

Rafał Podraza

# Ścieżka testera



Certyfikat

**ISTQB**

w pytaniach  
i odpowiedziach



Helion 

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Małgorzata Kulik

Projekt okładki: Studio Gravite / Olsztyn

Obarek, Pokoński, Pazdrijowski, Zaprucki

Grafika na okładce została wykorzystana za zgodą Shutterstock.com

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <https://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<https://helion.pl/user/opinie/istqbw>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-9802-3

Copyright © Helion S.A. 2023

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

O autorze	5
Wprowadzenie	7
<b>Część I. Pytania</b>	<b>  9</b>
<b>Część II. Odpowiedzi</b>	<b>  105</b>
Bibliografia	271



# Wprowadzenie

Książka *Ścieżka testera. Certyfikat ISTQB w pytaniach i odpowiedziach* idealnie sprawdzi się jako lektura dla osób, które przygotowują się do egzaminu ISTQB Poziomu Podstawowego (*Foundation Level*). Jej celem jest pokazanie czytelnikowi, z jakimi pytaniami może się zmierzyć na egzaminie. Materiał zawiera 400 przykładowych pytań wraz z odpowiedziami, obejmujących swoim zakresem różne rozdziały sylabusu, np. związanych z procesem testowym, technikami testowania czy procesami przeglądu. Pytania zostały dobrane tak, aby były jak najbardziej zbliżone do tych, które pojawiają się na oficjalnym egzaminie. Każde z nich zawiera cztery odpowiedzi, z których tylko jedna jest poprawna.

W drugiej części książki do wszystkich pytań dołączone są poprawne odpowiedzi wraz z uzasadnieniami, do których można zajrzeć w każdej chwili. Jednak warto na początku odpowiedzieć na nie samemu i zweryfikować swoją wiedzę.

Zaleca się, aby przed przystąpieniem do analizowania pytań zawartych w *Ścieżce testera* czytelnik sięgnął po sylabus poziomu podstawowego, który jest dostępny na oficjalnej stronie SJSI. Książki nie należy traktować jako zamiennika i nie może ona zastąpić oficjalnych źródeł/szkoleń.

Fragmety *Ścieżki testera* bazują na następujących źródłach:

- wyjaśnienia — *Certyfikowany tester. Sylabus poziomu podstawowego ISTQB®* opracowany przez SJSI (wersja 2018 3.1);
- pojęcia, definicje — słownik związany z testowaniem: *The ISTQB® Glossary*, © by International Software Testing Qualifications Board.



# Część I.

# Pytania







**001.** Która z poniższych opcji zapobiegnie ponownemu występowaniu defektów?

- A. Rotacja programistów, aby utrzymać ich motywację.
- B. Określenie warunków środowiskowych, które spowodowały awarię.
- C. Usprawnienie procesów na podstawie analizy przyczyn źródłowych.
- D. Nadanie priorytetu powtarzającym się defektom.

**002.** Poniżej znajduje się lista problemów, które można zaobserwować podczas testowania lub eksploatacji. Co jest **NAJBARDZIEJ** prawdopodobną awarią?

- A. Produkt uległ uszkodzeniu, gdy użytkownik wybrał opcję w oknie dialogowym.
- B. W wyniku kompilacji została użyta niewłaściwa wersja pliku z kodem źródłowym.
- C. Algorytm obliczeniowy użył niewłaściwych zmiennych wejściowych.
- D. Deweloper błędnie zinterpretował wymagania dotyczące algorytmu.

**003.** Które z poniższych stwierdzeń są **PRAWDZIWE**?

- I. Testy oprogramowania mogą być niezbędne w celu spełnienia wymagań prawnych lub umownych.
  - II. Testowanie oprogramowania jest potrzebne głównie do poprawy jakości pracy programisty.
  - III. Rygorystyczne testowanie i naprawianie wykrytych defektów mogą pomóc zmniejszyć ryzyko wystąpienia problemów w środowisku operacyjnym.
  - IV. Rygorystyczne testy są czasami stosowane w celu udowodnienia, że wszystkie awarie zostały znalezione.
- A. Stwierdzenia II i III są prawdziwe, a stwierdzenia I i IV są fałszywe.
  - B. Stwierdzenia I i IV są prawdziwe, a stwierdzenia II i III są fałszywe.
  - C. Stwierdzenia I i III są prawdziwe, a stwierdzenia II i IV są fałszywe.
  - D. Stwierdzenia III i IV są prawdziwe, a stwierdzenia I i II są fałszywe.

