

Certyfikowany tester

sylabus Poziomu Zaawansowanego

Zarządzanie Testami

wersja 3.0

(wersja PL 3.0.0.1)

International Software Testing Qualifications Board



Informacja o prawach autorskich

Informacja o prawach autorskich © International Software Testing Qualifications Board (zwana dalej ISTQB®). ISTQB® jest zastrzeżonym znakiem towarowym International Software Testing Qualifications Board.

Copyright © 2023, autorami aktualizacji 3.0 są Horst Pohlmann (właściciel produktu, wiceprezes AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma i Miroslav Renda.

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

Tłumaczenie z języka angielskiego wersji 3.0 – KONTEKST A. Wolski spółka komandytowa.

Przegląd końcowy przeprowadził zespół w składzie: Adam Roman (kierownik zespołu), Monika Petri-Starego.

Redakcja tekstu i korekta tłumaczenia: Monika Petri-Starego.

Copyright © 2010-2012 autorzy – Podgrupa Robocza ds. Poziomu Zaawansowanego Kierownik Testów: Rex Black (przewodniczący), Judy McKay (wiceprzewodnicząca), Graham Bath, Debra Friedenberg, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto.

Wszelkie prawa zastrzeżone. Autorzy niniejszym przenoszą prawa autorskie na ISTQB®. Autorzy (jako obecni posiadacze praw autorskich) oraz ISTQB® (jako przyszły posiadacz praw autorskich) wyrazili zgodę na następujące warunki użytkowania:

Kopiowanie fragmentów niniejszego dokumentu w celach niekomercyjnych jest dozwolone pod warunkiem wskazania źródła. Akredytowani dostawcy szkoleń mogą opracowywać na podstawie niniejszego sylabusu własne szkolenia pod warunkiem wskazania autorów i ISTQB® jako źródła sylabusu i właścicieli praw autorskich do niego. Zastrzega się jednak, że umieszczenie odwołań do niniejszego sylabusu w ewentualnych materiałach reklamowych dotyczących szkolenia jest dozwolone dopiero po uzyskaniu oficjalnej akredytacji materiałów szkoleniowych ze strony uznawanej przez ISTQB® Rady Krajowej.

Osoby fizyczne i grupy osób fizycznych mogą opracowywać na podstawie niniejszego sylabusu artykuły i książki pod warunkiem wskazania autorów i ISTQB® jako źródła sylabusu i właścicieli praw autorskich do niego.

Korzystanie z sylabusu do innych celów bez wcześniejszej pisemnej zgody ISTQB® jest zabronione.

Każda uznawana przez ISTQB® Rada Krajowa może dokonywać przekładu niniejszego sylabusu pod warunkiem powielenia powyższych uwag dotyczących praw autorskich w przetłumaczonej wersji dokumentu.

Historia zmian

Wersja	Data	Uwagi
ISEB v1.1	4.09.2001 r.	Sylabus Praktyczny ISEB
ISTQB 1.2E	09.2003 r.	Sylabus poziomu zaawansowanego ISTQB od EOQ-SG
V2007	12.10.2007 r.	Certyfikowany tester – poziom zaawansowany – sylabus w wersji 2007
D100626	10.06.2010 r.	Wprowadzenie zmian przyjętych w 2009 roku, rozdzielanie każdego rozdziału na osobne moduły
D101227	10.12.2010 r.	Przyjęcie zmian formatu i poprawek, które nie mają wpływu na znaczenie zdań
D2011	31.10.2011 r.	Zmiana na podzielony sylabus, sformułowanie na nowo celów nauczania (LO) i zmiany treści w celu dopasowania do celów nauczania; dodanie wyników biznesowych (BO)
Alpha 2012	9.02.2012 r.	Uwzględnienie wszystkich uwag NB otrzymanych od wydania październikowego
Beta 2012	26.03.2012 r.	Uwzględnienie uwag NB otrzymanych na czas od wydania Alpha
Beta 2012	7.04.2012 r.	Przesłanie wersji beta do GA
Beta 2012	8.06.2012 r.	Wersja z edytowaną treścią przekazana do NB
Beta 2012	27.06.2012 r.	Uwzględnienie uwag EWG i uwag grupy Glossary
RC 2012	15.08.2012 r.	Wersja do potencjalnego wydania – zawiera ostateczne poprawki NB
Beta v3.0	31.10.2023 r.	Uwzględnienie wszystkich uwag od rad krajowych na temat wszystkich rozdziałów (BOIncs) po przeglądzie wersji Alpha
POST Beta v3.0	31.01.2024 r.	Uwzględnienie wszystkich uwag od rad krajowych na temat wszystkich rozdziałów (BOIncs) po przeglądzie wersji Beta
POST Beta v3.0	29.02.2024 r.	Drobne korekty tekstu
RC v3.0	28.03.2024 r.	Wersja do potencjalnego wydania – uwzględniono najnowsze formalne zmiany szablonu zaproponowane przez PWG
V.3.0	3.05.2024 r.	Poprawki po wydaniu; wyeliminowanie wyłącznie literówek i niespójności

Historia zmian polskiej wersji sylabusa

Wersja	Data	Uwagi
w. 3.0.0.1	14.10.2024 r.	Zmiana tłumaczenia ze „śledzenie wymagań” na „śledzenie powiązań” (p. 2.3.5 oraz 5. Załącznik A).
w. 3.0.0.0	12.08.2024 r.	Publikacja polskiej wersji sylabusa

Spis treści

Informacja o prawach autorskich	2
Historia zmian	3
Historia zmian polskiej wersji sylabusu.....	3
Spis treści.....	4
Podziękowania	8
0. Wstęp.....	10
0.1 Cel niniejszego sylabusu.....	10
0.2 Certyfikowany Tester – Poziom Zaawansowany – Zarządzanie Testami w procesie testowania	10
0.3 Ścieżka kariery testerów	10
0.4 Cele biznesowe.....	11
0.5 Cele nauczania objęte egzaminem i poziomy wiedzy.....	11
0.6 Egzamin certyfikacyjny z zarządzania testami na poziomie zaawansowanym	12
0.7 Akredytacja	12
0.8 Odniesienia do norm.....	12
0.9 Poziom szczegółowości	13
0.10 Struktura sylabusu	13
0.11 Jakie są podstawowe założenia tego sylabusu?	14
1. Zarządzanie czynnościami testowymi – 750 minut	16
1.2 Kontekst testowania.....	16
1.3 Testowanie oparte na ryzyku	16
1.4 Projektowa strategia testowa	17
1.6 Narzędzia testowe	17
1.1 Proces testowy.....	18
1.1.1 Czynności związane z planowaniem testów	18
1.1.2 Czynności monitorowania testów i nadzoru nad testami.....	19
1.1.3 Czynności związane z ukończeniem testów	20

1.2	Kontekst testowania	21
1.2.1	Interesariusze testów	21
1.2.2	Znaczenie wiedzy interesariuszy w zarządzaniu testami	21
1.2.3	Zarządzanie testami w hybrydowym modelu wytwarzania oprogramowania	22
1.2.4	Czynności związane z zarządzaniem testami w różnych modelach cyklu wytwarzania oprogramowania	23
1.2.5	Czynności związane z zarządzaniem testami na różnych poziomach testów	24
1.2.6	Czynności związane z zarządzaniem testami w różnych typach testów	25
1.2.7	Czynności związane z zarządzaniem testami w celu planowania, monitorowania i nadzoru nad testami	26
1.3	Testowanie oparte na ryzyku	28
1.3.1	Testowanie jako działanie łagodzące ryzyko	28
1.3.2	Identyfikacja ryzyk jakościowych	28
1.3.3	Ocena ryzyka jakościowego	29
1.3.4	Łagodzenie ryzyka jakościowego poprzez odpowiednie testy	30
1.3.5	Techniki testowania oparte na ryzyku	31
1.3.6	Wskaźniki sukcesu i trudności związane z testowaniem opartym na ryzyku	32
1.4	Projektowa strategia testowa	34
1.4.1	Wybór podejścia do testów	34
1.4.2	Analiza organizacyjnej strategii testowej, kontekstu projektu i innych aspektów	35
1.4.3	Definicja celów testów	36
1.5	Usprawnianie procesu testowego	38
1.5.1	Proces usprawniania testów (IDEAL)	38
1.5.2	Usprawnianie procesu testowego w oparciu o model	39
1.5.3	Podejście do usprawniania procesu testowego w oparciu o analizę	40
1.6	Narzędzia testowe	43
1.6.2	Techniczne i biznesowe aspekty decyzji dotyczących narzędzi	44

1.6.3	Rozważania dotyczące procesu wyboru i ocena zwrotu z inwestycji	44
1.6.4	Cykl życia narzędzia	46
1.6.5	Metryki narzędzia	46
2.	Zarządzanie produktem – 390 minut	48
2.1	Metryki testów	49
2.1.1	Metryki czynności związanych z zarządzaniem testami.....	49
2.1.2	Monitorowanie, nadzór i ukończenie testów	50
2.1.3	Raportowanie testów.....	51
2.2	Szacowanie testów	53
2.2.1	Oszacowanie, jakie czynności będzie obejmować testowanie.....	53
2.2.2	Czynniki, które mogą wpływać na nakład pracy związany z testowaniem.....	53
2.2.3	Wybór technik szacowania testów.....	54
2.3	Zarządzanie defektami.....	57
2.3.1	Cykl życia defektu	57
2.3.2	Międzyfunkcyjne zarządzanie defektami	59
2.3.3	Specyfika zarządzania defektami w zespołach zwinnych	60
2.3.4	Wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania	60
2.3.5	Informacje zawarte w raporcie o defekcie.....	61
2.3.6	Określanie działań usprawniających proces przy użyciu informacji z raportów o defektach.....	62
3.	Zarządzanie zespołem – 225 minut.....	64
3.1	Zespół testowy	65
3.1.1	Typowe umiejętności w ramach czterech obszarów kompetencji	65
3.1.2	Analiza wymaganych umiejętności członków zespołu testowego	66
3.1.3	Ocena umiejętności członków zespołu testowego	67
3.1.4	Rozwijanie umiejętności członków zespołu testowego	68
3.1.5	Umiejętności kierownicze wymagane do zarządzania zespołem testowym	68

3.1.6	Czynniki motywujące lub demotywuujące zespół testowy w określonych sytuacjach	69
3.2	Relacje z interesariuszami	71
3.2.1	Koszt jakości	71
3.2.2	Stosunek kosztów do korzyści testowania	72
4.	Bibliografia	74
	Normy	74
	Dokumenty ISTQB®	74
	Książki	74
	Artykuły	75
	Strony internetowe	75
5.	Załącznik A – Cele nauczania i poziomy poznawcze	76
	Poziom 1: Zapamiętać (K1)	76
	Poziom 2: Zrozumieć (K2)	76
	Poziom 3: Zastosować (K3)	77
	Poziom 4: Przeanalizować (K4)	77
6.	Załącznik B – Macierz powiązań między celami biznesowymi a celami nauczania	78
7.	Załącznik C – Opis wydania	86
8.	Załącznik D – Słowa kluczowe specyficzne dla domeny	88
9.	Załącznik E – Znaki towarowe	89
10.	Indeks	90

Podziękowania

Niniejszy dokument został formalnie opublikowany przez Walne Zgromadzenie ISTQB® w dniu 3 maja 2024 r.

Został on opracowany przez zespół International Software Testing Qualifications Board: Horst Pohlmann (właściciel produktu, wiceprezes AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma i Miroslav Renda.

Zespół pragnie podziękować Gary'emu Mogyorodi za recenzję techniczną (w wersji Beta), Julii Sabatine za korektę, a także zespołowi recenzentów i Radom Krajowym za ich sugestie i wkład.

W procesie przeglądu, zgłaszania uwag i głosowania nad niniejszym sylabusem uczestniczyły następujące osoby:

Przegląd wersji alfa: Benjamin Timmermans, Mattijs Kemmink, Rik Marselis, Jean-Francois Riverin, Gary Mogyorodi, Ralf Bongard, Ingvar Nordström, Yaron Tsubery, Imre Mészáros, Mattijs Kemmink, Ádám Bíró, Ramit M Kaul, Chinthaka Indikadahena, Darvay Tamás Béla, Beata Karpinska, Young jae Choi, Stuart Reid, Tal Pe'er, Meile Postuma, Daniel van der Zwan, Klaudia Dussa-Zieger, Jörn Münzel, Ralf Bongard, Petr Neugebauer, Derk-Jan de Grood, Rik Kochuyt, Andreas Hetz, Laura Albert, Eszter Sebestyeni, Tamás Szőke, Henriett Braunné Bokor, Ágota Horváth, Péter Sótér, Ferenc Hamori, Darvay Tamás Béla, Paul Weymouth, Lloyd Roden, Kevin Chen, Huang qin, Pushparajan Balasubramanian, Szilard Szell, Tamas Stöckert, Lucjan Stapp, Adam Roman, Anna Miazek, Márton Siska, Erhardt Wunderlich, László Kvintovics, Murian Song, Mette Bruhn-Pedersen, Petra Schneider, Michael Stahl, Ramit M Kaul, Imre Mészáros, Dilhan Jayakody, Francisca Cano Ortiz, Johan Klintin, Liang Ren, Ole Chr. Hansen, Zsolt Hargitai, Tamás Rakamazi, Kenji Onishi, Arnika Hryszko, Rabi Arabi, Veronica Belcher, and Vignesh Balasubramanian.

Przegląd wersji beta: Maria-Therese Teichmann, Dominik Weber, Thomas Puffler, Peter Kunit, Martin Klonk, Michaël Pilaeten, Wim Decoutere, Arda Ender Torçuk, Piet de Roo, Rik Marselis, Jakub Platek, Ding Guofu, Zheng Dandan, Liang Ren, Yifan Chen, Hallur Helmsdal, Ole Chr. Hansen, Klaus Skafte, Gitte Ottosen, Tanzeela Gulzar, Arne Becher, Klaudia Dussa-Zieger, Jan Giesen, Florian Fieber, Carsten Weise, Arnd Pehl, Matthias Hamburg, Stephanie Ulrich, Jürgen Beniermann, Márton Siska, Sterbinszky Ádám, Ágnes Srancsik, Marton Matyas, Tamas Stöckert, Csilla Varga, Zsolt Hargitai, Bíró Ádám, Horváth Ágota, Sebestyén Eszter, Szilárd Széll, Péter Sótér, Giancarlo Tomasig, Nicola de Rosa, Kaiwalya Katyarmal, Pradeep Tiwari, Sreeja Padmakumari, Seunghee Choi, Stuart Reid, Dmitrij Nikolajev, Mantas Anilius, Monika Stoecklein-Olsen, Adam Roman, Mahmoud Khalaili, Ingvar Nordström, Beata Karpinska, Armin Born, Ferdinand Gramsamer, Mergole Kuate, Thomas Letzkus, Nishan Portoyan, Ainsley Rood, Lloyd Roden, Sarah Ireton.

Sylabus Poziomu Zaawansowanego Kierownik Testów 2010–2012 został opracowany przez zespół podstawowy International Software Testing Qualifications Board – Podgrupę Roboczą ds. Poziomu Zaawansowanego Kierownik Testów: Rex Black (przewodniczący), Judy McKay (wiceprzewodnicząca), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Członkowie zespołu podstawowego składają podziękowania zespołowi recenzentów oraz wszystkim Radom Krajowym za ich sugestie i wkład.

W momencie ukończenia sylabusu poziomu zaawansowanego Grupa Robocza ds. Poziomu Zaawansowanego składała się z następujących osób (w porządku alfabetycznym):

Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenberg, Bernard Homès (wiceprzewodniczący), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Posthuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (przewodniczący), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

W procesie przeglądu, zgłaszania uwag i głosowania nad niniejszym sylabusem uczestniczyły następujące osoby:

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh, Hans Weiberg.

0. Wstęp

0.1 Cel niniejszego sylabusu

Niniejszy sylabus stanowi podstawę egzaminu International Software Testing Qualification na poziomie zaawansowanym w zakresie zarządzania testami. ISTQB® udostępnia sylabus:

1. Radom Krajowym w celu przetłumaczenia na języki lokalne i dokonania akredytacji dostawców szkoleń (Rady Krajowe mogą dostosowywać sylabus do potrzeb danego języka oraz modyfikować odwołania do literatury w celu uwzględnienia publikacji lokalnych);
2. organom certyfikacyjnym w celu sformułowania pytań egzaminacyjnych w językach lokalnych (dostosowanych do celów nauczania niniejszego sylabusu);
3. akredytowanym przez ISTQB® dostawcom szkoleń w celu opracowania materiałów dydaktycznych i określenia odpowiednich metod nauczania;
4. kandydatom do certyfikacji w celu przygotowania do egzaminu certyfikacyjnego (ISTQB® zaleca udział w szkoleniu akredytowanym przez ISTQB® przed przystąpieniem do egzaminu ISTQB® na poziomie zaawansowanym);
5. międzynarodowej społeczności specjalistów w dziedzinie inżynierii oprogramowania i systemów w celu rozwijania zawodu testera oprogramowania i systemów oraz opracowywania książek i artykułów.

0.2 Certyfikowany Tester – Poziom Zaawansowany – Zarządzanie Testami w procesie testowania

Niniejsza kwalifikacja na poziomie zaawansowanym jest skierowana do wszystkich osób zaangażowanych w zarządzanie testowaniem oprogramowania. Obejmuje to osoby na stanowiskach takich jak testerzy, konsultanci ds. testów, kierownicy testów, testerzy testowania akceptacyjnego przez użytkownika, scrum masterzy, kierownicy projektów czy właściele produktów. Ponadto kwalifikacja na poziomie zaawansowanym w zakresie zarządzania testami jest odpowiednia dla osób, które pragną zdobyć zaawansowaną wiedzę w dziedzinie testowania oprogramowania, takich jak: kierownicy projektów, kierownicy ds. jakości, właściele produktów, kierownicy ds. rozwoju oprogramowania, analitycy biznesowi, dyrektorzy IT oraz konsultanci ds. zarządzania. Posiadacze certyfikatu zarządzania testami na poziomie zaawansowanym będą mogli przejść do kwalifikacji w zakresie testowania oprogramowania na poziomie eksperta ISTQB®. Certyfikat ISTQB® Certyfikowany tester – Poziom Zaawansowany – Kierownik Testów lub Zarządzanie Testami jest ważny dożywotnio i nie wymaga odnawiania. Certyfikat jest uznawany na całym świecie i świadczy o kompetencjach zawodowych i wiarygodności kandydatów w zarządzaniu testami.

0.3 Ścieżka kariery testerów

System opracowany przez ISTQB® umożliwia osobom zawodowo zajmującym się testowaniem wsparcie na każdym etapie kariery. Osoby, które uzyskają certyfikat ISTQB® Certyfikowany Tester – Poziom Zaawansowany – Zarządzanie Testami, mogą być również zainteresowane innymi certyfikatami w obszarze Core na poziomie zaawansowanym (tj. Analityk Testów i Techniczny Analityk Testów), a następnie certyfikatami ISTQB® na poziomie eksperckim (tj. Zarządzanie Testami lub Wdrażanie Usprawnień Procesu Testowego). Natomiast osoby, które chcą rozwijać swoje umiejętności

w dziedzinie praktyk testowania zwinnego powinny wziąć pod uwagę certyfikat technicznego testera zwinnego (Techniczny Tester Zwinny) lub certyfikat ATLaS (Agile Test Leadership at Scale). Z kolei obszar Specialist umożliwia ścisłą specjalizację w dziedzinach, w których stosowane są wyspecjalizowane podejścia do testowania i czynności testowe (np. automatyzacja testów, testowanie z wykorzystaniem sztucznej inteligencji lub testowanie aplikacji mobilnych), bądź w których skupia się specjalistyczna wiedza w dziedzinie testowania dotycząca określonych sektorów przemysłu (takich jak branża automotive czy gry). Najnowsze informacje na temat systemu certyfikacji testerów ISTQB® można uzyskać w serwisie www.istqb.org.

0.4 Cele biznesowe

W tym podrozdziale wymieniono jedenaście celów biznesowych, które powinna realizować osoba otrzymująca certyfikat Zarządzanie Testami na Poziomie Zaawansowanym.

Umiejętności certyfikowanego testera Zarządzanie Testami na Poziomie Zaawansowanym:

TM_01	Zarządzanie testowaniem w różnych projektach wytwarzania oprogramowania z zastosowaniem procesów zarządzania testami ustanowionych dla zespołu projektowego lub organizacji testującej
TM_02	Określanie interesariuszy w dziedzinie testów i modeli cyklu wytwarzania oprogramowania, które są istotne w danym kontekście
TM_03	Organizowanie sesji na potrzeby określania i oceny ryzyka w ramach dowolnego cyklu wytwarzania oprogramowania i wykorzystywanie wyników do kierowania testami w celu osiągnięcia celów testów
TM_04	Określanie projektowej strategii testów zgodnej z organizacyjną strategią testów i z kontekstem projektu
TM_05	Ciągłe monitorowanie i kontrolowanie testów w celu osiągnięcia celów projektu
TM_06	Ocena i raportowanie postępów testów interesariuszom projektu
TM_07	Określanie niezbędnych umiejętności i rozwijanie ich w zespole
TM_08	Przygotowywanie i przedstawianie uzasadnień biznesowych do przeprowadzania testów w różnych kontekstach, z określeniem kosztów i oczekiwanych korzyści
TM_09	Prowadzenie działań usprawniających procesy testowe w projektach lub strumieniach produktów związanych z rozwojem oprogramowania oraz wnoszenie wkładu w organizacyjne inicjatywy na rzecz usprawnienia procesu testowego
TM_10	Planowanie działań testowych, w tym wymaganej infrastruktury testowej, szacowanie nakładu wymaganego do przeprowadzenia testów
TM_11	Tworzenie raportów o defektach i przepływu pracy dotyczącego defektów odpowiedniego w kontekście danego cyklu wytwarzania oprogramowania

0.5 Cele nauczania objęte egzaminem i poziomy wiedzy

Cele nauczania wspomagają osiąganie celów biznesowych i są wykorzystywane do opracowywania egzaminów certyfikacyjnych dla testerów na poziomie zaawansowanym.

Ogólnie rzecz biorąc, wszystkie treści tego sylabusu można objąć egzaminem na poziomie K1, z wyjątkiem wstępu, bibliografii, podsumowania i załączników. Oznacza to, że od kandydata może być wymagane rozpoznanie, zapamiętanie lub przywołanie słowa kluczowego lub pojęcia wymienionego

w każdym z trzech rozdziałów dokumentu. Poziomy wiedzy w odniesieniu do poszczególnych celów nauczania przedstawiono na początku każdego rozdziału. Poziomy te sklasyfikowano następująco:

- K2: Zrozumieć
- K3: Zastosować
- K4: Przeanalizować

Dalsze szczegóły i przykłady celów nauczania podano w Załączniku A.

Definicje wszystkich pojęć wymienionych jako słowa kluczowe pod tytułami rozdziałów należy zapamiętać (K1), nawet jeśli nie wspomniano o tym wyraźnie w celach nauczania.

0.6 Egzamin certyfikacyjny z zarządzania testami na poziomie zaawansowanym

Egzamin z zarządzania testami na poziomie zaawansowanym jest oparty na tym sylabusie. Przy udzielaniu odpowiedzi na pytania egzaminacyjne może być konieczne skorzystanie z materiału obejmującego więcej niż jeden rozdział tego sylabusu. Przedmiotem egzaminu może być treść wszystkich części sylabusu z wyjątkiem wstępu, załączników i bibliografii. W dokumencie znajdują się również odwołania do norm i książek (rozdział 5), ale ich treść nie może być przedmiotem egzaminu w zakresie wykraczającym poza informacje streszczone w tym sylabusie.

Więcej informacji można znaleźć w dokumencie „Exam Structures and Rules 1.1 Compatible with Syllabus Foundation and Advanced Levels and Specialists Modules”.

- Kryterium przystąpienia do egzaminu w celu uzyskania certyfikatu Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami:
 - o przed przystąpieniem do egzaminu certyfikacyjnego z zarządzania testami na poziomie zaawansowanym kandydaci muszą uzyskać certyfikat ISTQB® na poziomie podstawowym. Zdecydowanie zaleca się jednak, aby kandydat miał przynajmniej minimalne doświadczenie w tworzeniu lub testowaniu oprogramowania, np. sześciomiesięczne doświadczenie jako tester lub programista.

0.7 Akredytacja

Rada Krajowa ISTQB® może dokonywać akredytacji dostawców szkoleń, którzy oferują materiały dydaktyczne zgodne z niniejszym sylabusem. Wytyczne dotyczące akredytacji należy uzyskać od Rady Krajowej lub organu dokonującego akredytacji. Akredytowany kurs jest uznawany za zgodny z niniejszym sylabusem i może obejmować egzamin ISTQB®.

Wytyczne dotyczące akredytacji w zakresie niniejszego sylabusu są zgodne z ogólnymi wytycznymi dotyczącymi akredytacji opublikowanymi przez Grupę Roboczą ds. Zarządzania Procesami i Zgodności ISTQB®.

0.8 Odniesienia do norm

W sylabusie Zarządzanie Testami na Poziomie Zaawansowanym znajdują się odniesienia do norm (np. ISO, IEC i IEEE). Celem tych odniesień jest zapewnienie ram lub zapewnienie źródła dodatkowych informacji, jeśli czytelnik tego potrzebuje. Należy pamiętać, że sylabus wykorzystuje te normy jako bibliografię. Treść norm nie jest przedmiotem egzaminu. Więcej informacji na temat norm można znaleźć w rozdziale 5.

0.9 Poziom szczegółowości

Szczegółowość informacji zawartych w niniejszym sylabusie umożliwia tworzenie spójnych pod względem treści nauczania kursów i przeprowadzanie egzaminów na skalę międzynarodową. Aby sprostać temu zadaniu, w sylabusie uwzględniono:

- ogólne cele instruktażowe opisujące intencję sylabusu Zarządzanie Testami na Poziomie Zaawansowanym;
- wykaz słów kluczowych, które muszą zapamiętać uczestnicy szkolenia;
- cele nauczania w poszczególnych obszarach wiedzy, opisujące efekty kształcenia o charakterze poznawczym;
- opis kluczowych pojęć, w tym odniesienia do źródeł, takich jak przyjęta literatura lub normy.

Treść sylabusu nie stanowi opisu całego obszaru wiedzy związanego z testowaniem oprogramowania. Odzwierciedla ona jedynie poziom szczegółowości, jaki należy uwzględnić w szkoleniach z zakresu Zarządzania Testami na Poziomie Zaawansowanym. W sylabusie skupiono się na pojęciach i technikach związanych z testowaniem, które mogą mieć zastosowanie do wszystkich projektów związanych z oprogramowaniem.

0.10 Struktura sylabusu

Istnieją trzy rozdziały z treściami egzaminacyjnymi. Nagłówek najwyższego poziomu każdego rozdziału określa minimalny czas wymagany dla akredytowanych kursów szkoleniowych w celu omówienia treści rozdziału; czas nie jest podany poniżej poziomu rozdziału. W przypadku akredytowanych szkoleń sylabus wymaga co najmniej 22,75 godzin zajęć, podzielonych na trzy rozdziały w następujący sposób:

- Rozdział 1: Zarządzanie czynnościami testowymi (750 minut)
 - o Uczestnik uczy się objaśniać czynności związane z zarządzaniem testami (tj. planowanie testów, monitorowanie testów, nadzór nad testami i ich ukończenie).
 - o Uczestnik uczy się określać projektową strategię testów, w tym cele testów, i wybierać odpowiednie podejście do testów zgodne z organizacyjną strategią testów i z kontekstem projektu.
 - o Uczestnik uczy się zarządzać projektami w różnych kontekstach.
 - o Uczestnik uczy się stosować testy oparte na ryzyku w celu skoncentrowania testów na zidentyfikowanych ryzykach.
 - o Uczestnik uczy się prowadzić działania usprawniające proces testowy poprzez przeprowadzenie retrospektywy projektu lub iteracji.
 - o Uczestnik uczy się, jak poprawić wsparcie narzędziowe procesu testowego poprzez uwzględnienie ryzyk, kosztów i korzyści związanych ze wsparciem narzędziowym.
- Rozdział 2: Zarządzanie produktem (390 minut)
 - o Uczestnik uczy się, jak monitorować i kontrolować testowanie, aby osiągnąć cele testów przy użyciu metryk testów oraz oceniać i raportować postęp testów.
 - o Uczestnik uczy się, jak wybierać odpowiednie techniki szacowania testów w różnych jednostkach zależnych i zgodnie z różnymi modelami wytwarzania oprogramowania.

- o Uczestnik uczy się, jak określać przepływ pracy dotyczący defektu w ramach zarządzania defektami, aby dopasować go do sekwencyjnego, zwinnego i hybrydowego modelu wytwarzania oprogramowania.
- Rozdział 3: Zarządzanie zespołem (225 minut)
 - o Uczestnik uczy się, jak analizować dany kontekst projektu w celu określenia wymaganych umiejętności w zespole testowym.
 - o Uczestnik uczy się zarządzać zespołem zgodnie z podejściem „cały zespół”.
 - o Uczestnik uczy się, jak określać uzasadnienie biznesowe działań testowych w projekcie.

Uwaga! W odniesieniu do każdego celu nauczania w aktualnym sylabusie istnieje odpowiednia sekcja z zawartością (np. do LO-1.2.3 istnieje sekcja 1.2.3).

0.11 Jakie są podstawowe założenia tego sylabusu?

Niniejszy sylabus jest przeznaczony dla każdego, kto chce osiągnąć zaawansowany poziom kompetencji w zakresie zarządzania testami, tj. dla osób takich jak kierownicy testów, analitycy testów, inżynierowie testów, konsultanci testów, koordynatorzy testów, liderzy testów i kierownicy projektów. Sylabus jest zgodny z sylabusem Certyfikowany Tester Poziom Podstawowy ISTQB® w. 4.0, który zapewnia podstawową wiedzę i zrozumienie w dziedzinie testowania oprogramowania.

