

Specyfikowanie wymagań

Wykładowca

Dr inż. Zofia Kruczkiewicz

Struktura wykładu

- I. Wprowadzenie (wykład1)
- II. Zbieranie, analiza i negocjowanie wymagań
- III. Wymagania funkcjonalne i niefunkcjonalne.
- IV. Opis oraz zarządzanie wymaganiami.
- V. Faza strategiczna: cele, koszty, harmonogram, ryzyko.

Literatura

1. K. Frączkowski, Zarządzanie projektem informatycznym. Projekty w środowisku wirtualnym. Czynniki sukcesu i niepowodzeń projektów., Oficyna Wydawnicza Politechniki Wrocławskiej
2. Roger S. Pressman, Praktyczne podejście do oprogramowania, WNT, 2004
3. Stephen H. Kan, Metryki i modele w inżynierii jakości oprogramowania, Mikom, 2006

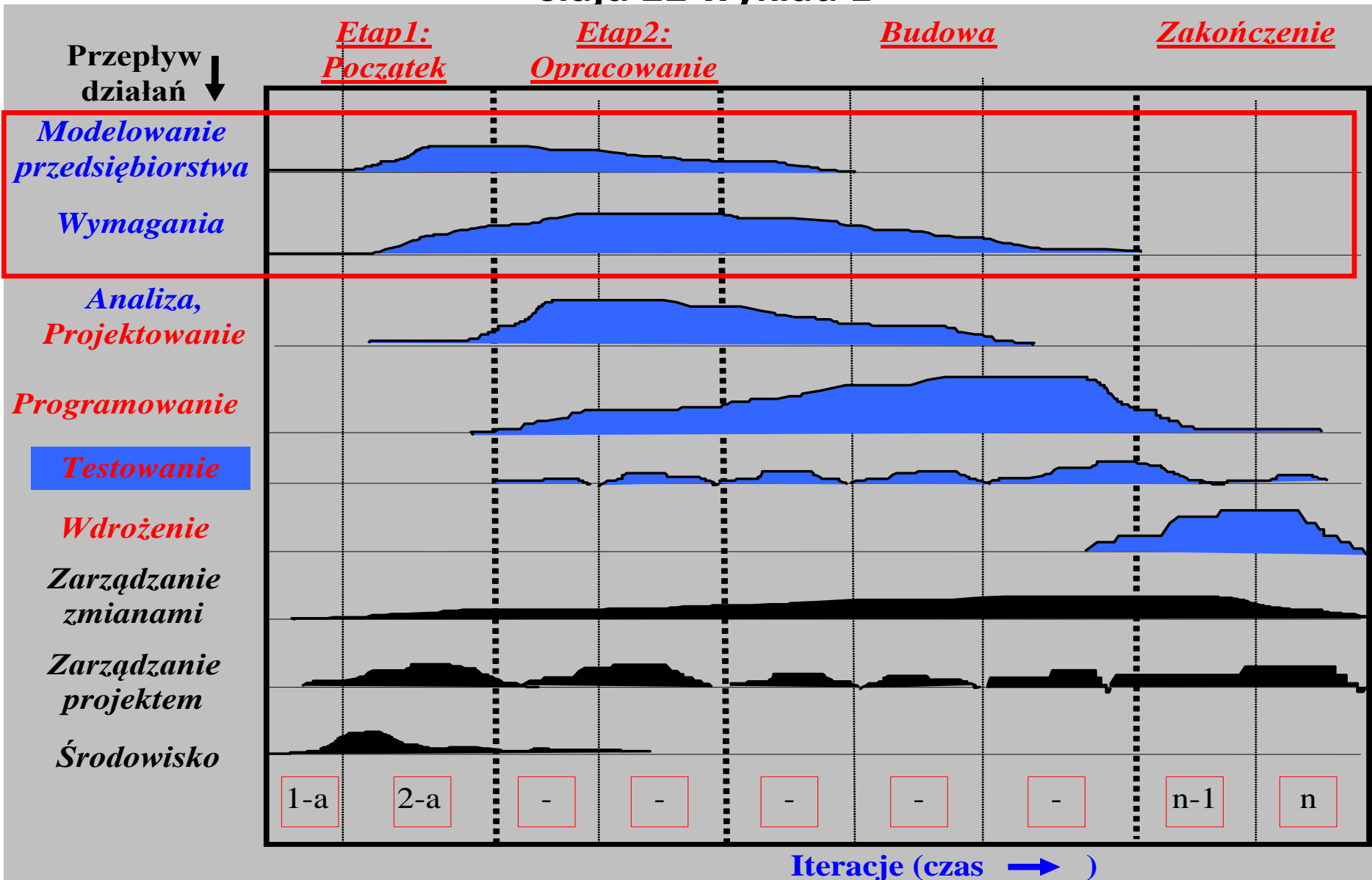
I. Wprowadzenie (wykład1)

Proces - model procesu wytwarzania oprogramowania - czyli model cyklu życia oprogramowania [3LU, 2LPW] – (slajd 20, Wykład 1)

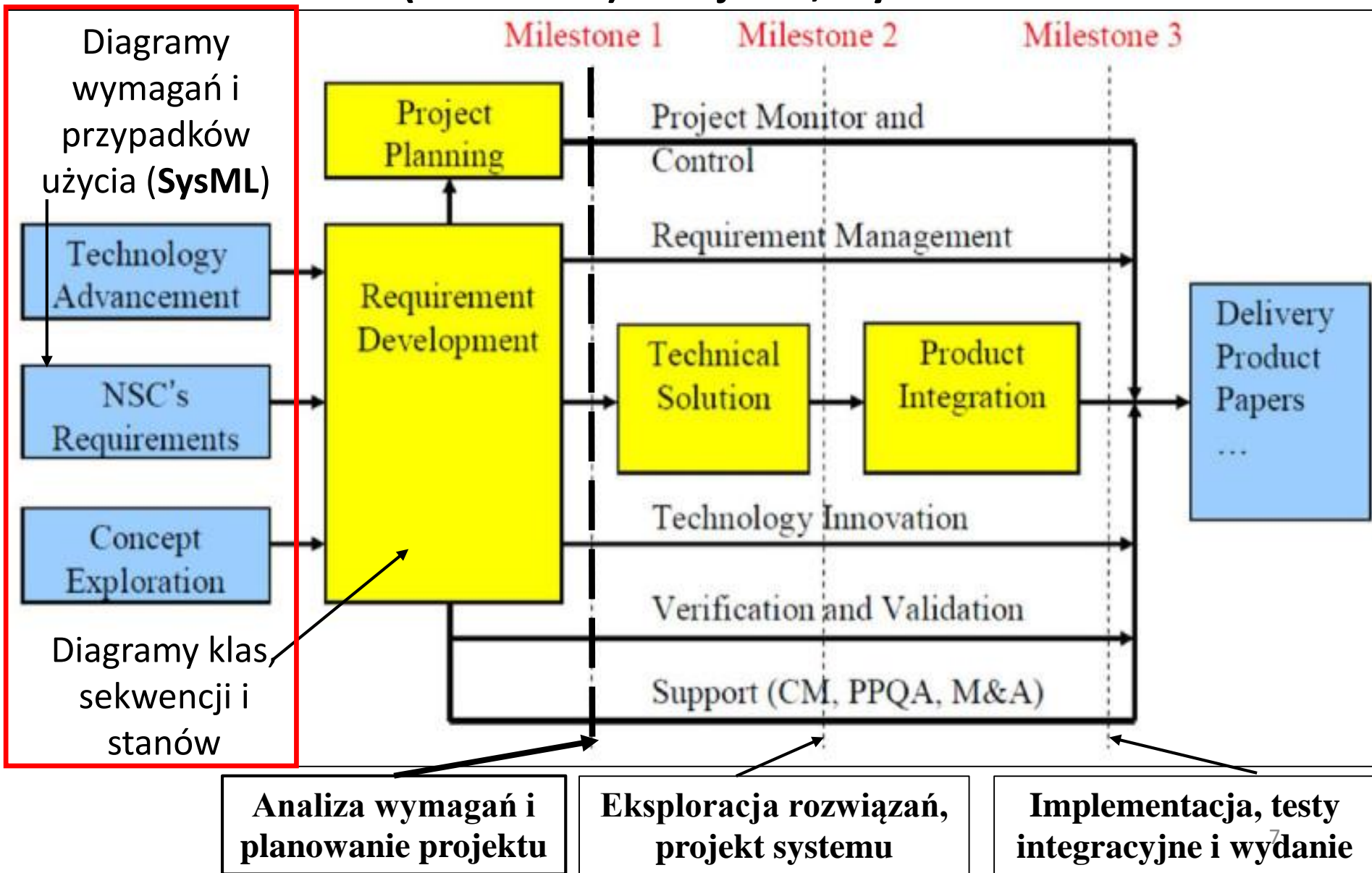
<p>Definiowanie</p> <p>Modelowanie struktury i dynamiki systemu</p>	<p>Prace techniczne, łączenie rozwiązań</p> <p>Implementacja systemu, struktury i dynamiki generowanie kodu</p>	
<p>Perspektywa koncepcji <i>co należy wykonać?</i></p>	<p>Perspektywa specyfikacji <i>jak należy używać?</i></p>	<p>Perspektywa implementacji <i>jak należy wykonać?</i></p>
<ul style="list-style-type: none"> • model problemu np. przedsiębiorstwa • <u>wymagania</u> • analiza (<i>model konceptualny</i>) • testy <i>modelu</i> 	<ul style="list-style-type: none"> • projektowanie (<i>model projektowy: architektura sprzętu i oprogramowania; dostęp użytkownika; przechowywanie danych</i>) • testy projektu 	<ul style="list-style-type: none"> • programowanie (<i>specyfikacja programu : deklaracje, definicje; dodatkowe struktury danych: struktury „pojemnikowe”, pliki, bazy danych</i>) • testy oprogramowania • wdrażanie • testy wdrażania

Proces - zunifikowany iteracyjno- przyrostowy proces tworzenia oprogramowania – **kiedy należy wykonać?** [3LU]

- slajd 22 wyklad 1



Cykl życia tworzenia oprogramowania w dziedzinie medycyny nuklearnej: **Light-Weight Capability Maturity Model Integration (LW-CMMI)** – slajd 59, wykład 1



II. Zbieranie, analiza i negocjowanie wymagań

1. Zasada siedmiu pytań (W5HH principle) – **dobry początek projektu**

1. **Why**: Dlaczego powstaje taki produkt ?
2. **What**: Co trzeba zrobić ?
3. **When**: Na kiedy trzeba zrobić ?
4. **Who**: Kto jest odpowiedzialny za poszczególne funkcje produktu ?
5. **Where**: Gdzie znajdują się ludzie związani z projektem ?
6. **How**: Jak powstanie produkt i jak będą przebiegać prace ?
7. **How**: Ile zasobów trzeba będzie poświęcić ?

