

Certyfikowany Tester

Poziom Zaawansowany
Zarządzanie Testami
(CTAL – TM)

Sylabus

wersja 3.0

wersja tłumaczenia PL 3.0.0



International Software Testing Qualifications Board



Informacja o prawach autorskich

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

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

Copyright © 2023, autorami aktualizacji 3.0 są Horst Pohlmann (właściciel produktu, wiceprzewodniczący 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 © 2010–2012 autorzy z Podgrupy Roboczej ds. Testów na Poziomie Zaawansowanym: Rex Black (przewodniczący), Judy McKay (wiceprzewodnicząca), Graham Bath, Debra Friedenberga, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto.

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

Tłumaczenie z języka angielskiego: Adam Roman.

Korekta: Monika Petri-Starego. Przegląd tłumaczenia: Monika Petri-Starego, Lucjan Stapp.

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

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

Historia zmian

Wersja	Data	Uwagi
ISEB v1.1	4.09.2001	Sylabus ISEB Practitioner
ISTQB 1.2E	09.2003	Sylabus ISTQB Advanced Level opracowany przez EOQ-SG
V2007	12.10.2007	Sylabus wersja 2007
D100626	10.06.2010	Wprowadzenie zmian przyjętych w 2009 r., rozdzielanie poszczególnych rozdziałów dla oddzielnych modułów
D101227	10.12.2010	Poprawki redakcyjne
D2011	31.10.2011	Zmiana polegająca na podziale sylabusa, przeróbka celów nauczania i zmiany tekstu w celu dostosowania go do celów nauczania
Alpha 2012	09.02.2012	Włączenie wszystkich komentarzy od rad krajowych
Beta 2012	26.03.2012	Włączenie uwag od rad krajowych otrzymanych w terminie od wydania wersji Alpha
Beta 2012	07.04.2012	Wersja beta przekazana Zgromadzeniu Ogólnemu ISTQB®
Beta 2012	08.06.2012	Wersja z poprawkami przekazana do rad krajowych
Beta 2012	27.06.2012	Uwzględnienie uwag grupy ds. egzaminów i ds. słownika
RC 2012	15.08.2012	Wersja kandydacka – uwzględniono ostateczne poprawki rad krajowych
Beta v3.0	31.10.2023	Włączenie wszystkich uwag otrzymanych od rad krajowych dotyczących wszystkich sekcji (BOIncs) z przeglądu wersji Alpha
POST Beta v3.0	31.01.2024	Włączenie wszystkich uwag otrzymanych od rad krajowych dotyczących wszystkich sekcji (BOIncs) z przeglądu wersji Beta
POST Beta v3.0	29.02.2024	Drobne modyfikacje w korekcie tekstu
RC v3.0	28.03.2024	Wersja kandydacka do wydania – zawiera najnowsze zmiany w szablonie formalnym zaproponowane przez Process WG
V.3.0	3.05.2024	Poprawki po publikacji; usunięto jedynie literówki i niespójności

Historia zmian polskiej wersji dokumentu

Wersja	Data	Uwagi
3.0.0	14.03.2024	Opublikowanie polskiego tłumaczenia sylabusa

Spis treści

Informacja o prawach autorskich	2
Historia zmian	3
Historia zmian polskiej wersji dokumentu	3
Podziękowania	7
Wstęp	9
Cel niniejszego sylabusu	9
Certyfikowany tester Poziom Zaawansowany – Zarządzanie Testami w procesie testowania	9
Ścieżki kariery dla testerów	9
Cele biznesowe	10
Cele nauczania objęte egzaminem oraz poziomy wiedzy	11
Egzamin certyfikacyjny z zarządzania testami na poziomie zaawansowanym	11
Akredytacja	11
Odniesienia do norm	12
Poziom szczegółowości	12
Struktura sylabusu	12
Podstawowe założenia sylabusu	13
1. Zarządzanie czynnościami testowymi – 750 minut	15
1.1. Proces testowy	17
1.1.1. Czynności związane z planowaniem testów	17
1.1.2. Czynności związane z monitorowaniem testów i nadzorem nad testami	18
1.1.3. Czynności związane z ukończeniem testów	19
1.2. Kontekst testowania	19
1.2.1. Interesariusze testów	20
1.2.2. Znaczenie wiedzy interesariuszy w zarządzaniu testami	20
1.2.3. Zarządzanie testami w hybrydowym modelu wytwarzania oprogramowania	22
1.2.4. Czynności związane z zarządzaniem testami w różnych modelach wytwarzania oprogramowania	22
1.2.5. Czynności związane z zarządzaniem testami na różnych poziomach testów	23
1.2.6. Czynności związane z zarządzaniem testami w różnych typach testów	24

1.2.7.	Czynności związane z zarządzaniem testami w celu planowania, monitorowania i nadzoru nad testami	25
1.3.	Testowanie oparte na ryzyku	26
1.3.1.	Testowanie jako czynność łagodząca ryzyko	27
1.3.2.	Identyfikacja ryzyk jakościowych	27
1.3.3.	Ocena ryzyka jakościowego	28
1.3.4.	Łagodzenie ryzyka jakościowego poprzez odpowiednie testy	29
1.3.5.	Techniki testowania oparte na ryzyku	31
1.3.6.	Metryki sukcesu i trudności związane z testowaniem opartym na ryzyku	32
1.4.	Projektowa strategia testowa	32
1.4.1.	Wybór podejścia do testów	33
1.4.2.	Analiza organizacyjnej strategii testowej, kontekstu projektu i innych aspektów	34
1.4.3.	Definicja celów testów	35
1.5.	Doskonalenie procesu testowego	36
1.5.1.	Proces doskonalenia testów (IDEAL)	37
1.5.2.	Doskonalenie procesu testowego oparte na modelu	37
1.5.3.	Podejście analityczne do doskonalenia procesu testowego	38
1.5.4.	Retrospektywy	40
1.6.	Narzędzia testowe	41
1.6.1.	Dobre praktyki dotyczące wprowadzania narzędzi	41
1.6.2.	Techniczne i biznesowe aspekty decyzji dotyczących narzędzi	42
1.6.3.	Proces wyboru narzędzia i ocena zwrotu z inwestycji	43
1.6.4.	Cykl życia narzędzia	44
1.6.5.	Metryki narzędzia	45
2.	Zarządzanie produktem – 390 minut	46
2.1.	Metryki testów	47
2.1.1.	Metryki czynności związanych z zarządzaniem testami	47
2.1.2.	Monitorowanie, nadzór i ukończenie testów	48
2.1.3.	Raportowanie testów	49
2.2.	Szacowanie testów	50

2.2.1.	Oszacowanie, jakie czynności będzie obejmować testowanie	50
2.2.2.	Czynniki, które mogą wpływać na wysiłek testowy	51
2.2.3.	Wybór technik szacowania testów	52
2.3.	Zarządzanie defektami	54
2.3.1.	Cykl życia defektu.....	54
2.3.2.	Interdyscyplinarne zarządzanie defektami	56
2.3.3.	Specyfika zarządzania defektami w zespołach zwinnych	57
2.3.4.	Wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania	58
2.3.5.	Informacje zawarte w raporcie o defekcie	58
2.3.6.	Określanie działań doskonalących proces przy użyciu informacji z raportów o defektach	60
3.	Zarządzanie zespołem – 225 minut.....	62
3.1.	Zespół testowy.....	63
3.1.1.	Typowe umiejętności w ramach czterech obszarów kompetencji.....	63
3.1.2.	Analiza wymaganych umiejętności członków zespołu testowego.....	64
3.1.3.	Ocena umiejętności członków zespołu testowego	65
3.1.4.	Rozwijanie umiejętności członków zespołu testowego	66
3.1.5.	Umiejętności kierownicze wymagane do zarządzania zespołem testowym	67
3.1.6.	Czynniki motywujące lub demotywuujące zespół testowy w określonych sytuacjach	68
3.2.	Relacje z interesariuszami	68
3.2.1.	Koszt jakości	69
3.2.2.	Stosunek kosztów do korzyści testowania	69
4.	Bibliografia.....	72
5.	Dodatek A – cele nauczania i poziomy wiedzy	74
6.	Dodatek B – macierz powiązań między celami biznesowymi a celami nauczania.....	76
7.	Dodatek C – nota wydania	79
8.	Dodatek D – Słowa kluczowe specyficzne dla dziedziny.....	81
9.	Dodatek E – Znaki towarowe	82
10.	Indeks	83

Podziękowania

Niniejszy dokument został formalnie opublikowany przez Zgromadzenie Ogólne ISTQB® w dniu 3.05.2024 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 przegląd techniczny (wersji beta), Julii Sabatine za korektę, a także zespołowi recenzentów i radom krajowym za ich sugestie i uwagi.

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 Posthuma, 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, Rabih 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, Seunghye Choi, Stuart Reid, Dmitrij Nikolajev, Mantas Aniulis, Monika Stoecklein-Olsen, Adam Roman, Mahmoud Khalaili, Ingvar Nordström, Beata Karpinska, Armin Born, Ferdinand Gramsamer, Mergole Kuaté, Thomas Letzkus, Nishan Portoyan, Ainsley Rood, Lloyd Roden, Sarah Ireton.

Sylabus Certyfikowany Tester Poziom Zaawansowany Kierownik Testów 2010–2012 został opracowany przez główny zespół 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 głównego składają podziękowania zespołowi recenzentów oraz wszystkim radom krajowym za ich sugestie i uwagi.

W momencie ukończenia sylabusa 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.

Wstęp

Cel niniejszego sylabusu

Niniejszy sylabus stanowi podstawę egzaminu ISTQB® na poziomie zaawansowanym w obszarze zarządzania testami. ISTQB® udostępnia sylabus:

1. Radom krajowym, w celu przetłumaczenia go na języki lokalne oraz dokonania akredytacji dostawców szkoleń (rady krajowe mogą dostosować sylabus do specyfiki języka oraz zmienić odniesienia do literatury, aby uwzględnić publikacje lokalne);
2. Organom certyfikacyjnym, w celu przygotowania pytań egzaminacyjnych w językach lokalnych, zgodnych z celami nauczania określonymi w niniejszym sylabusie;
3. Akredytowanym przez ISTQB® dostawcom szkoleń, w celu opracowania materiałów dydaktycznych oraz określenia odpowiednich metod nauczania;
4. Kandydatom do certyfikacji, w celu przygotowania do egzaminu certyfikacyjnego (ISTQB® rekomenduje udział w szkoleniu akredytowanym przez ISTQB® przed przystąpieniem do egzaminu na poziomie zaawansowanym);
5. Międzynarodowej społeczności specjalistów w dziedzinie inżynierii oprogramowania i systemów, w celu wspierania rozwoju zawodu testera oprogramowania i systemów oraz tworzenia publikacji, takich jak książki i artykuły.

Certyfikowany tester Poziom Zaawansowany – Zarządzanie Testami w procesie testowania

Niniejsza kwalifikacja na poziomie zaawansowanym jest przeznaczona dla wszystkich osób uczestniczących w zarządzaniu testowaniem oprogramowania. Obejmuje to stanowiska takie jak testerzy, konsultanci ds. testów, kierownicy testów, testerzy testowania akceptacyjnego przez użytkownika, scrum masterzy, kierownicy projektów oraz właściciele produktów. Ponadto kwalifikacja na poziomie zaawansowanym w obszarze zarządzania testami jest odpowiednia dla osób, które pragną poszerzyć swoją wiedzę w dziedzinie testowania oprogramowania, w tym kierowników projektów, kierowników ds. jakości, właścicieli produktów, kierowników ds. rozwoju oprogramowania, analityków biznesowych, dyrektorów IT oraz konsultantów ds. zarządzania. Posiadacze certyfikatu ISTQB® w zakresie zarządzania testami na poziomie zaawansowanym będą mogli kontynuować ścieżkę do kwalifikacji w testowaniu oprogramowania na poziomie eksperta ISTQB®. Certyfikat ISTQB® Certyfikowany Tester Poziom Zaawansowany Kierownik Testów (lub Zarządzanie Testami) jest ważny bezterminowo i nie wymaga odnawiania. Dokument ten jest uznawany globalnie i potwierdza kompetencje zawodowe oraz wiarygodność kandydatów w zakresie zarządzania testami.

Ścieżki kariery dla testerów

Program opracowany przez ISTQB® zapewnia profesjonalistom zajmującym się testowaniem wsparcie na każdym etapie ich kariery. Osoby, które uzyskają certyfikat ISTQB® Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami, mogą być również zainteresowane innymi certyfikatami w obszarze ścieżki głównej (ang. *core track*) na poziomie zaawansowanym (tj. Certyfikowany Tester Poziom Zaawansowany Analityk Testów, Certyfikowany Tester Poziom Zaawansowany Techniczny Analityk Testów i Certyfikowany Tester Poziom Zaawansowany

Inżynier Automatyzacji Testów), a następnie certyfikatami ISTQB® na poziomie eksperckim (tj. Certyfikowany Tester Poziom Ekspercki Zarządzanie Testami lub Certyfikowany Tester Poziom Ekspercki Doskonalenie Procesu Testowego). Osoby pragnące rozwijać swoje kompetencje w zakresie praktyk testowania zwinnego powinny rozważyć zdobycie certyfikatu techniczny tester zwinny lub certyfikatu ATLaS (*Agile Test Leadership at Scale*). Z kolei ścieżka specjalistyczna umożliwia pogłębioną specjalizację w dziedzinach, w których stosuje się wyspecjalizowane podejścia i czynności testowe (np. automatyzacja testów, testowanie z wykorzystaniem sztucznej inteligencji lub testowanie aplikacji mobilnych), bądź w których rozwija się specjalistyczną wiedzę w zakresie testowania dla określonych sektorów przemysłu (takich jak przemysł samochodowy (ang. *automotive*) czy gry). Najnowsze informacje dotyczące systemu certyfikacji ISTQB® dostępne są na stronie www.istqb.org.

Cele biznesowe

W tym podrozdziale wymieniono 11 celów biznesowych (ang. *business outcomes*), które powinna realizować osoba otrzymująca certyfikat zarządzanie testami na poziomie zaawansowanym.

Umiejętności testera posiadającego certyfikat Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami:

TM-01	Zarządzanie testowaniem w różnych projektach wytwarzania oprogramowania z zastosowaniem procesów zarządzania testami ustanowionych dla zespołu projektowego lub organizacji testowej
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 czynności doskonalących procesy testowe w projektach lub strumieniach produktów związanych z rozwojem oprogramowania oraz wnoszenie wkładu w organizacyjne inicjatywy na rzecz doskonalenia procesu testowego
TM-10	Planowanie czynności testowych, w tym wymaganej infrastruktury testowej, szacowanie wysiłku wymaganego do przeprowadzenia testów
TM-11	Tworzenie raportów o defektach i przepływu pracy dotyczącego defektów odpowiedniego w kontekście danego cyklu wytwarzania oprogramowania

Cele nauczania objęte egzaminem oraz poziomy wiedzy

Cele nauczania (ang. *learning objectives*) wspierają realizację celów biznesowych i stanowią podstawę do opracowywania egzaminów certyfikacyjnych dla testerów na poziomie Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami.

Zasadniczo wszystkie treści niniejszego sylabusu mogą podlegać egzaminowaniu na poziomie K1, z wyjątkiem wstępu, bibliografii, podsumowania oraz załączników. Oznacza to, że od kandydata może być wymagane rozpoznanie, zapamiętanie lub przywołanie słowa kluczowego bądź pojęcia przedstawionego w każdym z trzech głównych rozdziałów sylabusu. Poziomy wiedzy związane z poszczególnymi celami nauczania przedstawiono na początku każdego rozdziału. Zostały one sklasyfikowane w następujący sposób:

- K2: zrozumieć (ang. *understand*),
- K3: zastosować (ang. *apply*),
- K4: przeanalizować (ang. *analyze*).

Dalsze szczegóły oraz przykłady celów nauczania zawarte są w Załączniku A.

Ponadto, definicje pojęć wymienionych jako słowa kluczowe pod tytułami rozdziałów należy pamiętać (K1), nawet jeśli nie zostały one wyraźnie wskazane w celach nauczania.

Egzamin certyfikacyjny z zarządzania testami na poziomie zaawansowanym

Egzamin z zarządzania testami na poziomie zaawansowanym opiera się na treściach niniejszego sylabusu. W trakcie odpowiadania na pytania egzaminacyjne może być konieczne odwołanie się do materiału obejmującego więcej niż jeden rozdział sylabusu. Zakres egzaminu obejmuje wszystkie części sylabusu z wyjątkiem wstępu, załączników i bibliografii. W dokumencie znajdują się również odniesienia do norm i literatury (rozdział 4), jednak ich treść nie może być przedmiotem egzaminu poza informacjami podanymi wprost w niniejszym sylabusie.

Szczegółowe informacje dostępne są w dokumencie: „*Exam Structures and Rules*”.

Przed przystąpieniem do egzaminu certyfikacyjnego na poziomie zaawansowanym kandydaci muszą posiadać certyfikat ISTQB® na poziomie podstawowym. Zaleca się jednak, aby kandydat dysponował przynajmniej minimalnym doświadczeniem w tworzeniu lub testowaniu oprogramowania, np. sześciomiesięcznym doświadczeniem jako tester lub programista.

Akredytacja

Rada krajowa ISTQB® może przeprowadzać akredytację dostawców szkoleń, którzy oferują materiały dydaktyczne zgodne z niniejszym sylabusem. Szczegółowe wytyczne dotyczące akredytacji należy uzyskać bezpośrednio od rady krajowej lub właściwego organu akredytującego. Szkolenia posiadające akredytację są uznawane za zgodne z sylabusem i mogą obejmować egzamin ISTQB®.

Wytyczne akredytacyjne odnoszące się do niniejszego sylabusu są zgodne z ogólnymi zasadami akredytacji opublikowanymi przez Grupę Roboczą ds. Zarządzania Procesami i Zgodności ISTQB®.

Odniesienia do norm

W sylabusie Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami znajdują się odniesienia do norm (np. ISO, IEC oraz IEEE). Odniesienia te stanowią ramy (jak w przypadku odniesień do normy ISO 25010 dotyczącej charakterystyk jakościowych) lub źródło dodatkowych informacji dla czytelników, którzy wyrażą zainteresowanie określonym obszarem wiedzy. Treść norm nie podlega egzaminowaniu. Więcej informacji na temat norm znajduje się w rozdziale 4.

Poziom szczegółowości

Poziom szczegółowości niniejszego sylabusa pozwala na prowadzenie spójnych szkoleń i egzaminów na poziomie międzynarodowym. Aby osiągnąć ten cel, sylabus składa się z:

- celów biznesowych opisujących intencje sylabusa Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami,
- listy słów kluczowych, które studenci muszą być w stanie zapamiętać,
- celów nauczania dla każdego obszaru wiedzy, opisujące efekty uczenia się, które należy osiągnąć,
- opisu kluczowych pojęć, w tym odniesienia do uznanych źródeł lub norm.

Treść sylabusa nie stanowi opisu całego obszaru wiedzy dotyczącego testowania oprogramowania; odzwierciedla ona poziom szczegółowości, jaki należy uwzględnić w szkoleniach poziomu zaawansowanego zarządzania testami. Koncentruje się ona na koncepcjach i technikach testowania, które mogą mieć zastosowanie do wszystkich projektów wytwarzania oprogramowania, bez względu na przyjęty model wytwarzania oprogramowania.

Struktura sylabusa

Niniejszy sylabus obejmuje trzy rozdziały zawierające treści egzaminacyjne. Nagłówek najwyższego poziomu każdego rozdziału wskazuje minimalny czas wymagany dla akredytowanych szkoleń na omówienie jego zawartości; czas nie jest określony poniżej poziomu rozdziału. Dla akredytowanych szkoleń sylabus przewiduje łącznie co najmniej 22,75 godzin zajęć, podzielonych pomiędzy trzy rozdziały w następujący sposób:

Rozdział 1: Zarządzanie czynnościami testowymi (750 minut)

- Uczestnik uczy się objaśniać czynności związane z zarządzaniem testami, w tym planowanie testów, monitorowanie testów, nadzór nad testami oraz ukończenie testów.
- Uczestnik uczy się definiować projektową strategię testów, obejmującą cele testów oraz wybierać odpowiednie podejście do testów zgodnie z organizacyjną strategią testową i kontekstem projektu.
- Uczestnik uczy się zarządzać projektami w różnych kontekstach.
- Uczestnik uczy się stosować podejście oparte na ryzyku, aby skoncentrować czynności testowe na zidentyfikowanych ryzykach.
- Uczestnik uczy się przeprowadzać czynności doskonalące proces testowy poprzez retrospektywę projektu lub iteracji.
- Uczestnik uczy się doskonalić wsparcie narzędziowe procesu testowego, uwzględniając ryzyka, koszty i korzyści związane z wykorzystaniem narzędzi.

Rozdział 2: Zarządzanie produktem (390 minut)

- Uczestnik uczy się monitorować i kontrolować testowanie w celu osiągnięcia celów testów przy użyciu metryk oraz oceniać i raportować postęp testów.
- Uczestnik uczy się dobierać odpowiednie techniki szacowania testów zgodnie z różnymi modelami wytwarzania oprogramowania.
- Uczestnik uczy się określać przepływ pracy dotyczący defektu w ramach zarządzania defektami, dopasowując go do sekwencyjnego, zwinnego i hybrydowego modelu wytwarzania oprogramowania.

Rozdział 3: Zarządzanie zespołem (225 minut)

- Uczestnik uczy się analizować kontekst projektu w celu określenia wymaganych umiejętności w zespole testowym.
- Uczestnik uczy się zarządzać zespołem w oparciu o podejście „cały zespół”.
- Uczestnik uczy się określać uzasadnienie biznesowe czynności testowych w projekcie.

Uwaga: Dla każdego celu nauczania w niniejszym sylabusie istnieje odpowiadająca mu sekcja z treścią (np. dla celu LO-1.2.3 znajduje się sekcja 1.2.3).

Podstawowe założenia sylabusu

Niniejszy sylabus jest przeznaczony dla wszystkich osób, które pragną osiągnąć zaawansowany poziom kompetencji w zakresie zarządzania testami, w tym kierowników testów, analityków testów, inżynierów testów, konsultantów testów, koordynatorów testów, liderów testów oraz kierowników projektów. Sylabus jest zgodny z sylabusem Certyfikowany Tester Poziom Podstawowy w wersji 4.0, który zapewnia podstawową wiedzę i zrozumienie w dziedzinie testowania oprogramowania.

Sylabus obejmuje dwie główne role w testowaniu: rolę związaną z zarządzaniem testami i rolę związaną z testowaniem.

Rola związana z zarządzaniem testami, znana także jako rola kierownika testów w kontekście sekwencyjnego modelu wytwarzania, odpowiada za ogólny proces testowy, zespół testowy oraz zarządzanie testami. Obejmuje to określanie strategii testów, planowanie czynności testowych, monitorowanie i kontrolowanie postępów testów, raportowanie wyników, a także zarządzanie ryzykami i problemami związanymi z testami. Rola ta również odpowiada za zapewnienie zgodności celów testów z potrzebami biznesowymi i interesariuszy, a czynności testowe były koordynowane z innymi interesariuszami projektu.

Rola związana z testowaniem obejmuje ocenę testów, zarządzanie defektami oraz czynności związane z ukończeniem testów. Wykorzystuje ona różne techniki testowania w celu zapewnienia jakości i niezawodności produktów pracy oraz testowanego systemu. Rola ta korzysta również z narzędzi testowych i automatyzacji w celu wsparcia procesu testowego oraz poprawy efektywności i skuteczności testów.

Zakres czynności przypisanych do tych dwóch ról może się różnić w zależności od kontekstu, w tym projektu, produktu, kompetencji zespołu i struktury organizacyjnej (por. sylabus Certyfikowany Tester Poziom Podstawowy v4.0).

Termin „członek zespołu testowego” odnosi się do każdej osoby pełniącej rolę związaną z zarządzaniem testami lub testowaniem, która wykonuje czynności testowe, niezależnie od kontekstu organizacyjnego i innych ról. Zespoły testowe składają się z osób o różnych umiejętnościach i kompetencjach, posiadających różne poziomy doświadczenia i certyfikacji (poziom podstawowy, zaawansowany lub ekspercki). Członkowie zespołu mogą pełnić różne role i obowiązki w zależności od stosowanego podejścia i modelu procesu testowego, w tym testowania zwinnego, opartego na modelu lub opartego na ryzyku.

Sylabus koncentruje się na zarządzaniu testami na poziomie *projektu*, a nie organizacji, zapewniając informacje przydatne głównie w zarządzaniu testami na tym poziomie. W związku z tym niniejszy sylabus dostarcza informacje, które mogą być wykorzystane do zarządzania testami na poziomie projektu, ale w mniejszym stopniu do zarządzania testami na poziomie organizacji.

Hybrydowy model wytwarzania oprogramowania, opisywany w sylabusie, łączy elementy różnych modeli cyklu wytwarzania oprogramowania, takich jak model V, iteracyjny, przyrostowy czy zwinny. Ma on na celu wykorzystanie mocnych stron oraz ograniczenie słabości każdego modelu, w zależności od kontekstu i potrzeb projektu. Przykładowo, hybrydowe podejście do wytwarzania oprogramowania może wykorzystywać model V w fazach planowania i analizy wymagań, a model zwinny w fazach projektowania, wytwarzania i testowania. Alternatywnie, hybrydowe podejście do wytwarzania oprogramowania może wykorzystywać model iteracyjny do ogólnego zarządzania projektem, przy jednoczesnym wykorzystaniu modelu przyrostowego w każdej iteracji oraz modelu zwinnego dla poszczególnych przyrostów. Hybrydowy model wymaga wysokiej elastyczności, skutecznej komunikacji i współpracy między interesariuszami oraz jasnego zrozumienia celów, ryzyk i ograniczeń każdej fazy i modelu.

