

Egzamin przykładowy – odpowiedzi
zestaw C
(wersja 1.6)

wersja tłumaczenia PL 1.6.0

zgodny z sylabusem

Certyfikowany Tester
Poziom Podstawowy
wersja 4.0.1

International Software Testing Qualifications Board



Informacja o prawach autorskich

© International Software Testing Qualifications Board (zwana dalej „ISTQB”).

ISTQB® jest zarejestrowanym znakiem towarowym International Software Testing Qualifications Board.

Wszelkie prawa zastrzeżone. Autorzy niniejszym przenoszą prawa autorskie na ISTQB®. Autorzy (jako obecni właściciele praw autorskich) i ISTQB® (jako przyszły właściciel praw autorskich) uzgodnili następujące warunki użytkowania:

- Fragmenty niniejszego dokumentu mogą być kopiowane do użytku niekomercyjnego, pod warunkiem podania źródła. Każdy akredytowany dostawca szkoleń może wykorzystać niniejszy egzamin przykładowy w trakcie szkolenia, pod warunkiem podania autorów i ISTQB® jako źródła i właścicieli praw autorskich do egzaminu przykładowego oraz pod warunkiem, że wszelkie reklamy takiego szkolenia mogą zawierać wzmiankę o egzaminie przykładowym dopiero po otrzymaniu oficjalnej akredytacji materiałów szkoleniowych od uznanej przez ISTQB® rady krajowej.
- Każda osoba lub grupa osób może wykorzystać niniejszy egzamin przykładowy w artykułach i książkach, pod warunkiem podania autorów i ISTQB® jako źródła i właścicieli praw autorskich do egzaminu przykładowego.
- Wszelkie inne wykorzystanie niniejszego egzaminu przykładowego jest zabronione bez uprzedniej pisemnej zgody ISTQB®.
- Każda rada krajowa uznana przez ISTQB® może przetłumaczyć niniejszy egzamin przykładowy, pod warunkiem, że w przetłumaczonej wersji egzaminu przykładowego zamieści powyższą informację o prawach autorskich.

Prawa autorskie wersji polskiej: Copyright © Polish Quality Board (PQB).

Tłumaczenie z języka angielskiego: Adam Roman.

Przegląd tłumaczenia i korekta: Monika Petri-Starego.

Odpowiedzialność za dokument

Za niniejszy dokument odpowiada Grupa Robocza ISTQB® ds. Egzaminów.

Niniejszy dokument jest utrzymywany przez Główny Zespół ISTQB® składający się z Grupy Roboczej ds. Sylabusu oraz Grupy Roboczej ds. Egzaminów.

Podziękowania

Niniejszy dokument został opracowany przez zespół ISTQB®: Stuarta Reida i Adama Romana.

Zespół główny dziękuje zespołowi recenzentów Grupy Roboczej ds. Egzaminów, Grupie Roboczej Poziomu Podstawowego oraz radom krajowym za sugestie i uwagi.

Historia zmian

Wersja	Data	Uwagi
1.6	25.03.2025	Korekta dotycząca stosowania terminów ze słownika ISTQB®
1.5	4.07.2024	Zmiana pytania nr 20
1.4	29.05.2024	Korekta odpowiedzi nr 17
1.3	14.03.2024	Korekta odpowiedzi nr 20, 25
1.2	20.12.2023	Zmiana w celu dostosowania do wersji dokumentu z pytaniami
1.1	6.11.2023	Drobna korekta redakcyjna
1.0	16.10.2023	Pierwsza wersja

Historia zmian polskiej wersji dokumentu

Wersja	Data	Uwagi
1.6.0	14.03.2026	Publikacja polskiego tłumaczenia dokumentu

Spis treści

Informacja o prawach autorskich	2
Odpowiedzialność za dokument	2
Podziękowania	2
Historia zmian	3
Historia zmian polskiej wersji dokumentu	3
Wstęp	6
Cel niniejszego dokumentu	6
Struktura niniejszego dokumentu	6
Klucz odpowiedzi	7
Odpowiedzi	8
Pytanie 1 (FL-1.1.1, K1, 1 pkt)	8
Pytanie 2 (FL-1.1.2, K2, 1 pkt)	8
Pytanie 3 (FL-1.3.1, K2, 1 pkt)	9
Pytanie 4 (FL-1.4.1, K2, 1 pkt)	9
Pytanie 5 (FL-1.4.3, K2, 1 pkt)	10
Pytanie 6 (FL-1.4.5, K2, 1 pkt)	11
Pytanie 7 (FL-1.5.2, K1, 1 pkt)	11
Pytanie 8 (FL-1.5.3, K2, 1 pkt)	12
Pytanie 9 (FL-2.1.2, K1, 1 pkt)	12
Pytanie 10 (FL-2.1.3, K1, 1 pkt)	13
Pytanie 11 (FL-2.1.5, K2, 1 pkt)	13
Pytanie 12 (FL-2.1.6, K2, 1 pkt)	14
Pytanie 13 (FL-2.2.1, K2, 1 pkt)	14
Pytanie 14 (FL-2.2.3, K2, 1 pkt)	15
Pytanie 15 (FL-3.1.3, K2, 1 pkt)	15
Pytanie 16 (FL-3.2.1, K1, 1 pkt)	16
Pytanie 17 (FL-3.2.4, K2, 1 pkt)	16
Pytanie 18 (FL-3.2.5, K1, 1 pkt)	17
Pytanie 19 (FL-4.1.1, K2, 1 pkt)	18
Pytanie 20 (FL-4.2.1, K3, 1 pkt)	18
Pytanie 21 (FL-4.2.2, K3, 1 pkt)	18
Pytanie 22 (FL-4.2.3, K3, 1 pkt)	19
Pytanie 23 (FL-4.2.4, K3, 1 pkt)	19

Pytanie 24 (FL-4.3.2, K2, 1 pkt)	20
Pytanie 25 (FL-4.3.3, K2, 1 pkt)	20
Pytanie 26 (FL-4.4.1, K2, 1 pkt)	20
Pytanie 27 (FL-4.4.3, K2, 1 pkt)	21
Pytanie 28 (FL-4.5.2, K2, 1 pkt)	21
Pytanie 29 (FL-4.5.3, K3, 1 pkt)	22
Pytanie 30 (FL-5.1.1, K2, 1 pkt)	22
Pytanie 31 (FL-5.1.4, K3, 1 pkt)	23
Pytanie 32 (FL-5.1.5, K3, 1 pkt)	23
Pytanie 33 (FL-5.1.6, K1, 1 pkt)	23
Pytanie 34 (FL-5.1.7, K2, 1 pkt)	24
Pytanie 35 (FL-5.2.3, K2, 1 pkt)	24
Pytanie 36 (FL-5.3.2, K2, 1 pkt)	25
Pytanie 37 (FL-5.4.1, K2, 1 pkt)	25
Pytanie 38 (FL-5.5.1, K3, 1 pkt)	25
Pytanie 39 (FL-6.1.1, K2, 1 pkt)	26
Pytanie 40 (FL-6.2.1, K1, 1 pkt)	26

Wstęp

Cel niniejszego dokumentu

Przykładowe pytania i odpowiedzi oraz związane z nimi uzasadnienia w tym przykładowym egzaminie zostały opracowane przez zespół ekspertów merytorycznych i doświadczonych autorów pytań w celu:

- pomocy radom krajowym ISTQB® i komisjom egzaminacyjnym w tworzeniu pytań,
- dostarczenia organizatorom szkoleń i kandydatom do egzaminów przykładowych pytań egzaminacyjnych.

Pytania te nie mogą być wykorzystywane w niezmienionej formie w żadnym oficjalnym egzaminie.

Należy pamiętać, że rzeczywiste egzaminy mogą zawierać szeroki zakres pytań, a niniejszy przykładowy egzamin nie ma na celu przedstawienia przykładów wszystkich możliwych typów, stylów lub długości pytań. Ponadto ten przykładowy egzamin może być zarówno trudniejszy, jak i łatwiejszy od oficjalnego egzaminu.