Niniejszy sylabus obejmuje dwie główne role w testowaniu: rolę związaną z zarządzaniem testami i rolę związaną z testowaniem. Rola związana z zarządzaniem testami jest również znana jako rola kierownika testów w kontekście sekwencyjnego modelu wytwarzania, w którym kierownik testów jest zazwyczaj odrębną rolą od kierownika projektu lub właściciela produktu. Rola związana z zarządzaniem testami jest odpowiedzialna za ogólny proces testowy, zespół testowy i zarządzanie testami. Obejmuje to określanie strategii testów, planowanie działań testowych, monitorowanie i kontrolowanie postępów testów, raportowanie wyników testów oraz zarządzanie ryzykiem i problemami związanymi z testami. Rola związana z zarządzaniem testami zapewnia również, aby cele testów były dostosowane do potrzeb biznesu i interesariuszy, a czynności testowe były koordynowane z innymi interesariuszami projektu.

Rola związana z testowaniem obejmuje również ocenę testów, zarządzanie defektami i czynności związane z zamykaniem testów. Rola związana z testowaniem wykorzystuje różne techniki testowania, aby zapewnić jakość i niezawodność testowanych artefaktów i testowanego systemu. Rola związana z testowaniem wykorzystuje również narzędzia testowe i automatyzację w celu wsparcia procesu testowego oraz poprawy wydajności i skuteczności testów. Czynności i zadania przypisane do tych ról mogą się różnić w zależności od kontekstu, takiego jak projekt, produkt, umiejętności i organizacja (patrz sylabus Certyfikowany Tester Poziom Podstawowy ISTQB® w. 4.0).

Termin *członek zespołu testowego* jest używany w niniejszym sylabusie w odniesieniu do każdej osoby pełniącej rolę związaną z zarządzaniem testami lub rolę związaną z testowaniem, która przeprowadza testy, niezależnie od kontekstu organizacyjnego i innych ról. Zespoły testowe składają się z osób o różnych umiejętnościach i kompetencjach. Członkowie zespołu mogą również mieć różne poziomy doświadczenia i certyfikacji, takie jak poziom podstawowy, poziom zaawansowany lub poziom ekspercki. Członkowie zespołu testowego mogą mieć również różne role i obowiązki w zależności od zastosowanego podejścia do testów i modelu procesu testowego, takiego jak testowanie zwinne, testowanie oparte na modelu, testowanie oparte na ryzyku itp.

Ważną kwestią dotyczącą perspektywy, jaką zapewnia sylabus, jest fakt, że koncentruje się on na zarządzaniu testami na poziomie projektu, a nie na zarządzaniu testami na poziomie organizacyjnym.

W związku z tym niniejszy sylabus zawiera informacje, które można wykorzystać do zarządzania testami na poziomie projektu, ale w mniejszym stopniu do zarządzania testami na poziomie organizacyjnym.

Hybrydowy model wytwarzania oprogramowania jest używany w niniejszym sylabusie w odniesieniu do podejścia do wytwarzania oprogramowania, które łączy elementy różnych modeli cyklu wytwarzania oprogramowania, takich jak model V, model iteracyjny, przyrostowy i zwinny. Hybrydowy model wytwarzania oprogramowania ma na celu wykorzystanie mocnych stron i złagodzenie słabych stron każdego modelu, w zależności od kontekstu i potrzeb projektu. Hybrydowy model wytwarzania oprogramowania może np. wykorzystywać model V do początkowych faz planowania i analizy wymagań, a następnie model zwinny do faz projektowania, wytwarzania i testowania. Alternatywnie, hybrydowy model wytwarzania oprogramowania może wykorzystywać model iteracyjny do ogólnego zarządzania projektem, przy jednoczesnym stosowaniu modelu przyrostowego w odniesieniu do każdej iteracji oraz modelu zwinnego dla każdego przyrostu. Hybrydowy model wytwarzania oprogramowania wymaga wysokiego stopnia elastyczności, komunikacji i współpracy między interesariuszami, a także jasnego zrozumienia celów, ryzyk i ograniczeń każdej fazy i modelu.

Zgodnie z niniejszym sylabusem i Słownikiem terminów testowych ISTQB® strategia testów to opis sposobu przeprowadzania testowania w celu osiągnięcia celów testów w danych okolicznościach. Strategia testów określa ogólny zakres, podejście i zasoby do testowania systemu lub produktu. Jest ona zwykle dokumentowana w planie testów lub jako część innych dokumentów, w zależności od kontekstu testów. Na strategię testów ma wpływ organizacyjna strategia testowa, która jest ogólną strategią wysokiego poziomu, opisującą sposób przeprowadzania testów w organizacji. Strategia testów może również istnieć w odniesieniu do pojedynczego poziomu testów lub typu testu, które są konkretnymi aspektami testowania, koncentrującymi się na różnych celach, zadaniach i kryteriach. Ogólny termin „strategia testów” może być używany w dowolnym kontekście (projektu, organizacji, produktu). Podejście do testów to sposób, w jaki realizowane są zadania testowe, w szczególności wybór i kombinacja poziomów testów, typów testów i technik testowania do testowania statycznego i dynamicznego, a także innych praktyk testowych, takich jak testowanie skryptowe, testowanie manualne, testowanie back-to-back itp. Podejście do testów wybrane przez rolę związaną z zarządzaniem testami jest kluczową decyzją w formułowaniu odpowiedniej strategii testów w danym kontekście.

1. Zarządzanie czynnościami testowymi – 750 minut

Słowa kluczowe

analiza ryzyka, cel testów, cykl wytwarzania oprogramowania, hybrydowy model wytwarzania oprogramowania, identyfikacja ryzyka, iteracyjny model wytwarzania, łagodzenie ryzyka, metoda określania celów SMART, monitorowanie ryzyka, monitorowanie testów, nadzór nad testami, ocena ryzyka, plan testów, planowanie testów, poziom ryzyka, poziom testów, prawdopodobieństwo ryzyka, przyrostowy model wytwarzania oprogramowania, retrospektywa, ryzyko jakościowe, ryzyko produktowe, zintegrowany model dojrzałości testów (TMMi), sekwencyjny model wytwarzania oprogramowania, strategia testów, testowanie funkcjonalne, testowanie нефunkcjonalne, testowanie oparte na ryzyku, testowanie w oparciu o doświadczenie, typ testów, TPI NEXT, ukończenie testów, usprawnianie procesu testowego, wpływ ryzyka, zarządzanie ryzykiem

Słowa kluczowe specyficzne dla domeny

Cel Pytanie Metryka (GQM), IDEAL, wskaźnik, miara, metryka

Cele nauczania w rozdziale 1:

1.1 Proces testowy

- TM-1.1.1 (K2) Kandydat podsumowuje planowanie testów.
- TM-1.1.2 (K2) Kandydat podsumowuje monitorowanie testów i nadzór nad testami.
- TM-1.1.3 (K2) Kandydat podsumowuje ukończenie testów.

1.2 Kontekst testowania

- TM-1.2.1 (K2) Kandydat porównuje powody, dla których różni interesariusze są zainteresowani testowaniem.
- TM-1.2.2 (K2) Kandydat wyjaśnia, dlaczego wiedza interesariuszy jest ważna w zarządzaniu testami.
- TM-1.2.3 (K2) Kandydat wyjaśnia testowanie w hybrydowym modelu wytwarzania oprogramowania.
- TM-1.2.4 (K2) Kandydat podsumowuje czynności związane z zarządzaniem testami w różnych cyklach wytwarzania oprogramowania.
- TM-1.2.5 (K2) Kandydat porównuje czynności związane z zarządzaniem testami na różnych poziomach testów.
- TM-1.2.6 (K2) Kandydat porównuje czynności związane z zarządzaniem testami w różnych typach testów.
- TM-1.2.7 (K4) Kandydat analizuje dany projekt i określa czynności związane z zarządzaniem testami, które kładą nacisk na planowanie testów, monitorowanie testów i nadzór nad testami.

1.3 Testowanie oparte na ryzyku

- TM-1.3.1 (K2) Kandydat wyjaśnia różne środki podejmowane w ramach testów opartych na ryzyku w celu reagowania na ryzyko.
- TM-1.3.2 (K2) Kandydat podaje przykłady różnych technik, które kierownik testów może wykorzystać do określenia ryzyk jakościowych produktu.

- TM-1.3.3 (K2) Kandydat podsumowuje czynniki określające poziomy ryzyka związane z jakością produktu.
- TM-1.3.4 (K4) Kandydat wybiera odpowiednie czynności testowe w celu łagodzenia ryzyk zgodnie z poziomem ryzyka w danym kontekście.
- TM-1.3.5 (K2) Kandydat rozróżnia między przykładami ciężkich i lekkich technik testowania opartych na ryzyku.
- TM-1.3.6 (K2) Kandydat podaje przykłady wskaźników sukcesu i trudności związanych z testowaniem opartym na ryzyku.

1.4 Projektowa strategia testowa

- TM-1.4.1 (K2) Kandydat objaśnia typowe wybory dotyczące podejścia do testów.
- TM-1.4.2 (K4) Kandydat analizuje organizacyjną strategię testów i kontekst projektu w celu wyboru odpowiedniego podejścia do testów.
- TM-1.4.3 (K3) Kandydat wykorzystuje metodę określania celów SMART do określenia mierzalnych celów testów i kryteriów wyjścia.

1.5 Usprawnianie procesu testowego

- TM-1.5.1 (K2) Kandydat objaśnia, jak wykorzystać model IDEAL do usprawnienia procesu testowego w danym projekcie.
- TM-1.5.2 (K2) Kandydat podsumowuje podejścia do usprawniania opartego na modelu w celu przetestowania usprawnień procesów i zrozumienia, jak zastosować je w kontekście projektu.
- TM-1.5.3 (K2) Kandydat podsumowuje analityczne podejście do usprawniania procesów i rozumie, jak zastosować je w kontekście projektu.
- TM-1.5.4 (K3) Kandydat wdraża retrospektywy projektu lub iteracji w celu oceny procesów testowych i odkrycia obszarów testowania wymagających poprawy.

1.6 Narzędzia testowe

- TM-1.6.1 (K2) Kandydat podsumowuje najlepsze praktyki w zakresie wprowadzania narzędzi.
- TM-1.6.2 (K2) Kandydat objaśnia wpływ różnych aspektów technicznych i biznesowych przy podejmowaniu decyzji o wyborze odpowiedniego rodzaju narzędzia.
- TM-1.6.3 (K4) Kandydat analizuje daną sytuację w celu opracowania planu wyboru narzędzi, obejmującego ryzyka, koszty i korzyści.
- TM-1.6.4 (K2) Kandydat rozróżnia etapy cyklu życia narzędzi.
- TM-1.6.5 (K2) Kandydat podaje przykłady gromadzenia i oceny metryk za pomocą narzędzi.

1.1 Proces testowy

Wstęp

Sylabus Certyfikowany Tester Poziom Podstawowy ISTQB® w. 4.0 opisuje czynności procesu testowego obejmujące: planowanie testów, monitorowanie testów i nadzór nad testami, analizę testów, projektowanie testów, implementację testów, wykonywanie testów i ukończenie testów.

W sylabusie Poziomu Podstawowego ISTQB® w.4 stwierdzono, że czynności te w procesie testowym są często wdrażane iteracyjnie lub równolegle, w zależności od modelu cyklu wytwarzania oprogramowania (SDLC) i kontekstu projektu. Zazwyczaj wymagane jest dostosowanie tych czynności w kontekście produktu i projektu.

W tym sylabusie skupiono się na następujących kluczowych czynnościach związanych z zarządzaniem testami:

- **Planowanie testów:** określenie celów testów, podejścia do testów, zakresu testów, zasobów testowych, harmonogramu testów, wyników testów i uczestników testów (interesariuszy testów).
- **Monitorowanie testów i nadzór nad testami:** śledzenie postępów testów, wyników testów i odchyłań od wyników testów; podejmowanie działań naprawczych w razie potrzeby; raportowanie statusu testów i wyników odpowiednim interesariuszom.
- **Ukończenie testów:** finalizowanie i archiwizacja artefaktów testowych, ocena procesu testowego i testowanego produktu, określanie działań usprawniających proces testowy i przekazanie informacji o zamknięciu testów odpowiednim interesariuszom.

Norma ISO/IEC/IEEE 29119-2 określa procesy zarządzania testami, które obejmują te czynności związane z zarządzaniem testami. Te procesy zarządzania testami mogą być stosowane na różnych poziomach testowania, takich jak projekt, program lub portfel. Każdy poziom testów może mieć swój własny plan testów, który jest zgodny z planem testów wyższego poziomu.

1.1.1 Czynności związane z planowaniem testów

Ta sekcja koncentruje się na czynnościach związanych z planowaniem testów w różnych zakresach, takich jak cały projekt, poziom testów, rodzaj testów lub wydanie/iteracja/sprint w Agile. W zależności od zakresu planowanie testów może rozpoczynać się i kończyć na różnych etapach procesu wytwarzania. Planowanie testów to czynność, która obejmuje określenie działań i zasobów wymaganych do osiągnięcia celów testów określonych w polityce testów. Planowanie testów powinno zostać zainicjowane na jak najwcześniejszym etapie procesu wytwarzania, najlepiej przed określeniem wymagań, i powinno być aktualizowane w miarę postępu projektu. Planowanie testów jest często procesem iteracyjnym, który wymaga ponownego zaplanowania w trakcie projektu, aby uwzględnić zmiany i informacje zwrotne.

Częścią planowania testów są następujące zadania (podobne do tych, które można znaleźć w normie ISO/IEC/IEEE 29119-2):

- **Zrozumienie kontekstu i organizacja planowania testów**
Zrozumienie kontekstu organizacji (np. polityki testów i organizacyjnej strategii testów), zakresu testów i przedmiotu testów (tj. testowanego produktu) ma kluczowe znaczenie w kontekście planowania testów (patrz podrozdział 1.2). Obejmuje to również wszystkie czynności niezbędne do zorganizowania opracowania planu testów i uzyskania

zatwierdzenia tych działań oraz harmonogramu przez interesariuszy (np. właściciela produktu, kierownika projektu lub kierownika zespołu wytwórczego).

- **Określenie i analiza ryzyk produktowych**

Analiza ryzyka obejmuje określenie i ocenę potencjalnego wpływu i prawdopodobieństwa ryzyka produktowego w ramach planowania testów. Więcej informacji na temat ryzyka produktowego można znaleźć w podrozdziale 1.3 Testowanie oparte na ryzyku.

- **Określenie metod postępowania z ryzykiem**

Na podstawie analizy ryzyka wybiera się i dokumentuje w planie testów odpowiednie metody postępowania z ryzykiem. Mogą one obejmować działania zapobiegawcze, naprawcze lub łagodzące w celu zaradzenia wykrytym ryzykom.

- **Określenie podejścia do testów oraz oszacowanie i przydzielenie zasobów testowych**

W oparciu o organizacyjną strategię testów, normy regulacyjne, wszelkie ograniczenia wynikające z projektu oraz podejścia do postępowania z ryzykiem, określa się podejście do testów do bieżącego zakresu testów (patrz podrozdział 1.4). Po określeniu podejścia do testów ważne jest, aby oszacować wymagane zasoby testowe, takie jak personel testowy, narzędzia testowe, środowiska testowe i dane testowe, oraz przydzielić te zasoby do czynności testowych.

- **Ustalenie planu testów**

Plan testów musi zostać zaakceptowany przez wszystkich interesariuszy, a zatem spory między nimi powinny zostać rozstrzygnięte.

1.1.2 Czynności monitorowania testów i nadzoru nad testami

Aby zarządzanie testami zapewniało skuteczny nadzór nad testami, należy ustanowić harmonogram testów i ramy monitorowania, które umożliwią śledzenie statusu i postępu testów. Ramy te powinny obejmować szczegółowe środki i cele, które są wymagane do powiązania statusu produktów i zasobów prac testowych z planem i celami strategicznymi.

W przypadku małych i mniej złożonych projektów może być stosunkowo łatwo powiązać czynności i produkty prac testowych z planem i celami strategicznymi, ale zazwyczaj należy określić bardziej szczegółowe cele, aby to osiągnąć. Może to obejmować określenie środków i celów, które są wymagane do osiągnięcia celów testów i zakresu podstawy testów.

Szczególnie ważna jest potrzeba powiązania statusu czynności i produktów prac testowych w sposób zrozumiały i istotny dla projektu i interesariuszy biznesowych.

Monitorowanie testów i nadzór nad testami to działania ciągłe.

Nadzór nad testami polega na porównywaniu rzeczywistych postępów z planem testów i wdrażaniu działań naprawczych w razie potrzeby. Kieruje on testami w celu realizacji strategii i celów testów (patrz podrozdział 1.4) i w razie potrzeby powraca do czynności związanych z planowaniem testów. Dane kontrolne wymagają szczegółowych informacji dotyczących planowania testów w celu uzyskania odpowiednich reakcji. Ta czynność obejmuje:

- wdrażanie planu testów i dyrektyw nadzoru;
- zarządzanie odchyleniami od planowanych testów;
- postępowanie z nowo wykrytymi i zmienionymi ryzykami;
- ustalanie gotowości do rozpoczęcia testów;

- udzielanie i uzyskiwanie zgody na ukończenie testów w oparciu o kryteria wyjścia.

Monitorowanie testów obejmuje gromadzenie i rejestrowanie wyników testów, wykrywanie odchyleń od planowanych testów, wykrywanie i analizę nowych ryzyk, które wymagają testowania, oraz monitorowanie zmian w zakresie wykrytych ryzyk.

1.1.3 Czynności związane z ukończeniem testów

Ukończenie testów zwykle następuje podczas kamieni milowych projektu (np. wydanie, zakończenie iteracji lub zakończenie poziomu testów). W przypadku wszelkich nierozwiązanych defektów tworzone są wnioski o zmianę lub pozycje w backlogu produktu. Patrz sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0. Po spełnieniu kryteriów wyjścia kluczowe wyniki powinny zostać zarejestrowane, zarchiwizowane i przekazane odpowiednim interesariuszom. Ukończenie testów wymaga następujących zadań:

- **Utworzenie i zatwierdzanie sumarycznego raportu z testów**
To zadanie pozwala upewnić się, że wszystkie testy zostały zakończone, a wszystkie cele testów zostały osiągnięte. Zadanie to obejmuje zgromadzenie odpowiednich informacji z różnych testaliów, takich jak plany testów, wyniki testów, raporty z postępu testów, raporty z ukończenia testów i raporty o defektach. Zebrane informacje są oceniane i podsumowywane w sumarycznym raporcie z testów. Sumaryczny raport z testów jest zatwierdzany i przekazywany odpowiednim interesariuszom.
- **Zarchiwizowanie testaliów**
W ramach tego zadania określa się testalia, które mogą być przydatne w przyszłości lub co do których oczekuje się, że będą ponownie używane, co zwykle wiąże się z przypadkami testowymi. Dzięki temu testalia te są łatwo dostępne i zrozumiałe w przypadku potrzeby ponownego użycia w przyszłości. Ponadto wyniki testów, dzienniki testów, raporty z testów i inne testalia powinny być tymczasowo archiwizowane w systemie zarządzania konfiguracją.
- **Przekazanie testaliów**
Zadanie to dostarcza wartościowych produktów pracy osobom, które ich potrzebują. Np. znane odroczone lub zaakceptowane defekty powinny być zgłaszane osobom, które będą używać testaliów lub udzielać wsparcia w odniesieniu do nich.
- **Wykonanie wszystkich niezbędnych zadań w celu wyczyszczenia środowiska testowego i przywrócenia go do wstępnie określonego stanu**
To zadanie pozwala zapewnić, że środowisko testowe będzie gotowe do następnego cyklu testowego lub projektu. Polega ono na usunięciu ze środowiska testowego wszelkich danych testowych, narzędzi testowych, sterowników, zaślepek i skryptów testowych. Obejmuje również przywrócenie środowiska testowego do jego pierwotnego lub pożądanego stanu.
- **Przeprowadzenie/zebranie/udokumentowanie wyciągniętych wniosków**
To zadanie jest realizowane w ramach retrospektyw, podczas których omawiane i dokumentowane są ważne wnioski wyciągnięte podczas procesu testowego. Może to obejmować ustalenia dotyczące całego cyklu wytwarzania oprogramowania (SDLC). Wyciągnięte wnioski można wykorzystać do ulepszenia procesu testowego, jak opisano w podrozdziale 1.5 Usprawnianie procesu testowego.

1.2 Kontekst testowania

Wstęp

Kontekst testowania obejmuje unikalne warunki i ograniczenia, które wpływają na proces testowy, kształtowanie decyzji i strategii planowania, projektowania i wykonywania testów. Ważne jest, aby kierownicy testów rozumieli ten kontekst w celu dostosowania testów do konkretnych potrzeb i celów projektu wytwarzania oprogramowania. Kontekst ten może się różnić w zależności od rodzaju produktu, branży, wymagań regulacyjnych, a co najważniejsze, od zastosowanego cyklu wytwarzania oprogramowania (SDLC).

Kierownicy testów mają za zadanie stosować ustalone strategie testów i dobierać techniki testowania, a nie rozwijać je. Odgrywają oni kluczową rolę w formułowaniu planów testów, które są dostosowane do kontekstu projektu. Poprzez zrozumienie i rozważenie takich różnych czynników kierownicy testów mogą zapewnić, aby testy były adekwatne, skuteczne i wydajne w osiąganiu celów testów.

1.2.1 Interesariusze testów

Interesariusze testów to osoby lub grupy bezpośrednio lub pośrednio zainteresowane jakością produktu. Poniżej znajduje się typowa lista potencjalnych interesariuszy, zmieniona w celu odzwierciedlenia ich różnorodnych interesów związanych z testowaniem:

- **programiści, liderzy ds. rozwoju i kierownicy ds. rozwoju:** oprócz wdrażania systemu podlegającego testowaniu i prowadzenia działań na podstawie wyników testów, interesariusze ci są również zaangażowani w testowanie modułowe i wnoszą wkład w proces testowy;
- **testerzy, liderzy testów i kierownicy testów:** osoby te przygotowują testalia, co obejmuje opracowywanie planów testów oraz udział w procesie testowym poprzez czynności, takie jak analiza wymagań, projektowanie testów, wykonywanie testów, śledzenie i raportowanie defektów, automatyzacja testów i raportowanie o postępach testów;
- **kierownicy projektów, właściciele produktów i użytkownicy biznesowi:** określają wymagania i wymagany poziom jakości oraz rekomendują wymagany zakres na podstawie postrzeganych ryzyk; dokonują również przeglądu produktów pracy, uczestniczą w testowaniu akceptacyjnym przez użytkownika (UAT) i podejmują decyzje na podstawie wyników testów;
- **zespół operacyjny:** jest zaangażowany w operacyjne testy akceptacyjne, zapewnia gotowość systemu do produkcji i wnosi wkład w określanie wymagań нефункциональных;
- **klienci i użytkownicy:** klienci kupują produkt, a użytkownicy bezpośrednio z niego korzystają; obie te grupy mają kluczowe znaczenie w określaniu wymagań i powinny być zaangażowane w testy akceptacyjne użytkowników (UAT), aby zweryfikować, czy produkt odpowiada ich potrzebom.

Lista ta nie obejmuje wszystkich potencjalnych interesariuszy. Kierownicy testów muszą przeprowadzić analizę interesariuszy w ramach tworzenia strategii testów i planu testów, biorąc pod uwagę zakres testów w celu określenia konkretnych interesariuszy ich projektu.

1.2.2 Znaczenie wiedzy interesariuszy w zarządzaniu testami

W zarządzaniu testami kluczowe jest uwzględnienie perspektyw i wpływu różnych interesariuszy. Macierz interesariuszy, często określana jako macierz siła-interes, pomaga kierownikom testów

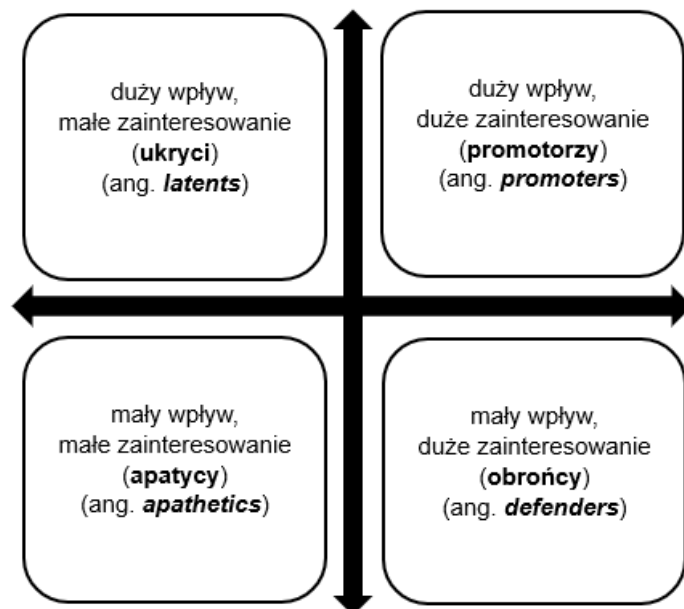
w ustalaniu priorytetów zaangażowania interesariuszy i skutecznym zarządzaniu oczekiwaniami. Macierz interesariuszy jest strategicznym narzędziem w zarządzaniu testami, które:

- wykorzystuje wiedzę interesariuszy, a użytkownicy końcowi i zespoły techniczne dostarczają informacji zwrotnych i spostrzeżeń na temat wydajności i zabezpieczeń;
- wspiera zarządzanie ryzykiem poprzez podkreślanie interesów i wpływów interesariuszy, zachęcając do proaktywnych działań łagodzących;
- ceni różne punkty widzenia dzięki cennym informacjom zwrotnym.

Macierz interesariuszy składa się z czterech ćwiartek:

- **Promotorzy (duży wpływ, duże zainteresowanie):** kluczowi współpracownicy o wysokim wpływie i zainteresowaniu, niezbędni do kształtowania strategii i planu testów.
- **Ukryci (duży wpływ, małe zainteresowanie):** mimo że mogą oni nie być mocno zainteresowani codziennymi zadaniami, ich decyzje mają kluczowe znaczenie dla alokacji zasobów i kierunku projektu na poziomie ogólnym.
- **Obrońcy (mały wpływ, duże zainteresowanie):** często dostarczają wysokiej jakości informacji zwrotnych i mogą być zaangażowani poprzez regularne aktualizacje i udział w konkretnych dyskusjach.
- **Apatycy (mały wpływ, małe zainteresowanie):** mimo że nie są ściśle zaangażowani, informowanie ich o istotnych kamieniach milowych i zachęcanie ich do wnoszenia wkładu w poszczególne zagadnienia może przynieść unikalne spostrzeżenia.

Rola kierownika testów obejmuje sporządzenie szczegółowej listy interesariuszy i zrozumienie powiązań każdego z nich z czynnościami testowymi, przy użyciu macierzy interesariuszy w celu zwiększenia skuteczności praktyk zarządzania testami.



Rys. 1. Różne typy interesariuszy

1.2.3 Zarządzanie testami w hybrydowym modelu wytwarzania oprogramowania

Hybrydowe modele wytwarzania oprogramowania integrują elementy zarówno tradycyjnego podejścia sekwencyjnego, jak i praktyk zwinnych, aby dostosować je do konkretnych potrzeb projektu lub zmian

organizacyjnych. Poniżej przedstawiono typowe powody korzystania z hybrydowego modelu wytwarzania oprogramowania, mimo że – w zależności od organizacji i projektu – mogą istnieć również inne powody:

- **Model hybrydowy jako przejście na model zwinny:** przejście z metodyki tradycyjnej na zwinną może być trudne ze względu na fundamentalne zmiany w przepływie pracy, kulturze i dynamice zespołu; modele hybrydowe zapewniają zrównoważone podejście, które ułatwia to przejście, łącząc strukturę metod tradycyjnych z elastycznością praktyk zwinnych.
- **Model hybrydowy jako rozwiązanie dopasowane do zamierzonego celu:** niektóre organizacje lub projekty mogą nie być w stanie przejść na metodyki zwinne; projekty, które są obciążone wysokim ryzykiem, mogą wymagać sekwencyjnych zadań w przypadku niektórych elementów i praktyk zwinnych w odniesieniu do innych; mogą korzystać z modelu hybrydowego, który odpowiada ich celom.

W środowisku hybrydowym czynności związane z zarządzaniem testami mogą obejmować:

- ocenę zrozumienia i zdolności zespołu do płynnego przejścia między metodykami tradycyjnymi i zwinnymi;
- określenie mocnych i słabych stron w dostosowaniu się do podejścia hybrydowego;
- zapewnienie, że zespół jest biegły w łączeniu ustrukturyzowanych procesów z elastycznością podejścia zwinnego;
- wzmocnienie współpracy między zespołem testowym a interesariuszami w celu lepszego zarządzania testami w ramach sprintów i tradycyjnych faz testowych;
- udział w skoordynowanych wysiłkach, takich jak scrum-of-scrums dla testerów, aby utrzymać koncentrację na testowaniu, jednocześnie przyczyniając się do realizacji ogólnych celów w zakresie wytwarzania;
- śledzenie i przeglądanie działań i przypadków testowych w ramach sprintów, aby upewnić się, że są one zgodne z praktykami zwinnymi.

Więcej informacji można znaleźć w (Fowler, 2010).

1.2.4 Czynności związane z zarządzaniem testami w różnych modelach cyklu wytwarzania oprogramowania

Aby prawidłowo dostosować testy do modelu SDLC, kierownik testów musi zrozumieć różne modele SDLC używane w jego organizacji i wykorzystać tę wiedzę do prawidłowego dostosowania testów do działań twórczych.

