

1. Testy funkcjonalne jak sama nazwa nawet wskazuje to testy funkcji programu czy systemu, czyli jak zachowywać ma się dany program, co on robi. Zwykle jak dany program czy system ma funkcjonować opisane jest w dokumentacji, jednak nie zawsze taka dokumentacja musi istnieć.

Testy niefunkcjonalne to testy, które sprawdzają jak się zachowuje program, lub co się z nim dzieje, np. pod jakimiś obciążeniami. Takie testy właśnie nazywamy min. Testy wydajnościowe czy obciążeniowe.

2. Smoke testy – Celem jego jest sprawdzenie czy program lub system w ogóle da się uruchomić. Czy włącza się np. interfejs i czy reaguje na działanie użytkownika. To typ testu, który wykonywany jest tuż przed oddaniem wersji aplikacji.  
Testy regresji – to testy wykonywane na już przetestowanym oprogramowaniu, które ma na celu sprawdzenie czy nie pojawiły się nowe usterki po dokonywanych zmianach.
3. Celem testowania jest znajdowanie błędów lub niepoprawnego działania danej funkcji oprogramowania, który powinien być klasyfikowany pod względem priorytetu.
4. Poprzez wykonanie testu potwierdzającego.
5. Przetestowałbym wartości brzegowe aplikacji czy inaczej zwane też BVA, w tym wypadku -50, -49 oraz 199 i 200
6. 4 przypadki testowe by pokryć wszystkie możliwości.
  - 1 przypadek: żadna instrukcja warunkowa nie zostaje spełniona
  - 2 przypadek: 1wszy warunek zostaje spełniony lecz 2gi nie
  - 3 przypadek: 1wszy warunek nie zostaje spełniony ale 2gi tak
  - 4 przypadek: oba warunki zostają spełnione
7. 17, 18, 60 oraz 61
8. Pytania, które bym zadał:
  - Jakie są wstępne warunki do wykonywania testu – zbiór danych wejściowych
  - jakie są oczekiwane rezultaty
  - jakie są warunki końcowe
9. Get – to próba żądania informacji od jakiegoś zasobu. POST – to przekazywanie danych (dodanie) do danego zasobu.
10. NIE. http jest protokołem bezstanowym.
11. To komendy SQL – INNER JOIN – zwraca część wspólna obu tabel, inaczej: elementy zbioru A należące do Zbioru B i na odwrót. LEFT JOIN – zwraca cała lewa tabelkę plus część wspólną, inaczej opisując: zwraca wszystkie elementy zbioru A oraz elementy, które są wspólne dla zbiorów A i B.
12. W katalogu /etc

13. Uczyłbym komendy cp w taki sposób: `>_ cp <bieżący folder> . <folder docelowy>`, spowoduje to skopiowanie wszystkich plików z bieżącego folderu do folderu docelowego (jeżeli nie istnieje to zostanie stworzony).