

Egzamin przykładowy Odpowiedzi

Sylabus ISTQB® Certyfikowany Tester – Poziom Podstawowy Agile – Tester Zwinny

v 1.3 (2014)
PL w. 1.3.0.2 (2023)



Prawa autorskie

Autorzy (jako obecni posiadacze praw autorskich), ISTQB® (jako przyszły właściciel praw autorskich), tłumacze oraz SJSI zgodzili się na następujące warunki użytkowania: każda uznawana przez ISTQB® Rada Krajowa może wykonać tłumaczenie niniejszego egzaminu przykładowego oraz udzielać zezwolenia na korzystanie z całości lub części tłumaczenia innym stronom.

Każdy akredytowany dostawca szkoleń może użyć tego przykładowego egzaminu w swoim kursie szkoleniowym, pod warunkiem, że autorzy i ISTQB® są podani jako źródło i właściciele praw autorskich przykładowego egzaminu. Powoływanie się na niniejszy egzamin przykładowy w materiałach reklamowych i promocyjnych dozwolone jest dopiero po oficjalnym rozpoczęciu procedury ubiegania się o akredytację w ISTQB® lub w Radzie Krajowej (National Board) uznawanej przez ISTQB®. Każda osoba lub grupa osób może używać tego przykładowego egzaminu w artykułach i książkach, pod warunkiem, że autorzy i ISTQB® są podani jako źródło i właściciele praw autorskich do przykładowego egzaminu. Jakikolwiek inne użycie tego przykładowego egzaminu jest zabronione bez uprzedniego uzyskania pisemnej zgody ISTQB®. Każda Rada Krajowa uznana przez ISTQB® może przetłumaczyć ten przykładowy egzamin pod warunkiem, że powieli wyżej wymienioną informację o prawach autorskich w przetłumaczonej wersji przykładowego egzaminu.

Niniejszy dokument może być kopiowany w całości lub publikowany w wybranych fragmentach z podaniem źródła.

Odpowiedzialność za dokument

Dokument ten jest utrzymywany przez podstawowy zespół z ISTQB® składający się z Grupy Roboczej ds. Syllabusu i Grupy Roboczej ds. Egzaminów.

Podziękowania

Dokument ten został opracowany przez zespół z Grupy Roboczej ds. Egzaminów ISTQB® (International Software Testing Qualifications Board Examination Working Group): Mette Bruhn-Pedersen, Debra Friedenberg, Jen Leger, Lloyd Roden, Lucjan Stapp, Patricia McQuaid oraz przez autorów sylabusu Rozszerzenia Poziomu Podstawowego - Tester Zwinny¹.

Zespół dziękuje zespołowi z Grupy Roboczej ds. Egzaminów ISTQB®, który dokonał przeglądu dokumentu, autorom sylabusu Rozszerzenia Poziomu Podstawowego – Tester Zwinny² oraz osobom z Rad Krajowych za ich uwagi i sugestie.

Dokument został formalnie dopuszczony przez Grupę Roboczą ds. Egzaminów ISTQB® 1 maja 2014 roku.

Wersję polską przygotował zespół: Adam Wałęska, Lucjan Stapp – tłumaczenie, Jan Sabak – przegląd, edycja po przeglądzie: Lucjan Stapp.

Redakcja tekstu i korekta tłumaczenia (2023): Monika Petri-Starego, Lucjan Stapp, Adam Roman.

Prawa autorskie zastrzeżone © Stowarzyszenie Jakości Systemów Informatycznych (SJSI).

¹ Aktualna nazwa tego poziomu certyfikacji to Certyfikowany Tester – Poziom Podstawowy Agile – Tester Zwinny

² Patrz przypis 1

Historia zmian

Wersja	Data	Uwagi
1.3	28.04.2023 r.	Korekta odpowiedzi nr 5, 14, 29, 38. Zmiana formatu odpowiedzi dla pytań nr: 5, 21, 22, 23, 33, 35, 36, 39.
1.2.2	12.05.2021 r.	Usunięcie błędnej, zduplikowanej informacji o prawach autorskich
1.2.1	07.05.2021 r.	Aktualizacja noty o prawach autorskich.
1.2.	05.08.2019 r.	Korekta układu dokumentu.
1.1	06.06.2019 r.	Drobne zmiany
1.0	maj 2014 r.	Pierwsza wersja zatwierdzona przez GA

Historia zmian wersji polskiej

Wersja	Data	Uwagi
1.3.0.2. PL	19.06.2023 r.	Aktualizacja do wersji 1.3. oryginału. Korekta odpowiedzi nr 5, 14, 29, 38. Zmiana formatu odpowiedzi dla pytań nr: 5, 21, 22, 23, 33, 35, 36, 39. Poprawa tłumaczenia terminu “framework” na “struktura”, poprawa tłumaczenia terminu “najpierw testuj” na “najpierw test”.
w.2014.01.	2014 r.	Tłumaczenie zespół SJSI.

0. Wstęp

0.1 Cel tego dokumentu.

Dokument zawiera pełny przykładowy egzamin zgodnie z zasadami opisanymi w dokumencie „ISTQB® Foundation Extensions Exam Structure and Rules”³.

Przykładowe pytania oraz zbiór odpowiedzi i uzasadnień w obu dokumentach zostały opracowane przez zespół ekspertów dziedzinowych i doświadczonych twórców pytań w celu wsparcia Rad Krajowych ISTQB® i Rad Egzaminacyjnych przy pisaniu pytań egzaminacyjnych, a także dla planujących zdawanie egzaminu ISTQB® Certyfikowany Tester - Tester Zwinny.

