

Inżynieria Oprogramowania inżynieria wymagań

Dr hab. inż. Ilona Bluemke

Plan wykładu

- Faza strategiczna
- Niepowodzenia projektów
- Koszty usuwania błędów
- Wymagania

Faza strategiczna

Wykonywana zanim zostanie podjęta decyzja o realizacji dalszych etapów przedsięwzięcia

- oprogramowanie zamawiane - negocjacje i/lub przetarg
- oprogramowanie rynkowe - rozważana i planowana produkcja nowego programu czy nowej wersji

Studium wykonalności (feasibility study) -1

czynności:

- rozmowy, wywiady z przedstawicielami klienta
- określenie celów przedsięwzięcia z punktu widzenia klienta
- określenie zakresu oraz kontekstu przedsięwzięcia
- określenie wymagań - ogólne, zgrubna analiza i projekt systemu
- propozycja kilku możliwych sposobów realizacji

Studium wykonalności (feasibility study) - 2

- oszacowanie kosztów
- analiza rozwiązań
- prezentacja wyników, korekcja
- określenie wstępnego harmonogramu oraz przedstawienie struktury zespołu realizującego
- określenie standardów zgodnie z którymi będzie realizacja

Decyzje strategiczne

- wybór modelu, zgodnie z którym będzie realizowane przedsięwzięcie
- wybór technik stosowanych w analizie
- wybór środowiska implementacji
- wybór narzędzia CASE
- określenie stopnia wykorzystania gotowych komponentów
- podjęcie decyzji o współpracy z innymi producentami i/lub zatrudnieniu ekspertów zewnętrznych

faza strategiczna

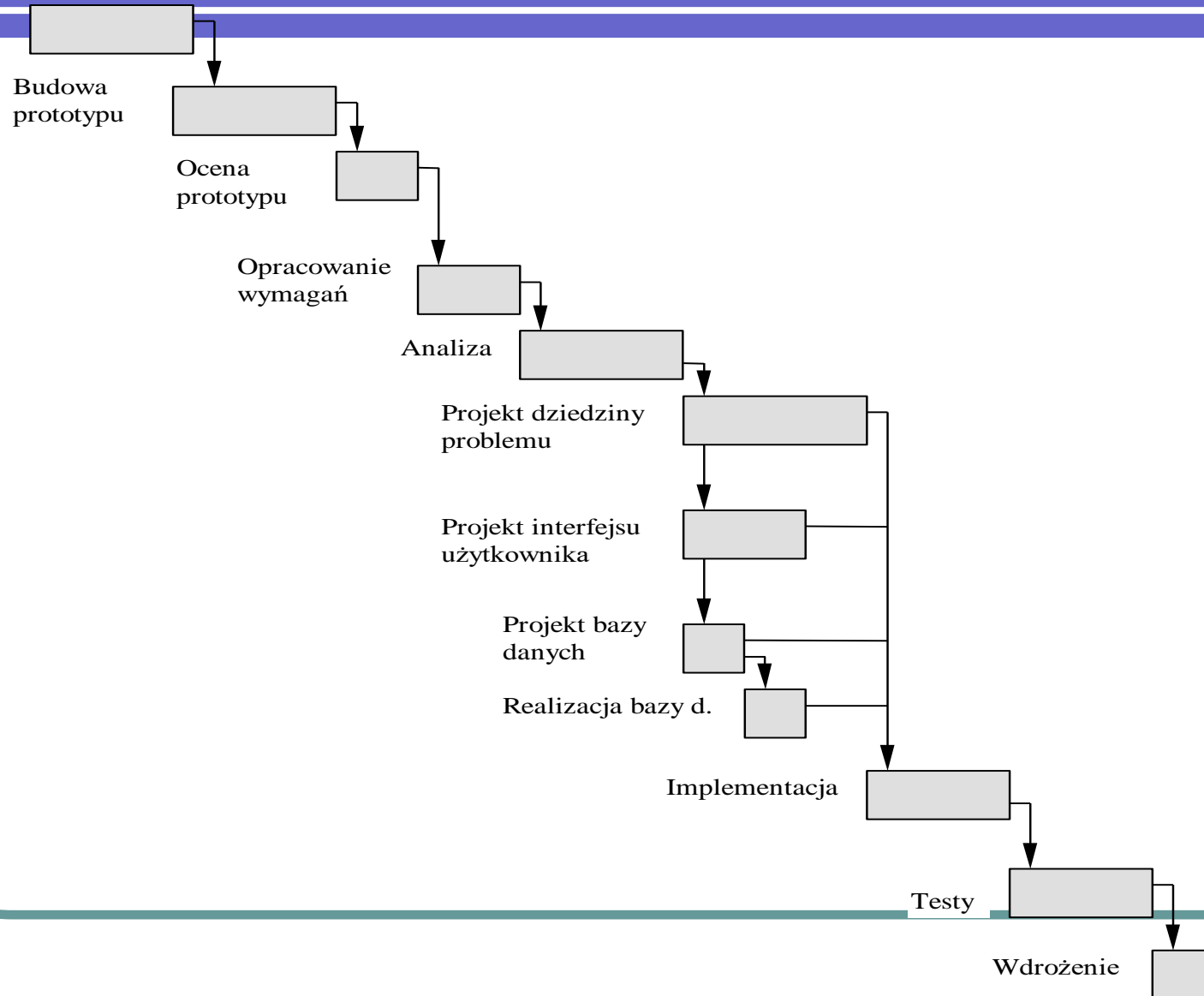
Rozważa się kilka możliwych rozwiązań.

