

## Grupa QA – zadanie 1

### ETAP 1

1.
  - a) Testy funkcjonalne – służą do przetestowania funkcji, które system powinien dostarczać. Znaczy to, że testy dotyczą tego „co” system robi. Funkcje, które należy przetestować mogą być znalezione w specyfikacji wymagań, w przypadkach użycia lub także w specyfikacji funkcjonalnej.
  - b) Testy нефункционалне – służą do tego aby przetestować działanie systemu pod kątem różnych cech np.: wydajności, niezawodności, dostępności. W skrócie można powiedzieć, że testy dotyczą tego „jak” działa system.
2.
  - a) Smoke testy – testy mające sprawdzić czy główne funkcjonalności systemu lub modułu działają, bez zbytniego zagłębiania się w szczegóły.
  - b) Testy regresji – są to testy powtórzeniowe. Wprowadza się je aby wykryć nowe usterki lub błędy powstałe wskutek napraw poprzednich błędów. Wykonuje się je po zmianach w oprogramowaniu jak i po zmianach w środowisku, w którym oprogramowanie działa.
3. Potrzeba cztery przypadki testowe.  
Zestawy testowe:
  - a=1, b=3;
  - a=1, b=2;
  - a=0, b=3;
  - a=0, b=2
4. Testowanie zwinne – metoda testowania, która jest stosowana projektach korzystających z metodologii zwinnych, która wytwarzanie oprogramowania traktuje jako klienta testowania. Do charakterystycznych cech tego podejścia do testowania można zaliczyć:
  - a) użytkownik jest członkiem zespołu wytwarzającego oprogramowanie (rozwiewa ewentualne wątpliwości),
  - b) testerze traktowani są jako pomost między programistami a użytkownikiem, a także
  - c) testerzy nie stanowią odrębnej grupy wykonującej swoje zadania,
  - d) testerzy nie muszą wysyłać raportów.
5. Analiza wartości brzegowych – metoda testowania, która polega na sprawdzaniu wartości znajdujących się na brzegach przedziałów, ponieważ przy obsłudze takich wartości pojawia się największe prawdopodobieństwo błędu. Dobrym przykładem sprawdzania wartości brzegowych jest np. sprawdzanie zakresu tablicy.

Wpisuje liczby: 18, 60, 17, 61

6. Pytania do analityka:
  - a) Jakiego typu dane powinny być przyjmowane przez pola input?
  - b) Jakie wartości mają być przyjmowane przez poszczególne pola input?
  - c) Jak system powinien zareagować na poprawne dane?
  - d) Jak system powinien zareagować na niepoprawne dane?
  - e) Czy pola input mogą przyjmować wartości, które już znajdują się w bazie?
7. Do celów testowania można zaliczyć m.in. zmniejszenie ilości błędów w oprogramowaniu, ocena jakości oprogramowania, dostarczenie pewnych informacji o oprogramowaniu w celu podjęcia decyzji.