Pytań tych nie wolno używać podczas prawdziwego egzaminu, ale mogą służyć one jako przewodnik dla tworzących pytania. Dając szeroki przekrój formatów i tematów, te przykładowe pytania egzaminacyjne (wraz z zestawem odpowiedzi) prezentują wiele pomysłów odnoszących się do tego, jak tworzyć dobre pytania, które mogą być wykorzystane przez poszczególne Rady Krajowe. Co więcej, firmy szkoleniowe mogą używać tych pytań jako część szkoleń przygotowujących do egzaminu.

0.2 Instrukcje

Zbiór pytań i odpowiedzi jest zorganizowany następująco:

- Cel nauczania i poziom K
- Pytanie – włączając w to scenariusz, po którym następuje właściwe pytanie
- Zbiór odpowiedzi

Następnie dla wszystkich pytań podano:

- Cel nauczania i poziom K
- Poprawna odpowiedź - razem z uzasadnieniem odpowiedzi.

³ Aktualna nazwa tego dokumentu to „Exam Structures and Rules Version 1.1 Compatible with Syllabus on Foundation and Advanced Levels, and Specialist Modules”

Pytanie 1

FA-1.1.1 (K1) Kandydat pamięta podstawowe zasady zwinnego wytwarzania oprogramowania w oparciu o Manifest Agile

Uzasadnienie:

- A. Niepoprawne – patrz uzasadnienie poprawnej odpowiedzi.
- B. Poprawne – Manifest Agile zawiera cztery zasady: ludzie i współpraca ponad procesy i narzędzia; działające oprogramowanie ponad obszerną dokumentację; współpraca z klientem ponad formalne ustalenia; reagowanie na zmiany ponad podążanie za planem.
- C. Niepoprawne – patrz uzasadnienie poprawnej odpowiedzi.
- D. Niepoprawne – patrz uzasadnienie poprawnej odpowiedzi.

Punktacja: 1 punkt

Pytanie 2

FA-1.1.1 (K1) Kandydat pamięta podstawowe zasady zwinnego wytwarzania oprogramowania w oparciu o Manifest Agile

Uzasadnienie:

- A. Poprawne – z perspektywy klienta działające oprogramowanie jest bardziej użyteczne i wartościowe niż obszerna dokumentacja i - co więcej - daje możliwość szybkiego przekazania zespołowi twórczemu szybkiej informacji zwrotnej.
- B. Niepoprawne – to jest typowa praktyka w wytwarzaniu sterowanym testami TDD, ale nie jest to zasada z Manifestu Agile.
- C. Niepoprawne – wartością jest: współpraca z klientem ponad formalne ustalenia.
- D. Niepoprawne – wartością jest: reagowanie na zmiany ponad podążanie za planem.

Punktacja: 1 punkt

Pytanie 3

FA-1.1.2 (K2) Kandydat rozumie korzyści wynikające z podejścia „cały zespół”.

Uzasadnienie:

- A. Niepoprawne – to zależy od umiejętności zespołu, programiści mogą wykonywać to zadanie.
- B. Niepoprawne – zespół powinien pracować razem przy wyborze narzędzi, które umożliwią im wspólne i efektywne działanie.
- C. Poprawne – testerzy wspomagają i współpracują z przedstawicielami biznesu, pomagając im tworzyć porządne testy akceptacyjne.
- D. Poprawne – w projektach zwinnych cały zespół odpowiada za jakość.
- E. Niepoprawne – programiści mogą pomagać przy tych zadaniach w zależności od umiejętności zespołu i indywidualnego obciążenia.

Punktacja: 1 punkt

Pytanie 4

FA-1.1.2 (K2) Kandydat rozumie korzyści wynikające z podejścia „cały zespół”.

Uzasadnienie:

- A. Niepoprawne – umiejętności związane z testowaniem oprogramowania powinny być przekazywane także członkom zespołu niebędącymi testerami.
- B. Niepoprawne – to zależy od umiejętności zespołu i dostępności, niektórzy testerzy mają podstawy programistyczne.
- C. Poprawne – umożliwia równoważenie różnych umiejętności, tak jak jest to potrzebne w projekcie.
- D. Niepoprawne – testerzy – specjaliści są ciągle potrzebni i są istotnym zasobem w projektach zwinnych.

Punktacja: 1 punkt

Pytanie 5

FA-1.1.3 (K2) Kandydat rozumie korzyści wynikające z wczesnego i częstego otrzymywania informacji zwrotnej.

Uzasadnienie:

Rozważ następujące kwestie:

1. Wczesne informacje zwrotne gwarantują, że programiści wdrażają to, co jest potrzebne.
2. Wczesna informacja zwrotna pozwala zespołom zwinnym dostarczyć cechy o największej wartości biznesowej jako pierwsze.
3. Może być wymagane więcej testów ze względu na częste zmiany.
4. Klienci wskazują, czy wymagania zostały pominięte lub źle zinterpretowane, i modyfikują żadaną funkcjonalność.

Stąd:

- A. Niepoprawne
- B. Niepoprawne
- C. Poprawne
- D. Niepoprawne

Punktacja: 1 punkt

Pytanie 6

FA-1.1.3 (K2) Kandydat rozumie korzyści wynikające z wczesnego i częstego otrzymywania informacji zwrotnej.

