**Certyfikowany tester**

ISTQB (International Software Testing Qualifications Board) jest certyfikatem testerskim. Certyfikat przez długi czas był podstawą do zatrudniania testerów, w dzisiejszych czasach trochę stracił już na rynku pracy, ale nadal warto wiedzieć z czym to się je. Co musi wiedzieć taki certyfikowany tester? Struktura sylabusa składa się z 6 rozdziałów, ale w tej notatce skupimy się na pierwszych czterech.

1. **Podstawy testowania**

**Typowe cele testowania:**

* zapobieganie defektom poprzez dokonywanie oceny produktów pracy, takich jak: wymagania, projekt czy kod
* weryfikacja wymagań
* sprawdzenie, czy przedmiot testów jest kompletny i czy działa zgodnie z oczekiwaniami.
* Wykrycie potencjalnych defektów i awarii => zmniejszenie ryzyka związanego z niedostateczną jakością programowania.
* dostarczanie interesariuszom informacji niezbędnych do podejmowania świadomych decyzji (dotyczących zwłaszcza poziomu jakości przedmiotu testów)
* przestrzeganie wymagań wynikających z umów, przepisów prawa i norm/standardów i/lub sprawdzanie, czy obiekt testów jest zgodny z tymi wymaganiami lub standardami.

**Testowanie/debugowanie:**

Testowanie i debugowanie nie jest tym samym. Testowanie ujawnia awarie, które są skutkiem defektów w oprogramowaniu, a debugowanie to czynność związana z wytwarzaniem oprogramowania (polegająca na znajdowaniu, analizowaniu i usuwaniu tych defektów). Debugowanie jest zwykle zadaniem dla programistów, a testowanie dla testerów.

**Pomyłki, defekty, i awarie – czym się różnią?**

Pomyłki (error/mistake) – są to zazwyczaj pomyłki popełniane przez programistę prowadzące do błędu w programie. Mogą one wynikać np. z niezrozumienia wymagań programu czy błędu w obliczaniu wartości. Wynikiem pomyłki jest błąd (bug).

Defekt/usterka – jest to odchylenie od wymagań. Jest to nieprawidłowość, przez którą program działa niepoprawnie, lub podaje nieoczekiwany rezultat. Niepowodzenie jest efektem usterki.

**Siedem zasad testowania**

1. **Testowanie ujawnia usterki:** Testowanie może wykazać obecność defektów, ale **nie może dowieść, że oprogramowanie jest od nich wolne.** Co oznacza, że ZAWSZE w oprogramowaniu mogą istnieć nieprawidłowości, które nie zostały wykryte przez testera.

2. **Testowanie gruntowne jest niemożliwe:** co prostszym językiem można powiedzieć - **przetestowanie wszystkiego jest niemożliwe** (może poza najprostszymi przypadkami). W związku z tym, tester powinien kierować swoje wysiłki w celu znalezienia najbardziej niebezpiecznych lub najpoważniejszych nieprawidłowości.

3. **Wczesne testowanie oszczędza czas i pieniądze.** Wykonywanie testów na wczesnym etapie cyklu życia oprogramowania pozwala ograniczyć lub wyeliminować kosztowne zmiany.

4. **Kumulowanie się defektów:** defekty oprogramowania często są **skumulowane w konkretnych miejscach oprogramowania.** Powodem tego może być skomplikowana logika funkcji. Zadaniem testera jest wskazanie takich „hot-spotów”, w których mogą istnieć skupiska błędów.

5. **Paradoks pestycydów**: spryskując swoje pole tym samym środkiem owadobójczym po jakimś czasie przestanie dawać wystarczające efekty. Tak samo jest z testowaniem – dlatego ważnym jest, aby ciągle projektować nowe testy.

6**. Testowanie zależy od kontekstu**: tester inaczej będzie testował aplikację bankową niż stronę internetową sklepu. Związane jest to z innymi podatnościami. Zależnie od poziomu ryzyka tester musi dobrać odpowiednie techniki i metody testowania.

