1. Różnica polega na tym, że w testowaniu funkcjonalnym testuje się w oparciu o wymagania, poprawność działania funkcji, proces biznesowy, a w niefunkcjonalnych chodzi o zbadanie wydajności i użyteczności.

W pierwszym rodzaju testowania soft jest oceniany pod kątem interesariusza powierzającego swój software testerom. Oni mają zbadać, czy program spełnia swoje zadania określone przez kierownika projektu, interesariuszy. Weźmy na przykład zautomatyzowany zakład produkcyjny. Wszystkie funkcje wgranego softu na maszynie mają działać poprawnie.

W przeciwieństwie do testów funkcjonalnych, testy niefunkcjonalne nie muszą koniecznie sprawdzać samego softu. Wg mnie jest to testowanie wszystkiego na około, co mogło by być związane z przyszłymi warunkami pracy, przy maszynach, jakie gwarantuje wgrany na nich soft. Także bardziej podchodzi to pod ergonomie i przyjazność w obsłudze dla pracownika. Badana tutaj jest użyteczność całego oprogramowania.

2. Smoke testy - pierwszy etap testowania. Są to testy funkcjonalne tylko sztandarowych funkcji. Stosuje się to, kiedy jest mało czasu na poprawki softu. Do drugiego etapu testowania tj. całkowitego przechodzi się pod warunkiem, że w smoke-testach najważniejsze przetestowane funkcje działają zgodnie z wymaganiami. W innym przypadku soft wraca do dewelopera.

Testy regresji - testy te to po prostu kolejna egzekucja kompleksowej kampanii testowej. Jednak przypadki testowe nie są identyczne. Te które nie przeszły pomyślnie testów zostają przed testami regresji poprawione. Te testy wykonuje się, aby upewnić się, że zmiany pomogły, a nie wywołały kolejnych usterek w oprogramowaniu.

3. Cztery przypadki testowe

**Wartosci wejsciowe np. x= 2**, **y=1**

1) TC -> a==5, b== 3, x== 3, y== 0

2) TC -> a==-1, b== 1, **x== 2**, **y== 1**

3) TC -> a==5, b== 1, x== 3, **y== 1**

4) TC-> a==-1, b== 3, **x== 2**, y== 0

4. Testowanie zwinne - wg słownika terminów testowych testowanie zwinne jest używane przy technikach wytwarzania sterowanego testami. To oznacza, że zanim deweloper zacznie pisanie kodu, tworzy najpierw przypadki testowe oraz koncepcje testowania. Kiedy ta metoda nie jest używana to czasem zdarza się, że niektóre wymagania mogą być po prostu nietestowalne.

5. Wiek: <18:60) Poprawne wartosci: 18;60

Wartosci niepoprawne: 17;61

Inne wartosci niepoprawne: -5; ;0;150;a;&;60.5

Inne wartosci poprawne: 20; 42;5

6. Jeżeli dołączam do projektu podczas developementu, to oznacza, że wytwarzanie oprogramowania jest wg V-modelu i testowanie pojedynczych modułów oraz integracyjne są juz wykonane. Na tym etapie robi sie testy systemowe. Pierwsze o co bym pytała to kto wcześniej zajmował się testowaniem, ponieważ samo logowanie sie do aplikacji wydaje się jak by to był jeden moduł. Ktoś więc przede mną musiał to testować. Dodatkowe pytania:

- Co oznacza pierwszy/drugi input (e-mail,telefon, jakis nick/haslo)?

- Jak sprawdzic poprawność/niepowtarzalnosc wpisanego nicku/mailu/telefonu? (Ilość/rodzaj znakow,spacje w srodku?

- Ile maksymalnie osób może się w jednym czasie zalogowac?

- Czy uzytkownik ma jakas dozwolona ilosc prob zalogowania się? Co się ma stać jeżeli tą dowolną ilość prób przekroczy?

- Do jakiego rodzaju serwisu/konta jest to logowanie?

- Czy istnieje proces odzyskiwania konta?

7. Celem testowania jest wykazanie usterek w oprogramowaniu tzn. podejrzenie niewłaściwego działania oprogramowania oraz próba znalezienia prawdopodobnej przyczyny.