



# Metoda inspekcji modeli obiektowych

*Aleksander Jarzębowicz*

[olek@eti.pg.gda.pl](mailto:olek@eti.pg.gda.pl)

Katedra Inżynierii Oprogramowania  
Politechnika Gdańska

Seminarium Katedralne 5 luty 2008

- 1 -



## Plan prezentacji

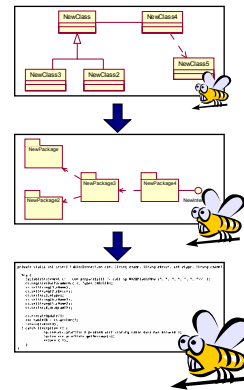
- **Motywacja: defekty**
- **Inspekcje oprogramowania**
- **Metoda UML-HAZOP**
  - Detekcja defektów
  - Proces inspekcji
  - Wspomaganie narzędziowe
- **Walidacja metody**
  - Studia przypadków
  - Kontrolowane eksperymenty
- **Podsumowanie**

- 2 -



## Defekty w cyklu życia oprogramowania

- W ramach projektu informatycznego powstaje szereg artefaktów projektowych
- Defekty pojawiają się już w najwcześniejszych etapach wytwarzania
- Jeśli nie zostaną wykryte, przenoszą się do kolejnych artefaktów (i tam często dalej „rozwijają”)
- Poprawienie defektów staje się coraz trudniejsze i wymaga coraz większych nakładów pracy
- **Wniosek: warto jak najwcześniej podjąć działania ukierunkowane na wykrywanie defektów**



- 3 -



## Wykrywanie defektów

- Istnieją dwa podstawowe sposoby wykrywania defektów:
  - Analiza dynamiczna (przede wszystkim testowanie)
  - Analiza statyczna (bez wykonywania kodu)
- Testowanie możliwe dopiero w końcowych etapach cyklu wytwarzania – kiedy dostępny jest kod, analiza statyczna możliwa w każdym etapie
- Analiza statyczna – przede wszystkim przeglądy
  - Inne np. formalne dowody poprawności, interpretacje symboliczne
- Przeglądy – kontrole artefaktów ukierunkowane na wykrywanie defektów prowadzone przez ludzi



- 4 -



## Dlaczego przeglądy? - Motywacja

- Ludzie popełniają błędy i wprowadzają defekty do tworzonych przez siebie dokumentów
- Mechanizacja wykrywania i usuwania tych defektów jest możliwa jedynie w ograniczonym zakresie
- Niezbędny jest więc wysiłek człowieka w celu wykrycia i usunięcia tych defektów
- Idea przeglądów:
  - Najtrudniej jest zauważyć własne błędy, dlatego dobrze jest, by poszukał ich ktoś inny



- 5 -



## Przeglądy

- Przegląd to działanie zespołowe, ukierunkowane na:
  - wykrycie (szeroko rozumianych) defektów w dokumencie, którego autor jest członkiem zespołu
  - potwierdzenie dobrej jakości dokumentu w zakresie nie wymagającym poprawy
- Przegląd danego produktu polega na:
  - przeanalizowaniu go (indywidualnie bądź zespołowo) przez grupę osób niezaangażowanych w jego wytworzenie,
  - rejestrację zauważonych defektów i innych wątpliwości,
  - przekazanie wyników autorowi, który dokonuje niezbędnych poprawek

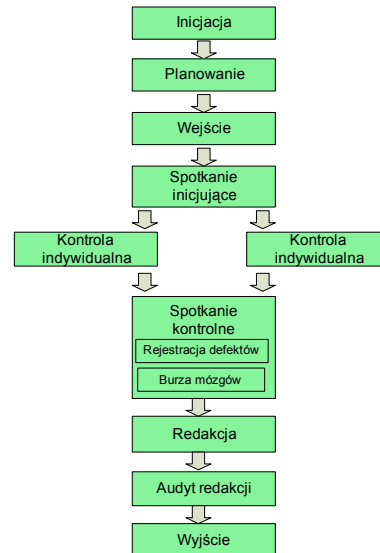


- 6 -



## Inspekcja oprogramowania

- Inspekcja - najbardziej sformalizowany rodzaj przeglądu
  - Zdefiniowane mechanizmy detekcji
  - Sformalizowany proces
  - Ustalone parametry inspekcji (np. tempo kontroli)
  - Statystyczna kontrola procesu