Propozycje rozwiązań powinny być poprzedzone określeniem ograniczeń, przy których przedsięwzięcie będzie realizowane

- maksymalne nakłady
- dostępny personel
- dostępne narzędzia
- ograniczenia czasowe

miesiące

Wstępne zbieranie wymagań



Normy jakości oprogramowania

ISO/IEC TR 9126 software engineering product quality standards:

- part 1- Quality model : 2001
- part 2 – External metrics : 2003
- part 3 - Internal metrics : 2002

Syndrom LOOP

L ate	- późno
O verbudget	- przekroczony budżet
O vertime	- nadgodziny
P oor quality	- kiepska jakość

Przyczyny niepowodzenia projektów

(wg Standish Group 1994)

- 13% - brak danych wejściowych
- 12% - niepełne wymagania i specyfikacje
- 12% - zmiany wymagań i specyfikacji
- 4% - nierealny plan, harmonogram
- 6% - nieodpowiedni personel
- 7% - nieznajomość technologii

1/3 projektów stwarza problemy z gromadzeniem, dokumentowaniem i zarządzaniem wymaganiami.

Udane projekty

projekty dostarczone na czas i w ramach budżetu:

- duże firmy - 9%
- małe firmy – 16%

Przyczyny sukcesu: (wg Standish Group 1994)

- 16% - zaangażowanie użytkownika
- 14% - wsparcie kierownictwa
- 12% - jasne określenie wymagań

Podsumowanie wad wg Capersa Jonesa

wada	Potencjalna wada	Skuteczność usuwania	Dostarczone wady
wymagania	1.00	77%	0.23
projekt	1.25	85%	0.19
kod	1.75	95%	0.09
dokumentacja	0.60	80%	0.12
niewłaściwe poprawki	0.40	70%	0.12
suma	5.00	85%	0.75

Względne koszty usuwania błędów na różnych etapach (Davies 1993):

Względne koszty usuwania błędów oprogramowania na różnych etapach (Davies 1993):

- **0.1-0.2** Określania wymagań
- **0.5** Projektowanie
- **1** **Kodowanie**
- **2** Testowanie jednostek
- **5** Testowanie akceptacyjne
- **20** Pielęgnacja

Kategorie błędów projektowych:

- projekt budowany z poprawnego zbioru wymagań
- projekt budowany na błędnych wymaganiach (**kosztowne**)
 - projekt będzie przerobiony lub odrzucony, marnotrawstwo czasu, wysiłku
 - błędy ukryte, wykrywane jako błędy wymagań po długim czasie

Przeciekanie błędów

74 % błędów wymagań wykrywanych na etapie analizy wymagań

„przeciekanie błędów:

- (Hughes Aircraft 15 lat)
- 4% - projekt wstępny, zaawansowany
- 7% - projekt szczegółowy
- 4% - pielęgnowanie

Błędy wymagań pochłaniają 25-40% sumy budżetu

Naprawa błędu może pociągnąć koszty w obszarach:

- Ponowna specyfikacja
- Ponowne projektowanie
- Ponowne kodowanie
- Ponowne testowanie
- Dokumentowanie
- Działania korygujące – likwidacja uszkodzeń
- Anulowanie np. kodu, projektu bazującego na błędnych wymaganiach
- Wycofanie gotowych wersji oprogramowania
- Koszty gwarancji
- Koszty serwisu (np. instalacja nowej wersji)
- Odpowiedzialność karna związana z produktem

Wymaganie (def Thyler)

- Możliwość rozwiązania problemu i osiągnięcia celu wymagana przez użytkownika
- Możliwość spełnienia umowy, normy, specyfikacji lub innej dokumentacji, którą musi mieć system

Zarządzanie wymaganiami

- Systematyczne podejście do **uzyskiwania**, **organizowania** i **dokumentowania** wymagań systemu oraz proces, który ustala i zachowuje umowę między klientem a zespołem realizującym przedsięwzięcie w zależności od zmieniających się wymagań systemu.
- Zbiór zorganizowanych, uniwersalnych i usystematyzowanych procesów i technik zajmowania się wymaganiami stawianymi złożonemu dużemu przedsięwzięciu.

Analiza problemu

Proces rozumienia rzeczywistych problemów i potrzeb użytkownika oraz proponowanie rozwiązań spełniających te potrzeby.

- Uzgodnienie definicji problemu
- Zrozumienie podstawowych przyczyn problemu kryjącego się za innym problemem
- Zidentyfikowanie udziałowców i użytkowników tworzonego systemu
- Zidentyfikowanie granicy systemu
- Zidentyfikowanie ograniczeń nałożonych na rozwiązanie

Uzgodnienie definicji problemu

Rozpisanie i sprawdzenie, czy zgadza się z tym każdy uczestnik przedsięwzięcia.

Format:

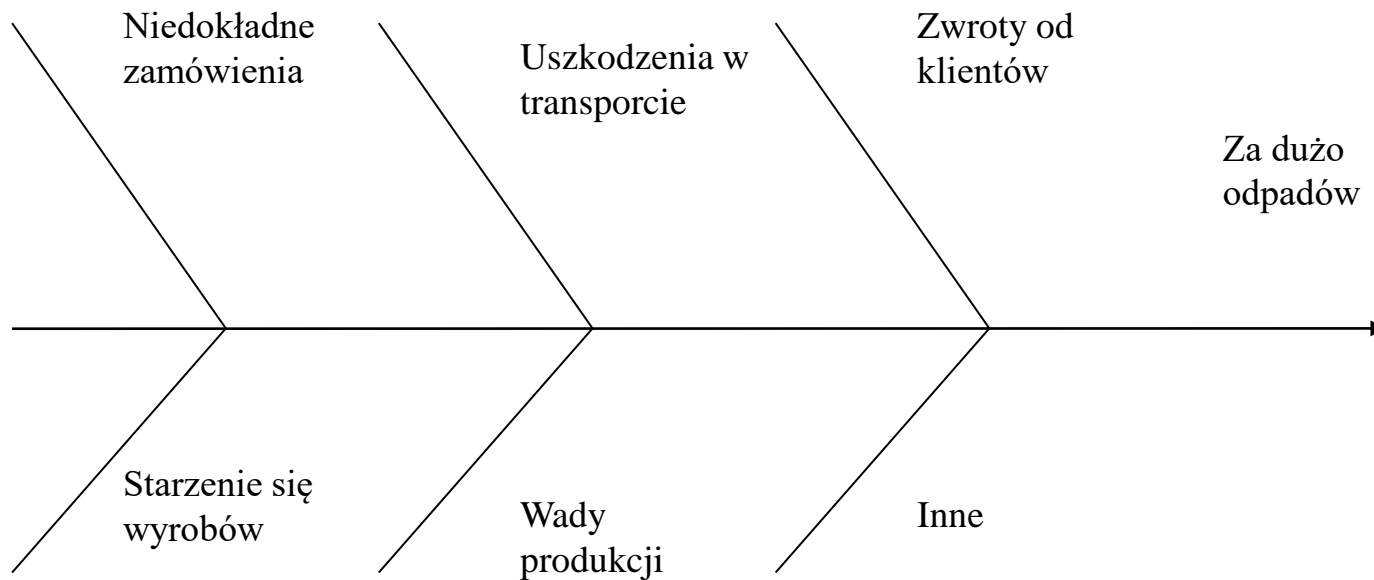
Problem polega na - opisz

Problem dotyczy - wskaż udziałowców

Rezultatem problemu jest - opisz wpływ na udziałowców i przedsiębiorstwo

Korzyści z rozwiązania problemu - wskaż proponowane rozwiązanie i wymień podstawowe korzyści

Zrozumienie podstawowych przyczyn problemu kryjącego się za innym



Zidentyfikowanie udziałowców i użytkowników

- **Udziałowiec** – każdy na kogo implementacja systemu ma zasadniczy wpływ
- Potrzeby udziałowców nie będących użytkownikami muszą być również określone i uwzględnione

Pomocne pytania

- Kim są użytkownicy
- Jaka jest rola klienta systemu
- Na kogo jeszcze będą miały wpływ wyniki działania systemu
- Kto będzie oceniał i zatwierdzał system po jego dostarczeniu
- Czy są inni zewnętrzni i wewnętrzni użytkownicy systemu, których potrzeby muszą być uwzględnione
- Kto będzie pielęgnował system

Identyfikacja aktorów

klucz w analizie problemu

- Kto dostarcza, używa lub usuwa informacje
- Kto obsługuje
- Kto utrzymuje, pielęgnuje, konserwuje
- Gdzie system będzie stosowany
- Gdzie zostaną wprowadzone dane
- Jakie inne systemy zewnętrzne będą z nim oddziaływać