2. Zakres działania produktu – ustalenia między klientem i wykonawcami

Zestawy przykładowych pytań o produkcie:

- **Zestaw początkowy:**
 - Kto potrzebuje?
 - Kto będzie używał?
 - Jakie korzyści ekonomiczne przyniesie?
 - Czy problem klienta można jednak inaczej rozwiązać?
- **Zestaw kolejny o oczekiwaniach klienta**
 - Jak klient opisz „dobry” wynik działania produktu ?
 - Jakie problemy mogłyby być rozwiązane w firmie dzięki zastosowaniu produktu ?
 - W jakim miejscu ma działać produkt ?
 - Jakie są wymagania odnośnie efektywności produktu oraz ograniczenia jego działania ?

2 (cd) Zakres działania produktu – ustalenia między klientem i wykonawcami

- **Zestaw „meta-pytań” skierowanych do klienta:**
 - Czy klient czuje, że jest właściwą osobą do udzielania odpowiedzi?
 - Czy zadane pytania przez wykonawców są sensowne ?
 - Czy nie zadano zbyt wielu pytań ?
 - Kto może udzielić dodatkowych informacji ze strony klienta ?
 - Czy analityk powinien zadać jeszcze jakieś pytania ?

Wykonalność produktu – wnioski po rozmowie z klientem

3. Wykonanie opisu „świata rzeczywistego”, przekazywany przez stronę zamawiającą oprogramowanie, wg następującego schematu:

1. Opis zasobów ludzkich - Co robią pracownicy?

Na podstawie tego opisu określa się po przystąpieniu do modelowania oprogramowania, jakie funkcje powinno wykonywać tworzone oprogramowanie, aby automatycznie wspomagać pracę ludzi, Uwaga: w przypadku systemów wbudowanych opis „świata rzeczywistego” powinien dotyczyć procesów technologicznych, które należy zautomatyzować.

2. Przepisy i strategia firmy - Co ogranicza działalność firmy?

Wykonanie opisu zawierającego specyfikację przepisów, ustaw, zarządzeń, strategii firmy itp. Po przystąpieniu do modelowania oprogramowania należy uwzględnić te ograniczenia na działanie funkcji, jakie system powinien wykonywać, aby wspomagać automatycznie pracę ludzi.

3. (cd) Wykonanie opisu „świata rzeczywistego”, przekazywany przez stronę zamawiającą oprogramowanie, wg następującego schematu

3. Dane techniczne

Opis warstwy technicznego systemu informacyjnego (rozmiary, architektura, liczba użytkowników, pozostałe funkcje programu, stosowane dotąd oprogramowanie i sprzęt):

- 3.1. Dane ilościowe związane z przetwarzaniem realizowanym przez projektowany program: ilu pracowników, ile danych, jak często wykonuje się poszczególne operacje na danych,
- 3.2. Dane o lokalizacji firmy
- 3.3. Dane o profilach klientów firmy oraz pracowników firmy,
- 3.4. Dane o używanym sprzęcie i oprogramowaniu.

Przykład: Opis biznesowy „świata rzeczywistego” biblioteki.

1. Opis zasobów ludzkich

Bibliotekarz może dodawać do katalogu tytułów nowe tytuły. Każdy tytuł jest reprezentowany przez dane: tytuł, autor, wydawnictwo, ISBN oraz informacje o liczbie egzemplarzy i miejscu ich przechowywania i występuje w bibliotece jako pojedyncza informacja dla każdego tytułu. Pewna grupa tytułów opisuje książki nagrane na wybranym nośniku, dlatego dodatkowo tytuł zawiera dane nagrania np nazwisko aktora. Każdy egzemplarz, niezależnie, czy jest książką czy kasetą, jest opisany odrębną informacją zawierającą numer egzemplarza, który może się powtarzać dla różnych tytułów. Bibliotekarz może dodawać nowe tytuły i egzemplarze oraz je przeszukiwać, natomiast klient może jedynie przeszukiwać tytuły i sprawdzać egzemplarze wybranych tytułów.

W celu wypożyczenia książki klient musi ją **zarezerwować** podając dane rejestracji, dane książki oraz datę rezerwacji. Klient musi **wypożyczyć** zarezerwowaną książkę w terminie jej rezerwacji podając dane rejestracji i rezerwacji, wyszukać rezerwację i następnie ją usunąć. Musi wykonać dane wypożyczenia zawierające: dane rejestracji, dane zarezerwowanej książki oraz datę zwrotu. Rezerwacje można usunąć bez konieczności jej wypożyczenia – w terminie rezerwacji.

W celu **zwrotu książki** należy podać dane rejestracji oraz dane wypożyczonej książki. Zwrot musi nastąpić w okresie wypożyczenia podanym w danych wypożyczenia.

Przykład:(cd) Opis biznesowy „świata rzeczywistego” biblioteki

2. Przepisy

Pracownik ponosi odpowiedzialność za poprawność danych - odpowiada materialnie za niezgodność danych ze stanem wypożyczalni.

3. Dane techniczne

Klient może przeglądać dane wypożyczalni za pośrednictwem strony internetowej lub bezpośrednio za pomocą specjalnego programu. Zakłada się, że klientów jednocześnie przeglądających dane wypożyczalni może być ponad 1000 oraz wypożyczalnia może zawierać kilkadziesiąt tysięcy tytułów oraz przynajmniej dwukrotnie więcej egzemplarzy. Biblioteka składa się z kilku ośrodków w różnych miastach na terenie kraju (lista miast jest dołączona do umowy). Zaleca się stosowanie technologii Java.

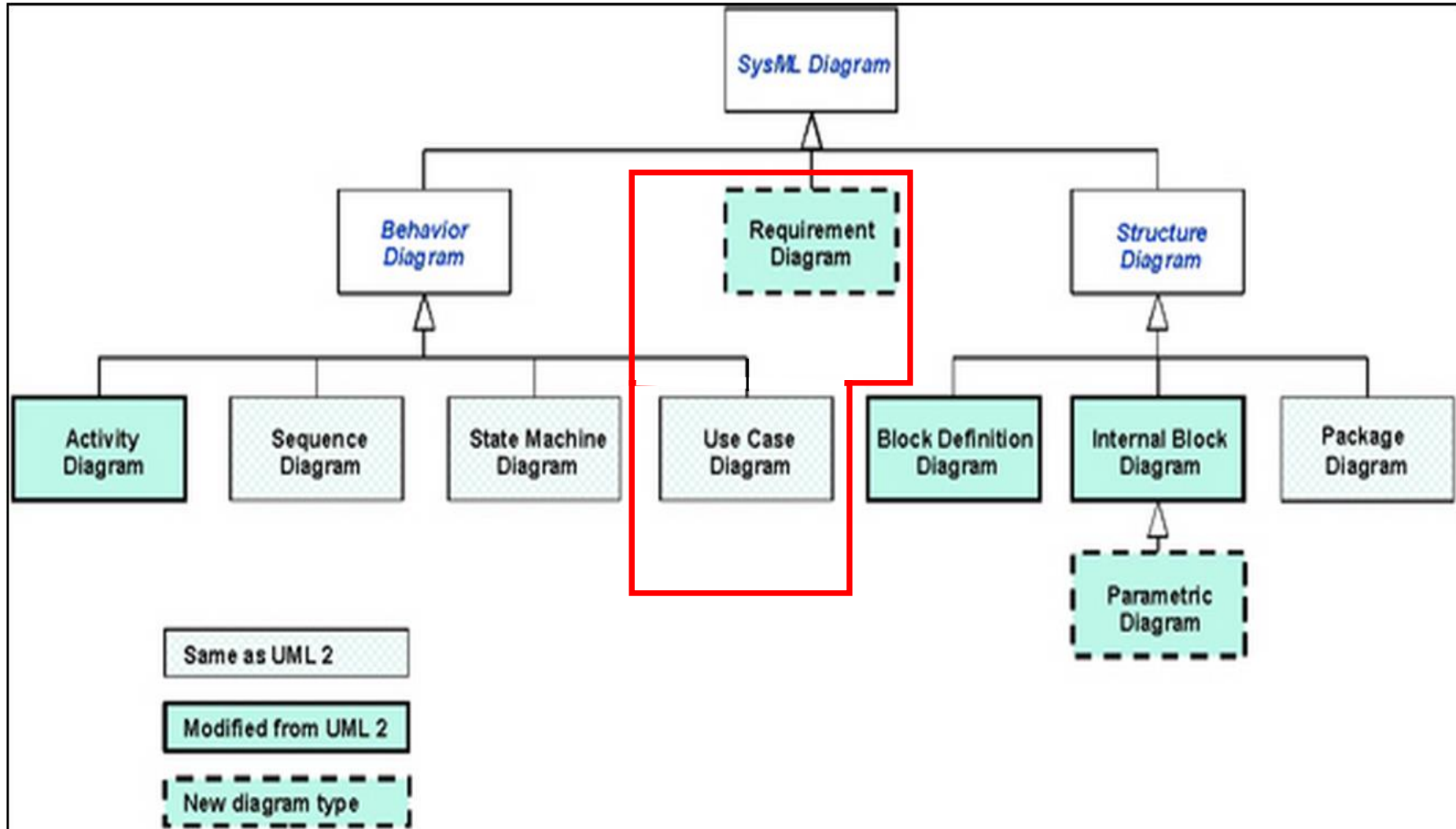
III. Wymagania funkcjonalne i niefunkcjonalne - diagramy wymagań SysML.