- 7 -



## Kontrola indywidualna - mechanizm detekcji

- Ad hoc – brak wspomaganie ale i brak ograniczenia zakresu
- Lista kontrolna (ang. *checklist*) – lista zagadnień do sprawdzenia przez inspektora
  - Czy wszystkie rozmiary tablic są zadeklarowane jako stałe?
  - Sprawdź czy wszystkie zmienne są zainicjalizowane
- Scenariusze do wykonania (ang. *scenario-based reading*) – scenariusze zgodnie z którymi ma działać inspektor
  - Dodatkowe zadania mające za cel umożliwienie inspektorowi lepszemu zrozumienia kontrolowanego artefaktu
  - Przykład: *Perspective-Based Reading* – kontrola specyfikacji wymagań z perspektywy projektanta, testera, użytkownika

- 8 -



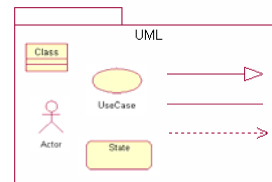
## Metoda UML-HAZOP

- 9 -



## Przedmiot analiz: modele UML

- Unified Modeling Language można uznać za nieformalny, ale faktyczny standard w praktyce przemysłowej
- UML pokrywa szeroki zakres procesu wytwarzania – również wczesne etapy gdzie uzysk z eliminacji defektów jest największy
- Stopień formalizacji UML pozwala na precyzyjniejsze opracowanie kryteriów inspekcji w odniesieniu do zdefiniowanych elementów notacji
- Inspekcje modeli UML (i ogólnie modeli obiektowych) nie zostały jak dotąd opracowane i przebadane w takim stopniu jak np. inspekcje wymagań czy kodu źródłowego



- 10 -

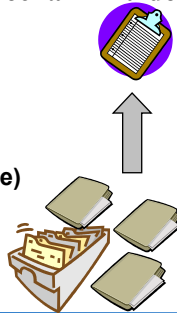


## Inspekcje modeli obiektowych

- Inspekcja musi być dostosowana do rodzaju produktów, których dotyczy np. modeli obiektowych
- Kluczową kwestią jest mechanizm detekcji (jakich defektów poszukiwać i w jaki sposób)
- Opracowanie mechanizmu detekcji:

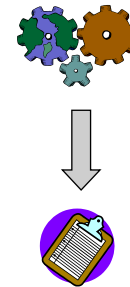
### BOTTOM-UP

(zebranie informacji o defektach z zakończonych projektów i uogólnienie)



### TOP-DOWN

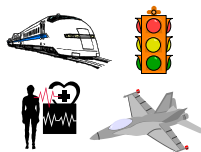
(systematyczne wygenerowanie wszystkich kategorii defektów w danym zakresie)



- 11 -



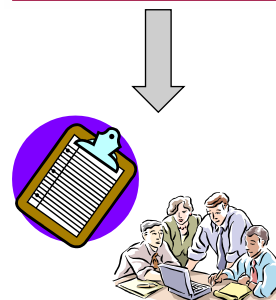
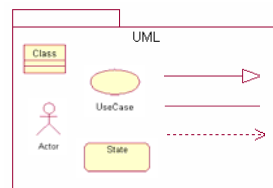
## Metoda UML HAZOP



Bezpieczeństwo systemów (ang. safety)

### Metoda HAZOP

HAZOP guidewords
NO
MORE
LESS
AS WELL AS
PART OF
REVERSE
OTHER THAN
EARLY
LATE
BEFORE
AFTER



- 12 -



## Geneza metody HAZOP



- Hazard and Operability Studies
- Opracowana w latach 60 na potrzeby przemysłu chemicznego
- Wykrywanie sytuacji zagrażających bezpieczeństwu (tzw. hazardów) w instalacjach fabryk chemicznych
- Systematyczny przegląd diagramu z projektem instalacji i jego poszczególnych elementów i ich atrybutów
- Wykorzystanie „słów kluczowych” – krótkich sformułowań sugerujących analitykowi w bardzo ogólny sposób rodzaj defektu