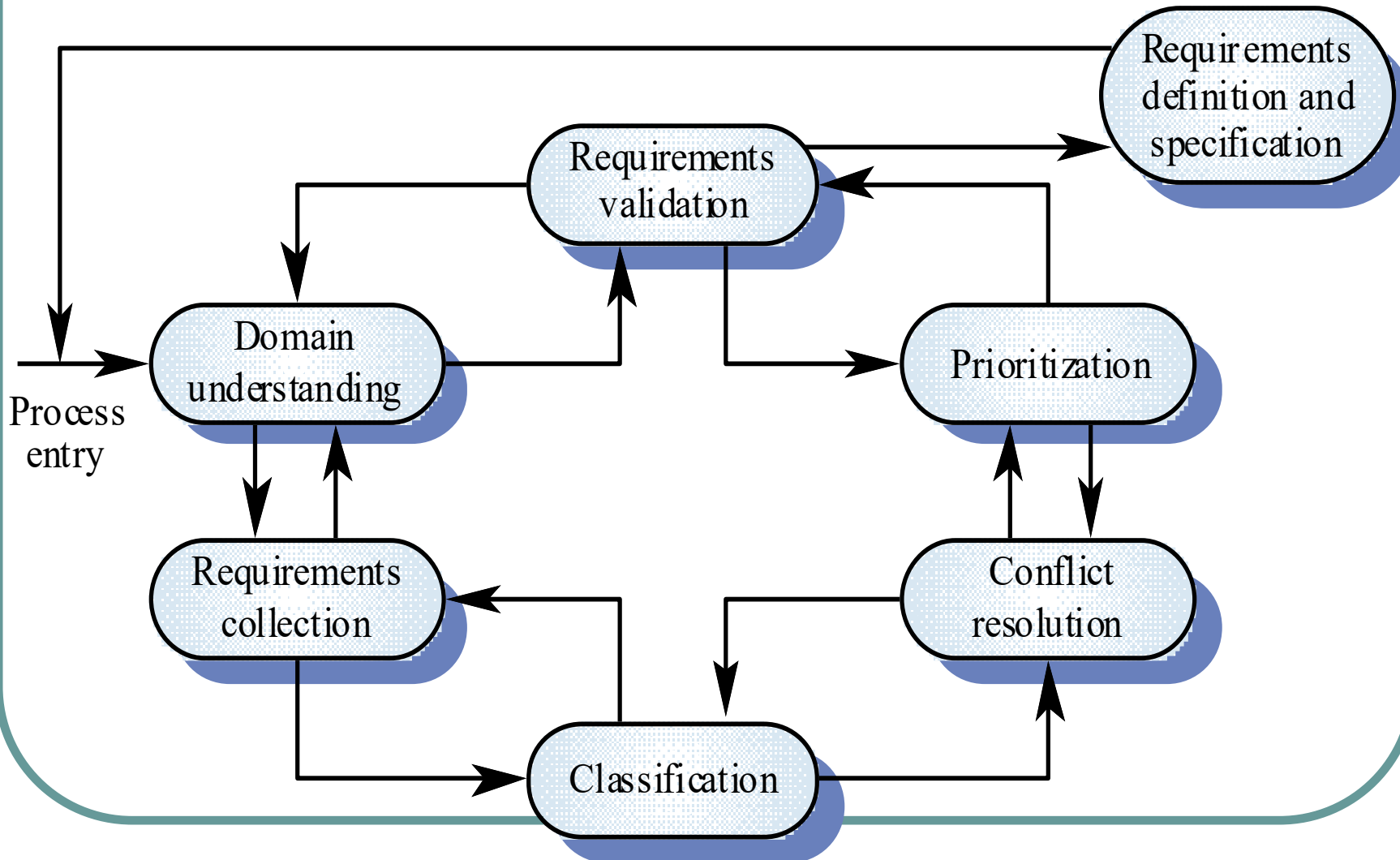
Zidentyfikowanie ograniczeń nałożonych na rozwiązanie

- **Ekonomiczne (ograniczenia finansowe, budżetowe, licencyjne)**
- **Polityczne**
- **Techniczne (ograniczenia w wyborze technologii, platformy, zakupione pakiety)**
- **Systemowe (rozwiązanie wbudowane w istniejący system, kompatybilność z innymi rozwiązaniami, systemy operacyjne)**
- **Środowiskowe (bezpieczeństwo, prawo, normy)**
- **Planowanie i zasoby (ograniczenia zasobów, ich zwiększenie stale, chwilowe, zatrudnienie osób spoza firmy)**

Uzyskiwanie wymagań

- Wywiad, ankieta
- Warsztaty wymagań
- Burza mózgów
- Przypadki użycia
- Odgrywanie ról
- Prototypy

Proces analizy wymagań



Rodzaje wymagań

- **Wymagania funkcjonalne** są to usługi oczekiwane przez użytkownika (bez uwag implementacyjnych).
- **Wymagania niefunkcjonalne** są to ograniczenia w jakich system ma pracować, standardy jakie spełnia np.
 - ✓ czas odpowiedzi,
 - ✓ zajętość pamięci,
 - ✓ niezawodność.

Wymagania funkcjonalne

Wymagania funkcjonalne powinny być:

- pełne, czyli zawierać wszystkie wymagania użytkownika,
- spójne, nie sprzeczne.