1. Zdefiniowanie wymagań funkcjonalnych i niefunkcjonalnych programu na podstawie opisu biznesowego

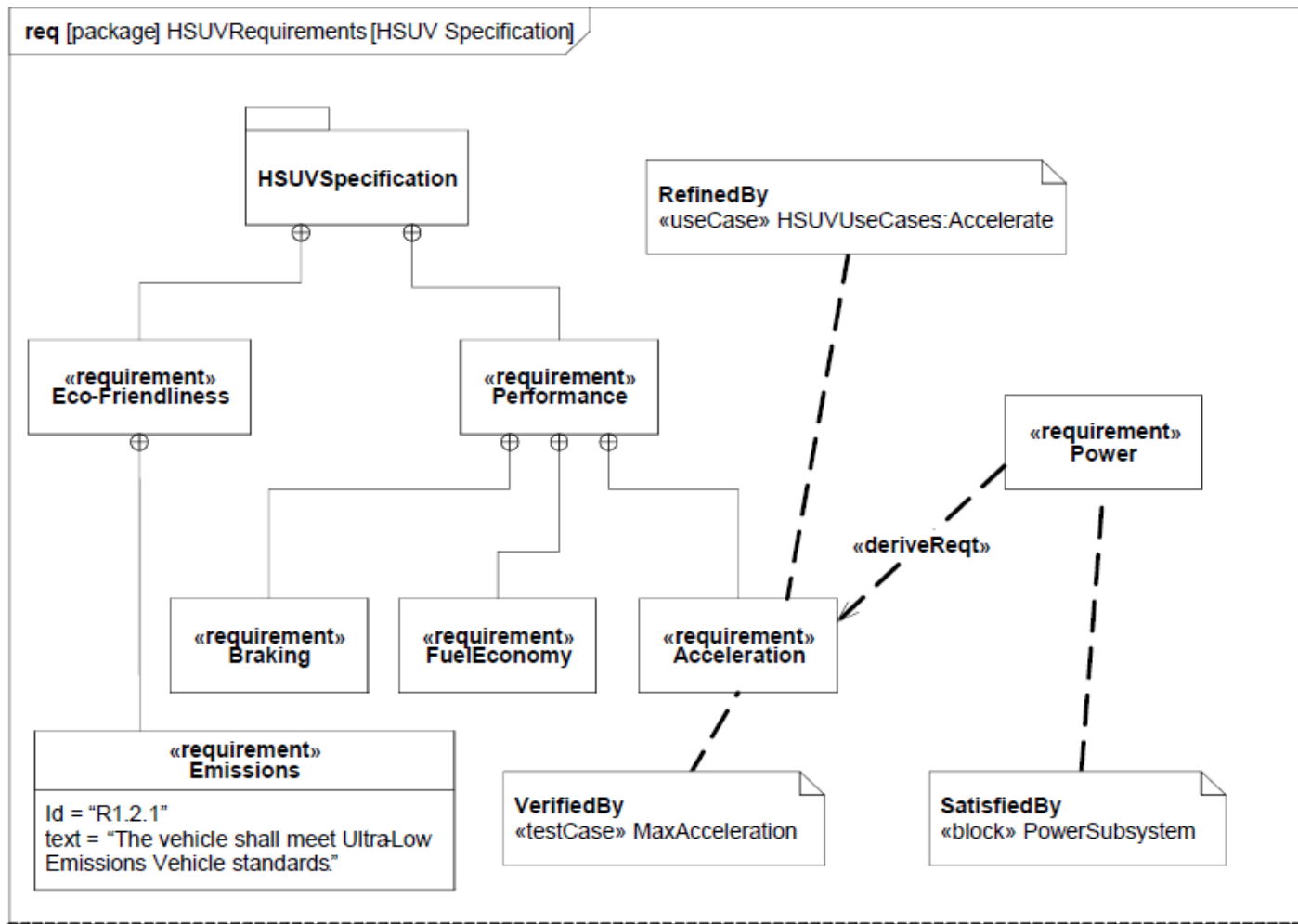
Wymagania te przedstawiają czynności projektowanego oprogramowania

1. Co program powinien robić, aby zautomatyzować procesy wykonywane przez tzw. zasoby ludzkie? – reprezentowane przez wymagania funkcjonalne
2. Jakie rozwiązania technologiczne zastosować, aby program spełniał oczekiwania strony zamawiającej w zakresie bezpieczeństwa (security), niezawodności (safety), skalowalności i wydajności.

2. Produkt – diagram wymagań SysML (slajd 42, wykład 1)

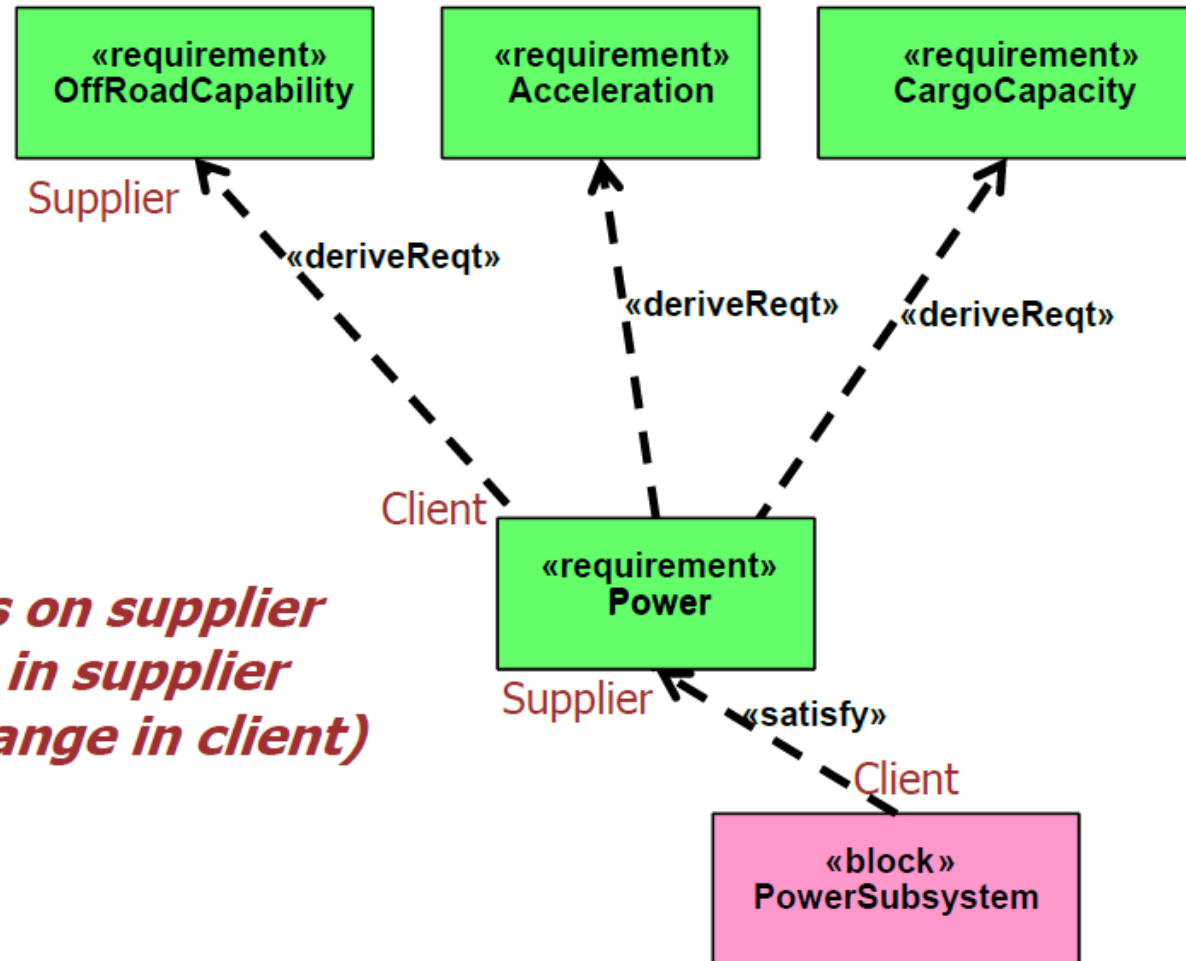


Requirements Breakdown



Requirement Relationships Model the Content of a Specification

Example of Derive/Satisfy Requirement Dependencies



*Client depends on supplier
(i.e., a change in supplier
results in a change in client)*

from OMG

Arrow Direction Opposite Typical Requirements Flow-Down

IV. Opis oraz zarządzanie wymaganiami. Przykład zastosowania diagramu wymagań.

1. Przykład - konfiguracja wymagania programu

Requirement Specification

General | Stereotypes | Attributes

Name: Korzystanie z zasobów biblioteki przez klienta

ID: REQ004

Source: Wypożyczalnia

Kind: Functional

Verify Method:

Risk: Medium

Status: Proposed

Text:

Zarejestrowany klient biblioteki może zarezerwować
książki, które nie są
dostępne w danej chwili. W momencie, kiedy
zamówione książki są dostępne
-albo po zwrocie lub dzięki zakupowi, można je
wypożyczyć i usunąć rezerwację.
Rezerwację można usunąć niezależnie.
Wszystkie operacje musi zatwierdzić bibliotekarz.