**004. Dlaczego ważne jest unikanie paradoksu pestycydów?**

- A. Testowanie dynamiczne jest mniej skuteczne w znajdowaniu błędów.
- B. Pestycydy wymieszane z testami statycznymi mogą sprawić, że błędy nie zostaną wykryte.
- C. Testowanie nie powinno być zależne od kontekstu.
- D. Ciągłe przeprowadzanie tych samych testów zmniejszy szansę na znalezienie nowych defektów.

**005. Które z poniższych stwierdzeń dotyczących testów gruntownych jest prawdziwe?**

- A. Jest to forma testów przeciążeniowych.
- B. Nie są one wykonalne (chyba że dotyczą bardzo prostego oprogramowania).
- C. Zwykle są one wykonywane za pomocą automatyzacji testów.
- D. Zwykle odpowiada za nie programista podczas testów jednostkowych.

**006. Które z poniższych stwierdzeń najlepiej opisuje jedną z siedmiu kluczowych zasad testowania oprogramowania?**

- A. Testy automatyczne lepiej niż testy manualne pomagają uniknąć testowania gruntownego.
- B. Testowanie gruntowne, przy wystarczającym wysiłku i wsparciu narzędziowym, jest wykonalne dla każdego oprogramowania.
- C. Zwykle przetestowanie wszystkich kombinacji wejścia/wyjścia dla danego systemu oprogramowania nie jest możliwe.
- D. Celem testowania jest wykazanie braku defektów.

**007. Zespół testowy konsekwentnie znajduje od 95% do 97% defektów w testowanym systemie. Kierownik testów wie, że jest to dobry wynik zarówno dla jego zespołu, jak i dla całej branży, jednak kierownictwo wyższego szczebla jest rozczarowane grupą testową i twierdzi, że pomija ona zbyt wiele błędów. Biorąc pod uwagę to, że użytkownicy są generalnie zadowoleni z systemu i że zaistniałe awarie miały niewielki wpływ na jego działanie — wskaż, która z poniższych zasad testowania najprawdopodobniej pomoże kierownikowi testów wyjaśnić menedżerom i kierownikom wyższego szczebla, dlaczego niektóre defekty mogą być pominięte.**

- A. Testowanie gruntowne jest niemożliwe.
- B. Paradoks pestycydów.
- C. Kumulowanie się defektów.
- D. Przekonanie o braku błędów jest błędem.

**008.** Programista pracuje nad bardzo złożonym kodem. Która z poniższych ogólnych zasad testowania może wpłynąć na jego pracę?

- A. Kumulowanie się defektów.
- B. Paradoks pestycydów.
- C. Testowanie gruntowne jest niemożliwe.
- D. Przekonanie o braku błędów jest błędem.

**009.** Według słownika ISTQB synonimem którego z poniższych słów jest „pluskwa”?

- A. Ulepszenie.
- B. Incydent.
- C. Defekt.
- D. Pomyłka.

**010.** Jaką cechę musi posiadać tester, aby odnieść sukces w pracy z zespołem projektowym?

- A. Umiejętność komunikowania się w konstruktywny sposób.
- B. Autorytarny styl przywództwa.
- C. Ekstrawertyczną osobowość.
- D. Brak poprawnej odpowiedzi.

**011.** Który z poniższych elementów — z perspektywy testera — jest kluczem do skutecznego komunikowania się i utrzymywania pozytywnych relacji z programistami w przypadku sporu dotyczącego priorytetu defektu?

- A. Przekazanie sprawy do działu kadr i podkreślenie znaczenia wzajemnego szacunku.
- B. Rozmowa z kierownictwem wyższego szczebla, aby upewnić się, że wszyscy rozumieją, na jakiej podstawie tester ustalił priorytet błędu.
- C. Przekonanie dewelopera, aby wziął na siebie winę za pomyłkę.
- D. Przypomnienie im o wspólnym celu tworzenia systemów dobrej jakości.

**012.** Które z poniższych sytuacji prowadziłyby do problemów (lub konfliktów) w zespołach podczas przeglądów i testów?

- A. Testerzy i przeglądający nie są na tyle ciekawi, aby znajdować defekty.
- B. Testerzy i przeglądający nie mają wystarczających kwalifikacji, aby znaleźć awarie i usterki.
- C. Testerzy i przeglądający komunikują defekty jako krytykę pod adresem osób, a nie oprogramowania.
- D. Testerzy i przeglądający oczekują, że defekty oprogramowania zostały już znalezione i naprawione przez programistów.