Uzasadnienie:

- A. Niepoprawne – taką samą liczbę defektów można znaleźć przy użyciu dowolnego procesu wytwarzania oprogramowania. Zaletą podejścia zwinnego jest możliwość szybszego znajdowania i usuwania defektów.
- B. Poprawne – wyjaśniając żądania klienta odnośnie cech, wcześniej i regularnie podczas wytwarzania, zwiększa się prawdopodobieństwo, że kluczowe cechy będą dostępne dla klienta wcześniej i produkt lepiej będzie odzwierciedlał życzenia klienta.
- C. Niepoprawne – w podejściu zwinnym nie wyróżnia się nikogo indywidualnie, ocenia się cały zespół.
- D. Niepoprawne – może nie być wystarczająco dużo czasu, by ukończyć wszystkie cechy w danej iteracji, ale proces zwinny umożliwia zespołowi skupienie się na cechach o największej wartości biznesowej.

Punktacja: 1 punkt

Pytanie 7

FA-1.2.1 (K1) Kandydat pamięta podejścia zwinne do wytwarzania oprogramowania.

Uzasadnienie:

- A. Niepoprawne – patrz na poprawną odpowiedź dla poprawnego odwzorowania.
- B. Poprawne – programowanie ekstremalne (XP) obejmuje 5 wartości kierujących wytwarzaniem oprogramowania: komunikacja, prostota, informacja zwrotna, odwaga oraz szacunek. Scrum dzieli projekt na krótkie iteracje zwane sprintami. W Kanbanie nie mamy ani iteracji, ani sprintu, służy do optymalizacji ciągłego przepływu zadań i minimalizacji czasu potrzebnego na przerobienie każdego zadania.
- C. Niepoprawne – patrz na poprawną odpowiedź dla poprawnego odwzorowania.
- D. Niepoprawne – patrz na poprawną odpowiedź dla poprawnego odwzorowania.

Punktacja: 1 punkt

Pytanie 8

FA-1.2.2 (K3) Kandydat potrafi napisać historyjki użytkownika we współpracy z wytwórcami, przedstawicielami biznesu i właścicielem produktu.

Uzasadnienie:

- A. Niepoprawne –rozważanie testowalności i automatyzacji jest istotne, ale projektowanie aplikacji w oparciu o ograniczenie wysiłku testowego może nie skutkować pożądanym rozwiązaniem dla użytkownika końcowego.
- B. Niepoprawne – to właściciel produktu priorytetyzuje różne charakterystyki jakościowe.
- C. Niepoprawne – wydajnościowe kryteria akceptacji na ogół powinny być określone przez właściciela produktu.
- D. Poprawne – tester przyczynia się do zapewnienia, że zespół tworzy kryteria akceptacji dla historyjki użytkownika.

Punktacja: 1 punkt

Pytanie 9

FA-1.2.3 (K2) Kandydat rozumie, jak spotkania retrospektywne mogą być wykorzystywane jako mechanizm do doskonalenia procesu w projektach zwinnych.

Uzasadnienie:

- A. Niepoprawne – testerzy powinni brać udział we wszystkich elementach spotkania retrospektywnego.
- B. Niepoprawne – testerzy powinni brać udział we wszystkich elementach spotkania retrospektywnego.
- C. Poprawne – wszyscy członkowie zespołu, testerzy i nie-testerzy, powinni wносить wkład zarówno do czynności testowych, jak i nietestowych.
- D. Niepoprawne – testerzy mogą nauczyć się wartościowych informacji podczas spotkań retrospektywnych, by zastosować je w następnych iteracjach.

Punktacja: 1 punkt

Pytanie 10

FA-1.2.3 (K2) Kandydat rozumie, jak spotkania retrospektywne mogą być wykorzystywane jako mechanizm do doskonalenia procesu w projektach zwinnych.

Uzasadnienie:

- A. Niepoprawne – to powinno być poruszane, by wspomóc wcześniejsze wykrywanie defektów w procesie.
- B. Niepoprawne – to powinno być poruszane jako udoskonalenie procesu.
- C. Poprawne – spotkania retrospektywne nie powinny być przeznaczane do oceny osób, ale powinny koncentrować się na udoskonalaniu procesu i zespołu jako całości.
- D. Niepoprawne - to powinno być poruszane jako udoskonalenie procesu.

Punktacja: 1 punkt

Pytanie 11

FA-1.2.4 (K2) Kandydat rozumie wykorzystanie i cele ciągłej integracji.

Uzasadnienie:

- A. Niepoprawne - to jest zasada ciągłej integracji, wersje są tworzone co najmniej raz dziennie z automatycznym rozmieszczeniem i wykonaniem zautomatyzowanych testów modułowych (jednostkowych) i integracyjnych.
- B. Niepoprawne – ciągła integracja pozwala na ciągłą dostępność wykonywalnego oprogramowania w dowolnym miejscu i czasie, dla testowania, pokazywania lub celów edukacyjnych.
- C. Niepoprawne – praktyka ciągłej integracji pozwala programistom na ciągłe dokonywanie integracji i ciągłe testowanie, więc defekty w kodzie mogą być szybko wykrywane.
- D. Poprawne – testowanie powinno być automatyzowane na poziomie modułowym (jednostkowym) i integracyjnym, by umożliwić szybką informację zwrotną na temat jakości wersji.

Punktacja: 1 punkt

Pytanie 12

FA-1.2.5 (K1) Kandydat zna różnice pomiędzy planowaniem iteracji i wydania, a także wie, w jaki sposób tester dodaje wartości każdej z tych aktywności.

Uzasadnienie:

- A. Niepoprawne – tego oczekuje się podczas planowania iteracji.
- B. Niepoprawne – tego oczekuje się podczas planowania iteracji.
- C. Niepoprawne – tego oczekuje się podczas planowania iteracji.
- D. Poprawne – tego oczekuje się podczas planowania wydania.