Istnieje wiele metod opisu wymagań np. w postaci formularzy, diagramów przypadków użycia, metod formalnych.

Przykłady

- **Wymaganie produktu**

Komunikacja pomiędzy systemem a użytkownikiem powinna być wyrażona w Unicode.

- **Wymaganie organizacyjne**

Proces rozwoju systemu i dostarczane dokumenty muszą być zgodne z normą (ISO 9000, CMMI).

- **Wymaganie zewnętrzne**

System nie powinien ujawniać operatorom żadnych danych osobowych klientów oprócz nazwisk i numerów identyfikacyjnych.

Wymagania niefunkcjonalne

- **Szybkość** – czas odpowiedzi użytkownika,
 - liczba przetworzonych transakcji w sekundzie
- **Rozmiar** – kilobajty
- **Łatwości użycia** - czas szkolenia,
 - liczba ekranów pomocy,
- **Niezawodność** – dostępność systemu,
 - średni czas pomiędzy błędami,
 - częstotliwość pojawiania się błędów,
 - prawdopodobieństwo błędu żądanej usługi,

Wymagania niefunkcjonalne-2

- **Solidność (robustness)** – czas uruchomienia po awarii,
 - procent zdarzeń powodujących awarie,
 - prawdopodobieństwo zniszczenia danych po awarii
- **Przenośność** – liczba systemów na których działa programowanie,
 - procent zależnych od systemu instrukcji.

Specyfikacje wymagań

- **Strukturalny język naturalny**
(formularze, szablony)
- **Pseudokod**
(if, else, while, ..., operatory logiczne, wcięcia, tekst)
- **Języki opisu projektu, opisu wymagań**
(PDL, PSL/PSA)
- **Notacje graficzne ze strukturalnymi opisami**
(SADT, przypadki użycia – use case)

Specyfikacje techniczne

- **Maszyny skończenie stanowe (FSM)**
(automaty, diagramy stanów)
- **Tabele decyzyjne, Drzewa decyzyjne**
- **Diagramy czynności (schematy blokowe)**
- **Modele związków encji**
- **Modele przepływu danych (DFD)**
- **Modele obiektowe dziedziny problemu**
- **Modele sieciowe (sieci Petri)**
- **Specyfikacje formalne**

Wymaganie użytkownika

3.5.1 Dodawanie węzłów do projektu

3.5.1.1 Edytor będzie udostępniał użytkownikom udogodnienia do dodawania do swoich projektów węzłów określonego typu

3.5.1.2 Sekwencja czynności:

1. Użytkownik powinien wybrać typ węzła, jaki należy dodać
2. Użytkownik powinien przesunąć wskaźnik w okolicy miejsca nowe węzła i zlecić jego dodanie
3. Użytkownik powinien przeciągnąć węzeł do jego ostatecznego położenia

Formularz specyfikacji wymagania

- Funkcja
- Opis
- Dane wejściowe
- Źródło danych wejściowych
- Dane wyjściowe, wynik
- Przeznaczenie
- Ograniczenia
- Warunek wstępny
- Warunek końcowy
- Efekty uboczne
- Uzasadnienie

Przykład specyfikacji wymagania

Funkcja *Dodaj węzeł*

Opis *Dodaje węzeł do istniejącego projektu. Użytkownik wybiera typ i położenie węzła (przesuwa wskaźnik na właściwy obszar). Po dodaniu węzeł jest zaznaczony.*

Dane wejściowe *Typ, położenie węzła, Identyfikator projektu*

Źródło danych wejściowych *Użytkownik, Baza danych*

Dane wyjściowe, wynik *Identyfikator projektu*

Przeznaczenie *Baza danych projektów*

Wymagania *Identyfikator określa korzeń grafu projektu*

Warunek wstępny *Projekt jest otwarty i wyświetlony*

Warunek końcowy *Pozostałe elementy projektu nie ulegają zmianie*

Efekty uboczne *Brak*

Klasy stałości wymagań

- **Wymagania stałe** – stabilne wymagania wynikające z podstawowej działalności firmy. Można je wywnioskować z modeli dziedziny zastosowania.
Np. szpitalu zawsze są wymagania dotyczące pacjentów, lekarzy, pielęgniarek itp.
- **Wymagania niestałe** – prawdopodobnie ulegną zmianie w trakcie tworzenia systemu lub po przekazaniu go użytkownikowi.
Np. zasady finansowania szpitala wg aktualnej ustawy o ochronie zdrowia