Struktura niniejszego dokumentu

W niniejszym dokumencie można znaleźć:

- tabelę z kluczem odpowiedzi, zawierającą dla każdej poprawnej odpowiedzi:
 - poziom K, cel nauczania i wartość punktową,
- zestawy odpowiedzi, zawierające dla wszystkich pytań:
 - poprawną odpowiedź,
 - uzasadnienie dla każdej opcji odpowiedzi,
 - poziom K, cel nauczania i wartość punktową.

Pytania znajdują się w oddzielnym dokumencie.

Klucz odpowiedzi

Pytanie	Poprawna odpowiedź	Cel nauczania	Poziom K	Punkty
1	b	FL-1.1.1	K1	1
2	c	FL-1.1.2	K2	1
3	b	FL-1.3.1	K2	1
4	b, e	FL-1.4.1	K2	1
5	a	FL-1.4.3	K2	1
6	c	FL-1.4.5	K2	1
7	b	FL-1.5.2	K1	1
8	a	FL-1.5.3	K2	1
9	d	FL-2.1.2	K1	1
10	d	FL-2.1.3	K1	1
11	b	FL-2.1.5	K2	1
12	c	FL-2.1.6	K2	1
13	d	FL-2.2.1	K2	1
14	b	FL-2.2.3	K2	1
15	d	FL-3.1.3	K2	1
16	a	FL-3.2.1	K1	1
17	b	FL-3.2.4	K2	1
18	b	FL-3.2.5	K1	1
19	c	FL-4.1.1	K2	1
20	a	FL-4.2.1	K3	1
21	d	FL-4.2.2	K3	1
22	d	FL-4.2.3	K3	1
23	a	FL-4.2.4	K3	1
24	c	FL-4.3.2	K2	1
25	a	FL-4.3.3	K2	1
26	b	FL-4.4.1	K2	1
27	d	FL-4.4.3	K2	1
28	b	FL-4.5.2	K2	1
29	d	FL-4.5.3	K3	1
30	a	FL-5.1.1	K2	1
31	c	FL-5.1.4	K3	1
32	a	FL-5.1.5	K3	1
33	b	FL-5.1.6	K1	1
34	d	FL-5.1.7	K2	1
35	c	FL-5.2.3	K2	1
36	b	FL-5.3.2	K2	1
37	d	FL-5.4.1	K2	1
38	b	FL-5.5.1	K3	1
39	d	FL-6.1.1	K2	1
40	d	FL-6.2.1	K1	1

Odpowiedzi

Pytanie 1 (FL-1.1.1, K1, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Sprawdzanie zgodności z udokumentowanymi wymaganiami jest niepoprawne, ponieważ walidacja dotyczy spełnienia wymagań i oczekiwań użytkowników, podczas gdy weryfikacja dotyczy spełnienia wyspecyfikowanych wymagań, więc byłoby to poprawne, gdybyśmy zastąpili „walidację” terminem „weryfikacja”.
- b) Odpowiedź poprawna. Wywoływanie awarii i identyfikowanie defektów jest prawdopodobnie najczęstszym celem testowania dynamicznego.
- c) Odpowiedź niepoprawna. Wywoływanie błędów i identyfikowanie podstawowych przyczyn jest niepoprawne, ponieważ testerzy nie wywołują błędów, lecz awarie. Błędy są zazwyczaj popełniane przez programistów (i nie można ich tak naprawdę wywołać) i powodują defekty, które testerzy próbują zidentyfikować bezpośrednio poprzez testowanie statyczne lub pośrednio poprzez awarie w testowaniu dynamicznym. Identyfikowanie podstawowych przyczyn jest przydatne, ale stanowi część debugowania, które jest odrębną czynnością od testowania.
- d) Odpowiedź niepoprawna. Sprawdzanie, czy przedmiot testów spełnia oczekiwania użytkownika, jest niepoprawne, ponieważ weryfikacja dotyczy sprawdzania wyspecyfikowanych (udokumentowanych) wymagań, podczas gdy walidacja dotyczy spełnienia wymagań i oczekiwań użytkownika, więc byłoby to poprawne, gdybyśmy zastąpili „weryfikację” terminem „walidacja”.

Pytanie 2 (FL-1.1.2, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

- a) Odpowiedź niepoprawna. Testowanie dynamiczne wywołuje awarie (na podstawie których można zlokalizować i usunąć defekty). Jednak debugowanie polega na lokalizowaniu defektów i ich usuwaniu. Dlatego debugowanie nie usuwa awarii.
- b) Odpowiedź niepoprawna. Zarówno testowanie, jak i debugowanie przyczyniają się do poprawy jakości przedmiotu testów, dlatego też należy je traktować jako działania pozytywne. Debugowanie jest ogólnie uważane za działanie pozytywne, ponieważ polega na naprawianiu (usuwaniu) czegoś. Testowanie dynamiczne polega na celowym wywołaniu awarii w przedmiocie testów, dlatego niektórzy uważają je za działanie negatywne, ale jest to bardzo wąskie spojrzenie (i nie jest ono typowe dla testerów). Możliwe są zarówno pozytywne, jak i negatywne przypadki testowe. Pozytywne przypadki testowe sprawdzają, czy przedmiot testów prawidłowo wykonuje to, co ma robić, podczas gdy negatywne sprawdzają, czy przedmiot testów nie robi tego, czego nie powinien robić.
- c) Odpowiedź poprawna. Testowanie pozwala stwierdzić istnienie defektów albo bezpośrednio poprzez obserwację defektu podczas przeglądów (lub za pomocą narzędzia do analizy statycznej), albo pośrednio poprzez wywołanie awarii podczas testowania dynamicznego. Debugowanie jest działaniem odrębnym od testowania (zwykle

wykonywanym przez programistów) i polega na lokalizowaniu defektów (tylko w przypadku testowania dynamicznego) i ich usuwaniu.

- d) Odpowiedź niepoprawna. Przyczyną defektów są zazwyczaj błędy ludzkie. Testowanie wykrywa defekty bezpośrednio poprzez testowanie statyczne lub pośrednio poprzez wywołanie awarii w testowaniu dynamicznym, a debugowanie usuwa defekty. Testowanie nie wykrywa więc przyczyn defektów, a debugowanie nie usuwa przyczyn defektów.

Pytanie 3 (FL-1.3.1, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

„Przekonanie o braku defektów jest błędem” dotyczy przekonania, że samo zapewnienie poprawności zgodnie z wyspecyfikowanymi wymaganiami (tj. weryfikacja braku defektów implementacyjnych) nie gwarantuje zadowolenia użytkowników z systemu. Aby temu zaradzić, konieczna jest również walidacja, czyli sprawdzenie, czy system spełnia potrzeby i oczekiwania użytkowników, realizuje cele biznesowe i przewyższa konkurencyjne systemy. Zatem:

- a) Odpowiedź niepoprawna. Zasada „testowanie ujawnia defekty, ale nie może dowieść ich braku” wyjaśnia, że chociaż testowanie może wykryć istnienie defektów w przedmiocie testów, nie jest możliwe wykazanie, że nie ma w nim żadnych defektów, a zatem zagwarantowanie jego poprawności. Dlatego wyjaśnienie, że testowanie nie może wykazać braku defektów, częściowo odnosi się do tej zasady, a nie do błędnego przekonania o braku defektów.
- b) Odpowiedź poprawna. Wspierając użytkownika końcowego w przeprowadzaniu testów akceptacyjnych, powinno być możliwe sprawdzenie, czy system spełnia potrzeby i oczekiwania użytkowników.
- c) Odpowiedź niepoprawna. Nie można zapewnić, że w dostarczonym systemie nie pozostały żadne defekty implementacyjne, ponieważ zasada „testowanie ujawnia defekty, ale nie może dowieść ich braku” wyjaśnia, że chociaż testowanie może wykryć obecność defektów w przedmiocie testów, nie jest możliwe wykazanie, że nie ma żadnych defektów, a zatem zagwarantowanie jego poprawności.
- d) Odpowiedź niepoprawna. Modyfikowanie testów, które nie powodują żadnych awarii w celu zapewnienia, że pozostało niewiele defektów, jest jednym ze sposobów rozwiązania problemu związanego z zasadą „testy ulegają zużyciu”. Zasada ta dotyczy przekonania, że powtarzanie identycznych testów na niezmiennym kodzie prawdopodobnie nie pozwoli wykryć nowych defektów, a zatem modyfikowanie testów może być niezbędne. Nie potwierdzi to jednak, że system spełnia potrzeby i oczekiwania użytkowników.

