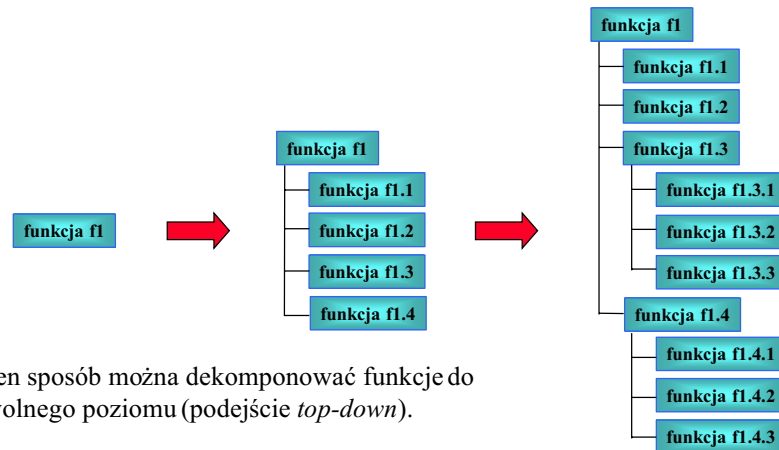
Funkcja nadrzędna f1  
 funkcja f1.1  
 funkcja f1.2  
 funkcja f1.3  
     funkcja f1.3.1  
     funkcja f1.3.2  
     funkcja f1.3.3  
 funkcja f1.4  
     funkcja f1.4.1  
     funkcja f1.4.2  
     funkcja f1.4.3

#### Notacje graficzne:



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 14

## Zstępujące konstruowanie hierarchii funkcji



- ✦ W ten sposób można dekomponować funkcje do dowolnego poziomu (podejście *top-down*).
- ✦ Możliwe jest również podejście odwrotne (*bottom-up*), kiedy składamy kilka funkcji bardziej elementarnych w jedną funkcję bardziej ogólną. Możliwa jest również technika mieszana.

## Przykład: program podatkowy

Program ułatwia przygotowanie formularzy zeznań podatkowych (PIT-ów) oraz przechowanie informacji o źródłach przychodów i ulg. Zeznanie może być tworzone przez pojedynczego podatnika lub małżeństwa. Zeznanie mogą obejmować informacje o rocznych przychodach (w przypadku małżeństwa z podziałem na przychody męża i żony) oraz o ulgach podatkowych. Przychody podzielone są na klasy ze względu na źródło uzyskania, np. pozarolnicza działalność gospodarcza, wolny zawód, ... W ramach danej klasy przychodów podatnik mógł osiągnąć szereg przychodów z różnych źródeł.

Wszystkie przychody opisane są przez kwotę przychodu, kwotę kosztów, kwotę zapłaconych zaliczek oraz kwotę dochodu. Powyższe informacje pozwalają obliczyć należny podatek oraz kwotę do zapłaty lub zwrotu. Zeznanie zawiera także informację o podatniku oraz adres Urzędu Skarbowego.

System pozwala wydrukować wzorzec zeznania zawierający wszystkie informacje, jakie podatnik musi umieścić w formularzu. Zeznanie można zabezpieczyć przed dalszymi zmianami (po złożeniu w Urzędzie Skarbowym). System pozwala na tworzenie listy podatników oraz urzędów skarbowych, które mogą być pomocne przy tworzeniu nowego zeznania. Przechowuje także informację o wszystkich złożonych zeznaniach.



## Program podatkowy: hierarchia funkcji



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 17

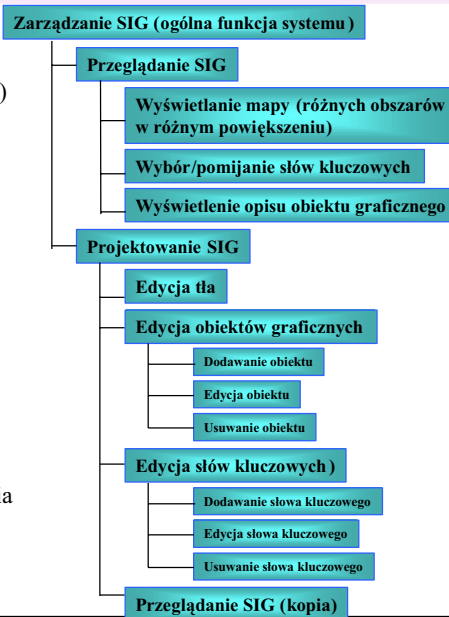
## Przykład: system informacji geograficznej - SIG

SIG jest rodzajem mapy komputerowej, składającej się z tła (np. mapy fizycznej) oraz szeregu obiektów graficznych umieszczonych na tym tle.

