

Pytanie 1.:

Testy funkcjonalne – funkcje są tym „co” system robi, testy dotyczą funkcji lub ich cech, oraz ich współdziałania z innymi systemami. Testy funkcjonalne zajmują się zewnętrznym zachowaniem oprogramowania (testy czarnoskrzynkowe). Np.: testowanie zabezpieczeń - sprawdza funkcje (np. zapory) pozwalające na wykrycie zagrożeń, takich jak wirusy, pochodzących od złośliwych obcych; testowanie współdziałania, ocenia zdolność oprogramowania do współpracy z jednym lub większą liczbą wskazanych modułów lub systemów.

Testy niefunkcjonalne - typy testów: testowanie wydajnościowe, testowanie obciążeniowe, testowanie przeciążeniowe, testowanie użyteczności, testowanie pielęgnowalności, testowanie niezawodności oraz testowanie przenaszalności. Testowanie niefunkcjonalne polega na sprawdzeniu "jak" system działa; na wszystkich poziomach testów.

Pytanie 2.:

Testy regresywne - wykonanie poprawionych testów lub wykonanie testów w celu sprawdzenia czy w nie zmienianych częściach oprogramowania nie pojawiły się usterki lub czy naprawa usterek nie ujawniła innych defektów (testowanie regresywne); testowanie regresywne jest bardzo ważne w każdej iteracji oprócz pierwszej. Każdy przyrost może podlegać zarówno weryfikacji jak i walidacji. Testy regresywne można wykonywać na wszystkich poziomach testów i dla wszystkich typów testów: funkcjonalnych, niefunkcjonalnych i strukturalnych.

Smoke test sprawdza czy program/system da się uruchomić, czy jego interfejsy są dostępne i czy reagują na działania użytkownika. Jeżeli smoke test nie powiedzie się nie ma powodu aby przechodzić do sanity testów. Ten typ testów przeprowadzany jest przez programistów tuż przed oddaniem wersji aplikacji lub przez testerów, przed zaakceptowaniem otrzymanej do testów aplikacji.

Pytanie 3.: (wszystkie scenariusze zachowania)

Cztery przypadki testowe:

1. $a > 0$;
2. $a \leq 0$;
3. $b == 3$;
4. $b \sim 3$;

Pytanie 4.:

Testowanie zwinne - metoda testowania stosowana w projektach korzystających z metodologii zwinnych stosująca techniki i metody takie jak programowanie ekstremalne (XP), traktujące wytwarzanie jako klienta testowania i kładąca nacisk na metodę „najpierw test”. Testerzy łącznik między programistami, a użytkownikami.

Pytanie 5.:

Minimum i maksimum klasy równoważności to jej wartości brzegowe. Wartość brzegowa poprawnego przedziału jest nazywana poprawną wartością brzegową, a wartość brzegowa niepoprawnego przedziału – niepoprawną wartością brzegową. Testy można zaprojektować tak, żeby pokrywały zarówno poprawne jak i niepoprawne wartości brzegowe.

Poprawne wartości brzegowe 18; 60 niepoprawne to 17 i mniej oraz 61 i więcej. Reasumując { 17; 18; 19; 59; 60; 61 }

Pytanie 6.:

Co należy wpisać do okienek input, pierwszego i drugiego?

Co ma być przechowywane w DB?

Jakie wartości podane powodują błędne, a jakie poprawne działanie?

Dalsze pytania są zależne od tego co będzie w oknach input, długość znaków, rodzaj, jakie znaki rozpoznawać.

Pytanie 7.:

Istnieją różne cele testowania:

- znajdowanie usterek,
- nabieranie zaufania do poziomu jakości,
- dostarczanie informacji potrzebnych do podejmowania decyzji,
- zapobieganie defektom.