W poniższej tabeli przedstawiono porównanie różnych czynności związanych z zarządzaniem testami w oparciu o różne modele SDLC:

Aspekt	Sekwencyjny model wytwarzania, np. model V	Iteracyjny model wytwarzania, np. SCRUM
Szacowanie	Wczesne szczegółowe szacowanie w odniesieniu do każdego poziomu testów.	Szacowanie iteracyjne, planowanie historyjek użytkownika w ramach planowania iteracji.
Testalia	Obejmują strategię, plan, przypadki testowe, harmonogram i raporty.	Skupiają się na kryteriach akceptacji i definicji ukończenia; minimalna dokumentacja.

Aspekt	Sekwencyjny model wytwarzania, np. model V	Iteracyjny model wytwarzania, np. SCRUM
Role	Kierownik testów nadzoruje decyzje i zarządzanie zespołem.	Role są zintegrowane; facylitator lub coach zastępuje tradycyjnego kierownika testów.
Narzędzia	Głównie narzędzia do zarządzania testami dostosowane do testowania w danej fazie.	Narzędzia do CI/CD i automatyzacji są kluczowe, ponieważ wspierają ciągłe testowanie.
Podejście do testów	Zaplanowane z wyprzedzeniem, odpowiadające fazom projektu.	Osadzone w iteracjach, z naciskiem na zdolność adaptacji i informacje zwrotne.
Automatyzacja testów	Wdrażana strategicznie, może występować na różnych etapach.	Wbudowana od samego początku, z naciskiem na zautomatyzowaną regresję w CI/CD.
Monitorowanie i tworzenie raportów	Raportowanie oparte na kamieniach milowych, z opcjonalnymi zautomatyzowanymi tablicami wskaźników.	Ciągłe raportowanie z tablicami wskaźników w czasie rzeczywistym i codziennymi aktualizacjami statusu.
Metryki	Koncentrują się na tradycyjnych metrykach testowych i zarządzaniu defektami (np. przeprowadzanie testów, wskaźniki defektów).	Obejmują zwinne metryki do śledzenia iteracji – oprócz tradycyjnych metryk (np. prędkość zespołu, wykresy spalania).

Tabela 1: Czynności związane z zarządzaniem testami w różnych modelach SDLC

1.2.5 Czynności związane z zarządzaniem testami na różnych poziomach testów

Testowanie modułowe:

- określenie zakresu, celów i kryteriów ukończenia testów modułowych (testów jednostkowych);
- angażowanie testerów w czynności wykraczające poza tradycyjne role związane z testowaniem, takie jak przeglądy kodu, w których ich umiejętności analityczne stanowią wartość dodaną;
- koordynacja z zespołem programistów w zakresie rozwiązywania problemów i wkładu w testy modułowe.

Testowanie integracji modułów:

- określenie sekwencji integracji i kombinacji testów we współpracy z zespołem wytwórczym, z uwzględnieniem modelu SDLC, narzędzi i procesów;
- nadzorowanie postępów w celu zapewnienia zgodności ze strategiami testowania systemowego i akceptacyjnego;
- zarządzanie tą fazą we współpracy z programistami, z uwzględnieniem testowania integracji modułów.

Testowanie integracji systemów:

- upewnienie się, że zakres i cele testowania integracji systemów są jasne i dostosowane do oceny ryzyka i celów jakościowych;
- prowadzenie nadzoru nad postępami, wynikami i zarządzaniem problemami podczas testowania integracji systemów.

Testowanie systemowe:

- dostosowanie planowania do modelu SDLC, ze staranną alokacją zasobów, wyborem narzędzi i harmonogramowaniem:
 - o w przypadku projektów zwinnych należy zintegrować testowanie systemowe z iteracyjnym testowaniem historyjek, unikając odrębnych faz testowania i zapewniając ciągłość i integrację testowania; w modelach sekwencyjnych testowanie może przebiegać zgodnie z zaplanowanymi etapami.

Testowanie akceptacyjne:

- współpraca z interesariuszami w celu przeglądu i potwierdzenia spełnienia kryteriów akceptacji i planowania czynności testowych, w tym zarządzania testami akceptacyjnymi użytkownika (ang. UAT – *User Acceptance Tests*);
- koordynowanie logistyki testowania akceptacyjnego, ułatwianie przeprowadzenia testów u klienta w celu upewnienia się, że produkt odpowiada potrzebom biznesowym i standardom jakości poza środowiskiem programistycznym;
- ułatwianie rozwiązywania wszelkich problemów związanych z UAT i prowadzenie interesariuszy przez proces zatwierdzania produktu po spełnieniu kryteriów akceptacji.

1.2.6 Czynności związane z zarządzaniem testami w różnych typach testów

Skuteczne zarządzanie testami wymaga zintegrowanego podejścia, które uwzględnia unikalne wymagania testowania funkcjonalnego, niefunkcjonalnego, czarnoskrzynkowego i białoskrzynkowego. Kierownicy przeprowadzający testy funkcjonalne koncentrują się na zapewnieniu, że wszystkie funkcje będą dokładnie przetestowane i spełnią określone wymagania. Zarządzanie testami niefunkcjonalnymi koncentruje się na weryfikacji atrybutów systemu, takich jak wydajność i zabezpieczenia. Zarządzanie testami czarnoskrzynkowymi polega na zapewnieniu, że testy koncentrują się na użytkowniku i pokrywają wszystkie możliwe interakcje zewnętrzne. Zarządzanie testami białoskrzynkowymi kładzie nacisk na zrozumienie struktury kodu i zapewnienie, że testy dokładnie pokrywają wewnętrzną logikę.

Zarządzanie testami funkcjonalnymi:

- planowanie strategiczne i śledzenie postępów: tworzenie szczegółowej strategii testów, która jest zgodna z wymaganiami funkcjonalnymi i celami projektu, a także monitorowanie postępów;
- koordynacja zasobów: efektywne przydzielanie zasobów ludzkich i technicznych w celu pokrycia wszystkich aspektów funkcjonalnych systemu.

Zarządzanie testami niefunkcjonalnymi:

- testy porównawcze wydajności: ustanawianie wzorców wydajności i zarządzanie czynnościami testowymi, które oceniają system pod kątem tych kryteriów;
- weryfikacja zgodności: nadzorowanie testów, które zapewniają, że system spełnia standardy niefunkcjonalne, takie jak zabezpieczenia, użyteczność i niezawodność.

Zarządzanie testami czarnoskrzynkowymi:

- analiza pokrycia testami: zapewnienie, że testy czarnoskrzynkowe pokrywają wszystkie scenariusze użytkownika i wymagania biznesowe;
- włączenie informacji zwrotnych: zarządzanie procesem gromadzenia informacji zwrotnych od interesariuszy w celu udoskonalenia podejść do testowania czarnoskrzynkowego i usuwania defektów.

Zarządzanie testami białoskrzynkowymi:

- optymalizacja pokrycia kodu: nadzorowanie korzystania z narzędzi pokrycia kodu w celu wykrywania luk w testach białoskrzynkowych i kierowanie zasobami w celu wyeliminowania tych obszarów;
- integracja ze spostrzeżeniami technicznymi: zarządzanie włączaniem spostrzeżeń technicznych do procesu planowania testów, zapewniając, aby testy były projektowane ze zrozumieniem wewnętrznego działania aplikacji.

1.2.7 Czynności związane z zarządzaniem testami w celu planowania, monitorowania i nadzoru nad testami

Skuteczne zarządzanie testami jest podstawą wszelkich udanych działań testowych. Obejmuje szeroki zakres czynności, które wymagają starannego planowania, czujnego monitorowania i strategicznego nadzoru. Kierownicy testów odgrywają kluczową rolę w zapewnieniu, aby proces testowy był nie tylko skuteczny i wydajny, ale także dostosowany do unikalnych wymagań danego projektu.

Planowanie testów:

- **Kompleksowe określanie zakresu:** plan testów musi być skrupulatnie opracowany i musi zawierać dokładną definicję zakresu; obejmuje to określenie wszystkich wymagań funkcjonalnych i niefunkcjonalnych w celu zapewnienia pełnego pokrycia testami; obejmuje to również rozważenie implikacji zarówno metodyki testowania czarnoskrzynkowego, jak i białoskrzynkowego, zapewniając, aby opracowane przypadki testowe były w stanie zwalidować system podlegający testowaniu pod każdym kątem.
- **Ocena ryzyka i plan łagodzenia ryzyka:** integralną częścią planu testów są solidne ramy zarządzania ryzykiem; kierownicy testów muszą przeprowadzić szczegółową analizę ryzyka, wskazując potencjalne luki w zabezpieczeniach i wyzwania, które mogą mieć wpływ zarówno na przepływ pracy w projekcie, jak i na produkt końcowy; kluczowe znaczenie ma opracowanie strategii łagodzących, co obejmuje planowanie zapobiegawcze w celu skutecznego obejścia lub zminimalizowania tych ryzyk.
- **Strategia alokacji zasobów:** kolejnym kluczowym elementem jest planowanie zasobów; wykracza to poza zwykłą alokację i obejmuje określenie struktury zespołu, wyznaczanie ról i ustanawianie protokołów komunikacji; w środowiskach, w których zespoły są rozproszone, jak w przypadku modeli na miejscu/poza miejscem pracy, staje się to szczególnie istotne z uwagi na utrzymanie synchronizacji i zapewnienie płynnej współpracy.

Monitorowanie testów:

- **Nadzór nad realizacją:** monitorowanie odgrywa kluczową rolę w procesie zarządzania testami; wiąże się to z ciągłym przeglądem wykonania testów w stosunku do ustalonego planu, śledzeniem postępu przypadków testowych i zarządzaniem wszelkimi pojawiającymi się defektami; dostosowanie priorytetów testów w oparciu o oceny ryzyka i wytwarzanie w czasie rzeczywistym zapewnia, że testy pozostaną skoncentrowane i dostosowane do najbardziej krytycznych obszarów.

- **Optymalizacja narzędzi i środowiska:** rozsądny wybór i wykorzystanie narzędzi testowych i środowisk ma kluczowe znaczenie dla wspierania strategii testów; ciągłe monitorowanie zapewnia ich skuteczną integrację z potokiem dostarczania CI/CD, ułatwiając ciągłe testowanie i natychmiastowe przekazywanie informacji zwrotnych, które mają kluczowe znaczenie dla procesu zwinnego wytwarzania oprogramowania.
- **Współpraca w zakresie rozwoju:** utrzymywanie ścisłej współpracy z zespołem wytwórczym jest niezbędne do uzyskania pomyślnych wyników testów; współpraca ta powinna wspierać kompleksowe podejście do testowania, wykorzystując spostrzeżenia zarówno z perspektywy białoskrzynkowej, jak i czarnoskrzynkowej, aby zapobiegać potencjalnym problemom.

Nadzór nad testami:

- **Adaptacyjne zarządzanie procesami:** nadzór nad testami polega na dynamicznym dostosowywaniu procesu testowego w odpowiedzi na nowe spostrzeżenia, wyzwania i ewoluującą dynamikę projektu; wymaga to od kierownika testów elastyczności i zdolności do wprowadzania zmian w podejściu do testowania, które będą odzwierciedlać aktualny stan projektu.
- **Zarządzanie bramkami jakości:** ustrukturyzowane podejście do zarządzania bramkami jakości ma fundamentalne znaczenie; obejmuje to określenie, co stanowi bramkę jakości w procesie testowym, i podejmowanie świadomych decyzji dotyczących postępów w fazie testowania, co ma zasadnicze znaczenie dla utrzymania integralności produktu.

Koncentrując się na tych konkretnych czynnościach w ramach planowania testów, monitorowania testów i nadzoru nad testami, kierownicy testów mogą zapewnić, aby proces testowy był dobrze zdefiniowany, dzięki czemu będzie można go dostosowywać do zmian w projekcie, a jego wynikiem będzie produkt spełniający zarówno wymagania projektu, jak i oczekiwania interesariuszy.

1.3 Testowanie oparte na ryzyku

Wstęp

Testowanie oparte na ryzyku obejmuje identyfikację, ocenę, monitorowanie i łagodzenie ryzyka w celu wspierania testów. Ryzyka te są określane przez różnych interesariuszy i mogą być wykorzystywane do wyboru i priorytetyzacji testów. Im wyższy poziom ryzyka, tym wcześniej powinny rozpocząć się testy, a nakład pracy związany z testowaniem powinien być bardziej intensywny i długotrwały.

1.3.1 Testowanie jako działanie łagodzące ryzyko

Ryzyko produktowe to potencjalna sytuacja, w której w produkcie mogą występować problemy z jakością. Gdy testy ujawniają defekty, testowanie pomaga złagodzić ryzyko produktowe, zapewniając świadomość defektów i możliwości ich usunięcia przed wydaniem. Gdy testy nie wykrywają defektów, testowanie wskazuje, że poziom ryzyka produktowego jest niższy, niż oczekiwano.

Kierownik testów jest odpowiedzialny między innymi za przekazanie prawidłowej i rzetelnej oceny jakości produktu. Wymaga to aktywnego zaangażowania w zarządzanie ryzykiem projektowym z naciskiem na ryzyka projektowe związane z zapewnieniem jakości (np. niejednoznaczne wymagania, które prowadzą do poważnych problemów z późną walidacją, niewystarczające środowiska testowe utrudniające wykonywanie testów).

Testowanie oparte na ryzyku koncentruje się na ryzykach związanych z jakością. Jest ono zgodne z ogólnym procesem zarządzania ryzykiem, na który składają się następujące główne czynności:

- analiza ryzyka, składająca się z identyfikacji i oceny ryzyka;
- kontrola ryzyka, polegająca na monitorowaniu ryzyka i łagodzeniu ryzyka.

Te główne czynności są logicznie uporządkowane w porządku sekwencyjnym, ale mogą się pokrywać.

Aby skoncentrować testy na ryzykach jakościowych, należy zidentyfikować i ocenić ryzyka jakościowe. Aby analiza ryzyka była najskuteczniejsza, powinna obejmować różnych interesariuszy. Jako główny interesariusz analizy ryzyka jakościowego, kierownik testów powinien rozumieć i monitorować te działania oraz być w stanie je moderować. Monitorowanie testów powinno obejmować monitorowanie ryzyka. Oprócz monitorowania ewolucji znanych ryzyk jakościowych, powinno to obejmować analizę wszelkich nowych ryzyk jakościowych i dostosowanie rejestru ryzyk.

Kierownik testów jest jedną z kilku osób, które kierują ograniczaniem ryzyka jakościowego. Łagodzenie ryzyka jest rozdzielone na kilka czynności testowych. Na przykład wyniki analizy ryzyka jakościowego są wykorzystywane w planowaniu testów w celu skoncentrowania testów na właściwych obszarach przy użyciu odpowiednich technik. W analizie testów poziomy ryzyka kierują wyborem warunków testowych, które mają zostać uwzględnione. Podczas wykonywania testów kolejność wykonywania testów jest regulowana przez priorytetyzację opartą na ryzyku.

1.3.2 Identyfikacja ryzyk jakościowych

Zadaniem kierownika testów jest zebranie informacji o ryzykach od interesariuszy. Interesariusze mogą określać ryzyka jakościowe za pomocą co najmniej jednej z następujących technik:

- wywiady z ekspertami;
- niezależne oceny;
- retrospektywy;
- warsztaty dotyczące ryzyka;

- burze mózgów;
- listy kontrolne;
- odwoływanie się do wcześniejszych doświadczeń.

Poprzez zaangażowanie jak najszerszej próbki interesariuszy, proces identyfikacji ryzyka często identyfikuje większość znaczących ryzyk produktowych. Określenie, którzy interesariusze mają uczestniczyć na tym etapie, jest bardzo ważne. Istotne jest zapewnienie, że lista uczestniczących interesariuszy jest wyczerpująca i uzgodniona z kierownikiem projektu. Identyfikacja ryzyka, w której brakuje kluczowych interesariuszy, może być bardzo problematyczna. Istotne jest zapewnienie wszystkim interesariuszom możliwości uczestnictwa. Jeśli interesariusze nie mogą uczestniczyć w identyfikacji ryzyka, powinni przynajmniej mieć możliwość delegowania zadania. Jeśli kluczowi interesariusze mogą nie być reprezentowani, można wykorzystać spotkanie rozpoczynające projekt, aby ustalić, czy ich brakuje.

W testowaniu opartym na ryzyku ważne jest, aby zrozumieć, że ryzyko nie jest równomiernie rozłożone w przedmiotach testów. Na przykład, moduły aplikacji samoobsługowej skierowane do klienta mogą mieć zupełnie inne ryzyka związane z użytecznością niż moduły administracyjne. Identyfikacja poszczególnych ryzyk związanych z różnymi elementami testowymi jest ważnym zadaniem w planowaniu testów.

Identyfikacja ryzyka często powoduje powstawanie produktów ubocznych (tj. wykrywanie problemów, które nie są ryzykami produktowymi). Przykłady obejmują ogólne pytania lub problemy dotyczące produktu lub projektu albo problemy w przywołanych dokumentach, takich jak wymagania i specyfikacje projektowe. Ryzyka projektowe są również często identyfikowane jako produkt uboczny identyfikacji ryzyka jakościowego, ale nie są przedmiotem testów opartych na ryzyku. Kierownik testów może często odgrywać ważną rolę w podkreślaniu tych produktów ubocznych i wyjaśnianiu, że jakość jest sprawą ważną dla wszystkich. Słabe lub brakujące wymagania są często oznaką bardziej fundamentalnego problemu w planowaniu i przygotowaniu, ponieważ zapewnienie jakości jest związane z całym SDLC.

1.3.3 Ocena ryzyka jakościowego

Po określeniu ryzyk można je ocenić. Ocena ryzyka jakościowego obejmuje kategoryzację ryzyk według typu (ryzyko produktowe lub ryzyko projektowe) oraz według charakterystyk jakościowych, na które ma ono wpływ.

Określenie poziomu ryzyka zwykle obejmuje ocenę, dla każdej pozycji ryzyka, prawdopodobieństwa wystąpienia ryzyka i wpływu ryzyka na wystąpienie. Czynniki wpływające na prawdopodobieństwo ryzyka w przypadku ryzyk jakościowych obejmują:

- złożoność technologii, narzędzi lub architektury systemu;
- dojrzałość organizacji;
- problemy kadrowe z umiejętnościami, dostępnością, motywacją lub autonomiczną pracą, w tym znajomością SDLC w użyciu;
- konflikt w zespole;
- problemy kontraktowe z dostawcami;
- geograficzne rozproszenie zespołów;
- słabe przywództwo menedżerskie lub techniczne;
- presja czasu, zasobów, budżetu i zarządzania;
- brak wczesnych działań w zakresie zapewnienia jakości;
- wysokie wskaźniki zmian podstawy testów, produktu lub personelu.

Czynniki oddziałujące na wpływ ryzyk:

- częstotliwość korzystania z danej funkcji;
- krytyczność dotkniętej ryzykiem funkcji;
- krytyczność dotkniętego ryzykiem celu biznesowego;
- szkody na reputacji;
- utrata dochodów z działalności;
- potencjalne straty finansowe, ekologiczne lub społeczne lub odpowiedzialność;
- sankcje cywilne lub karne;
- problemy z interfejsami i integracją;
- brak rozsądnych rozwiązań zastępczych;
- potrzeby bezpieczeństwa.

Kierownik testów łączy prawdopodobieństwo ryzyka i wpływ ryzyka w celu określenia poziomu ryzyka.

Jeśli analiza ryzyka opiera się na obszernych i statystycznie ważnych danych dotyczących ryzyka, właściwa jest ocena ilościowa. Na przykład prawdopodobieństwo ryzyka można wyrazić jako procent, a wpływ ryzyka – jako liczbę. W takim przypadku poziom ryzyka można obliczyć jako iloczyn tych dwóch czynników. Zazwyczaj jednak prawdopodobieństwo i wpływ ryzyka można ustalić tylko jakościowo w skalach porządkowych, np. jako bardzo wysokie, wysokie, średnie, niskie lub bardzo niskie. Prawdopodobieństwo ryzyka i wartości wpływu ryzyka są następnie łączone za pomocą macierzy ryzyka w celu utworzenia zagregowanego poziomu ryzyka. Ten zagregowany poziom ryzyka należy interpretować jako jakościową, względną ocenę w skali porządkowej.

O ile analiza ryzyka nie opiera się na obszernych i statystycznie ważnych danych dotyczących ryzyka, analiza ryzyka będzie miała charakter jakościowy, w oparciu o subiektywne postrzeganie prawdopodobieństwa ryzyka i wpływu ryzyka przez interesariuszy.

1.3.4 Łagodzenie ryzyka jakościowego poprzez odpowiednie testy

W procesie wytwarzania oprogramowania testowanie jest najważniejszą czynnością, która ogranicza ryzyko jakościowe i umożliwia zmniejszenie prawdopodobieństwa awarii. Inne możliwe środki łagodzące ryzyko obejmują plan awaryjny (np. poprzez zapewnienie rozwiązań zastępczych), przeniesienie ryzyka na stronę trzecią (np. dostawcę modułu) lub akceptację ryzyka.

W **planowaniu testów** czas i nakład pracy związany z opracowaniem i wykonaniem testu powinien być proporcjonalny do poziomu ryzyka: testowanie pod kątem wyższych poziomów ryzyka powinno rozpocząć się wcześniej i stosować bardziej rygorystyczne techniki testowania, podczas gdy testowanie pod kątem niższych poziomów ryzyka może rozpocząć się później i powinno stosować mniej rygorystyczne techniki testowania. Aby jak najlepiej złagodzić ogólne ryzyko poprzez testy, kierownik testów powinien przeanalizować następujące czynniki kontekstowe i wybrać odpowiednie podejście do testów:

- **Elementy testowe:** różne elementy testowe w przedmiocie testów mogą mieć różne poziomy tego samego typu ryzyka, dlatego przedmiot testów nie musi być testowany z jednolitą dokładnością.
- **Charakterystyki jakościowe:** ryzyka wpływające na określone charakterystyki jakościowe należy złagodzić za pomocą powiązanych typów testów, które wymagają określonego nakładu pracy związanego z testowaniem, środowisk testowych i umiejętności testowania.

- **Poziomy i typy testów:** niektóre ryzyka mogą być testowane dynamicznie tylko na określonych poziomach testów; inne poprzez testy statyczne (np. analizę statyczną i przeglądy kodu pod kątem utrzymywalności) lub poprzez połączenie obu (np. przegląd architektury i dynamiczne testowanie zintegrowanego systemu pod kątem luk w zabezpieczeniach); testowanie każdego elementu testowego tak wcześnie, jak to możliwe, zmniejsza ryzyko znalezienia krytycznych defektów na późniejszym etapie cyklu życia, co spowodowałoby wyższe koszty i opóźnienia związane z awariami wewnętrznymi.
- **SDLC:** czynności testowe mają swoje własne, specyficzne kryteria wejścia; różne SDLC spełniają je w różnym czasie.
- **Zespół testowy:** najbardziej wykwalifikowane osoby powinny testować elementy testowe o najwyższym poziomie ryzyka.
- **Wymogi regulacyjne:** niektóre normy związane z bezpieczeństwem (np. norma IEC 61508) określają techniki testowe i wymagane pokrycie w oparciu o poziom integralności; kierownik testów musi dopilnować, aby normy te były przestrzegane.

Ponadto poziom ryzyka powinien wpływać na decyzje dotyczące kontroli jakości, takie jak wykorzystanie przeglądów produktów pracy, takich jak przypadki testowe, poziom niezależności testów od wytwarzania oraz zakres przeprowadzonych testów regresji.

Podczas **monitorowania testów i nadzoru nad testami** testowanie oparte na ryzyku umożliwia raportowanie postępów testów pod względem poziomu ryzyka rezydualnego (reszkowego) w dowolnym momencie. Wspiera to zespół wytwórczy i interesariuszy w monitorowaniu i nadzorze nad wytwarzaniem oprogramowania, w tym w podejmowaniu decyzji o wydaniu, w oparciu o poziom ryzyka reszkowego. Wymaga to raportowania wyników testów pod kątem ryzyka w sposób zrozumiały dla interesariuszy.

Podczas **implementacji testów** priorytetyzacja testów jest oparta na poziomach ryzyka. Podczas wykonywania testów zapewnia to wczesne pokrycie najbardziej krytycznych obszarów i łagodzenie ryzyk na najwyższym poziomie.

- W niektórych przypadkach testy są priorytetyzowane do wykonania w ścisłej malejącej kolejności poziomów ryzyka, które obejmują, zaczynając od najwyższego. Podejście to nazywane jest podejściem w głąb i jest odpowiednie, gdy ważne jest jak najwcześniejsze łagodzenie ryzyk najwyższego poziomu.
- Alternatywnie co najmniej jeden test dla każdego ryzyka ma najwyższy priorytet. Wszystkie inne testy są priorytetyzowane w oparciu o ich poziomy ryzyka. Podejście to nazywane jest podejściem wszerz i jest odpowiednie, gdy interesariusze chcą jak najwcześniej uzyskać ogólny obraz jakości produktu. W praktyce testowanie często rozpoczyna się od podejścia w głąb, ale w miarę jak czas staje się bardziej ograniczony, przechodzi się do podejścia wszerz, testując wszystkie pozostałe elementy ryzyka co najmniej raz.

Niezależnie od tego, czy testowanie oparte na ryzyku odbywa się w głąb, wszerz, czy poprzez połączenie tych podejść, czas przeznaczony na testowanie może zostać wykorzystany bez przeprowadzania wszystkich zaplanowanych testów. Na tym etapie testowanie oparte na ryzyku ułatwia dostarczenie uzasadnionego zalecenia kierownictwu, czy rozszerzyć testowanie, czy zaakceptować pozostałe ryzyko.

1.3.5 Techniki testowania oparte na ryzyku

Istnieje wiele konkretnych technik o różnym stopniu sformalizowania w celu wdrożenia testowania opartego na ryzyku. Przydatność techniki zależy od projektu, procesu i rozważań nad produktem. Istnieją dwa podstawowe rodzaje technik: ciężka lub lekka. W systemach o krytycznym znaczeniu ze

w. 3.0 Strona 31 z 90 14.10.2024 r.

względów bezpieczeństwa bardzo często stosuje się techniki ciężkie. W aplikacjach niekrytycznych ze względów bezpieczeństwa zwykle stosuje się techniki lekkie.

Techniki ciężkie są formalne, wykorzystują określone procedury i szczegółową dokumentację. Obejmują szerokie grupy interesariuszy. Ocena ryzyka w ramach technik ciężkich wykorzystuje szczegółowe czynniki prawdopodobieństwa ryzyka i wpływu ryzyka oraz wzory matematyczne do obliczania prawdopodobieństwa ryzyka i wpływu ryzyka na podstawie tych czynników. Przykładami technik ciężkich są:

- **Analiza zagrożeń:** rozszerza proces analityczny, próbując określić zagrożenia leżące u podstaw każdego ryzyka.
- **Koszt ekspozycji:** w odniesieniu do każdego elementu ryzyka jakościowego określa prawdopodobieństwo wystąpienia awarii, koszt straty związanej z typową awarią oraz koszt testowania pod kątem takich awarii.
- **Analiza przyczyn i skutków awarii (FMEA) i jej warianty:** identyfikuje ryzyka jakościowe, ich potencjalne przyczyny i prawdopodobne skutki, a następnie przypisuje im poziom istotności, priorytet i wskaźniki wykrywania.
- **Analiza drzewa usterek:** odnosi potencjalne awarie do defektów, które mogą spowodować awarię, a następnie do błędów, które mogą spowodować te defekty, kontynuując do momentu określenia różnych przyczyn podstawowych.

Z kolei techniki lekkie są mniej dokładne i wymagają mniejszego nakładu pracy ze strony zespołu testowego i interesariuszy. Podobnie jak techniki ciężkie, opierają się one również na zaangażowaniu interesariuszy, wykorzystując wyniki analizy ryzyka jako podstawę planowania testów i warunków testowych. Grupa interesariuszy może jednak nie być tak szeroka, a czynniki ryzyka są zwykle zredukowane do wpływu ryzyka i prawdopodobieństwa ryzyka w skali porządkowej. Niektóre z tych technik, takie jak systematyczne testowanie oprogramowania (ang. *Systematic Software Testing* - SST) (Craig i Jaskiel, 2002), mogą być stosowane tylko wtedy, gdy podano specyfikacje wymagań. Inne techniki, w tym pragmatyczna analiza i zarządzanie ryzykiem (ang. *Pragmatic Risk Analysis and Management* - PRAM) (Black, 2009) oraz zarządzanie ryzykiem produktowym (ang. *Product Risk Management* - PRISMA) (van Veenendaal, 2012), wykorzystują wymagania i/lub inne specyfikacje jako dane wejściowe do analizy ryzyka, ale mogą funkcjonować w całości w oparciu o dane wejściowe interesariuszy.

1.3.6 Wskaźniki sukcesu i trudności związane z testowaniem opartym na ryzyku

W retrospektywie zespół testowy powinien zmierzyć zakres, w jakim zrealizował korzyści płynące z testowania opartego na ryzyku. W wielu przypadkach wiąże się to z udzieleniem odpowiedzi na niektóre lub wszystkie z następujących pytań za pomocą metryk i konsultacji:

- Czy odpowiedni interesariusze byli zaangażowani lub reprezentowani w analizie ryzyka?
- Czy zaangażowanie interesariuszy w analizę ryzyka było odpowiednie?
- Jeśli wystąpiły krytyczne incydenty w produkcji, które wskazują, że przeoczono krytyczne defekty, czy zostały one usunięte?
- Czy większość defektów o wysokim priorytecie została wykryta na początku wykonywania testów?
- Czy zespół testowy był w stanie objaśnić wyniki testów interesariuszom pod kątem ryzyka?
- Czy pominięte testy miały niższy poziom powiązanego ryzyka niż te wykonane?