Punktacja: 1 punkt

Pytanie 13

Termin (słowo kluczowe) z sylabusu Tester Zwinny (K1)

Uzasadnienie:

- A. Niepoprawne – tester bierze udział w tworzeniu historyjki użytkownika.
- B. Niepoprawne – historyjka użytkownika może zawierać zarówno wymagania funkcjonalne, jak i нефункционалне.
- C. Niepoprawne – historyjkę użytkownika piszą wspólnie programiści, testerzy i przedstawiciele biznesu.
- D. Poprawne – w środowisku zwinnym historyjki użytkownika są pisane tak, by uchwycić wymagania z punktu widzenia programistów, testerów i przedstawicieli biznesu. Wspólne tworzenie historyjki użytkownika może wykorzystywać takie techniki jak burza mózgów lub mapa myśli.

Punktacja: 1 punkt

Pytanie 14

FA-2.1.1 (K2) Kandydat potrafi opisać różnice pomiędzy testowaniem w projektach zwinnych i tradycyjnych (nie zwinnych).

Uzasadnienie:

- A. Niepoprawne – testowanie zwinne promuje lekką dokumentację.
- B. Poprawne – automatyzacja testów na wszystkich poziomach występuje w wielu zespołach zwinnych. O ile programiści koncentrują się na automatycznych testach na poziomie modułowym (jednostkowym), to testerzy powinni koncentrować się na automatyzacji testów na poziomie testów integracyjnych, systemowych i akceptacyjnych. W tradycyjnych projektach to nie jest typowe, by tak samo koncentrować się na automatyzacji. Czasami wykonuje się automatyzację raz, gdy zakończone są testy systemowe, by pracować ze stabilnym systemem lub tylko automatyzuje się regresję dla celów pielęgnacyjnych po produkcyjnym wdrożeniu systemu.
- C. Niepoprawne – decyduje zespół.
- D. Niepoprawne – zaangażowanie testera to dobra praktyka we wszystkich modelach wytwarzania.

Punktacja: 1 punkt

Pytanie 15

FA-2.1.2 (K2) Kandydat potrafi opisać, w jaki sposób czynności kodowania i testowania są zintegrowane w podejściu zwinnym.

Uzasadnienie:

- A. Poprawne – te trzy perspektywy (testera, programisty i przedstawiciela biznesu) są ważne, by określić, kiedy cecha jest ukończona.
- B. Niepoprawne – kryteria wejścia i wyjścia do danego poziomu testów są bardziej związane z tradycyjnym cyklem życia.
- C. Niepoprawne – cechy powinny być weryfikowane w tej samej iteracji, w której są wytwarzane.
- D. Niepoprawne – cechy powinny być weryfikowane w tej samej iteracji, w której są wytwarzane.

Punktacja: 1 punkt

Pytanie 16

FA-2.1.3 (K2) Kandydat potrafi opisać rolę niezależnego testowania w projektach zwinnych.

Uzasadnienie:

- A. Poprawne – jest to cecha charakterystyczna projektów zwinnych.
- B. Niepoprawne – wiele zespołów w projektach zwinnych ciągle ma niezależne zespoły testowe z kierownikami testów.
- C. Niepoprawne – testowanie to ciągle wyspecjalizowana rola w projekcie zwinnym, zwłaszcza wtedy, gdy wykonywana jest właściwie.
- D. Niepoprawne – programiści i testerzy pracują razem, by zbudować i przetestować cechę.
- E. Poprawne – zespoły zwinne mogą stosować różne formy testów akceptacyjnych.

Punktacja: 1 punkt

Pytanie 17

FA-2.1.3 (K2) Kandydat potrafi opisać rolę niezależnego testowania w projektach zwinnych.

Uzasadnienie:

- A. Niepoprawne – to jest prawdziwe stwierdzenie. To się zdarza, gdy tester i programista pracują razem.
- B. Poprawne – to jest nieprawdziwe stwierdzenie. Niezależny tester MOŻE znaleźć więcej defektów niż programista, ale to zależy od poziomu wykonywanych testów, a także od kompetencji niezależnego testera.
- C. Niepoprawne – to jest prawdziwe stwierdzenie. To jest opcja, która zachowuje poziom niezależności, gdy są oddzielne zespoły testerów i programistów, a testerzy są przydzielani na żądanie na końcu sprintu.
- D. Niepoprawne – to jest prawdziwe stwierdzenie. Ta opcja jest spełniona, gdy są pewne wyspecjalizowane prace testowe w czynnościach wykonywanych poza sprintem lub długookresowych.

Punktacja: 1 punkt

Pytanie 18

FA-2.2.1 (K2) Kandydat potrafi opisać narzędzia i techniki stosowane do informowania o statusie testów w projekcie zwinnym, włączając w to postęp testów i jakość produktu.

Uzasadnienie:

- A. Niepoprawne – to może być wskaźnik jakości, ale zakłada się tu, że wykonano dostateczne testowanie, by zidentyfikować wszystkie możliwe usterki. Co więcej, nie zidentyfikowano, czy system może być traktowany w tym momencie jako „działające oprogramowanie”.
- B. Poprawne – pozytywna informacja zwrotna od użytkownika i działające oprogramowanie to kluczowe wskaźniki jakości produktu.
- C. Niepoprawne – to jest dobry wskaźnik tempa pracy zespołu, ale nie dostarcza informacji o jakości produktu.
- D. Niepoprawne – to jest dobry wskaźnik tempa pracy zespołu, ale też nie dostarcza informacji o jakości produktu.

Punktacja: 1 punkt

Pytanie 19

FA-2.2.1 (K2) Kandydat potrafi opisać narzędzia i techniki stosowane do informowania o statusie testów w projekcie zwinnym, włączając w to postęp testów i jakość produktu.