Poziomy identyfikacji wymagań

1) Potrzeby udziałowców

Cele systemu (goals)

- często niejasne i niejednoznaczne

2) Cechy systemu – usługi, których dostarcza system w celu spełnienia jednej lub więcej potrzeb udziałowca

3) Wymagania stawiane oprogramowaniu

Atrybuty cech produktu (1)

- **status**

śledzi postęp np. zaproponowano, zatwierdzono, wprowadzono

- **priorytet/korzyść**

stosuje się w zarządzaniu zakresem
np. konieczne, ważne, użyteczne

- **wysiłek**

oszacowanie liczby osób, tygodni, linii kodu np.
poziom wysiłku niski, średni, wysoki

- **ryzyko**

wielkość prawdopodobieństwa, że cecha
spowoduje niepożądane zdarzenie np. poziom
ryzyka niski, średni, wysoki

Atrybuty cech produktu (2)

- **stabilność**
wielkość prawdopodobieństwa, że cecha zmieni się
- **wersja docelowa**
w której cecha pojawi się po raz pierwszy
- **przydzielenie do**
cechy muszą być przydzielone do przyszłych zespołów odpowiedzialnych za definiowanie, ew. realizację
- **powód**
do śledzenia źródła pożądanej cechy, np. odwołanie do strony i wiersza specyfikacji

Macierz atrybutów

Identyfikator

Krótki tekst

Atrybut

Atrybut

AT: Attribute Ma			Priority	Difficulty
Requirements:				
FEAT1: The QBS system shall, upon user request,...			Medium	High
FEAT2: The QBS system shall provide a loan officer...			Low	Medium
FEAT3: The QBS System shall calculate the blue...			Medium	Medium
FEAT4: The QBS system shall allow only...			High	Medium
FEAT5: The QBS system shall allow only...			High	Medium

FEAT1: The QBS system shall, upon user request, display detailed customer information including: □Name, □Address, □Phone Number□Account Numbers□L				
---	--	--	--	--

Pełny tekst

Miary jakości

Miary jakości do oceny specyfikacji wymagań stawianych oprogramowaniu (IEEE 830)

- poprawny
- jednoznaczny
- kompletny
- spójny
- uporządkowany wg ważności i stabilności
- sprawdzalny
- modyfikowalny
- możliwy do śledzenia oraz możliwy do zrozumienia

Zatwierdzanie wymagań

- **Przeglądy wymagań**
 - Systematyczna analiza przez zespół recenzentów
- **Prototypowanie**
 - Wykonywalny model systemu (poziomy prot.)
- **Generowanie przypadków testowych**
 - Zaprojektowanie testów akceptacyjnych dla wymagań funkcjonalnych i нефunkcjonalnych
- **Automatyczna weryfikacja niesprzeczności**
 - Wykrywanie niezgodności w bazie wymagań zgodnie z regułami (CASE, modele formalne)

Zarządzanie zmianami wymagań

- **Planowanie zmian**

(atrybut stabilności, rozróżnianie starych, znanych, nowych, modyfikowanych wymagań)

- **Tolerancja linii bazowej na zmiany**

- **Kanał kontroli zmian**

(gromadzenie żądań zmian, ocena legalności, wpływu na system, podejmowanie decyzji, rozpowszechnianie informacji o zmianach)

- **Dokumentacja historii zmian**

- **Zarządzanie konfiguracją wymagań**
(wersjonowanie)

Śledzenie zależności (Traceability)

Określanie i pielęgnowanie relacji zależności pomiędzy różnymi artefaktami tworzonymi w cyklu rozwoju oprogramowania.