- 13 -



## Słowa kluczowe

<i>Słowo kluczowe</i>	<i>Ogólna interpretacja</i>
NO	Kompletny brak zamierzonego rezultatu przy braku innych, „zastępczych” efektów
MORE	Zbyt wysoka wartość parametru ilościowego
LESS	Zbyt niska wartość parametru ilościowego
AS WELL AS	Oprócz zamierzonego rezultatu otrzymano inny, dodatkowy
PART OF	Zamierzony rezultat został uzyskany tylko częściowo
REVERSE	Otrzymano logiczne przeciwieństwo zamierzonego rezultatu
OTHER THAN	Brak zamierzonego rezultatu, ale otrzymano inny, odmienny
EARLY	Zdarzenie zachodzi wcześniej niż zamierzono
LATE	Zdarzenie zachodzi później niż zamierzono
BEFORE	Zdarzenie zachodzi przed innym zdarzeniem, w stosunku do którego miało być późniejsze
AFTER	Zdarzenie zachodzi po innym zdarzeniu, w stosunku do którego miało być wcześniejsze

- 14 -



## Adaptacja HAZOP na potrzeby inspekcji oprogramowania

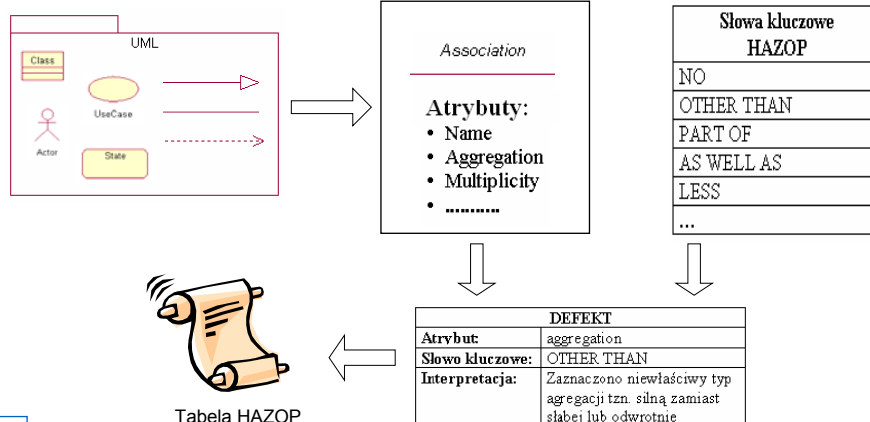
- **Obserwacja 1:** Słowa kluczowe mogą być wykorzystane także do innych celów np. sugerowania defektów projektowych (zamiast sytuacji niebezpiecznych)
- **Obserwacja 2:** Słowa kluczowe zastosowane do elementów diagramów UML pozwalają na wykrywanie defektów obecnych w tych diagramach

Guideword
NO
MORE
LESS
AS WELL AS
PART OF
REVERSE
OTHER THAN
EARLY
LATE
BEFORE
AFTER

- 15 -



## Zastosowanie HAZOP do analizy diagramów UML



- 16 -





## Przykłady defektów (1)

Atrybut: Association.aggregation	
Słowo kluczowe	Interpretacja
NO	Nie zaznaczono agregacji, która powinna wystąpić
AS WELL AS	Zaznaczono agregację, która w rzeczywistości nie ma miejsca
OTHER THAN	Zaznaczono niewłaściwy typ agregacji tzn. silną zamiast słabej lub odwrotnie
MORE	Agregacja obejmuje zbyt wiele klas składowych, niektóre nie powinny wchodzić w jej skład
LESS	Agregacja nie obejmuje niektórych klas (obecnych na diagramie lub też nie), które powinny wchodzić w jej skład

- 17 -



## Przykłady defektów (2)

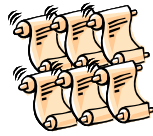
CLASS		
Atrybut	Słowo kluczowe	Interpretacja
Class	AS WELL AS (1)	Klasa nie powinna znajdować się na diagramie i należy ją usunąć.
	AS WELL AS (2)	Klasa jest zbyt złożona, powinna zostać rozbita na dwie lub więcej osobnych klas.
	PART OF	Klasa jest tożsama z inną klasą obecną na diagramie. Te dwie klasy powinny zostać połączone w jedną.
Class.Name	OTHER THAN	Nazwa klasy nie oddaje jej znaczenia i powinna zostać zmieniona.
Class.AttributeSet	PART OF	Brak atrybutu lub atrybutów klasy, które klasa powinna posiadać.
Class.Attribute	AS WELL AS	Atrybut jest nadmiarowy i powinien zostać usunięty.
	OTHER THAN	Nazwa atrybutu nie odzwierciedla jego znaczenia i powinna zostać zmieniona.
Class.OperationSet	PART OF	Brak operacji (jednej lub wielu), w które powinna być wyposażona ta klasa.
Class.Operation	AS WELL AS	Operacja jest nadmiarowa i powinna zostać usunięta.
	OTHER THAN	Nazwa operacji nie odzwierciedla jej działania i powinna zostać zmieniona.