7. **Przekonanie o braku błędów jest błędem**: to, że produkt nie posiada defektów nie znaczy, że spełnia wymagania. Klientów nie interesuje to, ile defektów zostało znalezionych tylko to, czy oprogramowanie wykonywało założone zadania w sposób efektywny.

1. **Testowanie w cyklu życia oprogramowania**

Istnieją różne modele projektowania oprogramowania, ale tutaj skupimy się na dwóch najpopularniejszych:

• sekwencyjne modele wytwarzania oprogramowania: sekwencyjne modele opierają się na założeniu, że kończąc jedną fazę dopiero możemy przejść do następnej fazy. Poszczególne fazy nie powinny na siebie nachodzić. Testowanie oprogramowania następuje dopiero wtedy, kiedy wszystkie poprzednie fazy zostaną zakończone. Przykładem takiego modelu może być tzw. waterfall (wodospad).

• iteracyjne i przyrostowe modele wytwarzania oprogramowania: w tym przypadku fazy wytwarzania oprogramowania mogą na siebie nachodzić. Testowanie następuje po zakończeniu każdej iteracji. Przykładem takiego modelu może być SCRUM, w którym dodawane funkcjonalności są stosunkowo małe, a iteracje trwają krótko.

1. **Testowanie statyczne**

Testowanie statyczne jest przeciwną metodą do testowania dynamicznego. Przypomnijmy – testowanie dynamiczne wymaga uruchomienia testowanego oprogramowania, a statyczne polega na badaniu produktów pracy lub dokonywaniu oceny kodu. Testowanie statyczne pozwala wykryć defekty bezpośrednio w produktach pracy, a nie na podstawie spowodowanych przez te defekty awarii.

Analiza statyczna jest ważna w przypadku oprogramowania stosowanego w dziedzinach, w których nieprawidłowość może mieć bardzo złe skutki (medycyna, inżynieria), ale staje się coraz bardziej istotna w innych kontekstach (testowanie zabezpieczeń).

Testowanie statyczne dostarcza wielu korzyści, może pomóc wykryć defekty jeszcze przed rozpoczęciem testowania dynamicznego (przegląd specyfikacji wymagań, niespójności, niejasności w założeniach projektu). Usunięcie defektów na tak wczesnym etapie projektu jest dużo tańsze niż zwlekanie z tym na później.

1. **Techniki testowania**

**Kategorie technik testowania:**

* testy czarnoskrzynkowe
* testy białoskrzynkowe
* testy oparte na doświadczeniu

Podstawowe różnice między testami czarnoskrzynkowymi i białoskrzynkowymi zostały omówione (NUMER ALBO NAZWA TWOJEJ NOTATKI Z TESTOW).

**Techniki testów czarnoskrzynkowych:**

- **podział na klasy równoważności:** technika ta polega na podzieleniu danych na grupy lub nawet podgrupy, za pomocą których tester będzie sprawdzał poprawność działania oprogramowania. **Przykład:** tester chce sprawdzić, czy hasło zapisuje się w sposób poprawny. Hasło powinno zawierać od 4 do 10 liter. Tester wybiera trzy klasy: 1. niepoprawną (hasło posiada od 0 do 3 liter), 2. poprawną (hasło posiada od 4 do 10 liter) oraz 3. niepoprawną (hasło posiada od 10 liter wzwyż).

- **badanie wartości brzegowych:** technika ta jest rozszerzeniem powyższej techniki. Polega ona na uwzględnieniu wartości brzegowych w badanym przypadku. **Przykład:** tester chce sprawdzić, czy hasło zapisuje się w sposób poprawny. Hasło powinno zawierać od 4 do 10 liter. Tester sprawdza następujące hasła: hasło posiada od 1 do 2 liter, hasło posiada 3 litery (warunek brzegowy pierwszej klasy z poprzedniego przykładu), hasło posiada 4 litery (warunek brzegowy drugiej klasy), hasło posiada od 5 do 9 liter, hasło posiada 10 liter (warunek brzegowy drugiej klasy), hasło posiada 11 liter (warunek brzegowy 3 klasy), hasło posiada więcej niż 11 liter.

