

TESTOWANIE OPROGRAMOWANIA

porównanie metod i technik

Szymon Cebula
Daniel Leśnik

Testowanie oprogramowania

Jest to proces związany z wytwarzaniem oprogramowania. Jest on jednym z procesów kontroli jakości oprogramowania.

Celem testowania jest:

- wykrycie i usunięcie błędów w systemie
- ocena niezawodności systemu

Fazy testowania

Weryfikacja(*verification*) - testowanie zgodności systemu z wymaganiami zdefiniowanymi w fazie określenia wymagań.

Atestowanie(*validation*) - ocena systemu lub komponentu podczas lub na końcu procesu jego rozwoju na zgodności z wyspecyfikowanymi wymaganiami. Atestowanie jest więc weryfikacją końcową.

Przegląd Oprogramowania

Przegląd - proces lub spotkanie, podczas którego produkt roboczy lub pewien zbiór produktów roboczych jest prezentowany dla personelu projektu, kierownictwa, użytkowników, klientów lub innych zainteresowanych stron celem uzyskania komentarzy, opinii i akceptacji.

Typy Przeglądów Oprogramowania

Przegląd techniczny – ocena zgodności elementów oprogramowania z przyjętym planem

Przejścia(*walkthrough*) – wczesna ocena dokumentów, modeli, projektów i kodu

Audyt – przeglądy potwierdzające zgodność oprogramowania z wymaganiami, specyfikacjami, zaleceniami, standardami, procedurami, instrukcjami, kontraktami i licencjami

Rodzaje testów

Wykrywanie błędów - głównym celem jest wykrycie jak największej liczby błędów w programie

Testy statystyczne - celem jest wykrycie przyczyn najczęstszych błędnych wykonania oraz ocena niezawodności systemu

Testy dynamiczne - polegają na wykonywaniu (fragmentów) programu i porównywaniu uzyskanych wyników z wynikami poprawnymi

Testy statyczne - oparte na analizie kodu

Fazy testowania systemu

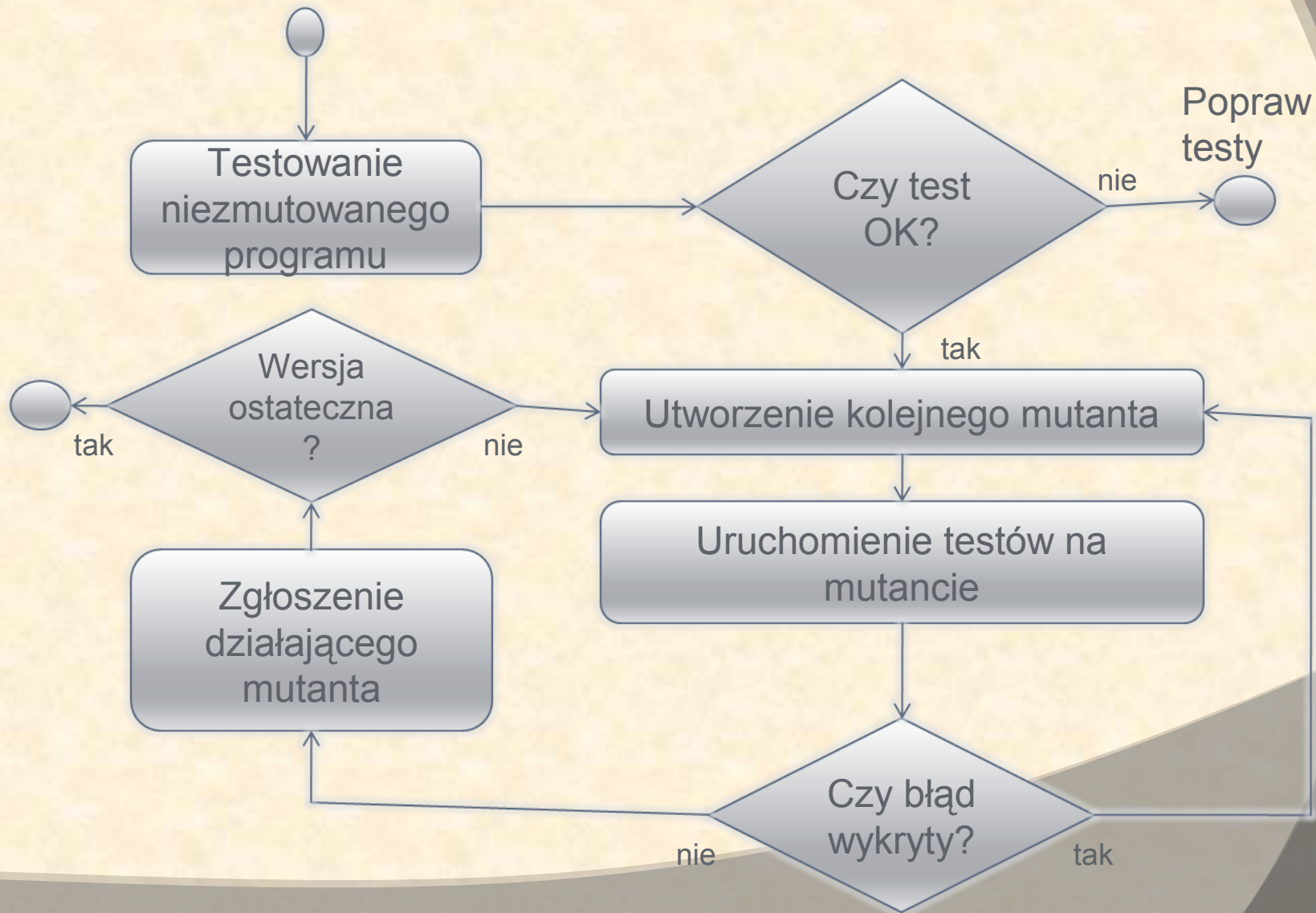
Testy modułów - wykonywane już w fazie implementacji bezpośrednio po zakończeniu realizacji poszczególnych modułów

