



Poziomy testów

Rodzaje i charakterystyka



Poziom testów

Poziom testów jest definiowany jako grupa czynności testowych wykonywanych na danym etapie projektu, które są razem zorganizowane i zarządzane. Poziomy testów dzielimy na testy modułowe, integracyjne, systemowe i akceptacyjne.

poziomy testów	modułowe
	integracyjne
	systemowe
	akceptacyjne



Obszary testowania

Na każdym poziomie testów określa się kolejno następujące zagadnienia:

- przedmiot testów (to, co jest testowane)
- cel (to, co chcemy osiągnąć)
- podstawę testów (na jakiej podstawie tworzymy przypadki testowe)
- typowe defekty (nieprawidłowości jakich szukamy na danym poziomie)
- preferowane środowisko testowe (warunki, jakie muszą być spełnione, aby wykonywane testy były miarodajne)
- odpowiedzialność i specyficzne podejście



Testy modułowe

Testy modułowe nie dotyczą całej aplikacji, lecz skupiają się na jej poszczególnych elementach, które można przetestować oddzielnie, w oderwaniu od reszty aplikacji — na modułach.

Przedmiot: moduły, jednostki, komponenty aplikacji, moduły baz danych

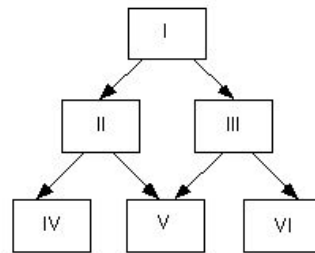
Cel: wykrycie defektów w module, zablokowanie przedostania się defektów do powiązanych modułów

Podstawa: kod, projekt szczegółowy, specyfikacja techniczna modułu

Typowe defekty: niepoprawna logika, martwy kod, niepoprawna funkcjonalność

Preferowane środowisko testowe: programistyczne

Podejście top-down



Istnieje kilka wypracowanych podejść do testów modułowych. Jednym z przykładów jest podejście **top-down** (od góry do dołu). Moduły znajdujące się na najwyższym poziomie są testowane jako pierwsze. Moduły będące w hierarchii poniżej są zastępowane zaślepkami. Testowane moduły są używane do testowania niżej położonych komponentów. Proces jest kontynuowany do momentu przetestowania komponentów znajdujących się na niższym poziomie.

Zaślepka to kawałek kodu zwracający odpowiedź która docelowo będzie pochodzić z innego modułu. Służy do testowania, czy ta odpowiedź jest poprawnie interpretowana. Może być użyta kiedy drugi moduł nie jest jeszcze napisany.



Testy modułowe (przykład)

Wyobraź sobie pojedyncze wagony pociągu. Zanim zostaną podłączone do lokomotywy, muszą zostać sprawdzone. Każdy taki wagon to moduł. Aby pociąg dotarł na czas do punktu docelowego, wszystkie wagony muszą być sprawne.



Testy integracyjne

Testy integracyjne są ukierunkowane na znajdowanie błędów w interfejsach na styku poszczególnych zintegrowanych elementów - głównie usterek dotyczących wymiany danych pomiędzy nimi.

Przedmiot: podsystemy, infrastruktura, interfejsy, interfejsy programowania aplikacji

Cel: budowanie zaufania do interfejsów, wykrycie błędów występujących w poszczególnych modułach, szczególnie w komunikacji

Podstawa: projekt oprogramowania, przepływ pracy (tzn. sposób przepływu informacji pomiędzy różnymi obiektami biorącymi udział w jej przetwarzaniu), przypadki użycia (czyli interakcje z całym systemem lub podsystemami, prowadzące do konkretnego rezultatu), specyfikacja wymagań



Testy integracyjne

Typowe defekty: błędy w komunikacji pomiędzy modułami, niezgodność interfejsów, niepoprawne lub brakujące dane

Preferowane środowisko testowe: niewielka ilość zintegrowanych modułów (im mniej, tym lepiej)

Interfejs odnosi się do interfejsu użytkownika, czyli sposobu, w jaki użytkownik komunikuje się z aplikacją. Interfejs użytkownika jest odpowiedzialny za prezentację informacji oraz udostępnianie narzędzi potrzebnych do wykonania określonych czynności.

API, czyli interfejs programowania aplikacji (ang. application programming interface) to zestaw reguł umożliwiających przesyłanie danych między aplikacjami.



Testy integracyjne

Na każdym etapie integracji, testerzy koncentrują się wyłącznie na samej integracji. Na przykład, gdy integrują moduł A z modułem B, interesują się tylko testowaniem komunikacji pomiędzy modułami, a nie funkcjonalnością poszczególnych modułów, gdyż ta była sprawdzona wcześniej w testach modułowych. Kiedy dane generowane przez moduł A nie docierają do modułu B lub docierają zniekształcone, może to powodować komplikacje.

Aby zminimalizować ryzyko usterek należy stosować **podstawowe zasady integracji**.