Obiekt może być punktem (budynek, firma), linią (rzeka, kolej) lub obszarem (park, osiedle).

Każdy obiekt ma swoją nazwę i ewentualny opis, który może być wyświetlony na żądanie użytkownika. Obiekt można opisać szeregiem słów kluczowych.

Użytkownik ma możliwość wyświetlenia tylko tych obiektów, które opisano słowami kluczowymi.



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 18

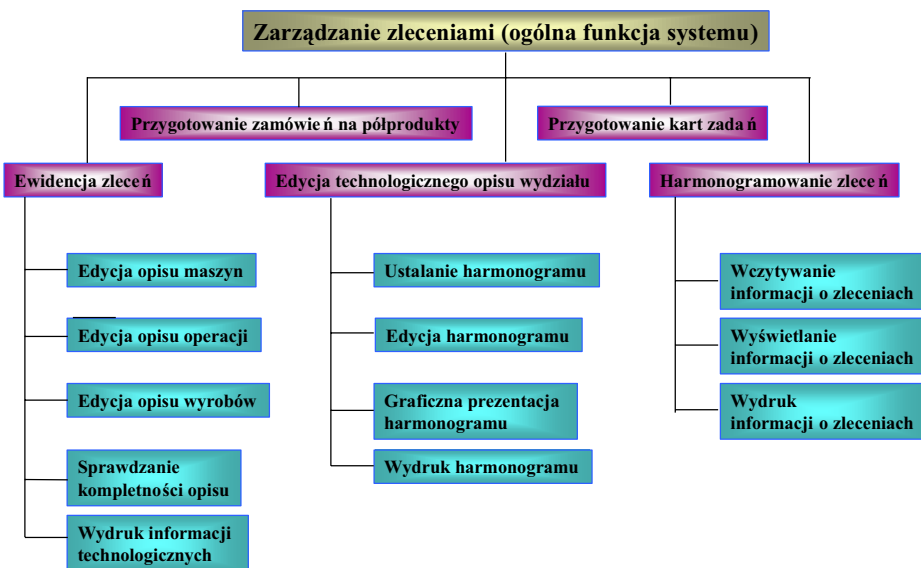
## Przykład: system harmonogramowania zleceń

Zlecenia dla wydziału przygotowywane są przez dział marketingu. Zlecenie oznacza konieczność wyprodukowania konkretnej ilości pewnego wyrobu przed upływem konkretnego terminu. Czasami może być określony najwcześniejszy pożądaný termin realizacji. Dział marketingu wykorzystuje własny system informatyczny, w którym między innymi umieszczane są informacje o zleceniach, požadane jest więc, aby system harmonogramowania zleceń automatycznie odczytywał te informacje.

Wyprodukowanie danego wyrobu (realizacja zlecenia) wymaga wykonania ciągu operacji. Pewne operacje mogą być wykonywane tylko na jednym urządzeniu; w innych przypadkach możliwe jest wykorzystanie kilku maszyn, które mogą różnić się efektywnością pracy. Po wykonaniu pewnych operacji może być konieczna przerwa, zanim rozpocznie się realizacja następnych; z drugiej strony taka przerwa może trwać przez ograniczony czas. Przystawienie maszyn na inne operacje może wymagać czasu. Co pewien czas (np. raz na miesiąc) ustalany jest harmonogram niezrealizowanych zleceń.

System powinien opracować harmonogramy w formie łatwej do wykorzystania przez kadre kierowniczą wydziału oraz przygotowywać zamówienia do magazynu na półprodukty. Zlecenia wykonane są usuwane ze zbioru niezrealizowanych zleceń.

## System harmonogramowania zleceń: funkcje



## Diagramy przypadków użycia

### Opis funkcji systemu z punktu widzenia jego użytkowników.

Opis wymagań poszczególnych użytkowników jest prostszy niż opis wszystkich wymagań wobec systemu.