W większości przypadków udane testy oparte na ryzyku dają odpowiedź twierdzącą na wszystkie te pytania. W dłuższej perspektywie należy wyznaczyć cele doskonalenia procesów pod kątem wskaźników sukcesu, a także dążyć do poprawy efektywności procesu analizy ryzyka jakościowego.

Zarządzanie ryzykiem często napotyka nieoczekiwane trudności ze względu na zawilgości, które często są pomijane.

- **Trudność w ocenie poziomu ryzyka:** oszacowanie wpływu ryzyka i prawdopodobieństwa ryzyka może być bardzo trudne; rozwiązanie: wykorzystanie danych historycznych i poproszenie kluczowych interesariuszy projektu o ich ocenę.
- **Dobre początki:** ustanowienie i utrzymanie odpowiedniego podejścia do testów opartego na ryzyku jest często zaniedbywane w obliczu dużej krótkoterminowej presji na sukces; rozwiązanie: regularne monitorowanie i raportowanie ryzyka do interesariuszy.
- **Déjà vu:** ten sam zestaw ryzyk jest stosowany w odniesieniu do każdego projektu, co prowadzi do rutyny; rozwiązanie: zaangażowanie właściwych osób w identyfikację ryzyka i łagodzenie tylko tych ryzyk, które są uważane za ważne.
- **Pomijanie kluczowych ryzyk:** główną przyczyną tego problemu jest zazwyczaj zaangażowanie w proces niedoświadczonych lub nieodpowiednich osób; rozwiązanie: zaangażowanie odpowiednich osób i ich przeszkolenie.
- **Migracja interesariuszy:** interesariusze mogą zmieniać się w czasie, a także mogą pojawić się nowe ryzyka, więc analiza ryzyka jest ciągłym, iteracyjnym działaniem i nie powinna być wykonywana tylko raz na początku.

1.4 Projektowa strategia testowa

Wstęp

W tym sylabusie organizacyjna strategia testowa jest traktowana jako zadana. Opracowanie i utrzymanie organizacyjnej strategii testowej jest rozważane w kontekście normy ISO/IEC/IEEE 29119-3 (gdzie jest ona określana jako „organizacyjna praktyka testowa”) oraz sylabusów ISTQB® Poziom Ekspercki Zarządzanie Testami na poziomie i ISTQB® ATLaS (Certified Tester Agile Test Leadership at Scale).

Jeśli organizacyjna strategia testowa nie istnieje lub nie obejmuje wymaganych aspektów, zarządzanie testami musi dążyć do wyjaśnienia brakujących szczegółów z odpowiednimi interesariuszami.

W kontekście tego podrozdziału definicja projektowej strategii testowej jest przykładem dowolnego rodzaju szczegółowej strategii testów w odniesieniu do projektu, wydania, produktu lub dowolnego innego rodzaju inicjatywy wytwarzania lub przejęcia systemu. Projektowa strategia testowa (określana jako „strategia testów” w normie ISO/IEC/IEEE 29119-3) opisuje podejście do testowania w określonym kontekście, aby można było osiągnąć cele organizacji, w szczególności te odnoszące się do jakości produktu i czynności testowych. Strategia testów może również istnieć w odniesieniu do pojedynczego poziomu testów lub typu testu.

Projektowa strategia testowa jest głównym wynikiem planowania testów w projekcie i jest zwykle dokumentowana w planie testów lub jako część innych dokumentów. Zaleca się dokumentowanie strategii testów, ale niekoniecznie w formie formalnego planu testów. Potrzeba dokumentacji zależy od kontekstu testowania (patrz podrozdział 1.2). Gdy projekt jest zgodny z sekwencyjnym modelem wytwarzania oprogramowania, strategia testów w projekcie jest zwykle dokumentowana, najlepiej w planie testów (patrz ISO/IEC/IEEE 29119-3). Dokumentacja jest również często wymagana przez umowy, porozumienia, organy regulacyjne lub przepisy prawa.

1.4.1 Wybór podejścia do testów

Projektowa strategia testowa kieruje wszystkimi działaniami testowymi w projekcie i wyszczególnia cele, zasoby, harmonogramy i odpowiedzialności. Strategia ta musi być dostosowana do unikalnych wymagań projektu. Kluczowe decyzje obejmują wybór poziomów testów, typów testów i technik testowych do testowania statycznego i dynamicznego oraz innych praktyk testowych (np. testowanie skryptowe, testowanie manualne, testowanie back-to-back).

Teoretycznie wszystkie typy testów mogą być wykonywane na dowolnym poziomie testów, a każda technika testowania może być zastosowana do dowolnego typu testów na dowolnym poziomie testów. W praktyce odpowiedni dobór i połączenie tych wyborów ma znaczący wpływ na skuteczność i efektywność testów. Na przykład łatwość utrzymania kodu może być często oceniana bardziej efektywnie i wydajnie przy użyciu statycznej analizy kodu lub przeglądu kodu. Z drugiej strony, wydajność może być lepiej oceniana za pomocą skryptowych testów systemowych ze względu na interakcję modułów wewnętrznych lub użyteczność funkcjonalności może być lepiej walidowana z użytkownikami za pomocą wspólnie opracowanych manualnych testów akceptacyjnych. Wybór najlepszego podejścia do strategii testów może być złożonym procesem, na który może wpływać organizacyjna strategia testowa, kontekst projektu i inne aspekty.

Wybór i połączenie poziomów testów, rodzajów testów i technik testowania ma zatem kluczowe znaczenie dla skutecznej strategii testowania w ramach projektu, ponieważ znacząco wpływa na wydajność i skuteczność testowania.

1.4.2 Analiza organizacyjnej strategii testowej, kontekstu projektu i innych aspektów

Aby umożliwić opracowanie projektowej strategii testowej, należy w pełni zrozumieć organizacyjną strategię testową, kontekst projektu oraz dodatkowe czynniki lub ograniczenia związane z testowaniem.

Aby wybrać odpowiednie podejście do testów, należy zazwyczaj przeanalizować następujące czynniki:

- **Domena:** domena, dla której produkt zostanie utworzony lub zmodyfikowany. Wszelkie regulacje, normy i praktyki specyficzne dla danej dziedziny mogą zmienić rygor testowania, wymaganą dokumentację, a także jej poziom szczegółowości. Na przykład w farmacji i medycynie podejście do testów często kładzie nacisk na intensywne testowanie akceptacyjne przez użytkownika skoncentrowane na zagrożeniach dla zdrowia pacjentów, wykorzystując przypadki testowe oparte na funkcjonalnych wymaganiach użytkowników, podczas gdy testowanie akceptacyjne przez użytkownika internetowych aplikacji ubezpieczeniowych może koncentrować się na użyteczności i zwiększaniu prawdopodobieństwa zawarcia nowych umów ubezpieczeniowych poprzez testowanie A/B.
- **Cele organizacyjne i nadrzędne charakterystyki jakościowe:** cele organizacyjne mogą obejmować potrzebę wykazania wartości testowania i zwiększenia stopnia automatyzacji testów lub charakterystyk jakościowych procesu testowego, takich jak poziom dojrzałości testów lub skuteczność wykrywania defektów. Mogą one określać poziomy i typy testów, których należy przestrzegać.
- **Cele projektu i rodzaj projektu:** cele projektu (np. dotyczące budżetu, czasu i jakości) oraz rodzaj projektu (tj. specyficzny dla klienta w porównaniu z wytwarzaniem produktu zorientowanym na rynek) zazwyczaj zawierają ograniczenia i ryzyka, a także możliwości, które wpływają na testowanie. Na przykład ściśle ograniczenia budżetowe i czasowe mogą wymagać rygorystycznego stosowania testów opartych na ryzyku w celu nadania priorytetu przypadkom testowym do wykonania, podczas gdy opracowanie produktu specjalnie dla klienta może wymagać testów, które obejmują wcześniej zdefiniowane umowne kryteria akceptacji.
- **Zasoby testowe:** należy wziąć pod uwagę wszelkie ograniczenia dotyczące dostępności zasobów testowych, w tym narzędzi testowych, infrastruktury testowej, technologii i środowiska programistycznego wykorzystywanego w projekcie, a także dostępnego personelu testującego i jego umiejętności (patrz podrozdział 3.1). Na przykład testowanie w oparciu o doświadczenie wymaga testerów z dobrą wiedzą na temat domeny; aplikacje mobilne zwykle muszą być testowane na zazwyczaj ograniczonej liczbie różnych urządzeń; korzystanie z narzędzi testujących może być ograniczone przez liczbę dostępnych licencji.
- **Model cyklu życia oprogramowania zastosowany w projekcie:** aby określić odpowiednie poziomy testów, nakład pracy związany z testowaniem, odpowiednie kryteria wejścia i wyjścia, patrz sylabus Certyfikowany Tester Poziom Podstawowy ISTQB® w. 4.0, podrozdziały 2.2 i 5.1. Cykl życia oprogramowania z ciągłą integracją wymaga bardziej zautomatyzowanego testowania niż jednorazowe wytwarzanie przy użyciu modelu kaskadowego, a zatem można stosować różne rodzaje testów i techniki testowania.
- **Interfejsy z innymi systemami:** w systemie systemów zasadnicze znaczenie ma dostosowanie testów do innych zespołów lub projektów i wybór odpowiednich poziomów testów, zwłaszcza do testowania integracji systemów. Na przykład testowanie oparte na ryzyku pomaga ustalić priorytety i skalować testowanie integracji systemów.
- **Dostępność danych testowych:** należy wziąć pod uwagę ograniczenia w dostępności danych testowych, takie jak potrzeba zanonimizowanych danych testowych z produkcji lub

tworzenie konkretnych danych testowych, które mogą być trudne do dostarczenia i wymagają walidacji, takich jak dane do testowania AI. Na przykład testowanie oparte na modelu może wspierać tworzenie danych testowych i zarządzanie nimi.

Kierownik testów powinien określić, która kombinacja technik testowania, poziomów testów i typów testów powinna zostać wykorzystana jako najlepsze podejście do realizacji organizacyjnej strategii testowej, kontekstu projektu oraz dodatkowych czynników lub ograniczeń związanych z testowaniem.

1.4.3 Definicja celów testów

Plan testów powinien być określony w odniesieniu do każdego projektu testowego i powinien zawierać, między innymi, zakres testów, cele testów i kryteria wyjścia. Plan testów można skonfigurować na poziomie wydania jako plan testów projektu (inaczej główny plan testów) oraz, w razie potrzeby, jako plan testów poziomu dla różnych poziomów testów. Dodatkowo można określić plany testów w odniesieniu do różnych charakterystyk jakościowych, takie jak plan testów zabezpieczeń lub plan testów wydajności. W projektach stosujących zwinne wytwarzanie oprogramowania lub hybrydowy model wytwarzania oprogramowania można uzgodnić plan testów iteracyjnych. Dla każdego wydania i iteracji zakres cech funkcjonalnych i ich charakterystyk нефункциональных, które mają zostać dostarczone, jest określany w planie testów i uzgadniany z interesariuszami.

W powiązaniu z funkcjami dostarczonymi do testowania w projekcie należy określić cele testowe projektu i kryteria wyjścia. Można to zrobić za pomocą metody określania celów SMART:

- S = [specific] konkretny. Projektowy cel testów i kryterium wyjścia powinny być jasne i jednoznaczne.
- M = [measurable] mierzalny. Powinien być wymierny i mieć określone kryteria pomiaru postępów w celu ustalenia, czy został osiągnięty.
- A = [achievable] osiągalny. Powinien być wykonalny, biorąc pod uwagę dostępne zasoby, ramy czasowe i możliwości.
- R = [relevant] istotny. Powinien być zgodny z ogólnymi celami projektu.
- T = [timely] terminowy. Powinien mieć konkretne ramy czasowe i określony termin realizacji.

Cele testów projektu powinny obejmować wszystkie ukierunkowane aspekty jakości i ilości, pod warunkiem, że są one mierzalne lub możliwe do oceny. Przykładami celów testów projektu są:

- osiągnięcie określonych kryteriów wyjścia w określonych ramach czasowych;
- osiągnięcie celów jakościowych organizacji (np. mierzonych jako kluczowy wskaźnik wydajności w odniesieniu do liczby reklamacji klientów dotyczących produktu);
- przestrzeganie zasad i przepisów danej branży;
- zapewnienie dostępności danych tylko upoważnionym użytkownikom (np. poprzez prawa dostępu);
- sprawdzenie kompletności funkcjonalnej, poprawności funkcjonalnej, wydajności, skuteczności, przenaszalności i bezpieczeństwa migracji danych;
- zwiększenie poziomu automatyzacji testów (np. w przypadku testów regresji lub wydajności o określony procent);
- pomyślna refaktoryzacja kodu i wykazanie, że nie wprowadziła ona nowych defektów (np. w celu usunięcia słabo ustrukturyzowanego kodu źródłowego lub długu technicznego przy jednoczesnym zachowaniu istniejącej funkcjonalności, potwierdzonego testem regresji);

- udowodnienie bezpieczeństwa interfejsów (np. poprzez weryfikację komunikatów XML (ang. *Extensible Markup Language*)) pod kątem ich definicji schematu XML w celu zapewnienia odrzucenia złośliwych danych);
- sprawdzenie użyteczności interfejsu użytkownika i uzyskanie pewnego stopnia podcharakterystyki (np. poprzez pomiar czasu potrzebnego na wykonanie określonego zadania w sklepie internetowym).

Oprócz liczenia i pomiaru celów testów projektu należy rozważyć ocenę poziomu jakości przez ekspertów i interesariuszy na poziomie domeny.

W zależności od kontekstu projektu i celów testów, czasami może być wymagane wiele środowisk testowych z dostępnymi zasobami i/lub narzędziami testowymi. Nie wszystkie środowiska testowe mogą być dostępne w tym samym czasie. Należy wziąć to pod uwagę przy formułowaniu możliwych do osiągnięcia celów testów i kryteriów wyjścia.

W zależności od kontekstu projektu, przy określaniu celów testów i zakresu testów projektu należy wziąć pod uwagę dodatkowe czynniki, jak opisano w podrozdziale 1.2 Kontekst testowania.

1.5 Usprawnianie procesu testowego

Wstęp

Testowanie jest ważną częścią wytwarzania oprogramowania i często stanowi co najmniej 30–40% całkowitych kosztów projektu. Oprócz wielu wyzwań (technicznych), przed którymi stoją projekty oprogramowania (np. rosnąca złożoność i rozmiar, nowe technologie, szeroka gama urządzeń i systemów operacyjnych oraz luki w zabezpieczeniach), istnieje potrzeba optymalizacji skuteczności i wydajności testowania, a także potrzeba odpowiedniego usprawnienia procesów testowych. Uczenie się na podstawie istniejących najlepszych praktyk i własnych błędów pozwala usprawnić proces testowy i sprawić, aby projekty były bardziej udane.

Proces usprawniania na poziomie organizacji jest zwykle bardziej przydatny niż proces usprawniania na poziomie projektu lub zespołu. Niemniej możliwe i korzystne jest również zastosowanie usprawniania procesów na poziomie projektu lub zespołu, ale powinno ono być dostosowane do potrzeb projektu lub zespołu. Usprawnianie testów może być zainicjowane np. przez niezadowolenie z wyników bieżących testów, nieoczekiwane defekty, zmieniające się okoliczności, wynik testu porównawczego lub brak komunikacji. Dostępne są różne techniki usprawniania testowania (Bath i van Veenendaal, 2014). Niektóre z tych technik opisano poniżej. Techniki opisane w tym sylabusie mogą być stosowane zarówno w sekwencyjnych, jak i zwinnych/przyrostowych modelach wytwarzania oprogramowania. Szczegółowe informacje można znaleźć w sylabusie ISTQB® Poziom Ekspertki Wdrażanie usprawnień procesu testowego.

1.5.1 Proces usprawniania testów (IDEAL)

Po uzgodnieniu, że należy usprawnić procesy testowe, czynności związane z wdrażaniem usprawnień procesów, które należy przyjąć na potrzeby tego działania, można określić jak w modelu IDEAL, który opiera się na podobnych pomysłach jak dobrze znany cykl Deminga (ang. *plan-do-check-act* - PDCA). IDEAL to akronim oznaczający inicjowanie, diagnozowanie, ustalanie, działanie i uczenie się (ang. *Initiating, Diagnosing, Establishing, Acting, Learning*).

Mimo że IDEAL został pierwotnie określony w celu wspierania działań usprawniających na poziomie organizacyjnym, może być również stosowany na poziomie projektu lub zwinnego zespołu twórczego. W kontekście projektu cele działań (patrz poniżej) nie zostały jeszcze osiągnięte. Główną różnicą jest prawdopodobnie faza inicjująca, która jest znacznie mniejsza na poziomie projektu lub zespołu niż na poziomie organizacji. Diagnozowanie poprzez retrospektywę i ustalenie planu byłoby najprawdopodobniej znacznie mniejsze niż na poziomie organizacyjnym. Działanie i uczenie się będą również istotne na poziomie projektu lub zespołu.

Inicjowanie procesu usprawniania

Na początku procesu usprawniania interesariusze uzgadniają cele i zakres usprawnień procesu.

Diagnozowanie bieżącej sytuacji

Bieżący proces testowy jest oceniany w celu określenia możliwych usprawnień. Ocena jest zwykle przeprowadzana w oparciu o standardowe ramy w przypadku usprawniania procesu testowego w oparciu o model (patrz sekcja 1.5.2) lub może być oparta na analizie konkretnych metryk w przypadku usprawniania procesu testowego w oparciu o analizę (patrz sekcja 1.5.3).

Ustanowienie planu usprawniania procesu testowego

Plan usprawnień procesu testowego może być formalnym dokumentem, który zawiera listę wszystkich szczegółowych działań, które należy wykonać, aby osiągnąć usprawnienia. W zależności od kontekstu, plan może być wysoce nieformalny i bardzo lekki. Lista możliwych usprawnień procesu powinna zostać uszeregowana według priorytetów. Priorytetyzacja może opierać się na zwrocie z inwestycji (ang. *return on investment* - ROI), ryzykach, zgodności ze strategiami projektu lub zespołu i/lub wymiernych korzyściach ilościowych lub jakościowych, które mają zostać osiągnięte.

Działanie w celu wdrożenia usprawnienia procesu testowego

W celu wprowadzenia usprawnień wdrażany jest plan usprawnień procesu testowego. Zwykle obejmuje to szkolenie i pilotowanie zmienionych procesów oraz ich pełne wdrożenie w projekcie lub zespole.

Uczenie się z usprawnień

Po pełnym wdrożeniu usprawnień procesu konieczne jest sprawdzenie, jakie korzyści, planowane lub nieoczekiwane, zostały osiągnięte. Dowiedziawszy się, co zadziało, a co nie, musimy podjąć działania w oparciu o te informacje, a dopiero potem można rozpocząć kolejny cykl usprawniania.

1.5.2 Usprawnianie procesu testowego w oparciu o model

Jedną z przesłanek zarówno doskonalenia procesu testowego w oparciu o model, jak i doskonalenia w oparciu o analizę, jest założenie, że jakość produktu jest w dużym stopniu uzależniona od jakości stosowanych procesów. Przy stosowaniu usprawniania procesu testowego w oparciu o model wykorzystuje się model usprawniania testów. Modele usprawniania testów opierają się na najlepszych praktykach w zakresie testowania i organizują usprawnianie testów w sposób stopniowy.

Pojawiło się kilka zalecanych modeli procesu, które wspierają usprawnianie procesu testowego. Obejmują one zintegrowany model dojrzałości testów (ang. *Test Maturity Model integration* - TMMi®) i TPI NEXT®.

Usprawnianie w oparciu o model może być również stosowane na poziomie projektu. W takich przypadkach proces oceny i usprawniania koncentruje się w szczególności na procesach testowych lub kluczowych obszarach zdefiniowanych w modelu, które odnoszą się do działań na poziomie projektu (np. planowanie testów i projektowanie testów) i często w dużej mierze pomija te, które są na poziomie organizacji (np. polityka testów i organizacja testów). Alternatywnie można również odpowiednio dostosować praktyki dotyczące poziomu organizacyjnego do kontekstu projektu.

Więcej informacji na temat usprawniania procesu testowego w oparciu o model można znaleźć w sylabusie ISTQB® Poziom Ekspercki Wdrażanie usprawnień procesu testowego.

Zintegrowany model dojrzałości testów (TMMi®)

Zintegrowany model dojrzałości testów (TMMi®) (van Veenendaal i Cannegieter, 2011) (van Veenendaal, 2020) składa się z pięciu poziomów dojrzałości. Każdy poziom dojrzałości, z wyjątkiem poziomu 1 TMMi®, zawiera obszary procesu testowego i cele usprawniania. Ponadto, aby ułatwić i wesprzeć jego wdrożenie, TMMi® określa praktyki, podpraktyki i przykłady. TMMi® został początkowo opracowany w celu uzupełnienia zintegrowanego modelu dojrzałości organizacyjnej (CMMI® - *Capability Maturity Model Integration*), ale dziś jest szeroko stosowany niezależnie od CMMI®.

Aby ułatwić i wesprzeć aktualizację TMMi® w zwinnym wytwarzaniu oprogramowania, opracowano specjalne wytyczne, które wyjaśniają, w jaki sposób TMMi® może być używany i korzystnie stosowany w zwinnym wytwarzaniu oprogramowania.

Więcej informacji na temat TMMi® można znaleźć pod adresem www.tmmi.org.

TPI NEXT®

Model TPI NEXT® (van Ewijk, 2013) określa 16 kluczowych obszarów, z których każdy obejmuje określony aspekt procesu testowego (np. strategię testów, metryki testów, narzędzia testowe i środowisko testowe). W modelu określono cztery poziomy dojrzałości dla każdego z 16 kluczowych obszarów.

Konkretne punkty kontrolne określa się w celu oceny każdego kluczowego obszaru na każdym z poziomów dojrzałości. Wyniki oceny są podsumowywane i wizualizowane za pomocą macierzy dojrzałości, która obejmuje wszystkie kluczowe obszary.

Więcej informacji na temat TPI NEXT® można znaleźć pod adresem www.tmap.net.

1.5.3 Podejście do usprawniania procesu testowego w oparciu o analizę

W przypadku korzystania z podejścia do usprawniania w oparciu o model, jak opisano w poprzedniej sekcji, usprawnienia są wprowadzane poprzez porównanie podejścia testowego projektu lub zespołu z zewnętrznymi najlepszymi praktykami. Podejścia analityczne pozwalają określić problemy w oparciu o dane z samego projektu lub zespołu. Odpowiednie usprawnienia można uzyskać na podstawie analizy tych problemów. Podejścia analityczne mogą być stosowane wraz z podejściem opartym na modelu w celu weryfikacji wyników i zapewnienia różnorodności.

Problemy można zidentyfikować za pomocą danych ilościowych i jakościowych. W sekcji 1.5.3 Podejście do usprawniania procesu testowego w oparciu o analizę wprowadzono podejścia analityczne, które wykorzystują głównie dane ilościowe z procesu testowego i dane z defektów do oceny obecnego podejścia. Sekcja 1.5.4 Retrospektywy wprowadza retrospektywy, w których dane jakościowe dotyczące tego, co działa dobrze, a co nie, są zbierane od członków zespołu wytwórczego i testowego.

Analiza danych jest ważna dla obiektywnego doskonalenia procesu testowego i stanowi cenne wsparcie do ocen czysto jakościowych, które w przeciwnym razie mogą skutkować nieprecyzyjnymi zaleceniami, które nie są poparte danymi. Zastosowanie analitycznego podejścia do doskonalenia najczęściej wiąże się z ilościową analizą procesu testowego w celu określenia obszarów problemowych i wyznaczenia celów specyficznych dla projektu. Określenie i pomiar kluczowych parametrów są wymagane do oceny procesu testowego i oceny, czy usprawnienia są skuteczne.

Przykładami podejść analitycznych są:

- analiza przyczyny podstawowej;
- analiza z wykorzystaniem miar, metryk i wskaźników;
- podejście GQM (ang. *Goal-Question-Metric* - Cel Pytanie Metryka).

Analiza przyczyny podstawowej to badanie problemów w celu określenia ich przyczyny podstawowej. Pozwala to na określenie rozwiązań, które usuwają przyczyny problemów, a nie tylko odnoszą się do natychmiastowych oczywistych objawów. Możliwa procedura analizy obejmowałaby wybranie odpowiedniego zestawu defektów, określenie klastrow w tych danych oraz użycie diagramów przyczynowo-skutkowych (zwanym również diagramami Ishikawy lub diagramami rybiej ości) w celu określenia głównych przyczyn ważnych klastrow defektów. Następnie wprowadza się usprawnienia, aby zapobiec wystąpieniu podobnych defektów.

Miary, metryki i wskaźniki są wykorzystywane w sposób ilościowy do oceny tego, jak dobrze wykonywany jest proces testowy w projekcie lub zespole. Kluczowe cechy procesu testowego, które należy wziąć pod uwagę, to skuteczność, efektywność i przewidywalność. W odniesieniu do każdego z tych atrybutów można wybrać jedną lub kilka metryk. Gromadząc i analizując odpowiednie dane, można określić kluczowe obszary wymagające poprawy.

Podejście GQM (Basili, et al., 2014) (van Solingen i Berghout, 1999) zapewnia ramy do określania i analizowania metryk, które są dostosowane do potrzeb informacyjnych odpowiednich interesariuszy projektu. Cele pomiarowe określają aspekt jakościowy obiektu, który wymaga pomiaru w określonym celu, perspektywie i kontekście. Cele te są przekształcane w pytania, które określają aspekt jakości z punktu widzenia interesariuszy. Następnie wybierane są metryki, które dostarczają niezbędnych informacji, aby odpowiedzieć na pytanie. Dane zebrane na potrzeby metryk odpowiadają na pytania, aby ocenić cel pomiaru i zaspokoić potrzeby informacyjne interesariuszy.

Więcej informacji na temat tych opartych na analizie podejść do usprawniania procesów testowych można znaleźć w sylabusie ISTQB® Poziom Ekspercki Wdrażanie usprawnień procesu testowego.

1.5.4 Retrospektywy

Retrospektywy to spotkania, podczas których zespół dokonuje przeglądu swoich metod i współpracy, rejestruje wyciągnięte wnioski (dobre i złe) oraz decyduje o zmianach i działaniach w celu osiągnięcia usprawnień (zarówno w zakresie testowania, jak i kwestii niezwiązanych z testowaniem). Retrospektywy dotyczą takich tematów jak proces, ludzie, organizacja, współpraca i narzędzia.

Retrospektywy są stosowane zarówno w sekwencyjnych modelach wytwarzania oprogramowania, jak i w zwinnym wytwarzaniu oprogramowania. W sekwencyjnych modelach wytwarzania są one częścią ukończenia testów. W tym kontekście retrospektywy mają na celu generowanie wyciągniętych wniosków w celu lepszego zarządzania przyszłymi projektami. W zwinnym wytwarzaniu oprogramowania retrospektywy są zwykle przeprowadzane na koniec każdej iteracji, aby omówić, co się udało, a co należy poprawić, oraz w jaki sposób te usprawnienia można włączyć do następnej iteracji. Retrospektywy są przeprowadzane przez cały zespół, a tym samym wspierają podejście „cały zespół” i sprzyjają ciągłemu usprawnianiu. Należy pamiętać, że czasami wymagane są dedykowane retrospektywy w celu rozwiązania problemów związanych z testowaniem.

Typowa retrospektywa składa się z następujących etapów:

Wprowadzenie: cel i program retrospektywy są poddawane przeglądowi oraz tworzona jest atmosfera wzajemnego zaufania, aby problemy mogły być omawiane bez obwiniania lub osądzania;

Zgromadzenie danych: gromadzone są dane dotyczące tego, co wydarzyło się podczas iteracji lub projektu. Możliwe jest gromadzenie danych jakościowych, takich jak oś czasu kluczowych wydarzeń, która określa problemy i wymienia opinie każdego członka zespołu na temat tych problemów. Ponadto można przedstawić dane ilościowe z metryk – na przykład dane dotyczące postępu testów, wykrywania defektów, skuteczności testów, wydajności testów i przewidywalności mogą zapewnić obiektywny wgląd w testowanie projektu lub iteracji.

Wprowadzenie usprawnień: zebrane dane są analizowane w celu zrozumienia obecnej sytuacji i wygenerowania pomysłów na usprawnienia. Na przykład analiza przyczyny podstawowej może być stosowana do określania przyczyn podstawowych wykrytych problemów, a sesja burzy mózgów może być przeprowadzana w celu wygenerowania pomysłów na rozwiązanie przyczyn podstawowych.

Określenie działań usprawniających: wyprowadzane są działania mające na celu wdrożenie pomysłów na usprawnienia oraz ustalane są ich priorytety. Określany jest plan usprawnień i obowiązki. Cele i powiązane z nimi metryki można określić w celu oceny wpływu działań na wykryte problemy. Wdrożenie zbyt wielu usprawnień jednocześnie jest trudne do wykonania za pomocą weryfikowalnych kroków.

Zamknięcie retrospektywy: w tym ostatnim kroku dokonuje się przeglądu samej retrospektywy w celu wskazania mocnych stron i usprawnień w procesie retrospektywy. Retrospektywa jest przeprowadzana

regularnie, szczególnie w zwinnym wytwarzaniu oprogramowania. Ciągłe doskonalenie dotyczy również samej retrospektywy.

Ważne jest, aby odpowiednio dokumentować wyniki retrospektywy. W sekwencyjnym modelu wytwarzania ustalenia, wnioski i zalecenia muszą być rozpowszechniane i przekazywane członkom organizacji w sposób zrozumiały. W zwinnym wytwarzaniu oprogramowania należy również udokumentować problemy i działania, aby umożliwić przegląd działań i ich potencjalnego wpływu na problemy w następnej iteracji.

