

Podstawy testowania

dr hab. inż. Michał Śmiałek, prof. uczelni
dr inż. Kamil Rybiński



**Wydział
Elektryczny**
POLITECHNIKA WARSZAWSKA

Inżynieria oprogramowania



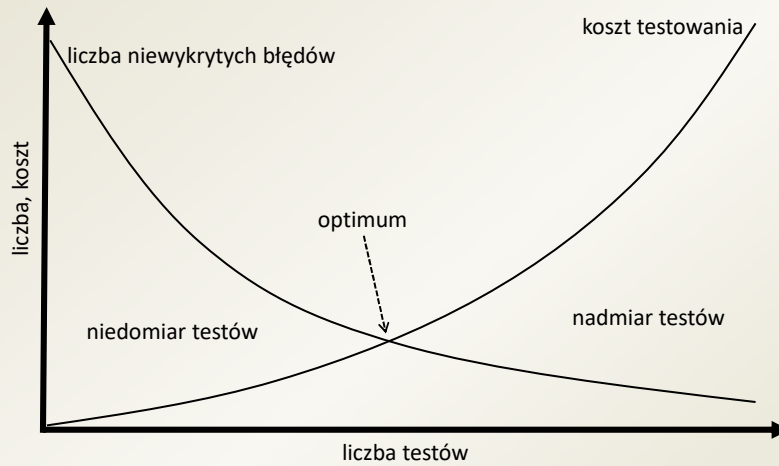
1

Błędy oprogramowania

- Odstępstwa od ustalonej specyfikacji
 - Oprogramowanie nie wykonuje czegoś, co powinno wykonywać
 - Oprogramowanie robi coś, czego nie powinno robić
 - Oprogramowanie nie wypełnia warunków dotyczących cech pozafunkcjonalnych (szybkość działania, łatwość użycia itd.)
 - Oprogramowanie zachowuje się w sposób nie przewidziany specyfikacją
- Pluskwa: usterka w kodzie programu wynikająca z nieuwagi lub przeoczenia programisty

2

Koszty a efektywność testów



- Równowaga między środkami poświęconymi na testowanie a liczbą wykrytych błędów

Badanie jakości testowania

- Metoda posiewowa
 - Wprowadzenie do kodu pewnej liczby sztucznie wytworzonych błędów
 - B - liczba wykrytych błędów
 - b - liczba posianych błędów, które zostały wykryte
 - P - liczba posianych błędów
 - szacunkowa liczba błędów:
 - $(B - b) * P/b$
 - szacunkowa liczba wykrytych błędów:
 - $(B - b) * (P/b - 1)$

Testy czarnej skrzynki

- Ocena działania programu jedynie na podstawie obserwacji danych wejściowych i wyjściowych
- Brak ingerencji oraz wiedzy o kodzie programu
- Badanie programu z punktu widzenia jego użytkowników
- Porównanie programu ze specyfikacją (np. wymaganiami)
- Techniki wykorzystywane w metodach czarnej skrzynki
 - Tworzenie klas równoważności, warunki graniczne, zmiany stanu, niedoświadczony użytkownik

Metoda klas równoważności

- Brak możliwości przebadania ogromnej liczby możliwych stanów systemu
- Określenie zbioru zadań testowych
 - Odpowiednie pokrycie równoważnych stanów
 - Dobranie optymalnej liczby klas równoważności
 - Identyfikacja klas równoważności
 - Pokrycie testami wartości domyślnych, zerowych, początkowych
 - Pokrycie testami wartości błędnych

Metoda warunków granicznych

- Określenie podziałów przyjmowanych przez dane wejściowe
 - Zakresy dopuszczalnych wartości danych i granice tych zakresów
- Sprawdzenie zachowania się systemu dla danych w okolicy podziałów
 - Dane mieszczące się w zakresie jak i spoza zakresów
- Przykład: kontrola zakresu liczbowego $\langle a, b \rangle$
 - Sprawdzamy działanie programu dla $x=a-1$, a , $a+1$, $b-1$, b , $b+1$
 - Sprawdzamy wybrane wartości z wnętrza przedziału, np. $x = (a+b)/2$, $a-2$

Metoda zmian stanów

- Liczba przejść między stanami rośnie lawinowo wraz z liczbą stanów
 - Problem komiwojażera (5 miejscowości, 120 dróg)
- Stworzenie mapy stanów (np. diagram maszyny stanów)
- Każdy stan powinien być odwiedzony przynajmniej raz
 - Badania często wykorzystywanych przejść między stanami
 - Badania stanów awaryjnych
- Konieczne wykorzystanie narzędzi do testów automatycznych
 - Możliwość wprowadzenia elementów losowości
 - Testowanie dla zmiennego obciążenia systemu

Metoda niedoświadczzonego użytkownika

- Uczestnictwo realnych użytkowników
- Osoby bez doświadczenia z systemem
- Wykorzystanie systemu w niestandardowy, nieoczekiwany sposób
- Testerzy dostają zadania odpowiadające przypadkom użycia systemu
 - Zadania wykonywane zgodnie z intuicją użytkowników
 - Możliwe wprowadzanie niewłaściwych danych itp.
 - Cenne wskazówki dotyczące oceny systemu przez użytkowników

Testy szklanej (białej) skrzynki

- Wykorzystują wiedzę o kodzie programu
- Badanie przebiegów przez kod programu
- Formalne i nieformalne przeglądy kodu
- Wykorzystanie analizatorów kodu
- Wykorzystanie modeli projektowych
 - Diagramy sekwencji opisują działanie kodu
 - Diagramy czynności opisują algorytmy