- Klasy użytkowników:**
- sekretarka
  - projektant
  - osoba przeglądająca mapę

Metoda modeluje **aktorów**, a nie konkretne **osoby**. Jedna osoba może pełnić rolę wielu aktorów; np. być jednocześnie projektantem i osobą przeglądającą mapę. I odwrotnie, jeden aktor może odpowiadać wielu osobom, np. sekretarka.

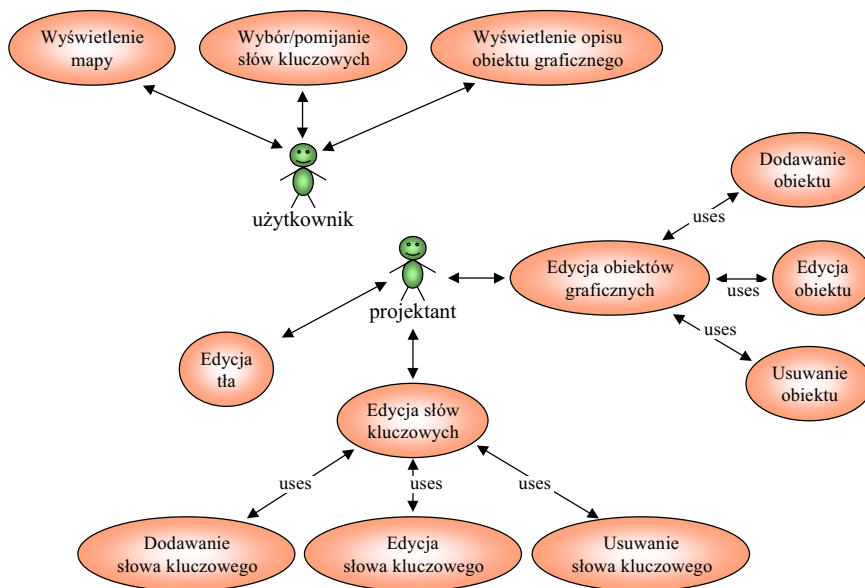
### Identyfikacja funkcji dla poszczególnych użytkowników.

Przeprowadzając wywiad z konkretną osobą należy koncentrować się na funkcjach istotnych z punktu widzenia roli (ról) odgrywanych przez tę osobę.

Metoda przypadków użycia nie jest sprzeczna z hierarchiczną dekompozycją funkcji.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 21

## Diagram przypadków użycia dla SIG



K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 22

## Wymagania niefunkcjonalne

Opisują ograniczenia, przy których system ma realizować swoje funkcje.

- ★ **Wymagania dotyczące produktu.**  
Np. musi istnieć możliwość przeglądania systemu informacji geograficznej wyłącznie za pomocą klawiatury.
- ★ **Wymagania dotyczące procesu.**  
Np. proces realizacji harmonogramowania zleceń musi być zgodny ze standardem opisanym w dokumencie XXXA/96.
- ★ **Wymagania zewnętrzne.**  
Np. system harmonogramowania musi współpracować z bazą danych systemu komputerowego działu marketingu opisaną w dokumencie YYYB/95.  
Niedopuszczalne są jakiegokolwiek zmiany w strukturze tej bazy.

Dobrze opisane wymagania niefunkcjonalne powinny być **weryfikowalne**, tj. powinna istnieć możliwość sprawdzenia czy system je rzeczywiście spełnia. Np. wymagania “system ma być łatwy w obsłudze”, “system ma być niezawodny” nie są weryfikowalne.

## Tabelaryczny zapis rozważanych rozwiązań

<i>Cecha</i>	<i>Miary</i>
Wydajność	Liczba transakcji obsługanych w ciągu sekundy Czas odpowiedzi Szybkość odświeżania ekranu
Rozmiar	Wymagana pamięć RAM Wymagana pamięć dyskowa
Łatwość użytkowania	Czas niezbędny dla przeszkolenia użytkowników Liczba stron dokumentacji
Niezawodność	Prawdopodobieństwo błędnego wykonania podczas realizacji transakcji Częstotliwość występowania błędnych wykonań Średni czas pomiędzy błędnymi wykonaniami Dostępność (procent czasu w którym system jest dostępny)
Odporność ( <i>robustness</i> )	Czas restartu po awarii systemu Prawdopodobieństwo zniszczenia danych w przypadku awarii
Przenaszalność	Procent kodu zależnego od platformy docelowej Liczba platform docelowych Koszt przeniesienia na nową platformę

