
PORÓWNANIE METOD I TECHNIK TESTOWANIA OPROGRAMOWANIA

Maciej Oleksy
Zenon Matuszyk

CO TO JEST TESTOWANIE OPROGRAMOWANIA?

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

GŁÓWNE CELE TESTOWANIA OPROGRAMOWANIA:

- ▮ Weryfikacja oprogramowania - testowanie zgodności systemu z wymaganiami zdefiniowanymi w fazie określenia wymagań.
- Walidacja (atestowanie) oprogramowania - ocena systemu lub komponentu podczas lub na końcu procesu jego rozwoju na zgodności z wyspecyfikowanymi wymaganiami.

CELE TESTOWANIA:

- Testowania umożliwia wykrycie błędów we wczesnych stadiach rozwoju oprogramowania, co pozwala zmniejszyć koszty usuwania tego błędu.
- Warto testować na każdym etapie tworzenia oprogramowania.
- Testować należy jak najwcześniej, ponieważ
im później wykryty zostanie błąd tym większy jest koszt jego usunięcia.

STANDARDY:

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

Test Plan – dokument planowania zarządzania projektem, który składa się z informacji o tym, w jaki sposób będą prowadzone testy, kto będzie je przeprowadzał, co będzie testowane, jak długo potrwa cały proces oraz jaki będzie zakres testów.

Test Design Specification – szczegóły na temat warunków testowania, oczekiwanych wyników a także kryteriach przejścia testu.

Test Case Specification – specyfikuje dane testowe do użycia podczas wdrażania warunków testowania określonych 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.

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 oraz wyceny jakości procesów testowania, jakości oprogramowania poddanego testowi, a także statystyki uzyskane z Incident Report. Raport referuje również do typów i czasu trwania wykonanych testów w celu usprawnienia wszelkich planów związanych z testami w przyszłości. Ostateczna forma dokumentu jest wykorzystywana w celach weryfikacji poprawności testowanego systemu względem wymagań zdefiniowanych przez zleceniodawców.

RODZAJE TESTÓW OPROGRAMOWANIA

- **Testy funkcjonalne**. Polegają one na tym, że wcielamy się w rolę użytkownika, traktując oprogramowanie jak „czarną skrzynkę”, która wykonuje określone zadania. Nie wnikamy wogóle w techniczne szczegóły działania programu. Testy te często są nazywane testami **czarnej skrzynki**
- **Testy strukturalne**. Tym razem tester ma dostęp do kodu źródłowego oprogramowania, może obserwować jak zachowują się różne części aplikacji, jakie moduły i biblioteki są wykorzystywane w trakcie testu. Te testy czasami są nazywane testami **białej skrzynki**

RODZAJE TESTÓW OPROGRAMOWANIA CD:

1. TESTY MANUALNE

Testy wykonywane ręcznie przez testera, który przechodzi przez interfejs użytkownika zgodnie z określoną sekwencją kroków:

- a) **testy integracyjne** pozwalają sprawdzić jak współpracują ze sobą różne komponenty oprogramowania. Obecnie rzadko mamy do czynienia z monolitycznymi aplikacjami. Są one raczej tworzone modułowo, dlatego należy sprawdzić, czy wszystko razem działa poprawnie
- b) **testy systemowe** dotyczą działania aplikacji jako całości. Zazwyczaj na tym poziomie testujemy różnego rodzaju wymagania niefunkcjonalne, takie jak – szybkość działania, bezpieczeństwo, niezawodność, dobrą współpracę z innymi aplikacjami i sprzętem.

Testy dopasowane do aktualnego zapotrzebowania/przeznaczenia:

- a) **testy funkcjonalne** – znane również jako testy czarnej skrzynki. Osoba testująca nie ma dostępu do informacji na temat budowy programu, który testuje. Wykonując testy nie opiera danych testowych na budowie wewnętrznej programu, lecz na założeniach funkcjonalnych, jakie powinien spełniać program zgodnie z dokumentacją

- b) **testy regresyjne** – mają na celu sprawdzenie wpływu nowych funkcjonalności na działanie systemu
- c) **testy akceptacyjne** z udziałem klienta – wykonywane w celu sprawdzenia na ile oprogramowanie działa zgodnie z wymaganiami klienta
- d) **testy dokumentacji**, których celem jest wykrycie niespójności i niezgodności w dokumentacji analitycznej, technicznej oraz dokumentacji użytkownika, sporządzonej w ramach realizowanego projektu informatycznego
- e) **testy użyteczności**, których celem jest weryfikacja interfejsu użytkownika w zakresie przystępności, wygody, szybkości oraz zgodności z oczekiwaniami przyszłych użytkowników.

2.TESTY AUTOMATYCZNE

Testy automatyczne skutecznie przyspieszają proces tworzenia testów systemowych, ich wykonywanie oraz analizę, a tym samym pozwalają na wcześniejsze wykrycie i wyeliminowanie błędów w aplikacjach. Testy automatyczne wykonywane są w oparciu o wysokiej jakości oprogramowanie.