- **Pochodzenie**

- Wskazanie na udziałowców, którzy proponowali to wymaganie

- **Uzależnienie wymagań**

- Powiązania pomiędzy wzajemnie zależnymi wymaganiami

- **Związki z projektem**

- Powiązania z modułami projektu implementującymi wymagania, przypadkami testującymi,...

Macierz zależności (traceability matrix)

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		D	R					
1.2			D			D		D
1.3	R			R				
2.1			R		D			D
2.2								D
2.3		R		D				
3.1								R
3.2							R	

Rational RequisitePro - RU e-st - [UC-FEAT: All UC->FEAT]

File Edit View Requirement Traceability Tools Window Help

STRQ12: Need account a...
 STRQ13: The system must...
 STRQ14: Need to be able ...
 STRQ15: The system must...
 STRQ16: The system must...
 STRQ17: The system shall...
 STRQ18: The system shall...
 STRQ19: The system must...
 STRQ20: Tax forms will be...
 STRQ21: Tax information ...
 STRQ22: The system perf...

Use Cases
 All Use Cases
All UC->FEAT*
 UC1: Apply for Trading Ac...
 UC2: Execute Trade
 UC3: Get Quote
 UC4: Manage Portfolio
 UC5: Distribute News
 UC6: Review Account
 UC7: Maintain Customer A...
 UC8: Report Tax Informati...
 UC9: Maintain Web Site
 UC10: Resolve Account P...
 UC11: Learn System Featu...

Relationships:
 - direct only

FEAT1: The system uses the...
 FEAT2: The system provides...
 FEAT3: The system supports the...
 FEAT4: The system allows funds to...
 FEAT5: The system allows a...
 FEAT6: The system can be upgraded...
 FEAT7: During upgrades the system...
 FEAT8: The system uses the Market...
 FEAT9: The system provides an FAQ...
 FEAT10: The system provides an "I...
 FEAT11: All Web pages shall be...
 FEAT12: All personal and financial...
 FEAT13: The system provides...
 FEAT14: The system runs on a...
 FEAT15: The system provides...
 FEAT16: The system provides...
 FEAT17: Transfer cash from one...
 FEAT18: The system executes...
 FEAT19: Charge to a credit card and...
 FEAT20: Rollover from a retirement...
 FEAT21: The system allows Market...
 FEAT22: The system allows Limit...
 FEAT23: The system allows mutual...
 FEAT24: The system allows stock...

UC	FEAT1	FEAT2	FEAT3	FEAT4	FEAT5	FEAT6	FEAT7	FEAT8	FEAT9	FEAT10	FEAT11	FEAT12	FEAT13	FEAT14	FEAT15	FEAT16	FEAT17	FEAT18	FEAT19	FEAT20	FEAT21	FEAT22	FEAT23	FEAT24
UC1: Apply for...																								
UC2: Execute Trade																								
UC3: Get Quote																								
UC4: Manage...																								
UC5: Distribute News																								
UC6: Review Account																								
UC7: Maintain...																								
UC8: Report Tax...																								
UC9: Maintain Web...																								
UC10: Resolve...																								
UC11: Learn System...																								

UC1: Apply for Trading Account
 FEAT1: The system uses the Financial Services Network to enable transfer of funds between other financial

View saved as STRQ Hierarchy 11 requirements

Narzędzia wspomagające CASE

- **Przechowywanie wymagań**
 - Gromadzenie w sposób bezpieczny i zorganizowany wymagań i ich atrybutów
- **Zarządzanie zmianami**
 - Dokumentowanie procesu zmian
 - Zachowanie spójności zbioru wymagań
 - Zarządzanie konfiguracjami wymagań
- **Wspomaganie śledzenia zależności**
 - Automatyczne identyfikowanie zależności
 - Przechowywanie, aktualizacja zależności.