- 18 -

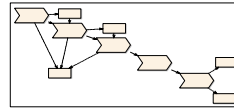


## Metoda UML-HAZOP

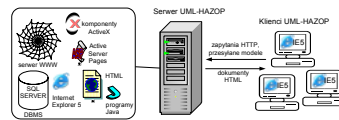
### UML-HAZOP



Tabele HAZOP – mechanizm detekcji



Proces inspekcji



System informatyczny

- 19 -



## Mechanizm detekcji UML-HAZOP

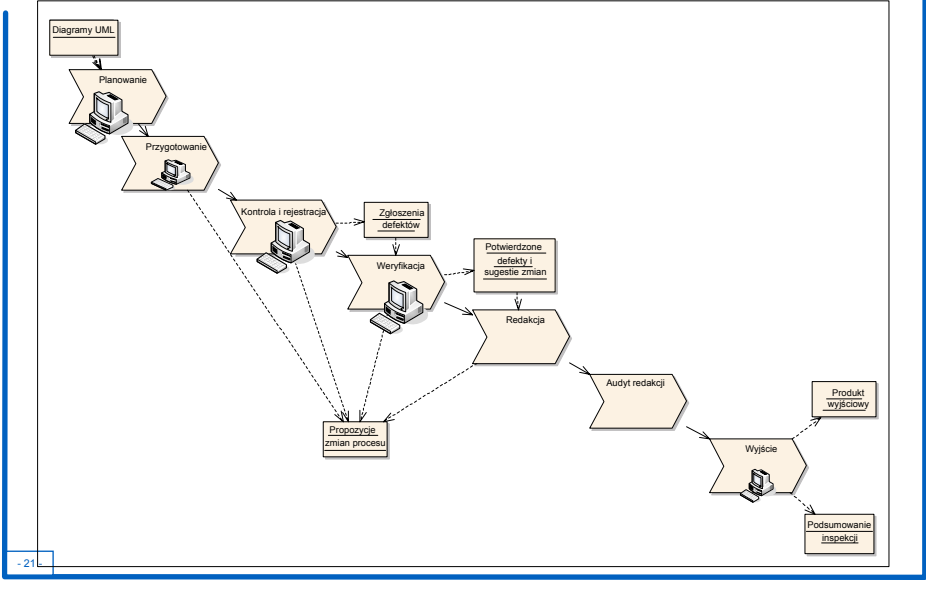
- Podczas inspekcji konkretnego modelu dla każdego elementu tworzona jest kopia tabeli HAZOP
- Zadaniem inspektora jest zaznaczenie dla każdej sugestii defektu decyzji (czy defekt taki występuje dla tego elementu czy też nie)
- Kontrola jest ma postać pracy sekwencyjnej sterowanej przez kolejne elementy i ich potencjalne defekty (nie według ogólnych kategorii defektów)
- Stanowi to zasadniczą różnicę w stosunku do tradycyjnych list kontrolnych zawierających pytania typu:
  - Czy wszystkie związki mają poprawne licznosci?
  - Czy nazwy klas są adekwatne do ich znaczenia?