Pytanie 4 (FL-1.4.1, K2, 1 pkt)

Poprawna odpowiedź: b, e

Uzasadnienie

Sylabus opisuje analizę testów w następujący sposób: „Analiza testów obejmuje analizę podstawy testów w celu zidentyfikowania cech, które można przetestować oraz zdefiniowanie i spriorytetyzowanie powiązanych z nimi warunków testowych, biorąc pod uwagę istniejące ryzyka. Analizuje się również podstawę testów i przedmiot testów w celu oceny ich testowalności

oraz zidentyfikowania ewentualnych defektów. Analiza testów jest często wspierana przez zastosowanie technik testowania.” Wynika stąd, że techniki testowania są często stosowane w analizie testów i projektowaniu testów. Analiza wartości brzegowych i podział na klasy równoważności są technikami testowania.

- a) Odpowiedź niepoprawna. Implementacja testów prawdopodobnie nie będzie wymagać zastosowania technik testowania, ponieważ dotyczy ona głównie zestawiania przypadków testowych w procedury testowe, podczas gdy techniki testowania służą do tworzenia przypadków testowych.
- b) Odpowiedź poprawna. Projektowanie testów prawdopodobnie wiąże się z wykorzystaniem technik testowania w celu tworzenia przypadków testowych na podstawie warunków testowych i elementów pokrycia.
- c) Odpowiedź niepoprawna. Wykonywanie testów prawdopodobnie nie wymaga stosowania technik testowania, ponieważ dotyczy głównie wykonywania procedur testowych (a więc przypadków testowych), podczas gdy techniki testowania służą do tworzenia przypadków testowych.
- d) Odpowiedź niepoprawna. Monitorowanie testów prawdopodobnie nie wymaga stosowania technik testowania. Monitorowanie testów dotyczy głównie bieżącej kontroli mającej na celu zapewnienie realizacji planu, podczas gdy techniki testowania służą do tworzenia przypadków testowych.
- e) Odpowiedź poprawna. Analiza testów prawdopodobnie będzie wymagała zastosowania technik testowania w celu zidentyfikowania warunków testowych.

Pytanie 5 (FL-1.4.3, K2, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

Biorąc pod uwagę każdą z wymienionych czynności testowych i związane z nimi testalia:

- A. Analiza testów – spriorytetyzowane warunki testowe (4) (np. kryteria akceptacji) oraz raporty o defektach dotyczące defektów wykrytych w podstawie testów.
- B. Projektowanie testów – spriorytetyzowane przypadki testowe, karty opisu testu, elementy pokrycia (1), wymagania dotyczące danych testowych i wymagania dotyczące środowiska testowego.
- C. Implementacja testów – procedury testowe, automatyczne skrypty testowe, zestawy testowe, dane testowe, harmonogram wykonywania testów (3) oraz elementy środowiska testowego, takie jak zaślepki, sterowniki, symulatory i wirtualizacje usług.
- D. Ukończenie testów – summaryczny raport z testów, udokumentowane wnioski, działania korygujące i żądania zmian (2) (jako elementy backlogu produktu)

Zatem:

- a) Odpowiedź poprawna. Prawidłowe dopasowanie to: 1B, 2D, 3C, 4A.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź niepoprawna.
- d) Odpowiedź niepoprawna.

Pytanie 6 (FL-1.4.5, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

- a) Odpowiedź niepoprawna. Chociaż prawdą jest, że w zwinnym wytwarzaniu oprogramowania niektóre zadania związane z zarządzaniem testami mogą być wykonywane przez sam zespół zwinny, rola związana z testowaniem nie jest przede wszystkim obowiązkiem jednej osoby spoza zespołu. Testowanie jest raczej wykonywane przez różnych członków zespołu zgodnie z podejściem „cały zespół”.
- b) Odpowiedź niepoprawna. Rola związana z zarządzaniem testami obejmuje przede wszystkim działania związane z planowaniem testów, monitorowaniem testów i nadzorem nad testami oraz ukończeniem testów. Tak więc, chociaż stwierdzenie to jest częściowo poprawne, błędne jest twierdzenie, że rola związana z testowaniem polega przede wszystkim na monitorowaniu testów i nadzorze nad testami.
- c) Odpowiedź poprawna. W zwinnym wytwarzaniu oprogramowania niektóre zadania związane z zarządzaniem testami mogą być realizowane przez sam zespół zwinny. Jednak w przypadku czynności testowych obejmujących wiele zespołów w organizacji zadania te mogą być realizowane przez kierowników testów spoza zespołu programistów.
- d) Odpowiedź niepoprawna. Rola związana z zarządzaniem testami obejmuje przede wszystkim działania związane z planowaniem testów, monitorowaniem testów i nadzorem nad testami oraz ukończeniem testów, podczas gdy rola związana z testowaniem odpowiada przede wszystkim za techniczne i inżynierskie aspekty testowania, takie jak analiza testów, projektowanie testów, implementacja testów i wykonywanie testów. Dlatego też rola związana z zarządzaniem testami nie odpowiada zazwyczaj za analizę testów i projektowanie testów, chociaż słuszne jest stwierdzenie, że rola związana z testowaniem odpowiada przede wszystkim za implementację i wykonywanie testów.

Pytanie 7 (FL-1.5.2, K1, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. W podejściu „cały zespół” testerzy odgrywają kluczową rolę, dzieląc się swoją wiedzą specjalistyczną z zespołem i kierując rozwojem produktu. Współpracują z innymi członkami zespołu, aby osiągnąć pożądaną jakość, oraz z przedstawicielami jednostek biznesowych, aby stworzyć testy akceptacyjne. Testerzy współpracują również z programistami w celu określenia optymalnej strategii testów i podejścia do automatyzacji.
- b) Odpowiedź poprawna. Wykorzystując w najbardziej efektywny sposób różnorodne umiejętności każdego członka zespołu, podejście „cały zespół” sprzyja doskonałej dynamice zespołu, promuje efektywną komunikację i współpracę oraz generuje efekt synergii, który przynosi korzyści całemu projektowi.
- c) Odpowiedź niepoprawna. Podejście „cały zespół” pozwala każdemu członkowi zespołu posiadającemu wymagane umiejętności i wiedzę podjąć się wykonania dowolnego zadania, dlatego też obecność specjalistów w zespole nie stanowi zalety tego podejścia.
- d) Odpowiedź niepoprawna. Nie ma konkretnych wytycznych dotyczących optymalnej wielkości zespołów stosujących podejście „cały zespół” i nie ma dowodów na to, że większe zespoły są lepsze.

Pytanie 8 (FL-1.5.3, K2, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

- a) Odpowiedź poprawna. Główną zaletą niezależności testowania jest to, że testerzy mają większe szanse na wykrycie różnych rodzajów defektów niż programiści, ze względu na swoje zróżnicowane doświadczenie, techniczny punkt widzenia i potencjalne uprzedzenia, w tym uprzedzenia poznawcze. Jednak główną wadą niezależności testowania jest to, że testerzy mogą zostać odizolowani od zespołu programistów, co prowadzi do problemów z komunikacją, braku współpracy i potencjalnie wrogich relacji, w których testerzy są obwiniani za opóźnienia i stwarzanie wąskich gardel w procesie wydawania nowych wersji.
- b) Odpowiedź niepoprawna. Znajomość kodu przez programistę nie oznacza, że rzadko znajduje on w nim defekty, ale raczej, że potrafi skutecznie znaleźć wiele defektów we własnym kodzie. Ponadto, zamiast *wspólnego* doświadczenia programistów i testerów, jako powód, dla którego testerzy i programiści znajdują różne rodzaje defektów, podaje się zazwyczaj *odmienne* doświadczenie programistów i testerów.
- c) Odpowiedź niepoprawna. Testowanie może być przeprowadzane na różnych poziomach niezależności, od całkowitego braku niezależności do bardzo wysokiej niezależności (np. testerzy spoza organizacji). W większości projektów wykorzystuje się wiele poziomów niezależności, przy czym programiści przeprowadzają testowanie modułowe i integracyjne modułów, zespół testujący przeprowadza testowanie systemowe i integracyjne systemów, a przedstawiciele jednostek biznesowych przeprowadzają testowanie akceptacyjne. Testerzy mogą więc należeć do zespołu programistów i nie muszą pochodzić spoza organizacji. Znajomość dziedziny biznesowej aplikacji będzie się zmieniać od przypadku do przypadku i nie zależy od poziomu niezależności.
- d) Odpowiedź niepoprawna. Testowanie może być przeprowadzane na różnych poziomach niezależności, od braku niezależności do bardzo wysokiej niezależności (np. testerzy spoza organizacji), przy czym testerzy spoza zespołu programistów są zazwyczaj bardziej niezależni niż testerzy z zespołu. Istnieje jednak więcej powodów, aby sądzić, że testerzy spoza zespołu są prawdopodobnie bardziej odizolowani od programistów i dlatego są bardziej narażeni na obwinianie ich za opóźnienia w wydaniu produktu.