Zgodnie z niniejszym sylabusem i słownikiem terminów testowych ISTQB®, strategia testów opisuje sposób prowadzenia testów w celu osiągnięcia określonych celów w danym kontekście. Określa ona ogólny zakres, podejście i zasoby potrzebne do testowania systemu lub produktu i jest zwykle dokumentowana w planie testów lub innych dokumentach stosownie do kontekstu. Na strategię testów wpływa organizacyjna strategia testowa, która określa ogólne podejście do testów w organizacji. Strategia testów może także dotyczyć pojedynczego poziomu testów lub typu testu, koncentrując się na konkretnych celach, zadaniach i kryteriach. Termin ogólny „strategia testów” może być stosowany w różnych kontekstach (projekt, organizacja, produkt).

Podejście do testów określa sposób realizacji zadań testowych, w tym wybór i kombinację poziomów testów, typów testów i technik testowania statycznego oraz dynamicznego, a także innych praktyk, takich jak testowanie skryptowe, manualne czy z wykorzystaniem pseudo-wyrocni (ang. *back-to-back testing*). Podejście wybrane przez rolę związaną z zarządzaniem testami stanowi kluczową decyzję przy formułowaniu odpowiedniej strategii testów w danym kontekście.

1. Zarządzanie czynnościami testowymi – 750 minut

Słowa kluczowe

analiza ryzyka, cel testów, cykl wytwarzania oprogramowania, doskonalenie procesu testowego, 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 niefunkcjonalne, testowanie oparte na ryzyku, testowanie oparte na doświadczeniu, typ testów, TPI NEXT, ukończenie testów, wpływ ryzyka, zarządzanie ryzykiem

Słowa kluczowe specyficzne dla dziedziny

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

Cele nauczania dla rozdziału 1

1.1 Proces testowy

TM-1.1.1 (K2) Podsumować planowanie testów.

TM-1.1.2 (K2) Podsumować monitorowanie testów i nadzór nad testami.

TM-1.1.3 (K2) Podsumować ukończenie testów.

1.2 Kontekst testowania

TM-1.2.1 (K2) Porównać powody, dla których różni interesariusze są zainteresowani testowaniem.

TM-1.2.2 (K2) Wyjaśnić, dlaczego wiedza interesariuszy jest ważna w zarządzaniu testami.

TM-1.2.3 (K2) Wyjaśnić testowanie w hybrydowym modelu wytwarzania oprogramowania.

TM-1.2.4 (K2) Podsumować czynności związane z zarządzaniem testami w różnych cyklach wytwarzania oprogramowania.

TM-1.2.5 (K2) Porównać czynności związane z zarządzaniem testami na różnych poziomach testów.

TM-1.2.6 (K2) Porównać czynności związane z zarządzaniem testami w różnych typach testów.

TM-1.2.7 (K4) Przeanalizować dany projekt i określić 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) Wyjaśnić różne środki podejmowane w ramach testowania opartego na ryzyku w celu reagowania na ryzyka.

TM-1.3.2 (K2) Podać przykłady różnych technik, które kierownik testów może wykorzystać do określenia ryzyk jakościowych produktu.

TM-1.3.3 (K2) Podsumować czynniki określające poziomy ryzyka związane z jakością produktu.

TM-1.3.4 (K4) Wybrać odpowiednie czynności testowe w celu łagodzenia ryzyk zgodnie z poziomem ryzyka w danym kontekście.

TM-1.3.5 (K2) Rozróżniać między przykładami ciężkich i lekkich technik testowania opartych na ryzyku.

TM-1.3.6 (K2) Podać przykłady wskaźników sukcesu i trudności związanych z testowaniem opartym na ryzyku.

1.4 Projektowa strategia testowa

TM-1.4.1 (K2) Objaśnić typowe wybory dotyczące podejścia do testów.

TM-1.4.2 (K4) Przeanalizować organizacyjną strategię testową i kontekst projektu w celu wyboru odpowiedniego podejścia do testów.

TM-1.4.3 (K3) Wykorzystać metodę określania celów SMART do określenia mierzalnych celów testów i kryteriów wyjścia.

1.5 Doskonalenie procesu testowego

TM-1.5.1 (K2) Objaśnić, jak wykorzystać model IDEAL do doskonalenia procesu testowego w danym projekcie.

TM-1.5.2 (K2) Podsumować podejścia do doskonalenia procesu testowego oparte na modelu i rozumieć, jak zastosować je w kontekście projektu.

TM-1.5.3 (K2) Podsumować analityczne podejście do doskonalenia procesów i rozumieć, jak zastosować je w kontekście projektu.

TM-1.5.4 (K3) Wdrożyć retrospektywy projektu lub iteracji w celu oceny procesu testowego i odkrycia obszarów testowania wymagających poprawy.

1.6 Narzędzia testowe

TM-1.6.1 (K2) Podsumować najlepsze praktyki w zakresie wprowadzania narzędzi.

TM-1.6.2 (K2) Objaśnić wpływ różnych aspektów technicznych i biznesowych przy podejmowaniu decyzji o wyborze odpowiedniego rodzaju narzędzia.

TM-1.6.3 (K4) Przeanalizować daną sytuację w celu opracowania planu wyboru narzędzi, obejmującego ryzyka, koszty i korzyści.

TM-1.6.4 (K2) Rozróżniać etapy cyklu życia narzędzi.

TM-1.6.5 (K2) Podać przykłady gromadzenia i oceny metryk za pomocą narzędzi.

1.1. Proces testowy

Wstęp. Sylabus Certyfikowany Tester Poziom Podstawowy w wersji 4.0 opisuje czynności procesu testowego, obejmujące: planowanie testów, monitorowanie testów i nadzór nad testami, analizę testów, projektowanie testów, implementację testów, wykonywanie testów oraz ukończenie testów. W sylabusie tym wskazano, że czynności te są często realizowane iteracyjnie lub równolegle, w zależności od wybranego modelu cyklu wytwarzania oprogramowania (ang. *Software Development Life Cycle*, SDLC) i kontekstu projektu. Wymagane jest zazwyczaj dostosowanie tych czynności do specyfiki produktu i projektu.

W niniejszym 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, oczekiwanych wyników oraz uczestników testów (interesariuszy testów).
- **Monitorowanie testów i nadzór nad testami:** śledzenie postępów testów, wyników testów oraz odchyłeń od oczekiwanych wyników; podejmowanie działań korygujących w razie potrzeby; raportowanie statusu testów i wyników odpowiednim interesariuszom.
- **Ukończenie testów:** finalizacja i archiwizacja testów, ocena procesu testowego i testowanego produktu, określanie działań doskonalących proces testowy oraz przekazanie informacji o ukończeniu testów odpowiednim interesariuszom.

Norma ISO/IEC/IEEE 29119-2 definiuje procesy zarządzania testami obejmujące te czynności. Procesy te mogą być stosowane na różnych poziomach testów, takich jak projekt, program lub portfel projektów. Każdy poziom testów może posiadać własny plan testów, zgodny z planem testów wyższego poziomu.

1.1.1. Czynności związane z planowaniem testów

W tej sekcji skupiono się na czynnościach związanych z planowaniem testów o różnym zakresie np. całego projektu, poziomu testów, typu testów lub wydania/iteracji w zwinnym wytwarzaniu oprogramowania. W zależności od zakresu, planowanie testów może rozpoczynać się i kończyć w różnych momentach procesu wytwarzania. Planowanie testów to działanie polegająca na identyfikacji czynności i zasobów niezbędnych do osiągnięcia celów testów określonych w polityce testów. Planowanie testów powinno rozpocząć się jak najwcześniej w procesie wytwarzania, najlepiej przed zidentyfikowaniem wymagań, i powinno być aktualizowane w miarę postępu projektu. Planowanie testów jest często procesem iteracyjnym, który wymaga ponownego planowania w trakcie projektu w celu uwzględnienia zmian i informacji zwrotnych.

Następujące zadania (podobnie jak te zawarte w normie ISO/IEC/IEEE 29119-2), są częścią planowania testów:

Zrozumienie kontekstu i organizacja planowania testów. Zrozumienie kontekstu organizacji (np. polityki testów i organizacyjnej strategii testowej), zakresu testów i elementu testowego (tj. testowanego produktu pracy) ma kluczowe znaczenie dla planowania testów (patrz podrozdział 1.2). Obejmuje to również wszystkie czynności niezbędne do opracowania

planu testów oraz uzyskania zgody na te czynności i harmonogram od interesariuszy (np. właściciela produktu, kierownika projektu lub kierownika zespołu programistów).

Identyfikacja i analiza ryzyk produktowych. Analiza ryzyka obejmuje identyfikację i ocenę potencjalnego wpływu i prawdopodobieństwa wystąpienia ryzyk produktowych w ramach planowania testów. Więcej szczegółów na temat ryzyk produktowych znajduje się w podrozdziale 1.3.

Określenie podejścia do zarządzania ryzykiem. Na podstawie analizy ryzyka wybiera się odpowiednie podejścia do zarządzania ryzykiem i dokumentuje je w planie testów. Mogą one obejmować działania zapobiegawcze, naprawcze lub łagodzące w celu wyeliminowania zidentyfikowanych ryzyk.

Zdefiniowanie podejścia do testowania oraz oszacowanie i przydzielenie zasobów testowych. W oparciu o organizacyjną strategię testową, obowiązujące normy, wszelkie ograniczenia wynikające z projektu oraz podejścia do zarządzania ryzykiem, definiuje się podejście do testów dla bieżącego zakresu testów (patrz podrozdział 1.4). Po zdefiniowaniu podejścia do testów ważne jest oszacowanie wymaganych zasobów testowych, takich jak personel testujący, narzędzia, środowiska testowe i dane testowe, oraz przydzielenie tych zasobów do czynności testowych.

Opracowanie planu testów. Plan testów musi zostać zaakceptowany przez wszystkich interesariuszy, dlatego też należy rozstrzygnąć wszelkie spory między nimi.

1.1.2. Czynności związane z monitorowaniem testów i nadzorem nad testami

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

W przypadku mniej złożonych projektów stosunkowo łatwo jest powiązać produkty prac testowych i czynności testowe z planem i celami strategicznymi, ale zazwyczaj wymaga to określenia bardziej szczegółowych celów, aby to osiągnąć. Może to obejmować zdefiniowanie środków i celów niezbędnych do osiągnięcia celów testów i pokrycia podstawy testów.

Szczególnie ważne jest powiązanie statusu produktów prac testowych i czynności związanych z testowaniem w sposób zrozumiały i istotny dla projektu i interesariuszy biznesowych.

Monitorowanie testów i nadzór nad testami to działania ciągłe.

Nadzór nad testami polega na porównaniu rzeczywistych postępów z planem testów i wdrażaniu działań korygujących w razie potrzeby. Nadzór kieruje testami tak, aby były one zgodne ze strategią i celami testów (patrz podrozdział 1.4) i w razie potrzeby powraca do czynności związanych z planowaniem testów. Nadzór wymaga szczegółowych informacji dotyczących planowania testów, aby umożliwić podjęcie odpowiednich działań. Nadzór obejmuje:

- wdrożenie planu testów i dyrektyw nadzoru,
- zarządzanie odchyleniami od planowanych testów,
- postępowanie w przypadku nowo zidentyfikowanych lub zmienionych ryzyk,

- 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, identyfikowanie odchyleń od planowanych testów, identyfikowanie i analizowanie nowych ryzyk wymagających testowania oraz monitorowanie zmian w zakresie zidentyfikowanych ryzyk.

1.1.3. Czynności związane z ukończeniem testów

Ukończenie testów następuje zazwyczaj w momencie osiągnięcia kamieni milowych projektu (np. wydanie, koniec iteracji lub ukończenie testów danego poziomu). W przypadku wszelkich nierozwiązanych defektów tworzone są żądania zmian lub pozycje w backlogu produktu (patrz sylabus Certyfikowany Tester Poziom Podstawowy). Po spełnieniu kryteriów wyjścia należy zarejestrować kluczowe wyniki, zarchiwizować je i przekazać odpowiednim interesariuszom. Ukończenie testów wymaga wykonania następujących zadań:

Utworzenie i zatwierdzenie sumarycznego raportu z testów. Zadanie to zapewnia, że wszystkie testy zostały wykonane, a wszystkie cele testów zostały osiągnięte. Zadanie to obejmuje zebranie z różnych testaliów odpowiednich informacji, takich jak plany testów, wyniki testów, raporty o postępie testów, wcześniejsze sumaryczne raporty z testów i raporty o defektach. Zebrane informacje są oceniane i podsumowywane w sumarycznym raporcie z testów. Raport ten jest zatwierdzany i przekazywany odpowiednim interesariuszom.

Archiwizacja testaliów. Zadanie to polega na zidentyfikowaniu testaliów, które mogą być przydatne w przyszłości lub które mogą zostać ponownie wykorzystane, np. przypadki testowe. Archiwizacja zapewnia, że są one dostępne i łatwe do zrozumienia w celu ponownego wykorzystania w przyszłości. Ponadto wyniki testów, dzienniki testów, raporty z testów i inne testalia powinny zostać tymczasowo zarchiwizowane w systemie zarządzania konfiguracją.

Przekazanie testaliów. Zadanie to polega na dostarczeniu wartościowych produktów pracy tym, którzy ich potrzebują. Na przykład znane defekty, które zostały odroczone lub zaakceptowane, powinny zostać przekazane osobom, które będą korzystać z testaliów lub wspierać ich użytkowanie.

Wykonanie niezbędnych zadań w celu wyczyszczenia środowiska testowego i przywrócenia go do wcześniej zdefiniowanego stanu. Zadanie to zapewnia gotowość środowiska testowego do następnego cyklu testów lub następnego projektu. Obejmuje ono usunięcie wszelkich danych testowych, narzędzi testowych, sterowników, zaślepek i skryptów testowych ze środowiska testowego. Obejmuje ono również przywrócenie środowiska testowego do stanu pierwotnego lub pożądanego.

Przeprowadzenie, zebranie, udokumentowanie wyciągniętych wniosków. Zadanie to jest wykonywane podczas retrospektyw, podczas których omawiane i dokumentowane są ważne wnioski wyciągnięte podczas procesu testowego. Może to obejmować ustalenia dotyczące całego cyklu wytwarzania oprogramowania. Wyciągnięte wnioski mogą zostać wykorzystane do doskonalenia procesu testowego, zgodnie z opisem w podrozdziale 1.5.

1.2. Kontekst testowania

Wstęp. Kontekst testowania obejmuje wszelkie warunki i ograniczenia, które mają wpływ na proces testowy, kształtując decyzje i strategię dotyczące planowania, projektowania

i wykonywania testów. Bardzo ważne jest, aby kierownicy testów rozumieli ten kontekst, w celu dostosowania testowania do konkretnych potrzeb i celów projektu. Kontekst ten może się różnić w zależności od rodzaju wytwarzanego produktu, dziedziny biznesowej, wymagań regulacyjnych i, co najważniejsze, stosowanego cyklu wytwarzania oprogramowania.

Kierownicy testów mają za zadanie stosować ustalone strategie testów i wybierać techniki testowania, a nie je opracowywać. Odgrywają oni kluczową rolę w formułowaniu planów testów dostosowanych do kontekstu projektu. Dzięki zrozumieniu i uwzględnieniu tych różnych czynników kierownicy testów mogą zapewnić, że testowanie jest adekwatne, skuteczne i wydajne w osiąganiu celów testów.

1.2.1. Interesariusze testów

Interesariusze testów to osoby lub grupy, które są bezpośrednio lub pośrednio zainteresowane jakością wytwarzanego produktu. Poniżej znajduje się typowa lista potencjalnych interesariuszy, sformułowana tak, aby odzwierciedlała ich różnorodne zainteresowania w zakresie testowania.

- **Programiści, liderzy rozwoju i kierownicy ds. rozwoju.** Oprócz wdrażania testowanego systemu i podejmowania działań w oparciu o wyniki testów, są również zaangażowani w testowanie modułowe i wnoszą wkład w proces testowy.
- **Testerzy, liderzy testów i kierownicy testów.** Przygotowują testalia, co obejmuje opracowywanie planów testów i udział w procesie testowym poprzez takie czynności jak analiza wymagań, projektowanie testów, wykonywanie testów, śledzenie i zgłaszanie defektów, automatyzacja testów oraz raportowanie postępów testów.
- **Kierownicy projektów, właściciele produktów i użytkownicy biznesowi.** Określają wymagania, definiują wymagany poziom jakości i zalecają wymagany zakres testów na podstawie postrzeganych ryzyk. Sprawdzają również produkty pracy, uczestniczą w testowaniu akceptacyjnym użytkowników (ang. *User Acceptance Testing*, UAT) i podejmują decyzje na podstawie wyników testów.
- **Zespół operacyjny.** Przeprowadza operacyjne testy akceptacyjne, zapewnia gotowość systemu do produkcji i przyczynia się do definiowania wymagań нефункциональных.
- **Klienci i użytkownicy.** Klienci kupują produkt, a użytkownicy bezpośrednio z niego korzystają. Obie grupy odgrywają kluczową rolę w definiowaniu wymagań i powinny być zaangażowane w testowanie akceptacyjne użytkowników (UAT) w celu sprawdzenia, czy produkt spełnia ich potrzeby.

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 przy identyfikowaniu konkretnych interesariuszy dla swojego projektu.

1.2.2. Znaczenie wiedzy interesariuszy w zarządzaniu testami

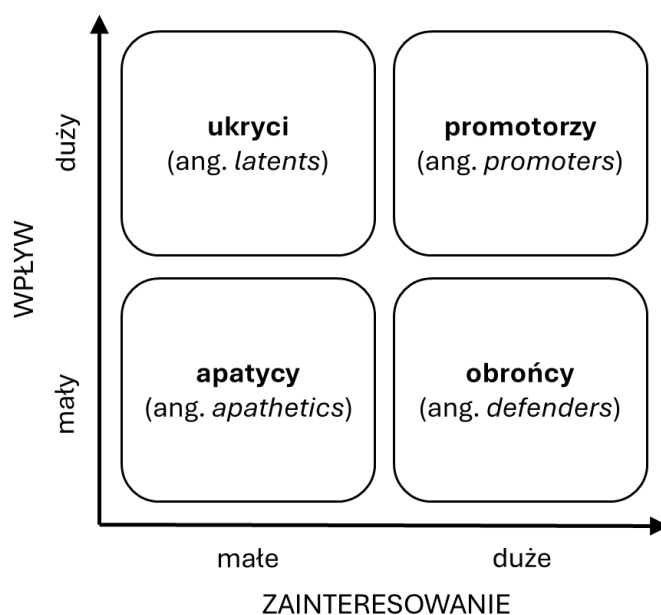
W zarządzaniu testami istotne jest uwzględnianie perspektyw oraz wpływu różnych interesariuszy. Macierz interesariuszy (patrz Rys. 1), często nazywana macierzą władza-vs.-zainteresowanie, wspiera kierowników testów w ustalaniu priorytetów zaangażowania interesariuszy oraz w skutecznym zarządzaniu ich oczekiwaniami. Macierz interesariuszy stanowi strategiczne narzędzie w zarządzaniu testami, które:

- wykorzystuje wiedzę interesariuszy, a użytkownicy końcowi oraz zespoły techniczne dostarczają informacji zwrotnych i spostrzeżeń dotyczących wydajności oraz zabezpieczeń,
- wspomaga zarządzanie ryzykiem poprzez uwidacznianie interesów i wpływu interesariuszy, zachęcając do podejmowania proaktywnych działań łagodzących,
- uwzględnia różnorodne punkty widzenia dzięki wartościowym informacjom zwrotnym.

Macierz interesariuszy obejmuje cztery ćwiartki:

- **Promotorzy (duży wpływ, duże zainteresowanie):** kluczowi interesariusze charakteryzujący się wysokim wpływem na projekt i dużym zainteresowaniem produktem końcowym, niezbędni do kształtowania strategii testów oraz planu testów.
- **Ukryci (duży wpływ, małe zainteresowanie):** mimo ograniczonego zaangażowania w bieżące zadania, ich decyzje wynikające z władzy w projekcie mają istotne znaczenie dla alokacji zasobów oraz ogólnego kierunku projektu.
- **Obrońcy (mały wpływ, duże zainteresowanie):** często dostarczają wartościowych informacji zwrotnych i mogą być angażowani poprzez regularne komunikaty oraz udział w dyskusjach.
- **Apatycy (mały wpływ, małe zainteresowanie):** chociaż ich zaangażowanie jest niewielkie, informowanie ich o kluczowych kamieniach milowych oraz zachęcanie do udziału w wybranych zagadnieniach projektowych może przynieść unikalne spostrzeżenia.

Rola kierownika testów obejmuje opracowanie szczegółowej listy interesariuszy oraz zrozumienie powiązań każdego z nich z czynnościami testowymi, z wykorzystaniem 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 łączą elementy tradycyjnego podejścia sekwencyjnego oraz praktyk zwinnych w celu dopasowania ich do specyficznych potrzeb projektu lub zmian zachodzących w organizacji. Poniżej przedstawiono typowe powody stosowania hybrydowego modelu wytwarzania oprogramowania, przy czym – w zależności od organizacji i projektu – mogą występować również inne przestanki.

- **Model hybrydowy jako ułatwienie przejścia na model zwinny.** Przechodzenie z metodyk tradycyjnych na zwinne może być wymagające ze względu na istotne zmiany w przepływie pracy, kulturze organizacyjnej oraz dynamice zespołu; modele hybrydowe oferują zrównoważone podejście, które ułatwia to przejście poprzez połączenie struktury metod tradycyjnych z elastycznością praktyk zwinnych.
- **Model hybrydowy jako rozwiązanie dopasowane do zamierzonego celu.** Niektóre organizacje lub projekty mogą nie mieć możliwości pełnego przejścia na metodyki zwinne; projekty o wysokim poziomie ryzyka mogą wymagać sekwencyjnej realizacji wybranych zadań oraz stosowania praktyk zwinnych w odniesieniu do innych, co uzasadnia wykorzystanie modelu hybrydowego dostosowanego do określonych celów.

W podejściu hybrydowym czynności związane z zarządzaniem testami mogą obejmować:

- ocenę poziomu zrozumienia oraz zdolności zespołu do sprawnego przechodzenia pomiędzy metodykami tradycyjnymi i zwinnymi,
- identyfikację mocnych i słabych stron w zakresie adaptacji do podejścia hybrydowego,
- zapewnienie, że zespół posiada kompetencje umożliwiające łączenie ustrukturyzowanych procesów z elastycznością podejścia zwinnego,
- wzmacnianie współpracy pomiędzy zespołem testowym a interesariuszami w celu skuteczniejszego zarządzania testami zarówno w ramach iteracji, jak i tradycyjnych faz testowych;
- uczestnictwo w skoordynowanych działaniach, takich jak *scrum-of-scrums* dla testerów, w celu utrzymania koncentracji na testowaniu przy jednoczesnym wspieraniu realizacji ogólnych celów wytwarzania,
- monitorowanie i przegląd działań oraz przypadków testowych realizowanych w iteracjach, aby zapewnić ich zgodność z praktykami zwinnymi.

Dodatkowe informacje można znaleźć w (Fowler, 2010).

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

Aby właściwie dostosować testowanie do modelu wytwarzania oprogramowania, kierownik testów musi zrozumieć różne modele wytwarzania oprogramowania stosowane w swojej organizacji i wykorzystać tę wiedzę do właściwego dostosowania testowania do czynności wytwórczych.

Tabela 1 przedstawia porównanie różnych czynności związanych z zarządzaniem testami w oparciu o różne modele wytwarzania oprogramowania.

Tabela 1. Działania związane z zarządzaniem testami w różnych modelach wytwarzania oprogramowania

Aspekt	Sekwencyjny model wytwarzania (np. model V)	Iteracyjny model wytwarzania (np. Scrum)
Szacowanie	Wczesne szczegółowe szacowanie dla każdego poziomu testów	Iteracyjne szacowanie, część planowania historyjek w ramach iteracji
Testalia	Obejmuje strategię, plan, przypadki testowe, harmonogram i raporty	Koncentruje się na kryteriach akceptacji i definicji ukończenia; lekka dokumentacja
Role	Kierownik testów nadzoruje decyzje i zarządzanie zespołem	Role są zintegrowane; facylitator lub coach zastępuje tradycyjnego kierownika testów
Narzędzia	Przeważają narzędzia do zarządzania testami dostosowane do testowania w określonej fazie	Narzędzia do CI/CD i automatyzacji mają kluczowe znaczenie, wspierając ciągłe testowanie
Podejście do testowania	Zaplanowane z wyprzedzeniem, odpowiadające fazom projektu	Wbudowane w iterację, z naciskiem na elastyczność i informacje zwrotne
Automatyzacja testów	Wdrożone strategicznie, może mieć miejsce na różnych etapach	Wbudowane od samego początku, z naciskiem na zautomatyzowane testy regresji w procesie CI/CD
Monitorowanie i raportowanie	Raportowanie oparte na kamieniach milowych, z opcjonalnymi zautomatyzowanymi tablicami wskaźników	Ciągłe raportowanie z tablicami wskaźników w czasie rzeczywistym i codziennymi aktualizacjami statusu
Metryki	Koncentracja na tradycyjnych metrykach testowych i zarządzaniu defektami (np. wykonywanie testów, wskaźniki defektów)	Oprócz tradycyjnych metryk obejmuje metryki zwinne do śledzenia iteracji (np. prędkość zespołu, wykresy spalania)

1.2.5. Czynności związane z zarządzaniem testami na różnych poziomach testów

Testowanie modułowe:

- określanie zakresu, celów oraz kryteriów ukończenia testów modułowych,
- angażowanie testerów w czynności wykraczające poza tradycyjne role testowe, takie jak przeglądy kodu, w których ich kompetencje analityczne są istotną wartością dodaną,
- koordynowanie współpracy z zespołem programistów w zakresie rozwiązywania problemów oraz wkładu w testy modułowe.