Uzasadnienie:

- A. Poprawne – wykres spalania pokazuje planowany postęp i datę wydania razem z bieżącym postępem prac nad historijkami użytkownika.
- B. Niepoprawne – rejestry (logi) automatyzacji pokazują, które testy przeszły, a które nie i nie jest to związane z żadną formą szacowania.
- C. Niepoprawne – jakkolwiek zwinna tablica zadań pokazuje postęp, ta informacja jest następnie wykorzystywana w wykresie spalania, jednakże tablica zadań pokazująca postęp historyjek użytkownika i zadań nie ma związku z szacowaniem.
- D. Niepoprawne - narzędzie do śledzenia defektów może pokazywać postęp w raportowaniu usterek i może być używane do określenia poziomu jakości produktu. Nie pokazuje jednak postępu prac zespołu względem szacowań.

Punktacja: 1 punkt

Pytanie 20

FA-2.2.2 (K2) Kandydat potrafi opisać proces ewoluowania testów w czasie wielu iteracji i potrafi wyjaśnić, dlaczego automatyzacja testów jest istotna przy zarządzaniu ryzykiem regresji w projektach zwinnych.

Uzasadnienie:

- A. Poprawne – ponieważ ta cecha była już dostarczona, przegląd wszystkich zasobów testowych jest wymagany, co może skutkować aktualizacją przypadków testowych, tak by pokrywały one nowe kryteria akceptacji. Zapewni to, że testy fałszywie negatywne (tzn. testy raportujące nieistniejące defekty) nie będą miały miejsca. To jest początkowe zadanie, które trzeba wykonać przed decyzją, czy jakieś zmiany trzeba przedsięwziąć.
- B. Niepoprawne – to mogłoby być początkowe zadanie do wykonania, gdyby tester wiedział, jakie nowe testy są wymagane dla tych zmian bez przeglądania uprzednio istniejących testów. Być może nie trzeba dodawać nowych testów – wystarczy aktualizacja istniejących.
- C. Niepoprawne – jakkolwiek jest to dobra praktyka, nie zostało zaadresowane specyficzne ryzyko regresji zidentyfikowane w tym scenariuszu.
- D. Niepoprawne – podobnie jak w B. Bez przeglądu istniejących testów dla tej cechy nie wiadomo, czy dodatkowa automatyzacja jest wymagana.

Punktacja: 1 punkt

Pytanie 21

FA-2.2.2 (K2) Kandydat potrafi opisać proces rozwoju testów w czasie wielu iteracji i potrafi wyjaśnić, dlaczego automatyzacja testów jest istotna przy zarządzaniu ryzykiem regresji w projektach zwinnych.

Uzasadnienie:

- i. Zwinność oczekuje zmiany i zarządza zmianą, więc każda iteracja będzie wymagała coraz więcej i więcej testów regresji. Jeżeli nie zostanie wykorzystana automatyzacja, prędkość zespołu spadnie.
- ii. To nie jest powód, by wprowadzać automatyzację do projektu.
- iii. Nie możemy ponownie przetestować/uruchomić wszystkich przypadków testowych z poprzedniej iteracji. Wytworzono wiele przypadków testowych, większość to ręczne testy eksploracyjne i nie jest możliwa automatyzacja wszystkiego.
- iv. Automatyzacja ma umożliwić ominięcie regresji w produkcie ze względu na dużą ilość zmian. Ale to nie gwarantuje, że nie pojawią się defekty.
- v. Narzędzia automatyzacji są połączone z narzędziami do ciągłej integracji, które będą wykonywane i wskażą, jeżeli nowy kod popsuje zbudowaną wersję.

Stąd:

- A. Niepoprawne
- B. Poprawne
- C. Niepoprawne
- D. Niepoprawne

Punktacja: 1 punkt

Pytanie 22

FA-2.3.1 (K2) Kandydat rozumie umiejętności (ludzkie, dziedzinowe i testowe) testera w zespole zwinnym.

Uzasadnienie:

Rozważ następujące kwestie:

- i. Projekty zwinne obejmują zmiany i oczekują zmian, ale to nie znaczy, że tak dzieje się codziennie.
- ii. To jest prawda; im wcześniej zespół zwinny otrzyma informację zwrotną o jakości, tym lepiej.
- iii. „Najpierw test” i „ciągła integracja” wymagają, by testy były zautomatyzowane, a to dostarcza informacji zwrotnej o jakości wersji jako część procesu automatyzacji wersji
- iv. Testowanie powinno być wykonywane podczas całej iteracji, a nie tylko na końcu.
- v. Projekty zwinne wymagają różnych poziomów testów, takich jak modułowe (jednostkowe), systemowe i akceptacyjne.

- A. Niepoprawne
- B. Niepoprawne
- C. Niepoprawne
- D. Poprawne

Punktacja: 1 punkt

Pytanie 23

FA-2.3.2 (K2) Kandydat rozumie rolę testera w zespole zwinnym.

Uzasadnienie:

Rozważ następujące kwestie:

- i. To zadanie to wspólny wysiłek całego zespołu.
- ii. Oczekuje się, że jest to działanie testera zwinnego.
- iii. W projektach zwinnych defekty są przedstawiane interesariuszom regularnie.
- iv. To jest typowa czynność testera zwinnego.
- v. Programowanie w parach w sytuacjach typowych jest wykonywane przez dwóch deweloperów; nie oczekuje się, by tester poprawiał logikę programu, jakkolwiek może przeglądać kod pod kątem testowalności lub pielęgnowalności.

- A. Niepoprawne
- B. Niepoprawne
- C. Poprawne
- D. Niepoprawne

Punktacja: 1 punkt

Pytanie 24

FA-2.3.2 (K2) Kandydat rozumie rolę testera w zespole zwinnym.