**013.** Które z poniższych zaleceń pomagają w dobrej komunikacji, a które ją utrudniają?

- I. Spróbuj zrozumieć, jak czuje się druga osoba.
  - II. Przekazuj subiektywne odczucia, skoncentrowane na konkretnych osobach.
  - III. Potwierdź, że druga osoba zrozumiała to, co powiedziałeś, i na odwrót.
  - IV. Podkreśl wspólny cel tworzenia lepszej jakości oprogramowania.
  - V. Każda dyskusja to bitwa do wygrania.
- A. Zalecenia I, II i III stanowią pomoc, a zalecenia IV i V stanowią utrudnienie.
  - B. Zalecenia III, IV i V stanowią pomoc, a zalecenia I i II stanowią utrudnienie.
  - C. Zalecenia I, III i IV stanowią pomoc, a zalecenia II i V stanowią utrudnienie.
  - D. Zalecenia II, III i IV stanowią pomoc, a zalecenia I i V stanowią utrudnienie.

**014.** Podczas których czynności w podstawowym procesie testowym następuje projektowanie i ustalanie priorytetów przypadków testowych?

- A. Podczas planowania testów.
- B. Podczas analizy testów.
- C. Podczas projektowania testów.
- D. Podczas ukończenia testów.

**015.** Podczas których czynności w podstawowym procesie testowym ma miejsce definiowanie i ustalanie priorytetów warunków testowych oraz uwzględnianie cech funkcjonalnych i niefunkcjonalnych?

- A. Podczas planowania testów.
- B. Podczas analizy testów.
- C. Podczas projektowania testów.
- D. Podczas ukończenia testów.

**016.** Podczas której czynności w podstawowym procesie testowym następuje sprawdzenie, czy wszystkie zgłoszenia defektów są zamknięte?

- A. Podczas planowania testów.
- B. Podczas analizy testów.
- C. Podczas projektowania testów.
- D. Podczas ukończenia testów.

# Część II.

# Odpowiedzi





**001.** Która z poniższych opcji zapobiegnie ponownemu występowaniu defektów?

- A. Rotacja programistów, aby utrzymać ich motywację.
- B. Określenie warunków środowiskowych, które spowodowały awarię.
- C. **Usprawnienie procesów na podstawie analizy przyczyn źródłowych.**
- D. Nadanie priorytetu powtarzającym się defektom.

#### Wyjaśnienie

- A. **Rotacja programistów** nie zapobiegnie ponownemu występowaniu defektów, a może nawet zwiększyć prawdopodobieństwo ich wystąpienia.
- B. **Określenie warunków środowiskowych** prędzej pomoże w rozwiązaniu już wykrytych wad, niż zapobiegnie ich ponownemu wystąpieniu.
- C. **Usprawnienie procesu na podstawie analizy przyczyn źródłowych** zapobiegnie występowaniu defektów w przyszłości, ponieważ przyczyna, która je spowodowała, nie będzie już istnieć.
- D. **Nadanie priorytetu powtarzającym się defektom** spowoduje szybsze rozwiązywanie usterek, ale nie zapobiegnie ich ponownemu występowaniu.

**002.** Poniżej znajduje się lista problemów, które można zaobserwować podczas testowania lub eksploatacji. Co jest NAJBARDZIEJ prawdopodobną awarią?

- A. **Produkt uległ uszkodzeniu, gdy użytkownik wybrał opcję w oknie dialogowym.**
- B. W wyniku kompilacji została użyta niewłaściwa wersja pliku z kodem źródłowym.
- C. Algorytm obliczeniowy użył niewłaściwych zmiennych wejściowych.
- D. Deweloper błędnie zinterpretował wymagania dotyczące algorytmu.

#### Wyjaśnienie

- A. **Produkt uległ uszkodzeniu** — w większości przypadków spowodowanie uszkodzenia lub zawieszenia produktu albo aplikacji jest traktowane jako awaria.
- B. **Niewłaściwa wersja kodu źródłowego** — jest to usterka spowodowana błędem popełnionym przez programistę.
- C. **Algorytm obliczeniowy użył niewłaściwych zmiennych wejściowych** — jest to usterka spowodowana błędem popełnionym przez programistę.
- D. **Deweloper błędnie zinterpretował wymagania** — jest to błąd popełniony przez dewelopera.

