

Inżynieria oprogramowania

Wykład 4: Inżynieria wymagań dla systemów informatycznych

Marek Krętowski
pokój 206
e-mail: mkret@ii.pb.bialystok.pl
http://aragorn.pb.bialystok.pl/~mkret

Wersja 1.11 st. podpłomowe

Inżynieria wymagań (ang. requirements engineering)

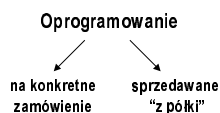
- **Inżynieria wymagań (IW)** reprezentuje całość działań związanych z pozyskiwaniem, reprezentowaniem, analizą i zarządzaniem wymaganiami w ramach kontekstu wyznaczonego przez cykl życia systemu inf.
- W większości organizacji zajmujących się budową systemów jakość procesów IW jest niska; wynika to często z lekceważenia wszelkiej działalności różnej od kodowania, a z drugiej strony wymuszana jest często przez krótkowzroczne oczekiwania klientów, którzy gotowi są płacić jedynie za fizyczne oprogramowanie
- Błędy popełnione podczas określania wymagań mogą być bardzo kosztowne; przyjmuje się, że rosną one zgodnie z zasadą 1:N w miarę rozwoju projektu (przykładowo: wykrycie poważnego błędu dopiero na etapie projektowanie jest N - razy kosztowniejsze niż odpowiednia analiza wymagań, ale już wykrycie tego samego błędu na etapie programowania może być nawet N*N kosztowniejsze)
- W ocenie niektórych specjalistów, IW jest najtrudniejszą częścią projektu związanego z wytworzeniem systemu inf. i zwłaszcza w przypadku dużych projektów od jej poprawności i efektywności często zależy powodzenie całych projektów

Inżynieria oprogramowania (Wyk. 4)

Slajd 2 z 20

Określanie wymagań

- Zamiana celów klienta na konkretne wymagania wobec tworzonego systemu zapewniające osiągnięcie tych celów
- Klient rzadko dokładnie wie jakie wymagania zapewnia określenie wyznaczonych celów
- Określanie wymagań należy rozumieć jako proces, w którym klient wspólnie z analitykami konstruuje zbiór wymagań zgodny z postawionymi celami



- Oprogramowanie na zamówienie - bezpośredni kontakt analityków z przyszłymi użytkownikami; niezbędne duże zaangażowanie ze strony klienta
- Oprogramowanie rynkowe - korzystny jak najszerszy kontakt z potencjalnymi klientami oraz ekspertami z danej dziedziny

Inżynieria oprogramowania (Wyk. 4)

Slajd 3 z 20

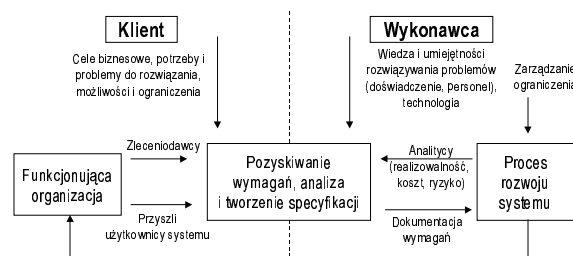
Trudności w określaniu wymagań

- Klient z reguły nie wie dokładnie w jaki sposób osiągnąć założone cele; cele klienta zwykle mogą być osiągnięte na wiele sposobów
- Duże systemy są wykorzystywane przez wielu użytkowników, których cele są często sprzeczne; różni użytkownicy mogą posługiwać się inną terminologią mówiąc o tych samych problemach
- Zleceniodawcy i użytkownicy są to często inne osoby; głos zleceniodawców jest oczywiście decydujący, ale nie zawsze potrafią oni właściwie przewidzieć potrzeby przyszłych użytkowników
- Najtrudniej wykryć wymagania, których użytkownicy nie są świadomi (prowadzi to do powstawania luk w specyfikacji)
- Często pracownicy obawiają się zmian, które spowoduje wprowadzenie systemu i postzegają analityków jako nieprzyjaciół
- Czasami próbuje się zatłoczyć przy okazji realizacji określonego przedsięwzięcia inne sprawy niezwiązane z projektem

Inżynieria oprogramowania (Wyk. 4)

Slajd 4 z 20

Iteracyjność pozyskiwania wymagań



Klient we własnym interesie powinien być zainteresowany jakością określonych wymagań i ich ewentualną modyfikacją, gdy ulegają one zmianie w czasie

Inżynieria oprogramowania (Wyk. 4)