- 20 -



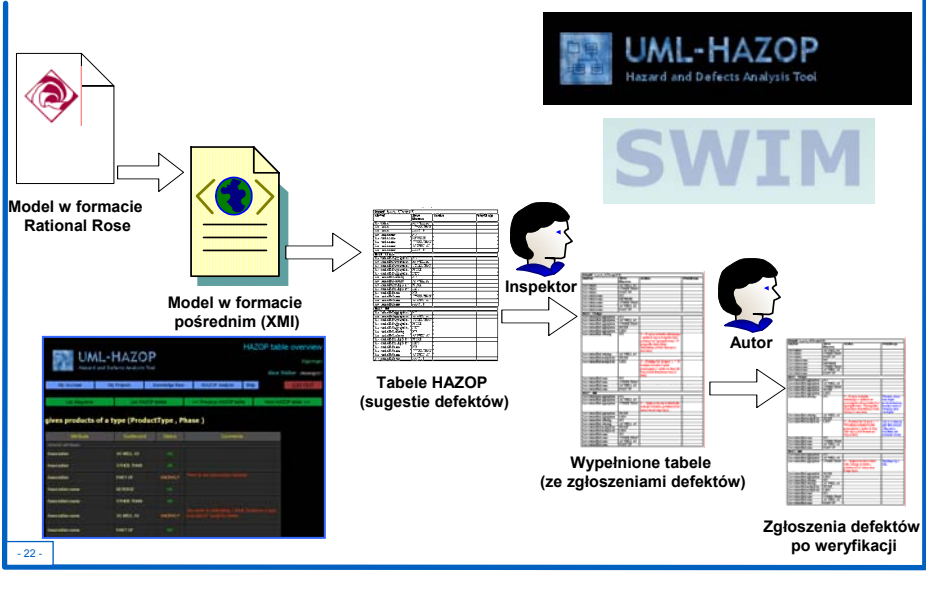
## Proces inspekcji UML-HAZOP



- 21 -



## System wspomagający inspekcję



- 22 -



## Narzędzie wspomagające SWIM

- Aplikacja internetowa w modelu cienki klient – serwer
- Technologia MS .NET
- Rozróżnienie ról
- Wspomaganie procesu kontroli i weryfikacji
- Komunikacja w rozproszonym zespole projektowym
- Wbudowany mechanizm adaptacji
  - dodatkowe diagramy
  - dodatkowe modele defektów
  - możliwość dowolnego definiowania zakresu inspekcji

Question	Result	Explanation
<b>GENERALIZATION SET</b>		
Generalization Set PART OF	OK	
<b>ASSOCIATION SET</b>		
Association Set PART OF	ERROR	There seems to be no association existing between the STEEP and RISE classes.
<b>CLASS SET</b>		
Class Set PART OF	OK	

Question	Result	Explanation	Default	Default comment
<b>ASSOCIATION SET</b>				
Association Set PART OF	ERROR	There seems to be an association existing between the STEEP and RISE classes.	ACCEPTED	This is indeed a very precise defect.
<b>CLASS SET</b>				
Class Set PART OF	unspecified		ACCEPTED	
<b>CLASS ATTRIBUTE</b>				
Class Attribute Set PART OF	unspecified		ACCEPTED	

- 23 -



## Walidacja zaproponowanych rozwiązań

- 24 -



## Walidacja metody UML-HAZOP

- Walidacja – sprawdzenie zaproponowanych rozwiązań na drodze eksperymentalnej
- Walidacja jest konieczna jeśli rozwiązanie ma mieć zastosowanie praktyczne
- Uzyskiwane wyniki stanowią punkt wyjścia do ewolucji, udoskonalenia rozwiązania poddawanego walidacji
- Cele walidacji metody UML-HAZOP:
  - Ocena jej *skuteczności*, tzn. stwierdzenie, w jakim stopniu metoda umożliwi wykrywanie defektów w modelach UML tworzonych we wczesnych etapach rozwoju systemu,
  - Ocena jej *wydajności*, tzn. stwierdzenie jakie nakłady pracy są niezbędne dla wykrycia defektu przy posłużeniu się tą metodą.
- Sposoby walidacji:
  - Studia przypadków
  - Kontrolowane eksperymenty

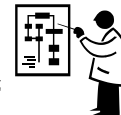


- 25 -



## Walidacja – studia przypadków

- Cztery studia przypadków dotyczące rzeczywistych (zaimplementowanych) systemów – inspekcje diagramów klas
- Możliwość zebrania jedynie liczb bezwzględnych – brak porównania z innymi metodami



Nr studium przypadku	Kontrolowany system	Rozmiar kontrolowanych modeli	Liczba potwierdzonych wykrytych defektów	Nakład (osobogodz.)
CS-1	System bilingowy	17 klas, 14 związków	8	12
CS-2	System wspomagający zarządzanie firmą (podsystem 1)	55 klas, 79 związków	80	45
CS-3	System wspomagający zarządzanie firmą (podsystem 2)	27 klas, 34 związków	39	21
CS-4	System dystrybucji leków w szpitalu	39 klas, 42 związków	81	Brak inf.