Aby przeprowadzić dobry test należy skupić się na:

- ❖ Poznaniu architektury systemu. Na tym etapie będzie można wstępnie określić na ile skomplikowany będzie proces testowania
- ❖ Określeniu jakiego typu testy są potrzebne. Może być to związane zarówno z charakterystyką testowanego systemu jak i z wymaganiami klienta, standardami zapewniania jakości przyjętymi w firmie
- ❖ Weryfikacji testowalności systemu. Jest to istotne przy dużych, wielowarstwowych aplikacjach. Tester, aby móc skutecznie testować system, musi mieć dostęp do wszelkich potrzebnych mu danych.
- ❖ Takie podejście do procesu testowania pozwala na dokładne oszacowanie czasochłonności procesu testowania a co za tym idzie określenie dokładnych kosztów całego procesu.

AUDYT PROJEKTU INFORMATYCZNEGO:

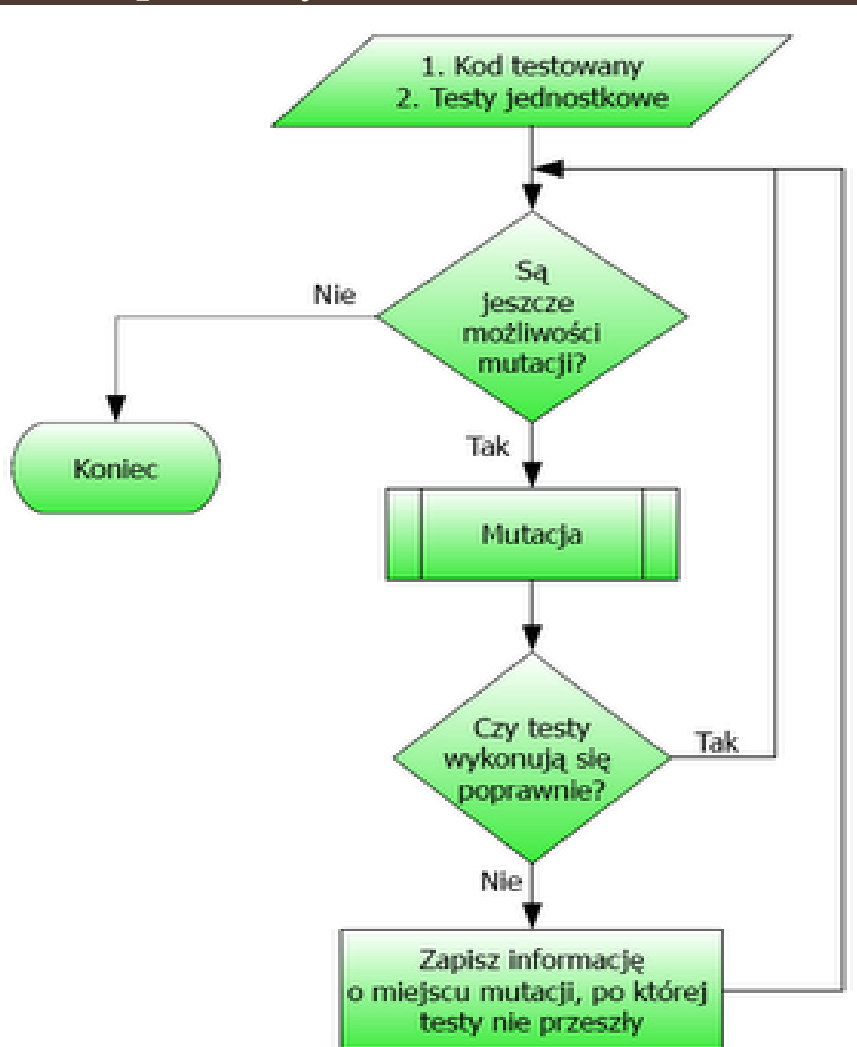
Celem audytu projektu informatycznego jest dostarczenie odbiorcy i dostawcy obiektywnych, aktualnych i syntetycznych informacji o stanie całego projektu

Jest to osiągane przez zbieranie dowodów, że projekt:
posiada możliwości (zasoby, kompetencje, metody, standardy)
by osiągnąć sukces,
optymalnie wykorzystuje te możliwości,
rzeczywiście osiąga założone cele (częstkowe).

Zebrane informacje służą jako podstawa do podejmowania strategicznych decyzji w projekcie

TESTOWANIE MUTACYJNE:

Przebieg procesu testowania mutacyjnego przedstawiony jest na poniższym schemacie:



Polega na automatycznym wielokrotnym wprowadzaniu małych losowych błędów w programie i uruchamianiu za każdym razem testów jednostkowych, które powinny te błędy wykryć. Testowanie mutacyjne wymaga dużej mocy obliczeniowej i dlatego dopiero od niedawna próbuje się je wykorzystywać w praktyce.

CO PODLEGA TESTOWANIU?

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

ZAKOŃCZENIE:

- ▣ Podsumowując testy możemy klasyfikować na różne sposoby, ze względu na to, co i w jaki sposób podlega procedurom testowym. Również oprogramowanie testujemy na bardzo różne sposoby, zależnie od tego jaki aspekt jego działania jest w danym momencie dla nas ważny.
- Stosowanie testowania oprogramowania może znacznie obniżyć koszty projektu, dzięki wczesnemu wykryciu błędów i ich naprawie.

BIBLIOGRAFIA:

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

KONIEC

DZIĘKUJEMY