**003.** Które z poniższych stwierdzeń są PRAWDZIWE?

- I. Testy oprogramowania mogą być niezbędne w celu spełnienia wymagań prawnych lub umownych.
  - II. Testowanie oprogramowania jest potrzebne głównie do poprawy jakości pracy programisty.
  - III. Rygorystyczne testowanie i naprawianie wykrytych defektów mogą pomóc zmniejszyć ryzyko wystąpienia problemów w środowisku operacyjnym.
  - IV. Rygorystyczne testy są czasami stosowane w celu udowodnienia, że wszystkie awarie zostały znalezione.
- A. Stwierdzenia II i III są prawdziwe, a stwierdzenia I i IV są fałszywe.  
B. Stwierdzenia I i IV są prawdziwe, a stwierdzenia II i III są fałszywe.
- C. **Stwierdzenia I i III są prawdziwe, a stwierdzenia II i IV są fałszywe.**  
D. Stwierdzenia III i IV są prawdziwe, a stwierdzenia I i II są fałszywe.

#### Wyjaśnienie

- I. **Testy mogą być niezbędne w przypadku wymagań prawnych lub umownych** — jest to uważane za „testowanie akceptacyjne zgodnie z umową lub przepisami”.
- II. **Testowanie oprogramowania jest potrzebne głównie do poprawy jakości pracy programisty** — może to stanowić jeden z celów, ale głównym zadaniem testowania oprogramowania jest identyfikowanie defektów i awarii w produktach pracy.
- III. **Naprawa defektów pomaga zmniejszyć ryzyko wystąpienia problemów w środowisku operacyjnym** — to prawda, ponieważ — zgodnie z zasadą wczesnego testowania — znajdowanie defektów na wczesnym etapie cyklu życia oprogramowania zapobiega ich występowaniu na późniejszych etapach.
- IV. **Testowanie służy do udowodnienia, że wszystkie awarie zostały znalezione** — to stwierdzenie jest całkowicie błędne. Nie da się udowodnić, że wszystkie awarie zostały wykryte, ponieważ gruntowne testowanie jest niemożliwe.

**004.** Dlaczego ważne jest unikanie paradoksu pestycydów?

- A. Testowanie dynamiczne jest mniej skuteczne w znajdowaniu błędów.
  - B. Pestycydy wymieszane z testami statycznymi mogą sprawić, że błędy nie zostaną wykryte.
  - C. Testowanie nie powinno być zależne od kontekstu.
- D. **Ciągle przeprowadzanie tych samych testów zmniejszy szansę na znalezienie nowych defektów.**

#### Wyjaśnienie

Jedną z siedmiu zasad testowania oprogramowania jest „Uważaj na paradoks pestycydów”, która mówi:

Jeśli te same testy są powtarzane w kółko, w końcu mogą nie znajdować już żadnych nowych defektów.

W celu wykrycia nowych defektów konieczne może być zmodyfikowanie istniejących testów i danych testowych, a także napisanie nowych testów. (Testy nie są już skuteczne w wykrywaniu defektów, podobnie jak pestycydy po pewnym czasie



nie są skuteczne w usuwaniu szkodników). W niektórych przypadkach, takich jak automatyczne testy regresji, paradoks pestycydów ma korzystny skutek, którym jest stosunkowo niewielka liczba defektów regresji.

**005.** Które z poniższych stwierdzeń dotyczących testów gruntownych jest prawdziwe?

- A. Jest to forma testów przeciążeniowych.
- B. **Nie są one wykonalne (chyba że dotyczą bardzo prostego oprogramowania).**
- C. Zwykle są one wykonywane za pomocą automatyzacji testów.
- D. Zwykle odpowiada za nie programista podczas testów jednostkowych.

#### Wyjaśnienie

Jedną z siedmiu zasad testowania oprogramowania stanowi „Testowanie gruntowne jest niemożliwe”, zgodnie z którą:

Testowanie wszystkiego (każdej kombinacji danych wejściowych i warunków wstępnych) nie jest wykonalne, z wyjątkiem najprostszyc przypadków. Zamiast próbować testować gruntownie, należy użyć analizy ryzyka, technik testowych i priorytetów, aby skoncentrować wysiłki testowe.

**006.** Które z poniższych stwierdzeń najlepiej opisuje jedną z siedmiu kluczowych zasad testowania oprogramowania?

- A. Testy automatyczne lepiej niż testy manualne pomagają uniknąć testowania gruntownego.
- B. Testowanie gruntowne, przy wystarczającym wysiłku i wsparciu narzędziowym, jest wykonalne dla każdego oprogramowania.
- C. **Zwykle przetestowanie wszystkich kombinacji wejścia/wyjścia dla danego systemu oprogramowania nie jest możliwe.**
- D. Celem testowania jest wykazanie braku defektów.