Testowanie integracji modułów:

- ustalanie kolejności integracji oraz kombinacji testów we współpracy z zespołem wytwórczym, z uwzględnieniem modelu wytwarzania oprogramowania, stosowanych narzędzi i procesów,
- nadzorowanie postępów w celu zapewnienia zgodności ze strategiami testowania systemowego i testowania akceptacyjnego,
- zarządzanie tą fazą we współpracy z programistami, z uwzględnieniem specyfiki testowania integracji modułów.

Testowanie systemowe:

- dostosowanie planowania do modelu wytwarzania oprogramowania, w tym staranna alokacja zasobów, dobór narzędzi oraz harmonogramowanie; w projektach zwinnych testowanie systemowe powinno być zintegrowane z iteracyjnym testowaniem historyjek użytkownika, z unikaniem odrębnych faz testowych oraz z zapewnieniem ciągłości i integracji testowania; w modelach sekwencyjnych testowanie może być realizowane zgodnie z zaplanowanymi etapami.

Testowanie integracji systemów:

- zapewnienie, że zakres oraz cele testowania integracji systemów są jednoznacznie określone i dostosowane do oceny ryzyka oraz celów jakościowych,
- sprawowanie nadzoru nad postępami, wynikami oraz zarządzaniem problemami w trakcie testowania integracji systemów.

Testowanie akceptacyjne:

- współpraca z interesariuszami w celu przeglądu i potwierdzenia spełnienia kryteriów akceptacji oraz planowania czynności testowych, w tym zarządzania testami akceptacyjnymi użytkownika,
- koordynowanie logistyki testowania akceptacyjnego oraz ułatwianie realizacji testów u klienta w celu potwierdzenia, że produkt spełnia potrzeby biznesowe i standardy jakości poza środowiskiem wytwórczym,
- wspieranie rozwiązywania problemów związanych z testowaniem akceptacyjnym użytkownika oraz 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 specyficzne wymagania testowania funkcjonalnego, niefunkcjonalnego, czarnoskrzynkowego i białoskrzynkowego. W przypadku kierowników testów zarządzających testowaniem funkcjonalnym nacisk kładzie się na zapewnienie, że wszystkie funkcje są dokładnie przetestowane i spełniają określone wymagania. Zarządzanie testami niefunkcjonalnymi koncentruje się na weryfikacji niefunkcjonalnych charakterystyk jakościowych systemu, takich jak wydajność czy zabezpieczenia. Zarządzanie testowaniem czarnoskrzynkowym polega na zapewnieniu, że testy są skoncentrowane na użytkowniku i obejmują wszystkie możliwe interakcje zewnętrzne. Zarządzanie testowaniem białoskrzynkowym kładzie nacisk na zrozumienie struktury kodu i zapewnienie, że testy dokładnie obejmują logikę wewnętrzną przedmiotu testów.

Zarządzanie testowaniem funkcjonalnym

- Planowanie strategiczne i śledzenie postępów: opracowanie szczegółowej strategii testów zgodnej z wymaganiami funkcjonalnymi i celami projektu, a także monitorowanie postępów.
- Koordynacja zasobów: efektywne przydzielanie zasobów ludzkich i technicznych w celu uwzględnienia wszystkich funkcjonalnych aspektów systemu.

Zarządzanie testowaniem нефункциональным

- Benchmarking wydajności: ustalenie benchmarków wydajności i zarządzanie czynnościami testowymi, które oceniają system pod kątem tych kryteriów.
- Weryfikacja zgodności: nadzorowanie testów zapewniających zgodność systemu z wymaganiami нефункциональными, takimi jak zabezpieczenia, użyteczność i niezawodność.

Zarządzanie testowaniem czarnoskrzynkowym

- Analiza pokrycia testowego: zapewnienie, że testy czarnoskrzynkowe obejmują wszystkie scenariusze użytkowników i wymagania biznesowe.
- Włączanie informacji zwrotnych: zarządzanie procesem zbierania informacji zwrotnych od interesariuszy w celu udoskonalenia podejścia do testów czarnoskrzynkowych i usuwania defektów.

Zarządzanie testowaniem białoskrzynkowym

- Optymalizacja pokrycia kodu: nadzorowanie wykorzystania narzędzi do pokrycia kodu w celu identyfikacji luk w testowaniu białoskrzynkowym i kierowanie zasobów do tych obszarów.
- Integracja wiedzy technicznej: zarządzanie włączaniem wiedzy technicznej do procesu planowania testów, zapewniając, że testy są projektowane z uwzględnieniem wewnętrznego działania aplikacji.

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

Skuteczne zarządzanie testami jest podstawą każdego udanego projektu testowego i obejmuje szeroki zakres czynności, które wymagają starannego planowania, czujnego monitorowania i strategicznej kontroli. Kierownicy testów odgrywają kluczową rolę w zapewnieniu, że proces testowy jest nie tylko skuteczny i wydajny, ale także dostosowany do unikalnych wymagań danego projektu.

Planowanie testów:

- **Kompleksowe określenie zakresu.** Plan testów musi być starannie opracowany i zawierać dokładne określenie zakresu testów. Obejmuje to identyfikację wszystkich wymagań funkcjonalnych i нефункциональных w celu zapewnienia pełnego pokrycia. Obejmuje to również rozważenie konsekwencji stosowania testowania czarno- i białoskrzynkowego, zapewniając, że opracowane przypadki testowe są w stanie zweryfikować testowany system pod każdym kątem.
- **Ocena ryzyka i plan łagodzenia ryzyka.** Integralną częścią planu testów są solidne ramy zarządzania ryzykiem. Kierownicy testów muszą przeprowadzać szczegółową analizę ryzyka, wskazując potencjalne słabe punkty i wyzwania, które mogą mieć wpływ zarówno na przebieg projektu, jak i na produkt końcowy. Kluczowe znaczenie ma opracowanie strategii łagodzenia ryzyka, obejmującej planowanie działań prewencyjnych w celu skutecznego uniknięcia lub zminimalizowania ryzyk.
- **Strategia planowania zasobów.** Kolejnym kluczowym elementem jest planowanie zasobów. Wykracza ono poza samą alokację i obejmuje definiowanie struktury zespołu,

wyznaczanie ról i ustanawianie protokołów komunikacyjnych. W środowiskach, w których zespoły są rozproszone, jak w modelach na miejscu/poza siedzibą firmy (ang. *onsite/offsite*), ma to szczególne znaczenie dla synchronizacji działań i zapewnienia płynnej współpracy.

Monitorowanie testów:

- **Nadzór nad wykonywaniem testów.** Monitorowanie testów odgrywa kluczową rolę w procesie zarządzania testami. Obejmuje ono ciągłą weryfikację wykonania testów w stosunku do ustalonego planu, śledzenie postępów w testowaniu i zarządzanie wszelkimi wykrytymi defektami. Priorytetyzacja testów w oparciu o ocenę ryzyka i rozwój sytuacji w czasie rzeczywistym zapewnia, że testy pozostają skoncentrowane i dostosowane do najbardziej krytycznych obszarów.
- **Optymalizacja narzędzi i środowiska.** Rozsądny wybór i wykorzystanie narzędzi testowych oraz środowisk ma kluczowe znaczenie dla wsparcia strategii testów. Ciągłe monitorowanie zapewnia ich skuteczną integrację z procesem ciągłej integracji i ciągłego dostarczania (CI/CD), ułatwiając ciągłe testowanie i natychmiastowe pętle informacji zwrotnej, które są niezbędne w zwinnym wytwarzaniu oprogramowania.
- **Współpraca w zakresie rozwoju.** Utrzymanie bliskich relacji z zespołem wytwórczym ma zasadnicze znaczenie dla pomyślnych wyników testów. Współpraca ta powinna wspierać kompleksowe podejście do testów, wykorzystując spostrzeżenia zarówno z perspektywy białoskrzynkowej, jak i czarnoskrzynkowej, aby prewencyjnie rozwiązywać potencjalne problemy.

Nadzór nad testami:

- **Adaptacyjne zarządzanie procesami.** Nadzór nad testami polega na dynamicznym dostosowywaniu procesu testowania w odpowiedzi na nowe spostrzeżenia, wyzwania i zmieniającą się dynamikę sytuacji projektowej. Wymaga to od kierownika testów elastyczności i zdolności do reagowania oraz wprowadzania zmian w podejściu do testów, które odzwierciedlają aktualny stan projektu.
- **Zarządzanie bramkami jakości.** Ustrukturyzowane podejście do zarządzania bramkami jakości ma fundamentalne znaczenie dla jakości produktu. Obejmuje ono zdefiniowanie, co stanowi bramki jakości w procesie testowym, oraz podejmowanie świadomych decyzji dotyczących postępów w testowaniu, co ma kluczowe znaczenie dla utrzymania integralności produktu.

Koncentrując się na powyższych czynnościach w ramach planowania testów, monitorowania testów i nadzoru nad testami, kierownicy testów mogą zapewnić, że proces testowy jest dobrze zdefiniowany, dostosowany do zmian w projekcie i skutkuje produktem, który spełnia 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. Ryzyko jest identyfikowane przez różnych interesariuszy i może być wykorzystane do wyboru i priorytetyzacji testów. Im wyższy poziom ryzyka, tym wcześniej należy rozpocząć testy i tym intensywniejsze i dłuższe powinny one być.

1.3.1. Testowanie jako czynność łagodząca ryzyko

Ryzyko produktowe dotyczy sytuacji, w których mogą wystąpić problemy z jakością produktu. Gdy testy ujawniają defekty, testowanie pomaga złagodzić ryzyko produktowe, zapewniając świadomość istnienia defektów i możliwość ich usunięcia przed wydaniem produktu. Gdy testy nie wykrywają defektów, oznacza to, że poziom ryzyka produktowego jest niższy niż oczekiwano.

Kierownik testów jest między innymi odpowiedzialny za dostarczenie prawidłowej i wiarygodnej oceny jakości produktu. Wymaga to aktywnego zaangażowania w zarządzanie ryzykiem projektowym, ze szczególnym uwzględnieniem ryzyk jakościowych (np. niejasne wymagania, które prowadzą do poważnych problemów w późniejszej walidacji czy niewystarczające środowiska testowe utrudniające przeprowadzenie testów).

Testowanie oparte na ryzyku koncentruje się na ryzykach jakościowych. Jest ono zgodne z ogólnym procesem zarządzania ryzykiem, który składa się z dwóch głównych czynności:

- analiza ryzyka, obejmująca identyfikację ryzyka i ocenę ryzyka,
- kontrola ryzyka, obejmująca monitorowanie ryzyka i łagodzenie ryzyka.

Czynności te są logicznie uporządkowane, ale w rzeczywistości mogą się nakładać.

Aby skoncentrować testowanie na ryzykach jakościowych, należy zidentyfikować i ocenić te ryzyka. Aby analiza ryzyka była jak najbardziej skuteczna, powinna obejmować różnych interesariuszy. Jako główny interesariusz analizy ryzyka jakościowego, kierownik testów powinien rozumieć i monitorować te czynności oraz być w stanie je moderować.

Monitorowanie testów powinno obejmować monitorowanie ryzyka. Oprócz monitorowania ewolucji znanych ryzyk jakościowych, powinno ono obejmować analizę wszelkich nowych ryzyk jakościowych oraz aktualizację rejestru ryzyk.

Kierownik testów jest jedną z kilku osób, które kierują łagodzeniem ryzyk jakościowych. Łagodzenie ryzyka jest rozłożone na kilka czynności testowych. Na przykład wyniki analizy ryzyka jakościowego są wykorzystywane w planowaniu testów, aby skoncentrować testy na właściwych obszarach przy użyciu właściwych technik. W analizie testów poziomy ryzyka kierują wyborem warunków testowych, które mają zostać pokryte testami. Podczas wykonywania testów priorytetyzacja oparta na ryzyku reguluje kolejność wykonywania testów.

1.3.2. Identyfikacja ryzyk jakościowych

Zadaniem kierownika testów jest zebranie informacji o ryzykach od interesariuszy. Interesariusze mogą identyfikować ryzyka jakościowe za pomocą jednej lub kilku z następujących technik:

- wywiady z ekspertami,
- niezależne oceny,
- retrospektywy,
- warsztaty dotyczące ryzyka,
- burze mózgów,
- listy kontrolne,
- odwołanie się do wcześniejszych doświadczeń.

Dzięki zaangażowaniu jak najszerszej grupy interesariuszy proces identyfikacji ryzyka często pozwala zidentyfikować większość istotnych ryzyk produktowych. Bardzo ważne jest określenie, którzy interesariusze powinni uczestniczyć w tym etapie. Konieczne jest zapewnienie, aby lista zaangażowanych interesariuszy była wyczerpująca i uzgodniona z kierownikiem projektu. Identyfikacja ryzyka, która pomija kluczowych interesariuszy, może być bardzo problematyczna. Kluczowe znaczenie ma zapewnienie wszystkim istotnym interesariuszom możliwości udziału w projekcie. Jeśli nie mogą oni uczestniczyć w tym zadaniu, powinni przynajmniej mieć możliwość delegowania udziału w nim. W przypadku braku reprezentacji kluczowych interesariuszy można zorganizować spotkanie inauguracyjne, aby ustalić, czy kogoś nie brakuje.

W testowaniu opartym na ryzyku ważne jest zrozumienie, że ryzyko nie jest równomiernie rozłożone w obrębie przedmiotów testów. Na przykład moduły aplikacji samoobsługowej przeznaczone dla klientów mogą wiązać się z zupełnie innym ryzykiem związanym z użytecznością niż moduły administracyjne. Identyfikacja indywidualnych rodzajów ryzyka dla różnych elementów testowych jest ważnym zadaniem w planowaniu testów.

Identyfikacja ryzyka często powoduje powstanie produktów ubocznych (tj. identyfikację problemów, które nie są ryzykami produktowymi). Przykłady obejmują ogólne pytania lub problemy dotyczące produktu lub projektu lub problemy w dokumentach referencyjnych, takich jak wymagania i specyfikacje projektowe. Ryzyka projektowe są również często identyfikowane jako produkt uboczny identyfikacji ryzyka jakościowego, ale nie jest to przedmiotem testowania opartego na ryzyku. Kierownik testów może często odgrywać znaczącą rolę w podkreślaniu tych produktów ubocznych i jasno pokazywać, że jakość jest obowiązkiem wszystkich. Słabe lub brakujące wymagania są często oznaką bardziej fundamentalnego problemu w planowaniu i przygotowaniu, ponieważ zapewnienie jakości jest obecne w całym cyklu wytwarzania oprogramowania.

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 dla każdego ze zidentyfikowanych ryzyk obejmuje zazwyczaj ocenę prawdopodobieństwa (wystąpienia) ryzyka i wpływ tego ryzyka. Czynniki wpływające na prawdopodobieństwo ryzyk jakościowych obejmują:

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

Czynniki wpływające na wpływ ryzyka obejmują:

- częstotliwość korzystania z danej funkcji,
- krytyczność dotkniętej ryzykiem funkcji,
- krytyczność dotkniętego ryzykiem celu biznesowego,
- utrata reputacji,
- utrata dochodów z działalności,
- potencjalne straty lub odpowiedzialność finansowa, ekologiczna lub społeczna,
- sankcje cywilne lub karne,
- problemy z interfejsami i integracją,
- brak rozsądnych rozwiązań zastępczych,
- potrzeby bezpieczeństwa.

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

Jeśli analiza ryzyka opiera się na obszernych i statystycznie istotnych 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 wartości. Zazwyczaj jednak prawdopodobieństwo ryzyka i wpływ ryzyka można ustalić tylko jakościowo w skalach porządkowych, np. jako bardzo wysokie, wysokie, średnie, niskie lub bardzo niskie. Prawdopodobieństwo ryzyka i wpływ ryzyka są następnie łączone za pomocą macierzy ryzyka w celu określenia poziomu ryzyka. Ten poziom ryzyka należy interpretować jako jakościową, względną ocenę na skali porządkowej.

Jeśli analiza ryzyka nie opiera się na obszernych i statystycznie istotnych danych dotyczących 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 tworzeniu oprogramowania testowanie jest najważniejszą czynnością łagodzącą ryzyko jakościowe. Umożliwia ono zmniejszenie prawdopodobieństwa wystąpienia awarii. Inne możliwe środki łagodzące ryzyko obejmują plany awaryjne (np. poprzez zapewnienie rozwiązań zastępczych), przeniesienie ryzyka na stronę trzecią (np. dostawcę modułu) lub akceptację ryzyka.

Podczas **planowania testów** czas i wysiłek związany z opracowaniem i wykonaniem testu powinny być proporcjonalne do poziomu związanego z nim ryzyka: testy dla wyższych poziomów ryzyka powinny rozpocząć się wcześniej i wykorzystywać bardziej rygorystyczne techniki testowania, natomiast testy dla niższych poziomów ryzyka mogą rozpocząć się później i mogą wykorzystywać mniej rygorystyczne techniki testowania. Aby jak najlepiej złagodzić całkowite ryzyko poprzez testowanie, kierownik testów powinien wybrać odpowiednie podejście do testów na podstawie analizy następujących czynników kontekstowych:

- **Elementy testowe:** różne elementy testowe w obrębie przedmiotu testów mogą mieć różne poziomy ryzyka tego samego typu, więc przedmiot testów nie musi być testowany z jednakową dokładnością.

- **Charakterystyki jakościowe:** Ryzyko wpływające na określone charakterystyki jakościowe powinno być ograniczane za pomocą powiązanych typów testów, które wymagają określonego wysiłku, środowisk testowych i umiejętności testowych.
- **Poziomy testów i typy testów:** Niektóre rodzaje ryzyk można testować wyłącznie dynamicznie na określonych poziomach testów, inne poprzez testowanie statyczne (np. analizę statyczną i przegląd kodu pod kątem utrzymywalności) lub poprzez połączenie obu metod (np. przegląd architektury i dynamiczne testowanie zintegrowanego systemu pod kątem luk w zabezpieczeniach). Przeprowadzenie testów wszystkich elementów testowych jak najwcześniej łagodzi ryzyko wykrycia krytycznych defektów w późniejszych fazach cyklu życia, co powodowałoby wyższe koszty awarii wewnętrznej oraz opóźnienia.
- **Model wytwarzania oprogramowania:** Czynności testowe mają swoje własne, specyficzne kryteria wejścia. W różnych modelach wytwarzania oprogramowania sprawdza się ich spełnienie 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) wymagają stosowania określonych technik testowania i spełnienie określonego pokrycia w oparciu o poziom nienaruszalności bezpieczeństwa (ang. *SIL – Safety Integrity Level*). Kierownik testów musi zapewnić przestrzeganie tych norm.

Ponadto poziom ryzyka powinien mieć wpływ na decyzje dotyczące kontroli jakości, takie jak stosowanie przeglądów produktów pracy (np. przypadków testowych), poziom niezależności testowania oraz zakres przeprowadzanych testów regresji.

Podczas **monitorowania i nadzoru nad testami** testowanie oparte na ryzyku umożliwia raportowanie postępów testów pod kątem poziomu ryzyka rezydualnego (resztkowego) w dowolnym momencie. Pomaga to zespołowi twórcy i interesariuszom w monitorowaniu i nadzorowaniu rozwoju oprogramowania, w tym w podejmowaniu decyzji dotyczących wydania, w oparciu o poziom ryzyka resztkowego. Wymaga to raportowania wyników testów pod kątem ryzyka w sposób zrozumiały dla interesariuszy.

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

- W niektórych przypadkach testy są priorytetyzowane do wykonania w ścisłej kolejności według malejących poziomów ryzyka, które pokrywają. Podejście to nazywa się podejściem „w głąb” (ang. *depth-first*) i jest odpowiednie, gdy ważne jest jak najwcześniejsze łagodzenie ryzyk najwyższego poziomu.
- Alternatywnie, co najmniej jeden test dla każdego ryzyka otrzymuje najwyższy priorytet. Wszystkie pozostałe testy są priorytetyzowane na podstawie poziomu ryzyka, które pokrywają. Podejście to nazywane jest podejściem „wszerz” (ang. *breadth-first*) i jest odpowiednie, gdy interesariusze chcą jak najwcześniej uzyskać ogólny obraz jakości całego produktu. W praktyce testowanie często rozpoczyna się od podejścia „w głąb”, ale gdy czas staje się coraz bardziej ograniczony, przechodzi się do podejścia „wszerz”, testując wszystkie pozostałe elementy ryzyka co najmniej raz.

Niezależnie od przyjętego podejścia, czas przeznaczony na testowanie może zostać wykorzystany bez przeprowadzenia wszystkich zaplanowanych testów. W tym momencie testowanie oparte na ryzyku ułatwia przedstawienie kierownictwu uzasadnionej rekomendacji dotyczącej przedłużenia testowania lub zaakceptowania pozostałego ryzyka.

1.3.5. Techniki testowania oparte na ryzyku

Istnieje wiele technik o różnym stopniu sformalizowania, które służą do wdrażania testowania opartego na ryzyku. Przydatność danej techniki zależy od projektu, procesu i produktu. Istnieją dwa podstawowe rodzaje technik: ciężkie i lekkie. W systemach o krytycznym znaczeniu ze względów bezpieczeństwa bardzo często stosuje się techniki ciężkie. W aplikacjach, które nie mają krytycznego znaczenia ze względów bezpieczeństwa, zazwyczaj stosuje się techniki lekkie.

Techniki ciężkie są formalne, wykorzystują określone procedury i szczegółową dokumentację. Angażują szerokie grupy interesariuszy. Ocena ryzyka w ramach technik ciężkich wykorzystuje szczegółowe czynniki ryzyka, prawdopodobieństwo ryzyka i wpływ ryzyka oraz wzory matematyczne do obliczenia prawdopodobieństwa i wpływu na podstawie tych czynników. Przykłady technik ciężkich (Roman, 2018):

- **Analiza zagrożeń** (ang. *hazard analysis*) rozszerza proces analityczny na wcześniejsze etapy, próbując zidentyfikować zagrożenia leżące u podstaw każdego ryzyka.
- **Koszt ekspozycji** (ang. *cost of exposure*) określa dla każdego elementu ryzyka jakościowego prawdopodobieństwo awarii, koszt straty związanej z typową awarią oraz koszt testowania takich awarii.
- **Analiza przyczyn i skutków awarii** (ang. *Failure Mode and Effect Analysis, FMEA*) oraz jej warianty identyfikują ryzyka jakościowe, ich potencjalne przyczyny i prawdopodobne skutki, a następnie przypisują im krytyczność, priorytet i wskaźnik wykrywalności.
- **Analiza drzewa usterek** (ang. *fault tree analysis*) łączy potencjalne awarie z defektami, które mogą ją spowodować, a następnie z błędami, które mogą spowodować te defekty, kontynuując to postępowanie aż do zidentyfikowania podstawowych przyczyn.

Techniki lekkie są mniej dokładne i wymagają mniejszego wysiłku ze strony zespołu testowego i interesariuszy (Roman, 2018). Podobnie jak techniki ciężkie, opierają się one również na zaangażowaniu interesariuszy, wykorzystując wyniki analizy ryzyka jako podstawę do planowania testów i definiowania warunków testowych. Jednak grupa interesariuszy może nie być tak szeroka, a czynniki ryzyka są zazwyczaj sprowadzane 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 dostarczone są specyfikacje wymagań. Inne techniki, w tym pragmatyczna analiza i zarządzanie ryzykiem (ang. *Pragmatic Risk Analysis and Management, PRAM*) (Black, 2009) oraz podejście do zarządzania 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 całkowicie w oparciu o dane wejściowe od interesariuszy.

1.3.6. Metryki sukcesu i trudności związane z testowaniem opartym na ryzyku

Podczas retrospektyw zespół testujący powinien zmierzyć, w jakim stopniu zrealizował korzyści płynące z testowania opartego na ryzyku. W wielu przypadkach wymaga to udzielenia odpowiedzi na niektóre lub wszystkie z poniższych pytań przy użyciu metryk i konsultacji.

- Czy odpowiedni interesariusze byli zaangażowani lub reprezentowani w trakcie analizy ryzyka?
- Czy zaangażowanie interesariuszy w analizę ryzyka było odpowiednie?
- Jeśli w produkcji wystąpiły krytyczne problemy wskazujące na przeoczenie krytycznych defektów, czy zostały one rozwiązane?
- Czy większość defektów o wysokim priorytecie została wykryta na wczesnym etapie wykonywania testów?
- Czy zespół testowy był w stanie wyjaśnić wyniki testów interesariuszom w kategoriach ryzyka?
- Czy pominięte testy wiązały się z niższym poziomem ryzyka niż testy wykonane?