Reset OK Cancel Apply Help

Functional

Functional

Performance
Interface

Analysis

Demonstration

Inspection

Test

Medium

High

Medium

Low

Proposed

Proposed

Approved

Rejected

Deferred

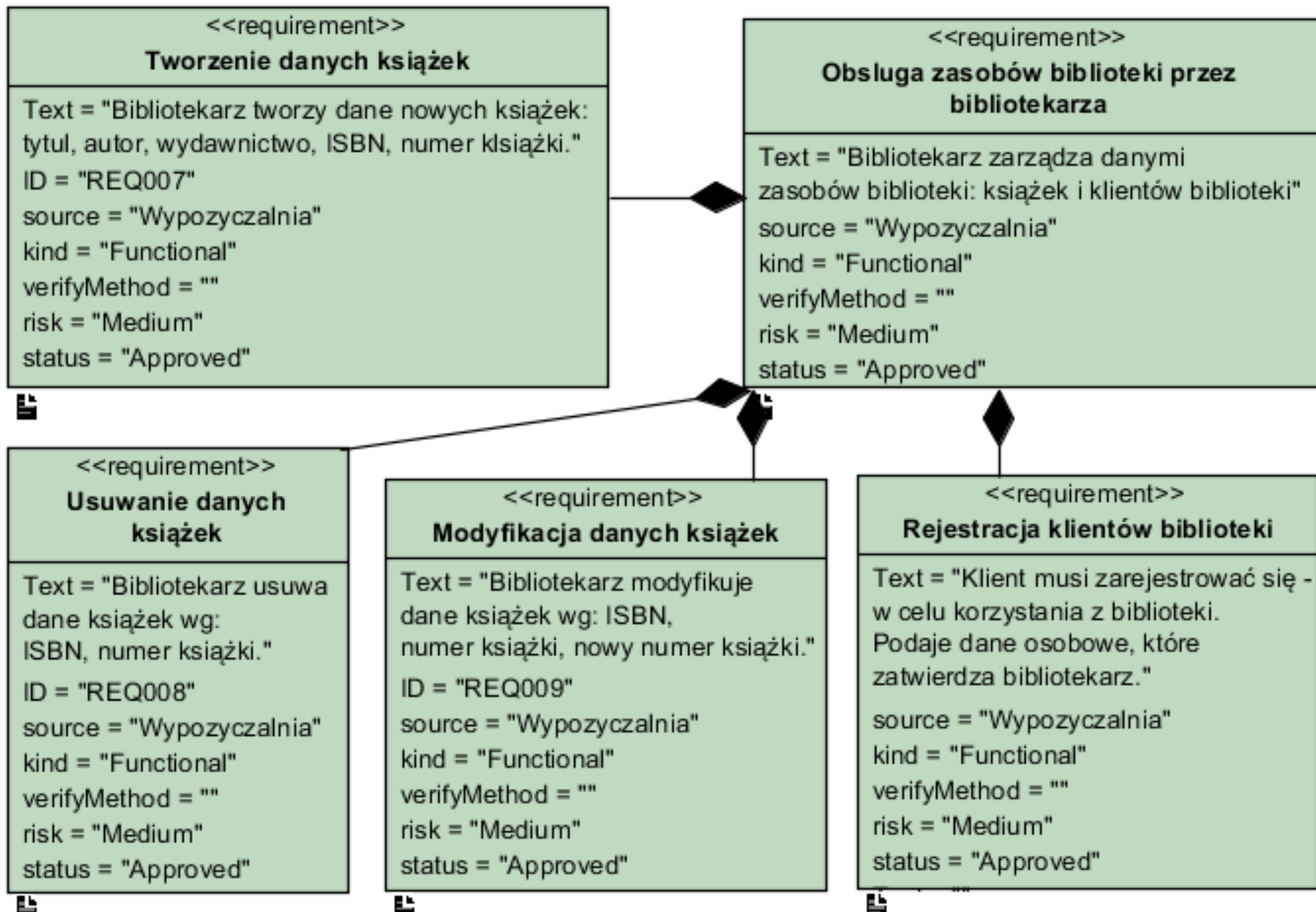
Implemented

Mandatory

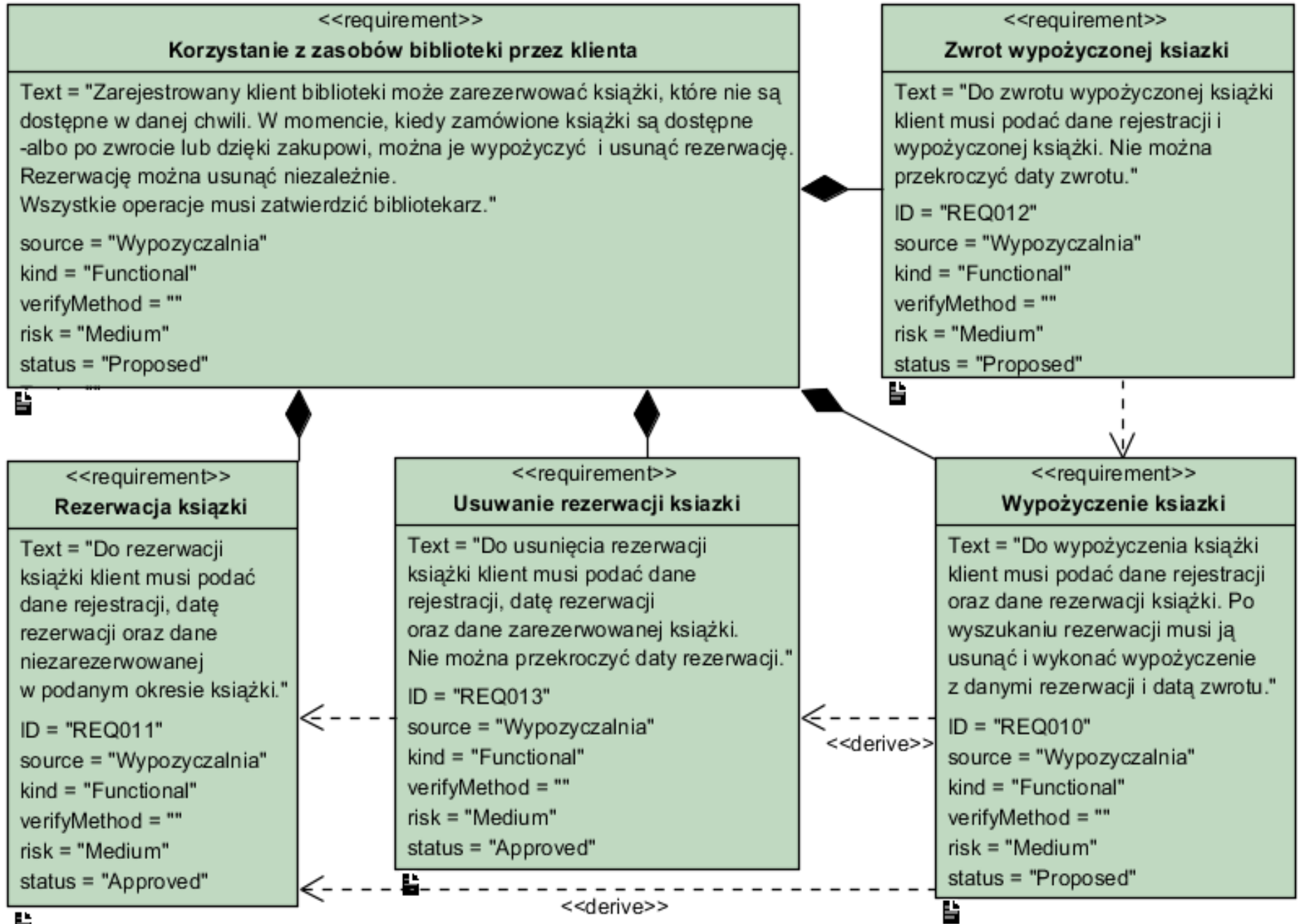
Obsolete

Wypoż

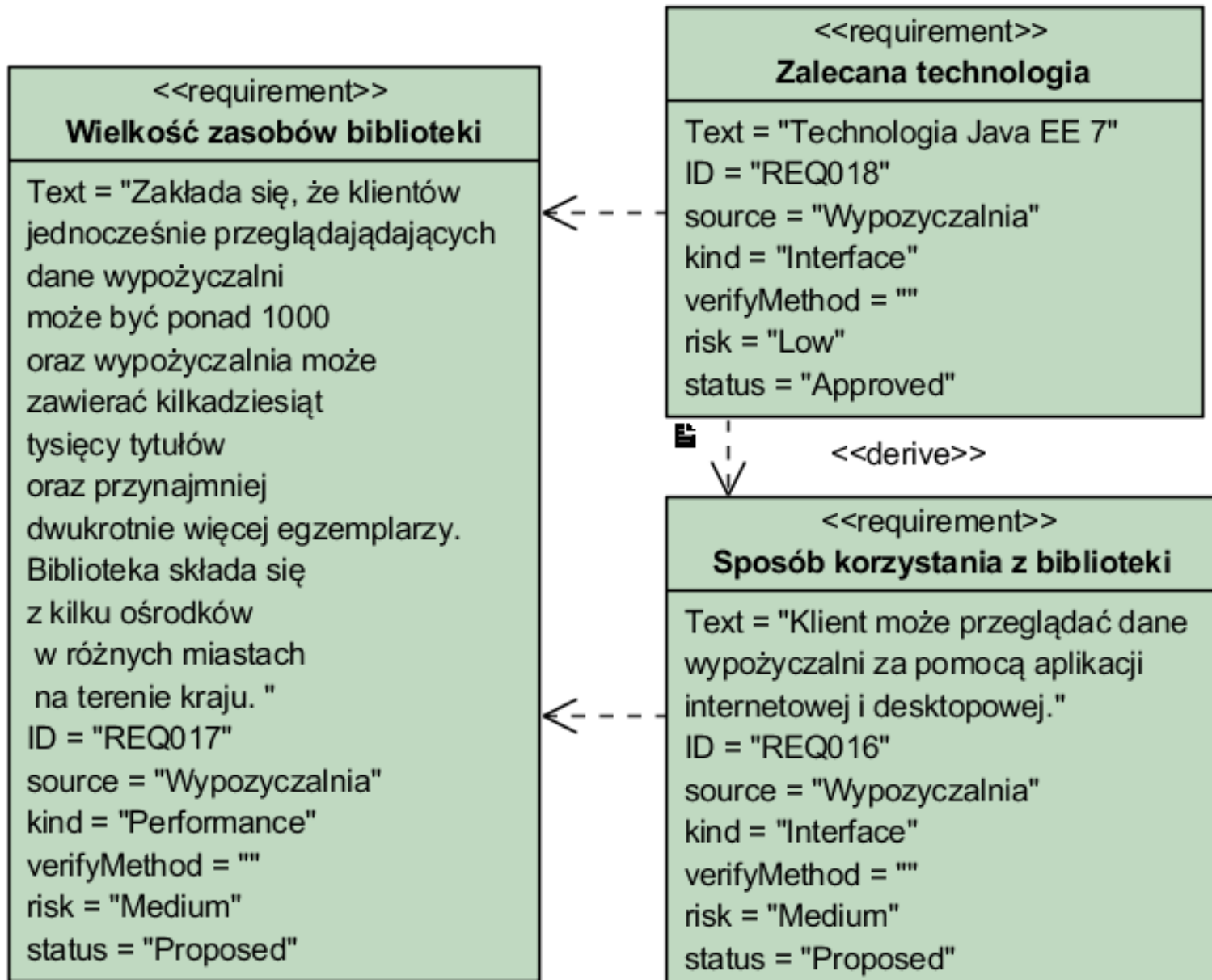
Przykład (cd) Diagram wymagań funkcjonalnych – program Biblioteka



Przykład (cd) Diagram wymagań funkcjonalnych – program Biblioteka



Przykład (cd) Diagram wymagań niefunkcjonalnych – program Biblioteka



V. Faza strategiczna: cele, koszty, harmonogram, ryzyko.

1. Planowanie projektów

Planowanie – **dokładność** zależy od:

1. Dokładności oceny złożoności i wielkości produktu – dobór miar produktu,
2. Umiejętności obliczenia kosztu, terminów i nakładu pracy na podstawie planowanej wielkości
3. Uwzględnienia w planie projektu kwalifikacji wykonawców,
4. Zmienności wymagań stawianych produktowi i stabilności technologii oraz narzędzi