#### Wyjaśnienie

- A. **Testowanie automatyczne pozwala uniknąć gruntownego testowania** — to niepoprawna odpowiedź, ponieważ testowanie gruntowne nie jest czymś, czego chcemy uniknąć.
- B. **Testowanie gruntowne, przy odpowiednim nakładzie pracy i wsparciu narzędzi, jest wykonalne** — nie można tego zrobić, ponieważ w przypadku dowolnego typu oprogramowania zawsze istnieje nieskończona liczba scenariuszy, których nie można pokryć nawet za pomocą narzędzi lub poprzez zwiększony wysiłek.
- C. **Testowanie gruntowne jest zwykle niemożliwe** — to prawda.
- D. **Celem testowania jest wykazanie braku defektów** — zgodnie z zasadą „Testowanie ujawnia usterki, ale nie może dowieść ich braku” testowanie pozwala zmniejszyć prawdopodobieństwo, że w oprogramowaniu pozostaną nieodkryte defekty. Ale nawet jeśli żadne defekty nie zostaną znalezione, to nie możemy dowieść, że rzeczywiście ich nie ma.

**007.** Zespół testowy konsekwentnie znajduje od 95% do 97% defektów w testowanym systemie. Kierownik testów wie, że jest to dobry wynik zarówno dla jego zespołu, jak i dla całej branży, jednak kierownictwo wyższego szczebla jest rozczarowane grupą testową i twierdzi, że pomija ona zbyt wiele błędów. Biorąc pod uwagę to, że użytkownicy są generalnie zadowoleni z systemu i że zaistniałe awarie miały niewielki wpływ na jego działanie — wskaż, która z poniższych zasad testowania najprawdopodobniej pomoże kierownikowi testów wyjaśnić menedżerom i kierownikom wyższego szczebla, dlaczego niektóre defekty mogą być pominięte.

→ **A. Testowanie gruntowne jest niemożliwe.**

- B. Paradoks pestycydów.
- C. Kumulowanie się defektów.
- D. Przekonanie o braku błędów jest błędem.

#### Wyjaśnienie

W naszej sytuacji mamy trzy punkty widzenia:

- 1) **kierownika testów**, który jest zadowolony z liczby wykrywanych defektów;
- 2) **kierownictwa**, które jest rozczarowane liczbą wykrywanych defektów;
- 3) **użytkowników**, którzy są zadowoleni z systemu.

W zależności od tych trzech punktów widzenia możemy stwierdzić, co następuje:

1. Testowany system charakteryzuje się wysokim poziomem jakości.
2. Kierownictwo nie rozumie jednej z siedmiu zasad testowania, związanej ze stwierdzeniem, że nie możemy znaleźć 100% defektów.

Dlatego wiemy, że zasada, która może pomóc kierownikowi testów wyjaśnić, dlaczego defekty nie zostały wykryte, to „Testowanie gruntowne jest niemożliwe”.

**008.** Programista pracuje nad bardzo złożonym kodem. Która z poniższych ogólnych zasad testowania może wpłynąć na jego pracę?

→ **A. Kumulowanie się defektów.**

- B. Paradoks pestycydów.
- C. Testowanie gruntowne jest niemożliwe.
- D. Przekonanie o braku błędów jest błędem.

#### Wyjaśnienie

W tym scenariuszu programista pracuje nad złożonym kodem, więc aby mógł sprawnie i szybko go testować, musi skupić swój wysiłek na modułach o największym ryzyku. Zasadą, która najlepiej pomoże określić te moduły, jest kumulowanie się defektów.

**009.** Według słownika ISTQB synonimem którego z poniższych słów jest „pluskwa”?

- A. Ulepszenie.
- B. Incydent.
- **C. Defekt.**
- D. Pomyłka.

#### Wyjaśnienie

„Pluskwa” jest inaczej nazywana „defektem”.

**010.** Jaką cechę musi posiadać tester, aby odnieść sukces w pracy z zespołem projektowym?

- **A. Umiejętność komunikowania się w konstruktywny sposób.**  
B. Autorytarny styl przywództwa.  
C. Ekstrawertyczną osobowość.  
D. Brak poprawnej odpowiedzi.

#### Wyjaśnienie

Niektórzy mogą postrzegać testowanie jako czynność destrukcyjną, mimo że w znacznym stopniu ma ono pozytywny wpływ na rozwój projektu i jakość produktu. Aby spróbować ograniczyć tego rodzaju opinie, informacje o defektach i awariach należy przekazywać w konstruktywny sposób. Dzięki temu można zmniejszyć napięcia pomiędzy testerami a analitykami, właścicielami produktów, projektantami i programistami. Dotyczy to zarówno testów statycznych, jak i dynamicznych.

Testerzy i kierownicy testów muszą posiadać wysokie umiejętności interpersonalne, aby móc skutecznie informować o defektach, niepowodzeniach, wynikach i postępkach testów oraz ryzykach, a także budować pozytywne relacje ze współpracownikami.