Uzasadnienie:

- A. Niepoprawne – to prawda. Część roli testera to tworzenie skryptów automatycznych, ich uruchamianie i zarządzanie nimi.
- B. Niepoprawne – to prawda. Tester powinien szkolić (coaching) wszystkich członków zespołu w zagadnieniach związanych z testowaniem.
- C. Poprawne – to jest fałsz. To rola Scrum Mastera (lub innej równoważnej roli w innych metodykach zwinnych), by stworzyć i uaktualniać wykres spalania na podstawie informacji otrzymywanych od innych członków zespołu.
- D. Niepoprawne – w projekcie zwinnym tester dostarcza informacji zwrotnej o produkcie na wszystkich etapach, co może zawierać czynności związane z analizą kodu.

Punktacja: 1 punkt

Pytanie 25

Termin (słowo kluczowe) z sylabusu Tester Zwinny (K1)

Uzasadnienie:

- A. Niepoprawne – to objaśnienie bardziej odnosi się do „wypalania” niż do całkowitego „spalenia”
- B. Niepoprawne – ta definicja opisuje tablicę zadań w projekcie zwinnym.
- C. Poprawne – wykres spalania pokazuje postęp prac nad historyjkami użytkownika, które są zakończone (zrobione) oraz oszacowanie pozostałego czasu potrzebnego do zakończenia pozostałych historyjek użytkownika w sprincie.
- D. Niepoprawne – wykres spalania nie zawiera żadnych odnośników do defektów usuniętych lub czekających na usunięcie.

Punktacja: 1 punkt

Pytanie 26

FA-3.1.1 (K1) Kandydat pamięta pojęcia: wytwarzanie sterowane testami (TDD), wytwarzanie sterowane testami akceptacyjnymi (ATDD), wytwarzanie sterowane zachowaniem (BDD).

Uzasadnienie:

- A. Niepoprawne – wytwarzanie sterowane testami (TDD) jest techniką używaną do wytwarzania kodu kierowaną przez zautomatyzowane przypadki testowe. Jest znana także jako zasada programowania „najpierw test”, bo testy pisane są przed stworzeniem kodu. Testy są zautomatyzowane i używane w ciągłej integracji.
- B. Niepoprawne – proces TDD jest powtarzany dla każdego małego fragmentu kodu, uruchamiamy zarówno poprzednie testy, jak i dodane testy.
- C. Niepoprawne – testy służą jako forma wykonywalnej specyfikacji projektu dla przyszłych prac pielęgnacyjnych.
- D. Poprawne – to jest prawdziwe dla BDD, a nie dla TDD.

Punktacja: 1 punkt

Pytanie 27

FA-3.1.2 (K1) Kandydat pamięta pojęcie piramida testów.

Uzasadnienie:

- A. Niepoprawne – nakład pracy testowej w każdym sprincie nie ma nic wspólnego z pojęciem piramidy testów.
- B. Niepoprawne – wielkość backlogu do testowania i ilość testów nie mają nic wspólnego z pojęciem piramidy testów.
- C. Poprawne – piramida testów eksponuje fakt, że mamy więcej testów na niższych poziomach i malejącą ilość testów na wyższych poziomach.
- D. Niepoprawne – ilość zautomatyzowanych testów nie ma nic wspólnego z pojęciem piramidy testów.

Punktacja: 1 punkt

Pytanie 28

FA-3.1.3 (K2) Kandydat potrafi opisać kwadranty testowe i ich związki z poziomami testów i typami testów.

Uzasadnienie:

- A. Poprawne – kwadranty testowe mogą być używane jako pomoc przy opisywaniu typów testów wszystkim interesariuszom.
- B. Niepoprawne – to nie jest dobra metryka, bo nie wszystkie poziomy/typy testów mają zastosowanie dla danego systemu.
- C. Niepoprawne – ilość testów dla każdego kwadrantu zależy od testowanego systemu i rzadko jest równa dla wszystkich kwadrantów. W pewnych sytuacjach może nawet nie być w ogóle testów dla kwadrantu.
- D. Niepoprawne – kwadranty testowe nie mają korelacji z poziomem ryzyka.

Punktacja: 1 punkt

Pytanie 29

FA-3.1.3 (K2) Kandydat potrafi opisać kwadranty testowe i ich związki z poziomami testów i typami testów.

Uzasadnienie:

- A. Niepoprawne – te przypadki testowe nie są testami jednostkowymi zorientowanymi na technologię (Q1), ani (testami systemowymi zorientowanymi na biznes (Q2)) .
- B. Niepoprawne – testy wydajności (TC3) nie są częścią ani 2^{go} kwadrantu, ani 3^{go} kwadrantu.
- C. Poprawne – testy użyteczności są częścią 3^{go} kwadrantu, akceptacji użytkownika – 4^{go}.
- D. Niepoprawne – testy użyteczności nie są częścią ani 2^{go}, ani 4^{go} kwadrantu.

Punktacja: 1 punkt

Pytanie 30

FA-3.1.4 (K3) Kandydat potrafi pełnić rolę testera w zespole scrumowym dla danego projektu zwinnego.

Uzasadnienie:

- A. Niepoprawne – modyfikacja struktury do testów automatycznych i skryptów po to, by wspomóc nowy typ przeglądarki, może nie być warte wysiłku, jeżeli ryzyko znalezienia nowych defektów jest niskie. Cały zespół powinien wykonać analizę ryzyka i podjąć wspólną decyzję.
- B. Poprawne – cały zespół powinien podjąć decyzję o modyfikacji struktury do testów automatycznych i skryptów. Tester jest następnie odpowiedzialny za dokonanie zmian w planie iteracji, o ile jest to niezbędne.
- C. Niepoprawne – tester musi powiadomić zespół, który następnie wspólnie zadecyduje, co zrobić z tą kwestią.
- D. Niepoprawne – tester sam nie określa zakresu prac. Ta kwestia powinna być zaadresowana poprzez stworzenie nowej historyjki użytkownika lub modyfikację istniejącej, i będzie adresowana do całego zespołu podczas planowania sprintu.

