**ETAP I**

1. **Testy dymne** to podstawowe testy sprawdzające najważniejsze funkcje i działanie aplikacji, np. czy program się uruchamia. **Warto je wykonywać** po dostarczeniu kolejnej wersji kodu   
   w celu sprawdzenia, czy przeprowadzenie dalszych testów jest w ogóle możliwe oraz bezpośrednio po wdrożeniu, by upewnić się, że aplikacja działa poprawnie w nowym środowisku.

**Różnica pomiędzy *smoke testing* a *sanity testing*** polega na tym, że smoke testing jest testowaniem ogólnym aplikacji, zaś sanity testing weryfikuje czy logika aplikacji   
i funkcjonalności są zgodne z wymaganiami.

1. **Różnica miedzy testami funkcjonalnymi a testami niefunkcjonalnymi** polega na tym,  
   że testy funkcjonalne opierają się na funkcjach systemu (czynnościach wykonywanych przez system), zaś testy niefunkcjonalne określają jak działa system (określają m.in. wydajność, użyteczność, niezawodność, poziom obciążenia).
2. Pokrycie instrukcji i pokrycie decyzji zaliczamy do **grupy technik testowania opartych na strukturze / białoskrzynkowych**. **Zależność** polega na tym, że jest wykorzystywana wiedza   
   o strukturze kodu przy tworzeniu przypadków testowych i istnieje możliwość miary pokrycia. Testowanie pokrycia decyzji zawiera się w pokryciu instrukcji i jest bardziej precyzyjne.
3. Testowanie formularza segmentu klienta – **Analiza wartości granicznych**:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Przyp.** | **Zarobki** | **Segment** | **Wart.1** | **Wart.2** | **Wart.3** | **Wart.4** | **Wart.5** | **Wart.6** |
| 1 | < 2 000zł | Brak dostępu dla użytkownika | 1 999zł | 2 000zł | - | - | - | - |
| 2 | 2 001zł – 4 000zł | Standard | 2 000zł | 2 001zł | 2 002zł | 3 999zł | 4 000zł | 4 001zł |
| 3 | 4 001zł – 6 000zł | Premium | 4 000zł | 4 001zł | 4 002zł | 5 999zł | 6 000zł | 6 001zł |
| 4 | 6 001zł < | VIP | 6 001zł | 6 002zł | - | - | - | - |

\*czerwono – niepoprawne wartości graniczne

1. Protokół **HTTP jest protokołem bezstanowym**. Oznacza to, że ani serwer (ani klient) nie przechowuje informacji o tym, jakie były wcześniej zapytania pomiędzy określonym serwerem i klientem oraz nie posiada stanu wewnętrznego. Dzięki temu każde zapytanie do serwera traktowane jest jako nowe, z punktu widzenia serwera aplikacji niemożliwe do powiązania z informacjami (np. o zalogowanym użytkowniku).
2. **Uwierzytelnianie** jest weryfikacją tożsamości konkretnego użytkownika (np. za pomocą hasła), natomiast **autoryzacja** to sprawdzenie czy dana operacja dla użytkownika jest dozwolona.

Za nieudane uwierzytelnianie/autoryzację odpowiadają **kody błędów 401 i 403**: 401,1; 401,2; 401,3; 401,4; 401,5; 403,1; 403,2; 403,3, 403,4; 403,5; 403,6; 403,7; 403,8; 403,12; 403,13; 403,13; 403,16; 403,17; 403,18; 403,19.

1. **Android ADB** jest wierszem poleceń pozwalającym na komunikację z urządzeniem.

Przydatne funkcje w testowaniu:

- instalacja i deinstalacja aplikacji na urządzeniu;

- kopiowanie i wysyłanie plików z i na urządzenie/komputer;

- restart telefonu w trybie bootloader/recovery/sideload;

- wykonania logcata – podgląd do aplikacji działających w tle i czynności wykonywanych przez system;

- wykonywanie zrzutów ekranu;

- nagrywanie ekranu urządzenia;

- usuwanie blokad ekranu: pin/hasło oraz gest/wzór;

- wyświetlanie informacji na temat urządzenia;

- wykonanie kopii zapasowej aplikacji wraz z jej ustawieniami.

1. **Aplikacja natywna** tworzona jest pod konkretny system/platformę w skutek czego jest bardziej wydajna, zaś **hybrydowa** jest tworzona jest pod wiele platform, dzięki czemu ma większą dostępność.
2. **Rodzaje relacji** między tabelami w bazach danych opartych o SQL: jeden do jednego, jeden do wielu oraz wiele do wielu.

**Przykład:**

Mamy cztery tabele: Tabela\_Producenci, Tabela\_Samochody, Tabela\_Zamówienia, Tabela\_Sprzedawca.

\*Jeden do jednego: Tabela\_Samochody - Tabela\_Zamówienia, samochód posiada unikalny identyfikator (VIN). Unikalnym kluczem w obu tabelach jest kolumna VIN.

\*Jeden do wielu: Tabela\_ Producenci -> Tabela\_Samochody, gdzie każdy samochód produkowany jest przez jednego producenta. W tym przypadku w Tabela\_Samochody występuje kolumna określająca Producenta (np. Nazwa Producenta).

\*Wiele do wielu: Tabela\_Sprzedawca jest tabelą łączącą tabele: Tabela\_Zamówienia oraz Tabela\_Samochody, zawierać będzie np. Id Zamówienia i VIN Samochodu oraz inne dane.

1. **Języki skryptowe: Perl, PHP, Python, JavaScript**

W momencie uruchamiania kod jest interpretowany i wykonywany. Piszę się w nich szybciej, ale lepiej sprawdzają się w prostych aplikacjach. Szybciej widać nanoszone zmiany w kodzie.

**Języki kompilowane: C, C++, Pascal**

Przed uruchomieniem kodu należy go skompilować, przez to wszelkie zmiany w kodzie nie są od razu widoczne. Programy pisane językiem kompilowanym działają szybciej, a błędy są widoczne przed uruchomieniem programu. Łatwiej utrzymać program i go rozwijać.