**011.** Który z poniższych elementów — z perspektywy testera — jest kluczem do skutecznego komunikowania się i utrzymywania pozytywnych relacji z programistami w przypadku sporu dotyczącego priorytetu defektu?

- A. Przekazanie sprawy do działu kadr i podkreślenie znaczenia wzajemnego szacunku.  
B. Rozmowa z kierownictwem wyższego szczebla, aby upewnić się, że wszyscy rozumieją, na jakiej podstawie tester ustalił priorytet błędu.  
C. Przekonanie dewelopera, aby wziął na siebie winę za pomyłkę.  
→ **D. Przypomnienie im o wspólnym celu tworzenia systemów dobrej jakości.**

#### Wyjaśnienie

Testerzy i kierownicy testów muszą posiadać dobre umiejętności interpersonalne, aby móc skutecznie informować o defektach, niepowodzeniach, wynikach i postępkach testów oraz ryzykach, a także budować pozytywne relacje ze współpracownikami. Sposoby dobrej komunikacji to między innymi:

- Zaczynj od współpracy, a nie od bitew. Przypomnij wszystkim o wspólnym celu, którym są systemy dobrej jakości.
- Podkreśl korzyści płynące z testowania. Na przykład z punktu widzenia autorów informacje o defektach mogą pomóc im ulepszyć produkty pracy i doskonalić umiejętności. Z perspektywy organizacji defekty znalezione i naprawione podczas testowania pozwolą zaoszczędzić czas i pieniądze oraz zmniejszą ryzyko związane z niepowodzeniem projektu.
- Przekazuj wyniki testów i inne ustalenia w sposób neutralny, skoncentrowany na faktach, bez krytykowania osoby, która stworzyła wadliwy element. Pisz obiektywne i oparte na faktach raporty o defektach oraz przeglądaj ustalenia.
- Spróbuj zrozumieć, jak czuje się druga osoba i dlaczego może negatywnie zareagować na przekazane informacje.

- 012.** Które z poniższych sytuacji prowadziłyby do problemów (lub konfliktów) w zespołach podczas przeglądów i testów?
- A. Testerzy i przeglądający nie są na tyle ciekawi, aby znajdować defekty.
  - B. Testerzy i przeglądający nie mają wystarczających kwalifikacji, aby znaleźć awarie i usterki.
  - C. **Testerzy i przeglądający komunikują defekty jako krytykę pod adresem osób, a nie oprogramowania.**
  - D. Testerzy i przeglądający oczekują, że defekty oprogramowania zostały już znalezione i naprawione przez programistów.

#### Wyjaśnienie

Element ludzkiej psychologii zwany efektem potwierdzenia może utrudniać akceptowanie informacji, które nie zgadzają się z aktualnie wyznawanymi przekonaniem. Na przykład: ponieważ programiści oczekują, że ich kod będzie poprawny, mogą mieć trudności z zaakceptowaniem tego, że kod jest błędny. Oprócz efektu potwierdzenia istnieją również inne błędy poznawcze, które mogą utrudniać ludziom zrozumienie lub zaakceptowanie informacji uzyskanych w wyniku testów. Co więcej, informacje te często zawierają złe wieści, a obwinianie osoby, która takie wieści przynosi, jest powszechną ludzką cechą.

Aby ograniczyć podobne reakcje, należy przekazywać informacje o defektach i awariach w sposób jak najbardziej konstruktywny, co pozwoli zmniejszyć napięcia między testerami a analitykami, właścicielami produktów, projektantami i programistami. Zasada ta ma zastosowanie zarówno w przypadku testowania statycznego, jak i dynamicznego.

- 013.** Które z poniższych zaleceń pomagają w dobrej komunikacji, a które ją utrudniają?
- I. Spróbuj zrozumieć, jak czuje się druga osoba.
  - II. Przekazuj subiektywne odczucia, skoncentrowane na konkretnych osobach.
  - III. Potwierdź, że druga osoba zrozumiała to, co powiedziałeś, i na odwrót.
  - IV. Podkreśl wspólny cel tworzenia lepszej jakości oprogramowania.
  - V. Każda dyskusja to bitwa do wygrania.
- A. Zalecenia I, II i III stanowią pomoc, a zalecenia IV i V stanowią utrudnienie.
  - B. Zalecenia III, IV i V stanowią pomoc, a zalecenia I i II stanowią utrudnienie.
  - C. **Zalecenia I, III i IV stanowią pomoc, a zalecenia II i V stanowią utrudnienie.**
  - D. Zalecenia II, III i IV stanowią pomoc, a zalecenia I i V stanowią utrudnienie.