Punktacja: 1 punkt

Pytanie 31

FA-3.2.1 (K3) Kandydat potrafi ocenić ryzyka jakościowe produktu w projekcie zwinnym.

Uzasadnienie:

- A. Poprawne – informacje z analizy ryzyka są używane podczas sesji pokera planistycznego do określenia priorytetów pozycji, które mają być zrobione podczas iteracji. Tylko po sesji pokera planistycznego dodaje się pozycje do backlogu, jeżeli zdecydowano, że nie wszystko da się wykonać podczas iteracji.
- B. Niepoprawne – w tym punkcie nie wiemy, czy mamy czas na zakończenie wszystkich zadań w iteracji. To, że coś ma wysokie ryzyko, nie oznacza, że wymaga dużego wysiłku, by to zrobić. Będziemy to wiedzieli po sesjach pokera planistycznego.
- C. Niepoprawne – długość iteracji w czasie nie zmienia się. Po sesji pokera planistycznego pewne pozycje mogą być przesunięte do backlogu, jeżeli zdecydowano, że nie wszystko da się wykonać podczas iteracji.
- D. Poprawne – łagodzenie ryzyka może być wykonane, zanim rozpocznie się wykonywanie testów, by ograniczyć poziom ryzyka.
- E. Niepoprawne – sesja pokera planistycznego powinna odbywać się najpierw, by określić, co będzie wykonane w danej iteracji. Jeżeli zdecydowano, że nie ma wystarczająco dużo czasu na zakończenie wszystkich pozycji, prawdopodobnie te z mniejszym ryzykiem będą dodane do backlogu przyszłych sprintów.

Punktacja: 1 punkt

Pytanie 32

FA-3.2.2 (K3) Kandydat potrafi oszacować nakład pracy testowej na podstawie zawartości iteracji i ryzyk jakościowych produktu.

Uzasadnienie:

- A. Niepoprawne – użytkownicy i deweloperzy mogą przeoczyć trudności techniki testowania niezbędnej do walidacji historyjki użytkownika. Dyskusja powinna być kontynuowana, aż cały zespół będzie zgodny w oszacowaniu.
- B. Poprawne – sesje pokera planistycznego dla historyjki użytkownika powinny być kontynuowane, aż zespół będzie usatysfakcjonowany oszacowaniem nakładu pracy.
- C. Niepoprawne – cały zespół musi zgodzić się na oszacowanie historyjki użytkownika. Sam użytkownik nie rozumie złożoności wytwarzania lub testowania funkcjonalności.
- D. Niepoprawne – nie jest konieczne, by się zgadzali, regułą może być branie najwyższego oszacowania lub średniej z trzech oszacowań. O tym decyduje zespół przed rozpoczęciem sesji pokera planistycznego.

Punktacja: 1 punkt

Pytanie 33

FA-3.3.1 (K3) Kandydat potrafi zinterpretować odpowiednie informacje w celu wsparcia czynności testowych.

Uzasadnienie:

Rozważ następujące kwestie:

- i. To jest pomocne, bo wiemy, że jest nowa wersja standardu; istniejące przypadki testowe powinny być zmodyfikowane lub nowe przypadki testowe powinny być dodane.
- ii. To jest pomocne podczas fazy analizy ryzyka.
- iii. Ta informacja nie jest pomocna, ponieważ dostęp użytkownika zmienia się z nowym wydaniem urządzenia i nowe historyjki użytkownika są dokumentowane.
- iv. Ponieważ wprowadzana jest nowa technologia, należy uzyskać wersje podstawowe (*baselines*) przy użyciu urządzeń o podobnej technologii lub zdefiniowanych wymagań wydajnościowych dla tego typu technologii.
- v. To jest pomocne podczas fazy analizy ryzyka.

Stąd:

- A. Niepoprawne
- B. Niepoprawne
- C. Poprawne
- D. Niepoprawne

Punktacja: 1 punkt

Pytanie 34

FA-3.3.2 (K2) Kandydat potrafi wyjaśnić interesariuszom biznesowym, jak definiować testowalne kryteria akceptacji.

Uzasadnienie:

- A. Niepoprawne – zarówno przypadki testowe, jak i karty opisu testów, są używane jako podstawa do określenia, co testować. Ilość wykonanych przypadków testowych nie podaje informacji o tym, co zostało pokryte. (Podobnie ilość kart opisu testów nie daje żadnej wartościowej informacji o pokryciu).
- B. Niepoprawne – to stwierdzenie samo w sobie jest niewystarczające. Powinno być podparte dodatkowymi informacjami dotyczącymi pokrycia testów i związanego ryzyka.
- C. Poprawne – otrzymane pokrycie testowe z dodatkowymi informacjami to najlepszy wybór, nawet jeżeli więcej informacji będzie potrzebnych. Powinno to zawierać informację o znalezionych defektach, częstości ich występowania oraz taksonomii (jak wiele ważnych problemów występuje w poszczególnych obszarach). Informacje te stanowią pełniejszą podstawę do wydania decyzji o wydaniu. Będziesz także potrzebował informacji dotyczącej ocenionych charakterystyk i informacje o tym, jak one wpływają na obraz całości ze względu na zakończenie systemu i związanego z tym testowania.
- D. Niepoprawne – zakończenie iteracji/sprintu implikuje, że kończysz testowanie, bo nie ma już czasu, co nie jest najlepszym kryterium potrzebnym do podjęcia decyzji, kiedy zakończyć testowanie.