Testy systemów - integrowane są poszczególne moduły i testowane są poszczególne podsystemy oraz system jako całość

Testy akceptacji(*acceptance testing*)

- alpha - system przekazywany jest do przetestowania przyszłemu użytkownikowi
- beta - polegają na nieodpłatnym przekazaniu pewnej liczby kopii systemu grupie użytkowników

Testowanie mutacyjne



Standardy testowania

Podstawowym standardem dla testowania oprogramowania jest *IEEE 829-1998 (829 Standard for Software Test Documentation)*.

- **Test Plan** –co będzie testowane, jak długo potrwa cały proces oraz jaki będzie zakres testów.
- **Test Design Specification** – szczegóły na temat warunków testowania
- **Test Case Specification** – specyfikuje dane testowe określone w Test Design Specification.
- **Test Procedure Specification** – zawiera szczegóły na temat przeprowadzenia każdego testu włączając w to założenia oraz poszczególne kroki testów.
- **Test Item Transmittal Report** – zawiera raporty na temat czasu przejścia testowanych fragmentów oprogramowania między etapami.

Standardy testowania

- **Test Log** – zawiera informacje o tym, które przypadki testowania zostały użyte, kto je użył i w jakim porządku oraz informacje o ich powodzeniu.
- **Test Incident Report** – zawiera informacje o testach zakończonych niepowodzeniem. Informacje o wynikach oraz dlaczego dany test nie powiódł się.
- **Test Summary Report** – raport ten zawiera wszystkie istotne informacje ujawnione podczas zakończonych testów .Ostateczna forma dokumentu jest wykorzystywana w celach weryfikacji poprawności testowanego systemu względem wymagań zdefiniowanych przez zleceniodawców.

Elementy podlegające testowaniu

- Wydajność systemu
- Interfejsy systemu
- Własności operacyjne systemu
- Testy zużycia zasobów
- Zabezpieczenie systemu
- Przenaszalność oprogramowania
- Niezawodność oprogramowania
- Odtwarzalność oprogramowania (*maintainability*)
- Bezpieczeństwo oprogramowania
- Kompletność i jakość złożonych funkcji systemu
- Nie przekraczanie ograniczeń
- Modyfikowalność oprogramowania
- Obciążalność oprogramowania
- Skalowalność systemu
- Akceptowalność systemu
- Jakość dokumentacji

Testowanie na zasadzie białej skrzynki

Testowanie n/z białej skrzynki pozwala sprawdzić wewnętrzną logikę programów poprzez odpowiedni dobór danych wejściowych, dzięki czemu można prześledzić wszystkie ścieżki przebiegu sterowania programu

Tradycyjnie programiści wstawiają kod diagnostyczny do programu aby śledzić wewnętrzne przetwarzanie. Debuggery pozwalają programistom obserwować wykonanie programu krok po kroku.

Testowanie na zasadzie białej skrzynki

Testowanie to cechuje:

- sprawdzanie wewnętrznej logiki oprogramowania
- dane testowe muszą uwzględniać każdą ścieżkę w programie
- kryterium doboru danych testowych:
 - kryterium pokrycia wszystkich instrukcji
 - kryterium pokrycia instrukcji warunkowych

Testowanie na zasadzie czarnej skrzynki

Tak określa się sprawdzanie funkcji oprogramowania bez zaglądania do środka programu.

Testujący traktuje sprawdzany moduł jak „czarną skrzynkę”, której wnętrze jest niewidoczne.