Techniki pokrycia instrukcji i gałęzi

- Przejście przez wszystkie instrukcje kodu przynajmniej raz
- Przejście przez wszystkie gałęzie wychodzące z instrukcji warunkowych
- Wykorzystanie diagramów blokowych
 - Pokazanie możliwych ścieżek wykonania kodu

Technika pokrycia warunków; testowanie pętli

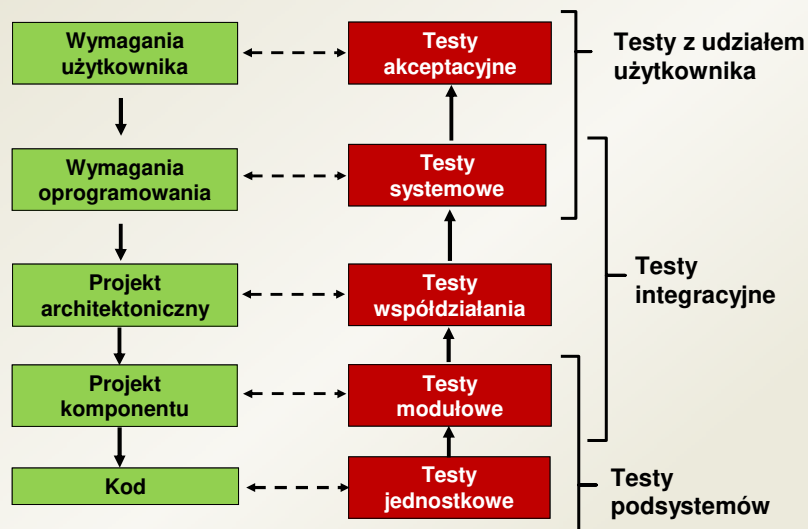
- Sprawdzenie działania kodu dla różnych kombinacji warunków w instrukcjach warunkowych
- Przykład: `if (x==0 || y==0)`
 - Sprawdzamy dla $x=0$ i $y=0$
 - Sprawdzamy dla $x=0$ i $y=2$ oraz $x=3$ i $y=0$
 - Dodatkowo sprawdzamy dla $x=2$ i $y=3$
- Ogólnie: sprawdzamy działanie dla wszystkich kombinacji wartości wyrażeń logicznych
- Sprawdzenie działania pętli
 - Szczególnie istotne pierwsze i ostatnie przebiegi
 - Wyjście z pętli przez zakończeniem
 - Próba wykonania większej liczby iteracji

Technika ścieżek bazowych

- Określenie złożoności cyklicznej kodu
- Wykorzystanie diagramu czynności
 - Węzły - instrukcje; przejścia - kolejność wykonania
- Złożoność liczymy na podstawie liczby węzłów decyzyjnych, węzłów złączenia oraz regionów
 - Region: obszar zamknięty krawędziami łączącymi węzły
- Złożoność cykliczna determinuje liczbę niezależnych ścieżek wykonania kodu
 - Określamy liczbę testów niezbędnych do pokrycia wszystkich instrukcji w danym fragmencie kod

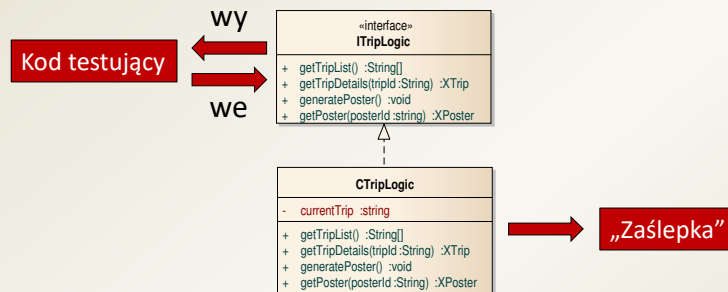
13

Poziomy testowania



14

Testowanie z użyciem zaślepek



- „Zaślepka”: kod podający standardowe dane testowe

Testowanie przypadków użycia

- Scenariusze testowe
 - Opis kolejnych kroków interakcji użytkownika z systemem
 - Metoda czarnej skrzynki
 - Wykorzystanie scenariuszy przypadków użycia
 - Złożenie scenariuszy dla kilku przypadków użycia
- Scenariusze pozytywne i negatywne
 - Sprawdzenie pozytywnego działania systemu
 - Wyeliminowanie niechcianego działania systemu

Przykładowe scenariusze testowe

ST213: Dodaj model samochodu – sukces (POZYTYWNY)

A. Wyświetl listę modeli samochodów – główny

1. Kierownik wybiera przycisk Lista modeli samochodów
2. System pobiera listę modeli samochodów
3. System wyświetla okno listy modeli samochodów
- Wynik → Okno listy modeli samochodów – pusta lista**
4. Kierownik wybiera przycisk Zamknij

B. Dodaj model samochodu – główny

1. Kierownik wybiera przycisk Dodaj
2. System wyświetla okno dodawania modelu samochodu
3. Kierownik wprowadza model samochodu
- Dane wejściowe → plik „modele_poprawne.txt”**
4. System waliduje model samochodu [model samochodu poprawny]
5. System zapisuje model samochodu
6. System wyświetla komunikat o sukcesie operacji
7. Kierownik wybiera przycisk OK

C. Wyświetl listę modeli samochodów – główny

1. Kierownik wybiera przycisk Lista modeli samochodów
2. System pobiera listę modeli samochodów
3. System wyświetla okno listy modeli samochodów
- Wynik → Okno listy modeli samochodów – lista zawiera dodany model samochodu**
4. Kierownik wybiera przycisk Zamknij

ST215: Dodaj model samochodu – sukces (NEGATYWNY)

- A. Wykonanie ST213-A
- B. Wykonanie ST213-B
- C. Wykonanie ST214-C