W większości przypadków pomyślnie przeprowadzone testowanie oparte na ryzyku skutkuje twierdzącą odpowiedzią na wszystkie te pytania. W perspektywie długoterminowej należy wyznaczyć cele doskonalenia procesów w zakresie metryk sukcesu, a także dążyć do poprawy efektywności procesu analizy ryzyka jakościowego.

Zarządzanie ryzykiem często napotyka nieoczekiwane trudności ze względu na złożoność, która jest często pomijana.

- **Trudności w ocenie poziomu ryzyka.** Oszacowanie prawdopodobieństwa ryzyka i wpływu ryzyka może być bardzo trudne. Rozwiązanie: wykorzystanie danych historycznych i poproszenie kluczowych interesariuszy projektu o ich ocenę.
- **Entuzjastyczne początki.** Tworzenie i utrzymywanie odpowiedniego podejścia do testów opartego na ryzyku jest często zaniebywane w obliczu dużej presji na osiągnięcie sukcesu w krótkim czasie. Rozwiązanie: regularne monitorowanie ryzyka i raportowanie go interesariuszom.
- **Déjà vu.** W każdym projekcie pojawia się ten sam zestaw ryzyk, co prowadzi do samozadowolenia w stosunku do ryzyka. Rozwiązanie: zaangażowanie odpowiednich osób w identyfikację ryzyka i łagodzenie tylko tych ryzyk, które są uznane za istotne.
- **Pomijanie kluczowych ryzyk.** Podstawową przyczyną tego problemu jest zazwyczaj zaangażowanie w proces osób niedoświadczonych lub nieodpowiednich. Rozwiązanie: zaangażowanie odpowiednich osób i przeszkolenie ich.
- **Fluktuacja interesariuszy.** Interesariusze mogą się zmieniać w czasie, a także mogą pojawić się nowe ryzyka. Rozwiązanie: przeprowadzać analizę ryzyka jako działanie ciągłe i powtarzalne, a nie tylko raz na początku projektu.

1.4. Projektowa strategia testowa

Wstęp. W niniejszym 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”), sylabusa

ISTQB® Certyfikowany Tester Poziom Ekspercki Zarządzanie Testami oraz sylabusu ISTQB® ATLaS (*Certified Tester Agile Test Leadership at Scale*).

Jeśli organizacyjna strategia testowa nie istnieje lub nie obejmuje wymaganych aspektów, w procesie zarządzania testami należy dążyć do wyjaśnienia brakujących szczegółów z odpowiednimi interesariuszami.

W kontekście tego podrozdziału definicja projektowej strategii testowej jest przykładem szczegółowej strategii testów w odniesieniu do projektu, wydania, produktu lub dowolnego innego rodzaju inicjatywy wytwarzania lub pozyskania systemu. Projektowa strategia testowa (określana jako „strategia testów” w normie ISO/IEC/IEEE 29119-3) opisuje podejście do testów 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). Strategia testów jest również często wymagana przez umowy, porozumienia, organy regulacyjne lub przepisy prawa.

1.4.1. Wybór podejścia do testów

Projektowa strategia testowa ukierunkowuje wszystkie czynności testowe realizowane w projekcie oraz określa cele, zasoby, harmonogramy i odpowiedzialności. Strategia ta musi być dostosowana do specyficznych wymagań danego projektu. Kluczowe decyzje obejmują dobór poziomów testów, typów testów i technik testowania stosowanych w testowaniu statycznym i dynamicznym, a także innych praktyk testowych (np. testowanie skryptowe, testowanie manualne, testowanie *back-to-back*).

Z teoretycznego punktu widzenia wszystkie typy testów mogą być realizowane na dowolnym poziomie testów, a każda technika testowania może zostać zastosowana do każdego typu testów na każdym poziomie. W praktyce jednak właściwy dobór oraz odpowiednie połączenie tych elementów mają istotny wpływ na skuteczność i efektywność testowania. Przykładowo, łatwość utrzymania kodu może być często oceniana w sposób bardziej efektywny i wydajny poprzez analizę statyczną kodu lub przeglądy kodu; wydajność bywa lepiej oceniana przy użyciu zautomatyzowanych testów systemowych, ze względu na interakcje pomiędzy wewnętrznymi modułami, natomiast adekwatność funkcjonalna może być skuteczniej walidowana z użytkownikami poprzez wspólnie opracowywane manualne testy akceptacyjne. Wybór odpowiedniego podejścia do strategii testów może być procesem złożonym i zależy od wielu czynników, takich jak organizacyjna strategia testowa, kontekst projektu oraz inne uwarunkowania.

Dobór oraz połączenie poziomów testów, typów testów i technik testowania mają zatem kluczowe znaczenie dla skutecznej strategii testów w projekcie, ponieważ w istotny sposób wpływają na wydajność i efektywność realizowanych testów.

1.4.2. Analiza organizacyjnej strategii testowej, kontekstu projektu i innych aspektów

Aby umożliwić opracowanie projektowej strategii testowej, konieczne jest pełne zrozumienie organizacyjnej strategii testowej, kontekstu projektu oraz dodatkowych czynników i ograniczeń związanych z testowaniem.

W celu wyboru odpowiedniego podejścia do testów zazwyczaj analizuje się następujące czynniki:

- **Dziedzina biznesowa**, dla której produkt jest tworzony lub modyfikowany. Regulacje, normy oraz praktyki charakterystyczne dla danej dziedziny biznesowej mogą wpływać na rygor testowania, wymagany zakres dokumentacji oraz poziom jej szczegółowości. Przykładowo w sektorze farmaceutycznym i medycznym podejście do testów często koncentruje się na intensywnym testowaniu akceptacyjnym przez użytkownika, ukierunkowanym na zagrożenia dla zdrowia pacjentów, z wykorzystaniem przypadków testowych opartych na funkcjonalnych wymaganiach użytkowników. Natomiast testowanie akceptacyjne przez użytkownika internetowych aplikacji ubezpieczeniowych może skupiać się na użyteczności oraz na zwiększaniu prawdopodobieństwa zawierania nowych umów poprzez testowanie A/B.
- **Cele organizacyjne i nadrzędne charakterystyki jakościowe** mogą obejmować potrzebę wykazania wartości testowania, zwiększenia poziomu automatyzacji testów lub doskonalenia charakterystyk jakościowych procesu testowego, takich jak dojrzałość testów czy skuteczność wykrywania defektów. Czynniki te mogą determinować poziomy testów i typy testów, które powinny być realizowane.
- **Cele projektu i rodzaj projektu** (np. cele związane z budżetem, harmonogramem i jakością; projekt dla klienta indywidualnego w porównaniu z produktem masowym) zwykle wiążą się z określonymi ograniczeniami, ryzykami i możliwościami wpływającymi na testowanie. Przykładowo rygorystyczne ograniczenia budżetowe i czasowe mogą wymagać konsekwentnego stosowania testowania opartego na ryzyku w celu nadania priorytetów przypadkom testowym, natomiast rozwój produktu przeznaczonego dla klienta może wymagać testów uwzględniających wcześniej zdefiniowane umowne kryteria akceptacji.
- **Zasoby testowe** posiadają ograniczenia dostępności, które należy uwzględnić; przykłady obejmują dostępność narzędzi testowych, infrastruktury testowej, technologii oraz środowiska wytwórczego wykorzystywanego w projekcie, a także personelu testowego i jego kompetencji (patrz podrozdział 3.1). Przykładowo testowanie oparte na doświadczeniu wymaga testerów posiadających dobrą znajomość dziedziny biznesowej; aplikacje mobilne są zazwyczaj testowane na ograniczonej liczbie urządzeń; wykorzystanie narzędzi testowych może być ograniczone liczbą dostępnych licencji.
- **Model wytwarzania oprogramowania zastosowany w projekcie** wpływa na określenie odpowiednich poziomów testów, wysiłku testowego oraz kryteriów wejścia i wyjścia (patrz sylabus Certyfikowany Tester Poziom Podstawowy v4.0, podrozdziały 2.2 i 5.1). Cykl życia oprogramowania oparty na ciągłej integracji wymaga wyższego poziomu automatyzacji testów niż jednorazowe wytwarzanie w modelu kaskadowym, co skutkuje zastosowaniem odmiennych typów testów i technik testowania.
- **Interfejsy z innymi systemami**. W systemie systemów (ang. *system of systems*) kluczowe jest dostosowanie testów do innych zespołów lub projektów oraz wybór właściwych

poziomów testów, w szczególności testowania integracji systemów. Testowanie oparte na ryzyku może wspierać priorytetyzację i skalowanie testów integracji systemów.

- **Dostępność danych testowych.** Należy uwzględnić ograniczenia w dostępności danych testowych, takie jak konieczność stosowania zanonimizowanych danych produkcyjnych lub tworzenia specyficznych danych testowych, które mogą być trudne do uzyskania i wymagają walidacji, np. danych do testowania AI. Przykładowo testowanie oparte na modelu może wspierać generowanie oraz zarządzanie danymi testowymi.

Kierownik testów powinien określić, jaka kombinacja technik testowania, poziomów testów oraz typów testów stanowi najlepsze podejście do realizacji organizacyjnej strategii testowej, z uwzględnieniem kontekstu projektu oraz dodatkowych czynników i ograniczeń związanych z testowaniem.

1.4.3. Definicja celów testów

Plan testów powinien być definiowany dla każdego projektu testowego i obejmować między innymi zakres testów, cele testów oraz kryteria wyjścia. Plan testów może być przygotowany na poziomie wydania jako plan testów projektu (nazywany również głównym planem testów), a w razie potrzeby także jako plany dla poszczególnych poziomów testów. Dodatkowo możliwe jest opracowanie planów testów ukierunkowanych na konkretne charakterystyki jakościowe, np. plan testów zabezpieczeń czy plan testów wydajności.

W projektach realizowanych z wykorzystaniem podejścia zwinnego lub hybrydowego modelu wytwarzania oprogramowania można uzgadniać iteracyjne plany testów. Dla każdego wydania oraz iteracji zakres cech funkcjonalnych oraz charakterystyk niefunkcjonalnych, które mają zostać dostarczone, jest definiowany w planie testów i uzgadniany z interesariuszami.

W odniesieniu do funkcjonalności przekazywanych do testowania w projekcie należy określić cele testów oraz kryteria wyjścia. Do ich definiowania można zastosować metodę SMART:

- S (*specific* – konkretny): cel testu oraz kryterium wyjścia powinny być jednoznaczne i jasno sformułowane.
- M (*measurable* – mierzalny): cel powinien być możliwy do zmierzenia oraz posiadać określone kryteria oceny postępu, umożliwiające stwierdzenie jego osiągnięcia.
- A (*achievable* – osiągalny): cel powinien być wykonalny przy uwzględnieniu dostępnych zasobów, harmonogramu oraz możliwości zespołu.
- R (*relevant* – istotny): cel powinien pozostawać w zgodzie z nadrzędnymi celami projektu.
- T (*timely* – terminowy): cel powinien posiadać jasno określone ramy czasowe oraz termin realizacji.

Cele testów projektu powinny obejmować wszystkie istotne aspekty jakościowe i ilościowe, o ile są one mierzalne lub możliwe do oceny. Przykładowe cele testów projektu obejmują:

- osiągnięcie zdefiniowanych kryteriów wyjścia w określonym czasie,
- realizację organizacyjnych celów jakościowych (np. mierzonych kluczowymi wskaźnikami wydajności, takimi jak liczba reklamacji klientów dotyczących produktu),
- spełnienie obowiązujących regulacji i przepisów branżowych,
- zapewnienie dostępu do danych wyłącznie dla upoważnionych użytkowników (np. poprzez odpowiednie mechanizmy kontroli dostępu),

- weryfikację kompletności i poprawności funkcjonalnej, wydajności, skuteczności, przenaszalności oraz zabezpieczeń migracji danych,
- zwiększenie poziomu automatyzacji testów (np. testów regresji lub testów wydajności o określony procent),
- potwierdzenie, że refaktoryzacja kodu zakończyła się sukcesem i nie wprowadziła nowych defektów (np. poprzez testy regresji, przy jednoczesnym usunięciu długu technicznego i zachowaniu istniejącej funkcjonalności),
- wykazanie zabezpieczenia interfejsów, na przykład poprzez weryfikację komunikatów XML pod kątem zgodności z definicjami schematów XML w celu odrzucania złośliwych danych,
- ocenę użyteczności interfejsu użytkownika oraz osiągnięcie określonego poziomu wybranych podcharakterystyk (np. poprzez pomiar czasu potrzebnego na wykonanie konkretnego zadania w sklepie internetowym).

Oprócz ilościowego pomiaru i oceny realizacji celów testów projektu należy również rozważyć ocenę poziomu jakości dokonywaną przez ekspertów oraz interesariuszy posiadających wiedzę dziedzinową.

W zależności od kontekstu projektu oraz przyjętych celów testów może być konieczne wykorzystanie wielu środowisk testowych, wraz z odpowiednimi zasobami i narzędziami testowymi. Należy uwzględnić fakt, że nie wszystkie środowiska testowe mogą być dostępne jednocześnie, co powinno zostać wzięte pod uwagę przy definiowaniu realistycznych celów testów oraz kryteriów wyjścia.

Ponadto, w zależności od kontekstu projektu, podczas określania celów testów oraz zakresu testów projektu należy uwzględnić dodatkowe czynniki opisane w podrozdziale 1.2.

1.5. Doskonalenie procesu testowego

Wstęp. Testowanie stanowi istotny element wytwarzania oprogramowania i jego koszt często stanowi co najmniej 30–40% całkowitych kosztów projektu. Oprócz licznych wyzwań technicznych, z jakimi mierzą się projekty programistyczne (takich jak rosnąca złożoność i skala rozwiązań, nowe technologie, duża różnorodność urządzeń i systemów operacyjnych oraz podatności zabezpieczeń), występuje potrzeba optymalizacji skuteczności i wydajności testowania, a także właściwego doskonalenia procesów testowych. Wykorzystywanie istniejących najlepszych praktyk oraz wyciąganie wniosków z własnych błędów umożliwia doskonalenie procesu testowego i zwiększenie szans powodzenia projektów.

Doskonalenie procesów na poziomie organizacyjnym jest zazwyczaj bardziej efektywne niż działania podejmowane wyłącznie na poziomie projektu lub zespołu. Mimo to możliwe i korzystne jest również wdrażanie udoskonaleń procesowych na poziomie projektu lub zespołu, pod warunkiem, że są one odpowiednio dostosowane do ich specyfiki i potrzeb. Udoskonalanie testów może zostać zainicjowane, na przykład, przez niezadowolające wyniki bieżących testów, wystąpienie nieoczekiwanych defektów, zmieniające się uwarunkowania, rezultaty testów porównawczych lub niedostateczną komunikację. Istnieje wiele technik doskonalenia testowania (Bath i van Veenendaal, 2014), z których część została przedstawiona poniżej. Techniki opisane w niniejszym sylabusie mogą być stosowane zarówno w sekwencyjnych, jak i zwinnych lub przyrostowych modelach wytwarzania oprogramowania. Szczegółowe informacje są dostępne w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspertski Doskonalenie Procesu Testowego.

1.5.1. Proces doskonalenia testów (IDEAL)

Po uzgodnieniu potrzeby doskonalenia procesów testowych można określić czynności związane z wdrażaniem udoskonaleń, korzystając z modelu IDEAL (Roman, 2018), który bazuje na podobnych założeniach jak znany cykl Deminga (*Plan-Do-Check-Act*, PDCA). Nazwa IDEAL jest akronimem etapów: inicjowanie, diagnozowanie, ustanawianie, działanie oraz uczenie się (ang. *Initiating, Diagnosing, Establishing, Acting, Learning*).

Choć model IDEAL został pierwotnie zaprojektowany z myślą o doskonaleniu czynności na poziomie organizacyjnym, może on być z powodzeniem stosowany również na poziomie projektu lub zespołu zwinnego. W kontekście projektowym zakłada się, że cele usprawnień nie zostały jeszcze osiągnięte. Kluczową różnicą jest zazwyczaj znacznie uproszczona faza inicjowania na poziomie projektu lub zespołu w porównaniu z poziomem organizacyjnym. Również diagnozowanie, realizowane np. poprzez retrospektywę, oraz ustalanie planu usprawnień są zwykle mniej rozbudowane. Etapy działania i uczenia się pozostają jednak równie istotne na każdym poziomie.

Inicjowanie doskonalenia procesu testowego. Na początku działań związanych z doskonaleniem procesu interesariusze wspólnie określają cele oraz zakres doskonalenia procesu testowego.

Diagnozowanie aktualnego stanu procesu testowego. Aktualny proces testowy podlega ocenie w celu zidentyfikowania potencjalnych obszarów do doskonalenia. Ocena ta może być przeprowadzona z wykorzystaniem standardowych ram w przypadku podejścia opartego na modelu (patrz sekcja 1.5.2) lub na podstawie analizy konkretnych metryk w podejściu analitycznym (patrz sekcja 1.5.3).

Ustanowienie planu doskonalenia procesu testowego. Plan doskonalenia procesu testowego może przyjąć formę formalnego dokumentu zawierającego szczegółowy wykaz działań niezbędnych do osiągnięcia zakładanych udoskonaleń. W zależności od kontekstu plan ten może być również nieformalny i bardzo uproszczony. Lista potencjalnych udoskonaleń procesu powinna zostać uporządkowana według priorytetów, które mogą być ustalane na podstawie zwrotu z inwestycji (ang. *Return of Investment*, ROI), ryzyk, zgodności ze strategią projektu lub zespołu oraz oczekiwanych mierzalnych korzyści ilościowych lub jakościowych.

Działanie w celu wdrożenia udoskonaleń procesu testowego. Na tym etapie realizowany jest plan usprawnień procesu testowego. Zazwyczaj obejmuje on szkolenia, pilotaż zmodyfikowanych procesów oraz ich pełne wdrożenie w ramach projektu lub zespołu.

Uczenie się na podstawie wdrożonych udoskonaleń procesu testowego. Po zakończeniu wdrażania udoskonaleń należy ocenić, jakie korzyści zostały osiągnięte, zarówno planowane, jak i nieoczekiwane. Analiza tego, co przyniosło pozytywne rezultaty, a co nie spełniło oczekiwań, stanowi podstawę do podjęcia dalszych działań i rozpoczęcia kolejnego cyklu doskonalenia procesu testowego.

1.5.2. Doskonalenie procesu testowego oparte na modelu

Jednym z podstawowych założeń doskonalenia procesu testowego opartego na modelu, jak i doskonalenia opartego na podejściu analitycznym, jest przekonanie, że jakość produktu w dużym stopniu zależy od jakości stosowanych procesów. W podejściu do doskonalenia procesu

testowego opartym na modelu wykorzystywany jest formalny model bazujący na najlepszych praktykach w obszarze testowania i porządkujący działania doskonalące testy w sposób stopniowy.

Istnieje kilka rekomendowanych modeli wspierających doskonalenie procesu testowego, w tym Test Maturity Model integration (TMMi®) oraz TPI NEXT®.

Doskonalenie procesu testowego oparte na modelu może być również realizowane na poziomie projektu. W takich przypadkach proces oceny i doskonalenia koncentrują się przede wszystkim na procesach testowych lub kluczowych obszarach zdefiniowanych w modelu, które odnoszą się do działań projektowych (np. planowanie testów czy projektowanie testów), a często w dużym stopniu pomijają obszary charakterystyczne dla poziomu organizacyjnego (np. politykę testów czy organizację testów). Alternatywnie praktyki przypisane do poziomu organizacyjnego mogą zostać odpowiednio dostosowane do kontekstu projektu.

Szczegółowe informacje dotyczące doskonalenia procesu testowego opartego na modelu są dostępne w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Doskonalenie Procesu Testowego.

Zintegrowany model dojrzałości testów (TMMi®) (van Veenendaal i Cannegieter, 2011; van Veenendaal, 2020) obejmuje pięć poziomów dojrzałości. Każdy poziom, z wyjątkiem poziomu 1., zawiera zdefiniowane obszary procesu testowego oraz cele udoskonaleń. W celu wsparcia i ułatwienia wdrożenia modelu TMMi® opisano również praktyki, podpraktyki oraz przykłady. Model TMMi® został pierwotnie opracowany jako uzupełnienie modelu CMMI® (ang. *Capability Maturity Model Integration*), jednak obecnie jest szeroko stosowany niezależnie od CMMI®.

W celu wsparcia stosowania i aktualizacji TMMi® w kontekście zwinnego wytwarzania oprogramowania przygotowano dedykowane wytyczne, które wyjaśniają, w jaki sposób model ten może być efektywnie wykorzystywany w środowiskach zwinnych.

Dodatkowe informacje na temat TMMi® można znaleźć na stronie www.tmmi.org.

Model TPI Next® (van Ewijk, 2013) definiuje 16 kluczowych obszarów, z których każdy odnosi się do konkretnego aspektu procesu testowego (np. strategii testów, metryk testowych, narzędzi testowych czy środowiska testowego). Dla każdego z tych kluczowych obszarów model określa cztery poziomy dojrzałości.

Dla oceny poszczególnych kluczowych obszarów na każdym poziomie dojrzałości definiowane są szczegółowe punkty kontrolne. Wyniki przeprowadzonej oceny są następnie podsumowywane i prezentowane w postaci macierzy dojrzałości obejmującej wszystkie kluczowe obszary.

Więcej informacji dotyczących TPI Next® jest dostępnych na stronie www.tmap.net.

1.5.3. Podejście analityczne do doskonalenia procesu testowego

W przypadku stosowania podejścia do doskonalenia opartego na modelu, opisanego w poprzedniej sekcji, udoskonalenia są realizowane poprzez porównanie podejścia testowego projektu lub zespołu z zewnętrznymi najlepszymi praktykami. Podejścia analityczne umożliwiają natomiast identyfikację problemów na podstawie danych pochodzących bezpośrednio z projektu lub zespołu. Na bazie analizy tych problemów można określić właściwe działania doskonalące.

Podejścia analityczne mogą być wykorzystywane łącznie z podejściem opartym na modelu w celu weryfikacji wyników oraz zapewnienia większej różnorodności perspektyw.

Problemy mogą być identyfikowane przy użyciu danych ilościowych oraz jakościowych. W tej sekcji przedstawiono podejścia analityczne, które w głównej mierze wykorzystują dane ilościowe z procesu testowego oraz dane dotyczące defektów w celu oceny aktualnego podejścia. Z kolei sekcja 1.5.4 opisuje retrospektywy, w ramach których od członków zespołu wytwórczego i testowego pozyskiwane są dane jakościowe dotyczące elementów funkcjonujących dobrze oraz tych wymagających poprawy.

Analiza danych odgrywa istotną rolę w obiektywnym doskonaleniu procesu testowego i stanowi wartościowe uzupełnienie ocen czysto jakościowych, które w przeciwnym razie mogą prowadzić do nieprecyzyjnych zaleceń nieopartych danymi. Zastosowanie podejścia analitycznego do doskonalenia procesu testowego najczęściej wiąże się z ilościową analizą procesu w celu identyfikacji obszarów problemowych oraz zdefiniowania celów specyficznych dla projektu. Określenie i pomiar kluczowych parametrów są niezbędne do oceny procesu testowego oraz sprawdzenia, czy wdrażane usprawnienia przynoszą oczekiwane rezultaty.

Przykładowe podejścia analityczne obejmują:

- analizę przyczyny podstawowej (ang. *root cause analysis*),
- analizę z wykorzystaniem miar, metryk i wskaźników,
- podejście GQM, czyli Cel-Pytanie-Metryka (ang. *Goal-Question-Metric*).

Analiza przyczyny podstawowej polega na badaniu problemów w celu zidentyfikowania ich rzeczywistych źródeł. Umożliwia to zaprojektowanie rozwiązań eliminujących przyczyny problemów, a nie jedynie reagujących na ich bezpośrednie, widoczne symptomy. Przykładowa procedura może obejmować wybór odpowiedniego zestawu defektów, identyfikację klastrów w tych defektach oraz zastosowanie diagramów przyczynowo-skutkowych (znanych również jako diagramy Ishikawy lub diagramy rybiej ości) w celu określenia podstawowych przyczyn istotnych klastrów defektów. Następnie wdrażane są udoskonalenia mające na celu zapobieganie występowaniu podobnych typów defektów w przyszłości.

Miary, metryki i wskaźniki są stosowane w sposób ilościowy do oceny, jak skutecznie realizowany jest proces testowy w projekcie lub zespole. Kluczowe atrybuty procesu testowego, które należy uwzględnić, to skuteczność, wydajność oraz przewidywalność. Dla każdego z tych atrybutów można dobrać jedną lub więcej odpowiednich metryk. Gromadzenie i analiza właściwych danych umożliwiają identyfikację kluczowych obszarów wymagających udoskonalień.

GQM (Basili i in., 2014; van Solingen i Berghout, 1999) dostarcza ram do definiowania i analizowania metryk dostosowanych do potrzeb informacyjnych właściwych interesariuszy projektu. Cele pomiarowe określają jakościowy aspekt obiektu, który ma zostać zmierzony, w określonym celu, perspektywie i kontekście. Cele te są następnie przekształcane w pytania opisujące dany aspekt jakości z punktu widzenia interesariuszy. Na tej podstawie dobierane są metryki, które dostarczają informacji niezbędnych do udzielenia odpowiedzi na te pytania. Dane zbierane dla metryk umożliwiają odpowiedzi na pytania, ocenę realizacji celu pomiarowego oraz zaspokojenie potrzeb informacyjnych interesariuszy.