Ryzyko w planowaniu mierzone jest:

1. Stopniem niepewności prognoz ilościowych (koszty, terminy i wielkość produktu)
2. Dokładnością specyfikacji produktu

2. Prognozowanie przebiegu projektu: kosztu, pracochłonności, czasu trwania – wspomagane narzędziami

1. Sporządzać prognozę po uzyskaniu pełnej informacji o produkcie i procesie - **zakres działania oprogramowania**
2. Oprzeć prognozę o **zakończone projekty**
3. Zastosować proste **techniki dekompozycji** do oszacowania kosztu i pracochłonności
lub
4. Użyć jeden lub kilka **modeli prognostycznych:**

$$d = f(v_i)$$

gdzie:

d – prognozowana wielkość np. pracochłonność, koszt lub termin

v_i – wybrany zestaw parametrów np. oszacowana wielkość produktu jako liczba linii kodu LOC lub liczba punktów funkcyjnych FP

3. Techniki dekompozycji – prognozowanie **wielkości oprogramowania**

- 1. Zastosowanie logiki rozmytej** - wnioskowania przybliżonego techniki
 - Rodzaj planowanego produktu
 - Początkowy jakościowy rozmiar produktu
 - Stopniowe szacowanie spodziewanej wielkości produktu
- 2. Szacowanie miary punktów funkcyjnych** – szacowanie wielkości związaną z charakterystyką informacyjną produktu
- 3. Liczenie składników standardowych** - podsystemy, moduły, okna, raporty, programy interakcyjne, programy wsadowe, pliki, wiersze kodu źródłowego itp. **i porównanie z danymi projektów zakończonych**
- 4. Szacowanie wielkości zmian**
 - Przydatna, gdy wprowadza się zmiany do istniejącego produktu, oparta na oszacowanej liczbie i rodzaju zmian (powtórne użycie, dodanie, modyfikacja lub usunięcie kodu)

3.1. Techniki dekompozycji – prognozowanie oparte na analizie produktu

Procedura *dekompozycji* oparta na LOC

Zdefiniuj zakres działania produktu

Zidentyfikuj funkcje, dekomponując zakres działania

Do while pozostałe pewne funkcje

Wybierz j-tą funkcję

Ustal liczbę podfunkcji

Do while pozostały pewne podfunkcje

Wybierz k-tą podfunkcję

If k-ta podfunkcja przypomina n-tą podfunkcję w bazie historycznej danych porównawczych

Then

Sprawdź **koszt, pracochońność i wielkość** n-tej podfunkcji;

Zmodyfikuj te wielkości, biorąc pod uwagę wszelkie różnice ;

Przygotuj **częściową prognozę E_p** , bazując na zmodyfikowanych danych;

Całkowita prognoza = suma wszystkich E_p

3.1. (cd) Techniki dekompozycji – prognozowanie oparte na **analizie produktu**

```
Else      If koszt, pracochność i wielkość k-tej podfunkcji  
          można oszacować  
          Then Przygotuj częściową prognozę Ep;  
              Całkowita prognoza = suma wszystkich Ep  
          Else Podziel k-tą podfunkcję na mniejsze podfunkcje  
              Dodaj nowe podfunkcje do listy podfunkcji  
          Endif  
        Endif  
      Enddo  
    Enddo
```

Koszt wyprodukowania 1 linii kodu - 13 [USD]

3.1 (cd). Techniki dekompozycji – prognozowanie oparte na analizie produktu

Procedura dekompozycji oparta na punktach funkcyjnych FP (podobna do procedury opartej na LOC).

Opiera się nie na rozmiarach funkcji, lecz aspektach dziedziny informacyjnej programu, którymi są:

- wejścia, wyjścia programu,
- zapytania użytkownika,
- pliki danych
- interfejsy zewnętrzne,
- 14 współczynników złożoności,

oblicza się miary FP, zestawia z danymi historycznymi porównawczymi i prognozuje wielkość.

3.1 (cd). Techniki dekompozycji – prognozowanie oparte na analizie produktu

Procedura prognozowania niezależna od przyjętej miary wielkości oprogramowania (LOC, FP)

- Oszacowanie przedziału wartości, w której mogą znaleźć się miary (LOC lub FP) – **wielkości produktu**
- Używając danych historycznych, intuicji (niezalecane) i doświadczenia oszacować **optymistyczną, najbardziej prawdopodobną i pesymistyczną** wartość miary wielkości dla poszczególnych funkcji lub aspektów dziedziny informacyjnej programu – ocena stopnia niepewności przygotowywanych prognoz
- Obliczenie **wartości oczekiwanej** badanej zmiennej prognostycznej **S** na podstawie **wartości optymistycznej S_{opt} , najbardziej prawdopodobnej S_m i pesymistycznej S_{pes}** .

$$S = 1/6 (S_{opt} + 4 S_m + S_{pes})$$

- Porównuje się wartość oczekiwaną **S** z danymi historycznymi na temat wydajności pracy (mierzonej z użyciem LOC i FP) i sporządza prognozy

3.1.1. Przykład prognoz - szacowanie wielkości miarą LOC (wg p.3.1)

Funkcja	S_{opt} [LOC] Wartość optymistyczna	S_m [LOC] Wartość najbardziej prawdopodobna	S_{pes} [LOC] Wartość pesymistyczna	S [LOC] Wartość oczekiwana (prognoza)
Interfejs użytkownika i mechanizmy kontroli (IUMK)	2300
Analiza dwuwymiarowych obiektów geometrycznych (A2OG)	5300
Analiza trójwymiarowych obiektów geometrycznych (A3OG)	4600	6900	8600	6800
Zarządzanie bazą danych (ZBD)	3350
Prezentowanie grafiki (MWG)	4950
Obsługa urządzeń peryferyjnych (FOUP)	2100
Analiza projektów (MAP)	8400
Prognozowana wielkość kodu	33200

3.1.2. Przykład prognoz - szacowanie wielkości miarą FP (wg p. 3.2) Wyznaczenie wielkości aspektów

Wielkość	Optymistyczna S_{opt}	Prawdopodobna S_m	Pesymistyczna S_{pes}	Oczekiwana S	Waga	FP
Liczba wejść	20	24	30	24.33	4	97
Liczba wyjść	12	15	22	15.66	5	78
Liczba zapytań	16	22	28	22	4	88
Liczba plików	4	4	5	4	10	42
Liczba interfejsów zewnętrznych	2	2	3	2.16	7	15
Suma (S)						320

3.1.2. (cd) Przykład prognoz - szacowanie wielkości miarą FP

Wyznaczenie wartości 14 współczynników złożoności FP_i

LP	Współczynniki złożoności F_i	Wartość
1	Zapasowe kopie danych i ich odtwarzanie	4
2	Wymiana danych	2
3	Rozproszone przetwarzanie danych	0
4	Znaczenie efektywności	4
5	Istniejące środowisko użytkowe	3
6	Wprowadzanie danych przez użytkownika	4
7	Wprowadzanie danych przez wiele okien	5
8	Modyfikowanie plików podczas interakcji	3
9	Złożoność aspektów dziedziny informacyjnej	5
10	Złożoność przetwarzania danych	5
11	Wielokrotne użycie kodu	4
12	Planowane wdrożenie /instalacja	3
13	Wiele instancji	5
14	Możliwość modyfikowania	5

3.1.2. (cd) Przykład prognoz - szacowanie wielkości miarą FP

- Przewidywana liczba punktów funkcyjnych FP_e

$$FP_e = S * [0.65 + 0.01 * \sum(F_i)]$$

czyli:

$$FP_e = 320 * [1.17] = 375$$

(wg tabeli z p.3.3.2 (S) i 3.3.2 (cd) (F_i))

- Obliczenia kosztu wykonania kodu i pracochłonności (osobomiesiące)
- **Koszt wykonania 1 FP – 1230 [USD]**

3.2. Przykład prognozowania opartego na analizie procesu

Czynność	Kontakt z klientem	Planowanie	Analizowanie ryzyka	Projektowanie		Implementacja i wdrażanie		OK	Razem
Zadania do wykonania →				Analizowanie	Projektowanie	Kodowanie	Testowanie		
Funkcje produktu ↓									
IUMK				0.50	2.50	0.40	5.00	n/a	8.40
A2OG				0.75	4.00	0.60	2.00	n/a	7.35
A3OG				0.50	4.00	1.00	3.00	n/a	8.50
ZBD				0.50	3.00	1.00	1.50	n/a	6.00
MWG				0.50	3.00	0.75	1.50	n/a	5.75
FOUP				0.25	2.00	0.50	1.50	n/a	4.25
MAP				0.50	2.00	0.50	2.00	n/a	5.00
Suma	0.25	0.25	0.25	3.50	20.50	4.75	16.50		46.00
% pracy	0.6%	0.6%	0.6%	7.6%	44.5%	10%	36%		

Jednostka – [osobomiesiące]

3.3. Prognozowanie wielkości produktu. Model dynamiczny - równanie programistyczne

Pracochłonność mierzona jest rozkładem:

$$E = [\text{LOC} * B^{0.333}/P]^3 * t^{-4}$$

E –pracochłonność

Parametry:

t – czas trwania projektu (miesiące lub lata)

LOC – liczba linii kodu

Współczynniki:

B – współczynnik specjalnych umiejętności

P – współczynnik wydajności, który uwzględnia:

- Ogólną dojrzałość procesu wytwórczego i metod zarządzania
- Zakres stosowania sprawdzonych metod inżynierii oprogramowania
- Poziom używanych języków programowania
- Stan środowiska programowego
- Umiejętności i doświadczenie pracowników
- Złożoność planowanej aplikacji

Przykłady P:

Wbudowane systemy czasu rzeczywistego: **P = 2000,**

Systemy operacyjne i oprogramowania telekomunikacyjnego: **P = 10000**

Oprogramowanie dla firm: **P = 28000**

3.4. Przykłady zastosowania metod prognozowania wg p.3.1 – p.3.3

Uproszczone wersje wzorów dla równania programistycznego:

- $t_{\min} = 8.14 * (LOC/P)^{0.43}$

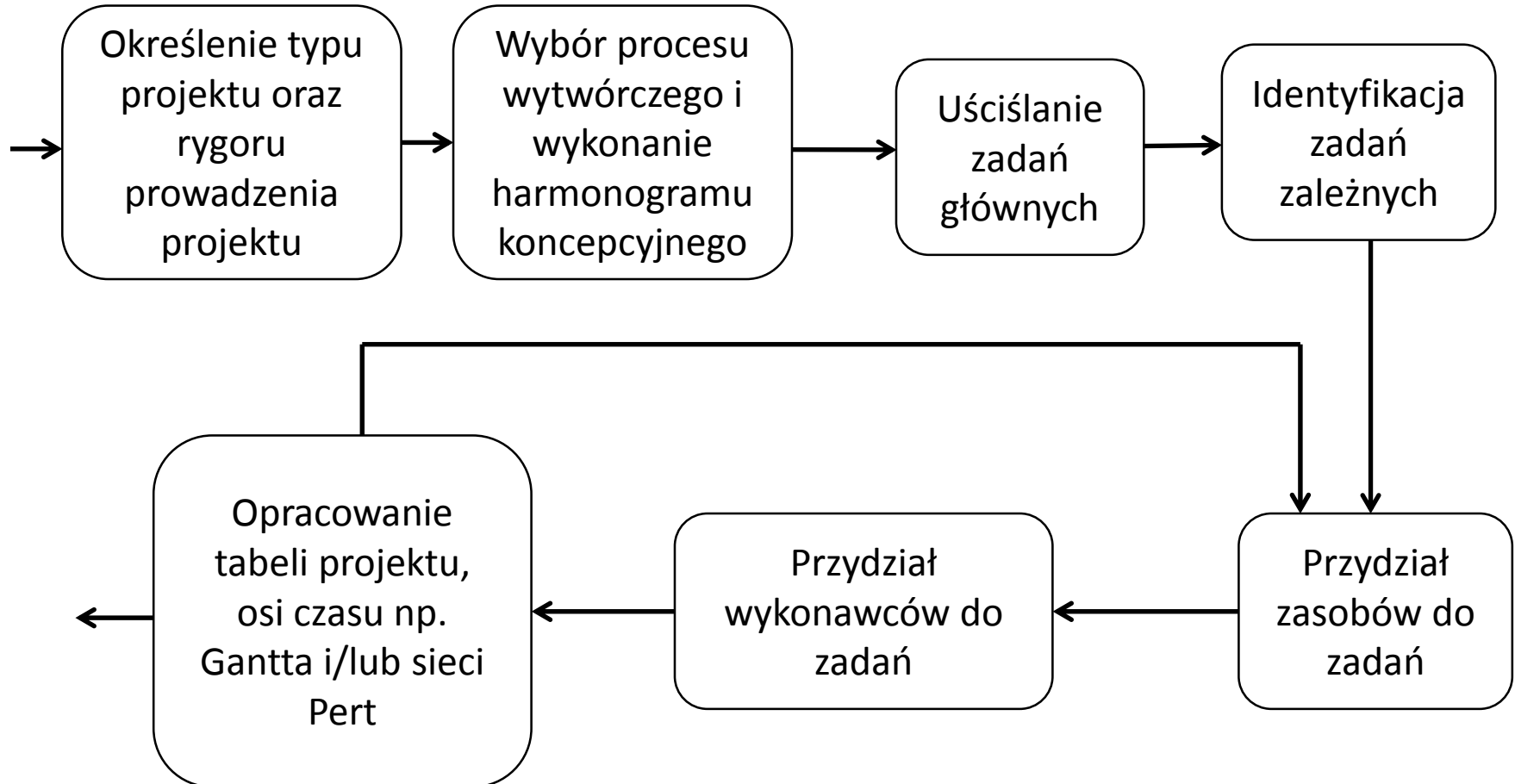
w miesiącach, dla $t_{\min} > 6$

- $E = 180 * B * t^3$,

w osobomiesiącach dla $E \geq 20$, t [lata]

	Koszt miesięcznej pracy 1 pracownika [USD]	E [Liczba osobomiesięcy]	t_{\min} Czas [miesiące]	Koszt całego projektu [USD]
Analiza procesu (p.3.2)	8000	46		46 * 8000 = 368 000
Analiza produktu LOC (p. 3.1. 3.1.1)	8000	33200 / 620 = 54		33200 * 13 = 431 000
Analiza produktu FP (p.3.1, 3.1.2)	8000	375 / 6.5 = 58		375 * 1230 = 461 250
Równanie programistyczne (p. 3.3)	8000	180 * 0.28 * (1.05) ³ = 58	8.14 * (33200 / 12000) ^{0.43} = 12.6 → 1.05 lat	58 * 8000 = 464 000

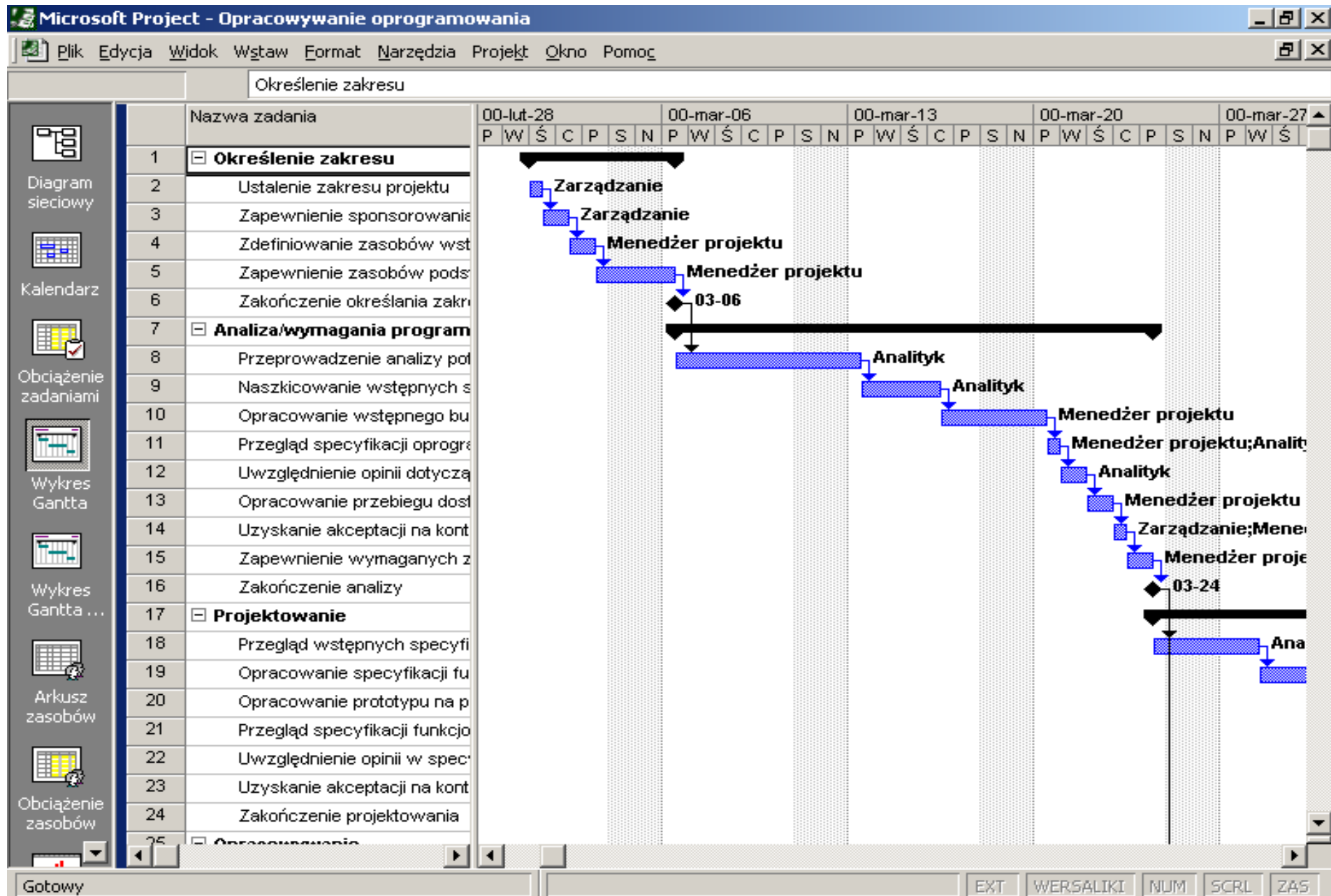
4. Metody tworzenia harmonogramów - schemat