Slajd 5 z 20

Zaangażowanie po stronie klienta

- **Udziałowcy (ang. stakeholders)** - wszystkie osoby, które zainteresowane są powodzeniem (lub porażką) przedsięwzięcia; użytkownicy, menadżerowie, zarząd decydujący o budżecie; na różnych poziomach organizacji
- W dużych projektach tworzy się często komitet sterujący, który zarządza projektem po stronie klienta:
 - przedstawiciele kierownictwa firmy
 - służby finansowe - kontrola od strony budżetu projektu
 - kierownicy zainteresowanych działów
 - przedstawiciele użytkowników końcowych
- Użytkownicy występują w różnych rolach podczas trwania projektu
 - źródła wiedzy o rozwiązywanym problemie podczas zbierania wymagań
 - reprezentanci w Komitecie projektu organizacji
 - oceniający prototypy
 - testujący zaproponowane rozwiązania
 - szkoleni podczas wdrożenia
 - końcowi użytkownicy nowego systemu

Inżynieria oprogramowania (Wyk. 4)

Slajd 6 z 20

Metody zbierania informacji

- **Lektura**
 - szczególnie użyteczna we wstępnej fazie, pozwala na poznanie organizacji i lepsze przygotowanie się do innych technik
- **Rozmowy - wywiady**
 - podstawowa forma; umożliwia szybką reakcję na uzyskiwane informacje oraz pozwala na dokładne rozpracowanie interesujących zagadnień;
 - może być jednak czasochłonna i kosztowna
- **Obserwacja**
 - bezpośrednia weryfikacja informacji z innych źródeł; umożliwia sprawdzenie jak naprawdę przebiegają procesy a nie tylko opieranie się np. na relacjach prac.
- **Próbkowanie dokumentów**
- **Ankietowanie**
 - uzyskanie informacji od znacznej liczby osób, w sposób, który pozwala na analizę statystyczną

Inżynieria oprogramowania (Wyk. 4)

Slajd 7 z 20

Lektura

Najczęściej prowadzi do poznania aktualnej sytuacji (istniejący system informacyjny), ale nie do określenia wymagań wobec nowego systemu

Potencjalne źródła wiedzy o organizacji i jej celach biznesowych:

- opis struktury organizacyjnej
- opisy stanowisk pracy i/lub obowiązków pracowników
- różne dokumenty wewnętrzne
- raporty z wcześniejszych analiz
- dokumentacja funkcjonującego oprogramowania
- akty prawne(!): ustawy, rozporządzenia, przepisy wykonawcze, ...
- plany inwestycyjne, projekty zmian, ... (nowe wymagania!)

Zagrożenie:

- dokumenty mogą być nieaktualne lub nie odzwierciedlać rzeczywistych procesów zachodzących w organizacji

Inżynieria oprogramowania (Wyk. 4)

Slajd 8 z 20

Rozmowy z pracownikami klienta - zalecenia

Przed rozmową:

- zaczynać zawsze od kierownictwa i dopiero później schodzić na niższe szczeble
- umawiać się na rozmowy
- określić temat, cel i czas rozmowy
- być przygotowanym (poznać co najmniej trochę dyskutowany temat, ustalić co chcemy się dowiedzieć, sporządzić listę pytań, ...)

Sama rozmowa:

- odpowiednie zachowanie (punktualność, nie zapomnieć się przedstawić, starać się stworzyć miłą atmosferę, okazywać szacunek rozmówcy)

- Pytania otwarte (np. Dlaczego?) zadawane pojedynczo, modyfikowane w razie potrzeby
 - kieruj rozmową, ale unikaj przerywania
 - używaj terminologii rozmowy, nie udawaj "eksperta od wszystkiego"
 - proś o wyjaśnienia w przypadku niejasności oraz kopie dokumentów o których mówi rozmówca
 - staraj się odróżnić fakty od opinii
 - nie wyrażaj własnych opinii
- Po rozmowie:
- udokumentowanie i autoryzacja rozmowy
 - ustalenie listy otwartych spraw
 - weryfikacja uzyskanych danych, zwłaszcza wyświadczenie sprzeczności

Inżynieria oprogramowania (Wyk. 4)

Slajd 9 z 20

Obserwacja

Rodzaje:

- formalna (stosowanie różnych technik pomiarowych)
- nieformalna

Cele obserwacji nieformalnej:

- uzyskanie ogólnego obrazu
- odkrycie potencjalnych ekspertów (osób kompetentnych i otwartych)
- przebieg przetwarzania związanego z danym dokumentem (od początku do końca)
- zebranie danych ilościowych
- wykrycie związków
- zidentyfikowanie możliwych zakłóceń oraz reakcji na sytuacje nietypowe
- zmiany w obciążeniu pracą
- magazyny danych - kartoteki

Plusy:

- wysoka wiarygodność uzyskanych w ten sposób informacji
- możliwość wykrycia sytuacji nietypowych i wymagań, których użytkownicy nie są świadomi

Zagrożenia:

- większość ludzi nie lubi być obserwowanym lub zachowuje się inaczej niż zwykle
- problemy etyczne z danymi personalnymi

Inżynieria oprogramowania (Wyk. 4)

Slajd 10 z 20

Próbkowanie dokumentów

Dwa tryby pracy:

- analiza jakościowa - gromadzenie pustych oraz wypełnionych dokumentów podczas wywiadów i obserwacji
- analiza ilościowa - statystyczna analiza zbiorów dokumentów => wykrywanie wzorów i prawidłowości

Celem jest ustalenie przetwarzanych danych w poszczególnych procesach (zarówno "ręcznie" jak i przy użyciu istniejących systemów):

- rodzaje i rozmiary danych
- ilość i częstotliwość napływu danych
- tendencje zmian, wartości średnie i ekstremalne
- stopień zautomatyzowania organizacji
- podział czasu pomiędzy poszczególne czynności
- ilości błędów w dokumentach

W przypadku drastycznej zmiany systemu, aktualnie przetwarzane dokumenty mogą znacznie odbiegać od proponowanych

Inżynieria oprogramowania (Wyk. 4)

Slajd 11 z 20

Ankietowanie

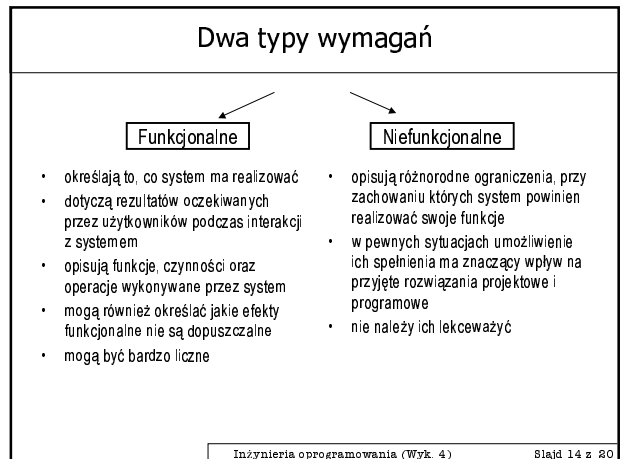
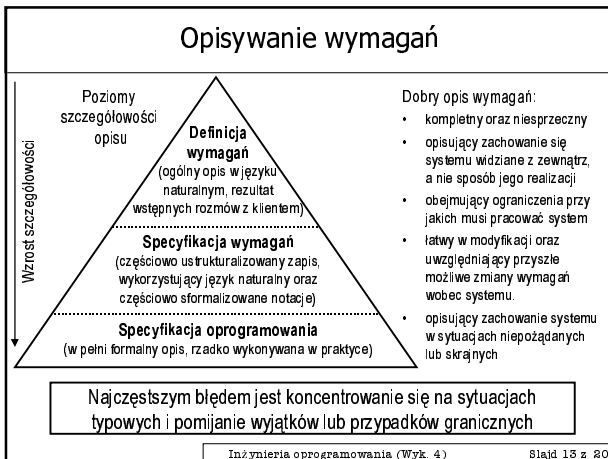
- Ankiety służą do zbierania zarówno opinii jak i faktów
- Sensowne tylko gdy jest to naprawdę uzasadnione
- Wymagają dobrych stosunków z ankietowanymi (odpowiednia motywacja, ankietowani muszą mieć też czas na ich wypełnienie)
- Efektywne w sytuacji zbierania informacji z rozproszonych geograficznie źródeł
- Dobrze zaprojektowany kwestionariusz może być analizowany przez odpowiedni system wspomagający

Zalecenia do ankietowania:

- najpierw sprecyzować, dlaczego wysyłamy ankiety i czego chcemy się dowiedzieć
- następnie przygotować pytania (najlepiej proste pytania z wyborem odpowiedzi, w przypadku subiektywnych ocen wykorzystanie skali)
- nie pytać o sprawy oczywiste
- pismo przewodnie (cel, data zwrotu)
- warto przetestować na współpracownikach
- nie zaniżać kosztów ankietowania