Dodatkowe informacje dotyczące podejść do doskonalenia procesów testowych opartych na analizie można znaleźć w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie Udoskonaleń Procesu Testowego.

1.5.4. Retrospektywy

Retrospektywy są spotkaniami, podczas których zespół analizuje stosowane metody pracy oraz współpracę, dokumentuje wyciągnięte wnioski (zarówno pozytywne, jak i negatywne), a także podejmuje decyzje dotyczące zmian i działań mających na celu wprowadzenie udoskonaleń, obejmujących zarówno testowanie, jak i obszary niezwiązane bezpośrednio z testami. Retrospektywy koncentrują się na zagadnieniach takich jak procesy, ludzie, organizacja, współpraca oraz narzędzia.

Retrospektywy są wykorzystywane zarówno w sekwencyjnych modelach wytwarzania oprogramowania, jak i w zwinnym wytwarzaniu oprogramowania. W modelach sekwencyjnych stanowią one element czynności związanych z ukończeniem testów i mają na celu generowanie wniosków, które wspierają lepsze zarządzanie przyszłymi projektami. W zwinnym wytwarzaniu oprogramowania retrospektywy są zazwyczaj realizowane na zakończenie każdej iteracji w celu omówienia elementów zakończonych sukcesem, obszarów wymagających poprawy oraz sposobów uwzględnienia tych udoskonaleń w kolejnej iteracji. Retrospektywy są prowadzone z udziałem całego zespołu, co wspiera podejście „cały zespół” oraz sprzyja ciągłemu doskonaleniu. W niektórych sytuacjach konieczne mogą być dedykowane retrospektywy ukierunkowane na problemy związane z testowaniem.

Typowa retrospektywa obejmuje następujące etapy:

- **Wprowadzenie.** Omawiany jest cel oraz agenda retrospektywy, a także budowana jest atmosfera wzajemnego zaufania, umożliwiająca otwartą dyskusję problemów bez obwiniania lub oceniania uczestników.
- **Zgromadzenie danych.** Zbierane są informacje dotyczące przebiegu iteracji lub projektu. Mogą to być dane jakościowe, takie jak oś czasu kluczowych zdarzeń identyfikująca problemy oraz opinie poszczególnych członków zespołu na temat zdarzeń. Dodatkowo mogą być prezentowane dane ilościowe pochodzące z metryk, na przykład dotyczące postępów testów, wykrywania defektów, skuteczności testów, wydajności testów oraz przewidywalności, które zapewniają obiektywny obraz testowania w projekcie lub iteracji.
- **Wprowadzenie udoskonaleń.** Zebrane dane są analizowane w celu zrozumienia aktualnej sytuacji oraz wygenerowania propozycji udoskonaleń. Przykładowo, analiza podstawowej przyczyny może zostać zastosowana do identyfikacji podstawowych przyczyn wykrytych problemów, a sesja burzy mózgów może posłużyć do opracowania rozwiązań eliminujących te podstawowe przyczyny.
- **Określenie działań doskonalących.** Definiowane są działania niezbędne do wdrożenia zaproponowanych udoskonaleń oraz ustalane są ich priorytety. Określany jest plan udoskonaleń oraz zakres odpowiedzialności. W celu oceny wpływu działań na zidentyfikowane problemy mogą zostać zdefiniowane cele oraz powiązane z nimi metryki. Należy unikać jednoczesnego wdrażania zbyt wielu udoskonaleń, ponieważ utrudnia to realizację działań w sposób możliwy do zweryfikowania.
- **Zamknięcie retrospektywy.** Na tym etapie dokonywany jest przegląd samego przebiegu retrospektywy w celu wskazania jej mocnych stron oraz obszarów wymagających udoskonalenia.

Retrospektywy są realizowane regularnie, w szczególności w zwinnym wytwarzaniu oprogramowania, a zasada ciągłego doskonalenia odnosi się również do samego procesu retrospektywy.

Istotne jest właściwe dokumentowanie rezultatów retrospektywy. W sekwencyjnych modelach wytwarzania ustalenia, wnioski oraz zalecenia powinny być rozpowszechniane i przekazywane członkom organizacji w sposób jasny i zrozumiały. W zwinnym wytwarzaniu oprogramowania problemy oraz działania również powinny być dokumentowane, aby umożliwić przegląd realizowanych działań oraz ocenę ich potencjalnego wpływu na problemy w kolejnej iteracji.

Testerzy, jako członkowie zespołu, wnoszą do retrospektyw unikalną perspektywę. Mogą identyfikować i zgłaszać problemy związane z testowaniem oraz inne zagadnienia, a także zachęcać zespół do rozważania możliwych udoskonaleń.

Dodatkowe informacje na temat retrospektyw 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 prawa.

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 v4.0.

1.6.1. Dobre praktyki dotyczące wprowadzania narzędzi

Niniejsza sekcja opisuje kroki wymagane do przeprowadzenia oceny oraz wdrożenia narzędzia testowego.

Kierownik testów może bezpośrednio uczestniczyć w procesie wdrażania narzędzia lub pełnić rolę wspierającą bądź koordynującą ten proces. Zazwyczaj to właśnie kierownicy testów ponoszą odpowiedzialność za dedykowane narzędzia testowe oraz inne narzędzia powiązane z testowaniem, takie jak narzędzia do zarządzania wymaganiami, defektami czy monitorowaniem.

Podczas **oceny i wyboru narzędzia** testowego należy uwzględnić ogólne dobre praktyki oraz istotne aspekty decyzyjne. Obejmują one w szczególności:

- identyfikację obszarów, w których zastosowanie narzędzia może udoskonalić procesy,
- zrozumienie technologii wykorzystywanych w organizacji oraz wybór narzędzia zgodnego z tym środowiskiem,
- analizę sposobu technicznej i organizacyjnej integracji narzędzia z cyklem wytwarzania oprogramowania,
- ocenę narzędzia w odniesieniu do jasno zdefiniowanych wymagań i obiektywnych kryteriów,
- ocenę dostawcy w przypadku narzędzi komercyjnych lub ocenę dostępnego wsparcia w przypadku narzędzi niekomercyjnych (np. *open source*),

- określenie potrzeb w zakresie coachingu, mentoringu lub szkoleń związanych z użytkowaniem narzędzia,
- analizę zalet i ograniczeń różnych modeli licencjonowania narzędzia,
- końcowy etap w postaci przeprowadzenia oceny słuszności koncepcji (ang. *proof of concept*).

Ogólne dobre praktyki związane z **pozyskaniem i wdrożeniem narzędzia** obejmują:

- realizację projektu pilotażowego w celu potwierdzenia kryteriów wyboru oraz oceny dopasowania narzędzia do istniejących procesów i praktyk,
- dostosowanie i doskonalenie procesów do wymagań narzędzia, a w razie potrzeby również konfigurację narzędzia zgodnie z obowiązującymi procesami,
- zdefiniowanie zasad i wytycznych dotyczących korzystania z narzędzia,
- zapewnienie użytkownikom odpowiednich szkoleń, coachingu oraz mentoringu,
- wdrażanie narzędzia w organizacji w sposób przyrostowy,
- ustanowienie mechanizmów gromadzenia informacji na temat rzeczywistego wykorzystania narzędzia w celu dalszego doskonalenia,
- jednoznaczne określenie odpowiedzialności za narzędzie (ang. *ownership*).

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 uwzględnić i odnieść się do nich.

Przepisy i bezpieczeństwo. Organizacje, które tworzą oprogramowanie o krytycznym znaczeniu dla bezpieczeństwa (ang. *safety-critical*) lub działalności (ang. *mission-critical*) lub podlegają zgodności z przepisami prawa, mogą preferować narzędzia komercyjne, ponieważ częściej spełniają one wymagane standardy i często posiadają odpowiednie, wymagane przepisami, certyfikaty.

Aspekty finansowe. Narzędzia o otwartym kodzie (ang. *open source*) zwykle charakteryzują się niższym kosztem początkowym ze względu na darmowy rozwój i wsparcie przez społeczność jego użytkowników. 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 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 prawa.

1.6.3. Proces wyboru narzędzia i ocena zwrotu z inwestycji

Narzędzia testowe mogą stanowić inwestycję długoterminową, potencjalnie obejmującą wiele iteracji jednego projektu lub znajdującą zastosowanie w wielu projektach. Z tego względu potencjalne narzędzie powinno być analizowane z różnych perspektyw.

- Z punktu widzenia kierownictwa wyższego szczebla wymagany jest dodatni zwrot z inwestycji.
- Z perspektywy zespołu wsparcia i zespołu operacyjnego preferowana jest ograniczona, lecz wystarczająca liczba narzędzi stosowanych w całej organizacji. Utrzymywanie nadmiernej liczby narzędzi, zarządzanie ich licencjami oraz administrowanie stosem narzędzi nie powinno generować nadmiernych kosztów ani wymagać dużego nakładu czasu.
- Dla liderów projektu istotne jest, aby narzędzie wносиło mierzalną wartość do projektu lub organizacji.
- Dla użytkowników końcowych narzędzia kluczowa jest użyteczność, obejmująca m.in. wsparcie dla realizowanych zadań, łatwość uczenia się oraz operacyjność.
- Z punktu widzenia personelu operacyjnego istotna jest łatwość utrzymania narzędzia.

Funkcjonalności powinny być analizowane w odniesieniu do każdego rodzaju narzędzia biznesowego i technicznego. Na taką analizę wpływają różne perspektywy i interesy, takie jak zarządzanie testami, analiza testów technicznych, automatyzacja testów czy wytwarzanie. Osoba odpowiedzialna w organizacji za dane narzędzie (właściciel narzędzia) musi dopilnować, aby analiza została przeprowadzona oraz aby uwzględniono wszystkie wymienione wcześniej aspekty.

Wszystkie narzędzia wprowadzane do procesu testowego powinny zapewniać organizacji dodatni zwrot z inwestycji (ROI). Odpowiedzialność za obliczenie oraz dalszą ocenę ROI spoczywa na kierowniku testów, natomiast w zwinnym wytwarzaniu oprogramowania może ona spoczywać na całym zespole zwinnym.

Przed zakupem lub wytworzeniem narzędzia należy przeprowadzić analizę kosztów i korzyści, aby potwierdzić opłacalność narzędzia dla organizacji. Analiza ta powinna obejmować zarówno koszty jednorazowe, jak i koszty powtarzające się.

Jednorazowe działania i koszty obejmują:

- określenie wymagań wobec narzędzia umożliwiających osiągnięcie założonych celów,
- ocenę i wybór odpowiedniego narzędzia i dostawcy, w tym dowód słuszności koncepcji,
- zakup, dostosowanie lub opracowanie narzędzia do pierwszego użycia,
- zdefiniowanie wytycznych dotyczących użytkowania narzędzia,
- przeprowadzenie szkoleń wstępnych dla użytkowników narzędzia,
- integrację narzędzia z istniejącym środowiskiem,
- zakup sprzętu lub oprogramowania niezbędnego do obsługi narzędzia.

Powtarzające się działania i koszty obejmują:

- cykliczne opłaty licencyjne oraz koszty wsparcia technicznego,
- koszty utrzymania,
- koszty ciągłych szkoleń,
- przenoszenie narzędzia pomiędzy różnymi środowiskami.

Należy również uwzględnić koszty alternatywne, co oznacza, że czas poświęcony na ocenę, administrowanie, szkolenia oraz korzystanie z narzędzia mógłby zostać przeznaczony na faktyczne zadania testowe. W związku z tym przed osiągnięciem zakładanych korzyści może być konieczne zaangażowanie dodatkowych zasobów testowych.

Podczas wyboru narzędzi należy rozważyć następujące ryzyka związane z ROI:

- niedojrzałość organizacji może skutkować nieefektywnym wykorzystaniem narzędzia,
- zmiany w polityce utrzymania po stronie dostawcy mogą zwiększyć wysiłek,
- rzeczywiste koszty mogą okazać się wyższe od przewidywanych,
- uzyskane korzyści mogą być niższe niż oczekiwano.

W odniesieniu do narzędzi testowych można wskazać następujące potencjalne korzyści:

- ograniczenie manualnej, powtarzalnej pracy (np. w testach regresji),
- skrócenie cyklu testowego dzięki automatyzacji,
- obniżenie kosztów wykonania testów przez zmniejszenie zakresu czynności manualnych,
- zwiększenie pokrycia wybranych typów testów obsługiwanych przez narzędzie,
- redukcję błędów ludzkich wynikającą z mniejszej liczby czynności manualnych,
- szybszy dostęp do informacji dotyczących testów.

Dodatkowe korzyści oraz ryzyka, w szczególności związane z narzędziami do automatyzacji testów, zostały opisane w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0 oraz w sylabusie ISTQB® Certyfikowany Tester Poziom Zaawansowany Inżynieria Automatyzacji Testów v2.0.

Z reguły organizacja testowa nie korzysta z jednego narzędzia. Całkowity zwrot z inwestycji dla organizacji jest zazwyczaj wynikiem łącznego ROI wszystkich narzędzi wykorzystywanych w testowaniu. Narzędzia powinny wymieniać informacje i współdziałać ze sobą. Wskazane jest zatem opracowanie długoterminowej, kompleksowej strategii narzędzi testowych, uwzględniającej ryzyka, koszty oraz korzyści.

1.6.4. Cykl życia narzędzia

Cykl życia narzędzia testowego obejmuje cztery główne etapy. Należy wyznaczyć administratora narzędzia, aby zapewnić właściwe planowanie, realizację i zarządzanie działaniami na każdym z tych etapów.

Pozyskanie. Pierwszym krokiem jest podjęcie decyzji o wyborze narzędzia. Następnie należy wyznaczyć właściciela narzędzia, który będzie podejmował decyzje dotyczące jego użytkowania, np. ustalenie konwencji nazewnictwa produktów pracy i miejsca ich przechowywania. Wczesne decyzje w tym zakresie mogą znacząco wpłynąć na ostateczny zwrot z inwestycji w narzędzie.

Wsparcie i utrzymanie. Właściciel narzędzia odpowiada za jego utrzymanie. Zadania pielęgnacyjne mogą być realizowane przez administratora narzędzia lub przeznaczony do tego zespół utrzymujący. W przypadku współdzielonych narzędzi należy uwzględnić procesy wymiany danych, komunikacji oraz współpracy. Konieczne jest także określenie zasad tworzenia kopii zapasowych i przywracania artefaktów związanych z narzędziem.

Ewolucja. Wraz z upływem czasu zmieniają się potrzeby biznesowe, środowisko narzędzia lub decyzje dostawców, co może wymagać modyfikacji narzędzia. Im bardziej złożone staje się środowisko pracy narzędzia, tym większe ryzyko, że zmiany zakłócą jego użytkowanie.

Wycofanie. Po zakończeniu okresu użytkowania narzędzie należy wycofać. Funkcjonalność dostarczana przez narzędzie zostaje zwykle zastąpiona nowym rozwiązaniem, a dane muszą zostać zachowane lub zarchiwizowane. Wycofanie może wynikać z decyzji dostawcy lub z oceny, że przejście na nowe narzędzie przynosi większe korzyści przy akceptowalnych kosztach i ryzykach.

1.6.5. Metryki narzędzia

Metryki generowane przez narzędzia testowe są zbierane w oparciu o potrzeby zespołu testowego oraz innych interesariuszy. Dzięki temu narzędzia często rejestrują cenne dane w czasie rzeczywistym, zmniejszając wysiłek związany z ich manualnym gromadzeniem. Dane te wspierają zarządzanie wysiłkiem przy testach oraz identyfikowanie obszarów wymagających optymalizacji.

Różne typy narzędzi gromadzą różne rodzaje informacji:

- Narzędzia do zarządzania testami dostarczają metryki dotyczące dostępnych elementów testowych, przypadków testowych, planowanych testów oraz statusów ich wykonania (np. zaliczony, niezaliczony, pominięty, zablokowany, planowany).
- Narzędzia do zarządzania wymaganiami zapewniają śledzenie pokrycia wymagań przez zaliczone i niezaliczone przypadki testowe.
- Narzędzia do zarządzania defektami dostarczają informacje o defektach, takie jak status, krytyczność, priorytet, gęstość defektów, a także dane wspomagające doskonalenie procesu, np. procent wykrywalności defektów, poziomy testów, na których występują defekty czy czas naprawy.
- Narzędzia do analizy statycznej dostarczają metryki związane ze złożonością kodu.
- Narzędzia do testowania wydajności mierzą czasy reakcji i wskaźniki awaryjności przy szczytowym obciążeniu.
- Narzędzia pokrycia kodu dostarczają informacje o tym, które części przedmiotu testów zostały przetestowane.
- 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 informacji na temat gromadzenia i wykorzystania metryk znajduje się w podrozdziale 2.1.

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 dziedziny

poker planistyczny, szacowanie trójpunktowe, szerokopasmowa technika delficka

Cele nauczania dla rozdziału 2

2.1 Metryki testów

TM-2.1.1 (K2) Podać przykłady metryk służących osiągnięciu celów testów.

TM-2.1.2 (K2) Objaśnić, w jaki sposób kontrolować postęp testów przy użyciu metryk testów.

TM-2.1.3 (K4) Przeanalizować wyniki testów w celu tworzenia raportów z testów, które umożliwiają interesariuszom podejmowanie decyzji.

2.2 Szacowanie testów

TM-2.2.1 (K2) Wyjaśnić czynniki, które należy wziąć pod uwagę przy szacowaniu testów.

TM-2.2.2 (K2) Podać przykłady czynników, które mogą mieć wpływ na szacowanie testów.

TM-2.2.3 (K4) Wybrać odpowiednią technikę lub podejście do szacowania testów w danym kontekście.

2.3 Zarządzanie defektami

TM-2.3.1 (K3) Wdrożyć proces zarządzania defektami, w tym przepływ pracy dotyczący defektu, który można wykorzystać do monitorowania i kontroli defektów.

TM-2.3.2 (K2) Objaśnić proces zarządzania defektami i identyfikować jego uczestników.

TM-2.3.3 (K2) Wyjaśnić specyfikę zarządzania defektami w zwinnym wytwarzaniu oprogramowania.

TM-2.3.4 (K2) Wyjaśnić wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania.

TM-2.3.5 (K3) Wykorzystać dane i informacje klasyfikacyjne, które należy gromadzić podczas zarządzania defektami.

TM-2.3.6 (K2) Wyjaśnić, w jaki sposób statystyki zgłoszeń defektów można wykorzystać do opracowania udoskonalenia procesu.

2.1. Metryki testów

Wstęp. W zarządzaniu często mówi się: „Co zostanie zmierzone, zostanie zrobione”. Oznacza to, że jeśli coś nie jest monitorowane, łatwo to przeoczyć i prawdopodobnie nie zostanie to wykonane. Dlatego kluczowe jest określenie odpowiednich metryk dla każdego przedsięwzięcia, w tym dla testowania oprogramowania.

Cele testów odpowiadają na pytanie, dlaczego testujemy (patrz podrozdział 1.4). W celu określenia, czy cele testów zostały zrealizowane, należy zdefiniować sposób ich pomiaru. Metryki wskazują, jak ocenić realizację tych celów.

Metryki testów dzielą się na trzy główne kategorie:

- **metryki projektu** – mierzą postęp projektu względem ustalonych kryteriów wyjścia, np. odsetek testów wykonanych, zaliczonych lub niezaliczonych,
- **metryki produktu** – oceniają atrybuty produktu, np. stopień, w jakim spełnia on oczekiwania jakościowe użytkowników,
- **metryki procesu** – mierzą efektywność procesu testowego i skuteczność testowania, wspierając raportowanie efektywności i wydajności procesu.

Szczegółowe informacje o zarządzaniu metrykami produktu i procesu znajdują się w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Zarządzanie Testami, natomiast o wykorzystaniu metryk procesu w doskonaleniu testów – w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie Uoskonalień Procesu Testowego.

W kolejnych sekcjach omawiane są metryki w kontekście czterech głównych działań zarządczych: planowania testów, monitorowania testów, nadzoru nad testami oraz ukończenia testów.

2.1.1. Metryki czynności związanych z zarządzaniem testami

Niniejszy sylabus skupia się na trzech głównych czynnościach zarządczych w testowaniu (patrz podrozdział 1.1):

- planowanie testów,
- monitorowanie testów i nadzór nad testami,
- ukończenie testów.

Kierownik testów musi określić odpowiedni zestaw metryk testów dla każdej z tych czynności. Każda metryka powinna być jasno zdefiniowana, mierzona, monitorowana i raportowana.

Na potrzeby planowania testów wybiera się metryki dopasowane do celów testów określonych w projektowej strategii testów.

Metryki stosowane podczas monitorowania testów i nadzoru nad testami mogą różnić się od tych używanych w trakcie ukończenia testów. Na potrzeby monitorowania i nadzoru wybiera się metryki koncentrujące się na postępie czynności testowych.

Na potrzeby ukończenia testów wybiera się metryki pozwalające ocenić, czy cele testów zostały osiągnięte, często łącząc kilka metryk w celu zmierzenia kryteriów wyjścia przypisanych do konkretnych celów testu.

Tabela 2 prezentuje przykłady metryk wykorzystywanych w zarządzaniu testami (lista nie jest wyczerpująca).

Tabela 2. Przykłady metryk stosowanych 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 czasu czynności testowych	X	
Odsetek wykonanych przypadków testowych wg statusu w porównaniu z planowanymi przypadkami testowymi	X	X
Skumulowana liczba usuniętych defektów vs. łączna liczba defektów	X	
Rzeczywiste zautomatyzowane przypadki testowe vs. planowane zautomatyzowane przypadki testowe		X
Rzeczywisty a planowany koszt testów	X	

W tabeli przedstawiono metryki przypisane do konkretnych czynności testowych. Interpretacja znaków X w kolumnach wygląda następująco: znak X przy monitorowaniu i nadzorze nad testami oznacza, że metryka jest wykorzystywana głównie do pomiaru postępu testów i raportowana w raportach o postępie testów (CTFL). Znak X przy ukończeniu testów – że metryka służy głównie do oceny osiągnięcia celów testów i jest uwzględniana w sumarycznym raporcie z testów (CTFL). Znak X w obu kolumnach – że metryka może być stosowana zarówno do monitorowania postępu, jak i do oceny ukończenia testów.

Istnieją również metryki, które służą do monitorowania skuteczności testów, np. odsetek wykrytych defektów (ang. *Defect Detection Percentage* – DDP). Metryka DDP jest szczegółowo omówiona w sylabusie ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie Udoskonaleń Procesu Testowego.

2.1.2. Monitorowanie, nadzór i ukończenie testów

Metryki testów informują o stopniu zaawansowania testów oraz o tym, czy osiągnięto kryteria wyjścia lub wykonano powiązane zadania testowe.

Monitorowanie testów polega na zbieraniu danych dotyczących testów oraz ich ocenie i analizie. Jest stosowane w celu śledzenia postępów testów oraz weryfikacji spełnienia kryteriów wyjścia lub wykonania powiązanych czynności testowych (patrz sekcja 1.1.2). Kryteria wyjścia wynikają z celów testów.

Nadzór nad testami wykorzystuje informacje pozyskane w trakcie monitorowania testów, aby zapewnić wskazówki i działania korygujące w celu efektywnego i wydajnego przeprowadzenia testów. Przykłady dyrektyw nadzoru obejmują: zmianę priorytetów testów w przypadku wystąpienia istotnego ryzyka, ponowną ocenę zgodności elementu testowego z kryteriami wejścia lub wyjścia po wprowadzeniu zmian, dostosowanie harmonogramu testów z uwagi na opóźnienia w udostępnieniu środowiska testowego oraz przydzielanie dodatkowych zasobów, jeśli jest to możliwe i konieczne.

Ukończenie testów obejmuje zebranie danych z wykonanych czynności testowych w celu skonsolidowania wniosków, testaliów i innych istotnych informacji. Ma ono miejsce w kluczowych

punktach projektu, takich jak ukończenie testów danego poziomu, zakończenie iteracji, zakończenie lub anulowanie projektu testowego, wydanie produktu lub zakończenie wydania pielęgnacyjnego.

Typowe metryki testów stosowane w zarządzaniu testami obejmują metryki postępu projektu oraz metryki pokazujące stopień realizacji harmonogramu i budżetu, aktualną jakość elementów testowych oraz skuteczność testowania względem celów testów lub celów iteracji.

2.1.3. Raportowanie testów

Kierownik testów powinien posiadać wiedzę na temat interpretacji i wykorzystywania metryk w celu zrozumienia oraz raportowania statusu testów. Na wyższych poziomach testów, takich jak testowanie systemowe, testowanie integracji systemów czy testowanie akceptacyjne, główną podstawą testów są zwykle produkty pracy takie jak specyfikacje wymagań, przypadki użycia, historyjki użytkownika oraz ryzyka produktowe. Metryki pokrycia strukturalnego znajdują większe zastosowanie na niższych poziomach testów, takich jak testowanie modułowe (np. pokrycie instrukcji) oraz testowanie integracji modułów (np. pokrycie interfejsów). Choć w zarządzaniu testami metryki pokrycia kodu mogą służyć do oceny zakresu, w jakim testy obejmują strukturę testowanego systemu, raportowanie wyników testów na wyższych poziomach powinno być dostosowane do specyfiki projektu i jego potrzeb. Na przykład w dynamicznie zmieniających się środowiskach metryki pokrycia kodu mogą wspierać monitorowanie wpływu zmian kodu na zestaw testowy oraz identyfikację potencjalnych luk lub ryzyk. Kierownik testów musi również rozumieć, że nawet przy osiągnięciu 100% pokrycia strukturalnego przez testy modułowe i integracji modułów, nadal mogą występować defekty i ryzyka jakościowe wymagające eliminacji na wyższych poziomach testów.