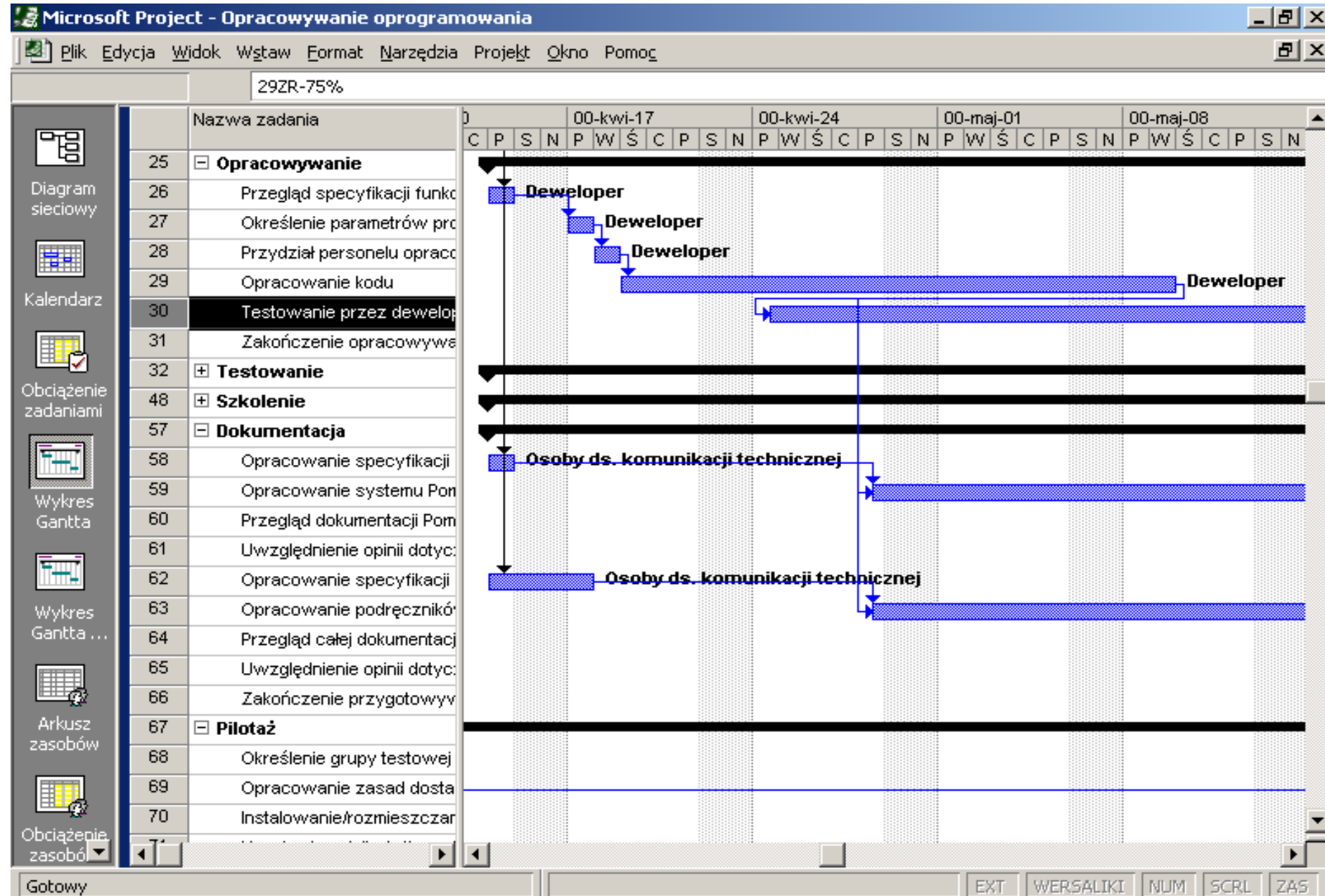
4.1. Metody tworzenia harmonogramów

- **Metoda ścieżki krytycznej CPM** (*Critical Path Method*) opiera się na zależnościach pomiędzy poszczególnymi etapami projektu i przewidywania czasu ich trwania. Wyznacza się czas trwania całego projektu i jednocześnie łączny czas tych etapów, których przebieg jest krytyczny dla realizacji całego zadania. Łączny czas trwania etapów krytycznych jest **ścieżką krytyczną**.
- **Metoda PERT** (*Program Evaluation and Review Technique*) jest systemem planowania kolejnych działań, gdzie przewidywany czas trwania opiera się na metodach statystycznych – jest używany w ograniczonym zakresie do wizualizacji zależności między zadaniami
- **Wykres Gantta – wykres osi czasu** jest systemem graficznym do przedstawiania kolejnych zadań na osi czasowej, ukazujących następstwo poszczególnych etapów:
 - Określenie czasochłonności
 - Określenie czasu wykonania
 - Określenie chwili rozpoczęcia zadania

4.2. Wykres Gantta – ścieżka krytyczna



4.2. (cd) Wykres Gantta - zadania zależne np. zadanie 30, 59, 63 zależne od 29



5. Strategie zarządzania ryzykiem

- **Strategia reakcji**

Działania naprawcze w momencie wystąpienia zagrożenia, często prowadzące do „zarządzania kryzysowego”

- **Strategia akcji – przedstawiona w ramach wykładu**

- Identyfikacja potencjalnych zagrożeń
- Oszacowanie prawdopodobieństwa ich wystąpienia
- Oszacowanie skali ewentualnych skutków
- Opracowanie planu kontrolowania zagrożeń w celu unikania ryzyka wystąpienia
- Opracowanie planów awaryjnych – skuteczna reakcja podczas wystąpienia zagrożenia

5.1. Rodzaje zagrożeń dla projektów - klasyfikacja

• Pierwsza klasyfikacja

- Zagrożenia projektowe – utrudnienie realizacji produktu
- Zagrożenia techniczne – wpływ na jakość produktu
- Zagrożenia ekonomiczne – utrudnienie sukcesu rynkowego
 - Dobry produkt, którego nikt nie potrzebuje (ryzyko marketingowe)
 - Powstanie produktu, który nie odpowiada profilowi firmy (ryzyko strategiczne)
 - Firma nie potrafi sprzedać swojego produktu
 - Utrata zainteresowania produktem przez zarząd firmy (ryzyko zarządzania)
 - Zmniejszenie budżetu lub liczby wykonawców (ryzyko budżetowe)

• Druga klasyfikacja

- Znane (nierealistyczny termin, brak dokładnej specyfikacji wymagań)
- Przewidywalne (rotacja personelu, zła komunikacja z klientem)
- Nieprzewidywalne (losowe)

• Trzecia klasyfikacja

- Ogólne – każdy projekt
- Specyficzne – związane z konkretnym projektem

5.2. Identyfikacja zagrożeń

- **Lista kontrolna zagrożeń - zawartość**
 - Wielkość produktu
 - Wpływ działalności handlowej
 - Cechy klienta
 - Określenie procesu wytwórczego
 - Środowisko tworzenia aplikacji
 - Technologia
 - Wielkość i doświadczenie zespołu
- **Forma**
 - **Lista pytań** i odpowiedzi
 - Lista czynników powodujących zagrożenia
 - Lista źródeł i składników ryzyka z określeniem prawdopodobieństwa wystąpienia (funkcjonalnością, pielęgnacją, kosztami i harmonogramami)

5.2. (cd) **Lista pytań** – wg wpływu na powodzenie projektu

1. Czy kierownictwo firmy formalnie popiera realizację przedsięwzięcia?
2. Czy użytkownicy są entuzjastycznie nastawieni do tworzenia produktu ?
3. Czy wykonawcy i klienci dobrze rozumieją wymagania stawiane produktowi ?
4. Czy klienci brali udział w przygotowaniu specyfikacji wymagań ?
5. Czy oczekiwania użytkowników produktów są realistyczne ?
6. Czy zakres działania produktu jest nieznany ?
7. Czy członkowie zespołu mają wszystkie potrzebne umiejętności ?
8. Czy wymagania stawiane produktowi są niezmiennie ?
9. Czy członkowie zespołu mają doświadczenie w stosowaniu technologii użytej do tworzenia produktu ?
10. Czy wielkość zespołu odpowiada potrzebom produktu ?
11. Czy klienci i użytkownicy uważają , że produkt jest ważny i stawiają takie same wymagania produktowi ?

5.3. Źródła i składniki ryzyka - Tab.1

Składnik		Funkcjonalność	Pielęgnacja	Koszt	Harmonogram
Kategoria					
Katastrofalne	1	Niespełnienie wymagania - klęska całego projektu		Niepowodzenie = opóźnienie i straty finansowe 100%	
	2	Znaczne ograniczenie funkcjonalności lub całkowite unieruchomienie produktu	Oprogramowania nie można uruchomić lub pielęgnować	Braki finansowe, przekroczenie budżetu	Nie można dotrzymać terminu realizacji
Krytyczne	1	Niepełnienie wymagania ogranicza funkcjonalność produktu = sukces projektu wątpliwy		Niepowodzenie = opóźnienie i straty 20-100%	
	2	Istotne ograniczenia funkcjonalności	Niewielkie opóźnienia we wprowadzaniu modyfikacji	Braki finansowe, możliwość przekroczenia budżetu	Możliwe opóźnienia

5.3 (cd) Źródła i składniki ryzyka - Tab.1

Składnik		Funkcjonalność	Pielęgnacja	Koszt	Harmonogram
Kategoria					
Marginalne	1	Niespełnienie wymagania – degradacja jednej z drugorzędnych funkcji produktu		Niepowodzenie = opóźnienie i straty finansowe 1-20%	
	2	Niewielkie ograniczenie funkcjonalności	Akceptowalna szybkość i jakość pielęgnacji	Wystarczające środki finansowe	Realistyczne terminy zakończenia projektu
Pomijalne	1	Niepełnienie wymagania = niewygodna lub inne kłopoty nie związane z działaniem produktu		Niewielkie opóźnienia lub straty nie przekraczające 0.2%	
	2	Brak ograniczenia funkcjonalności	Łatwa pielęgnacja	Możliwe oszczędności	Możliwe zakończenie prac przed terminem

1 – Konsekwencja wystąpienia awarii i niewykrytych błędów

2 – Konsekwencja braku osiągnięcia założonych celów

Tab.2 Tabela zagrożeń – sortowanie wg prawdopodobieństwa i skutków wystąpienia zagrożeń podczas szacowania ryzyka

Zagrożenie	Kategoria	Prawdopodobieństwo	Skutki	Plan
Zaniżony szacunek wielkości produktu	Wielkość produktu	60%	2	
Większa liczba użytkowników	Wielkość produktu	30%	3	
Trudności z ponownym użyciem komponentów	Wielkość produktu	70%	2	
Opór użytkowników	Rynek	40%	3	
Zmiana terminów	Rynek	50%	2	
Utrata źródeł finansowania	Cechy klienta	40%	1	
Zmiana wymagań klienta	Wielkość produktu	80%	2	
Technologia nie spełnia oczekiwań	Technologia	30%	1	
Brak niezbędnych szkoleń na temat narzędzi	Środowisko tworzenia aplikacji	80%	3	
Niedoświadczeni pracownicy	Doświadczenie zespołu	30%	2	
Duża rotacja personelu	Doświadczenie zespołu	60%	2	
.....				