#### Wyjaśnienie

- I. **Spróbuj zrozumieć, jak czuje się druga osoba** — według sylabusu sposoby dobrej komunikacji obejmują stwierdzenie: „Należy wczuć się w sytuację drugiej osoby i zrozumieć, dlaczego negatywnie reaguje ona na podane informacje”.
- II. **Przekazuj subiektywne odczucia, skoncentrowane na konkretnych osobach** — według sylabusu sposoby dobrej komunikacji obejmują stwierdzenie: „Informacje na temat rezultatów testów i inne wnioski należy przekazywać w sposób neutralny, koncentrując się na faktach i nie krytykując osoby, która stworzyła wadliwy element.

W związku z powyższym należy zadbać o to, by raporty o defektach i wnioski z przeglądu były obiektywne oraz znajdowały oparcie w faktach”.

- III. **Potwierdź, że druga osoba zrozumiała to, co powiedziałeś, i na odwrót** — według sylabusa sposoby dobrej komunikacji obejmują stwierdzenie: „Należy upewnić się, że rozmówca rozumie przekazywane informacje i *vice versa*”.
- IV. **Podkreśl wspólny cel tworzenia lepszej jakości oprogramowania** — według sylabusa sposoby dobrej komunikacji obejmują stwierdzenie: „Należy zacząć od współpracy, a nie od konfliktu. Wszystkich powinna łączyć świadomość wspólnego celu, jakim jest podnoszenie jakości systemów”.
- V. **Każda dyskusja to bitwa do wygrania** — według sylabusa sposoby dobrej komunikacji obejmują stwierdzenie: „Należy zacząć od współpracy, a nie od konfliktu. Wszystkich powinna łączyć świadomość wspólnego celu, jakim jest podnoszenie jakości systemów”.

**014.** Podczas których czynności w podstawowym procesie testowym następuje projektowanie i ustalanie priorytetów przypadków testowych?

- A. Podczas planowania testów.
- B. Podczas analizy testów.
- C. **Podczas projektowania testów.**
- D. Podczas ukończenia testów.

#### Wyjaśnienie

##### Projektowanie testów

Podczas projektowania testów warunki testowe są opracowywane w postaci przypadków testowych wysokiego poziomu, zestawów przypadków testowych wysokiego poziomu i innych testaliów. Zatem analiza testów odpowiada na pytanie „co testować?”, natomiast projektowanie testów — na pytanie „jak testować?”.

**015.** Podczas których czynności w podstawowym procesie testowym ma miejsce definiowanie i ustalanie priorytetów warunków testowych oraz uwzględnianie cech funkcjonalnych i nefunkcjonalnych?

- A. Podczas planowania testów.
- B. **Podczas analizy testów.**
- C. Podczas projektowania testów.
- D. Podczas ukończenia testów.

#### Wyjaśnienie

##### Analiza testów

Grupa czynności określana jako „analiza testów” polega na przeanalizowaniu podstawy testów w celu zidentyfikowania testowalnych cech i zdefiniowania związanych z nimi warunków testowych. Innymi słowy, analiza testów służy do ustalenia tego, co należy przetestować (w kategoriach mierzalnych kryteriów pokrycia).

Analiza testów obejmuje następujące główne czynności:

- dokonywanie analizy podstawy testów właściwej dla rozważanego poziomu testów;
- dokonywanie oceny testowalności podstawy testów i elementów testowych w celu zidentyfikowania często występujących typów defektów, które mogą powodować problemy z testowalnością;
- identyfikowanie cech i zbiorów cech, które mają zostać przetestowane;
- definiowanie warunków testowych w odniesieniu do poszczególnych cech oraz określanie ich priorytetów za pomocą analizy podstawy testów — z uwzględnieniem parametrów funkcjonalnych, niefunkcjonalnych i strukturalnych, innych czynników biznesowych i technicznych oraz poziomów ryzyka.

**016.** Podczas której czynności w podstawowym procesie testowym następuje sprawdzenie, czy wszystkie zgłoszenia defektów są zamknięte?

- A. Podczas planowania testów.
- B. Podczas analizy testów.
- C. Podczas projektowania testów.
- D. **Podczas ukończenia testów.**

#### Wyjaśnienie

##### Ukończenie testów

Ukończenie testów polega na zebraniu danych pochodzących z wykonanych czynności testowych w celu usystematyzowania i połączenia zdobytych doświadczeń, testaliów oraz innych istotnych informacji. Działania związane z ukończeniem testów wykonywane są w momencie osiągnięcia kamieni milowych projektu, takich jak: przekazanie systemu lub oprogramowania do eksploatacji, zakończenie realizacji (lub anulowanie) projektu, zakończenie iteracji projektu zwinnego, ukończenie testów danego poziomu bądź zakończenie prac nad wydaniem pielęgnacyjnym.