Celem raportowania metryk jest umożliwienie natychmiastowego zrozumienia informacji dla potrzeb zarządzania. Metryki mogą być raportowane jako „migawka” w określonym momencie (pojedyncza wartość) lub w formie ewolucji w czasie w celu analizy trendu metryki.

Ryzyka produktowe, defekty, postęp testów, pokrycie oraz powiązane koszty i wysiłek testowy są mierzone i raportowane w określony sposób na zakończenie projektu.

Poniżej podano przykłady metryk stosowanych do różnych celów.

Metryki ryzyk produktowych

- odsetek ryzyk, które przeszły pomyślnie wszystkie testy z nimi związane,
- odsetek ryzyk, w przypadku których niektóre lub wszystkie testy nie zostały zaliczone,
- odsetek ryzyk, które nie zostały jeszcze w pełni przetestowane.

Metryki te służą ocenie jakości podstawy testów oraz skuteczności przypadków testowych w pokrywaniu ryzyk produktowych.

Metryki defektów

- skumulowana liczba usuniętych defektów w stosunku do całkowitej liczby defektów,
- podział liczby lub odsetka defektów według kategorii:
 - miejsce defektu (np. określony moduł lub funkcja),
 - źródło defektu (np. specyfikacja wymagań, nowa funkcja, regresja),
 - wersja testu, który wykrył defekt,

- o poziom testu lub iteracja, w której defekt został wprowadzony, wykryty i usunięty,
- o priorytet/krytyczność,
- o podstawowa przyczyna,
- o status (np. odrzucony, duplikat, otwarty, zamknięty).

Metryki te umożliwiają monitorowanie procesu wykrywania i usuwania defektów, identyfikację obszarów o wysokiej gęstości defektów lub dużej krytyczności defektów oraz ocenę wydajności i skuteczności testów.

Metryki postępu testów

- status wykonania testów: całkowita liczba zaplanowanych, wdrożonych, wykonanych, zaliczonych, niezaliczonych, zablokowanych i pominiętych testów,
- wysiłek związany z testowaniem: rzeczywista i planowana liczba godzin poświęconych na testowanie.

Metryki pokrycia

- pokrycie wymagań: odsetek wymagań pokrytych przypadkami testowymi,
- pokrycie ryzyka produktowego: odsetek zidentyfikowanych ryzyk produktowych, które są łagodzone przez przypadki testowe,
- pokrycie kodu: odsetek instrukcji kodu, gałęzi, ścieżek lub warunków wykonywanych przez przypadki testowe.

Metryki kosztów i wysiłku związanego z testowaniem

- ryzyka rezydualne (resztkowe) w nieprzetestowanych modułach: potencjalny wpływ i prawdopodobieństwo wystąpienia defektów w modułach, które nie zostały przetestowane,
- koszt testów: rzeczywisty w stosunku do planowanego.

Przydatne jest też łączenie metryk z różnych kategorii (np. metryki pokazujące korelację między trendami otwartych defektów a trendami wykonanych testów lub metryki oceniające jakość podstawy testów na podstawie liczby defektów wykrytych w wymaganiach). W miarę postępu testów i spadku liczby wykrywanych defektów można podjąć decyzję o zakończeniu testów. Decyzja ta powinna być oparta na raportowanych metrykach oraz uzgodnionych kryteriach wyjścia.

2.2. Szacowanie testów

Wstęp. Najlepsze praktyki zarządzania projektami w zakresie szacowania w inżynierii systemów i oprogramowania obejmują wszystkie rodzaje zasobów, takie jak koszty, zasoby ludzkie czy czas. Szacowanie testów polega na zastosowaniu tych sprawdzonych praktyk w kontekście testowania związanego z projektem lub prowadzonymi działaniami.

2.2.1. Oszacowanie, jakie czynności będzie obejmować testowanie

Szacowanie testów jest działaniem w ramach zarządzania testami, umożliwiającym oszacowanie czasu, wysiłku (nakładu pracy) oraz kosztów potrzebnych do wykonania określonego zadania. Szacowanie testów stanowi jedno z kluczowych zadań w zarządzaniu testami.

Główne cechy szacowania w zarządzaniu testami obejmują:

- **Wysiłek (nakład pracy)** (ile czasu/osobogodzin będzie wymagane?) – zwykle wyrażany w osobogodzinach lub punktach historyjek użytkownika niezbędnych do realizacji zadań testowych w projekcie. Często czas pracy wynikający z wysiłku związanego z testowaniem oraz czas trwania testów mogą się różnić. Zarządzanie testami może wymagać oszacowania całkowitego czasu potrzebnego na wykonanie czynności.
- **Czas potrzebny do ukończenia projektu** (jak długo potrwa projekt testowy?) – czas jest krytycznym zasobem w projekcie. Planowanie testów wymaga oszacowania wysiłku testowego w dniach kalendarzowych i roboczych. Każdy projekt posiada kamienie milowe i określony termin realizacji.
- **Koszt** (ile będzie kosztować realizacja projektu testowego?) – obejmuje budżet projektu, w tym wydatki na zasoby testowe, narzędzia oraz infrastrukturę.

Testowanie często funkcjonuje jako podprojekt w ramach większego projektu i może być rozproszone w kilku lokalizacjach testowych (np. w centrach testowych). Pierwszym krokiem w szacowaniu testów jest określenie poziomów testów, czynności testowych oraz zadań testowych. Następnie projekt testowy należy podzielić na główne czynności testowe (np. planowanie i wykonywanie testów) w ramach procesu testowego (patrz sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0). W projektach zwinnych czynności testowe są często szacowane w kontekście prac wytwórczych, a nie jako odrębne elementy. Kolejnym etapem jest oszacowanie wysiłku testowego potrzebnego do wykonania zadań lub produktów pracy oraz przewidywanych kosztów z tym związanych.

Ponieważ testowanie stanowi część projektu, zawsze istnieją pewne naturalne ograniczenia projektowe, które na nie wpływają i wymagają kompromisu – nie można arbitralnie manipulować tymi wartościami. W zarządzaniu jakością jest to znane jako trójkąt czas-koszt-jakość. W zarządzaniu projektami trójkąt ten obejmuje trzy współzależne wartości, które są ściśle powiązane i wzajemnie na siebie oddziałują. Zależność ta jest powszechnie obserwowana w scenariuszach projektowych.

2.2.2. Czynniki, które mogą wpływać na wysiłek testowy

Szacowanie wysiłku testowego polega na przewidywaniu ilości pracy związanej z czynnościami testowymi, niezbędnej do osiągnięcia celów testów w kontekście konkretnego projektu, wydania lub iteracji. Czynniki wpływające na wysiłek testowy mogą obejmować następujące aspekty:

Produkt

- jakość podstawy testów,
- rozmiar przedmiotu testów,
- złożoność dziedziny biznesowej produktu (np. środowisko, infrastruktura, historia produktu),
- wymagania dotyczące testowania charakterystyk jakościowych (np. zabezpieczenia, niezawodność).

Czynniki związane z produktem wpływają na szacowanie testów, ponieważ definiują specyficzny kontekst testowanego systemu.

Proces wytwórczy

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

Czynniki związane z procesem wytwórczym wpływają na szacowanie testów, gdyż testowanie jest bezpośrednio powiązane z procesem wytwarzania.

Ludzie

- czynniki dotyczące dostępności i zadowolenia osób (np. dni ustawowo wolne od pracy, urlopy, inne przewidziane korzyści),
- umiejętności i doświadczenie zaangażowanych osób, zwłaszcza w odniesieniu do podobnych projektów i produktów (np. znajomość dziedziny biznesowej).

Ludzie są kluczowym zasobem, dlatego każda niestabilność w tym obszarze musi zostać uwzględniona. Z tego względu są istotnym czynnikiem przy szacowaniu wysiłku związanego z testowaniem (patrz podrozdział 3.1).

Wyniki testów

- liczba i krytyczność defektów wykrytych podczas wykonywania testów,
- liczba wymaganych przeróbek.

Statystyki wyników testów wspierają dokładniejsze szacowanie testów, a znajomość tych czynników pozwala przewidzieć bardziej precyzyjne wartości dotyczące defektów i przeróbek.

Kontekst testów

- rozproszenie testów w wielu lokalizacjach, skład i rozmieszczenie zespołów, złożoność projektu (np. wiele podsystemów),
- rodzaj pracy (np. zdalna lub wykonywana na miejscu).

Czynniki kontekstowe wpływają na całe szacowanie testów i powinny być uwzględnione w planowaniu (patrz podrozdział 1.2).

2.2.3. Wybór technik szacowania testów

Szacowanie testów powinno obejmować wszystkie czynności wchodzące w skład procesu testowego. Szacowanie kosztów, wysiłku oraz szczególnie czasu wykonywania testów jest często kluczowe z punktu widzenia zarządzania testami, ponieważ mają istotny wpływ na projekt. Jednocześnie szacowanie dotyczące wykonywania testów mogą być trudne do wykonania, gdy ogólna jakość oprogramowania jest niska lub nieznana. Znajomość produktu oraz doświadczenie w jego zakresie mają istotny wpływ na dokładność szacowania. Powszechną praktyką jest szacowanie liczby przypadków testowych w oparciu o podstawę testów, taką jak wymagania lub historyjki użytkownika. Założenia przyjęte w trakcie szacowania powinny być zawsze udokumentowane jako część procesu szacowania.

Techniki szacowania testów można podzielić na techniki oparte na miarach i techniki oparte na wiedzy eksperckiej. Szczegółowe informacje na temat technik szacowania testów znajdują się w sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0.

W większości przypadków przygotowane oszacowanie musi zostać przedstawione kierownictwu projektu wraz z uzasadnieniem. Parametry wejściowe, takie jak zakres testów, mogą ulegać zmianie, co często wymaga aktualizacji szacowania. W optymalnych warunkach ostateczne szacowanie testów stanowi najlepszą możliwą równowagę między celami organizacyjnymi i projektowymi w obszarach jakości, harmonogramu, budżetu i funkcji.

Należy pamiętać, że szacowanie opiera się na informacjach dostępnych w momencie ich sporządzenia, które na wczesnym etapie projektu mogą być ograniczone, a ponadto mogą ulegać zmianie w czasie. W celu zachowania dokładności szacowanie powinno być aktualizowane w odpowiedzi na nowe i zmieniające się informacje.

Wybór techniki szacowania zależy od różnych czynników, w tym tych podanych poniżej.

- **Błąd szacowania.** Niektóre techniki umożliwiają obliczenie odchylenia standardowego, które mierzy niepewność (lub błąd) oszacowania. Na przykład technika szacowania trójpunktowego wykorzystuje wartości optymistyczne, pesymistyczne i najbardziej prawdopodobne do obliczenia wartości oczekiwanej i odchylenia standardowego (patrz sylabus Certyfikowany Tester Poziom Podstawowy v4.0, sekcja 5.1.4).
- **Dostępność danych.** Niektóre techniki wymagają danych historycznych z wcześniejszych lub podobnych projektów, które mogą być niedostępne lub nie w pełni wiarygodne. Na przykład szacowanie na podstawie proporcji czy ekstrapolacja wykorzystują dane historyczne do określenia wskaźników lub trendów dla bieżącego projektu.
- **Dostępność ekspertów.** Niektóre techniki wymagają udziału ekspertów posiadających odpowiednią wiedzę i doświadczenie w celu uzyskania dokładnych i realistycznych oszacowań. Na przykład szerokopasmowa technika delficka oraz poker planistyczny bazują na opiniach i osądach ekspertów lub członków zespołu.
- **Wiedza w zakresie modelowania.** Niektóre techniki wymagają stosowania modeli matematycznych do przeprowadzenia szacowania, co wymaga umiejętności i wiedzy w zakresie modelowania. Przykłady obejmują ekstrapolację i szacowanie trójpunktowe, które wykorzystują odpowiednie wzory do obliczenia wartości oczekiwanej i odchylenia standardowego oszacowania.
- **Ograniczenia czasowe.** Niektóre techniki są bardziej czasochłonne i wymagają większego wysiłku, co może wpływać na ich wykonalność i przydatność. Na przykład poker planistyczny to technika prosta do przeprowadzenia, podczas gdy ekstrapolacja może się okazać znacznie trudniejsza.

Kryteria wyboru odpowiednich technik szacowania testów w dużej mierze zależą od kontekstu testowania, w tym od modelu wytwarzania oprogramowania, interesariuszy, poziomów testów i typów testów stosowanych w projekcie (patrz podrozdział 1.2). Kierownik testów musi być w stanie koordynować oraz stosować techniki szacowania testów w różnych kontekstach projektowych, w tym przy wykorzystaniu różnych modeli wytwarzania oprogramowania w różnych lokalizacjach projektu. Na przykład przy wyborze techniki szacowania należy najpierw określić złożoność problemu. W przypadku niskiej złożoności można stosować techniki oparte na metrykach. Przy wysokiej złożoności bardziej odpowiednie są techniki oparte na wiedzy eksperckiej. Dla projektów realizowanych w sekwencyjnym modelu wytwarzania można zastosować szerokopasmową technikę delficką, natomiast w projektach zwinnych – poker planistyczny.

2.3. Zarządzanie defektami

Wstęp. W sylabusie ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0 opisano działania, które rozpoczynają się po zaobserwowaniu rzeczywistych wyników testów, które różnią się od wyników oczekiwanych. W sylabusie czynności te nazywa się zarządzaniem defektami. Inne źródła używają terminu „zarządzanie incydentami” (norma ISO/IEC/IEEE 29119-3) lub „zarządzanie anomaliami” (metodyka TMap), aby podkreślić fakt, że na początku procesu możemy nie wiedzieć, czy rozbieżność jest spowodowana rzeczywistym defektem produktu pracy, czy też czymś innym (np. awarią automatyzacji testów lub niezrozumieniem wymagań przez testera). Zarządzanie defektami i narzędzia 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 życia. Zarządzanie defektami ma również kluczowe znaczenie przy podejmowaniu decyzji o tym, które defekty zostaną usunięte. Zapewnia to, że wysiłek będzie optymalnie poświęcany 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 cyklu życia (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 analizy tych procesów. Kierownik testów musi właściwie wykorzystywać zarówno proces zarządzania defektami, jak i wybrane narzędzia do zarządzania defektami.

2.3.1. Cykl życia defektu

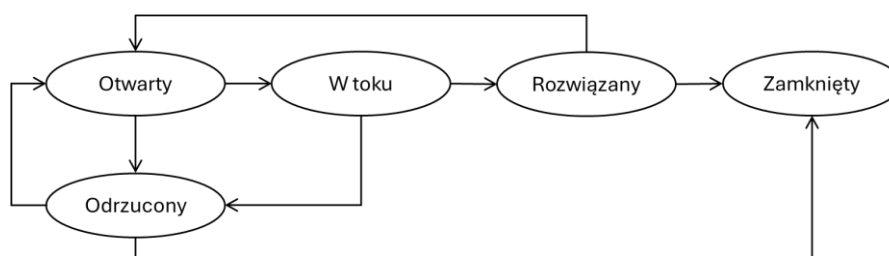
Każda faza cyklu życia oprogramowania powinna obejmować czynności mające na celu wykrywanie i eliminowanie potencjalnych defektów. Na przykład techniki testowania statycznego, takie jak przeglądy i analiza statyczna, mogą być stosowane do specyfikacji projektowych, specyfikacji wymagań oraz kodu przed ich przekazaniem do kolejnych etapów prac, aby umożliwić podjęcie dalszych działań. Im wcześniej defekt zostanie wykryty i usunięty, tym niższy jest całkowity koszt jakości produktu. Minimalizacja kosztu jakości następuje wówczas, gdy każdy defekt jest usuwany w tej samej fazie, w której został wprowadzony, co odpowiada idei tzw. idealnego powstrzymania fazowego (ang. *phase containment*) w procesie wytwarzania oprogramowania.

Podczas testowania statycznego celem jest wykrycie defektów. W testowaniu dynamicznym defekt staje się widoczny, gdy powoduje awarię (określaną jako anomalia), skutkując rozbieżnością między rzeczywistymi a oczekiwanymi wynikami testu. W niektórych przypadkach może wystąpić wynik fałszywie negatywny, gdy tester nie zauważy anomalii. W przypadku wykrycia anomalii konieczne jest przeprowadzenie dalszych badań, które zazwyczaj rozpoczynają się od wypełnienia raportu o defekcie zgodnie z ustalonym procesem zarządzania defektami.

Niezaliczenie testu nie zawsze prowadzi do utworzenia raportu o defekcie. Przykładem są procesy wytwarzania sterowanego testami, w których testy modułowe – często zautomatyzowane – pełnią funkcję wykonywalnej specyfikacji projektu. Do momentu zakończenia opracowywania modułu

niektóre lub wszystkie testy mogą początkowo kończyć się niezaliczeniem. Dlatego wynik takiego testu nie musi wynikać z defektu i zwykle nie jest rejestrowany w raporcie o defekcie.

Raport o defekcie podlega ustalonemu przepływowi pracy (w dalszej części dla uproszczenia oraz w zgodności z większością narzędzi do zarządzania defektami używany będzie termin „przeływ pracy dotyczący defektu”) i przechodzi przez sekwencję stanów (statusów) defektu. W większości tych stanów jedna osoba pełni rolę właściciela raportu o defekcie i odpowiada za wykonanie przypisanego zadania, takiego jak analiza, usunięcie defektu lub wykonanie testu potwierdzającego. Rys. 2 przedstawia przykładowy prosty przeływ pracy dotyczący defektu.



Rys. 2. Przykładowy prosty przeływ pracy dotyczący defektu

Prosty przeływ pracy dotyczący defektu może obejmować następujące stany defektu:

- OTWARTY (lub NOWY): stan początkowy podczas tworzenia raportu o defekcie.
- W TOKU: zespół pracuje nad analizą raportu o defekcie i usunięciem defektu.
- 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 (lub USUNIĘTY, GOTOWY 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 stosowane są przepływy pracy dotyczące defektu, które są rozszerzane o inne stany defektów istotne dla danego kontekstu (np. ponownie otwarty, zaakceptowany, doprecyzowany, odroczone).

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

- Jeżeli to możliwe, przepływ pracy dotyczący defektu powinien być zdefiniowany na poziomie całej organizacji, aby zapewnić spójne zarządzanie defektami we wszystkich realizowanych projektach.
- Duplikaty oraz zgłoszenia fałszywie pozytywne powinny być oznaczone odrębnym stanem lub jako kombinacja statusu ODRZUCONY z określeniem przyczyny odrzucenia. Takie podejście może wspierać późniejszą analizę defektów oraz działania doskonalące proces testowy.
- Zaleca się stosowanie jednego stanu końcowego (np. ZAMKNIĘTY). Przejście do tego stanu zazwyczaj wymaga wskazania powodu zamknięcia, co jest przydatne do oceny procesu oraz identyfikacji możliwości jego doskonalenia.
- Nazwy stanów w przepływie pracy dotyczącym defektu powinny być spójne z nazwami analogicznych stanów stosowanych dla innych elementów pracy (np. historyjek użytkownika lub zadań testowych), aby uprościć sposób pracy z nimi.
- Kolejne stany defektu powinny być przypisane do różnych ról odpowiedzialnych za obsługę zgłoszenia. Jeżeli co najmniej dwa następujące po sobie stany są przypisane do tej samej roli, powinno to mieć uzasadnienie (np. potrzeba pomiaru czasu spędzonego w danym stanie defektu).
- Każdy stan defektu, z wyjątkiem stanu końcowego, powinien posiadać więcej niż jedno przejście wychodzące, aby umożliwić odpowiedzialnej roli podjęcie decyzji dotyczącej kolejnego kroku. Odstępstwa od tej zasady powinny być uzasadnione (np. w celu monitorowania czasu poświęconego na określoną czynność).
- Zestaw atrybutów wymaganych przy wykonywaniu przejścia do danego stanu powinien być ograniczony do tych, które wnoszą istotną wartość do zarządzania defektami.

2.3.2. Interdyscyplinarne zarządzanie defektami

Chociaż organizacja testowa oraz kierownik testów często pełnią rolę właścicieli ogólnego procesu zarządzania defektami oraz narzędzia do zarządzania defektami, odpowiedzialność za zarządzanie defektami w ramach konkretnego projektu zazwyczaj spoczywa na zespole interdyscyplinarnym (ang. *cross-functional*). Zespół ten, określany niekiedy jako komitet ds. zarządzania defektami, może obejmować kierownika testów, przedstawicieli zespołów wytwórczych, dostawców, kierownictwo projektu, kierownictwo produktu lub właściciela produktu, a także innych interesariuszy zainteresowanych testowanym oprogramowaniem.

W miarę identyfikowania anomalii i ich rejestrowania w narzędziu do zarządzania defektami komitet ds. zarządzania defektami powinien decydować, czy dany raport o defekcie rzeczywiście opisuje defekt oraz czy zgłoszony defekt powinien zostać usunięty (a jeśli w dostawie uczestniczy wiele zespołów wytwórczych – przez który zespół), odrzucony lub odroczone. Podjęcie takiej decyzji wymaga rozważenia korzyści, ryzyk oraz kosztów związanych z usunięciem defektu. Zwykle pomocne jest omawianie defektów podczas spotkań zespołu, często określanych jako spotkania segregujące (lub triażowe, od fr. *triage*). Jeżeli zapadnie decyzja o usunięciu defektu, zespół powinien ustalić jego priorytet w odniesieniu do pozostałych zadań. W tym zakresie można konsultować się z kierownikiem testów i zespołem testowym, którzy powinni dostarczyć dostępne, obiektywne informacje dotyczące względnego znaczenia defektu.

W bardzo dużych projektach uzasadnione może być wyznaczenie pełnoetatowego kierownika ds. defektów, ze względu na wysiłek wymagany do przygotowania oraz monitorowania decyzji podejmowanych przez komitet ds. zarządzania defektami, przynajmniej w tych fazach cyklu życia,

w których testowanie jest najbardziej intensywne. W innych przypadkach jeden kierownik ds. defektów może być współdzielony przez kilka dużych projektów.

Narzędzie do zarządzania defektami nie powinno zastępować dobrej komunikacji, podobnie jak komitet ds. zarządzania defektami nie powinien zastępować skutecznego wykorzystania odpowiedniego narzędzia do zarządzania defektami. Skuteczne i wydajne zarządzanie defektami wymaga potężnej efektywnej komunikacji, właściwego wsparcia narzędziowego, dobrze zdefiniowanego przepływu pracy dotyczącego defektu (w tym odpowiednich atrybutów raportów o defektach) oraz zaangażowanego zespołu ds. zarządzania defektami.

2.3.3. Specyfika zarządzania defektami w zespołach zwinnych

Zarządzanie defektami w organizacjach stosujących zwinne wytwarzanie oprogramowania jest zazwyczaj mniej rozbudowane, mniej sformalizowane lub bardziej uproszczone niż w sekwencyjnych modelach wytwarzania. Jeżeli zespoły zwinne pracują w jednej lokalizacji lub dysponują skutecznymi środkami komunikacji, informacje o defektach lub awariach są często przekazywane pomiędzy testerami, przedstawicielami klientów i programistami bez tworzenia formalnego raportu o defekcie. Niemniej jednak raporty o defektach powinny być sporządzane w przypadku:

- defektów, które blokują inne trwające czynności w iteracji (tj. wytwarzanie, testowanie lub inne) i nie mogą zostać natychmiast usunięte przez zespół zwinny,
- defektów, których nie można usunąć w tej samej iteracji; niektóre zespoły zwinne stosują zasadę tworzenia raportu o defekcie, jeżeli nie może on zostać usunięty w dniu wykrycia awarii,
- defektów wymagających rozwiązania przez inne zespoły lub we współpracy z nimi w środowiskach wielozespołowych,
- defektów, które muszą zostać usunięte przez dostawcę,
- defektów, dla których raport o defekcie jest jednoznacznie wymagany (np. gdy programista nie jest w stanie natychmiast rozpocząć prac nad usunięciem defektu).

Powszechną praktyką jest dodawanie do backlogu produktu defektów, których nie można usunąć w tej samej iteracji, aby możliwe było nadanie im priorytetu wśród innych defektów oraz historyjek użytkownika na potrzeby kolejnych iteracji.

Chociaż podstawowe zasady zarządzania defektami powinny być określone w organizacyjnej strategii testów, wiele elementów, takich jak poziom sformalizowania, wyzwalacze tworzenia raportu o defekcie oraz atrybuty defektu podlegające rejestracji, może zostać uzgodnionych pomiędzy członkami zespołu zwinnego. Poziom sformalizowania zarządzania defektami oraz podejście do raportowania defektów powinny uwzględniać m.in.:

- współlokalizację członków zespołu,
- rozmieszczenie zespołu w różnych strefach czasowych,
- liczbę zespołów współpracujących przy wytwarzaniu produktu,
- dojrzałość zespołu lub zespołów,
- wielkość zespołu lub zespołów,
- ryzyka produktowe,
- wymogi regulacyjne, umowne lub inne (jeżeli mają zastosowanie).