Pytanie 9 (FL-2.1.2, K1, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Kontrola jakości dotyczy wszystkich czynności związanych z wytwarzaniem, co oznacza, że każdej czynności związanej z wytwarzaniem oprogramowania odpowiada odpowiednia czynność testowa. Jednak w tym przypadku próbujemy zrównać poziomy testów z poziomami wytwarzania i chociaż wiemy, co oznacza termin „poziomy testów”, nie ma powszechnego konsensusu w rozumieniu terminu „poziom wytwarzania”.
- b) Odpowiedź niepoprawna. Każda czynność związana z wytwarzaniem oprogramowania ma odpowiadającą jej czynność testową, jednak cele testów są zupełnie inne. Na przykład celem testu może być zapewnienie, że przedmiot testów spełnia wymagania umowy,

zgodnie z którym przed dostawą należy przeprowadzić określony typ testów. W tym przypadku nie ma powodu, aby istniał odpowiadający mu cel wytwarzania.

- c) Odpowiedź niepoprawna. Kontrola jakości dotyczy wszystkich czynności związanych z wytwarzaniem, co oznacza, że każdej czynności związanej z wytwarzaniem oprogramowania odpowiada czynność testowa. Jednak ta sama symetria nie dotyczy testowania i działań użytkowników. Na przykład w przypadku niektórych systemów trudno jest nawet zidentyfikować użytkowników końcowych. Ponadto niektóre czynności testowe koncentrują się na programistach (np. testowanie łatwości utrzymywania), co nie ma żadnego związku z działaniami użytkowników.
- d) Odpowiedź poprawna. Kontrola jakości dotyczy wszystkich czynności związanych z wytwarzaniem, co oznacza, że każdej czynności związanej z wytwarzaniem oprogramowania odpowiada czynność testowa.

Pytanie 10 (FL-2.1.3, K1, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Wytwarzanie sterowane testami modułowymi nie jest poprawnym przykładem podejścia „najpierw test”.
- b) Odpowiedź niepoprawna. Wytwarzanie sterowane testami integracyjnymi nie jest poprawnym przykładem podejścia „najpierw test”.
- c) Wytwarzanie sterowane testami systemowymi nie jest poprawnym przykładem podejścia „najpierw test”.
- d) Odpowiedź poprawna. Wytwarzanie sterowane testami akceptacyjnymi (ATDD) jest dobrze znanym przykładem podejścia „najpierw test”.

Pytanie 11 (FL-2.1.5, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Praktyki związane z „przesunięciem w lewo” mają na celu wdrożenie większej liczby czynności testowych we wczesnych fazach cyklu życia oprogramowania, przedstawiając ten cykl jako proces przebiegający od lewej do prawej strony. Nie ma czegoś takiego jak „przesunięcie w lewo” procesu testowego.
- b) Odpowiedź poprawna. „Przesunięcie w lewo” podkreśla znaczenie rozpoczęcia testowania na wcześniejszym etapie cyklu życia oprogramowania. Wdrożenie testowania w podejściu „przesunięcie w lewo” wymaga dodatkowych szkoleń oraz zwiększonego wysiłku i kosztów na wczesnych etapach cyklu życia oprogramowania, jednak ogólne oszczędności powinny być wyższe niż koszty.
- c) Odpowiedź niepoprawna. Chociaż automatyczne testy modułowe i testy integracyjne modułów jako testy regresji są generalnie cenne, tworzenie tych testów należy zazwyczaj do obowiązków programistów, a jeśli stosuje się podejście ciągłej integracji/ciągłego dostarczania (CI/CD), testy te zostaną przestane wraz z kodem. W niektórych sytuacjach tester może zautomatyzować testy regresji, a czasem nawet testy modułowe i testy integracyjne modułów, jednak nie jest to częścią podejścia „przesunięcie w lewo”, które przenosi testowanie na wcześniejszy etap cyklu życia oprogramowania.
- d) Odpowiedź niepoprawna. Szkolenie testerów w zakresie wykonywania zadań na wczesnym etapie cyklu życia oprogramowania wspierałoby podejście „przesunięcie

w lewo”, podkreślając znaczenie rozpoczęcia testowania na wcześniejszym etapie cyklu życia. Jednak automatyzacja większej liczby czynności testowych, które mają być wykonywane na późniejszym etapie cyklu życia, nie jest częścią tego podejścia.

Pytanie 12 (FL-2.1.6, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

- a) Odpowiedź niepoprawna. Jednym z celów retrospektyw jest identyfikacja potencjalnych udoskonaleń procesów, które po wdrożeniu powinny skutkować wyższą jakością przyszłych wyników procesu rozwoju (przedmiotów testów). Jest to więc prawdopodobne w wyniku retrospektywy.
- b) Odpowiedź niepoprawna. Korzyścią płynącą z retrospektyw dla testowania jest zwiększona wydajność testów dzięki udoskonaleniom procesów. Jest to więc prawdopodobne w wyniku retrospektywy.
- c) Odpowiedź poprawna. Uczestnikami retrospektyw są zazwyczaj testerzy, programiści, architekci, właściciele produktu i analitycy biznesowi, ale użytkownicy końcowi rzadko są zapraszani lub uczestniczą w tych spotkaniach – i jest również mało prawdopodobne, aby otrzymywali jakiegokolwiek raporty z tych spotkań. Jest więc bardzo mało prawdopodobne, aby dzięki retrospektywom dowiedzieli się więcej o procesach rozwoju i testowania i lepiej je zrozumieli.
- d) Odpowiedź niepoprawna. Korzyścią płynącą z retrospektyw dla testowania jest poprawa jakości testaliów (w tym automatycznych skryptów testowych) poprzez wspólne przeglądy z programistami. Jest to więc prawdopodobne w wyniku retrospektywy.

Pytanie 13 (FL-2.2.1, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Testowanie modułowe (zwane również testowaniem jednostkowym) polega na testowaniu poszczególnych modułów w izolacji i polega głównie na weryfikacji zgodności ze specyfikacją, a nie na walidacji pod kątem potrzeb użytkowników. Jednak testy te nie są zazwyczaj wykonywane przez testerów, lecz programistów, którzy zazwyczaj przeprowadzają je w swoim środowisku programistycznym.
- b) Odpowiedź niepoprawna. Testowanie integracyjne modułów polega na testowaniu interfejsów i interakcji między modułami oraz na weryfikacji zgodności ze specyfikacją, a nie na walidacji pod kątem potrzeb użytkowników. Jednak testy te nie są zazwyczaj przeprowadzane przez testerów, lecz programistów, którzy zazwyczaj przeprowadzają je w swoim środowisku programistycznym.
- c) Odpowiedź niepoprawna. Testowanie integracyjne systemów obejmuje badanie interfejsów z innymi systemami i usługami zewnętrznymi i polega głównie na weryfikacji zgodności ze specyfikacją, a nie na walidacji zgodności z potrzebami użytkowników. Tego typu testy są również najczęściej przeprowadzane przez testerów.
- d) Odpowiedź poprawna. Testowanie akceptacyjne koncentruje się na sprawdzeniu, czy system spełnia potrzeby biznesowe użytkownika i jest gotowy do wdrożenia. Idealnie byłoby, gdyby testy te były przeprowadzane przez użytkowników końcowych.