- **badanie w oparciu o tablicę decyzyjną:** technika ta polega na stworzeniu tablicy decyzyjnej. Przykład: tester chce sprawdzić, czy hasło zapisuje się w sposób poprawny. Hasło powinno zawierać o 4 do 10 liter. Tester następnie przygotowuje tablice, w której są następujące warunki 1. hasło posiada od 0 do 3 liter, 2. hasło posiada od 4 do 10 liter oraz 3. hasło posiada od 10 liter wzwyż. Następnie zaznacza T jeśli udało mu się utworzyć konto, oraz N jeśli nie udało. Tabela wyglądałaby następująco:

|  |  |  |
| --- | --- | --- |
| 1. hasło 0-3 | 2. hasło 4-10 | 3. hasło 10 wzwyż |
| N | T | N |

**Techniki oparte na doświadczeniu:**

- **zgadywanie błędów:** technika ta polega na utworzeniu listy najbardziej popularnych błędów w oprogramowaniu i sprawdzaniu, czy nie występują. Listę te można opracować na podstawie własnego doświadczenia lub powszechnej wiedzy na temat przyczyn awarii oprogramowania.

- **testowanie eksploracyjne:** technika ta polega na projektowaniu, wykonywaniu i ocenianiu testów w sposób dynamiczny podczas ich wykonywania. To jest właśnie to przeklikiwanie się przez stronę, aż znajdzie się jakąś nieprawidłowość.

- **testowanie w oparciu o listę kontrolną:** w tej technice testowania testerzy projektują testy tak, aby pokryć warunki testowe wymienione w liście kontrolnej. Listy te są tworzone wcześniej przez testerów, ale można też skorzystać z gotowych. Są one opracowywane na podstawie własnego doświadczenia, oczekiwań użytkowników lub wiedzy na temat przyczyn awarii oprogramowania.

**Pytania sprawdzające:**

**1. Która z poniższych odpowiedzi prawidłowo wyraża jeden z celów testowania?**

**a)** Testy powinny rozpoczynać się tak późno, jak to tylko możliwe, aby programiści mieli wystarczająco dużo czasu na stworzenie dobrego produktu.

**b)** Sprawdzenie, czy obiekt testowy działa zgodnie z oczekiwaniami użytkowników i innych interesariuszy.

**c)** Potwierdzenie, że zidentyfikowano wszystkie możliwe defekty.

**d)** Potwierdzenie, że ewentualne nieusunięte defekty nie spowodują żadnych awarii.

**2. Która z poniższych odpowiedzi opisujących typowe sytuacje zaistniałe podczas testowania lub eksploatacji opisuje awarię?**

**a)** Produkt przestał działać po wybraniu przez użytkownika jednej z opcji w oknie dialogowym.

**b)** W kompilacji uwzględniono niewłaściwą wersję jednego z plików kodu źródłowego.

**c)** Algorytm obliczeniowy korzystał z niewłaściwych zmiennych wejściowych.

**d)** Programista błędnie zinterpretował jedno z wymagań dotyczących algorytmu.

**3. Wskaż DWA stwierdzenia dotyczące testowania statycznego, które są ZASADNICZO prawdziwe.**

**a)** Testowanie statyczne umożliwia wykrywanie i usuwanie defektów niewielkim kosztem.

**b)** Testowanie statyczne ułatwia późniejsze testowanie dynamiczne.

**c)** Testowanie statyczne pozwala dokonać wczesnej walidacji wymagań użytkowników.

**d)** Testowanie statyczne umożliwia wykrycie problemów związanych z wykonywaniem kodu na wczesnym etapie cyklu życia oprogramowania.