Ostateczna decyzja zespołu zwinnego dotycząca szczegółów zarządzania defektami powinna być zawsze udokumentowana, na przykład w postaci wytycznych zapisanych w narzędziu do zarządzania wiedzą.

2.3.4. Wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania

W praktyce przy dostarczaniu systemu lub systemu systemów często współdziela wiele zespołów. Przykładem mogą być hybrydowe modele wytwarzania oprogramowania, w których klient stosuje zwinne wytwarzanie oprogramowania, natomiast jeden z dostawców korzysta z sekwencyjnego modelu wytwarzania, lub sytuacje, gdy organizacja pracująca w modelu sekwencyjnym zleca dostarczenie podsystemu zespołowi wykorzystującemu zwinne wytwarzanie oprogramowania. Takie środowiska wielozespołowe generują różnorodne wyzwania:

Dostosowanie atrybutów defektów oraz narzędzi do zarządzania defektami. W idealnym przypadku wszystkie zespoły korzystają z jednego narzędzia do zarządzania defektami. W praktyce jednak często każdy zespół używa innego narzędzia, w szczególności, gdy w realizację projektu zaangażowanych jest wielu dostawców. W takich sytuacjach zaleca się ustanowienie synchronizacji pomiędzy narzędziami do zarządzania defektami, najlepiej w sposób zautomatyzowany.

Priorytetyzacja defektów. Właściciele produktu powinni uczestniczyć w spotkaniach dotyczących zarządzania defektami oraz aktywnie pozyskiwać informacje o konsekwencjach i ryzykach związanych z defektami. Spotkania te powinny odbywać się częściej w środowiskach zwinnych niż w sekwencyjnych modelach wytwarzania, aby nadążyć za szybszym tempem dostarczania przyrostów produktu przez zespoły zwinne, przy czym w zespołach zwinnych spotkania te mogą być krótsze. W niektórych przypadkach korzystne może być powierzenie ostatecznej decyzji dotyczącej priorytetyzacji defektów mniejszej grupie interesariuszy odpowiedzialnych za zarządzanie defektami.

Dostosowanie oraz przejrzystość planu testów w kontekście nowego wytwarzania i usuwania defektów. Prace wszystkich zespołów powinny być spójne z tym samym planem projektu, niezależnie od stosowanego modelu wytwarzania oprogramowania – zwinnego lub sekwencyjnego. Wszystkie rezultaty, w tym poprawki defektów, powinny być zgodne z tym planem. Lepsze dostosowanie można osiągnąć poprzez aktywne uczestnictwo przedstawicieli wszystkich zespołów w planowaniu, na przykład poprzez udział zespołów stosujących sekwencyjny model wytwarzania w spotkaniach zespołów zwinnych, podczas których omawiane są defekty i ustalane ich priorytety. Przejrzystość planów wytwarzania można zwiększyć poprzez ich udostępnianie pomiędzy zespołami, na przykład za pomocą tablic wskaźników lub backlogu produktu.

2.3.5. Informacje zawarte w raporcie o defekcie

Informacje zawarte w raporcie o defekcie powinny wystarczać do realizacji następujących celów:

- zarządzania raportem o defekcie w całym cyklu życia defektu,
- oceny ogólnego statusu projektu, w szczególności w odniesieniu do jakości produktu oraz postępów testowania,
- oceny jakości przyrostu produktu,
- oceny zdolności i skuteczności procesu.

Zakres informacji niezbędnych do zarządzania defektami oraz monitorowania statusu projektu może się różnić w zależności od etapu cyklu życia, na którym defekt został wykryty. Dodatkowo raporty o defektach dotyczących niefunkcjonalnych charakterystyk jakościowych mogą wymagać bardziej szczegółowych danych, takich jak warunki obciążenia w przypadku problemów wydajnościowych. Niezależnie od tego, podstawowy zestaw informacji powinien być spójny w całym cyklu życia, a najlepiej również we wszystkich projektach organizacji, aby umożliwić rzetelne porównywanie danych o defektach.

Raport o defekcie może zawierać wiele różnych atrybutów. Kierownik testów powinien określić, które z nich są niezbędne do skutecznego zarządzania defektami w danym projekcie. Ponieważ każdy dodatkowy atrybut zwiększa wysiłek związany z raportowaniem defektów i może powodować dezorientację osoby zgłaszającej, zaleca się ograniczenie zbieranych danych do tych, które są rzeczywiście potrzebne do zarządzania defektami oraz do doskonalenia procesu.

W większości środowisk do zarządzania raportem o defekcie wymagane są co najmniej następujące informacje:

- tytuł defektu zawierający krótkie podsumowanie anomalii,
- szczegółowy opis anomalii, najlepiej wraz z krokami umożliwiającymi odtworzenie awarii,
- określenie wpływu defektu na system lub interesariuszy (krytyczność),
- priorytet usunięcia defektu.

Narzędzia do zarządzania defektami zazwyczaj automatycznie generują dodatkowe istotne informacje, takie jak:

- unikalny identyfikator raportu o defekcie,
- data i godzina utworzenia raportu,
- dane osoby, która wykryła lub zgłosiła defekt,
- faza projektu oraz etap cyklu życia, w którym wykryto anomalię,
- aktualny status raportu o defekcie,
- bieżący właściciel raportu (osoba odpowiedzialna za dalsze działania związane z defektem),
- historia zmian, obejmująca sekwencję podjętych działań wraz z datami i godzinami, dotyczących analizy, naprawy oraz potwierdzenia usunięcia defektu,
- powiązania i odniesienia, na przykład do przypadków testowych lub innych defektów.

W zależności od kontekstu projektu w raporcie o defekcie mogą być również gromadzone dodatkowe informacje, takie jak powiązania z wymaganiami (szczegółowe wytyczne zawiera norma ISO/IEC/IEEE 29119-3). Informacje te można pogrupować według ich przeznaczenia:

- **Wsparcie usuwania defektu:** dane dotyczące podsystemu lub modułu, w którym występuje defekt, konkretnego elementu testowego oraz jego wersji, a także środowiska testowego, w którym defekt został zaobserwowany.
- **Ocena ogólnego statusu projektu:** informacje umożliwiające monitorowanie postępów, w tym dotyczące ryzyk, kosztów, potencjalnych korzyści i konsekwencji usunięcia lub pozostawienia defektu, a także dostępnych obejść lub wymagań, na które defekt ma wpływ.

- **Ocena jakości przyrostu produktu:** klasyfikacja defektu (zgodnie z przyjętą taksonomią), wskazanie produktu pracy, w którym defekt został wprowadzony, lub charakterystyki i podcharakterystyki jakościowej, której defekt dotyczy.
- **Ocena zdolności procesu:** informacje pozwalające analizować skuteczność i wydajność procesów wytwarzania, takie jak etap cyklu życia wprowadzenia, wykrycia i usunięcia defektu oraz jego podstawowa przyczyna.

2.3.6. Określanie działań doskonalących proces przy użyciu informacji z raportów o defektach

Jak podano w sekcji 2.3.5, raporty o defektach mogą być przydatne do monitorowania i raportowania statusu projektu. Podczas gdy konsekwencje stosowania metryk dla procesu testowego zostały przede wszystkim uwzględnione w sylabusie ISTQB® Certyfikowany Tester 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 omówionych w sekcjach 2.1.2 oraz 2.1.3, informacje o defektach powinny wspierać inicjatywy doskonalenia procesu, wynikające z przeprowadzonych retrospektyw. Na przykład:

- wykorzystanie informacji o fazach wprowadzania, wykrywania i usuwania defektów do oceny powstrzymania fazowego 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 defektu do analizy faz, w których wprowadzana jest największa liczba defektów, aby umożliwić ukierunkowane doskonalenie zapobiegania defektom,
- wykorzystanie informacji o podstawowych przyczynach defektów w celu określenia pierwotnych przyczyn wprowadzenia defektów, aby umożliwić udoskonalenia procesu, które zmniejszą całkowitą liczbę defektów,
- wykorzystanie informacji o lokalizacji defektów w celu przeprowadzenia analizy klastrow 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 procesu debugowania,
- wykorzystanie informacji o zduplikowanych i odrzuconych defektach do oceny jakości tworzenia raportów o defektach,
- umożliwienie doskonalenia procesu w celu zmniejszenia całkowitej liczby defektów, poprzez wprowadzenie proaktywnych środków, aby zawczasu uniknąć popełniania błędów.

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

W niektórych przypadkach zespoły decydują się nie śledzić defektów wykrytych podczas niektórych lub we wszystkich fazach cyklu życia. 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 udoskonalenia 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, oszacowanie, wykrycie, zapobieganie defektom

Cele nauczania dla rozdziału 3

3.1 Zespół testowy

TM-3.1.1 (K2) Podać przykłady typowych umiejętności potrzebnych członkom zespołu testowego w czterech obszarach kompetencji.

TM-3.1.2 (K4) Przeanalizować dany kontekst projektu w celu określenia wymaganych umiejętności członków zespołu testowego.

TM-3.1.3 (K2) Objąć typowe techniki oceny umiejętności członków zespołu testowego.

TM-3.1.4 (K2) Rozróżniać między typowymi podejściami do rozwijania umiejętności członków zespołu testowego.

TM-3.1.5 (K2) Objąć umiejętności wymagane do zarządzania zespołem testowym.

TM-3.1.6 (K2) Podać przykłady czynników motywujących i czynników higienicznych dla członków zespołu testowego.

3.2 Relacje z interesariuszami

TM-3.2.1 (K2) Podać przykłady określające koszt jakości w każdej z czterech kategorii.

TM-3.2.2 (K3) Zastosować obliczenia kosztów i korzyści w celu oszacowania wartości dodanej testów dla interesariuszy.

3.1. Zespół testowy

Wstęp. Każdy zespół realizujący zadania testowe składa się z osób o zróżnicowanych kompetencjach. W niektórych organizacjach zespoły są samoorganizujące się, natomiast w innych ich tworzenie oraz rozwój należą do odpowiedzialności kierownika testów. Niezależnie od modelu organizacyjnego, właściwa kombinacja umiejętności członków zespołu stanowi kluczowy czynnik warunkujący skuteczną realizację działań testowych.

Zakres kompetencji wymaganych od członków zespołu testowego może zmieniać się w czasie. Z tego względu istotne jest zarówno odpowiednie dobranie osób do zespołu, jak i zapewnienie im adekwatnych szkoleń oraz możliwości dalszego rozwoju. Dodatkowo określone, wyspecjalizowane umiejętności mogą być wniesione przez osoby spoza zespołu testowego.

W niniejszym rozdziale omówiono podstawowy proces identyfikacji oraz rozwijania kompetencji niezbędnych członkom zespołu testowego, a także umiejętności wymagane do skutecznego kierowania zespołem lub prowadzenia działań szkoleniowych. Uwzględniono również wiedzę dotyczącą czynników motywujących i demotyujących członków zespołu testowego oraz innych aspektów sprzyjających efektywnej współpracy zespołowej.

Każda osoba posiada określony zestaw umiejętności, które mogą być rozwijane poprzez doświadczenie zawodowe, edukację oraz szkolenia. Idealny zespół testowy dysponuje wszystkimi kompetencjami niezbędnymi do realizacji powierzonych mu zadań testowych lub odpowiada wyłącznie za takie zadania, do których posiada wymagane kwalifikacje. Skuteczne funkcjonowanie zespołu testowego wymaga posiadania różnorodnych umiejętności na różnych poziomach zaawansowania. W zależności od kontekstu projektu znaczenie poszczególnych kompetencji może się różnić. W przypadku zadań testowych wykraczających poza możliwości zespołu zasadne jest zaangażowanie zewnętrznych ekspertów.

3.1.1. Typowe umiejętności w ramach czterech obszarów kompetencji

Kompetencje posiadane przez daną osobę można sklasyfikować w czterech głównych obszarach (Sonntag i Schmidt-Rathjens, 2005; Erpenbeck i von Rosenstiel, 2017):

- **Kompetencje zawodowe** – obejmują zdolność do realizacji wyspecjalizowanych zadań. Zaliczają się do nich m.in. umiejętności stosowania technik testowania, wiedza techniczna i biznesowa dotycząca dziedziny aplikacji, a także kompetencje z zakresu zarządzania projektami.
- **Kompetencje metodologiczne** – odnoszą się do ogólnych umiejętności, które mogą być wykorzystywane niezależnie od konkretnej dziedziny i umożliwiają samodzielne wykonywanie złożonych lub nowych zadań. Przykładami są umiejętności analityczne, koncepcyjne oraz zdolność oceny sytuacji.
- **Kompetencje społeczne** – dotyczą umiejętności komunikacji, współpracy oraz rozwiązywania konfliktów w kontekście wewnątrz- i międzykulturowym. Umożliwiają one budowanie relacji z innymi osobami w celu właściwego działania w danej sytuacji oraz realizacji celów indywidualnych i zespołowych. Obejmują m.in. umiejętności efektywnego porozumiewania się, zdolność rozwiązywania konfliktów, pracę zespołową, elastyczność oraz asertywność.
- **Kompetencje osobiste** – obejmują zdolność i gotowość do samorozwoju, rozwijania własnych talentów, motywacji oraz postaw, a także kształtowania osobowości. Przykładami są

zarządzanie sobą, odpowiedzialność osobista, umiejętność przyjmowania informacji zwrotnej, rzetelność, odporność psychiczna, pewność działania, samodyscyplina, otwartość na zmiany, gotowość do pomocy i uczenia się oraz zdolność delegowania zadań.

Wszystkie wymienione obszary kompetencji mają istotne znaczenie dla skutecznego funkcjonowania zespołu testowego. Ponieważ kompetencje metodologiczne, społeczne i osobiste nie są specyficzne wyłącznie dla testowania, ISTQB® koncentruje się przede wszystkim na rozwijaniu kompetencji zawodowych. Obejmują one m.in. umiejętność zarządzania zadaniami testowymi, analizowania podstawy testów, projektowania testów, identyfikowania i analizowania ryzyk oraz opracowywania, konfigurowania i utrzymywania danych testowych, środowisk testowych oraz skryptów testowych.

Cztery opisane wyżej i wykorzystane w niniejszym sylabusie obszary kompetencji opierają się na modelu opisanym w zacytowanych źródłach, który jest szeroko stosowany. Istnieją inne modele opisane w literaturze, które grupują umiejętności w inny sposób. Nie są one częścią tego sylabusu.

3.1.2. Analiza wymaganych umiejętności członków zespołu testowego

Zatrudnianie personelu stanowi element planowania testów i obejmuje określenie ról oraz kompetencji wymaganych do realizacji testowania zgodnie ze strategią testów. Aby właściwie zidentyfikować niezbędne umiejętności w danym projekcie, konieczne jest przeprowadzenie szczegółowej analizy jego kontekstu.

Kompetencje zawodowe i metodologiczne

W testowaniu szczególny nacisk kładzie się na umiejętności niezbędne do realizacji poszczególnych zadań testowych, w tym m.in.:

- planowanie testów – wymaga kompetencji koncepcyjnych umożliwiających opracowanie strategii testów,
- monitorowanie testów i nadzór nad testami – wymagają umiejętności z zakresu zarządzania projektami, niezbędnych do koordynowania wszystkich zadań testowych,
- analiza testów – wymaga zdolności analitycznych potrzebnych do analizy podstawy testów oraz ryzyk produktowych,
- projektowanie testów – wymaga znajomości technik testowania umożliwiających projektowanie przypadków testowych oraz wiedzy koncepcyjnej w zakresie projektowania środowisk testowych,
- implementacja testów – wymaga umiejętności oceny przy wyborze testów oraz specjalistycznej wiedzy technicznej, m.in. w zakresie tworzenia skryptów testowych i konfiguracji środowisk testowych,
- wykonywanie testów – wymaga specjalistycznych kompetencji technicznych niezbędnych do realizacji testów automatycznych, prowadzenia testów eksploracyjnych oraz oceny wyników testów,
- ukończenie testów – wymaga umiejętności skutecznego komunikowania wyników projektu oraz osobistej odpowiedzialności za podejmowane decyzje.

Różne poziomy i typy testów wiążą się z odmiennymi wymaganiami kompetencyjnymi, np. wiedzą biznesową w celu oceny funkcjonalnej przydatności systemu lub zaawansowaną wiedzą techniczną potrzebną do oceny utrzymywalności kodu.

Dodatkowo kontekst projektu dostarcza istotnych informacji dotyczących wymaganych kompetencji zawodowych, takich jak:

- dziedzina biznesowa systemu – wymaga wiedzy biznesowej w danym obszarze zastosowań, np. technologii informacyjnej, motoryzacji czy branży hazardowej,
- architektura systemu i technologie – mogą wymagać specjalistycznej wiedzy technicznej, np. w zakresie języków programowania, technologii interfejsów czy zagadnień podatności zabezpieczeń,
- model wytwarzania oprogramowania – wymaga znajomości poziomów testów, ról w testowaniu oraz odpowiednich technik testowania.

Kompetencje społeczne. W kontekście testowania kompetencje społeczne umożliwiają członkom zespołu testowego skuteczną współpracę z innymi interesariuszami oraz realizację celów testów. Obejmują one w szczególności umiejętności komunikacyjne, zdolność do współpracy oraz rozwiązywania konfliktów, np. konstruktywne reagowanie na nieoptymalne warunki testowe czy skuteczne zgłaszanie defektów programistom.

Ponieważ wytwarzanie i testowanie oprogramowania są zazwyczaj realizowane przez różne grupy interesariuszy, koordynacja działań opiera się głównie na komunikacji. Z tego względu umiejętność pracy zespołowej, efektywnej komunikacji i rozwiązywania konfliktów ma kluczowe znaczenie dla powodzenia projektu. Wymagany poziom kompetencji społecznych zależy jednak od kontekstu, np. zwinne wytwarzanie oprogramowania zazwyczaj stawia wyższe wymagania w tym zakresie niż sekwencyjne, silnie udokumentowane modele wytwarzania lub testowanie realizowane poza siedzibą organizacji.

Kompetencje osobiste. Skuteczność i wydajność pracy członków zespołu testowego zależą również od ich gotowości do samorozwoju, doskonalenia umiejętności oraz kształtowania odpowiednich postaw. Na przykład praca w samoorganizującym się zespole zwinnym wymaga wysokiego poziomu samodzielności, samodyscypliny i odpowiedzialności od wszystkich jego członków, natomiast kierownik testów w strukturze hierarchicznej musi posiadać umiejętność delegowania zadań.

Wysoki poziom rzetelności i odporności psychicznej jest szczególnie istotny w projektach o dużej presji czasowej. Ponadto gotowość do współpracy, uczenia się oraz otwartość na zmiany stanowią ważne czynniki wspierające skuteczne funkcjonowanie zespołów testowych we wszystkich modelach wytwarzania oprogramowania.

3.1.3. Ocena umiejętności członków zespołu testowego

W wielu sytuacjach zespoły testowe są budowane w oparciu o już dostępny personel. Aby poznać potencjał członków zespołu oraz potrzeby w zakresie rozwoju osobistego, kierownik testów powinien przeanalizować aktualne umiejętności zespołu testowego i zestawić je z wymaganymi kompetencjami. Porównanie to może zostać udokumentowane przy użyciu macierzy umiejętności.

Dostępne są modele wspierające zespoły i ich członków w bardziej efektywnej współpracy (np. „Role zespołowe” M. Belbina, DISG® lub PCM®). Zgodnie z koncepcją Belbina (Belbin, 2010) zespoły osiągają wysoką efektywność, gdy składają się z osób reprezentujących różne typy osobowości i role zespołowe. Powyższe modele pomagają zespołom zidentyfikować posiadane kompetencje oraz obszary, w których mogą występować braki.

Kompetencje zawodowe członków zespołu testowego mogą być oceniane na podstawie wykonywanych przez nich typowych zadań testowych, takich jak:

- przedstawienie strategii testów oraz omówienie informacji zwrotnych ze współpracownikami,
- przeprowadzenie przeglądu podstawy testów i przekazanie wyników, co może również ujawnić kompetencje komunikacyjne,
- określanie technik testowania w celu realizacji konkretnych celów testów w kontekście danego projektu,
- właściwe stosowanie różnych technik testowania,
- przygotowanie sumarycznego raportu z testów, zawierającego ocenę wyników testów.

Dodatkowo umiejętności te mogą być oceniane na podstawie zewnętrznych poświadczeń, certyfikatów, doświadczenia zawodowego oraz stopni naukowych.

Szczególnie w zwinnym wytwarzaniu oprogramowania zespoły regularnie identyfikują potrzebne kompetencje poprzez udział w retrospektywach oraz uzyskiwanie informacji zwrotnych. Doświadczeni trenerzy lub mentorzy wspierają zespoły w rozwijaniu umiejętności oraz w identyfikowaniu i uzupełnianiu luk kompetencyjnych.

3.1.4. Rozwijanie umiejętności członków zespołu testowego

Na początku projektu zespół testowy nie zawsze dysponuje kompletem wymaganych umiejętności. Nawet jeśli idealny zestaw osób nie jest dostępny, dobrze zbudowany zespół może skutecznie równoważyć indywidualne mocne i słabe strony swoich poszczególnych członków.

Kierownik testów lub sam zespół testowy może zidentyfikować potrzeby rozwojowe, porównując wymagane kompetencje z aktualnie dostępnymi umiejętnościami, np. za pomocą macierzy umiejętności. Na tej podstawie można dobrać odpowiednie podejścia do rozwoju kompetencji, takie jak:

- szkolenia i edukacja – przekazują z góry zdefiniowaną wiedzę i praktyki, są realizowane zazwyczaj w formie zajęć stacjonarnych lub wirtualnych (np. udział w kursach, szkoleniach wewnętrznych, szkoleniach „szytych na miarę”, kursach e-learningowych na żywo),
- samokształcenie – samodzielne zdobywanie wiedzy poza (wirtualną) salą szkoleniową, np. poprzez czytanie książek, oglądanie nagranych materiałów wideo czy korzystanie z zasobów internetowych,
- wzajemne uczenie się – dzielenie się wiedzą, doświadczeniami i pomysłami pomiędzy współpracownikami oraz uczenie się od siebie nawzajem,
- mentoring lub coaching – formy wsparcia, w których nowy lub mniej doświadczony członek zespołu otrzymuje indywidualne wskazówki od trenera lub korzysta z wiedzy i doświadczenia mentora, będącego stałym źródłem porad i pomocy,

- szkolenie w miejscu pracy – podejście łączące elementy samokształcenia, wzajemnego uczenia się oraz mentoringu lub coachingu.

Nie wszystkie metody rozwoju kompetencji są równie skuteczne i efektywne. Na przykład samokształcenie oraz szkolenia dobrze sprawdzają się w rozwijaniu kompetencji zawodowych i metodologicznych. Z tego względu podstawową wiedzę z zakresu testowania można rozwijać poprzez udział w szkoleniach ISTQB® lub samodzielną naukę opartą na sylabusach ISTQB®. Natomiast w przypadku kompetencji społecznych i osobistych bardziej efektywne są podejścia takie jak szkolenia interaktywne, mentoring i coaching. Wymiana społeczna, regularna informacja zwrotna oraz refleksja stanowią kluczowe czynniki sukcesu w rozwijaniu tych obszarów kompetencji.

3.1.5. Umiejętności kierownicze wymagane do zarządzania zespołem testowym

Osoba, która chce efektywnie kierować zespołem testowym, powinna posiadać wysokie kompetencje zarządcze. Obejmują one zarówno kompetencje zawodowe, jak i metodologiczne, niezbędne do realizacji podstawowych działań menedżerskich, takich jak: planowanie, monitorowanie postępów, nadzór czy raportowanie. W obszarze testowania wymagana jest również odpowiednia wiedza i umiejętności związane z zarządzaniem testami, np. znajomość różnych podejść do testowania, opracowywanie strategii testów, stosowanie technik testowania oraz znajomość wykorzystywanego modelu wytwarzania oprogramowania.

Skuteczne kierowanie lub coaching zespołu testowego wymaga odpowiedniego działania w relacjach z członkami zespołu, a także gotowości do rozwoju w zmieniających się okolicznościach. Z tego powodu kompetencje społeczne i osobiste są kluczowym czynnikiem sukcesu kierownika testów. Należą do nich odporność, umiejętność delegowania zadań oraz zdolność do zdobycia akceptacji zespołu. Dodatkowo obejmują one reprezentowanie interesów testowych w projekcie, promowanie wartości testowania oraz efektywną komunikację i rozwiązywanie konfliktów ze wszystkimi interesariuszami.

Rekrutacja członków zespołu testowego wymaga umiejętności analizy warunków społecznych, charakterystyki zespołu oraz środowiska pracy. Umiejętności te pozwalają zapewnić właściwe dopasowanie zespołu do warunków pracy lub, jeśli to możliwe, dostosowanie środowiska pracy do potrzeb zespołu. Ponadto, zespoły testowe przechodzą przez dynamiczne procesy rozwojowe, co wymaga kompetencji sytuacyjnych, np. zgodnie z fazami modelu rozwoju małych grup Tuckmana (Tuckman, 1965; Bonebright, 2010):

- formowanie (ang. *forming*) – wspieranie nowych członków zespołu w adaptacji do grupy,
- docieranie się (ang. *storming*) – rozwiązywanie konfliktów wewnątrz zespołu,
- normowanie (ang. *norming*) – wprowadzanie dyscypliny i przywództwa ukierunkowanego na cele, aby zapewnić wspólne wartości i zasady,
- działanie (ang. *performing*) – delegowanie zadań w sposób umożliwiający zespołowi przejęcie osobistej odpowiedzialności,
- rozwiązanie (ang. *adjourning*) – zarządzanie odejściem członków zespołu z poszanowaniem i uznaniem.