Testerzy, będąc częścią zespołu (projektowego), wnoszą swoją unikalną perspektywę. Mogą zgłaszać problemy związane z testowaniem (i inne) i zachęcać zespół do myślenia o możliwych usprawnieniach.

Więcej informacji można znaleźć w (Derby i Larsen, 2006).

1.6 Narzędzia testowe

Wstęp

Istnieją trzy rodzaje narzędzi biznesowych:

- narzędzia komercyjne;
- narzędzia *open source*;
- narzędzia niestandardowe.

Przy wyborze narzędzia biznesowego należy wziąć pod uwagę wszystkie wymagania organizacyjne i interesariuszy oraz przepisy.

Istnieją również narzędzia techniczne, takie jak narzędzia do automatyzacji testów, narzędzia do zarządzania testami i wiele innych.

Przykłady zastosowania narzędzi testowych można znaleźć w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0.

1.6.1 Dobre praktyki dotyczące wprowadzania narzędzi

Ta sekcja zawiera kroki niezbędne do oceny i wprowadzenia narzędzia testowego.

Kierownik testów może być zaangażowany we wprowadzanie narzędzia lub może wspierać lub ułatwiać proces jego wprowadzania. Kierownicy testów są zazwyczaj odpowiedzialni za dedykowane narzędzie testowe lub dowolne narzędzie związane z testowaniem, takie jak narzędzie do zarządzania wymaganiami, defektami lub monitorowania.

Istnieją ogólne dobre praktyki i rozważania brane pod uwagę przy ocenie i wyborze narzędzia testowego. Te praktyki i rozważania obejmują następujące kwestie:

- określenie możliwości usprawnienia procesów przy wsparciu odpowiednich narzędzi;
- zrozumienie technologii stosowanych w organizacji i wybór narzędzia zgodnego z tymi technologiami;
- zrozumienie, w jaki sposób narzędzie jest technicznie i organizacyjnie zintegrowane z SDLC;
- ocena narzędzia pod kątem jasnych wymagań i obiektywnych kryteriów;
- ocena dostawcy, jeśli rozważa się użycie narzędzia komercyjnego; ocena wsparcia dla narzędzi niekomercyjnych (np. *open source*);
- określenie wewnętrznych wymagań dotyczących coachingu, mentoringu lub szkolenia w zakresie korzystania z narzędzia;
- rozważenie zalet i wad różnych modeli licencjonowania;
- ostatni krok: przeprowadzenie oceny poprawności koncepcji.

Ogólne dobre praktyki w zakresie przyjmowania i wdrażania narzędzia obejmują:

- przeprowadzenie projektu pilotażowego w celu walidacji kryteriów i wymagań wyboru oraz oceny dopasowania narzędzia do istniejących procesów i praktyk;
- dostosowywanie i usprawnianie procesów zgodnie z potrzebami narzędzia, a także dostosowywanie narzędzia do istniejących procesów, jeśli to konieczne;
- określenie wytycznych dotyczących korzystania z narzędzia;
- zapewnienie szkoleń, coachingu i mentoringu użytkownikom narzędzia;
- wdrożenie narzędzia w organizacji w przyrostach;

- wdrożenie sposobu gromadzenia informacji dotyczących rzeczywistego korzystania z narzędzia w celu dalszych ulepszeń;
- określenie własności narzędzia.

1.6.2 Techniczne i biznesowe aspekty decyzji dotyczących narzędzi

Na decyzję o wdrożeniu i wykorzystaniu narzędzia wpływa wiele czynników. Dla kierownika testów ważne jest, aby je poznać i odnieść się do nich.

- **Przepisy i bezpieczeństwo:** organizacje, które tworzą oprogramowanie o krytycznym znaczeniu dla bezpieczeństwa lub działalności lub podlegają zgodności z przepisami, mogą preferować narzędzia komercyjne, ponieważ częściej spełniają one wymagane standardy i często posiadają odpowiednie certyfikaty.
- **Aspekty finansowe:** narzędzia *open source* zwykle charakteryzują się niższym kosztem początkowym ze względu na rozwój i wsparcie społeczności. Narzędzia komercyjne mogą mieć jednorazową cenę zakupu, a także powtarzające się koszty licencji. Początkowy koszt narzędzia niestandardowego jest trudny do określenia, ponieważ zależy on od wymagań i etapu rozwoju narzędzia. Oprócz kosztów początkowych należy obliczyć i uwzględnić koszty szkolenia i utrzymania przez cały okres korzystania z narzędzia. Wszystkie narzędzia mogą wiązać się z wysokimi kosztami utrzymania i wsparcia.
- **Wymagania interesariuszy:** ważne jest, aby zebrać wymagania od wszystkich interesariuszy w celu oceny i określenia najbardziej odpowiedniego narzędzia. Narzędzia komercyjne i narzędzia *open source* niekoniecznie spełniają szczegółowo wszystkie wymagania. Narzędzia niestandardowe mogą być najlepszym wyborem, aby sprostać wszystkim indywidualnym wymaganiom, a także w przypadkach, gdy żadne inne narzędzie nie zapewnia wymaganej funkcjonalności.
- **Istniejące oprogramowanie i strategia narzędziowa:** należy ocenić istniejący zestaw narzędzi (środowisko oprogramowania) i związaną z nim strategię narzędziową, ponieważ mogą istnieć preferowani lub zablokowani dostawcy, zintegrowane systemy, które są zależne od innych produktów, lub specjalny model pełnego wsparcia dla całego środowiska oprogramowania z określonymi przepisami.

1.6.3 Rozważania dotyczące procesu wyboru i ocena zwrotu z inwestycji

Narzędzia testowe mogą być inwestycją długoterminową, być może rozciągającą się na wiele iteracji pojedynczego projektu i/lub mającą zastosowanie do wielu projektów. Potencjalne narzędzie należy zatem rozpatrywać z różnych punktów widzenia.

- W przypadku kierownictwa wyższego szczebla wymagany jest pozytywny zwrot z inwestycji.
- W przypadku zespołu wsparcia i zespołu operacyjnego preferowana jest ograniczona, ale niezbędna liczba narzędzi używana w całej organizacji. Utrzymywanie większej liczby narzędzi, śledzenie ich licencji i zarządzanie stosem narzędzi nie powinno być kosztowne ani czasochłonne.
- W przypadku liderów projektu narzędzie musi dodawać wymierną wartość do projektu lub organizacji.
- W przypadku osób korzystających z narzędzia bardzo ważna jest użyteczność. Użyteczność obejmuje np. wsparcie dla danych zadań, zdolność uczenia się i operacyjność.
- W przypadku członków personelu operacyjnego ważna jest łatwość utrzymania.

Funkcje muszą być analizowane w odniesieniu do każdego rodzaju narzędzia biznesowego i technicznego. Na tę analizę mają wpływ różne perspektywy i interesy: zarządzanie testami, analiza testów technicznych, automatyzacja testów lub wytwarzanie. Osoba w organizacji, która jest odpowiedzialna za narzędzie (właściciel narzędzia) musi upewnić się, że analiza została wykonana, a wyżej wymienione punkty zostały wzięte pod uwagę.

Wszystkie narzędzia wprowadzone do procesu testowego powinny również zapewniać pozytywny zwrot z inwestycji dla organizacji. Obowiązkiem kierownika testów jest zadbanie o obliczenie i dalszą ocenę ROI. W zwinnym wytwarzaniu oprogramowania może to leżeć w gestii całego zespołu wytwórczego.

Przed nabyciem lub zbudowaniem narzędzia należy przeprowadzić analizę kosztów i korzyści, aby upewnić się, że przynosi ono korzyści organizacji. Analiza ta powinna uwzględniać zarówno koszty powtarzające się, jak i jednorazowe.

Jednorazowe działania i koszty:

- określanie wymagań dotyczących narzędzi, aby osiągnąć cele;
- ocena i wybór właściwego narzędzia i dostawcy, weryfikacja koncepcji;
- zakup, dostosowanie lub opracowanie narzędzia do pierwszego użycia;
- określenie wytycznych dotyczących korzystania z narzędzia;
- szkolenie wstępne dotyczące narzędzia;
- integracja narzędzia z istniejącym środowiskiem;
- zakup sprzętu/oprogramowania potrzebnego do obsługi narzędzia.

Powtarzające się działania i koszty:

- powtarzające się opłaty licencyjne i opłaty za wsparcie;
- koszty utrzymania;
- koszty ciągłego szkolenia;
- przenoszenie narzędzia do różnych środowisk.

Należy również wziąć pod uwagę koszty związane z możliwościami. Oznacza to, że czas poświęcony na ocenę, administrowanie, szkolenie i korzystanie z narzędzia mógł zamiast tego zostać poświęcony na rzeczywiste zadania testowe. W związku z tym może być potrzebne więcej zasobów testowych, zanim narzędzie będzie mogło zostać wykorzystane do zamierzonych działań.

Przy wyborze narzędzi należy wziąć pod uwagę następujące ryzyka dotyczące ROI:

- niedojrzałość organizacji może prowadzić do nieefektywnego korzystania z narzędzia;
- zmiany w polityce utrzymania przez dostawcę mogą zwiększyć obciążenie pracą;
- wyższe koszty niż oczekiwane;
- niższa korzyść niż oczekiwana.

W przypadku narzędzi testowych mogą mieć zastosowanie następujące korzyści:

- ograniczenie manualnej powtarzalnej pracy (np. testy regresji);
- przyspieszenie cyklu testowego poprzez automatyzację;
- oszczędność kosztów wykonania testów dzięki zmniejszeniu liczby czynności manualnych;
- zwiększone pokrycie niektórych typów testów obsługiwanych przez narzędzie;
- redukcja błędów ludzkich dzięki mniejszej liczbie czynności manualnych;
- szybszy dostęp do informacji o testach.

Dodatkowe korzyści i ryzyka, szczególnie w przypadku narzędzi do automatyzacji testów, można znaleźć w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0 i sylabusie ISTQB® Certyfikowany Tester Poziom Zaawansowany Inżynieria Automatyzacji Testów w. 2.0.

Ogólnie rzecz biorąc, organizacja testowa rzadko korzysta z jednego narzędzia. Całkowity zwrot z inwestycji dla organizacji jest zwykle połączeniem zwrotu z inwestycji wszystkich narzędzi używanych do testowania. Narzędzia muszą udostępniać informacje i współpracować ze sobą. Wskazana jest długoterminowa, kompleksowa strategia narzędzi testowych, która obejmuje ryzyko, koszty i korzyści.

1.6.4 Cykl życia narzędzia

Istnieją cztery różne etapy w cyklu życia narzędzia. Należy wyznaczyć administratora narzędzia w celu zapewnienia, aby działania w ramach tych etapów były określone, przeprowadzane i zarządzane.

- **Pozyskanie:** przede wszystkim podjęto decyzję o wyborze narzędzia. W drugim kroku należy przypisać właściciela narzędzia. Osoba ta podejmuje decyzje dotyczące korzystania z narzędzia (np. konwencje nazewnictwa produktów pracy i miejsca przechowywania tych produktów). Podejmowanie tych decyzji z góry może mieć znaczący wpływ na ostateczny zwrot z inwestycji w narzędzie.
- **Wsparcie i utrzymanie:** właściciel narzędzia jest odpowiedzialny za jego utrzymanie. Odpowiedzialność za czynności konserwacyjne powinna spoczywać na administratorze narzędzia lub dedykowanej grupie utrzymującej narzędzia. W przypadku współdziałania należy wziąć pod uwagę wymianę danych oraz procesy współpracy i komunikacji. Wymagane są również decyzje dotyczące tworzenia kopii zapasowych i przywracania artefaktów związanych z narzędziem.
- **Ewolucja:** w miarę upływu czasu środowisko, potrzeby biznesowe lub decyzje dostawców mogą wymagać zmian w narzędziu. Im bardziej złożone staje się środowisko pracy narzędzia, tym łatwiej zmiana może zakłócić jego użytkowanie.
- **Wycofanie:** po zakończeniu okresu użytkowania narzędzie powinno zostać wycofane z użytku. W większości przypadków funkcjonalność dostarczona przez narzędzie zostanie wymieniona, a dane muszą zostać zachowane i/lub zarchiwizowane. Może to wynikać z decyzji dostawcy lub z faktu, że osiągnął on punkt, w którym korzyści i możliwości związane z przejściem na nowe narzędzie przekraczają jego koszty i ryzyko.

1.6.5 Metryki narzędzia

Obiektywne metryki z narzędzi są projektowane i gromadzone w oparciu o potrzeby zespołu testowego i innych interesariuszy. Narzędzia testowe w większości przypadków przechwytyują cenne dane w czasie rzeczywistym i ograniczają nakłady pracy związane z gromadzeniem danych. Dane te są wykorzystywane do zarządzania ogólnym nakładem pracy związanym z testowaniem i określania obszarów do optymalizacji.

Różne narzędzia koncentrują się na gromadzeniu różnych rodzajów danych. Na przykład:

- Narzędzia do zarządzania testami mogą dostarczać różnych metryk związanych z dostępnymi elementami testowymi, testami, planowanymi testami, a także bieżącym i zakończonym statusem wykonania testów (np. zaliczony, niezaliczony, pominięty, zablokowany lub planowany).
- Narzędzia do zarządzania wymaganiami zapewniają identyfikowalność w zakresie pokrycia wymagań przez zaliczone i niezaliczone przypadki testowe.

- Narzędzia do zarządzania defektami mogą dostarczać informacji o defektach, takich jak status, krytyczność, priorytet i gęstość defektów w przedmiocie testów. Inne cenne dane, takie jak procent wykrywalności defektów, poziomy testów, na których wprowadzane są defekty, i czas realizacji naprawy wykrytych defektów, pomagają w usprawnianiu procesu, ale nie wszystkie mogą być dostarczane wyłącznie przez narzędzie do zarządzania defektami.
- Narzędzia do analizy statycznej dostarczają między innymi metryk związanych ze złożonością kodu.
- Narzędzia do testowania wydajności mogą dostarczyć cennych informacji, takich jak czasy reakcji i wskaźniki awaryjności przy szczytowym obciążeniu.
- Narzędzia pokrycia kodu pomagają zrozumieć, które części przedmiotu testów zostały sprawdzone za pomocą testów.
- Mimo że narzędzia testowe mogą być wykorzystywane do zbierania metryk, powinny one również monitorować same siebie. W tym kontekście można zmierzyć jakość procesu testowego (np. liczbę defektów znalezionych z wykorzystaniem narzędzi i bez nich oraz pokrycie wymagań).
- Wydajność testów (np. czas wykonywania testów i liczba wykonanych testów).

Więcej szczegółów na temat gromadzenia i wykorzystywania metryk można znaleźć w podrozdziale 2.1 Metryki testów.

2. Zarządzanie produktem – 390 minut

Słowa kluczowe

anomalia, awaria, cel testów, defekt, metryka, postęp testów, przepływ pracy dotyczący defektu, raport o defekcie, szacowanie testów

Słowa kluczowe specyficzne dla domeny

poker planistyczny, szacowanie trójpunktowe, szerokopasmowa technika delficka

Cele nauczania w rozdziale 2:

2.1 Metryki testów

- TM-2.1.1 (K2) Kandydat podaje przykłady metryk umożliwiających osiągnięcie celów testów.
- TM-2.1.2 (K2) Kandydat objaśnia, w jaki sposób kontrolować postęp testów przy użyciu metryk testów.
- TM-2.1.3 (K4) Kandydat analizuje wyniki testów w celu tworzenia raportów z testów, które umożliwiają interesariuszom podejmowanie decyzji.

2.2 Szacowanie testów

- TM-2.2.1 (K2) Kandydat wyjaśnia czynniki, które należy wziąć pod uwagę przy szacowaniu testów.
- TM-2.2.2 (K2) Kandydat podaje przykłady czynników, które mogą mieć wpływ na szacowanie testów.
- TM-2.2.3 (K4) Kandydat wybiera odpowiednią technikę lub podejście do szacowania testów w danym kontekście.

2.3 Zarządzanie defektami

- TM-2.3.1 (K3) Kandydat wdraża proces zarządzania defektami, w tym przepływ pracy dotyczący defektów, który można wykorzystać do monitorowania i kontroli defektów.
- TM-2.3.2 (K2) Kandydat objaśnia proces zarządzania defektami i identyfikuje jego uczestników.
- TM-2.3.3 (K2) Kandydat wyjaśnia specyfikę zarządzania defektami w zwinnym wytwarzaniu oprogramowania.
- TM-2.3.4 (K2) Kandydat wyjaśnia wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania.
- TM-2.3.5 (K3) Kandydat wykorzystuje dane i informacje klasyfikacyjne, które należy gromadzić podczas zarządzania defektami.
- TM-2.3.6 (K2) Kandydat wyjaśnia, w jaki sposób statystyki zgłoszeń defektów można wykorzystać do opracowania usprawnienia procesu.

2.1 Metryki testów

Wprowadzenie – dlaczego warto mieć metryki testów?

W zarządzaniu istnieje powiedzenie: „Co zostanie zmierzone, zostanie zrobione”. Podobnie to, co nie jest mierzone, prawdopodobnie nie zostanie wykonane, ponieważ łatwo to zignorować. Dlatego ważne jest, aby ustalić odpowiedni zestaw metryk na potrzeby każdego przedsięwzięcia, w tym testowania.

Cele testów są odpowiedzią na pytanie, dlaczego testujemy (patrz podrozdział 1.4). Aby określić, czy cele testów zostały osiągnięte, należy określić sposób ich pomiaru. Metryki testów to wskaźniki, które pomagają nam odpowiedzieć na to pytanie.

Metryki testów można podzielić na następujące kategorie:

- **Metryki projektu** mierzą postęp w oparciu o istniejące kryteria wyjścia projektu, takie jak odsetek testów przeprowadzonych, zaliczonych i niezaliczonych;
- **Metryki produktu** mierzą atrybuty produktu, takie jak stopień, w jakim produkt spełnia oczekiwania jakościowe przewidywanych użytkowników;
- **Metryki procesu** mierzą zdolność procesu testowego i skuteczność testowania. Metryki procesów są zatem wykorzystywane do raportowania efektywności i wydajności związanej z procesami.

Więcej informacji na temat zarządzania metrykami produktu i procesu można znaleźć w sylabusie ISTQB® Poziom Ekspercki Zarządzanie Testami.

Więcej informacji na temat wykorzystania metryk procesu można znaleźć w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie usprawnień procesu testowego.

W poniższych sekcjach omówiono metryki dotyczące planowania testów, monitorowania testów, nadzoru nad testami i ukończenia testów. Są to cztery główne czynności związane z zarządzaniem metrykami.

2.1.1 Metryki czynności związanych z zarządzaniem testami

Sylabus Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami koncentruje się na następujących ogólnych czynnościach związanych z zarządzaniem testami:

- planowanie testów;
- monitorowanie testów i nadzór nad testami;
- ukończenie testów (patrz podrozdział 1.1).

Kierownictwo testów musi być w stanie określić zestaw metryk testów do monitorowania testów, nadzoru nad testami i ich ukończenia w ramach działań związanych z planowaniem testów. Każda metryka musi być określona, mierzona, monitorowana i raportowana.

Podczas planowania testów definiuje się odpowiednie metryki testów, które odpowiadają celom testów określonym w projektowej strategii testowej.

Metryki używane podczas monitorowania i nadzoru nad testami mogą się różnić od tych używanych przy ich ukończeniu. Podczas monitorowania i nadzoru nad testami metryki dotyczą postępu czynności testowych. Po ukończeniu testów cele testów powinny zostać osiągnięte. Można połączyć jedną lub więcej metryk, aby zmierzyć kryteria wyjścia testu przypisane do danych celów testu.

Poniższa tabela zawiera przykłady metryk (jest ich znacznie więcej) wykorzystywanych w czynnościach związanych z zarządzaniem testami:

Metryka (planowana/monitorowana dla określonych kamieni milowych)	Monitorowanie i nadzór nad testami	Ukończenie testów
Pokrycie wymagań	X	X
Pokrycie ryzyka produktowego	X	X
Pokrycie kodu	X	
Rzeczywiste a planowane oszacowanie (w godzinach) czynności testowych	X	
Odsetek wykonanych przypadków testowych według statusu (np. niezaliczone, zablokowane) w porównaniu z planowanymi przypadkami testowymi	X	X
Skumulowana liczba usuniętych defektów w stosunku do łącznej liczby defektów	X	
Rzeczywiste zautomatyzowane przypadki testowe w stosunku do planowanych zautomatyzowanych przypadków testowych		X
Rzeczywisty a planowany koszt testów	X	

Tabela 2: Przykłady metryk stosowanych w czynnościach związanych z zarządzaniem testami

Metrykę zastosowaną w konkretnej czynności testowej przedstawiono w tabeli. Metryki ze znakiem X przy monitorowaniu i nadzorze nad testami są wykorzystywane głównie do pomiaru postępów i są zgłaszane w raportach o postępie testów (CTFL). Metryki ze znakiem X przy ukończeniu testów są przede wszystkim wykorzystywane do pomiaru osiągnięcia celów testów i są zgłaszane w sumarycznym raporcie z testów (CTFL). Metryki ze znakiem X w obu kolumnach mogą być używane w obu przypadkach.

Istnieją również metryki do monitorowania skuteczności testów (np. procent wykrywania defektów; ang. defect detection percentage – DDP).

DDP omówiono w sylabusie ISTQB® na poziomie eksperckim Wdrażanie usprawnień procesu testowego, w szczególności w sekcji poświęconej metrykom efektywności testów (akapit dotyczący DDP).

2.1.2 Monitorowanie, nadzór i ukończenie testów

Metryki testów to wskaźniki, które pokazują, jak daleko posunął się test i czy kryteria wyjścia lub powiązane zadania testowe zostały osiągnięte.

Monitorowanie testów to działanie polegające na gromadzeniu danych dotyczących testów oraz związanej z nimi ocenie i oszacowaniu. Służy do oceny postępów testów i sprawdzenia realizacji kryteriów wyjścia lub powiązanych czynności testowych (patrz sekcja 1.1.2). Kryteria wyjścia są wprowadzane z celów testów.

Nadzór nad testami wykorzystuje informacje z monitorowania testów, aby zapewnić wskazówki i działania naprawcze w celu osiągnięcia skutecznego i wydajnego testowania. Przykłady dyrektyw dotyczących nadzoru nad testami obejmują zmianę priorytetów testów, gdy wykryte ryzyko stało się problemem, ponowną ocenę, czy element testowy spełnia kryteria wejścia lub wyjścia z powodu przeróbek, dostosowanie harmonogramu testów w celu uwzględnienia opóźnienia w dostarczeniu środowiska testowego oraz dodawanie nowych zasobów w razie potrzeby.

Ukończenie testów polega na zebraniu danych pochodzących z wykonanych czynności testowych w celu skonsolidowania wyciągniętych wniosków, testaliów i innych istotnych informacji. Ukończenie testów ma miejsce w kamieniach milowych projektu, takich jak ukończenie poziomu testów, ukończenie iteracji, ukończenie (lub anulowanie) projektu testowego, wydanie produktu lub ukończenie wydania pielęgnacyjnego.

Typowe metryki testów wykorzystywane w czynnościach związanych z zarządzaniem testami obejmują metryki postępu projektu i metryki, które pokazują postęp w stosunku do planowanego harmonogramu i budżetu, bieżącą jakość elementów testowych oraz skuteczność testowania w stosunku do celów testów lub celów iteracji.

2.1.3 Raportowanie testów

Kierownictwo testów powinno rozumieć, jak interpretować i wykorzystywać metryki w celu zrozumienia i raportowania statusu testów. W przypadku wyższych poziomów testów, takich jak testowanie systemowe, testowanie integracji systemów, testowanie akceptacyjne i testowanie zabezpieczeń, główną podstawą testów są zazwyczaj produkty pracy, takie jak specyfikacje wymagań, przypadki użycia, historyjki użytkowników i ryzyka produktowe. Metryki pokrycia strukturalnego mają większe zastosowanie do niższych poziomów testów, takich jak testowanie modułowe (np. pokrycie instrukcji) i testowanie integracji modułów (np. pokrycie interfejsu). Podczas gdy w zarządzaniu testami można wykorzystywać metryki pokrycia kodu do pomiaru zakresu, w jakim ich testy badają strukturę testowanego systemu, raportowanie wyników testów wyższego poziomu powinno być dostosowane do konkretnego kontekstu i potrzeb projektu. Na przykład w często zmieniających się środowiskach metryki pokrycia kodu mogą być przydatne do monitorowania wpływu zmian kodu na zestaw testowy i określania potencjalnych luk lub ryzyk. Ponadto kierownictwo testów musi rozumieć, że nawet jeśli testy modułów i testy integracji modułów osiągną 100% pokrycia strukturalnego, defekty i ryzyka jakościowe pozostaną do wyeliminowania na wyższych poziomach testów.

Celem raportowania metryk jest zapewnienie natychmiastowego zrozumienia informacji do celów zarządzania. Metryki mogą być raportowane jako migawka metryki w danym momencie lub jako ewolucja metryki w czasie w celu oceny trendów.

Ryzyka produktowe, defekty, postęp testów, pokrycie i powiązane koszty oraz nakład pracy związany z testami są mierzone i raportowane w określony sposób na zakończenie projektu.

Poniżej przedstawiono przykłady metryk, które można wykorzystać do różnych celów:

Metryki związane z ryzykami produktowymi:

- odsetek ryzyk, które przeszły pomyślnie wszystkie testy;
- odsetek ryzyk, w przypadku których niektóre lub wszystkie testy zakończyły się niepowodzeniem;
- odsetek ryzyk, które nie zostały jeszcze w pełni przetestowane.

Metryki te można wykorzystać do oceny jakości podstawy testów i skuteczności przypadków testowych w pokrywaniu ryzyka produktowego.

Metryki związane z defektami:

- skumulowana liczba usuniętych defektów w stosunku do łącznej liczby defektów;
- podział liczby lub odsetka defektów na kategorie:
 - o elementy lub komponenty testowe;
 - o źródło defektu (np. specyfikacja wymagań, nowa funkcja lub regresja);

- o wersja testu, który wykrył defekt;
- o wprowadzony, wykryty i usunięty poziom lub iteracja testu;
- o priorytet/krytyczność;
- o przyczyna podstawowa;
- o status (np. odrzucony, zduplikowany, otwarty, zamknięty).

Metryki te mogą być wykorzystywane do monitorowania procesu wykrywania i usuwania defektów, wykrywania obszarów o wysokiej gęstości defektów lub ich krytyczności oraz oceny wydajności i skuteczności testów.

Metryki związane z postępem testów:

- status wykonania testów: łączna liczba zaplanowanych, wdrożonych, wykonanych, zaliczonych, niezaliczonych, zablokowanych i pominiętych testów;
- nakład pracy związany z testowaniem: liczba rzeczywistych i planowanych godzin zasobów poświęconych na testowanie.

Metryki związane z pokryciem:

- pokrycie wymagań: odsetek wymagań, które są pokryte przypadkami testowymi;
- pokrycie ryzyka produktowego: odsetek wykrytych ryzyk produktowych, które są łagodzone przez przypadki testowe;
- pokrycie kodu: odsetek instrukcji kodu, gałęzi, ścieżek lub warunków, które są wykonywane przez przypadki testowe.

Metryki związane z kosztami i nakładem pracy związanym z testowaniem:

- ryzyka rezydualne (resztkowe) w przypadku nieprzetestowanych modułów: potencjalny wpływ i prawdopodobieństwo wystąpienia defektów w modułach, które nie są testowane;
- koszt testów: rzeczywisty a planowany koszt testów.

Dodatkowo przydatne jest łączenie metryk z różnych kategorii (np. metryki pokazujące korelację między trendami otwartych defektów a trendami wykonanych testów lub metryki pokazujące jakość podstawy testów na podstawie liczby defektów znalezionych w wymaganiach). Gdy wykonywanie testów jest kontynuowane i wykrywanych jest coraz mniej defektów, można podjąć decyzję o ukończeniu testów. Decyzja ta powinna opierać się na raportowaniu metryk i uzgodnionych kryteriach wyjścia.

2.2 Szacowanie testów

Wstęp

Istnieją najlepsze praktyki zarządzania projektami w zakresie szacowania inżynierii systemów i oprogramowania, dotyczące wszystkich rodzajów zasobów (np. kosztów, ludzi lub czasu). Szacowanie testów to zastosowanie tych najlepszych praktyk do testowania związanego z projektem lub działaniem.

2.2.1 Oszacowanie, jakie czynności będzie obejmować testowanie

Szacowanie testów to czynność związana z zarządzaniem testami, która pozwala oszacować, ile czasu, nakładu pracy (wysiłku) i kosztów zajmie wykonanie danego zadania. Szacowanie testów jest jednym z głównych i ważnych zadań w zarządzaniu testami.

Główne cechy szacowania w zarządzaniu testami to:

- **Nakład pracy** jest zwykle obliczany w osobogodzinach lub punktach historyjek wymaganych do ukończenia zadań testowych projektu. Często nakład pracy związany z testowaniem i czas trwania testów (czas, który upłynął) mogą być różne, a zarządzanie testami może wymagać oszacowania całkowitego czasu trwania aktywności. Ile osobogodzin to zajmie?
- **Czas** wymagany do ukończenia projektu. Czas jest krytycznym zasobem w projekcie. Planowanie testów wymaga oszacowania nakładu pracy związanego z testowaniem w dniach kalendarzowych i roboczych. Każdy projekt ma kamienie milowe i termin realizacji. Jak długo potrwa ukończenie projektu testowego?
- **Koszt** to budżet projektu. Obejmuje wydatki na zasoby testowe, narzędzia i infrastrukturę. Jaki będzie koszt projektu testowego?