Pytanie 14 (FL-2.2.3, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Konieczne jest przeprowadzenie testowania potwierdzającego, aby sprawdzić, czy aktualizacje doprowadziły do prawidłowej implementacji, jednak rozsądne byłoby przeprowadzenie następnie testowania regresji w celu upewnienia się, że nie wprowadzono żadnych defektów ani nie wykryto ich w niezmiennych obszarach systemu.
- b) Odpowiedź poprawna. Testowanie potwierdzające sprawdza, czy aktualizacje doprowadziły do prawidłowej implementacji, a następnie przeprowadzone testowanie regresji sprawdza, czy nie wprowadzono żadnych defektów ani nie wykryto żadnych defektów w niezmiennych obszarach systemu.
- c) Odpowiedź niepoprawna. Testowanie regresji powinno być wykorzystane w celu zapewnienia, że podczas aktualizacji nie wprowadzono ani nie wykryto żadnych defektów w niezmiennych obszarach systemu, jednak konieczne jest również przeprowadzenie testowania potwierdzającego, które sprawdza, czy aktualizacje doprowadziły do prawidłowej implementacji.
- d) Odpowiedź niepoprawna. Testowanie potwierdzające sprawdza, czy aktualizacje doprowadziły do prawidłowej implementacji, a testowanie regresji służy do upewnienia się, że w niezmiennych obszarach systemu nie wprowadzono ani nie wykryto żadnych defektów. Jednak w przypadku ich przeprowadzenia (tj. gdy aktualizacja wymaga przetestowania) testy potwierdzające poprzedzają testy regresji.

Pytanie 15 (FL-3.1.3, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- i. Dwie różne części specyfikacji projektowej są ze sobą niezgodne ze względu na złożoność projektu – jest to przykład defektu specyfikacji, która obejmuje niespójności, niejasności, sprzeczności, pominięcia, nieścisłości i powtórzenia, które najłatwiej wykryć za pomocą testów statycznych.
- ii. Czas odpowiedzi jest zbyt długi, co powoduje utratę cierpliwości użytkowników – jest to przykład defektu czasu odpowiedzi, który można wykryć w praktyce jedynie poprzez uruchomienie programu i pomiar czasu odpowiedzi, co najłatwiej wykryć za pomocą testów dynamicznych.
- iii. Ścieżka w kodzie jest nieosiągalna podczas wykonywania – jest to przykład defektu kodowania, który obejmuje zmienne o nieokreślonych wartościach, zmienne niezadeklarowane, zduplikowany lub nieosiągalny kod oraz nadmierną złożoność kodu, które najłatwiej wykryć za pomocą testów statycznych.
- iv. Zmienna została zadeklarowana, ale nigdy nie została użyta w programie – jest to przykład defektu kodowania, który obejmuje zmienne o nieokreślonych wartościach, zmienne niezadeklarowane, zduplikowany lub nieosiągalny kod oraz nadmierną złożoność kodu, które najłatwiej wykryć za pomocą testów statycznych.
- v. Ilość pamięci potrzebna programowi do wygenerowania raportu jest zbyt duża – jest to przykład defektu wydajności, który można wykryć w praktyce jedynie poprzez uruchomienie programu i pomiar wykorzystanej pamięci, co najłatwiej wykryć za pomocą testów dynamicznych.

Zatem:

- a) Odpowiedź niepoprawna.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź niepoprawna.
- d) Odpowiedź poprawna.

Pytanie 16 (FL-3.2.1, K1, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

- a) Odpowiedź poprawna. Uzyskiwanie wczesnych i częstych informacji zwrotnych od interesariuszy może być bardzo korzystne. Ułatwia wczesną komunikację potencjalnych problemów związanych z jakością, może zapobiec nieporozumieniom dotyczącym wymagań i zapewnia, że wszelkie zmiany w wymaganiach interesariuszy są lepiej zrozumiałe i wdrażane szybciej.
- b) Odpowiedź niepoprawna. Informacje zwrotne pochodzą od interesariuszy, a przekazywanie przez nich informacji zwrotnych raczej nie poprawi zrozumienia ich własnych wymagań.
- c) Odpowiedź niepoprawna. Uzyskiwanie wczesnych i częstych informacji zwrotnych od interesariuszy może być bardzo korzystne. Ułatwia wczesną komunikację potencjalnych problemów związanych z jakością, może zapobiec nieporozumieniom dotyczącym wymagań i zapewnia, że wszelkie zmiany w wymaganiach interesariuszy są lepiej zrozumiałe i wdrażane szybciej. Jednak fakt, że zmiany w wymaganiach mogą być lepiej zrozumiałe i szybciej wdrażane, nie oznacza, że zachęca się do nieograniczonych zmian w wymaganiach.
- d) Odpowiedź niepoprawna. Informacje zwrotne pochodzą od interesariuszy i nie obejmują komunikacji z nimi. Komunikacja z użytkownikami końcowymi może obejmować poinformowanie ich o tym, które wymagania nie zostaną wdrożone przed wydaniem produktu, ale w idealnym przypadku nie powinno to mieć miejsca.

Pytanie 17 (FL-3.2.4, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

Biorąc pod uwagę każdy z wymienionych typów przeglądów:

1. Przegląd techniczny – ten typ przeglądu jest przeprowadzany przez przeglądających posiadających kwalifikacje techniczne i prowadzony przez moderatora. Jego celem jest osiągnięcie konsensusu i podjęcie decyzji dotyczących problemów technicznych, a także ocena jakości i budowanie zaufania do produktu pracy, generowanie nowych pomysłów, motywowanie i umożliwianie autorom doskonalenia się oraz wykrywanie anomalii.
2. Przegląd nieformalny – głównym celem jest wykrycie anomalii. Proces nie jest zdefiniowany i nie wymaga formalnego, udokumentowanego wyniku.
3. Inspekcja – jest to najbardziej formalny typ przeglądu, który przebiega zgodnie z kompletnym procesem przeglądu. Głównym celem jest wykrycie jak największej liczby anomalii, a inne cele obejmują ocenę jakości i budowanie zaufania do produktu pracy, motywowanie i umożliwianie autorom doskonalenia się oraz gromadzenie

metryk, które można wykorzystać do usprawnienia cyklu wytwarzania oprogramowania, w tym procesu samej inspekcji. Autor nie może pełnić funkcji lidera przeglądu ani protokolanta.

4. Przejrzenie – ten typ przeglądu, prowadzony przez autora, służy różnym celom, takim jak ocena jakości i budowanie zaufania do produktu pracy, edukacja przeglądających, osiągnięcie konsensusu, generowanie nowych pomysłów, motywowanie i umożliwianie autorom doskonalenia się oraz wykrywanie anomalii. Przeglądający mogą przeprowadzić indywidualny przegląd przed przejrzaniem (np. w ramach spotkania przeglądowego), ale nie jest to obowiązkowe.
- A. Obejmuje takie cele, jak osiągnięcie konsensusu, generowanie nowych pomysłów i motywowanie autorów do doskonalenia się.
B. Obejmuje takie cele, jak edukacja przeglądających, osiągnięcie konsensusu, generowanie nowych pomysłów i wykrywanie anomalii.
C. Głównym celem jest wykrywanie anomalii i wymaga gromadzenia metryk w celu wsparcia usprawniania procesów.
D. Głównym celem jest wykrywanie anomalii i nie generuje żadnych formalnych, udokumentowanych wyników.

W związku z tym:

- a) Odpowiedź niepoprawna.
b) Odpowiedź poprawna.
c) Odpowiedź niepoprawna.
d) Odpowiedź niepoprawna.

Pytanie 18 (FL-3.2.5, K1, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Aby zapewnić pomyślny przebieg przeglądów, ważne jest zapewnienie wsparcia kierownictwa dla procesu przeglądu, jednak nie oznacza to, że powinno ono uczestniczyć w przeglądach jako przeglądający.
- b) Odpowiedź poprawna. Aby zapewnić pomyślny przebieg przeglądów, ważne jest podzielenie produktu pracy na części, które są na tyle małe, że można je przejrzeć w rozsądnym czasie, aby przeglądający nie tracili koncentracji podczas indywidualnych przeglądów lub spotkań przeglądowych.
- c) Odpowiedź niepoprawna. Aby zapewnić pomyślny przebieg przeglądów, ważne jest jasne zdefiniowanie celów i mierzalnych kryteriów wyjścia, bez oceny uczestników.
- d) Odpowiedź niepoprawna. Aby zapewnić pomyślny przebieg przeglądów, należy podzielić przegląd na mniejsze części, aby przeglądający nie tracili koncentracji podczas indywidualnych przeglądów lub spotkań przeglądowych. Nie należy więc planować przeglądu jednego dokumentu podczas jednego przeglądu, jeśli jest on np. bardzo duży.