Punktacja: 1 punkt

Pytanie 35

FA-3.3.2 (K2) Kandydat potrafi wyjaśnić interesariuszom biznesowym, jak definiować testowalne kryteria akceptacji.

Uzasadnienie:

- A. Niepoprawne – to nie jest testowalne; brak szczegółów dotyczących typu testów białoskrzynkowych lub oczekiwanego pokrycia.
- B. Poprawne – to jest testowalne.
- C. Poprawne – to jest testowalne.
- D. Niepoprawne – to nie jest testowalne; nie wiemy, jaki jest rozsądny czas odpowiedzi.
- E. Niepoprawne – to nie jest testowalne; powinna być podana specyfikacja, jakie przeglądarki powinny być przetestowane. Musiano by przyjąć założenie, które przeglądarki są główne.

Punktacja: 1 punkt

Pytanie 36

FA-3.3.3 (K3) Kandydat potrafi napisać przypadki testowe dla wytwarzania sterowanego testami akceptacyjnymi na podstawie danej historyjki użytkownika.

Uzasadnienie:

- i. Test odnosi się do historii transakcji użytkownika.
- ii. Test odnosi się do roli kasjera bankowego i w wyniku pokazuje transakcje klienta banku.
- iii. Test odnosi się do roli kasjera bankowego i w wyniku pokazuje transakcje klienta banku.
- iv. Test odnosi się do roli kasjera bankowego i w wyniku pokazuje transakcje klienta banku.
- v. W historyjce użytkownika nie wspomina się o wymaganiach wydajnościowych.

Stąd:

- A. Niepoprawne
- B. Niepoprawne
- C. Niepoprawne
- D. Poprawne

Punktacja: 1 punkt

Pytanie 37

FA-3.3.4 (K3) Kandydat potrafi napisać przypadki testowe zarówno funkcjonalne, jak i нефunkcjonalne, używając technik czarnoskrzynkowych na podstawie danej historyjki użytkownika.

Uzasadnienie:

- A. Niepoprawne – podstawowym zagadnieniem historyjki użytkownika nie jest stan systemu; oczekuje się, że będzie testowany koszt wysyłki.
- B. Niepoprawne – podstawowym zagadnieniem historyjki użytkownika nie jest to, czy przesyłka zostanie nadana zgodnie z oczekiwaniami; oczekuje się, że będzie testowany koszt wysyłki.
- C. Poprawna – analiza wartości brzegowych to najlepsza opcja, gdy testujemy koszt wysyłki.
- D. Niepoprawne – podstawowym zagadnieniem historyjki użytkownika nie jest to, czy przesyłka zostanie nadana zgodnie z oczekiwaniami; oczekuje się, że będzie testowany koszt wysyłki.

Punktacja: 1 punkt

Pytanie 38

FA-3.3.5 (K3) Kandydat potrafi wykonać testy eksploracyjne, by wspomóc testowanie w projekcie zwinnym.

Uzasadnienie:

- A. Poprawne – w testach eksploracyjnych dokumentacja wykonywania testów i rzeczywistych rezultatów jest niezbędna. Rejestrowanie wyłącznie defektów nie jest wystarczające.
- B. Niepoprawne – testowanie eksploracyjne jest znane jako podejście do testów oparte na doświadczeniu, które jest tak efektywne, jak tester wykonujący test. Zyskiem z tego podejścia jest to, że testy, które zostaną zaprojektowane i wykonane, będą wpływały na następny zbiór testów, który będzie zaprojektowany i wykonany.
- C. Niepoprawne – testowanie eksploracyjne to nie technika testowania, ale podejście do testów, gdzie można używać innych technik takich jak testowanie sposobem par, drzewo klasyfikacji, analiza wartości brzegowych itp.
- D. Niepoprawne – zaleca się ograniczenie czasu trwania sesji testów eksploracyjnych w karcie opisu testu. Odpowiedź ta oznacza, że nigdy nie powinno być limitu czasu.

Punktacja: 1 punkt

Pytanie 39

FA-3.4.1 (K1) Kandydat zna różne narzędzia dostępne testerom oraz ich podział według przeznaczenia i czynności w projekcie zwinnym.

Uzasadnienie:

- A. Niepoprawne – to mógłby być jeden z celów użycia wiki, a nie narzędzia ALM.
- B. Niepoprawne – to mógłby być jeden z celów narzędzia do ciągłej integracji (CI), a nie narzędzia ALM.
- C. Poprawne – to jest jeden z wielu celów używania narzędzia ALM, ale użycie narzędzia umożliwia lepszą współpracę w zespole rozproszonym niż fizyczna tablica zadań.
- D. Niepoprawne – to mógłby być jeden z celów użycia narzędzia do generowania i ładowania danych, a nie narzędzie ALM.

Punktacja: 1 punkt

Pytanie 40

Termin (słowo kluczowe) z sylabusa Tester Zwinny (K1)

Uzasadnienie:

- A. Niepoprawne – to jest prawda, patrz sekcja 3.3.5 sylabusa.
- B. Poprawne – karty opisu testów są tworzone przed wykonaniem testów i zawierają cele i pomysły dotyczące testów.
- C. Niepoprawne – to jest prawda, patrz sekcja 3.3.4 sylabusa.
- D. Niepoprawne – to jest prawda; tester powinien dobrze rozumieć, jak system jest używany i co spowoduje, że zawiedzie.

Punktacja: 1 punkt