Testowanie jest często podprojektem w ramach (dużego) projektu, czasami rozproszonym w kilku lokalizacjach testowych (np. centrach testowych). Aby przeprowadzić szacowanie testów, pierwszym krokiem jest określenie poziomów testów, czynności testowych i zadań testowych. Następnie należy podzielić projekt testowy na główne czynności testowe (np. planowanie i wykonywanie testów) w ramach procesu testowego (patrz sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0). W projektach zwinnych czynności testowe są często szacowane w ramach prac wytwórczych, a nie jako odrębne wartości. Kolejnym krokiem jest oszacowanie nakładu pracy związanego z testowaniem wymaganego do ukończenia zadań lub produktów pracy oraz oczekiwanych kosztów z tym związanych.

Ponieważ testowanie jest częścią projektu, zawsze istnieją pewne naturalne ograniczenia projektowe, które na niego wpływają i wymagają kompromisu, a my nie możemy arbitralnie manipulować tymi wartościami. Można to znaleźć w zarządzaniu jakością jako trójkąt czas-koszt-jakość. W zarządzaniu projektami trójkąt czas-koszt-jakość obejmuje trzy wartości, które są współzależne, co oznacza, że są ze sobą ściśle powiązane i wzajemnie na siebie wpływają. Zależność ta jest powszechnie obserwowana w scenariuszach projektów.

2.2.2 Czynniki, które mogą wpływać na nakład pracy związany z testowaniem

Szacowanie wysiłku testowego polega na przewidywaniu ilości prac związanych z czynnościami testowymi, które będą potrzebne do osiągnięcia celów testów w odniesieniu do konkretnego projektu, wydania lub iteracji. Czynniki wpływające na nakład pracy związany z testowaniem mogą mieć następujące cechy:

Produkt:

- jakość podstawy testów;
- rozmiar testowanego produktu (tj. przedmiotu testów);
- złożoność domeny produktu (np. środowisko, infrastruktura i historia);
- wymagania dotyczące testowania charakterystyk jakościowych (np. zabezpieczeń i niezawodności).

Te czynniki związane z produktem mogą wpływać na szacunki dotyczące testów, ponieważ tworzą kontekst specyficzny dla systemu podlegającego testowaniu.

Proces wytwarzania:

- stabilność i dojrzałość procesów wytwarzania organizacji;
- stosowany model wytwarzania oprogramowania (np. zwinny/iteracyjny lub hybrydowy);
- czynniki materialne (np. dostępność automatyzacji testów, narzędzi i środowisk testowych).

Te czynniki związane z procesem wytwórczym mogą wpływać na szacunki dotyczące testów, ponieważ testowanie jest bezpośrednio związane z wytwarzaniem.

Ludzie:

- zadowolenie ludzi (np. ze względu na dni ustawowo wolne od pracy, urlopy, inne oczekiwane korzyści);
- umiejętności i doświadczenie zaangażowanych osób, zwłaszcza w odniesieniu do podobnych projektów i produktów (np. znajomość domeny).

Ludzie są najbardziej niezbędnym zasobem, dlatego należy wziąć pod uwagę każdą niestabilność. W związku z tym ludzie są ważnym czynnikiem w szacowaniu nakładu pracy związanego z testowaniem. Patrz też podrozdział 3.1.

Wyniki testów:

- liczba i krytyczność defektów wykrytych podczas wykonywania testów;
- wymagana ilość przeróbek.

Statystyki historyczne wspierają szacowanie testów. W związku z tym znajomość tych czynników pomoże w oszacowaniu dokładniejszych wartości.

Kontekst testów:

- rozłożenie testów na kilka oddziałów, skład i lokalizacja zespołów, złożoność projektu (np. wiele podsystemów);
- rodzaj pracy (np. wirtualna lub na miejscu).

Czynniki związane z kontekstem to te, które wpływają na całe szacowanie testów. Patrz też podrozdział 1.2.

2.2.3 Wybór technik szacowania testów

Szacowanie testów powinno obejmować wszystkie czynności związane z procesem testowym. Szacowany koszt, nakład pracy, a zwłaszcza czas wykonywania testów jest często najważniejszy z punktu widzenia zarządzania testami, ponieważ wartości te będą miały wpływ na projekt. Mimo to szacunki dotyczące wykonywania testów są zwykle trudne do wykonania, gdy ogólna jakość oprogramowania jest niska lub nieznana. Ponadto znajomość i doświadczenie związane z produktem

prawdopodobnie wpłynie na jakość szacunków. Powszechną praktyką jest szacowanie liczby przypadków testowych na podstawie podstawy testów (np. wymagań lub historyjek użytkowników). Założenia przyjęte podczas szacowania testów powinny być zawsze udokumentowane jako część oszacowania.

Techniki lub podejścia do szacowania testów można podzielić na oparte na metrykach i oparte na ekspertach. Dalsze szczegóły dotyczące technik szacowania testów zostały wyjaśnione w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0.

W większości przypadków oszacowanie, po jego przygotowaniu, musi zostać dostarczone kierownictwu projektu wraz z uzasadnieniem. Często niektóre parametry wejściowe ulegają zmianie (np. zakres testów), co często skutkuje dostosowaniem oszacowania. W idealnej sytuacji ostateczne oszacowanie testów stanowi najlepszą możliwą równowagę celów organizacyjnych i projektowych w obszarach jakości, harmonogramu, budżetu i funkcji.

Należy pamiętać, że wszelkie szacunki opierają się na informacjach dostępnych w momencie ich przygotowania. Na wczesnym etapie projektu informacje mogą być dość ograniczone. Ponadto informacje te mogą z czasem ulec zmianie. Aby zachować dokładność, szacunki powinny być aktualizowane w celu odzwierciedlenia nowych i zmieniających się informacji.

Wybór techniki szacowania zależy od różnych czynników, takich jak:

- **Błąd oszacowania:** niektóre techniki umożliwiają obliczenie odchylenia standardowego, które jest miarą niepewności lub zmienności oszacowania. Na przykład technika szacowania trójpunktowego wykorzystuje optymistyczne, pesymistyczne i najbardziej prawdopodobne szacunki do obliczenia wartości oczekiwanej i odchylenia standardowego szacunku (patrz sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0, sekcja 5.1.4).
- **Dostępność danych:** niektóre techniki wymagają danych historycznych z poprzednich lub podobnych projektów, które mogą nie być dostępne lub wiarygodne. Na przykład szacunki oparte na wskaźnikach i ekstrapolacji opierają się na danych historycznych w celu uzyskania wskaźników lub trendów na potrzeby bieżącego projektu.
- **Dostępność ekspertów:** niektóre techniki wymagają zaangażowania ekspertów, którzy mają wiedzę i doświadczenie, aby zapewnić dokładne i realistyczne szacunki. Na przykład szerokopasmowa technika delficka i poker planistyczny opierają się na opiniach i osądach ekspertów lub członków zespołu.
- **Wiedza z zakresu modelowania:** niektóre techniki wymagają użycia modeli matematycznych lub wzorów do obliczenia szacunków, co może wymagać pewnych umiejętności i wiedzy w zakresie modelowania. Na przykład ekstrapolacja i szacowanie trójpunktowe wykorzystują wzory do uzyskania wartości oczekiwanej i odchylenia standardowego szacunku.
- **Ograniczenia czasowe:** niektóre techniki wymagają więcej czasu i nakładu pracy niż inne, co może wpływać na ich wykonalność i przydatność.

Pokazuje to, że kryteria wyboru właściwych technik szacowania testów są w dużym stopniu zależne od kontekstu testowania (np. SDLC, interesariusze, poziomy testów i typy testów stosowane w projekcie, patrz podrozdział 1.2). Kierownik testów musi być w stanie koordynować i stosować techniki szacowania testów (np. z różnymi modelami SDLC w jednym projekcie w różnych oddziałach). Na przykład, aby wybrać odpowiednią technikę szacowania, należy najpierw określić złożoność problemu. Jeśli złożoność jest niska, można zastosować techniki oparte na metrykach. Jeśli złożoność jest wysoka, można zastosować techniki oparte na ekspertach. Jeśli stosowany jest sekwencyjny model

wytwarzania, można zastosować szerokopasmową technikę delficką. Jeśli używany jest zwinny model wytwarzania oprogramowania, można zastosować poker planistyczny.

2.3 Zarządzanie defektami

Wstęp

W sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0 opisano działania, które rozpoczynają się po zaobserwowaniu rzeczywistych wyników, które różnią się od oczekiwanych. W sylabusie działania te nazywa się zarządzaniem defektami. Inne normy używają terminu „zarządzanie incydentami” (ISO/IEC/IEEE 29119-3) lub „zarządzanie anomaliami” (TMAP), aby podkreślić fakt, że na początku procesu możemy nie wiedzieć, czy rozbieżność jest spowodowana defektem produktu pracy, czy też czymś innym (np. błędem automatyzacji testów lub niezrozumieniem wymagań przez testera). Zarządzanie defektami i narzędzie używane do zarządzania defektami mają kluczowe znaczenie dla testerów i innych członków zespołu zaangażowanych w wytwarzanie oprogramowania. Informacje pochodzące ze skutecznego procesu zarządzania defektami pozwalają zespołowi testowemu i innym interesariuszom projektu uzyskać wgląd w stan projektu w całym jego cyklu SDLC. Zarządzanie defektami ma również kluczowe znaczenie przy podejmowaniu decyzji o tym, które defekty zostaną naprawione. Zapewnia to, że nakłady będą poświęcane na pracę z odpowiednimi defektami. Gromadzenie i analizowanie danych związanych z defektami w czasie może pomóc w zlokalizowaniu obszarów potencjalnej poprawy zarówno w przypadku testowania, jak i innych procesów w ramach SDLC (np. lepszego zapobiegania defektom poprzez ulepszoną architekturę i projekt techniczny).

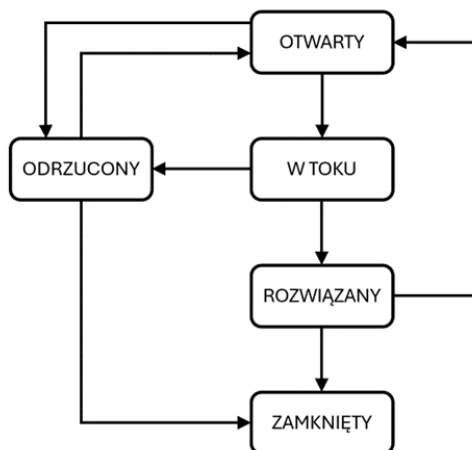
Oprócz zrozumienia ogólnego cyklu życia defektów i sposobu jego wykorzystania do monitorowania i nadzoru zarówno procesów wytwarzania oprogramowania, jak i procesów testowania, kierownik testów i testerzy (lub cały zwinny zespół w zwinnym wytwarzaniu oprogramowania) muszą również być zaznajomieni z tym, które dane mają kluczowe znaczenie dla ich analizy. Kierownik testów musi być zwolennikiem właściwego wykorzystania zarówno procesu zarządzania defektami, jak i wybranego narzędzia do zarządzania defektami.

2.3.1 Cykl życia defektu

Każda faza SDLC powinna obejmować działania mające na celu wykrywanie i usuwanie potencjalnych defektów. Na przykład techniki testowania statycznego (tj. przeglądy i analiza statyczna) mogą być stosowane do specyfikacji projektowych, specyfikacji wymagań i kodu przed dostarczeniem tych produktów pracy w celu podjęcia kolejnych działań. Im wcześniej każdy defekt zostanie wykryty i usunięty, tym niższy będzie ogólny koszt jakości produktu. Koszt jakości jest minimalizowany, gdy każdy defekt jest usuwany w tej samej fazie, w której został wprowadzony (tj. gdy proces oprogramowania osiąga idealne powstrzymanie fazowe).

Podczas testowania statycznego szukamy defektów. Podczas testowania dynamicznego obecność defektu jest ujawniana, gdy powoduje on awarię, co skutkuje rozbieżnością między rzeczywistymi a oczekiwanymi wynikami testu (tj. anomalią). W niektórych przypadkach wynik fałszywie negatywny występuje, gdy tester nie zaobserwuje anomalii. W przypadku zaobserwowania anomalii należy przeprowadzić dalsze badania. Badania te zwykle rozpoczynają się od wypełnienia raportu o defekcie zgodnie z określonym procesem testowym i zarządzania defektami. Niezaliczony test nie zawsze skutkuje utworzeniem raportu o defekcie (np. w wytwarzaniu sterowanym testami, gdzie testy modułów, zwykle zautomatyzowane, są używane jako forma wykonywalnej specyfikacji projektu). Do czasu zakończenia opracowywania modułu niektóre lub wszystkie testy mogą początkowo zakończyć się niepowodzeniem. Dlatego wynik takiego testu niekoniecznie jest spowodowany defektem i zazwyczaj nie jest śledzony za pomocą raportu o defekcie.

Raport o defekcie postępuje zgodnie z przepływem pracy (dla uproszczenia i spójności z większością narzędzi do zarządzania defektami będziemy dalej używać terminu „przepływ pracy dotyczący defektu”) i przechodzi przez sekwencję stanów defektu. W większości tych stanów jedna osoba jest właścicielem raportu o defekcie i jest odpowiedzialna za wykonanie zadania (np. analizę, usunięcie defektu lub test potwierdzający). Na poniższym schemacie przedstawiono prosty przepływ pracy dotyczący defektu:



Rys. 2: Prosty przepływ pracy dotyczący defektu

Prosty przepływ pracy dotyczący defektu może obejmować następujące stany defektu:

- OTWARTY (może być nazywany NOWYM): stan początkowy podczas tworzenia raportu o defekcie.
- W TOKU: zespół pracuje nad analizą raportu o defekcie i/lub naprawą.
- ODRZUCONY: raport o defekcie jest odrzucany przez osobę, która go przetworzyła (zwykle programistę lub analityka). Przyczyn odrzucenia może być wiele (np. nieprawidłowe informacje, nieprawidłowy test, zduplikowany raport o defekcie), a informacje te są dodawane do raportu o defekcie.
- ROZWIĄZANY (może być nazywany STAŁYM, GOTOWYM DO RETESTU): tester przeprowadza test potwierdzający, często wykonując kroki w celu odtworzenia awarii z raportu o defekcie, aby ustalić, czy poprawka rzeczywiście usunęła defekt.
- ZAMKNIĘTY: raport o defekcie osiągnął stan końcowy i nie przewiduje się dalszych prac. Tester przenosi raport o defekcie do tego stanu po pomyślnym teście potwierdzającym lub w celu potwierdzenia odrzucenia raportu o defekcie.

W wielu organizacjach stosowany jest prosty przepływ pracy dotyczący defektu, który jest rozszerzany o inne stany defektów istotne dla danego kontekstu (np. PONOWNIE OTWARTY, ZAAKCEPTOWANY, DOPRECYZOWANY lub ODROZCZONY).

Przepływ pracy dotyczący defektu może się różnić w różnych organizacjach pod względem różnych nazw stanów defektu, zasad przechodzenia między stanami defektu i ról odpowiedzialnych za zadania przy poszczególnych stanach defektu. Często przepływ pracy dotyczący defektu jest prostszy w zwinnym wytwarzaniu oprogramowania niż w sekwencyjnych modelach wytwarzania. Przepływ pracy dotyczący defektu powinien być dostosowany do danego kontekstu. Podczas projektowania przepływu pracy dotyczącego defektu zaleca się przestrzeganie kilku dobrych praktyk:

- Jeśli to możliwe, przepływ pracy dotyczący defektu powinien być zdefiniowany w całej organizacji, aby zapewnić jednolite zarządzanie defektami we wszystkich projektach.

- Defekty zduplikowane i fałszywie pozytywne powinny być reprezentowane przez oddzielny stan lub połączenie statusu ODRZUCONY z wyborem przyczyny odrzucenia. Mogą one być pomocne w dalszych analizach defektów w celu usprawnienia procesu testowego.
- Zaleca się stosowanie tylko jednego stanu końcowego (np. ZAMKNIĘTY). Przejście do tego stanu często wymaga wybrania powodu zamknięcia, przydatnego do oceny procesu i czynności związanych z jego doskonaleniem.
- Nazwy stanów w przepływie pracy dotyczącym defektu powinny być takie same jak dla analogicznych stanów innych jednostek (np. historyjek użytkowników i zadań testowych), aby uprościć pracę nad nimi.
- Kolejne stany defektu powinny należeć do różnych odpowiedzialnych ról. Jeśli dwa lub więcej kolejnych stanów należy do tej samej odpowiedzialnej roli, powinien istnieć dobry powód (np. do pomiaru czasu spędzonego w stanie defektu).
- Każdy stan defektu, z wyjątkiem stanu końcowego, powinien mieć więcej niż jedno przejście wychodzące, aby umożliwić odpowiedzialnej roli podjęcie decyzji dotyczącej następnego kroku. Wyjątki od tej zasady powinny być uzasadnione (np. w celu monitorowania czasu spędzonego na danej czynności).
- Zbiór atrybutów, które należy wprowadzić podczas wykonywania przejścia do stanu, powinien być ograniczony do tych, które nadają znaczną wartość zarządzaniu defektami.

2.3.2 Międzyfunkcyjne zarządzanie defektami

Mimo że organizacja testowa i kierownik testów często są właścicielami ogólnego procesu zarządzania defektami i narzędzia do zarządzania defektami, zespół wielofunkcyjny jest zazwyczaj odpowiedzialny za zarządzanie defektami w danym projekcie. Zespół ten, czasami nazywany komitetem ds. zarządzania defektami, może obejmować kierownika testów, przedstawicieli działu rozwoju, dostawców, kierownictwa projektu, kierownictwa produktu lub właściciela produktu oraz innych interesariuszy, którzy są zainteresowani testowanym oprogramowaniem.

W miarę wykrywania anomalii i wprowadzania ich do narzędzia do zarządzania defektami komitet ds. zarządzania defektami powinien ustalać, czy każdy raport o defekcie prezentuje rzeczywisty defekt i czy powinien on zostać naprawiony (oraz przez którą stronę w przypadku, gdy w dostawie uczestniczy kilka zespołów wytwórczych), odrzucony, czy odroczony. Decyzja ta wymaga, aby komitet ds. zarządzania defektami rozważył korzyści, ryzyko i koszty związane z usunięciem defektu. Korzystne jest omówienie defektów na spotkaniu (często nazywanym spotkaniem triażowym). Jeśli defekt ma zostać usunięty, zespół powinien ustalić priorytet usunięcia defektu w stosunku do innych zadań. Można skonsultować się z kierownikiem testów i zespołem testowym w sprawie względnego znaczenia defektu i powinni oni dostarczyć dostępnych obiektywnych informacji.

W bardzo dużych projektach powołanie pełnoetatowego kierownika ds. defektów może być uzasadnione nakładem pracy potrzebnym do przygotowania się i monitorowania decyzji podejmowanych na spotkaniach komitetu ds. zarządzania defektami, przynajmniej podczas tych faz SDLC, w których testowanie jest najbardziej intensywne. W innych sytuacjach kilka dużych projektów może współdzielić kierownika ds. defektów.

Narzędzie do zarządzania defektami nie powinno być używane jako substytut dobrej komunikacji, ani komitet ds. zarządzania defektami nie powinien być używany jako substytut skutecznego korzystania z dobrego narzędzia do zarządzania defektami. Komunikacja, odpowiednie wsparcie narzędziowe, dobrze zdefiniowany przepływ pracy dotyczący defektu (w tym właściwości raportów o defektach) i zaangażowany zespół ds. zarządzania defektami są niezbędne do skutecznego i wydajnego zarządzania defektami.

2.3.3 Specyfika zarządzania defektami w zespołach zwinnych

Zarządzanie defektami w organizacjach korzystających ze zwinnego wytwarzania oprogramowania jest często lżejsze, uproszczone i/lub mniej formalne niż w sekwencyjnych modelach wytwarzania. Jeśli zespoły zwinne są zlokalizowane w jednym miejscu lub mają efektywne środki komunikacji, informacje o defekcie lub awarii są często wymieniane między testerami, przedstawicielami klientów i programistami bez formalnego raportu o defekcie. Raporty o defektach należy jednak sporządzać w przypadku:

- defektów, które blokują inne bieżące działania sprintu (tj. wytwarzanie, testowanie lub inne) i nie mogą być natychmiast naprawione w zespole zwinnym;
- defektów, których nie można usunąć w ramach tej samej iteracji; niektóre zespoły zwinne mają zasadę tworzenia raportu o defekcie, jeśli defekt nie może zostać usunięty w dniu, w którym wykryto awarię;
- defektów, które muszą być rozwiązane przez lub we współpracy z innymi zespołami w organizacjach wielozespołowych;
- defektów, które muszą zostać usunięte przez dostawcę;
- defektów, w przypadku których raport o defekcie jest wyraźnie wymagany (np. gdy programista nie może natychmiast pracować nad poprawką).

Powszechną praktyką jest dodawanie defektów, których nie można usunąć w ramach tej samej iteracji, do backlogu produktu, aby można było nadać im priorytet wśród innych defektów i historyjek użytkowników w na potrzeby późniejszej iteracji.

Mimo że podstawy zarządzania defektami powinny być określone w organizacyjnej strategii testowej, wiele aspektów, w tym poziom sformalizowania, wyzwalacze tworzenia raportu o defekcie i atrybuty defektu, które należy określić, można pozostawić do uzgodnienia między członkami zespołu zwinnego. Ogólnie rzecz biorąc, poziom sformalizowania zarządzania defektami i podejście do tworzenia raportów o defektach powinny odzwierciedlać kwestie takie jak:

- lokalizacja członków zespołu w jednym miejscu;
- rozmieszczenie członków zespołu w różnych strefach czasowych;
- liczba zespołów, które współpracują przy wytwarzaniu produktu;
- dojrzałość zespołu(-ów);
- wielkość zespołu(ów);
- ryzyka produktowe;
- wymogi regulacyjne, umowne lub inne (jeśli dotyczy).

Ostateczna decyzja zespołu zwinnego dotycząca szczegółów zarządzania defektami powinna być zawsze udokumentowana (np. za pomocą wytycznych w narzędziu do zarządzania wiedzą).

2.3.4 Wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania

W praktyce wiele zespołów często współpracuje ze sobą przy dostarczaniu systemu lub systemu systemów. Przykłady obejmują hybrydowe modele wytwarzania oprogramowania, gdy klient korzysta ze zwinnego wytwarzania oprogramowania, a jeden z jego dostawców korzysta z sekwencyjnego modelu wytwarzania, lub gdy organizacja korzystająca z sekwencyjnego modelu wytwarzania wymaga dostarczenia podsystemu od zespołu korzystającego ze zwinnego wytwarzania oprogramowania. Takie środowisko wielozespołowe stawia różne wyzwania:

- **Dostosowanie atrybutów defektów i narzędzi wykorzystywanych do zarządzania defektami:** w idealnym scenariuszu wszystkie zespoły korzystają z jednego narzędzia do zarządzania defektami. W praktyce często zdarza się, że każdy zespół korzysta z innego narzędzia do zarządzania defektami, zwłaszcza gdy kilka zespołów dostawców uczestniczy w realizacji projektu. W takich przypadkach dobrze jest ustanowić synchronizację między narzędziami do zarządzania defektami (najlepiej automatyczną).
- **Priorytetyzacja defektów:** właściciel(e) produktu powinni być włączeni w spotkania dotyczące zarządzania defektami i aktywnie poszukiwać informacji o konsekwencjach i ryzykach związanych z defektami. Spotkania dotyczące zarządzania defektami powinny odbywać się częściej w przypadku zwinnego wytwarzania oprogramowania niż w przypadku sekwencyjnych modeli wytwarzania, aby dotrzymać kroku szybszemu tempu dostarczania przyrostów produktu przez zespół zwinny. Spotkania te mogą być jednak krótsze w przypadku zespołów zwinnych. Czasami korzystne jest, aby mniejsza grupa interesariuszy zajmujących się zarządzaniem defektami miała ostateczne słowo w kwestii ustalania priorytetów defektów.
- **Dostosowanie i przejrzystość planu testów na potrzeby nowego wytwarzania i napraw defektów:** praca wszystkich zespołów powinna być dostosowana do tego samego planu projektu, niezależnie od tego, czy korzystają one ze zwinnych, czy sekwencyjnych modeli wytwarzania oprogramowania. Wszystkie rezultaty, w tym poprawki defektów, powinny być dostosowane do tego planu projektu. Lepsze dostosowanie można osiągnąć poprzez aktywne uczestnictwo członków wszystkich zespołów w procesie planowania (np. udział zespołów zajmujących się sekwencyjnym modelem wytwarzania w spotkaniach dotyczących zwinnego wytwarzania oprogramowania, podczas których omawiane są defekty i ustalone ich priorytety). Przejrzystość planów wytwarzania można poprawić, udostępniając je między zespołami (np. za pośrednictwem tablic wskaźników lub backlogu produktu).

2.3.5 Informacje zawarte w raporcie o defekcie

Informacje zawarte w raporcie o defekcie powinny wystarczyć do następujących celów:

- zarządzanie raportem o defekcie w cyklu życia defektu;
- ocena ogólnego statusu projektu, zwłaszcza pod względem jakości produktu i postępów w testach;
- ocena statusu przyrostu produktu pod względem jakości produktu;
- ocena zdolności procesu.

Informacje potrzebne do zarządzania defektami i statusem projektu mogą się różnić w zależności od tego, kiedy defekt zostanie wykryty w SDLC. Ponadto raporty o defektach związane z нефункциональными характеристиками jakościowymi mogą wymagać więcej informacji (np. warunki obciążenia w przypadku problemów z wydajnością). Zebrane podstawowe informacje powinny jednak być spójne w całym SDLC, a najlepiej we wszystkich projektach w organizacji, aby umożliwić znaczące porównanie danych o defektach w całym projekcie i we wszystkich projektach.

W raporcie o defekcie można zebrać wiele elementów danych. Kierownik testów powinien zdecydować, które informacje są odpowiednie do skutecznego zarządzania defektami w kontekście danego projektu. Ze względu na fakt, że każdy dodatkowy atrybut wydłuża czas poświęcany na raportowanie defektów i może zwiększać dezorientację osoby, która wprowadza raport o defekcie, zaleca się gromadzenie tylko tych danych, które są potrzebne do zarządzania defektami w danym kontekście i/lub będą wykorzystywane do doskonalenia procesu.

Aby zarządzać raportem o defekcie w większości środowisk, obowiązkowe są następujące elementy:

- tytuł defektu z krótkim podsumowaniem anomalii;
- szczegółowy opis anomalii, najlepiej zawierający kroki do odtworzenia awarii;
- krytyczność wpływu na testowany system i/lub interesariuszy produktu;
- priorytet usunięcia anomalii.

Narzędzie do zarządzania defektami często tworzy dodatkowe ważne elementy danych:

- unikalny identyfikator raportu o defekcie;
- datę/godzinę utworzenia raportu o defekcie;
- imię i nazwisko osoby, która odkryła i/lub zgłosiła anomalię;
- fazę projektu i SDLC, w której wykryto anomalię;
- bieżący stan raportu o defekcie;
- aktualnego właściciela (tj. osobę aktualnie przypisaną do pracy nad defektem);
- historię zmian, takich jak sekwencja czynności, wraz z informacjami o dacie/godzinie, podjętych przez członków zespołu projektowego działaniach w celu wyizolowania, usunięcia i potwierdzenia defektu jako usuniętego;
- odniesienia (np. do przypadku testowego, do powiązanych defektów).

W zależności od kontekstu, dalsze informacje (np. śledzenie powiązań) mogą być również gromadzone w raporcie o defekcie (więcej informacji można znaleźć w normie ISO/IEC/IEEE 29119-3). Poniżej za pomocą punktów pogrupowano informacje zgodnie z zamierzonym celem:

- **Pomoc w usunięciu defektu:** podsystem lub moduł, w którym znajduje się defekt, konkretny element testowy i jego numer wydania, w którym zaobserwowano anomalię, lub środowisko testowe, w którym zaobserwowano defekt.
- **Ocena ogólnego statusu projektu:** informacje służące do monitorowania postępów (np. ryzyk, kosztów, możliwości i korzyści związanych z usunięciem lub nieusunięciem defektu, opis wszelkich dostępnych rozwiązań zastępczych lub wymagań, na które mają wpływ defekty).
- **Ocena stanu przyrostu produktu pod względem jego jakości:** typ defektu (zazwyczaj odpowiadający taksonomii defektów), produkt pracy, w którym defekt został wprowadzony, lub charakterystyka/podcharakterystyka jakościowa, na którą defekt ma wpływ.
- **Ocena zdolności procesu:** informacje umożliwiające monitorowanie skuteczności i wydajności procesów wytwarzania (np. faza SDLC wprowadzenia, wykrycia i usunięcia defektu lub podstawowej przyczyny defektu).

2.3.6 Określanie działań usprawniających proces przy użyciu informacji z raportów o defektach

Jak omówiono w sekcji 2.3.5 Informacje zawarte w raporcie o defekcie, raporty o defektach mogą być przydatne do monitorowania i raportowania statusu projektu. Podczas gdy implikacje metryk dla procesu testowego zostały przede wszystkim uwzględnione w sylabusie ISTQB® Poziom Ekspercki Zarządzanie Testami, w niniejszym sylabusie podkreśla się, że kierownicy testów powinni być świadomi tego, co oznaczają raporty o defektach w ocenie zdolności procesów wytwarzania i testowania oprogramowania.