Pytanie 19 (FL-4.1.1, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

- a) Odpowiedź niepoprawna. W większości przypadków zarówno czarnoskrzynkowe techniki testowania, jak i techniki testowania oparte na doświadczeniu mogą być stosowane do tych samych przedmiotów testów.
- b) Odpowiedź niepoprawna. Zarówno czarnoskrzynkowe techniki testowania, jak i techniki testowania oparte na doświadczeniu mogą być stosowane na wszystkich poziomach testów.
- c) Odpowiedź poprawna. Czarnoskrzynkowe techniki testowania (znane również technikami opartymi na specyfikacji) opierają się na analizie określonego zachowania przedmiotu testów bez odniesienia do jego wewnętrznej struktury. Podstawą testu jest więc zazwyczaj specyfikacja. Techniki testowania oparte na doświadczeniu skutecznie wykorzystują wiedzę i doświadczenie testerów do projektowania i implementacji przypadków testowych. Oznacza to, że tester podczas projektowania testów może w ogóle nie korzystać ze specyfikacji.
- d) Odpowiedź niepoprawna. Techniki testowania oparte na doświadczeniu mogą wykrywać defekty, które mogą zostać pominięte przy użyciu czarnoskrzynkowych i białoskrzynkowych technik testowania. Dlatego techniki testowania oparte na doświadczeniu stanowią uzupełnienie technik czarno- i białoskrzynkowych, a zarówno techniki czarnoskrzynkowe, jak i techniki testowania oparte na doświadczeniu mogą być stosowane we wszystkich cyklach życia oprogramowania.

Pytanie 20 (FL-4.2.1, K3, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

Istnieje sześć klas równoważności KR1–KR6 w trzech zidentyfikowanych dziedzinach, które muszą być pokryte przypadkami testowymi.

- a) Odpowiedź poprawna. 1111 pokrywa KR2, KR3 i KR6. 1234 dodatkowo pokrywa KR1 i KR5. 12345 dodatkowo pokrywa KR4. Wszystkie klasy równoważności są pokryte.
- b) Odpowiedź niepoprawna. Żaden przypadek testowy nie pokrywa KR1.
- c) Odpowiedź niepoprawna. Żaden przypadek testowy nie pokrywa klas KR1 i KR5.
- d) Odpowiedź niepoprawna. Żaden przypadek testowy nie pokrywa KR4.

Pytanie 21 (FL-4.2.2, K3, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

Klasy równoważności to: {..., 99, 100}, {101, 102, ..., 198, 199}, {200, 201, ...}.

W tym podziale dziedziny istnieją 4 wartości brzegowe klas równoważności: 100, 101, 199 i 200. W dwupunktowej analizie wartości brzegowych dla każdej wartości brzegowej istnieją dwa elementy pokrycia: ta wartość brzegowa oraz jej najbliższy sąsiad należący do sąsiedniej klasy

równoważności. Ponieważ najbliżsi sąsiedzi są również wartościami brzegowymi sąsiednich klas równoważności, istnieją tylko cztery elementy pokrycia będące czterema wyżej wymienionymi wartościami brzegowymi.

Zatem:

- a) Odpowiedź niepoprawna. Tylko 100 i 200 są prawidłowymi elementami pokrycia dla dwupunktowej analizy wartości brzegowych, więc osiągamy 50% pokrycia.
- b) Odpowiedź niepoprawna. Tylko 100 i 200 są prawidłowymi elementami pokrycia dla dwupunktowej analizy wartości brzegowych, więc osiągamy 50% pokrycia.
- c) Odpowiedź niepoprawna. Tylko 100 i 101 są prawidłowymi elementami pokrycia dla dwupunktowej analizy wartości brzegowych, więc osiągamy 50% pokrycia.
- d) Odpowiedź poprawna. 101, 199 i 200 są poprawnymi elementami pokrycia dla dwupunktowej analizy wartości brzegowych, więc osiągamy 75% pokrycia.

Pytanie 22 (FL-4.2.3, K3, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Kombinacja (TAK, TAK, NIE) nie pasuje do żadnej reguły. Jest to przykład pominięcia, a nie sprzeczności.
- b) Odpowiedź niepoprawna. Kombinacja (TAK, NIE, TAK) pasuje tylko do jednej kolumny, „Reguła 2”, więc nie ma sprzeczności.
- c) Odpowiedź niepoprawna. Zarówno kombinacja (TAK, TAK, TAK), jak i (NIE, TAK, TAK) pasują tylko do jednej kolumny, „Reguła 1”, więc nie ma sprzeczności.
- d) Odpowiedź poprawna. Kombinacja (NIE, NIE, NIE) pasuje zarówno do kolumny „Reguła 2”, jak i „Reguła 3”, ale akcje dla tych kolumn są różne, więc pokazuje to sprzeczność między tymi dwoma regułami.

Pytanie 23 (FL-4.2.4, K3, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

Trzy przejścia etykietowane zdarzeniami „Available” oraz „Cancel” nie mogą pojawić się w tym samym przypadku testowym, co oznacza, że wymagane są co najmniej trzy przypadki testowe, z których każdy pokrywa jedno z tych trzech przejść. Pozostałe przejścia można pokryć rzeczywiście w ramach tych trzech przypadków testowych:

TC1: Room request, Available, Pay,

TC2: Room request, Not available, Available, Pay,

TC3: Room request, Not available, Cancel.

A zatem:

- a) Odpowiedź poprawna.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź niepoprawna.
- d) Odpowiedź niepoprawna.

Pytanie 24 (FL-4.3.2, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

W testowaniu gałęzi elementami pokrycia są gałęzie reprezentowane przez krawędzie w grafie przepływu sterowania. Graf przepływu sterowania zawiera 8 krawędzi.

Zatem:

- a) Odpowiedź niepoprawna.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź poprawna.
- d) Odpowiedź niepoprawna.

Pytanie 25 (FL-4.3.3, K2, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

- a) Odpowiedź poprawna. Przeprowadzanie wyłącznie testów czarnoskrzynkowych nie zapewnia pomiaru rzeczywistego pokrycia kodu. Pomiar pokrycia białoskrzynkowego zapewnia obiektywny pomiar pokrycia i dostarcza informacji niezbędnych do wygenerowania dodatkowych testów w celu zwiększenia tego pokrycia, a tym samym zwiększenia zaufania do kodu.
- b) Odpowiedź niepoprawna. To stwierdzenie jest poprawne, ale nie ma nic wspólnego z testowaniem czarnoskrzynkowym.
- c) Odpowiedź niepoprawna. Ogólnie rzecz biorąc, nie ma relacji subsumpcji między technikami czarno- i białoskrzynkowymi.
- d) Odpowiedź niepoprawna. Techniki białoskrzynkowe są wykorzystywane do projektowania testów w oparciu o sam przedmiot testów, natomiast techniki czarnoskrzynkowe są wykorzystywane do projektowania testów w oparciu o specyfikację. Dlatego też nie ma związku między elementami pokrycia wynikającymi z tych dwóch typów technik.

Pytanie 26 (FL-4.4.1, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Testowanie eksploracyjne wykorzystuje karty opisu testu, a nie listę możliwych defektów/awarii. Chociaż testowanie eksploracyjne może obejmować wykorzystanie innych technik testowania, w tym przypadku atak usterek jest najbardziej prawdopodobną opcją.
- b) Odpowiedź poprawna. Jest to lista możliwych awarii. Ataki usterek są metodycznym podejściem do zgadywania błędów i wymagają od testera stworzenia lub uzyskania listy możliwych błędów, defektów i awarii oraz zaprojektowania testów, które zidentyfikują defekty związane z błędami, ujawnią defekty lub spowodują awarie.
- c) Odpowiedź niepoprawna. Tester korzysta z listy kontrolnej w celu wsparcia testowania. Zarówno zgadywanie błędów, jak i testowanie oparte na liście kontrolnej wykorzystują

takie listy, jednak lista ta zawiera możliwe awarie, a nie warunki testowe, dlatego *najbardziej prawdopodobną* techniką testowania jest atak usterek, który koncentruje się na błędach, defektach i awariach.