Skutki: 1- katastrofalne, 2 – krytyczne, 3-marginalne, 4- pomijalne wg Tab.1

5.4. Szacowanie ryzyka – procedura

1. Ustalenie stopnia przewidywanego prawdopodobieństwa wystąpienia każdego zagrożenia
2. Szacowanie skutków wystąpienia
3. Ocena ryzyka - Szacowanie skutków wystąpienia zagrożenia na przebieg projektu i własności produktu
4. Ocena dokładności oszacowania

Odp. p.1 (p. 5.4). Wyznaczanie prawdopodobieństwa wystąpienia każdego zagrożenia

- **Procedura 1:**
 - Korzystanie z jakościowej skali: niemożliwe, mało prawdopodobne, prawdopodobne i częste
 - Przypisanie pewnego matematycznego prawdopodobieństwa np. od 0.7 do 1 dla pojęcia częste
- **Procedura 2:**
 - Sporządzanie indywidualnej prognozy i
 - Wyznaczenie wartości średniej

Odp. p.2 (p. 5.4). Szacowanie skutków wystąpienia zagrożeń

- **Procedura:**

1. Ustalić średnie prawdopodobieństwo wystąpienia zagrożenia dla każdego składnika (**do p.1**)
2. Ustalić wpływ zagrożenia na każdy składnik ryzyka (**do p.2, tab. 1**)
3. Uzpełnić tabelę zagrożeń i przeanalizować ją (**tab. 2**)

Podatność na zagrożenie:

$$RE = P * C$$

P – prawdopodobieństwo wystąpienia zagrożenia

C – koszt strat poniesionych w wyniku wystąpienia zagrożenia

Odp.p.2 (p. 5.4). Szacowanie skutków wystąpienia zagrożeń – Przykład 1

Identyfikacja zagrożenia - 70% komponentów można użyć

P = Prawdopodobieństwo wystąpienia - 80%

100% komponentów - 60 komponentów

70% komponentów nadaje się do użycia - 42 komponentów

30% komponentów należy utworzyć od podstaw- **18 komponentów**

Średnia wielkość komponentu - **100 LOC**,

Koszt wytworzenia 1 LOC – **14 USD/LOC**,

C = Całkowity koszt utworzenia: $18 * 100 * 14 = 25200$ USD

Podatność na zagrożenie:

$RE = P * C = 0.8 * 25200 = 20160$ USD

Odp. p.3 (p. 5.4) cd. Ocena ryzyka - Szacowanie skutków wystąpienia zagrożenia na przebieg projektu i własności produktu

Lista informacji o zagrożeniach:

$[r_i, l_i, x_i]$

gdzie :

r_i - i-te zagrożenie

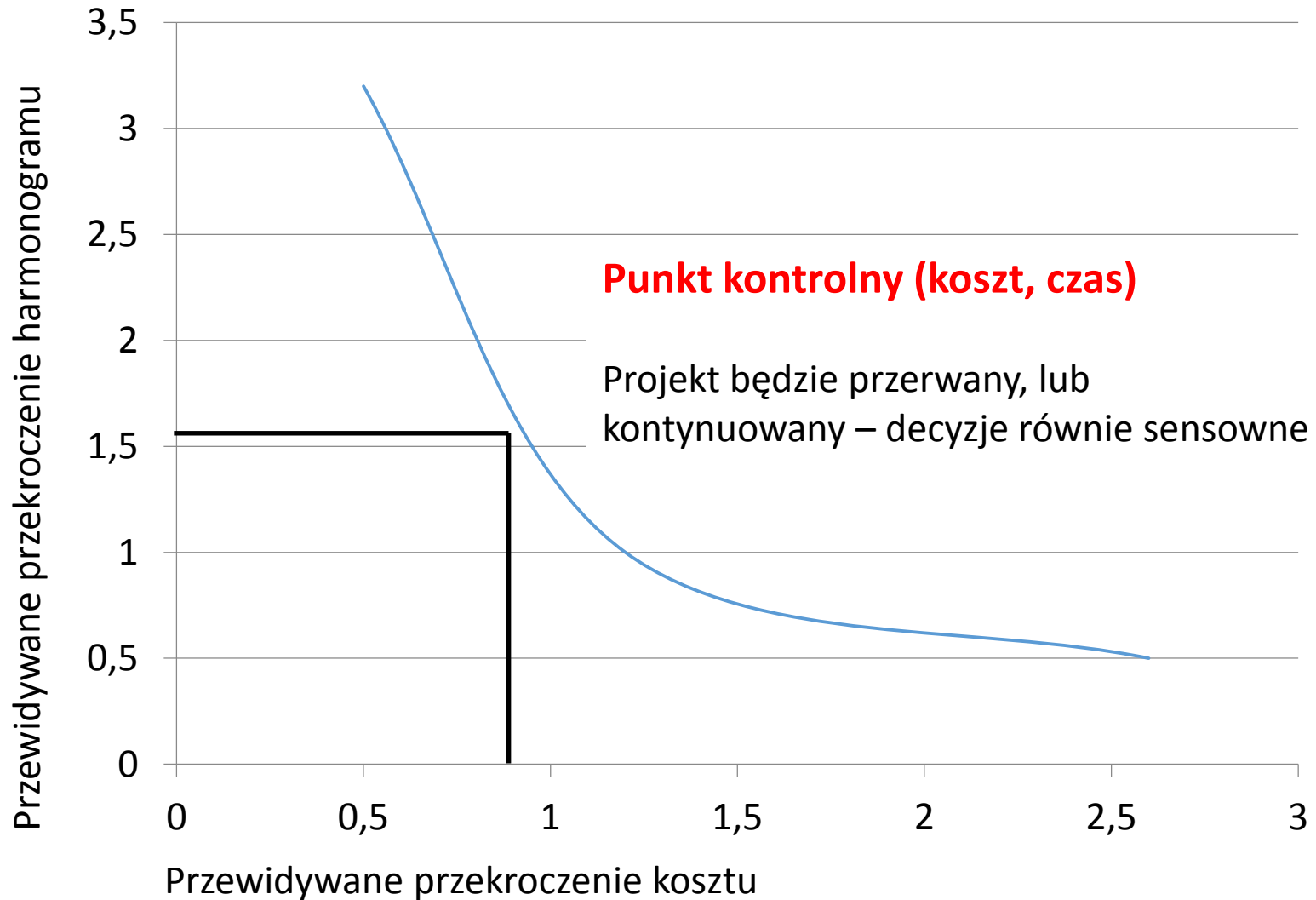
l_i - prawdopodobieństwo jego wystąpienia

x_i - skutki zagrożenia

Procedura:

1. Zweryfikowanie uzyskanych prognoz
2. Uporządkowanie zidentyfikowanych zagrożeń
3. Wstępne sposoby uniknięcia lub kontrolowania tych zagrożeń, które prawdopodobnie wystąpią

Odp. p.3 (p. 5.4) cd. Ocena ryzyka – zakres dopuszczalnego ryzyka – wyznaczenie punktu kontrolnego zazwyczaj trudne do wyznaczenia



Odp. p.3 (p. 5.4) cd. Ocena ryzyka – zakres dopuszczalnego ryzyka – procedura pomijająca wyznaczenie punktu kontrolnego

1. Zdefiniować zakresy dopuszczalnego ryzyka
2. Spróbować porównać każdą z uzyskanych trójek $[r_i, l_i, x_i]$ z odpowiednimi zakresami
3. Ustalić zbiór punktów kontrolnych, które wyznaczą obszar obejmujący sytuacje prowadzące do przerwania projektu. Będzie on ograniczony linią lub obszarem niepewności
4. Spróbować oszacować, czy kombinacja różnych zagrożeń może doprowadzić do przekroczenia zakresu dopuszczalnego ryzyka

Odp. p.4 (p. 5.4) Uściślanie zagrożeń

Przyczyny → **Skutek**

Przykład 1 (odp. p.2), cd.

Przyczyny:

1. Część komponentów poza standardem
2. Brak jasnego standardu wykonania komponentów
3. Część komponentów dedykowana innemu środowisku

Skutek:

- 30% komponentów należy utworzyć od podstaw

Dodatek - Prognozowanie **wielkości** **oprogramowania**

- Modele prognostyczne
- Model COCOMO (*Constructive Cost MOdel*) II

1. Struktura modeli prognostycznych

$$E = A + B * (ev)^C$$

gdzie

A, B, C – stałe ustalone doświadczalnie

E – pracochłonność

ev – zmienna prognostyczna: miara wielkości
oprogramowania LOC lub FP

Miary LOC i FP wyznacza się metodami podanymi wcześniej, ale zamiast porównywania ich z danymi historycznymi, można je podstawić do wzoru stanowiącego dany model prognostyczny

1.1. Modele oparte na LOC

Nazwa modelu	Wzór [osobomiesiące]
Walstona-Felixa	$E = 5.2 * (KLOC)^{0.91}$
Baileya-Basiliego	$E = 5.5 + 0.73 * (KLOC)^{1.16}$
Uproszczony model Boehma	$E = 3.2 * (KLOC)^{1.05}$
Doty'ego dla KLOC >9	$E = 5.288 * (KLOC)^{1.047}$

Modele należy przystosować do potrzeb produktu w projekcie

1.2. Modele oparte na FP

Nazwa modelu	Wzór [osobomiesiące]
Albrechta-Gaffneya	$E = -13.39 + 0.0545 FP$
Kemerera	$E = 60.62 * 7.728 + 10^{-8} FP^3$
Matsona-Barnetta-Mellichampa	$E = 585.7 + 15.12 FP$

Modele należy przystosować do potrzeb produktu w projekcie

2. Model COCOMO (*Constructive Cost MOdel*) II

m- typy modeli odzwierciedlających dojrzałość produktu

- **Model struktury aplikacji**

- Początkowa faza procesu
- Tworzenie prototypu interfejsu graficznego
- Analiza metod komunikacji oprogramowania z otoczeniem
- Szacowanie efektywności produktu i dojrzałości stosowanej technologii

- **Początkowy model projektowy**

- po ustaleniu wymagań oprogramowania
- Ustaleniu architektury

- **Model gotowej architektury**

- Faza konstruowania oprogramowania

2.1. Model COCOMOII - wagi złożoności dla różnych elementów produktu (Waga)

Rodzaj obiektu	Waga		
	Prosty	Średni	Złożony
Ekran	1	2	3
Raport	2	5	8
Komponent			10

Procedura wyznaczania punktów obiektowych

1. Określenie stopnia złożoności elementów produktu (**Prosty**, **Średni** lub **Złożony**)
2. Należy przydzielić **wagi** wg podanej tabeli (**Waga**)
3. Całkowitą liczbę punktów obiektowych **PO** oblicza się mnożąc liczbę elementów obiektowych (określonych przez **Rodzaj obiektu**) przez **współczynniki wagowe** (**Waga** wg podanej wyżej tabeli) i sumując uzyskane wyniki
4. W przypadku ponownego użycia komponentów należy wyznaczyć **NPO** - nową liczbę punktów obiektowych (poniżej)

$NPO = PO * [(100 - R)/100]$ – nowa miara punktów obiektowych,
gdzie R – liczba powtórnie używanych komponentów

2.2. Model COCOMOII – oszacowanie pracochłonności E

Doświadczenie /umiejętności pracowników	Bardzo małe	Małe	Przeciętne	Duże	Bardzo duże
Dojrzałość/ Bogactwo środowiska	Bardzo mała	Mała	Przeciętna	Duża	Bardzo duża
W	4	7	13	25	50

- Wyznaczenie pracochłonności **E**:
 - Ustalenie współczynnika wydajności pracy **W** (wg powyższej tabeli):

$$W = NPO / \text{osobomiesiące}$$
 - Oszacowanie pracochłonności **E [osobomiesiące]**

$$E = NPO / W$$