Oprócz informacji dotyczących monitorowania postępów w testach, o których mowa w niniejszym sylabusie, w sekcji 2.1.2 Monitorowanie, nadzór i ukończenie testów oraz w sekcji 2.1.3 Raportowanie

testów informacje o defektach powinny wspierać inicjatywy usprawniania procesu, jak omówiono podczas retrospektyw. Na przykład:

- wykorzystanie informacji o fazach wprowadzania, wykrywania i usuwania defektów do oceny powstrzymania fazowego i/lub przeprowadzenia analizy kosztów jakości w celu zaproponowania sposobów poprawy skuteczności wykrywania defektów w każdej fazie i zminimalizowania kosztów związanych z defektami;
- wykorzystanie informacji o fazie wprowadzania do analizy faz, w których wprowadzana jest największa liczba defektów, aby umożliwić ukierunkowane usprawnianie na potrzeby zapobiegania defektom;
- wykorzystanie informacji o podstawowych przyczynach defektów w celu określenia przyczyn wprowadzenia defektów, aby umożliwić usprawnienia procesu, które zmniejszą całkowitą liczbę defektów;
- wykorzystanie informacji o lokalizacji defektów w celu przeprowadzenia analizy klastrów defektów, lepszego zrozumienia ryzyk technicznych (na potrzeby testowania opartego na ryzyku) i umożliwienia refaktoryzacji kłopotliwych modułów;
- wykorzystanie informacji o ponownie otwartych defektach do oceny jakości implementacji debugowania;
- wykorzystanie informacji o zduplikowanych i odrzuconych defektach do oceny jakości tworzenia raportów o defektach;
- umożliwienie usprawnień procesu, które zmniejszą całkowitą liczbę defektów, poprzez wprowadzenie proaktywnych środków, aby z góry uniknąć błędów.

Wykorzystanie metryk do oceny skuteczności i wydajności procesu testowego omówiono w sylabusie ISTQB® Poziom Ekspertski Zarządzanie Testami.

W niektórych przypadkach zespoły decydują się nie śledzić defektów wykrytych podczas niektórych lub wszystkich faz SDLC. Mimo że jest to często przeprowadzane w imię wydajności i w celu zmniejszenia kosztów ogólnych procesu, znacznie ogranicza to wgląd w możliwości procesu wytwarzania i testowania oprogramowania. Sprawia to, że zaproponowane powyżej usprawnienia są trudne do przeprowadzenia ze względu na brak wiarygodnych danych pomocniczych.

3. Zarządzanie zespołem – 225 minut

Słowa kluczowe

awaria, awaria wewnętrzna, awaria zewnętrzna, defekt, koszt jakości, ocena, zapobieganie defektom

Cele nauczania w rozdziale 3:

3.1 Zespół testowy

- TM-3.1.1 (K2) Kandydat podaje przykłady typowych umiejętności potrzebnych członkom zespołu testowego w czterech obszarach kompetencji.
- TM-3.1.2 (K4) Kandydat analizuje dany kontekst projektu w celu określenia wymaganych umiejętności członków zespołu testowego.
- TM-3.1.3 (K2) Kandydat objaśnia typowe techniki oceny umiejętności członków zespołu testowego.
- TM-3.1.4 (K2) Kandydat rozróżnia między typowymi podejściami do rozwijania umiejętności członków zespołu testowego.
- TM-3.1.5 (K2) Kandydat objaśnia umiejętności wymagane do zarządzania zespołem testowym.
- TM-3.1.6 (K2) Kandydat podaje przykłady czynników motywujących i czynników higienicznych dla członków zespołu testowego.

3.2 Relacje z interesariuszami

- TM-3.2.1 (K2) Kandydat podaje przykłady w każdej z czterech kategorii określających koszt jakości.
- TM-3.2.2 (K3) Kandydat stosuje obliczenia kosztów i korzyści w celu oszacowania wartości dodanej testów dla interesariuszy.

3.1 Zespół testowy

Wstęp

Każdy zespół, który wykonuje zadania testowe, składa się z osób o różnych kompetencjach. Podczas gdy w niektórych organizacjach zespoły są samoorganizujące się, w innych kierownicy testów rekrutują i rozwijają te zespoły. Odpowiednie połączenie umiejętności¹ jest krytycznym czynnikiem w przypadku wszystkich zespołów, aby zapewnić pomyślne wykonywanie zadań testowych.

Umiejętności wymagane od członka zespołu testowego mogą z czasem ulec zmianie. Ważne jest, aby wybrać odpowiednie osoby do zespołu testowego oraz zapewnić im odpowiednie szkolenia i możliwości rozwoju. Ponadto osoby spoza zespołu testowego mogą zapewnić dodatkowe konkretne umiejętności.

W tym rozdziale przyjrzymy się podstawowemu procesowi analizowania i rozwijania wymaganych umiejętności członków zespołu testowego, a także umiejętnościom, które są wymagane do kierowania lub szkolenia zespołu testowego. Obejmuje to również wiedzę na temat czynników, które motywują lub demotywują członków zespołu testowego, oraz innych czynników zapewniających udaną pracę zespołową.

Każda osoba ma już umiejętności i może je dalej rozwijać na różne sposoby, takie jak doświadczenie zawodowe, edukacja i szkolenia. Idealny zespół testowy ma wszystkie niezbędne umiejętności do wykonywania danych zadań testowych lub jest odpowiedzialny tylko za zadania, do których ma wymagane umiejętności. Aby odnieść sukces, zespół testowy potrzebuje różnych umiejętności na różnych poziomach. W zależności od kontekstu projektu, niektóre umiejętności będą ważniejsze lub bardziej potrzebne niż inne. W przypadku konkretnych zadań testowych, które wykraczają poza możliwości zespołu testowego, warto zatrudnić zewnętrznych ekspertów.

3.1.1 Typowe umiejętności w ramach czterech obszarów kompetencji

Umiejętności danej osoby można podzielić na cztery obszary kompetencji (Sonntag i Schmidt-Rathjens, 2005) (Erpenbeck i von Rosenstiel, 2017)²:

- **Kompetencje zawodowe:** składają się z umiejętności wykonywania specjalistycznych zadań. Przykłady obejmują umiejętności w zakresie technik testowania, wiedzę technologiczną i biznesową w dziedzinie aplikacji, a także umiejętności zarządzania projektami.
- **Kompetencje metodologiczne:** obejmuje ogólne umiejętności, których dana osoba może używać niezależnie w danej domenie i które umożliwiają niezależne wykonywanie złożonych lub nowych zadań. Przykłady obejmują umiejętności analityczne, koncepcyjne i oceny sytuacji.
- **Kompetencje społeczne:** obejmuje umiejętności związane z komunikacją, współpracą i zarządzaniem konfliktami w kontekście wewnątrz- i międzykulturowym. Umożliwiają one nawiązywanie relacji z innymi w celu odpowiedniego działania w danej sytuacji oraz osiągania indywidualnych i wspólnych celów. Przykłady obejmują umiejętności komunikacyjne, umiejętności rozwiązywania konfliktów, umiejętność pracy w zespole, zdolność adaptacji i asertywność.

¹ Termin „umiejętność” jest używany jako ogólny termin określający same umiejętności, posiadanie wiedzy o czymś i zdolności do zrobienia czegoś.

² Cztery obszary kompetencji wykorzystane w niniejszym dokumencie opierają się na modelu opisanym w tych źródłach, który jest szeroko stosowany. Istnieją inne modele opisane w literaturze, które grupują umiejętności w różny sposób. Nie są one częścią tego sylabusu.

- **Kompetencje osobiste:** obejmują zdolność i chęć do rozwijania siebie i własnego talentu, motywacji i chęci do działania, a także rozwijania określonych postaw i indywidualnej osobowości. Przykłady obejmują zarządzanie sobą, osobistą odpowiedzialność, zdolność do przyjmowania krytyki, niezawodność, odporność, zdolność do działania z pewnością siebie, dyscyplinę, otwartość na zmiany, chęć pomocy i uczenia się oraz zdolność do delegowania zadań.

Wszystkie obszary kompetencji są ważne dla sukcesu każdego zespołu testowego. Kompetencje metodologiczne, społeczne i osobiste nie są specyficzne dla testowania, dlatego ISTQB® koncentruje się na rozwijaniu kompetencji zawodowych. Obejmuje to umiejętności zarządzania zadaniami testowymi, analizowania podstawy testów, projektowania testów, wykrywania i analizowania ryzyk oraz opracowywania, konfigurowania i utrzymywania danych testowych, środowisk testowych i skryptów testowych.

3.1.2 Analiza wymaganych umiejętności członków zespołu testowego

Zatrudnianie personelu jest działaniem podejmowanym w ramach planowania testów. Obejmuje to zadanie określenia ról i umiejętności personelu wymaganych do wdrożenia testowania w strategii testów. Szczegółowa analiza kontekstu jest wymagana do określenia wymaganych umiejętności na potrzeby danego projektu.

Kompetencje zawodowe i metodologiczne

W przypadku testów nacisk kładziony jest na umiejętności testowe wymagane do wykonania zadań testowych. Oto kilka przykładów:

- Planowanie testów wymaga wiedzy koncepcyjnej do opracowania strategii testów.
- Monitorowanie i nadzór nad testami wymagają umiejętności zarządzania projektami w zakresie zarządzania wszystkimi zadaniami testowymi.
- Analiza testów wymaga umiejętności analitycznych w zakresie analizy podstawy testów i ryzyk produktowych.
- Projektowanie testów wymaga umiejętności w zakresie technik testowania do projektowania przypadków testowych i wiedzy koncepcyjnej do projektowania środowisk testowych.
- Implementacja testów wymaga umiejętności oceny przy wyborze testów oraz specjalistycznej wiedzy technicznej w zakresie programowania skryptów testowych i konfigurowania środowisk testowych.
- Wykonywanie testów wymaga specjalistycznej wiedzy technicznej do wykonywania testów automatycznych, przeprowadzania testów eksploracyjnych i oceny wyników testów.
- Ukończenie testów wymaga umiejętności komunikowania wyników projektu i osobistej odpowiedzialności za podejmowane decyzje.

Różne typy testów i poziomy testów wymagają różnych umiejętności (np. wiedzy biznesowej w dziedzinie aplikacji w celu oceny przydatności funkcjonalnej systemu lub specjalistycznej wiedzy technicznej w celu oceny możliwości utrzymania kodu).

Ponadto kontekst projektu dostarcza cennych informacji na temat wymaganych kompetencji zawodowych:

- Domena systemu wymaga wiedzy biznesowej w obszarze zastosowań, takich jak technologia informacyjna, przemysł motoryzacyjny lub branża hazardowa.

- Architektura oprogramowania i systemu oraz technologie wykorzystywane w projekcie wymagają, na przykład, specjalistycznej wiedzy technicznej w zakresie języków programowania, technologii interfejsów lub luk w zabezpieczeniach.
- SDLC wymaga na przykład wiedzy na temat poziomów testów, ról testerów i konkretnych technik testowania.

Kompetencje społeczne

W kontekście testowania kompetencje społeczne umożliwiają członkom zespołu testowego odpowiednie zachowanie w relacjach z innymi członkami zespołu i osiągnięcie celów testów. W szczególności obejmuje to umiejętności komunikacji, współpracy i rozwiązywania konfliktów (np. konstruktywne radzenie sobie z nieoptymalnymi warunkami testowymi lub zgłaszanie defektów programistom).

Wytwarzanie i testowanie oprogramowania jest zwykle wykonywane przez różnych członków (różnych) grup, którzy koordynują swoje zadania poprzez komunikację. Umiejętności komunikacyjne, zdolność do pracy w zespołach i umiejętność rozwiązywania konfliktów są wymagane dla powodzenia projektu. Wymagany poziom umiejętności społecznych może się jednak różnić w zależności od kontekstu projektu. Na przykład zwinne wytwarzanie oprogramowania może wymagać wyższych wymagań w zakresie umiejętności społecznych niż sekwencyjne modele wytwarzania zorientowane na dokumenty, a także testowanie poza siedzibą firmy.

Kompetencje osobiste

Efektywność i wydajność członków zespołu testowego zależy również od ich zdolności i chęci do rozwijania siebie, swoich umiejętności i postaw. Praca w samoorganizującym się zespole zwinnym może np. wymagać wyższego poziomu samodzielnego zarządzania i dyscypliny od wszystkich członków zespołu, podczas gdy kierownik testów w hierarchicznym zespole testowym musi być w stanie np. delegować pracę. Wysoki stopień niezawodności i odporności jest często wymagany, szczególnie w projektach o krytycznym znaczeniu czasowym. Ponadto chęć pomocy, uczenia się i otwartość na zmiany są ważne w procesie zmian we wszystkich modelach SDLC.

3.1.3 Ocena umiejętności członków zespołu testowego

W wielu przypadkach zespoły testowe są tworzone z istniejącego personelu. Aby zrozumieć możliwości członków zespołu i potrzebę rozwoju osobistego, kierownictwo testów musi ocenić istniejące umiejętności zespołu testowego i porównać je z wymaganymi umiejętnościami, które można udokumentować za pomocą macierzy umiejętności.

Istnieją modele, które pomagają zespołom i członkom zespołu pracować bardziej efektywnie (np. „Role zespołowe” Meredith Belbin, DISG® lub PCM®). Według Belbina (Belbin, 2010) zespoły pracują efektywnie, gdy składają się z różnych typów osobowości i ról. Modele te pomagają zespołom określić, jakie umiejętności mają, a jakich może im brakować.

Kompetencje zawodowe i metodologiczne członków zespołu testowego można ocenić poprzez zademonstrowanie typowych zadań testowych:

- przedstawienie strategii testów i omówienie informacji zwrotnych ze współpracownikami;
- przegląd podstawy testów i przekazanie wyników, co może również ujawnić umiejętności komunikacyjne;
- określanie technik testowania w celu osiągnięcia konkretnych celów testów w kontekście danego projektu;
- odpowiednie stosowanie różnych technik testowania;

- sporządzenie sumarycznego raportu z testów, który zawiera ocenę wyników testów.

Ponadto umiejętności można ocenić za pomocą zewnętrznych poświadczeń, certyfikatów, doświadczenia zawodowego i stopni naukowych.

Zwłaszcza w zwinnym wytwarzaniu oprogramowania zespoły określają umiejętności, których potrzebują, regularnie uczestnicząc w retrospektywach i otrzymując informacje zwrotne. Doświadczeni trenerzy lub mentorzy pomagają im rozwijać umiejętności oraz określać i uzupełniać luki w wiedzy.

3.1.4 Rozwijanie umiejętności członków zespołu testowego

Na początku projektu zespół testowy może nie posiadać wszystkich wymaganych umiejętności. Mimo że idealny zestaw osób może nie być dostępny, silny zespół może zrównoważyć mocne i słabe strony poszczególnych osób.

Kierownictwo testów lub zespół testowy może określić niezbędne potrzeby rozwojowe, porównując wymagane umiejętności z dostępnymi umiejętnościami w macierzy umiejętności. Na tej podstawie mogą określić podejścia do rozwoju kompetencji:

- szkolenia i edukacja uczą wstępnie zdefiniowanej wiedzy i praktyk, zwykle w (wirtualnej) klasie (np. wysyłając ludzi na kurs szkoleniowy, organizując sesje szkoleniowe we własnym zakresie, opracowując niestandardowe szkolenia lub korzystając z kursów e-learningowych na żywo);
- samokształcenie to sposób uczenia się o przedmiocie, który polega na samodzielnej nauce, a nie w (wirtualnej) klasie (np. czytanie książek, oglądanie nagranych filmów lub przeszukiwanie zasobów internetowych);
- wzajemne uczenie się, w ramach którego współpracownicy dzielą się wiedzą, pomysłami i doświadczeniami oraz uczą się od siebie nawzajem;
- mentoring lub coaching to podejścia, w których członek zespołu, który jest nowy na danym stanowisku, otrzymuje indywidualne wskazówki od trenera lub wiedzę, umiejętności i/lub doświadczenie od doświadczonego mentora; doświadczona osoba jest stałym źródłem porad i pomocy;
- szkolenie w miejscu pracy jest również dobrze znane i stanowi połączenie samokształcenia, wzajemnego uczenia się i mentoringu lub coachingu.

Nie wszystkie podejścia do rozwoju kompetencji są równie skuteczne i wydajne. Na przykład samokształcenie i szkolenia dobrze nadają się do rozwijania kompetencji zawodowych i metodologicznych. Z tego powodu podstawową wiedzę z zakresu testowania można rozwijać poprzez uczestnictwo w sesjach szkoleniowych ISTQB® lub poprzez samokształcenie w oparciu o sylabusy ISTQB®. W celu rozwijania kompetencji społecznych i osobistych zaleca się jednak stosowanie podejść, takich jak szkolenia i coaching, które często są bardziej obiecujące niż samokształcenie. Wymiana społeczna, informacje zwrotne i refleksja są jednymi z kluczowych czynników sukcesu w rozwijaniu kompetencji społecznych i osobistych.

3.1.5 Umiejętności kierownicze wymagane do zarządzania zespołem testowym

Każdy, kto chce z powodzeniem kierować zespołem testowym, musi dysponować umiejętnościami zarządzania. Obejmują one kompetencje zawodowe i metodologiczne w zakresie podstawowych czynności związanych z zarządzaniem (np. planowania, monitorowania postępów, nadzoru i raportowania). Do testowania wymagana jest określona wiedza i umiejętności w zakresie zarządzania testami (np. znajomość różnych podejść do testowania, opracowywanie strategii testów lub stosowanie technik testowania lub stosowanego SDLC).

Kierowanie zespołem testowym lub coaching zespołu testowego oznacza odpowiednie działanie w relacjach z innymi członkami zespołu testowego oraz umiejętność i chęć rozwoju w zmieniających się okolicznościach. Z tego powodu kompetencje społeczne i osobiste są niezbędnymi czynnikami sukcesu w kierowaniu zespołem testowym. Obejmuje to odporność, umiejętność delegowania zadań i zdolność do bycia akceptowanym przez zespół testowy. Co więcej, obejmuje to możliwość dochodzenia interesów testowych w projekcie, zdolność do potwierdzania interesów testowych w projekcie, promowania korzyści płynących z testowania oraz komunikowania się i rozwiązywania konfliktów ze wszystkimi interesariuszami.

Pozyskiwanie osób do zespołu testowego wymaga umiejętności analizowania warunków społecznych, zespołowych i warunków pracy. Umiejętności te pomagają zapewnić, aby zespół testowy pasował do warunków pracy lub, jeśli to możliwe, aby warunki pracy były dostosowane do zespołu testowego. Ponadto zespoły testowe podlegają dynamicznym procesom rozwoju i dlatego wymagają umiejętności sytuacyjnych (np. zgodnie z fazami modelu rozwoju małych grup Tuckmana) (Tuckman, 1965) (Bonebright, 2010):

- gotowość do pomocy członkom zespołu testowego w dołączeniu do zespołu testowego (formowanie – ang. *forming*);
- umiejętność rozwiązywania konfliktów w zespole testowym (docieranie – ang. *storming*);
- dyscyplina i przywództwo zorientowane na cel w celu zapewnienia uzgodnionych wartości i zasad (normowanie – ang. *norming*);
- umiejętność delegowania zadań w celu przekazania zespołowi testowemu osobistej odpowiedzialności (działanie – ang. *performing*);
- umiejętność postępowania z uznaniem i zaufaniem w stosunku do odchodzących członków zespołu testowego (rozwiązanie – ang. *adjourning*).

3.1.6 Czynniki motywujące lub demotywujące zespół testowy w określonych sytuacjach

Zadowoleni i zmotywowani członkowie zespołu testowego zwiększają produktywność i wydajność, a tym samym mają znaczący wpływ na powodzenie projektów. Gdy uda się to osiągnąć, szkolenia przekrojowe odbywają się w sposób nieformalny, członkowie zespołu testowego mogą zarządzać własnym obciążeniem pracą, a kierownictwo testów ma więcej czasu na zajmowanie się zagadnieniami zewnętrznymi. Teoria motywacji i higieny (Herzberg et al., 1993) rozróżnia motywatory i czynniki higieniczne:

Motywatory są postrzegane świadomie i mogą prowadzić do wzrostu i satysfakcji. Może to obejmować:

- uznanie i docenienie wykonanej pracy (np. zachęty i wszelkie inne indywidualne podejścia, które członkowie zespołu testowego uznają za cenne);
- większą odpowiedzialność i autonomię (np. definiowanie procesów testowych w zespole testowym);
- interesujące, znaczące i wymagające zadania, które członkowie zespołu testowego postrzegają jako osiągalne, a jednocześnie warte podjęcia działań (np. wybór i wprowadzenie nowego narzędzia do automatyzacji testów);
- awans i rozwój zawodowy (np. rozwój doświadczonego testera do roli kierownika testów lub właściciela procesu testowego).

Czynniki higieniczne zwykle uważa się za oczywiste. Ich zapewnienie nie prowadzi automatycznie do większej satysfakcji. Ich brak może jednak mieć demotywujący wpływ na członków zespołu testowego:

- odpowiednie wynagrodzenie (np. wynagrodzenie zgodne z warunkami rynku, płatne nadgodziny, dobre świadczenia socjalne);
- doceniająca polityka personalna i styl zarządzania (np. szczupłe zarządzanie, realistyczne cele, ochrona przed dostępem z zewnątrz i przeciążeniem);
- przyjemne warunki pracy (np. jednoznaczne specyfikacje, dojrzałe przedmioty testów, odpowiednio usuwane defekty, odpowiednie miejsce pracy, stabilne środowisko testowe);
- bezpieczeństwo jako potrzeba egzystencjalna (np. bezpieczne miejsce pracy i przestrzeganie umów);
- dobre relacje międzyludzkie (np. ze współpracownikami, przełożonymi).

W konsekwencji zarządzanie testami powinno stale eliminować czynniki demotywujące, a jednocześnie tworzyć i wzmacniać czynniki motywujące.

Więcej informacji można znaleźć w (Belbin, 2010) (Marston, 1999) (Kahler, 2008).

3.2 Relacje z interesariuszami

Wstęp

W zarządzaniu testami ważne jest, aby zoptymalizować testowanie w celu zapewnienia dobrej wartości biznesowej. Nadmierne testowanie może skutkować nieuzasadnionymi opóźnieniami i kosztami, które przewyższają korzyści, podczas gdy niewystarczające testowanie może prowadzić do dostarczenia użytkownikom produktu o niskiej jakości. Optymalne podejście leży pomiędzy tymi dwiema skrajnościami. Obowiązkiem kierownika testów jest pomoc interesariuszom w zrozumieniu tej równowagi i wartości dodanej testowania w osiągnięciu tej równowagi, mając jednocześnie na uwadze np. typowe ograniczenia czasowe projektu.

3.2.1 Koszt jakości

Korzyści z testowania są równoważone przez koszty jakości. Sposób ilościowego określenia całkowitego kosztu nakładu pracy związanego z testowaniem i defektów związanych z jakością nazywa się kosztem jakości. Koszt jakości obejmuje klasyfikację kosztów projektu i kosztów operacyjnych w czterech kategoriach związanych z kosztami defektów produktu:

- **Koszty zapobiegania defektom:** koszt wszystkich działań, które są zaplanowane i proaktywne, aby zapobiec niskiej jakości (np. kwalifikacja programistów do ich zadań, takich jak szkolenie w zakresie tworzenia łatwego w utrzymaniu lub bezpiecznego kodu, przegląd podstawy testów tak wcześnie, jak to możliwe, oraz odpowiednia komunikacja w zespole).
- **Koszty wykrycia³ defektów:** koszt wszystkich działań mających na celu wykrywanie defektów (np. przeprowadzanie testów statycznych i dynamicznych oraz przegląd produktów pracy).
- **Koszty awarii wewnętrznej:** koszt wszystkich działań reaktywnych (np. usuwanie defektów wykrytych podczas testowania, zapewnianie rozwiązań zastępczych).
- **Koszty awarii zewnętrznej:** koszt wszystkich działań nietworzących wartości dodanej i reaktywnych (np. utrata przychodów, aktywów, zdrowia ludzkiego, życia ludzkiego lub środowiska, koszty prawne związane z usuwaniem defektów, testowaniem, wdrażaniem i wsparciem z powodu dostarczenia wadliwego produktu klientowi („po wydaniu”), usuwanie defektów polowych (zgłoszonych przez klientów)).

Całkowite koszty wykrycia defektów i koszty awarii wewnętrznej są zwykle znacznie niższe niż koszty awarii zewnętrznej. To sprawia, że testowanie jest niezwykle cenne. Określając koszty w tych czterech kategoriach, kierownicy testów mogą stworzyć przekonujące uzasadnienie biznesowe testowania.

Istnieje więcej podejść, które można rozważyć w celu określenia kosztu jakości. Sylabus ISTQB® obejmuje dwa z nich. Ten sylabus jest oparty na podejściu Feigenbauma, a sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0 przedstawia podejście Boehma (patrz sylabus ISTQB® CTFL w. 4.0, podrozdział 1.3). Te dwa podejścia zostały wybrane w celu osiągnięcia szerszego zrozumienia kosztu jakości. Podejście Feigenbauma (Feigenbaum, listopad/grudzień 1956) traktuje jakość jako proces zorientowany na klienta i obejmujący całą firmę, podczas gdy podejście Boehma (Boehm, 1979) koncentruje się na kompromisie między kosztem zapobiegania a kosztem awarii w wytwarzaniu oprogramowania (Hadjicostas, 2004).

³ Koszty wykrycia są związane z oceną jakości systemu (przyp. tłumaczy).
w. 3.0

3.2.2 Stosunek kosztów do korzyści testowania

Podczas gdy większość organizacji uważa testowanie za wartościowe w pewnym sensie, niewielu kierowników, w tym kierowników testów, potrafi określić ilościowo, opisać lub wyrazić tę wartość. Ponadto wielu kierowników testów, liderów testów i testerów koncentruje się na operacyjnych szczegółach testowania (tj. aspektach specyficznych dla zadania testowego lub poziomu testu), ignorując większe problemy taktyczne i strategiczne (wyższego poziomu) związane z testowaniem, na których zależy innym interesariuszom, zwłaszcza kierownikom.

Testowanie przynosi korzyści organizacji, projektowi i/lub operacjom zarówno w sensie ilościowym, jak i jakościowym:

- **Korzyści jakościowe** obejmują poprawę reputacji w zakresie jakości, płynniejsze i bardziej przewidywalne wydania, zwiększone zaufanie, ochronę przed odpowiedzialnością prawną oraz zmniejszenie ryzyka utraty całej działalności, a nawet życia.
- **Korzyści ilościowe** obejmują defekty wykryte lub usunięte przed wydaniem, defekty, które były znane przed wydaniem (tj. nie zostały usunięte, ale udokumentowane, być może z rozwiązaniami zastępczymi), korzyści kosztowe (Bohm 1981, Böhler 2008), zmniejszenie poziomu ryzyka poprzez przeprowadzanie testów i dostarczanie informacji o statusie projektu, procesu i produktu.

Dodatkową zaletą testowania jest to, że wszyscy interesariusze otrzymują odpowiednie informacje, aby podejmować świadome decyzje dotyczące tego, czy jakość produktu jest wystarczająca do uruchomienia, z defektami lub bez. Czasami wydanie produktu ze znanymi defektami jest znacznie lepsze niż czekanie, aż defekty zostaną usunięte. W tych przypadkach, w których defekt może być tolerowany, jest to silnie uzależnione od prawdopodobieństwa wystąpienia i krytyczności defektu.

Koszt jakości na defekt w procesie testowym oblicza się w następujący sposób:

Średnia oszczędność na defekt = średnie koszty awarii zewnętrznej - (średni koszt wykrycia defektu + średnie koszty awarii wewnętrznej)

Całkowity koszt jakości = (koszty zapobiegania defektom + (koszty wykrycia defektów * liczba defektów znalezionych przed wydaniem)) + ((koszt awarii wewnętrznej * liczba defektów znalezionych przed wydaniem) + (koszty awarii zewnętrznej * liczba defektów znalezionych po wydaniu))

Jako przykład przyjmijmy, że obliczono następujący koszt jakości na defekt dla produktu:

- Średni koszt zapobiegania defektowi: 180 USD
- Średni koszt wykrycia defektu: 500 USD
- Średni koszt awarii wewnętrznej na defekt: 200 USD
- Średni koszt awarii zewnętrznej na defekt: 4 000 USD

Średnie koszty zapobiegania, wykrycia i awarii wewnętrznej oblicza się na podstawie liczby defektów wykrytych przed wydaniem, podczas gdy średni koszt awarii zewnętrznej oblicza się na podstawie liczby defektów wykrytych po wydaniu. Mając te wartości, możemy obliczyć średnie oszczędności na defekt w następujący sposób:

Średnia oszczędność na defekt = 4 000 USD - (500 USD + 200 USD) = 3 300 USD.

Krzywa Boehma jest graficzną reprezentacją kosztów usuwania defektów w czasie w SDLC. Wynika z niej, że testowanie powinno odbywać się na wcześniejszym etapie SDLC, aby zmniejszyć koszty usuwania defektów. Krzywa Boehma pokazuje, że koszty awarii wewnętrznej lub koszty naprawy

defektów rosną, im później defekt zostanie wykryty w SDLC. Korzystając z tych informacji, kierownik testów powinien dążyć do znalezienia optymalnej relacji między kosztami zapobiegania defektom a kosztami wewnętrznej i zewnętrznej awarii.

Nakład pracy związany z testowaniem musi opierać się na konkretnych ryzykach projektowych i produktowych oraz ryzyku, jakie firma jest skłonna podjąć. Zbyt wiele testów może powodować wyższe koszty niż korzyści wynikające z obniżenia poziomu ryzyka. Jeśli testowanie jest niewystarczające, pominięte defekty mogą stwarzać wysokie ryzyko generowania wyższych kosztów niż koszt pominiętych testów. Testowanie oparte na ryzyku (patrz podrozdział 1.3) wspiera relację kosztów i korzyści testowania poprzez inwestowanie poziomów nakładu pracy związanego z testowaniem proporcjonalnie do poziomów ryzyka oraz poprzez ustalanie priorytetów testów w oparciu o ich poziomy ryzyka.