## Czynniki, które należy uwzględnić przy konstruowaniu wymagań (1)

Wg Software Engineering Guides

- ✦ **Możliwości systemu** (*capability requirements*): co użytkownicy chcą od systemu. Zestaw funkcji, które ma wykonywać system, uporządkowany hierarchicznie.
- ✦ **Objętość** (*capacity*): Ilu użytkowników będzie pracować jednocześnie? Ile terminali ma być podłączone do systemu? Ile czujników będzie kontrolowanych jednocześnie? Ile danych będzie przechowywane?
- ✦ **Szybkość** (*speed*): Jak długo może trwać operacja lub sekwencja operacji? Liczba operacji na jednostkę czasu. Średni czas niezbędny dla jednej operacji.
- ✦ **Dokładność** (*accuracy*): Określenie stopnia precyzji pomiarów lub przetwarzania. Określenie wymaganej dokładności wyników. Zastąpienie wyników ilościowych jakościowymi lub odwrotnie.
- ✦ **Ograniczenia** (*constraint requirements*): ograniczenia na interfejsy, jakość, skalę czasową, sprzęt, oprogramowanie, skalowalność, ...

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 25

## Czynniki, które należy uwzględnić przy konstruowaniu wymagań (2)

Wg Software Engineering Guides

- ✦ **Interfejsy komunikacyjne** (*communication interfaces*): sieć, protokoły, wydajność sieci, poziom abstrakcji protokołów komunikacyjnych ...
- ✦ **Interfejsy sprzętowe** (*hardware interfaces*): specyfikacja wszystkich elementów sprzętowych, które będą składały się na system, fizyczne ograniczenia (rozmiar, waga), wydajność (szybkość, RAM, dysk, inne pamięci), wymagania co do powierzchni lokalowych, wymagania co do wilgotności, temperatury i ciśnienia, ...
- ✦ **Interfejsy oprogramowania** (*software interfaces*): Określenie kompatybilności z innym oprogramowaniem, określenie systemów operacyjnych, języków programowania, kompilatorów, edytorów, systemów zarządzania bazą danych,...
- ✦ **Interakcja człowiek-maszyna** (*human-computer interaction*): Wszystkie aspekty interfejsu użytkownika, rodzaj języka interakcji (język komend, system menu, ikony), rodzaj sprzętu (monitor, mysz, klawiatura), określenie formatów (układu raportów i ich zawartości), określenie komunikatów dla użytkowników (język, forma), pomocy, komunikatów o błędach, ...

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 26

## Czynniki, które należy uwzględnić przy konstruowaniu wymagań (3)

Wg. *Software Engineering Guides*

- ✦ **Adaptowalność** (*adaptability*): Określenie w jaki sposób będzie organizowana reakcja na zmiany wymagań: dodanie nowej komendy, dodanie nowego okna interakcji, ...
- ✦ **Bezpieczeństwo** (*security*): założenia co do poufności, prywatności, integralności, odporności na hakerów, wirusy, wandalizm, sabotaż, ...
- ✦ **Odporność na awarie** (*safety*): konsekwencje błędów w oprogramowaniu, przerwy w zasilaniu, kopii zabezpieczających, częstotliwości składowania, dziennika zmian,...
- ✦ **Standardy** (*standards*): Określenie dokumentów standardyzacyjnych, które mają zastosowanie do systemu: formaty plików, normy czcionek, polonizacja, standardy procesów i produktów, ...
- ✦ **Zasoby** (*resources*): Określenie ograniczeń finansowych, ludzkich i materiałowych.
- ✦ **Skala czasowa** (*timescales*): ograniczenia na czas wykonania systemu, czas szkolenia, wdrażania, ...

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 27

## Kluczowe czynniki sukcesu

- ✦ Zaangażowanie właściwych osób ze strony klienta
- ✦ Pełne rozpoznanie wymagań, wykrycie sytuacji szczególnych i nietypowych. Typowy błąd popełniany w tej fazie to koncentrowanie się na sytuacjach typowych.
- ✦ Sprawdzenie kompletności i spójności wymagań. Przed przystąpieniem do dalszych prac, wymagania powinny być przejrzane pod kątem ich kompletności i spójności.
- ✦ Określenie wymagań niefunkcjonalnych w sposób umożliwiający ich weryfikację.

