



Techniki testów

Podstawy

Zanim przejdziemy do technik zastanówmy się, czym jest testowanie. Podstawową i najkrótszą definicją jest - znajdowanie anomalii. Nie rozstrzygamy, czym jest anomalia i kto ma stwierdzić czy rzeczywiście jest to anomalia czy też nie. Pozostawiamy dopasowanie tej definicji do Państwa środowiska i pojmowania testowania.

Możemy ją rozszerzyć do większego obszaru jak na przykład, testowaniem jest znajdowanie defektów lub ich zapobieganie oraz zapewnienie określonej jakości produktu (oprogramowanie lub też urządzenie). Testować możemy wszystko, co zostało wytworzone przez człowieka, od igły po odrzutowce i od napojów po budynki. Co więcej każdy z nas, czy tego chce czy nie, jest częściowo testerem. W codziennym życiu podejmujemy niezliczoną ilość decyzji, które wynikają z naszego doświadczenia, wiedzy lub innych czynników jak chociażby lenistwo.

Prosty przykład: wybierając drogę, którą pójdziemy do szkoły/pracy/sklepu kierujemy się potrzebami oraz jakością trasy. Przetestowaliśmy wszystkie opcje i wybieramy najlepsze dla nas, dlatego aby końcowy rezultat był dla nas jak najbardziej korzystny.

Testowanie jest w każdym aspekcie naszego życia i jest nierozdzielnie związane z kwestią poznania i percepcji. Ma to bezpośrednie przełożenie na testowanie najróżniejszych produktów.

Kiedy kończy się testowanie?

Choć wielu w to nie wierzy testowanie produktu nie kończy się nigdy. Trwa od momentu powstania koncepcji po unicestwienie produktu. Nie możemy powiedzieć, że jakość jest doskonała i nic nie da się już poprawić. Nie prawdą jest, że produkt nie ma żadnego błędu. Stwierdzenie to możemy zastąpić innym: żaden błąd nie został jeszcze znaleziony. Może ocieramy się w tym momencie o Prawo Murphego, ale tak właśnie jest.

Planowanie testów

Jeśli podejmujemy decyzję o testowaniu musimy określić cele testowania: Do czego dążymy/Co chcemy osiągnąć?

Odpowiedź na te pytania powinny być zawarte w planach testów.

Inne tematy podjęte przez test plany to:

- zasięg testowania i ryzyko,
- założenia testowania: techniki testowania, narzędzia testowe, pokrycie testów, grupa testowa etc
- zasoby testowe: ludzie, środowisko, środki etc
- implementacja polityki i strategii testowej,
- plan analizy testów i zajęć projektowych,
- określenie kryterium zakończenia testów.

Kontrolowanie testów

Plan testów musi być na bieżąco weryfikowany w każdym momencie procesu testowania.

Zdefiniowano, więc proces kontroli. W każdym momencie projektu musimy sprawdzać jak daleko "odchodzimy" od wstępnych założeń. Podstawowe czynniki to:

- pomiar i analiza rezultatów,
- monitorowanie i dokumentowanie postępu, pokrycie testów i kryteria zakończenia testów,
- inicjowania właściwych reakcji w odpowiedzi na zmiany,

- podejmowanie decyzji.

Projektowanie i analiza testów

Planowanie i analiza testów jest czynnością, podczas której ogólne postanowienia testowe zostają przemienione w namacalne warunki testowe i projektowanie testów.

Analiza i projektowanie testów obejmuje następujące bardzo ważne zadania:

- przegląd podstaw testów (takich jak wymagania, architektura, projekt, interfejsy)
- identyfikacja warunków testowych lub wymagań testowych i wymaganych danych testowych bazujących na analizie czynników testowych, specyfikacji, zachowań i struktury
- projektowanie testów
- sprawdzenie testowalności wymagań systemu
- projektowanie ustawień środowiska testowego i identyfikacja wymaganej infrastruktury i narzędzi

Wdrażanie i wykonywanie testów

Implementacja testów i ich wykonywanie jest czynnością gdzie warunki testów zostają przemienione w przypadki testowe i zbiór czynników testowych oraz konfigurację środowiska.

Implementacja testów i ich wykonywanie składa się z następujących zadań:

- rozwijanie i ustalanie ważności przypadków testowych, tworzenie danych testowych, pisanie procedur testowych i opcjonalnie przygotowanie środowisk symulatorów testowych oraz pisanie skryptów testowych dla automatyzacji.
- tworzenie zestawów testowych z przypadków testowych dla efektywnego wykonywania testów
- wykonywanie przypadków testowych ręcznie lub przy użyciu narzędzi automatycznych, zgodnie z zaplanowaną sekwencją
- zapamiętywanie wyników wykonywanych testów i zapisywanie wersji oprogramowania będącego testowanym i narzędzi testowych
- porównywanie aktualnych wyników z oczekiwanymi wynikami
- raportowanie różnic jako przypadków i ich analiza w celu ustalenia ich przyczyn (np. błąd w kodzie, złe dane testowe, błąd w dokumentacji lub złe wykonanie instrukcji testowej)
- powtarzanie czynności testowych jako rezultat wykrycia przypadków. Dla przykładu, ponowne wykonanie instrukcji testowej, która poprzednio zwróciła wynik negatywny, dla upewnienia się, że błąd już nie występuje; ponowne wykonanie poprawionej instrukcji testowej by upewnić się, że błąd został przypadkowo wprowadzony w poprzednio niezmiennych obszarach lub czy błąd nie odkrył kolejnych defektów (testy regresji).