- d) Odpowiedź niepoprawna. Analiza wartości brzegowych opiera się na wartościach brzegowych klas równoważności. Powyższa lista nie wspomina o klasach równoważności ani ich brzegach.

Pytanie 27 (FL-4.4.3, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Chociaż prawdą jest, że tester może zaimplementować i wykonać szczegółowe przypadki testowe w oparciu o listę kontrolną, nie wyjaśnia to, w jaki sposób miałyby to skutkować zwiększeniem pokrycia.
- b) Odpowiedź niepoprawna. Elementy listy kontrolnej nie powinny być automatyzowane. Ale nawet jeśli są, automatyczne skrypty testowe zawsze wykonują testy w ten sam sposób, co zazwyczaj nie skutkuje zwiększeniem pokrycia.
- c) Odpowiedź niepoprawna. Prawdą jest, że każdy element listy kontrolnej powinien być testowany oddzielnie i niezależnie. Ma to jednak wpływ na kolejność wykonywania testów, a nie na osiągnięte pokrycie, więc nie skutkuje zwiększeniem pokrycia.
- d) Odpowiedź poprawna. Jeśli listy kontrolne są wysokiego poziomu, prawdopodobnie wystąpi pewna zmienność w rzeczywistym testowaniu, co może skutkować większym pokryciem, ale mniejszą powtarzalnością. Jeśli dwóch testerów postępuje zgodnie z listą kontrolną zawierającą pozycje wysokiego poziomu, każdy z nich może używać innych danych testowych, kroków testowych itp. W ten sposób każdy tester prawdopodobnie pokryje niektóre obszary nieobjęte przez drugiego testera, co spowoduje zwiększenie pokrycia.

Pytanie 28 (FL-4.5.2, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. To kryterium akceptacji opisuje zasady lub przepisy, których system musi przestrzegać (w tym przypadku prawo do bycia zapomnianym). Jest to przykład kryterium akceptacji zapisanego w formacie zorientowanym na reguły.
- b) Odpowiedź poprawna. To kryterium akceptacji opisuje przykładowy scenariusz, który musi być możliwy do zrealizowania przez system. Jest to przykład kryterium akceptacji zapisanego w formacie zorientowanym na scenariusze.
- c) Odpowiedź niepoprawna. To zdanie wygląda bardziej jak linia kodu, która implementuje regułę biznesową. Kryteria akceptacji powinny być opracowywane we współpracy z przedstawicielami jednostek biznesowych, a zatem powinny być napisane językiem, który rozumieją. To zdanie najprawdopodobniej będzie niezrozumiałe dla tych interesariuszy.

- d) Odpowiedź niepoprawna. To kryterium akceptacji opisuje, jakich zasad lub przepisów musi przestrzegać system i w jaki sposób zapewniona zostanie zgodność. Dlatego jest to przykład kryterium akceptacji zapisanego w formacie zorientowanym na reguły, a nie na scenariusze.

Pytanie 29 (FL-4.5.3, K3, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Chcemy sprawdzić, czy użytkownicy specjalni mają uprawnienia użytkowników zwykłych, więc musimy przetestować uprawnienia dostępu dla użytkownika specjalnego, a nie dla użytkownika zwykłego.
- b) Odpowiedź niepoprawna. Chcemy sprawdzić, czy użytkownicy specjalni mają uprawnienia użytkowników zwykłych, więc musimy przetestować uprawnienia dostępu dla użytkownika specjalnego, a nie dla użytkownika zwykłego.
- c) Odpowiedź niepoprawna. W kryteriach akceptacji nie ma opisu piętra 5. Przypadki testowe nie powinny wykraczać poza zakres historii użytkownika. Ale nawet jeśli chcielibyśmy przeprowadzić testy negatywne, ten test nie jest bezpośrednio związany z KA3.
- d) Odpowiedź poprawna. W ten sposób możemy sprawdzić, czy użytkownik specjalny ma dostęp do pięter, które są dostępne dla użytkownika zwykłego.

Pytanie 30 (FL-5.1.1, K2, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

- a) Odpowiedź poprawna. Plan testów może zawierać *wymagania* dotyczące danych testowych (jako część podejścia do testów), ale nie może zawierać szczegółowych danych testowych dla przypadków testowych. Dane testowe są częścią przypadków testowych, a nie planu testów. Ponadto zazwyczaj nie jest możliwe zdefiniowanie takich danych podczas tworzenia planu testów, ponieważ nie wiadomo dokładnie, jak będą wyglądały moduły.
- b) Odpowiedź niepoprawna. Jednym z celów planu testów jest zapewnienie, że przeprowadzone czynności testowe spełnią ustalone kryteria, poprzez uwzględnienie kryteriów wejścia i wyjścia. Kryteria pokrycia kodu są przykładem takich kryteriów dla poziomu testowania modułowego.
- c) Odpowiedź niepoprawna. Szablony dokumentacji są typową częścią planu testów. Pomaga to ułatwić komunikację między interesariuszami poprzez zdefiniowanie standardowego sposobu komunikacji lub raportowania.
- d) Odpowiedź niepoprawna. Jednym z celów planu testów jest wykazanie, że testy będą zgodne z istniejącą polityką testów i strategią testów, lub wyjaśnienie, dlaczego testy będą od nich odbiegać. Jest to przykład wyjaśnienia odchylenia w odniesieniu do poziomów testów, które będą (lub nie będą) przestrzegane.

Pytanie 31 (FL-5.1.4, K3, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

Z wykresu wynika, że: $A(4) = 6$ i $A(3) = 8$ (dwa ostatnie szare pola).

Podstawiając do wzoru otrzymujemy:

$$E(5) = \frac{3 \cdot A(4) + A(3)}{4} = \frac{3 \cdot 6 + 8}{4} = \frac{26}{4} = 6,5 \text{ osobodni}$$

Zatem:

- a) Odpowiedź niepoprawna.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź poprawna.
- d) Odpowiedź niepoprawna.

Pytanie 32 (FL-5.1.5, K3, 1 pkt)

Poprawna odpowiedź: a

Uzasadnienie

Chcemy wykonać przypadki testowe zgodnie z ich priorytetami, ale musimy również wziąć pod uwagę zależności.

Jeśli weźmiemy pod uwagę tylko priorytety, najpierw należy wykonać TC 5 i TC 7 (najwyższy priorytet), następnie TC 1, TC 3 i TC 4, a na końcu TC 2 i TC 6 (najniższy priorytet). Jednak, aby uruchomić TC 7, musimy najpierw uruchomić TC 4. Z kolei, aby uruchomić TC 5, musimy uruchomić TC 4 i TC 2, ale TC 2 jest blokowane przez TC 1, które powinno zostać uruchomione przed TC 2.

Aby więc jak najwcześniej uruchomić przypadki testowe o priorytecie 1, pierwsze pięć przypadków testowych powinno wyglądać następująco: TC 4 – TC 7 – TC 1 – TC 2 – TC 5.

Następnie musimy uruchomić TC 3, ponieważ ma on wyższy priorytet niż TC 6. Pełny harmonogram będzie więc wyglądał następująco: TC 4 – TC 7 – TC 1 – TC 2 – TC 5 – TC 3 – TC 6. Dlatego szóstym przypadkiem testowym będzie TC 3.

A zatem:

- a) Odpowiedź poprawna.
- b) Odpowiedź niepoprawna.
- c) Odpowiedź niepoprawna.
- d) Odpowiedź niepoprawna.

Pytanie 33 (FL-5.1.6, K1, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Model piramidy testów nie dostarcza informacji o priorytetach testów.

- b) Odpowiedź poprawna. Model piramidy testów pokazuje, że różne testy mają różne poziomy szczegółowości.
- c) Odpowiedź niepoprawna. Model piramidy testów jest niezależny od kryteriów pokrycia.
- d) Odpowiedź niepoprawna. Model piramidy testów nie pokazuje żadnych relacji między różnymi testami.