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 28

## Podstawowe rezultaty fazy określania wymagań

✦ Dokument opisujący wymagania składający się z:

- Wprowadzenia opisującego cele, zakres i kontekst systemu
- Opisu ewolucji systemu, tj. przewidywanych zmian wymagań wobec systemu
- Opisu wymagań funkcjonalnych
- Opisu wymagań нефункциональных
- Modelu systemu
- Słownika

oraz następujących dodatków:

- Specyfikacji wymagań funkcjonalnych
- Specyfikacji wymagań нефункциональных
- Opisu wymagań sprzętowych
- Opisu wymagań dotyczących bazy danych
- Indeksu

✦ Plan testów

## Dokument wymagań użytkownika

*Wg Software  
Engineering Guides*

### Styl

**Jasność:** jednoznaczne sformułowania, zrozumiałe dla użytkowników i projektantów.  
Strukturalna organizacja dokumentu  
**Spójność:** brak konfliktów w wymaganiach.  
**Modyfikowalność:** wszystkie wymagania są sformułowane w jasnych punktach, które mogą być wyizolowane z kontekstu i zastąpione przez inne.

### Ewolucja

Możliwość dodawania nowych wymagań, możliwość istotnej ich modyfikacji

### Odpowiedzialność

Określenie kto jest odpowiedzialny za sformułowanie danego wymagania i istotne w przypadku modyfikacji.

### Medium

Dokument papierowy lub elektroniczny. Staranne kontrolowanie wersji dokumentu

# Dokument wymagań użytkownika - zawartość

Wg Software  
Engineering Guides

Informacje  
organizacyjne

Zalecany spis treści



**a - Streszczenie**  
**b - Spis treści**  
**c - Status dokumentu** (roboczy, 1-sza faza, zatwierdzony, ...)  
**d - Zmiany w stosunku do wersji poprzedniej**

## 1. Wstęp

- 1.1. Cel
- 1.2. Zakres
- 1.3. Definicje, akronimy i skróty
- 1.4. Referencje, odsyłacze do innych dokumentów
- 1.5. Krótki przegląd

## 2. Ogólny opis

- 2.1. Walory użytkowe i przydatność projektowanego systemu
- 2.2. Ogólne możliwości projektowanego systemu
- 2.3. Ogólne ograniczenia
- 2.4. Charakterystyka użytkowników
- 2.5. Środowisko operacyjne
- 2.6. Założenia i zależności

## 3. Specyficzne wymagania

- 3.1. Wymagania co do możliwości systemu
- 3.2. Przyjęte lub narzucone ograniczenia

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 31

# Narzędzia dla fazy określenia wymagań

Mogą być pomocne różnorodne narzędzia CASE, szczególnie do rozrysowania i modyfikacji różnych diagramów, np. diagramów przypadków użycia.

Produkcja dokumentów może (powinna) odbywać się przy pomocy popularnych edytorów takich jak WORD. Wykorzystuje się także arkusze kalkulacyjne, itp.

## Możliwości narzędzi:

- ☞ wstawienie nowego wymagania
- ☞ modyfikacja istniejącego wymagania
- ☞ usunięcie wymagania
- ☞ zapamiętanie atrybutów (tekstowych i znakowych)
- ☞ wyszukiwanie na podstawie atrybutów tekstowych
- ☞ wyszukiwanie w pełnym tekście na podstawie zadanego ciągu znaków
- ☞ wzajemne powiązania (cross-referencing)
- ☞ zapamiętanie historii aktualizacji
- ☞ sterowanie i kontrola dostępu
- ☞ wyświetlenie
- ☞ drukowanie w różnych formatach

K.Subieta. Wytwarzanie, integracja i testowanie SI, Wykład 3, Folia 32



## Określenie wymagań jakości dla następnych faz projektu

*Wg Software  
Engineering Guides*

Plan zapewnienia jakości powinien uwzględniać kontrolowanie następujących cech:

- ◆ zarządzanie projektem
- ◆ dokumentacja projektu
- ◆ standardy, praktyka, konwencje, miary
- ◆ recenzje, raporty i audyty
- ◆ aktywności w zakresie testowania
- ◆ techniki sygnalizowania problemów i akcji korygujących
- ◆ stosowane narzędzia, techniki i metody
- ◆ zarządzanie kodem i nośnikami
- ◆ zarządzanie dostawami
- ◆ obsługa/przechowywanie zapisów o wykonanych pracach
- ◆ szkolenie uczestników projektu
- ◆ zarządzanie ryzykiem