**e)** W przypadku testowania systemu krytycznego ze względów bezpieczeństwa testowanie statyczne ma mniejszą wartość, ponieważ testowanie dynamiczne pozwala skuteczniej wykryć defekty.

**4. Testowanie w oparciu o listę kontrolną to:**

**a)** Technika testowania, w której testy są wyprowadzane w oparciu o wiedzę testera na temat przeszłych awarii lub generalną wiedzę o trybach możliwych awarii.

**b)** Technika testowa oparta o analizę specyfikacji modułu lub systemu.

**c)** Technika projektowania testów oparta na doświadczeniu, w której doświadczony tester używa listy ogólnych zagadnień, które powinny być odnotowywane, sprawdzone, zapamiętane, lub zbioru reguł bądź kryteriów, względem których produkt ma być sprawdzany.

**d)** Podejście do testowania, w którym testerzy dynamicznie projektują i przeprowadzają testy na podstawie swojej wiedzy, badania obiektu testów i wyników z poprzednich testów.

**5. W przypadku aplikacji do odtwarzania wideo obowiązuje następujące wymaganie: aplikacja musi umożliwiać odtwarzanie wideo na ekranach o następującej rozdzielczości:**

1. 640×480

2. 1280×720

3. 1600×1200

4. 1920×1080

**Który z poniższych przypadków testowych jest rezultatem zastosowania technik podziału na klasy równoważności w odniesieniu do tego wymagania?**

**a)** Zweryfikowanie, czy aplikacja umożliwia odtwarzanie filmu wideo na ekranie o rozdzielczości 1920×1080 (1 przypadek testowy).

**b)** Zweryfikowanie, czy aplikacja umożliwia odtwarzanie filmu wideo na ekranach o rozdzielczości 640×480 i 1920×1080 (2 przypadki testowe).

**c)** Zweryfikowanie, czy aplikacja umożliwia odtwarzanie filmu wideo we wszystkich rozdzielczościach wymienionych w wymaganiu (4 przypadki testowe).

**d)** Zweryfikowanie, czy aplikacja umożliwia odtwarzanie filmu wideo w dowolnej z rozdzielczości wymienionych w wymaganiu (1 przypadek testowy).

**6. Która z poniższych odpowiedzi NAJLEPIEJ opisuje testowanie eksploracyjne?**

**a)** Sposób testowania polegający na wnikliwym zbadaniu kontekstu przedmiotu testów w celu zidentyfikowania potencjalnych słabych punktów, które należy sprawdzić przy użyciu przypadków testowych.

**b)** Podejście do testowania, w którym testerzy dynamicznie projektują i przeprowadzają testy na podstawie swojej wiedzy, badania obiektu testów i wyników z poprzednich testów.

**c)** Podejście do testowania, w którym zadania testowe są planowane jako nieprzerywalne sesje projektowania i wykonywania testów, często stosowane w połączeniu z testowaniem opartym na liście kontrolnej.

**d)** Testowanie oparte na doświadczeniu testera, jego wiedzy i intuicji.

**7. Właściciel produktu twierdzi, że Twoja rola jako testera w zespole zwinnym polega na znajdowaniu wszystkich defektów przed końcem każdej iteracji. Którą z poniższych zasad testowania należy zastosować, aby odpowiedzieć na to (nieprawdziwe) stwierdzenie?**

**a)** Kumulowanie się defektów.

**b)** Testowanie ujawnia usterki.

**c)** Mylne przekonanie o braku błędów.

**d)** Analiza przyczyny podstawowej.

**Zakończenie**

To są podstawy tego, co powinien wiedzieć zarówno certyfikowany tester, jak i osoba będąca devem. Tworzenie oprogramowania musi polegać na współpracy testerów jak i programistów, dlatego tak ważne jest, aby każda ze stron rozumiała zadania tej drugiej.

Odpowiedzi:

1 - B

2 - A

3 - A, C

4 - C

5 - C

6 - B

7 - B