Główne czynności wykonywane w ramach ukończenia testów to:

- sprawdzenie, czy wszystkie raporty o defektach są zamknięte, oraz wprowadzenie do rejestru produktu żądań zmian lub pozycji dotyczących wszelkich defektów, które nie zostały rozwiązane do momentu zakończenia wykonywania testów;
- utworzenie sumarycznego raportu z testów, który zostanie przekazany interesariuszom;
- dla wersji końcowych: zarchiwizowanie środowiska testowego, danych testowych, infrastruktury testowej oraz innych testaliów, do ponownego wykorzystania w przyszłości;
- przekazanie testaliów zespołom odpowiedzialnym za pielęgnację, innym zespołom projektowym i/lub innym interesariuszom, którzy mogą odnieść korzyść z ich użycia;
- przeanalizowanie wniosków z ukończonych czynności testowych (ang. *lessons learned*) w celu ustalenia, jakie zmiany będą konieczne w przypadku przyszłych iteracji, wydań i projektów;
- wykorzystanie zebranych informacji do zwiększenia dojrzałości procesu testowego.

# PROGRAM PARTNERSKI

— GRUPY HELION —

- 
1. ZAREJESTRUJ SIĘ
  2. PREZENTUJ KSIĄŻKI
  3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>

GRUPA  
**Helion**

## Rozpocznij karierę testera oprogramowania!

IT przeżywa rozkwit, czego dowodem jest to, że w pierwszym półroczu 2022 roku 25 procent ogłoszeń o pracy było skierowanych właśnie do przedstawicieli tej branży. Na ten imponujący wynik składają się oferty dla specjalistów do spraw rozwoju oprogramowania (16 procent wszystkich ogłoszeń). Nic więc dziwnego, że na rynku nie brakuje książek przeznaczonych dla adeptów tej sztuki. Jedną z takich publikacji jest *Ścieżka testera. Certyfikat ISTQB w pytaniach i odpowiedziach*, która może stanowić podstawowe szkolenie na ścieżce certyfikacji dla wszystkich osób zajmujących się testowaniem oprogramowania.

Jednym ze sposobów, by rozwinąć warsztat testera, jest zdanie egzaminu i uzyskanie świadectwa ISTQB. Certyfikat ten, wydawany przez międzynarodową organizację International Software Testing Qualifications Board, jest wysoko ceniony przez pracodawców, ma dużą rozpoznawalność i stanowi dowód znajomości terminologii testerskiej.

Do zdobycia certyfikatu ISTQB na poziomie podstawowym z pewnością przyda się ta książka, mająca na celu przedstawić zagadnienia, które można napotkać na wspomnianym egzaminie. Zawiera ona 400 przykładowych pytań i odpowiedzi związanych z metodami testowania czy procesami przeglądu. Pytania dobrano tak, aby były jak najbardziej zbliżone do egzaminacyjnych; każdemu z nich towarzyszą cztery odpowiedzi, z których tylko jedna jest poprawna. W drugiej części publikacji natomiast znalazły się odpowiedzi na wszystkie pytania, podane wraz z uzasadnieniami. Zatem — do dzieła!

## Zdobądź wiedzę, certyfikat ISTQB i... dobrze płatną pracę

**Rafał Podraza** — QA Engineer, instruktor i twórca treści. Na co dzień zajmuje się usprawnianiem procesów związanych z testowaniem, dba o dostarczaną jakość oprogramowania, a także implementuje automatyczne skrypty testowe z wykorzystaniem języków JavaScript i TypeScript. Od listopada 2021 roku tworzy profil *Testowanie Oprogramowania* na platformie Udemy, gdzie publikuje kursy edukacyjne — część z nich uzyskała status best-sellerów. Jest twórcą bloga *testowanie-oprogramowania.pl* i aktywnym działaczem w społeczności testerskiej, regularnie tworzy treści związane z testowaniem i programowaniem na kanale YouTube *Technika Programowania*.

**Helion**



helion.pl



**HELION SA**  
ul. Kościuszki 1c  
44-100 Gliwice  
tel.: 32 230 98 63  
helion@helion.pl

**KOD KORZYŚCI**  
Sięgnij po więcej! ▶



ISBN 978-83-283-9802-3



9 788328 398023

Cena: 69,00 zł