Sprawdzanie kryterium zakończenia testów

Sprawdzanie kryterium zakończenia testów jest czynnością testową względem zdefiniowanych celów. Musi zostać to zrobione na każdym poziomie testów.

Sprawdzanie kryterium składa się z następujących zadań:

- sprawdzenie zapisów potestowych względem kryterium wyjścia określonego w planach testów
- testowanie czy więcej testów jest potrzebne lub czy kryterium zakończenia powinno być zmienione
- napisanie podsumowania testów w postaci raportu dla udziałowców

Zakończenie testów

Czynności kończące testy polegają na zbieraniu wszystkich danych ze wszystkich ukończonych czynności testowych dla konsolidacji doświadczenia, czynników testowych, faktów i liczb. Dla przykładu, kiedy oprogramowanie zostaje wypuszczone na rynek projekt zostaje zakończony (lub wstrzymany), kamienie milowe projektu zostały osiągnięte lub wersja oprogramowania z uwzględnieniem komentarzy i błędów zgłoszonych przez użytkowników zostało opublikowana.

Na czynności zamykająca aktywności testowe składają się:

- sprawdzenie, które z zaplanowanych dostaw rzeczywiście zostały dostarczone, sprawdzenie raportów zamknięcia przypadków lub sprawdzenie raportów zgłoszenia zmian dla tych, które pozostały otwarte, sprawdzenie dokumentacji akceptacyjnych systemu

- zakończenie i zarchiwizowanie wszystkich czynników testowych dla późniejszego, ponownego ich użycia
- przejęcie czynników testowych przez organizację zajmującą się serwisem produktu.
- analiza tego, czego nauczyliśmy się podczas projektu, aby wiedza ta posłużyła nam dla przyszłych projektów i dla poprawy dojrzałości organizacji testowej.

Psychologia testów

Charakterystyka pracy podczas testowania i przeglądu dokumentów jest różna do tego, jaka powstaje podczas analizy i rozwoju oprogramowania. Z właściwą charakterystyką programiści są zdolni do testowania własnego kodu, ale rozdzielenie tych obowiązków na testera i programistę jest działaniem mającym za zadanie skoncentrowanie wysiłków i zdobycie dodatkowych korzyści, takich jak niezależny wgląd w kod dokonany przez wykwalifikowaną i profesjonalnie przygotowaną kadrę testerów. Niezależne testowanie może odbywać się na każdym poziomie weryfikacji.

Wysoki poziom niezależności (unikając stronniczości autora) jest bardzo efektywny w znajdowaniu defektów i uszkodzeń z nich wynikających. Niezależność nie wyklucza jednak, że również programiści mogą samodzielnie wychwycić dużo błędów we własnym kodzie. Zdefiniowano kilka poziomów niezależności:

- projektowanie testów przez osobę, która pisze testowane oprogramowanie (niski poziom niezależności)
- projektowanie testów przez inną osobę (np. z ekipy programistów)
- projektowanie testów przez osobę z innej grupy organizacji (np. niezależny zespół testowy)
- projektowanie testów przez osobę z innej organizacji lub firmy (np. certyfikowanie przez zewnętrzną organizację)

Ludzie i projekty kierowane są poprzez różne cele. Ludzie są skłaniani do dopasowywania swoich planów do celów przez zarząd oraz innych udziałowców np. dla znalezienia defektów lub potwierdzenia, że oprogramowanie działa. Ważne jest, więc aby cele były czytelne dla całej organizacji.

Identyfikacja problemów podczas testowania może być postrzegana jako kryzys w projekcie i wymierzony w autora. Testowanie jest więc rozważane jako destruktywna czynność nawet, gdy jest bardzo konstruktywna w zarządzaniu ryzykiem. Szukanie problemów w systemie wymaga ciekawości, profesjonalnego pesymizmu, krytycznego spojrzenia, przypisywania uwagi do szczegółów, dobrej komunikacji z kolegami programistami i doświadczeniu, na którym można oprzeć zgadywanie błędów.

Gdy błąd, defekt i jego konsekwencje są zakomunikowane w konstruktywny sposób, negatywne emocje pomiędzy testerami a analitykami, projektantami i programistami mogą być wykluczone. Zasada ta stosuje się nie tylko do testowania, ale i do przeglądu dokumentacji.

Testerzy i liderzy grup testowych muszą potrafić komunikować defekty, postęp, ryzyko z użyciem świetnych umiejętności interpersonalnych i w konstruktywny sposób. Dla autora oprogramowania lub dokumentu informacja o defektach może być pomocna w uczeniu się na własnych błędach. Znalezione defekty, które zostały naprawione podczas testowania oszczędzają organizacji czas i pieniądze i obniżają ryzyko.

Problemy komunikacyjne mogą pojawić się, gdy tester jest postrzegany jako dostarczyciel złych wiadomości o defektach. Jednakże istnieje wiele dróg dla poprawienia zdolności komunikacji i współpracy między testerami i innymi pracownikami firmy:

- rozpocząć współpracę raczej niż walkę – przypomnieć wszystkim o wspólnych celach polepszenia jakości systemu
- zakomunikować informacje o produkcie w sposób neutralny i skoncentrowany na faktach, bez krytyki osób, które go stworzyły; dla przykładu zapisać obiektywnie i zgodnie z faktami raporty na temat przypadków i przejrzeć spostrzeżenia
- spróbować zrozumieć jak odczuwają ludzie i dlaczego reagują tak jak reagują
- upewnić się, że każda osoba rozumie co zostało powiedziane i vice versa