Testowanie na zasadzie czarnej skrzynki

To testowanie cechuje się:

- nie wnikaniem w kod programu
- testujemy specyfikacje, której „nie ma”
- sprawdzaniem dokumentu pod względem błędów i przeoczeń
- badaniem standardowe
- szczegółowym testowanie

Test Driven development

Polega na wielokrotnym powtarzaniu kilku kroków:

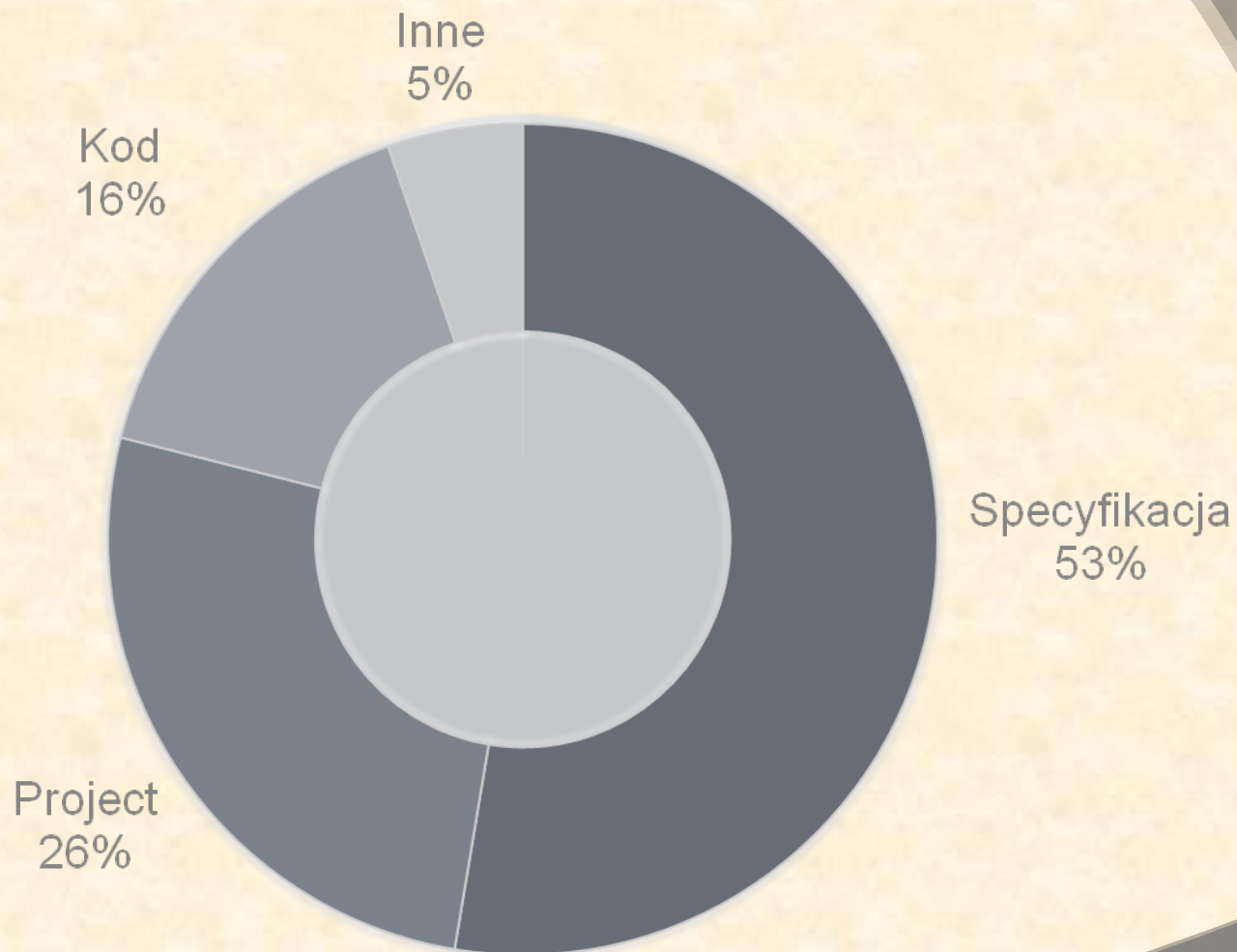
- programista pisze automatyczny test sprawdzający dodawaną funkcjonalność. Test w tym momencie nie powinien się udać
- implementacja funkcjonalności. W tym momencie wcześniej napisany test powinien się udać
- programista dokonuje refaktoryzacji napisanego kodu, żeby spełniał on oczekiwane standardy.

Test Driven development

To testowanie cechuje:

- jest oparte o testy jednostkowe
- testy pisane są przed implementacją metod właściwych
- metodę tą wykorzystuje się w MSF v.3 i MSF v.4

Przyczyny powstawania błędów



Podsumowanie

- Stosowanie testowania oprogramowania może znacznie obniżyć koszty projektu, dzięki wczesnemu wykryciu błędów
- Wskazuje błędy w działaniu programowania, błędne linie kodu
- Sprawdza stabilność systemu

Bibliografia

- www.wikipedia.pl
- www.testowanie.net
- www.testerzy.pl