Pytanie 34 (FL-5.1.7, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Kwadranty testowe grupują poziomy testów i typy testów według kilku kryteriów. Nie reprezentują one żadnych kombinacji poziomów testów i typów testów i nie są powiązane z żadnym etapem cyklu życia oprogramowania. Zarówno poziomy testów, jak i typy testów są traktowane oddzielnie w modelu kwadrantów testowych.
- b) Odpowiedź niepoprawna. Kwadranty testowe grupują poziomy testów i typy testów według kilku kryteriów. Nie opisują one stopnia szczegółowości poszczególnych typów testów wykonywanych na każdym poziomie testów. Taki model, odnoszący się do poziomów testów, nazywany jest piramidą testów.
- c) Odpowiedź niepoprawna. Stwierdzenie jest błędne, ponieważ ogólnie rzecz biorąc, każdy typ testu może być wykonywany na każdym poziomie testów.
- d) Odpowiedź poprawna. Kwadranty testowe grupują poziomy testów, typy testów, czynności testowe, techniki testowania i produkty pracy w zwinnym wytwarzaniu oprogramowania. W tym modelu testy mogą być ukierunkowane na biznes lub technologię. Testy mogą wspierać zespół (tj. kierować rozwojem) lub krytykować produkt (tj. mierzyć jego zachowanie w stosunku do oczekiwań). Połączenie tych dwóch punktów widzenia określa cztery kwadranty.

Pytanie 35 (FL-5.2.3, K2, 1 pkt)

Poprawna odpowiedź: c

Uzasadnienie

- a) Odpowiedź niepoprawna. Monitorowanie ryzyka jest częścią kontroli ryzyka, a nie analizy ryzyka.
- b) Odpowiedź niepoprawna. Sama identyfikacja ryzyka nie pozwala nam wdrożyć działań łagodzących ryzyko. Działania te są wdrażane podczas fazy kontroli ryzyka.
- c) Odpowiedź poprawna. Jest to przykład wpływu analizy ryzyka na dokładność i zakres testowania.
- d) Odpowiedź niepoprawna. Elementy pokrycia są określane przy użyciu technik testowania, a nie poprzez analizę ryzyka.

Pytanie 36 (FL-5.3.2, K2, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Raporty o postępie testów są najczęściej wykorzystywane podczas monitorowania testów i nadzoru nad testami oraz po ich ukończeniu, a nie podczas projektowania testów.
- b) Odpowiedź poprawna. Sumaryczny raport z testów jest przygotowywany po ukończeniu testów, gdy projekt, poziom testów lub typ testów został ukończony i, w idealnym przypadku, spełnione zostały kryteria wyjścia. Raport ten wykorzystuje informacje zawarte w raportach o postępie testów oraz inne dane.
- c) Odpowiedź niepoprawna. Raporty o postępie testów są najczęściej wykorzystywane podczas monitorowania testów i nadzoru nad testami oraz ukończenia testów, a nie podczas analizy testów.
- d) Odpowiedź niepoprawna. Raporty o postępie testów są najczęściej wykorzystywane podczas monitorowania testów i nadzoru nad testami oraz ukończenia testów, a nie podczas planowania testów.

Pytanie 37 (FL-5.4.1, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Gdy użytkownik zgłasza awarię oprogramowania, dzięki unikalnej identyfikacji zatwierdzeń zmian (ang. *commit*) możliwe jest ponowne odtworzenie plików z wersji oprogramowania używanej przez użytkownika (a także odpowiednich wersji skryptów testowych), a tym samym odtworzenie awarii i szybsze zlokalizowanie defektu.
- b) Odpowiedź niepoprawna. Jeśli zmiana w środowisku testowym powoduje nieoczekiwane problemy podczas testowania, zarządzanie konfiguracją pozwala testerom przywrócić poprzednią wersję środowiska. Dzięki temu testy mogą być kontynuowane bez wpływu zmiany na nie.
- c) Odpowiedź niepoprawna. Zarządzanie konfiguracją zapewnia, że wszystkie zidentyfikowane dokumenty (np. specyfikacje wymagań) i elementy oprogramowania są jednoznacznie przywołane w dokumentacji testowej (np. w planach testów).
- d) Odpowiedź poprawna. Zapewnia to proces zarządzania defektami, a nie proces zarządzania konfiguracją.

Pytanie 38 (FL-5.5.1, K3, 1 pkt)

Poprawna odpowiedź: b

Uzasadnienie

- a) Odpowiedź niepoprawna. Jest to ważne, ale nie tak ważne jak elementy środowiska testowego.
- b) Odpowiedź poprawna. Brakuje jednak ważnej informacji, jaką jest identyfikacja przeglądarki i urządzenia użytego do testowania. Informacje o przeglądarce i urządzeniu

są ważne, ponieważ taka usterka może być specyficzna dla danej przeglądarki lub urządzenia. Na przykład przycisk logowania może działać poprawnie w jednej przeglądarce (lub jednej wersji konkretnej przeglądarki), ale nie w innej. Dlatego informacje o przeglądarce i urządzeniu mogą pomóc programistom w odtworzeniu problemu i szybszym znalezieniu jego przyczyny.

- c) Odpowiedź niepoprawna. Przedmiot testów jest zidentyfikowany (WebShop v0.99).
- d) Odpowiedź niepoprawna. Uwzględniono wpływ – jest to krytyczność (wysoka).

Pytanie 39 (FL-6.1.1, K2, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. Narzędzia do wykonywania testów i pomiaru pokrycia ułatwiają automatyczne wykonywanie przypadków testowych oraz pomiar pokrycia osiągniętego poprzez uruchomienie tych przypadków testowych. Jednak narzędzia te nie pomagają w organizacji defektów i zarządzaniu konfiguracją.
- b) Odpowiedź niepoprawna. Narzędzia do projektowania i implementacji testów ułatwiają generowanie przypadków testowych, danych testowych i procedur testowych, ale nie pomagają w organizacji defektów i zarządzaniu konfiguracją.
- c) Odpowiedź niepoprawna. Narzędzia do zarządzania defektami służą do zarządzania defektami, ale nie są narzędziami testowymi i nie są używane do organizowania przypadków testowych ani zarządzania konfiguracją.
- d) Odpowiedź poprawna. Narzędzia do zarządzania testami zwiększają wydajność procesu testowego, ułatwiając zarządzanie cyklem życia oprogramowania, wymaganiami, testami, defektami i konfiguracją.

Pytanie 40 (FL-6.2.1, K1, 1 pkt)

Poprawna odpowiedź: d

Uzasadnienie

- a) Odpowiedź niepoprawna. „Możliwość generowania przypadków testowych bez dostępu do podstawy testów” nie jest zaletą automatyzacji testów (jest to niemożliwe). Generowanie przypadków testowych przez testerów lub narzędzia wymaga dostępu do podstawy testów.
- b) Odpowiedź niepoprawna. „Osiągnięcie większego pokrycia dzięki bardziej obiektywnej ocenie” nie jest bezpośrednią korzyścią wynikającą z automatyzacji testów. Automatyzacja testów zapewni bardziej obiektywną ocenę pokrycia, jednak ta obiektywna ocena nie zwiększy pokrycia. Tylko wykorzystanie wyników pokrycia do napisania dalszych przypadków testowych może potencjalnie zwiększyć pokrycie.
- c) Odpowiedź niepoprawna. „Wydłużenie czasu wykonywania testów dzięki większej mocy obliczeniowej” jest stwierdzeniem sprzecznym, ponieważ większa moc obliczeniowa zazwyczaj skraca czas wykonywania, a wydłużenie czasu wykonywania nie jest korzyścią, ponieważ testowanie trwałoby dłużej.

- d) Odpowiedź poprawna. „Zapobieganie błędom ludzkim dzięki większej spójności i powtarzalności” jest korzyścią wynikającą z automatyzacji testów, ponieważ automatyzacja testów nie jest narażona na błędy ludzkie. Oznacza to na przykład, że testy są konsekwentnie opracowywane na podstawie wymagań, dane testowe są tworzone w sposób systematyczny, a testy są wykonywane przez narzędzie w tej samej kolejności, w ten sam sposób i z tą samą częstotliwością.