- 26 -



## Kontrolowane eksperymenty

- **Udział większej grupy osób**
- **Wykonanie analiz dokumentacji metodą UML-HAZOP i inną metodą inspekcji**
- **Zebranie i porównanie metryk**
- **Zebranie i uwzględnienie opinii uczestników**



- 27 -



## Walidacja – eksperymenty

- **Cele eksperymentów:**
  - Porównanie UML-HAZOP i innych technik inspekcji
  - Uwzględnienie wyników do optymalizacji metody
- **Uczestnicy: studenci 5 roku Wydziału ETI**
- **Wykorzystywane materiały: Diagram klas z opisem elementów (przedmiot analizy), streszczenie SWS (punkt odniesienia)**
- **Inspekcja ograniczona do etapów przygotowania i kontroli indywidualnej**
- **Pomiar liczby defektów wykrytych przez pojedynczego uczestnika i czasu poświęconego na kontrolę**
- **Zaplanowano i wykonano 3 eksperymenty:**
  - Były prowadzone na przestrzeni 3 lat
  - W sumie udział wzięło 89 studentów
  - Każdy eksperyment stanowił spore przedsięwzięcie organizacyjne
  - Problematyka projektowania eksperymentów: zagrożenia, powtarzalność

- 28 -



## Eksperyment EXP-1

- **Cel:** porównanie UML-HAZOP z podejściem Ad hoc
- **Rezultaty:**
  - Duża część defektów poza zakresem tabel UML-HAZOP
  - UML-HAZOP miał słabsze wyniki ogólne, był natomiast dużo lepszy w swoim (ograniczonym) zakresie
  - Pewne trudności w posługiwaniu się metodą przez niedoświadczonych użytkowników
- **Wnioski:**
  - Rozszerzenie zakresu metody
  - Ewolucja tabel HAZOP
  - Opracowanie dodatkowych materiałów szkoleniowych

- 29 -



## Eksperyment EXP-2

**Cel:** porównanie zmodyfikowanego UML-HAZOP z wynikami Ad hoc

- **Rezultaty:**
  - Rozszerzenie zakresu i modyfikacje tabel HAZOP dały pozytywne efekty
  - Osiągane rezultaty bardzo zbliżone do zapamiętanych dla Ad hoc
  - Niewystarczający czas kontroli – średnio jedynie 65% tabel zostało wypełnionych przez inspektora
- **Wnioski:**
  - Obserwacja przyrostu zgłaszanych defektów skłania do hipotezy, że przyrost ten dla UML-HAZOP jest liniowy
  - W przypadku ograniczeń czasowych należy zawęzić zakres kontroli dla pojedynczego inspektora
  - Sprzężenie zwrotne od uczestników wykazało potrzebę dalszej optymalizacji tabel HAZOP

- 30 -



## Eksperyment EXP-3

### Cele eksperymentu:

- Replikacja poprzednich eksperymentów z udziałem większej liczby uczestników
- Zbadanie wyników dla par inspektorów z rozdzielonymi zakresami poszukiwanych defektów
- Ocena skutków rozszerzenia UML-HAZOP o elementy podejścia scenariuszowego (ang. *scenario-based reading*)
- Weryfikacja przypuszczenia o liniowym przyroście defektów
- Zbadanie jaki wpływ na wyniki UML-HAZOP ma ograniczenie czasu dostępnego na kontrolę

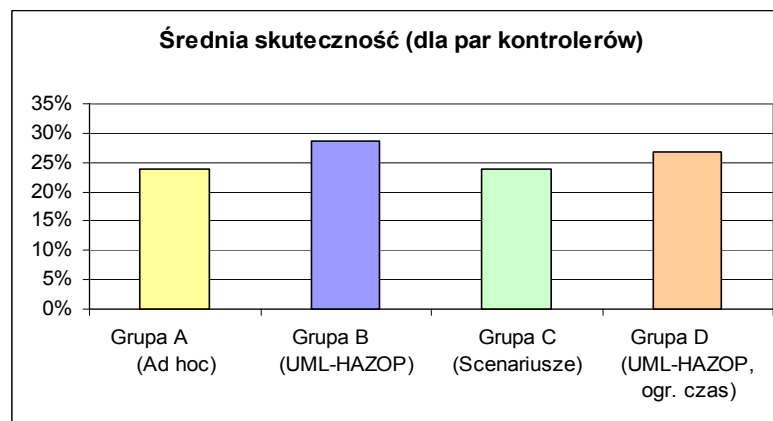
### Grupy eksperymentalne:

- Grupa A: Ad hoc
- Grupa B: UML-HAZOP (zwykły)
- Grupa C: Podejście scenariuszowe
- Grupa D: UML-HAZOP (ograniczony czas)

- 31 -



## Wyniki EXP-3 (1)



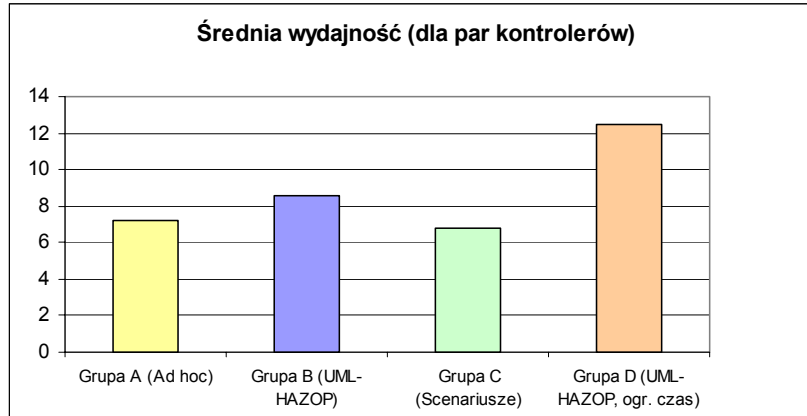
- Skuteczność – jaki procent wszystkich defektów wykryto

- 32 -





## Wyniki EXP-3 (2)

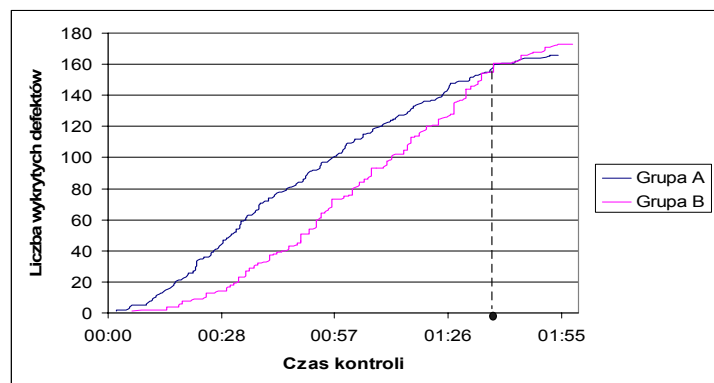


- Wydajność - liczba defektów wykrytych na 1 godz. kontroli
- Wynik UML-HAZOP wyższy o 73%

- 33 -



## Wyniki EXP-3 (3)



Liczba wykrywanych defektów jako funkcja czasu

- 34 -



## Podsumowanie

### Wyniki uzyskane do tej pory:

1. Zaproponowanie nowej metody inspekcji diagramów UML polegającej na zastosowaniu znanej metody HAZOP do systematycznej identyfikacji defektów
2. Zaproponowanie struktury i organizacji procesu inspekcji
3. Zaprojektowanie i implementacja informatycznego narzędzia wspomagającego
4. Wykorzystanie wyników inspekcji w budowie dowodów zaufania uzasadniających wiarygodność systemów
5. Realizacja czterech studiów przypadków zastosowania metody UML-HAZOP w odniesieniu do rzeczywistych systemów (w tym wytworzonych w warunkach przemysłowych)
6. Zaplanowanie, organizacja i realizacja trzech kontrolowanych eksperymentów

- 35 -



## Perspektywy

### Perspektywy dalszych prac:

1. Adaptacja metody HAZOP do innych modeli, przede wszystkim modeli procesów biznesowych
2. Dalsza optymalizacja procesu UML-HAZOP na drodze kontrolowanych eksperymentów
3. Wdrożenie metody i narzędzia w organizacji wytwarzającej oprogramowanie (granty badawczo-rozwojowe KBN)

- 36 -



**Dziękuję za uwagę**

**Pytania ?**