Kierownicy testów powinni rozumieć, które z tych korzyści i kosztów odnoszą się do ich organizacji, projektu i/lub działania oraz powinni być w stanie zakomunikować wartość dodaną testów w odniesieniu do tych korzyści i kosztu jakości przypadającego na defekt.

4. Bibliografia

Normy

- **IEC 61508** (2010) Functional Safety of electrical/electronic/programmable electronic safety-related systems – Part 1 to 7
- **ISO/IEC/IEEE 29119-2** (2021) ISO/IEC/IEEE 29119-2 Software and systems Engineering - Software testing – Part 2 Test Processes
- **ISO/IEC/IEEE 29119-3** (2021) ISO/IEC/IEEE 29119-3 Software and systems Engineering - Software testing – Part 3 Test documentation

Dokumenty ISTQB®

- Sylabus ISTQB® Certyfikowany Tester Agile Test Leadership at Scale w. 2.0 (2023)
- Sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0 (2023)
- Sylabus ISTQB® Certyfikowany Tester Poziom Ekspercki Zarządzanie Testami ISTQB® v1.0 (2011)
- Sylabus ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie usprawnień procesu testowego w. 1.0.1 (2011)

Książki

- Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., Rombach, D. (2014). *Aligning Organizations Through Measurement - The GQM+ Strategies Approach*. Springer International.
- Bath, G., van Veenendaal, E. (2014). *Improving the Test Process - rozdział 6: Process for Improvement*. Rocky Nook.
- Belbin, R. M. (2010). *Management Teams: Why They Succeed or Fail*. Londyn: Routledge.
- Black, R. (2009). *Managing the Testing Process, 3rd Edition*. John Wiley & Sons.
- Boehm, B. (1979). *Software Engineering Economics*. Prentice-Hall.
- Bonebright, D. A. (2010). *40 years of storming: a historical review of Tuckman's model of small group development* (1 Ausg., Bd. 13). Human Resource Development International, 1, 2010, Vol. 13.
- Craig, R., Jaskiel, S. P. (2002). *Systematic Software Testing*. Artech House.
- Derby, E., Larsen, D. (2006). *Agile Retrospectives - Making Good Teams Great*. The Pragmatic Bookshelf.
- Erpenbeck, J., von Rosenstiel, L. (2017). *Handbuch Kompetenzmessung*. Stuttgart: Schäffer- Poeschel.
- Fowler, M. (2010). *Hybrid development processes*. IEEE Software, 27(2), 57-63.
- Herzberg, F., Mausner, B., Bloch Snyderman, B. (1993). *Motivation to Work*. Londyn: Routledge.
- Kahler, T. (2008). *The Process Therapy Model: The Six Personality Types with Adaptations*. Taibi Kahler Associates, Inc.
- Marston, W. M. (1999). *Emotions Of Normal People*. Londyn: Routledge.

- Sonntag, K., Schmidt-Rathjens, C. (2005). *Anforderungsanalyse und Kompetenzmodelle*. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Tuckman, B. W. (1965). *Developmental sequence in small groups* (Bd. 63(6)). Psychological Bulletin.
- van Ewijk, A. (2013). *TPI NEXT - Business Driven Test Process Improvement*,. Sogeti Nederland B.V.
- van Solingen, R., Berghout, E. (1999). *The Goal Question Metric Method - A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill.
- van Veenendaal, E. (2012). *The PRISMA Approach: Practical Risk-Based Testing*. UTN Publishers.
- van Veenendaal, E. (2020). *TMMi in the Agile world, version 1.4*. TMMi Foundation.
- van Veenendaal, E., Cannegieter, J. J. (2011). *The Little TMMi - Objective-Driven Test Process Improvement*. UTN Publishers.

Artykuły

- Feigenbaum, Armand V. (listopad/grudzień 1956) Harvard Business Review, tom 34, wyd. 6, str. 93–101
- Hadjicostas, Evsevios (2004) Total Quality Management and Cost of Quality, Springer

Strony internetowe

- www.tmmi.org Test Maturity Model integration (TMMi®); ostatnia wizyta 31 stycznia 2024 r.
- www.tmap.net Test Process Improvement (TPI); ostatnia wizyta 31 stycznia 2024 r.
- www.wikipedia.org/wiki/PDCA Plan-Do-Check-Act; ostatnia wizyta 31 stycznia 2024 r.

Poprzednie odniesienia wskazują na informacje dostępne w Internecie i innych miejscach. Nawet jeśli te odniesienia zostały sprawdzone w momencie publikacji niniejszego sylabusu, ISTQB® nie ponosi odpowiedzialności, jeśli odniesienia te nie są już dostępne.

5. Załącznik A – Cele nauczania i poziomy poznawcze

Konkretne cele nauczania, które mają zastosowanie do tego sylabusu, przedstawiono na początku każdego rozdziału. Znajomość każdego z zagadnień poruszonych w sylabusie będzie sprawdzana na egzaminie zgodnie z przypisanym celem nauczania.

Opis celu nauczania zawiera czasownik wyrażający czynność, który odpowiada właściwemu poziomowi wiedzy wymienionemu poniżej.

Poziom 1: Zapamiętać (K1)

Kandydat pamięta, rozpoznaje lub umie sobie przypomnieć dany termin lub dane pojęcie.

Czynności (czasowniki akcji): pamiętać, rozpoznawać.

Przykłady
Kandydat pamięta pojęcia związane z piramidą testów.
Kandydat rozpoznaje typowe cele testów.

Poziom 2: Zrozumieć (K2)

Kandydat potrafi uzasadnić lub wyjaśnić stwierdzenia dotyczące danego zagadnienia, a także podsumować, porównać i sklasyfikować pojęcia z zakresu testowania oraz podać odpowiednie przykłady.

Czynności (czasowniki akcji): klasyfikować, porównywać, rozróżniać, wyjaśniać, podawać przykłady, interpretować, podsumowywać

Przykłady	Uwagi
Kandydat wyjaśnia, w jaki sposób różnego typu narzędzia testowe wspomagają testowanie.	
Kandydat rozróżnia poszczególne poziomy testów.	Może służyć do wyszukiwania podobieństw, różnic lub obu.
Kandydat odróżnia testowanie od debugowania.	Szuka różnic między koncepcjami.
Kandydat rozróżnia ryzyka projektowe i produktowe.	Umożliwia oddzielne klasyfikowanie dwóch (lub więcej) pojęć.
Kandydat wyjaśnia wpływ kontekstu na proces testowy.	
Kandydat podaje przykłady wskazujące, dlaczego testowanie jest niezbędne.	
Kandydat wyciąga wnioski o przyczynie podstawowej defektu na podstawie danego profilu awarii.	
Kandydat podsumowuje czynności wykonywane w ramach procesu przeglądu.	

Poziom 3: Zastosować (K3)

Kandydat potrafi wykonać odpowiednią procedurę, gdy zostanie mu postawione znane mu zadanie, bądź wybrać właściwą procedurę i zastosować ją w danym kontekście.

Czynności (czasowniki akcji): zastosować, zaimplementować, przygotować, użyć.

Przykłady	Uwagi
Kandydat używa techniki analizy wartości brzegowych, aby zaprojektować przypadki testowe.	Powinno odnosić się do procedury/techniki/procesu itp.
Kandydat potrafi wdrożyć metody gromadzenia metryk w celu spełnienia wymagań technicznych i zarządczych.	
Kandydat potrafi przygotować testy instalowalności aplikacji mobilnych.	
Kandydat potrafi wykorzystać śledzenie powiązań do monitorowania postępów testów pod kątem kompletności i spójności z celami testów, strategią testów i planem testów.	Może być stosowane w LO, w którym kandydat musi być w stanie zastosować technikę lub procedurę. Podobne do „zastosować”.

Poziom 4: Przeanalizować (K4)

Kandydat potrafi rozdzielić informacje związane z procedurą lub techniką na ich części składowe w celu lepszego zrozumienia i potrafi odróżnić fakty od wniosków. Typowym zastosowaniem jest analiza dokumentu, oprogramowania lub sytuacji projektowej i zaproponowanie odpowiednich działań w celu rozwiązania problemu lub wykonania zadania.

Czynności (czasowniki akcji): analizować, dekonstruować, nakreślać, ustalać priorytety, wybierać

Przykłady	Uwagi
Kandydat potrafi przeanalizować daną sytuację projektową, aby określić, które techniki testowania czarnoskrzynkowego lub oparte na doświadczeniu należy zastosować, aby osiągnąć określone cele.	Może być przedmiotem egzaminu tylko w połączeniu z mierzalnym celem analizy. Powinno mieć formę „Przeanalizuj xxxx do xxxx” (lub podobną).
Kandydat potrafi nadawać priorytety przypadkom testowym w danym zestawie testowym do wykonania w oparciu o powiązane ryzyka produktowe.	
Kandydat potrafi wybrać odpowiednie poziomy i typy testów w celu weryfikacji danego zestawu wymagań.	Potrzebne, gdy wybór wymaga analizy.

Materiały referencyjne

(w zakresie poziomów poznawczych celów nauczania)

Anderson, L. W. i Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

Konkretne cele nauczania, które mają zastosowanie do tego sylabusu, przedstawiono na początku każdego rozdziału.

6. Załącznik B – Macierz powiązań między celami biznesowymi a celami nauczania

Ten rozdział zawiera powiązania między celami biznesowymi a celami nauczania sylabusu Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami.

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami		BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM_01	Zarządzanie testowaniem w różnych projektach wytwarzania oprogramowania z zastosowaniem procesów zarządzania testami ustanowionych dla zespołu projektowego lub organizacji testującej.	12										
TM_02	Określanie interesariuszy procesu testowego i modeli cyklu wytwarzania oprogramowania, które są istotne w danym kontekście.		4									
TM_03	Organizowanie sesji na potrzeby określania i oceny ryzyka w ramach dowolnego cyklu wytwarzania oprogramowania i wykorzystywanie wyników do kierowania testami w celu osiągnięcia celów testów.			6								
TM_04	Określanie strategii testów projektu zgodnej z organizacyjną strategią testów i z kontekstem projektu.				11							
TM_05	Ciągłe monitorowanie i nadzór nad testami w celu osiągnięcia celów projektu.					4						
TM_06	Ocena i raportowanie postępów testów interesariuszom projektu.						3					
TM_07	Określanie niezbędnych umiejętności i rozwijanie ich w zespole.							6				
TM_08	Przygotowywanie i przedstawianie uzasadnień biznesowych do przeprowadzania testów w różnych								5			

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
	kontekstach, z określeniem kosztów i oczekiwanych korzyści.												
TM_09	Prowadzenie działań usprawniających procesy testowe w projektach lub strumieniach produktów związanych z wytwarzaniem oprogramowania oraz wnoszenie wkładu w organizacyjne inicjatywy na rzecz usprawnienia procesów testowych.										5		
TM_10	Planowanie czynności testowych i szacowanie nakładu pracy związanego z testowaniem.											9	
TM_11	Tworzenie raportów o defektach i przepływu pracy dotyczącego defektów odpowiedniego w kontekście danego cyklu wytwarzania oprogramowania.												6
Unikalne LO	Cel nauczania	Poziom wiedzy											
1	Zarządzanie czynnościami testowymi												
1.1	Proces testowy												
TM-1.1.1	Kandydat podsumowuje planowanie testów.	K2	X			X							
TM-1.1.2	Kandydat podsumowuje monitorowanie testów i nadzór nad testami.	K2	X				X						
TM-1.1.3	Kandydat podsumowuje ukończenie testów.	K2	X					X					
1.2	Kontekst testowania												
TM-1.2.1	Kandydat porównuje powody, dla których różni interesariusze są zainteresowani testowaniem.	K2		X		X							

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.2.2	Kandydat wyjaśnia, dlaczego wiedza interesariuszy jest ważna w zarządzaniu testami.	K2		X		X							
TM-1.2.3	Kandydat wyjaśnia testowanie w hybrydowym modelu wytwarzania oprogramowania.	K2		X		X							
TM-1.2.4	Kandydat podsumowuje czynności związane z zarządzaniem testami w różnych cyklach wytwarzania oprogramowania.	K2	X	X		X							
TM-1.2.5	Kandydat porównuje czynności związane z zarządzaniem testami na różnych poziomach testów.	K2	X			X							
TM-1.2.6	Kandydat porównuje czynności związane z zarządzaniem testami w różnych typach testów.	K2	X			X							
TM-1.2.7	Kandydat analizuje dany projekt i określa czynności związane z zarządzaniem testami, które kładą nacisk na planowanie testów, monitorowanie testów i nadzór nad testami.	K4	X			X							
1.3	Testowanie oparte na ryzyku												
TM-1.3.1	Kandydat wyjaśnia różne środki podejmowane w ramach testów opartych na ryzyku w celu reagowania na ryzyko.	K2			X								

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.3.2	Kandydat podaje przykłady różnych technik, które kierownik testów może wykorzystać do określenia ryzyk jakościowych produktu.	K2			X								
TM-1.3.3	Kandydat podsumowuje czynniki określające poziomy ryzyka związane z jakością produktu.	K2			X								
TM-1.3.4	Kandydat wybiera odpowiednie czynności testowe w celu łagodzenia ryzyk zgodnie z poziomem ryzyka w danym kontekście.	K4			X								
TM-1.3.5	Kandydat rozróżnia między przykładami ciężkich i lekkich technik testowania opartych na ryzyku.	K2			X								
TM-1.3.6	Kandydat podaje przykłady wskaźników sukcesu i trudności związanych z testowaniem opartym na ryzyku.	K2			X								
1.4	Projektowa strategia testowa												
TM-1.4.1	Kandydat objaśnia typowe wybory dotyczące podejścia do testów.	K2				X							
TM-1.4.2	Kandydat analizuje organizacyjną strategię testów i kontekst projektu w celu wyboru odpowiedniego podejścia do testów.	K4				X							
TM-1.4.3	Kandydat wykorzystuje metodę określania celów SMART do określenia mierzalnych celów testów i kryteriów wyjścia.	K3				X							

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
1.5	Usprawnianie procesu testowego												
TM-1.5.1	Kandydat objaśnia, jak wykorzystać model IDEAL do usprawnienia procesu testowego w danym projekcie.	K2									X		
TM-1.5.2	Kandydat podsumowuje podejścia do usprawniania opartego na modelu w celu przetestowania usprawnień procesów i zrozumienia, jak zastosować je w kontekście projektu.	K2									X		
TM-1.5.3	Kandydat podsumowuje analityczne podejście do usprawniania procesów i rozumie, jak zastosować je w kontekście projektu.	K2									X		
TM-1.5.4	Kandydat wdraża retrospektywy projektu lub iteracji w celu oceny procesów testowania i odkrycia obszarów testowania wymagających poprawy.	K3									X		
1.6	Narzędzia testowe												
TM-1.6.1	Kandydat podsumowuje najlepsze praktyki w zakresie wprowadzania narzędzi.	K2										X	
TM-1.6.2	Kandydat objaśnia wpływ różnych aspektów technicznych i biznesowych przy podejmowaniu decyzji o wyborze odpowiedniego rodzaju narzędzia.	K2										X	

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.6.3	Kandydat analizuje daną sytuację w celu opracowania planu wyboru narzędzi, obejmującego ryzyka, koszty i korzyści.	K4										X	
TM-1.6.4	Kandydat rozróżnia etapy cyklu życia narzędzi.	K2										X	
TM-1.6.5	Kandydat podaje przykłady gromadzenia i oceny metryk za pomocą narzędzi.	K2									X	X	
2	Zarządzanie produktem												
2.1	Metryki testów												
TM-2.1.1	Kandydat podaje przykłady metryk umożliwiających osiągnięcie celów testów.	K2					X						
TM-2.1.2	Kandydat objaśnia, w jaki sposób kontrolować postęp testów przy użyciu metryk testów.	K2					X						
TM-2.1.3	Kandydat analizuje wyniki testów w celu tworzenia raportów z testów, które umożliwiają interesariuszom podejmowanie decyzji.	K4					X	X					
2.2	Szacowanie testów												
TM-2.2.1	Kandydat wyjaśnia czynniki, które należy wziąć pod uwagę przy szacowaniu testów.	K2	X							X		X	
TM-2.2.2	Kandydat podaje przykłady czynników, które mogą mieć wpływ na szacowanie testów.	K2	X							X		X	

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-2.2.3	Kandydat wybiera odpowiednią technikę lub podejście do szacowania testów w danym kontekście.	K4	X							X		X	
2.3	Zarządzanie defektami												
TM-2.3.1	Kandydat wdraża proces zarządzania defektami, w tym przepływ pracy dotyczący defektów, który można wykorzystać do monitorowania i kontroli defektów.	K3											X
TM-2.3.2	Kandydat objaśnia proces zarządzania defektami i identyfikuje jego uczestników.	K2											X
TM-2.3.3	Kandydat wyjaśnia specyfikę zarządzania defektami w zwinnym wytwarzaniu oprogramowania.	K2	X										X
TM-2.3.4	Kandydat wyjaśnia wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania.	K2	X										X
TM-2.3.5	Kandydat wykorzystuje dane i informacje klasyfikacyjne, które należy gromadzić podczas zarządzania defektami.	K3											X
TM-2.3.6	Kandydat wyjaśnia, w jaki sposób statystyki zgłoszeń defektów można wykorzystać do opracowania usprawnienia procesu.	K2										X	X
3	Zarządzanie zespołem												
3.1	Zespół testowy												

Cele biznesowe w: Poziom Zaawansowany – Zarządzanie Testami			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-3.1.1	Kandydat podaje przykłady typowych umiejętności potrzebnych członkom zespołu testowego w czterech obszarach kompetencji.	K2							X				
TM-3.1.2	Kandydat analizuje dany kontekst projektu w celu określenia wymaganych umiejętności członków zespołu testowego.	K4							X				
TM-3.1.3	Kandydat objaśnia typowe techniki oceny umiejętności członków zespołu testowego.	K2							X				
TM-3.1.4	Kandydat rozróżnia między typowymi podejściami do rozwijania umiejętności członków zespołu testowego.	K2							X				
TM-3.1.5	Kandydat objaśnia umiejętności wymagane do zarządzania zespołem testowym.	K2							X				
TM-3.1.6	Kandydat podaje przykłady czynników motywujących i czynników higienicznych dla członków zespołu testowego.	K2							X				
3.2	Relacje z interesariuszami												
TM-3.2.1	Kandydat podaje przykłady w każdej z czterech kategorii określających koszt jakości.	K2								X			
TM-3.2.2	Kandydat stosuje obliczenia kosztów i korzyści w celu oszacowania wartości dodanej testów dla interesariuszy.	K3						X		X			

7. Załącznik C – Opis wydania

Sylabus ISTQB® Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami w. 3.0 jest główną aktualizacją opartą na sylabusie Certyfikowany Tester Poziom Zaawansowany Kierownik Testów 2012. W związku z tym nie sporządzono szczegółowego opisu wydania w podziale na rozdziały i podrozdziały, a jedynie przedstawiono podsumowanie najważniejszych zmian.

W bieżącej wersji wszystkie cele nauczania zostały przeredagowane w taki sposób, aby każdy z nich stanowił niepodzielną całość oraz aby można było jednoznacznie powiązać cele nauczania z treścią podrozdziałów sylabusa. Zadbano też o to, aby sylabus nie zawierał treści niepowiązanych z żadnymi celami nauczania. Autorzy skupili się na możliwości praktycznego wykorzystania przedstawionych treści oraz na równowadze między wiedzą a umiejętnościami, dzięki czemu nowa wersja powinna być bardziej przystępna i zrozumiała oraz łatwiejsza do przetłumaczenia, a zawarty w niej materiał — łatwiejszy do opanowania.

W bieżącym wydaniu głównym wprowadzono następujące zmiany:

- Zmniejszono objętość całego sylabusa. Sylabus nie jest podręcznikiem, lecz dokumentem mającym na celu przedstawienie w skrócie podstawowych elementów kursu zaawansowanego do tematyki testowania oprogramowania, w tym wskazanie, jakie tematy należy poruszyć i na jakim poziomie. W szczególności:
 - o z tekstu usunięto w większości przypadków przykłady, ponieważ przygotowanie przykładów i ćwiczeń do wykorzystania w trakcie kursu jest zadaniem dostawcy szkoleń;
 - o przestrzegano „Listy kontrolnej pisania sylabusa”, która sugeruje maksymalną długość treści celów nauczania na poziomie wiedzy (K2 = maksymalnie 1 500 znaków bez spacji, K3 = maksymalnie 2 500 znaków bez spacji, K4 = maksymalnie 3 000 znaków bez spacji, +/- 20%).
- Zmniejszono liczbę celów nauczania (LO) w porównaniu z sylabusem CTAL TM 2012:
 - o 36 LO K2 w porównaniu z 39 LO w CTAL TM 2012;
 - o 5 LO K3 w porównaniu z 12 LO w CTAL TM 2012;
 - o 7 LO K4 w porównaniu z 10 LO w CTAL TM 2012.
- Poprawiono pełną strukturę sylabusa.
- Dostosowano do sylabusa ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0.
- Wprowadzono duże zmiany w dotychczasowym (2012) rozdziale 1 (Proces testowy):
 - o ograniczono do zarządzania czynnościami testowymi (planowania testów, monitorowania testów, nadzoru nad testami i ich ukończenia);
 - o umieszczono jako sekcję w nowym rozdziale „Zarządzanie czynnościami testowymi”;
- dodano nowy rozdział **Zarządzanie czynnościami testowymi**;
 - o podrozdział 1.1 – Proces testowy: patrz wyżej;
 - o podrozdział 1.2 – Kontekst testowania: rozszerzono o niesekwencyjne modele wytwarzania oprogramowania;

- o podrozdział 1.3 – Testowanie oparte na ryzyku: całkowicie przeredagowano, aby był bardziej przydatny na poziomie projektu;
 - o podrozdział 1.4 – Projektowa strategia testowa: plan testów jest już określony w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0, dlatego skoncentrowano się na wyborze odpowiedniego podejścia testowego i sposobie definiowania mierzalnych celów testów;
 - o podrozdział 1.5 – Usprawnianie procesu testowego: włączono do Zarządzania czynnościami testowymi, pokazano, jak zastosować je w kontekście projektu i wdrażać za pomocą retrospektyw w ramach iteracji lub projektu;
 - o podrozdział 1.6 – Narzędzia testowe: wprowadzanie narzędzi zostało przeniesione z sylabusa ISTQB® Certyfikowany Tester Poziom Podstawowy w.3.1 (nie występują w sylabusie ISTQB® CTFL w. 4.0);
- dodano nowy rozdział **Zarządzanie produktem**;
 - o podrozdział 2.1 – Metryki testów: wcześniejsze punkty definiujące metryki i wykorzystanie metryk – Metryki testów;
 - o podrozdział 2.2 – Szacowanie testów: sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy w. 4.0 obejmuje już obliczanie szacunków testowych; rozwinięto, aby wybrać na poziomie K4 odpowiednie techniki szacowania testów i stosować szacowanie testów w modelach SDLC;
 - o podrozdział 2.3 – Zarządzanie defektami: dostosowano do najnowszych wydań norm i rozszerzono o wykorzystanie w podejściu zwinnym i w hybrydowym modelu wytwarzaniu oprogramowania;
- dodano nowy rozdział **Zarządzanie zespołem**:
 - o podrozdział 3.1 – Zespół testowy: główne tematy są takie same jak w sylabusie TM 2012; określanie indywidualnych umiejętności i tworzenie zespołów testowych;
 - o podrozdział 3.2 – Relacje z interesariuszami: jest to dotychczasowy podrozdział sylabusa TM 2012 zatytułowany „Wartość biznesowa testowania”;
- Wprowadzono duże zmiany i usunięto podrozdziały/rozdziały w sylabusie CTAL TM 2012:
 - o usunięto podrozdział dotyczący testów rozproszonych, zewnętrznych i wewnętrznych;
 - o usunięto podrozdział dotyczący zarządzania stosowaniem norm branżowych;
 - o usunięto rozdział dotyczący przeglądów;
 - o usunięto sekcje dotyczące usprawniania procesu testowego CTP i STEP;
 - o usunięto sekcje dotyczące analizy testów, projektowania testów, implementacji testów i wykonywania testów.

8. Załącznik D – Słowa kluczowe specyficzne dla domeny

Termin	Definicja
Cel Pytanie Metryka (GQM)	Podejście do pomiaru oprogramowania z wykorzystaniem trójpoziomowego modelu składającego się z poziomu pojęciowego (cel), poziomu operacyjnego (pytanie) oraz poziomu ilościowego (metryka).
IDEAL	Model doskonalenia organizacyjnego, który służy jako mapa drogowa do inicjowania, planowania i wdrażania działań usprawniających.
wskaźnik	Miara, która zapewnia oszacowanie lub ocenę określonych atrybutów pochodzących z modelu w odniesieniu do określonych potrzeb informacyjnych.
miara	Liczba lub kategoria przypisana do atrybutu jednostki poprzez dokonanie pomiaru.
metryka	Skala pomiarowa i metoda pomiaru.
poker planistyczny	Technika szacowania oparta na kompromisie, zwykle używana do szacowania nakładu pracy lub relatywnej wielkości historyjek użytkownika, wykorzystywana na ogół w zwinnym wytwarzaniu oprogramowania. Jest odmianą szerokopasmowej techniki delfickiej, używającą talii kart z wartościami reprezentującymi jednostki, w których zespół dokonuje szacowania.
szacowanie trójpunktowe	W przypadku tej techniki opartej na ekspertach eksperci przygotowują trzy oszacowania: najbardziej optymistyczne (a), najbardziej prawdopodobne (m) i najbardziej pesymistyczne (b), a ostateczna szacowana wartość (E) jest ich średnią ważoną arytmetyczną.
szerokopasmowa technika delficka	Bazująca na wiedzy eksperckiej technika estymacji opierająca się na zbiorowej wiedzy członków zespołu polegająca na dokładnym szacowaniu.

9. Załącznik E – Znaki towarowe

CMMI® jest znakiem towarowym zarejestrowanym w Urzędzie Patentów i Znaków Towarowych Stanów Zjednoczonych przez Carnegie Mellon University.

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

TMMi® jest zastrzeżonym znakiem towarowym TMMi Foundation.

TPI-Next® jest zastrzeżonym znakiem towarowym firmy Sogeti, Holandia.

10. Indeks

Słowa kluczowe są zdefiniowane w Słowniku terminów testowych ISTQB® (<http://glossary.istqb.org/>).

analiza ryzyka, 19, 28, 30, 33
anomalia, 57
awaria, 32, 57, 58, 60, 62, 71
awaria wewnętrzna, 31, 71, 72
awaria zewnętrzna, 71, 72
cel testów, 20, 36, 49
cykl wytwarzania oprogramowania, 15, 18, 20, 21, 23, 69
defekt, 51, 57, 58, 61, 62, 67, 71, 72
hybrydowy model wytwarzania oprogramowania, 15, 23, 36, 54, 60
IDEAL, 38, 89
iteracyjny model wytwarzania oprogramowania, 15, 23, 36, 54
komitet ds. zarządzania defektami, 59
koszt jakości, 57, 71, 72
krytyczność defektu, 52, 54, 72
łagodzenie ryzyka, 26, 28, 30, 31, 33
metoda określania celów SMART, 36
metryka, 24, 38, 40, 41, 49, 50, 51, 52, 89
metryki, 51
metryki procesu, 49
metryki produktu, 49
metryki projektu, 49
monitorowanie ryzyka, 28
monitorowanie testów, 18, 19, 20, 26, 27, 28, 31, 49, 50, 66, 87
nadzór nad testami, 18, 19, 26, 27, 31, 49, 50, 66, 87
ocena ryzyka, 26, 29, 32
określenie ryzyka, 19, 28, 29, 32
plan testów, 18, 19, 21, 22, 26, 34, 36, 61
planowanie testów, 18, 19, 26, 27, 32, 34, 39, 49, 53, 66, 87
podstawowa przyczyna, 40, 62
poker planistyczny, 55, 56, 89
postęp testów, 19, 21, 41, 51
powstrzymanie fazowe, 57, 63
poziom ryzyka, 28, 30, 31, 73
poziom testów, 18, 35, 47, 55, 66
prawdopodobieństwo ryzyka, 29, 30, 32, 33
priorytet, 59, 60, 61
przepływ pracy dotyczący defektu, 14, 48, 58, 60, 84
przyczyna podstawowa, 52
przyrostowy model wytwarzania oprogramowania, 15, 38
raport o defekcie, 57, 58, 59, 60, 61, 62, 63
retrospektywa, 13, 20, 28, 32, 38, 40, 41, 42, 63, 68
ryzyko jakościowe, 28, 29, 30, 32, 33, 51
ryzyko produktowe, 19, 28, 29, 50, 51, 52, 60, 66
ryzyko projektowe, 29
sekwencyjny model wytwarzania oprogramowania, 23, 25, 34, 38, 41, 42, 56, 58, 60, 61
strategia testów, 14, 15, 21, 25, 27, 34, 66, 67, 69
szacowanie testów, 53, 54, 55
test porównawczy, 25
testowanie funkcjonalne, 25
testowanie niefunkcjonalne, 25
testowanie oparte na ryzyku, 14, 16, 28, 31, 32, 33, 35, 73
testowanie w oparciu o doświadczenie, 35
TPI NEXT, 39, 40
typ testu, 15
ukończenie testów, 18, 20, 49, 50, 51, 66
usprawnianie procesu testowego, 38, 39, 40
wpływ ryzyka, 29, 30, 32, 33
wynik fałszywie negatywny, 57
zapobieganie defektom, 71, 72, 73
zarządzanie ryzykiem, 14, 22, 28, 33
zintegrowany model dojrzałości testów (TMMi), 39, 40
zwrot z inwestycji (ROI), 39, 44, 45