Inżynieria oprogramowania (Wyk. 4)

Slajd 12 z 20



Wymagania niefunkcjonalne

Rodzaje wymagań niefunkcjonalnych:

- Dotyczące **produktu** (np. wynikające ze specyfiki sprzętu wykorzystywanego przez klienta)
- Dotyczące **procesu** (np. zgodność z systemem prawnym lub z narzuconymi standardami)
- **Zewnętrzne** (określają zasady współpracy z innymi systemami)

Przykłady wymagań:

- **Rozmiar:** liczba terminali i jednocześnie pracujących użytkowników; liczba kontrolowanych urządzeń (czujników); rozmiar przechowywanych danych

- **Szybkość działania:** średni (maksymalny) czas operacji lub ich sekwencji; liczba operacji na jednostkę czasu
- **Dokładność:** stopień precyzji pomiarów lub przetwarzania (wymagana dokładność wyników)
- **Ograniczenia:**
 - interfejsy komunikacyjne - sieć, protokoły, wydajność sieci;
 - jakościowe
 - sprzętowe - istniejące elementy, fizyczne ograniczenia, wydajność, odporność (wilgotność, temperatura, ...)
 - interfejsy oprogramowania - zgodność z sys. oper., innym oprogramowaniem, ...
 - związane z interakcją człowiek-komputer (typ interfejsu użytkownika, ...)

Inżynieria oprogramowania (Wyk. 4) Slajd 16 z 20

Wymagania niefunkcjonalne (2)

- **Adaptowalność:** reakcja na zmiany wymagań
- **Bezpieczeństwo:** sposób zapewnienia poufności (systemy identyfikacji użytkowników i określania praw dostępu), wykrywanie nieuprawnionego dostępu, odporność na ataki z zewnątrz, wirusy, ...
- **Odporność na awarie:** konsekwencje błędów w progr. lub innych zdarzeń (przerwy w zasilaniu), zapewnienie integralności danych, kopie zabezpieczające, dzienniki zmian, ...
- **Standardy:** formaty plików, polonizacja, standardy procesów i produktów, ...
- **Zasoby:** Określenie ograniczeń finansowych, ludzkich i materiałowych.
- **Skala czasowa:** ograniczenia na czas wykonania systemu, czas szkolenia, wdrażania

- Należy dążyć do umożliwienia weryfikacji wymagań; powinna istnieć możliwość sprawdzenia lub zmierzenia czy system rzeczywiście spełnia oczekiwania
- Przykładowo wymagania: „system ma być łatwy w obsłudze”, „system ma być niezawodny”, „system ma być dostatecznie szybki”, itd. nie są praktycznie weryfikowalne
- Konieczne jest posługiwanie się wielkościami mierzalnymi w celu określania wymagań tego typu

Inżynieria oprogramowania (Wyk. 4) Slajd 17 z 20

Przykłady metryk dla wymagań niefunkcjonalnych

Cecha	Metryka
Wydajność	Liczba transakcji obsługanych w ciągu sekundy Czas odpowiedzi
Rozmiar	Liczba rekordów w bazie danych Wymagana przestrzeń dyskowa
Łatwość użytkowania	Czas niezbędny dla przeszkolenia użytkowników Rozmiar dokumentacji
Niezawodność i odporność	Prawdopodobieństwo błędu podczas realizacji transakcji Średni czas pomiędzy błędnymi wykonaniami Dostępność (procent czasu w którym system jest dostępny) 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

Inżynieria oprogramowania (Wyk. 4) Slajd 18 z 20

Dokument opisu wymagań

- Wszystkie wymagania uzgodnione z klientem powinny być zebrane w dokumencie opisu wymagań
- Dokument ten jest zwykle podstawą szczegółowego kontraktu między klientem, a producentem oprogramowania
- Dokument opisu wymagań musi być zrozumiały dla obydwu stron
- Powinien pozwalać na weryfikację umożliwiającą stwierdzenie, czy wykonany system rzeczywiście spełnia postawione wymagania (zarówno od strony funkcjonalnej jak i niefunkcjonalnej)
- Firmy, które zajmują się produkcją oprogramowania często nie są zainteresowane w precyzyjnym formułowaniu wymagań pozwalających na rzeczywistą weryfikację stworzonego systemu; może to prowadzić do konfliktów

Inżynieria oprogramowania (Wyk. 4) Slajd 19 z 20