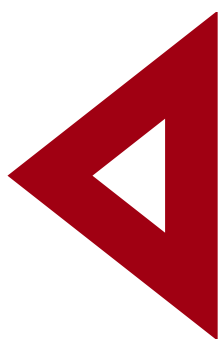




**21 PYTAŃ
REKRUTACYJNYCH
DLA TESTERA**



CZATESTOW.PL

Chyba każdy przed rozmową rekrutacyjną chciałby sprawdzić oraz odświeżyć swoją wiedzę. Poniżej znajdziecie listę 21 pytań które mogą paść na rozmowie rekrutacyjnej. To nie są wszystkie możliwe pytania a jedynie przykłady z czym można się spotkać. Warto pamiętać, że podstawą dla rekruterów jest CV jakim dysponują, dlatego przygotowując się na rozmowę warto nie tylko prześledzić pytania z wiedzy teoretycznej, ale również narzędzia które pojawiają się w naszym życiorysie.

1. Co to są testy funkcjonalne i niefunkcjonalne?

Testy funkcjonalne to testy sprawdzające czy dana funkcjonalność spełnia założone wymagania. Sprowadzają się do testowania funkcjonalności i zadanej logiki działania. Ogólnie mówiąc, testy te sprawdzają jak działa dana rzecz.

Testy niefunkcjonalne to w dużej mierze określania parametrów działania, można wyróżnić m.in. testy obciążeniowe, testy wydajnościowe, testy użyteczności, testy niezawodności.

Dla przykładu, jeśli mamy samochód, i testujemy hamulce, to sam fakt hamowania jest testem funkcjonalnym. Określenie parametrów, np. ile wynosi droga hamowania – jest testem niefunkcjonalnym.

2. Czym są testy czarnoskrzynkowe? I Jaka jest różnica w porównaniu do testów białej skrzynki?

Test czarnej skrzynki to test funkcjonalny. Polega on na zbadaniu rezultatu przy zadanych wartościach wejściowych. Wszystko co jest pomiędzy startem a wynikiem jest traktowane jako czarna skrzynka i nie wiemy co w niej jest.

Testowanie białoskrzynkowe polega na przeglądzie, a właściwie podglądzie kodu. Jest to sprawdzanie kodu i algorytmów testowanego programu. W tym przypadku testujący ma dostęp do całej logiki zawartej w programie lub funkcji.

3. Metody wytwarzania oprogramowania?

- Waterfall
- Model V
- Model iteratywny(przyrostowy)

4. Czym są wartości brzegowe?

Wartości brzegowe dotyczą testowania przedziałów. Na końcach przedziałów jest największe prawdopodobieństwo wystąpienia nieprawidłowości w działaniu. Są dwa rodzaje wartości brzegowych poprawne (pozytywne) i niepoprawne (negatywne). Dla przykładu, powiedzmy że mamy zmienną która może przyjmować wartości od 1 do 10. Dla takiego przedziału wartości brzegowe pozytywne to 1 i 10, natomiast wartości negatywne to 0 i 11.

5. Co robisz gdy znajdziesz błąd?

- Sprawdzasz czy błąd jest powtarzalny
- Sprawdzasz czy zostały spełnione wszystkie warunki początkowe
- Raportujesz błąd

6. Skąd wiadomo, że coś jest błędne?

Po pierwsze dokumentacja – wszystko co jest niezgodne z wymaganiami jest błędne. W przypadku gdy brak dokumentacji lub jest ona niewystarczająca trzeba ratować się doświadczeniem (swoim jak i całego zespołu), logiką i zdrowym rozsądkiem. Zawsze lepiej jest zgłosić potencjalny problem, niż przeoczyć ewentualny kłopot.

7. Co powinien zawierać scenariusz testowy?

- Czego dotyczą testy (opis)
- Wersja która powinna być testowana
- Warunki początkowe
- Dokładny opis poszczególnych kroków
- Wartość oczekiwana
- Kryteria Akceptacji

8. Czy można stworzyć idealne testy automatyczne zastępujące człowieka?

To pytanie jest podchwytliwe, ponieważ odpowiedź brzmi teoretycznie 'tak', ale w praktyce 'nie'. Chodzi o to, że napisanie testów automatycznych które pokryją 100% przypadków testowych jest nieopłacalne i prawie niemożliwe do wykonania ze względu na brak elastyczności testów automatycznych.

9. Różnica między weryfikacją a walidacją

Oba słowa oznaczają sprawdzanie jednak diabeł tkwi w szczegółach i nie są sobie równoznaczne. Weryfikacja jest podejściem statycznym i odpowiada na pytanie czy system spełnia wymogi swojej specyfikacji. Walidacja jest podejściem dynamicznym i daje rezultat czy oprogramowanie zachowuje się właściwie.

10. Jakie znasz poziomy testów

- Modułowe
Testy modułowe polegają na testowaniu pojedynczych funkcjonalności.
- Integracyjne
Testy integracyjne sprawdzają interakcje pomiędzy modułami (zbiorami funkcjonalności)
- Systemowe
Testy systemowe są przeprowadzane globalnie dla kompletnego systemu
- Akceptacyjne
Testy akceptacyjne służą jako metoda ostatniej weryfikacji dostarczanego oprogramowania.