Podstawowe zasady integracji

- poszczególne moduły/systemy powinny być integrowane tylko raz
- kolejność i terminy integracji kolejnych elementów powinny być udokumentowane
- należy integrować możliwie małą liczbę modułów w jednym czasie
- nie powinno się integrować więcej niż dwóch modułów naraz
- powinno się dążyć do minimalizacji nakładów związanych z budową zaślepek



Testy integracyjne (przykłady)

Zakładając, że przeprowadziliśmy już testy pojedynczych modułów (wagonów) i wszystkie testy przebiegły prawidłowo. Czy to oznacza, że pociąg może wyruszyć w drogę? Oczywiście nie. Należy bowiem sprawdzić, czy wagony są trwale i bezpiecznie ze sobą połączone, w szczególności — czy pierwszy z nich jest poprawnie podłączony do lokomotywy. W taki sposób badamy integrację.

Przeprowadzanie transakcji finansowych

Dokonując przelewu bankowego z naszego konta odejmowana jest pewna kwota pieniędzy, która od tak nie znika. Musi pojawić się na innym koncie — tym, na które dokonujemy przelewu. Aby mieć pewność, że transakcja została przeprowadzona poprawnie, należy upewnić się, że pieniądze dotarły na wskazany rachunek docelowy.



Testy systemowe

Testy systemowe skupiają się na całościowym działaniu systemu z uwzględnieniem każdej funkcjonalności.

Przedmiot: aplikacje, systemy operacyjne, system podlegający testowaniu

Cel: działanie całego systemu, walidacja zgodności zachowań systemu z projektem i specyfikacją wymagań, budowanie zaufania do jakości produktu

Podstawa: specyfikacja wymagań, przypadki użycia, historyjki użytkownika

Typowe defekty: niepoprawne lub nieoczekiwane zachowania systemu, problemy z prawidłowym działaniem systemu w środowisku testów, niezgodność działania systemu z opisami zawartymi w instrukcji użytkownika

Preferowane środowisko testowe: w pełni zintegrowane środowisko testowe odzwierciedlające docelowe środowisko



Testy systemowe (przykład)

Zbadaliśmy już wagony, upewniliśmy się także, że połączenia pomiędzy nimi są poprawne, trwałe i bezpieczne. Jednak przed wpuszczeniem do pociągu pasażerów, powinniśmy przeprowadzić jazdę próbną, aby sprawdzić jak pociąg zachowuje się na torach. Jak reaguje na zakręty, szczególnie te bardziej ostre? Czy nie ma ryzyka wykolejenia? Jaka jest jego droga hamowania przy różnych prędkościach?



Testy akceptacyjne

Testy akceptacyjne odgrywają decydującą rolę w kwestii wdrożenia rozwiązania do docelowego środowiska. Odpowiedzialność za testy akceptacyjne leży często po stronie klientów lub użytkowników systemu.

Przedmiot: aplikacje, systemy operacyjne, system podlegający testowaniu

Cel: walidacja i weryfikacja zgodności zachowań systemu z projektem i specyfikacją wymagań, budowanie zaufania do jakości produktu

Podstawa: specyfikacja wymagań, procesy biznesowe, przypadki użycia, historyjki użytkownika, wymogi prawne

Typowe defekty: niepoprawne lub nieoczekiwane zachowania systemu, problemy z prawidłowym działaniem systemu w środowisku testów, niezgodność działania systemu z opisami zawartymi w instrukcji użytkownika, niedostateczna wydajność, słabe zabezpieczenia

Preferowane środowisko testowe: w pełni zintegrowane środowisko testowe odzwierciedlające docelowe środowisko



Testy akceptacyjne zgodności z umową i testy zgodności legislacyjnej.

Testy akceptacyjne zostały podzielone na dwie kategorie:

- testy zgodności z umową i testy zgodności legislacyjnej
- testy alfa i beta

Testy zgodności z umową są wykonywane przez sprawdzenie spełnienia kryteriów akceptacji zapisanych w kontrakcie na wykonanie oprogramowania na zamówienie. Te kryteria akceptacji powinny być zdefiniowane w momencie negocjacji umowy. Testy zgodności legislacyjnej wykonuje się sprawdzając, czy oprogramowanie jest zgodne z wszystkimi przepisami prawnymi, z którymi musi być zgodne, takimi jak rozporządzenia rządowe, inne akty prawne lub przepisy dotyczące bezpieczeństwa.



Testy alfa i beta

Czynniki wpływające na podział to:

- miejsce wykonywania testów
- środowisko testów
- profil działalności testerów

Testy alfa to symulowane lub rzeczywiste testy przeprowadzane u producenta bez udziału twórców oprogramowania. Testy alfa przeprowadza się w środowisku jak najbardziej zbliżonym do środowiska docelowego, przed wypuszczeniem produktu na rynek, w chwili gdy jest on już dostatecznie stabilny, zazwyczaj w siedzibie producenta oprogramowania. Testy te są wykonywane przez potencjalnych użytkowników, a nie przez twórców. Celem testów alfa jest uzyskanie opinii potencjalnych lub obecnych klientów.



Testy alfa i beta

Testy beta to testy produkcyjne przeprowadzane w środowisku niezwiązanym z zespołem projektowym. Są wykonywane głównie przez klientów lub potencjalnych klientów w ich własnych lokalizacjach. Z reguły po pomyślnym przeprowadzeniu testów beta dochodzi do akceptacji produktu i finalizacji zlecenia.