11. Czy testowanie jest potrzebne?

Odpowiedź może być tylko jedna, tak! Dlaczego? Po pierwsze człowiek jest omylny i często nie widzimy własnych błędów. Kolejną sprawą jest sprawdzenie wymagań, tu znowu wchodzi w grę czynnik ludzki, ta sama informacja może być rozumiana inaczej przez różne osoby. Ponadto, testowanie to zapewnianie jakości, a więc sprawdzenie i zagwarantowanie że dany produkt, będzie spełniał odpowiednie normy i wymagania.

12. Co to jest TDD i BDD?

TDD czyli Test Driven Development

BDD czyli Behaviour Driven Development

Oba skróty dotyczą metod wytwarzania oprogramowania. W przypadku TDD testy powstają jako pierwsze i program jest pisany tak aby testy zostały spełnione. BDD jest podobne, jednak opiera się na zachowaniu programu zamiast na testach. BDD opiera się na podstawowym scenariuszu Given -> When -> Then, czyli „dana rzecz” w określonym „przypadku” daje takie „rezultaty”.

13. Czym są testy regresywne

Wg definicji: Testowanie regresywne: Ponowne przetestowanie uprzednio testowanego programu po dokonaniu w nim modyfikacji, w celu upewnienia się, że w wyniku zmian nie powstały nowe defekty lub nie ujawniły się wcześniej nie wykryte w nie zmienionej części oprogramowania. Testy takie są przeprowadzane po zmianach oprogramowania lub jego środowiska pracy.

14. Jak zareagować na słynne “U mnie działa”?

Ma działać na środowisku produkcyjnym. Środowisko testowe powinno być zbliżone do środowiska produkcyjnego, więc jak coś nie działa na testowym środowisku, to nie działa. Tester czasami musi być lekko gruboskórny i opierać się jedynie na faktach.

15. Co to jest smoke test?

Smoke test to szybkie powierzchowne sprawdzenie całego systemu. Zgodnie z definicją: Smoke test (test dymny) to podzbiór przypadków testowych, które pokrywają główne funkcjonalności modułu lub systemu, mający na celu potwierdzenie, że kluczowe funkcjonalności programu działają, bez zagłębiania się w szczegóły.

16. Jaka jest rola testera w Scrumie?

To pytanie może być bardziej ogólne i dotyczyć samego procesu Scruma. Praca w Scrumie opiera się na samoorganizacji zespołu. Dla testera oznacza to brak oczekiwania na koniec prac programisty, tylko współpracę. Umożliwia to szybsze znalezienie defektów, jeszcze przed finalnym zbudowaniem funkcji. Warto jeszcze raz podkreślić, że w Scrumie każdy członek jest tak samo ważny i powinien brać udział we wszystkich wydarzeniach.

17. Czy jest możliwe testowanie bez wymagań?

Jest to możliwe, jednak zawsze potrzebne są jakieś wymagania. Przy testowaniu bez wymagań musimy zrobić założenia które będą kryteriami akceptacji dla naszych testów.

18. Kiedy stosować testy automatyczne?

Jest kilka przypadków gdy stosowanie testów automatycznych jest zalecane. Najczęściej jest to automatyzacja testów regresyjnych oraz automatyzacja testów powtarzalnych. Trzeba pamiętać że napisanie testu automatycznego pochłania więcej czasu niż testy manualne, dlatego zawsze warto zastanowić się, czy dany test będzie łatwy w utrzymaniu oraz przyniesie korzyści.

19. Co powinno zawierać zgłoszenie błędu?

Zgłoszenie błędu powinno zawierać wszystkie potrzebne informacje konieczne do identyfikacji błędu, mianowicie:

- wersja i środowisko na którym przeprowadzony został test
- opis błędu (najlepiej poparty odniesieniem do dokumentacji)
- scenariusz pozwalający na odtworzenie błędu

20. Jak przetestować "szklankę"?

Jak przetestować krzesło? Jak przetestować telefon? Ogólnie jak przetestować cokolwiek. Pytanie ma za zadanie sprawdzić nasze podejście do nowego zagadnienia. Należy zacząć zgodnie ze sztuką, czyli od dokumentacji. Jeśli brakuje dokumentacji trzeba zrobić założenia. Szklanka z pytania może służyć do picia, podsłuchiwania, jako świecznik czy przyrząd do wykrawania ciasteczek. Trzeba wiedzieć co się testuje. Gdy już mamy zdefiniowane funkcjonalności, możemy przejść do wymyślenia przypadków i scenariuszy testowych. Dobrym pomysłem jest przejście przez 'happy path' czyli scenariusz zgodny z oczekiwanym zachowaniem, oraz scenariusz który powinien być niepoprawny.

21. Ile jest drzew w lesie?

Pytanie dziwne i z gatunku „kreatywnych” pytań rekrutacyjnych. Sama odpowiedź jest drugorzędna, zdecydowanie ważniejszy jest proces uzyskania odpowiedzi. Głównie chodzi o robienie założeń, czyli zakładamy że las to duże skupisko drzew, czyli o powierzchni powyżej 1ha. Powiedzmy że nasz las ma 5ha, czyli 50000 m². Kolejnym założeniem jest fakt, że jedno drzewo zajmuje ok 10 m². Dzielimy więc obszar lasu, przez obszar drzewa i uzyskujemy wynik 5000 drzew. Trochę dziwne prawda? Dla mnie też, ale niektóre pytania takie są.