3.1.6. Czynniki motywujące lub demotywujące zespół testowy w określonych sytuacjach

Zadowoleni i zmotywowani członkowie zespołu testowego przyczyniają się do zwiększenia produktywności i wydajności, co znacząco wpływa na sukces realizowanych projektów. Gdy uda się osiągnąć odpowiedni poziom zaangażowania, szkolenia przekrojowe mogą odbywać się nieformalnie, członkowie zespołu testowego samodzielnie zarządzają swoim obciążeniem pracą, a kierownik testów zyskuje więcej czasu na zajmowanie się zagadnieniami zewnętrznymi.

Dwuczynnikowa teoria Herzberga (Herzberg et al., 1993) wyróżnia dwa rodzaje czynników mobilizujących pracowników: motywujące oraz higieniczne.

Czynniki motywujące są dostrzegane w sposób świadomy i mogą prowadzić do wzrostu satysfakcji oraz rozwoju zawodowego, np.:

- uznanie i docenienie pracy – np. nagrody, zachęty lub inne indywidualne formy docenienia, które członkowie zespołu uznają za wartościowe,
- większa odpowiedzialność i autonomia – np. możliwość samodzielnego definiowania procesów testowych w zespole,
- interesujące i znaczące zadania – wyzwania, które są osiągalne, a jednocześnie warte podjęcia, np. wdrożenie nowego narzędzia do automatyzacji testów,
- awans i rozwój zawodowy – np. możliwość awansu doświadczonego testera na kierownika testów lub właściciela procesu testowego.

Czynniki higieniczne są zwykle uznawane za oczywiste; ich obecność nie zwiększa automatycznie satysfakcji, natomiast ich brak może demotywować zespół testowy, np.:

- odpowiednie wynagrodzenie – np. zgodne z rynkowymi stawki, płatne nadgodziny, korzystne świadczenia socjalne,
- polityka personalna i styl zarządzania – np. realistyczne cele, szczupłe zarządzanie (ang. *lean management*), ochrona przed przeciążeniem i nadmiernym wpływem zewnętrznym,
- przyjazne warunki pracy – np. jednoznaczne specyfikacje, dojrzałe przedmioty testów, właściwe usuwanie defektów, ergonomiczne miejsce pracy, stabilne środowisko testowe,
- bezpieczeństwo egzystencjalne – np. stabilne miejsce pracy i przestrzeganie umów,
- dobre relacje interpersonalne – np. bezkonfliktowa współpraca ze współpracownikami i przełożonymi.

W praktyce zarządzania testami oznacza to konieczność stałego eliminowania czynników demotywujących przy jednoczesnym tworzeniu i wzmacnianiu czynników motywujących.

Więcej informacji na ten temat można znaleźć w pracach Belbina (2010), Marstona (1999) oraz Kahlera (2008).

3.2. Relacje z interesariuszami

Wstęp. W zarządzaniu testami kluczowe jest optymalizowanie procesów testowych w celu zapewnienia maksymalnej wartości biznesowej. Nadmierne testowanie może prowadzić do niepotrzebnych opóźnień i kosztów przewyższających korzyści, natomiast zbyt ograniczone

testy mogą skutkować dostarczeniem użytkownikom produktu o niewystarczającej jakości. Optymalny poziom testowania znajduje się pomiędzy tymi skrajnościami. Kierownik testów odpowiada za wspieranie interesariuszy w zrozumieniu tej równowagi oraz w docenieniu wartości dodanej testowania, uwzględniając jednocześnie typowe ograniczenia czasowe i zasobowe projektu.

3.2.1. Koszt jakości

Korzyści z testowania są równoważone kosztami jakości. Sposób ilościowego określenia całkowitego kosztu wysiłku związanego z testowaniem i defektów związanych z jakością nazywa się kosztem jakości (ang. *cost of quality*). Koszt jakości obejmuje klasyfikację kosztów projektu i kosztów operacyjnych według czterech kategorii:

- **Koszty prewencji** (zapobiegania defektom) – koszt wszystkich działań, które są zaplanowane i proaktywne, aby zapobiec niskiej jakości (np. szkolenie programistów w zakresie tworzenia łatwego w utrzymaniu lub bezpiecznego kodu, przegląd podstawy testów tak wcześnie, jak to możliwe oraz odpowiednia komunikacja w zespole).
- **Koszty wykrycia** defektów – koszt wszystkich czynności mających na celu wykrywanie defektów (np. przeprowadzanie testowania statycznego i dynamicznego 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 po wydaniu, związanych z niską jakością produktu, 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, usuwanie defektów polowych (zgłoszonych przez klientów)).

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

Istnieje więcej podejść, które można rozważyć w celu określenia kosztu jakości. Niniejszy sylabus jest oparty na opisanym powyżej podejściu (Feigenbaum, 1956), a np. sylabus Certyfikowany Tester Poziom Podstawowy v4.0 (podrozdział 1.3) przedstawia podejście Boehma. Te dwa podejścia zostały wybrane w celu osiągnięcia szerszego zrozumienia kosztu jakości. Podejście Feigenbauma traktuje jakość jako proces zorientowany na klienta i obejmujący całą organizację, podczas gdy podejście Boehma (Boehm, 1979) koncentruje się na kompromisie między kosztem zapobiegania a kosztem awarii w wytwarzaniu oprogramowania (Hadjicostas, 2004).

3.2.2. Stosunek kosztów do korzyści testowania

Choć większość organizacji uznaje testowanie za wartościowe w pewnym zakresie, niewielu kierowników, w tym kierowników testów, potrafi ilościowo określić lub jasno przedstawić tę wartość. Wielu kierowników testów, liderów testów i testerów koncentruje się na operacyjnych szczegółach testowania – takich jak konkretne zadania testowe czy poziomy testów – często

pomijając kwestie o charakterze taktycznym i strategicznym, które są istotne dla innych interesariuszy, zwłaszcza kadry kierowniczej.

Testowanie przynosi korzyści zarówno organizacji, projektowi, jak i procesom operacyjnym, w wymiarze jakościowym i ilościowym:

- **Korzyści jakościowe** obejmują: poprawę reputacji w zakresie jakości, bardziej przewidywalne i płynne wydania, wzrost zaufania, ochronę przed odpowiedzialnością prawną oraz ograniczenie ryzyka utraty biznesu, a nawet utraty życia.
- **Korzyści ilościowe** obejmują: liczbę wykrytych lub usuniętych defektów przed wydaniem, dokumentację defektów, które nie zostały usunięte, oszczędności kosztowe (Bohm 1981; Böhler 2008) oraz zmniejszenie ryzyka dzięki przeprowadzaniu testów i dostarczaniu informacji o statusie projektu, procesie i produkcie.

Dodatkową wartością testowania jest umożliwienie interesariuszom podejmowania świadomych decyzji dotyczących gotowości produktu do wydania, zarówno z defektami, jak i bez nich. W niektórych przypadkach dopuszczenie defektów w produkcie może być bardziej korzystne niż opóźnianie wydania, w zależności od prawdopodobieństwa i krytyczności defektu.

Obliczanie kosztu jakości na defekt odbywa się w następujący sposób.

$$\text{Średnia oszczędność na defekt} = \begin{aligned} &\text{średni koszt awarii zewnętrznej} \\ &- (\text{średni koszt wykrycia} \\ &+ \text{średni koszt awarii wewnętrznej}) \end{aligned}$$

$$\begin{aligned} \text{Całkowity koszt jakości} = & \text{koszty prewencji} \\ & + (\text{koszty wykrycia} + \text{koszty awarii wewnętrznej}) \times \text{liczba defektów} \\ & \text{wykrytych przed wydaniem} \\ & + \text{koszty awarii zewnętrznej} \times \text{liczba defektów wykrytych} \\ & \text{po wydaniu} \end{aligned}$$

Jako przykład przyjmijmy, że obliczono w następujący sposób koszt jakości:

- średni koszt zapobiegania defektowi: 180 \$,
- średni koszt wykrycia defektu: 500 \$,
- średni koszt awarii wewnętrznej na defekt: 200 \$,
- średni koszt awarii zewnętrznej na defekt: 4000 \$.

Średni koszt zapobiegania, wykrycia i awarii wewnętrznej oblicza się na podstawie liczby defektów wykrytych przed wydaniem, podczas gdy średni koszt awarii zewnętrznej – na podstawie liczby defektów wykrytych po wydaniu. Mając te wartości, możemy obliczyć średnią oszczędność na defekt w następujący sposób, korzystając z powyższego wzoru:

$$\text{Średnia oszczędność na defekt} = 4000 \$ - (500 \$ + 200 \$) = 3300 \$.$$

Krzywa Boehma pokazuje, że koszty usuwania defektów rosną w miarę ich późniejszego wykrycia w cyklu życia. Wynika z tego, że testy powinny być prowadzone możliwie wcześnie, aby zminimalizować koszty awarii wewnętrznych i zewnętrznych. Kierownik testów powinien dążyć do optymalnego balansowania między kosztami zapobiegania defektom a kosztami ich usunięcia.

Wysilek testowy musi opierać się na konkretnych ryzykach projektowych i produktowych oraz na ryzyku, jakie organizacja 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 w wysilek testowy proporcjonalnie do poziomów ryzyka oraz poprzez ustalanie priorytetów testów w oparciu o poziomy ryzyk, które pokrywają.

Kierownicy testów powinni rozumieć, które z tych korzyści i kosztów odnoszą się do ich organizacji, projektu bądź działania. Powinni też 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 – Parts 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 v2.0 (2023).

Sylabus ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0 (2023).

Sylabus ISTQB® Certyfikowany Tester Poziom Ekspercki Zarządzanie Testami v1.0 (2011).

Sylabus ISTQB® Certyfikowany Tester Poziom Ekspercki Wdrażanie Udoskonaleń Procesu Testowego v1.0.1 (2011).

Exam Structure and Rules v1.2 (2025), https://istqb.org/wp-content/uploads/sdm-uploads/ISTQB_Exam-Structures-and-Rules_v1.2.pdf

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.

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.

Roman, A. (2018). Testowanie i jakość oprogramowania. Modele, techniki, narzędzia. PWN.

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. (1956) Harvard Business Review, tom 34, wyd. 6, str. 93–101

Fowler, M. (2010). Hybrid development processes. IEEE Software, 27(2), 57–63.

Hadjicostas, Evsevios (2004) Total Quality Management and Cost of Quality, in: Quality Assurance in Analytical Chemistry, pp. 111–135, Springer.

Strony internetowe

www.tmmi.org Test Maturity Model integration (TMMi®); dostęp 31 stycznia 2024 r.

www.tmap.net Test Process Improvement (TPI); dostęp 31 stycznia 2024 r.

www.wikipedia.org/wiki/PDCA Plan-Do-Check-Act; dostęp 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. Dodatek A – cele nauczania i poziomy wiedzy

Poniżej przedstawiono cele nauczania obowiązujące w przypadku niniejszego sylabusu. Znajomość każdego z zagadnień poruszonych w sylabusie będzie sprawdzana na egzaminie zgodnie z przypisanym celem nauczania. Opis celu nauczania zawiera czasownik wyrażający czynność, która odpowiada właściwemu poziomowi wiedzy wymienionemu poniżej.

Poziom 1 – zapamiętać (K1): kandydat pamięta, rozpoznaje lub umie sobie przypomnieć dany termin lub dane pojęcie.

Czasownik wyrażający czynność: przypomnieć sobie, rozpoznać.

Przykłady:

„Pamiętać pojęcia związane z piramidą testów”.
„Rozpoznawać 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.

Czasownik wyrażający czynność: sklasyfikować, porównać, zestawić ze sobą, rozróżnić, wyjaśnić, podać przykłady, zinterpretować, podsumować.

Przykłady:

„Klasyfikować różne sposoby pisania kryteriów akceptacji”.
„Porównać poszczególne role występujące w testowaniu”.
„Rozróżniać ryzyka projektowe i produktowe”.
„Omówić na przykładach cel i treść planu testów”.
„Wyjaśnić wpływ wybranego modelu cyklu wytwarzania oprogramowania na testowanie”.
„Podsumować 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.

Czasownik wyrażający czynność: zastosować, wdrożyć, sporządzić, użyć.

Przykłady:

„Stosować priorytetyzację przypadków testowych”.
„Sporządzić raport o defekcie”.
„Zastosować technikę analizy wartości brzegowych, aby zaprojektować przypadki testowe”.

Poziom 4 – przeanalizować (K4): Kandydat potrafi rozdzielić informacje związane z procedurą lub techniką na poszczególne części składowe w celu lepszego zrozumienia oraz potrafi odróżnić fakty od wniosków. Typowym zastosowaniem jest analiza dokumentu, oprogramowania lub sytuacji projektowej oraz zaproponowanie odpowiednich działań w celu rozwiązania problemu lub wykonania zadania.

Czasownik wyrażający czynność: przeanalizować, zdekonstruować, nakreślić, ustalić priorytety, wybrać.

Przykłady:

„Przeanalizować daną sytuację projektową, aby określić, które czarnoskrzynkowe lub oparte na doświadczeniu techniki testowania należy zastosować, aby osiągnąć określone cele”.

„Ustalić priorytety przypadków testowych w danym zestawie testów do wykonania w oparciu o związane z nimi ryzyko produktowe”.

„Wybrać odpowiednie poziomy testów i typy testów, aby zweryfikować dany zestaw wymagań”.

Materiały dodatkowe w zakresie poziomów poznawczych celów nauczania:

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

Określone cele nauczania, które mają zastosowanie do tego sylabusu, znajdują się na początku każdego rozdziału.

6. Dodatek B – macierz powiązań między celami biznesowymi a celami nauczania

Cele nauczania	Poziom K	TM01	TM02	TM03	TM04	TM05	TM06	TM07	TM08	TM09	TM10	TM11
1. Zarządzanie czynnościami testowymi												
1.1 Proces testowy												
TM-1.1.1 Podsumować planowanie testów.	K2	X			X							
TM-1.1.2 Podsumować monitorowanie testów i nadzór nad testami.	K2	X				X						
TM-1.1.3 Podsumować ukończenie testów.	K2	X					X					
1.2 Kontekst testowania												
TM-1.2.1 Porównać powody, dla których różni interesariusze są zainteresowani testowaniem.	K2		X		X							
TM-1.2.2 Wyjaśnić, dlaczego wiedza interesariuszy jest ważna w zarządzaniu testami.	K2		X		X							
TM-1.2.3 Wyjaśnić testowanie w hybrydowym modelu wytwarzania oprogramowania.	K2		X		X							
TM-1.2.4 Podsumować czynności związane z zarządzaniem testami w różnych cyklach wytwarzania oprogramowania.	K2	X	X		X							
TM-1.2.5 Porównać czynności związane z zarządzaniem testami na różnych poziomach testów.	K2	X			X							
TM-1.2.6 Porównać czynności związane z zarządzaniem testami w różnych typach testów.	K2	X			X							
TM-1.2.7 Przeanalizować dany projekt i określić 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 Wyjaśnić różne środki podejmowane w ramach testowania opartego na ryzyku w celu reagowania na ryzyka.	K2			X								
TM-1.3.2 Podać przykłady różnych technik, które kierownik testów może wykorzystać do określenia ryzyk jakościowych produktu.	K2			X								
TM-1.3.3 Podsumować czynniki określające poziomy ryzyka związane z jakością produktu.	K2			X								
TM-1.3.4 Wybrać odpowiednie czynności testowe w celu łagodzenia ryzyk zgodnie z poziomem ryzyka w danym kontekście.	K4			X								
TM-1.3.5 Rozróżniać między przykładami ciężkich i lekkich technik testowania opartych na ryzyku.	K2			X								

Cele nauczania	Poziom K	TM01	TM02	TM03	TM04	TM05	TM06	TM07	TM08	TM09	TM10	TM11
TM-1.3.6 Podać przykłady wskaźników sukcesu i trudności związanych z testowaniem opartym na ryzyku.	K2			X								
1.4 Projektowa strategia testowa												
TM-1.4.1 Objaśnić typowe wybory dotyczące podejścia do testów.	K2				X							
TM-1.4.2 Przeanalizować organizacyjną strategię testową i kontekst projektu w celu wyboru odpowiedniego podejścia do testów.	K4				X							
TM-1.4.3 Wykorzystać metodę określania celów SMART do określenia mierzalnych celów testów i kryteriów wyjścia.	K3				X							
1.5 Doskonalenie procesu testowego												
TM-1.5.1 Objaśnić, jak wykorzystać model IDEAL do doskonalenia procesu testowego w danym projekcie.	K2									X		
TM-1.5.2 Podsumować podejścia do doskonalenia opartego na modelu w celu przetestowania usprawnień procesów i rozumieć, jak zastosować je w kontekście projektu.	K2									X		
TM-1.5.3 Podsumować analityczne podejście do usprawniania procesów i rozumieć, jak zastosować je w kontekście projektu.	K2									X		
TM-1.5.4 Wdrożyć retrospektywy projektu lub iteracji w celu oceny procesu testowego i odkrycia obszarów testowania wymagających poprawy.	K3									X		
1.6 Narzędzia testowe												
TM-1.6.1 Podsumować najlepsze praktyki w zakresie wprowadzania narzędzi.	K2										X	
TM-1.6.2 Objaśnić wpływ różnych aspektów technicznych i biznesowych przy podejmowaniu decyzji o wyborze odpowiedniego rodzaju narzędzia.	K2										X	
TM-1.6.3 Przeanalizować daną sytuację w celu opracowania planu wyboru narzędzi, obejmującego ryzyka, koszty i korzyści.	K4										X	
TM-1.6.4 Rozróżniać etapy cyklu życia narzędzi.	K2										X	
TM-1.6.5 Podać przykłady gromadzenia i oceny metryk za pomocą narzędzi.	K2									X	X	
TM-1.6.6 Podsumować najlepsze praktyki w zakresie wprowadzania narzędzi.	K2										X	
2. Zarządzanie produktem												
2.1 Metryki testów												
TM-2.1.1 Podać przykłady metryk służących osiągnięciu celów testów.	K2					X						
TM-2.1.2 Objaśnić, w jaki sposób kontrolować postęp testów przy użyciu metryk testów.	K2					X						
TM-2.1.3 Przeanalizować wyniki testów w celu tworzenia raportów z testów, które umożliwiają interesariuszom podejmowanie decyzji.	K4					X	X					

Cele nauczania	Poziom K	TM01	TM02	TM03	TM04	TM05	TM06	TM07	TM08	TM09	TM10	TM11
2.2 Szacowanie testów												
TM-2.2.1 Wyjaśnić czynniki, które należy wziąć pod uwagę przy szacowaniu testów.	K2	X							X		X	
TM-2.2.2 Podać przykłady czynników, które mogą mieć wpływ na szacowanie testów.	K2	X							X		X	
TM-2.2.3 Wybrać odpowiednią technikę lub podejście do szacowania testów w danym kontekście.	K4	X							X		X	
2.3 Zarządzanie defektami												
TM-2.3.1 Wdrożyć proces zarządzania defektami, w tym przepływ pracy dotyczący defektu, który można wykorzystać do monitorowania i kontroli defektów.	K3											X
TM-2.3.2 Objaśnić proces zarządzania defektami i zidentyfikować jego uczestników.	K2											X
TM-2.3.3 Wyjaśnić specyfikę zarządzania defektami w zwinnym wytwarzaniu oprogramowania.	K2	X										X
TM-2.3.4 Wyjaśnić wyzwania związane z zarządzaniem defektami w hybrydowym modelu wytwarzaniu oprogramowania.	K2	X										X
TM-2.3.5 Wykorzystać dane i informacje klasyfikacyjne, które należy gromadzić podczas zarządzania defektami.	K3											X
TM-2.3.6 Wyjaśnić, w jaki sposób statystyki zgłoszeń defektów można wykorzystać do opracowania udoskonalenia procesu.	K2										X	X
3. Zarządzanie zespołem												
3.1 Zespół testowy												
TM-3.1.1 Podać przykłady typowych umiejętności potrzebnych członkom zespołu testowego w czterech obszarach kompetencji.	K2							X				
TM-3.1.2 Przeanalizować dany kontekst projektu w celu określenia wymaganych umiejętności członków zespołu testowego.	K4							X				
TM-3.1.3 Objaśnić typowe techniki oceny umiejętności członków zespołu testowego.	K2							X				
TM-3.1.4 Rozróżniać między typowymi podejściami do rozwijania umiejętności członków zespołu testowego.	K2							X				
TM-3.1.5 Objaśnić umiejętności wymagane do zarządzania zespołem testowym.	K2							X				
TM-3.1.6 Podać przykłady czynników motywujących i czynników higienicznych dla członków zespołu testowego.	K2							X				
3.2 Relacje z interesariuszami												
TM-3.2.1 Podać przykłady w każdej z czterech kategorii określających koszt jakości.	K2								X			
TM-3.2.2 Stosować obliczenia kosztów i korzyści w celu oszacowania wartości dodanej testów dla interesariuszy.	K3						X		X			
RAZEM liczba celów nauczania pokrywających cel biznesowy		12	4	6	11	4	3	6	5	5	10	6

7. Dodatek C – nota wydania

Niniejsza wersja 3.0 sylabusa ISTQB® Certyfikowany Tester Poziom Zaawansowany Zarządzanie Testami jest zasadniczą aktualizacją opartą na poprzedniej wersji sylabusa z roku 2012 (o nazwie Certyfikowany Tester Poziom Zaawansowany Kierownik Testów). 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 tak, 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 jest bardziej przystępna, zrozumiała oraz łatwiejsza do przetłumaczenia, a zawarty w niej materiał jest łatwiejszy do opanowania.

W bieżącej wersji (3.0) wprowadzono następujące zmiany:

- Zmniejszono objętość całego sylabusa. Sylabus nie jest podręcznikiem, lecz dokumentem skrótowo opisującym podstawowe elementy szkolenia obowiązujące na egzaminie, w tym wskazanie, jakie tematy należy poruszyć i na jakim poziomie. W szczególności:
 - z tekstu usunięto w większości przypadków przykłady, ponieważ przygotowanie przykładów i ćwiczeń do wykorzystania w trakcie szkolenia jest zadaniem dostawcy szkoleń;
 - przestrzegano „listy kontrolnej pisania sylabusa”, która sugeruje maksymalną objętość treści celów nauczania na danym poziomie wiedzy (K2 = maksymalnie 1500 znaków bez spacji, K3 = maksymalnie 2500 znaków bez spacji, K4 = maksymalnie 3000 znaków bez spacji, +/- 20%).
- Zmniejszono liczbę celów nauczania (LO) w porównaniu z wersją z 2012 roku:
 - 36 LO K2 w porównaniu z 39 LO w CTAL TM 2012;
 - 5 LO K3 w porównaniu z 12 LO w CTAL TM 2012;
 - 7 LO K4 w porównaniu z 10 LO w CTAL TM 2012.
- Poprawiono strukturę sylabusa.
- Treści dostosowano do sylabusa ISTQB® Certyfikowany Tester Poziom Podstawowy v4.0.
- Wprowadzono istotne zmiany w dotychczasowym rozdziale 1 (Proces testowy):
 - ograniczono się do zarządzania czynnościami testowymi (planowania testów, monitorowania testów, nadzoru nad testami i ukończenia testów);
 - zintegrowano go jako sekcję w nowym rozdziale „Zarządzanie czynnościami testowymi”.
- Dodano nowy rozdział „Zarządzanie czynnościami testowymi”:
 - podrozdział 1.1 – Proces testowy: patrz powyżej;
 - podrozdział 1.2 – Kontekst testowania: rozszerzono o niesekwencyjne modele wytwarzania oprogramowania;
 - podrozdział 1.3 – Testowanie oparte na ryzyku: całkowicie przeredagowano, aby był bardziej przydatny na poziomie projektu;

- podrozdział 1.4 – Projektowa strategia testowa: plan testów jest już określony w sylabusie Certyfikowany Tester Poziom Podstawowy v4.0, dlatego skoncentrowano się na wyborze odpowiedniego podejścia testowego i sposobie definiowania mierzalnych celów testów;
- podrozdział 1.5 – Doskonalenie procesu testowego: treści te włączono do „Zarządzania czynnościami testowymi”, pokazano, jak zastosować doskonalenie w kontekście projektu i wdrażać je za pomocą retrospektyw w ramach iteracji lub projektu;
- podrozdział 1.6 – Narzędzia testowe: wprowadzanie narzędzi przeniesione z sylabusu Certyfikowany Tester Poziom Podstawowy v3.1 (ta treść nie występuje w sylabusie Certyfikowany Tester Poziom Podstawowy v4.0).
- Dodano nowy rozdział „Zarządzanie produktem”:
 - podrozdział 2.1 – Metryki testów: gromadzi wcześniejsze sekcje definiujące metryki i ich wykorzystanie;
 - podrozdział 2.2 – Szacowanie testów: sylabus Certyfikowany Tester Poziom Podstawowy v4.0 opisuje techniki szacowania; zagadnienie to rozwinęto, aby wybrać (poziom K4) odpowiednie techniki szacowania testów i stosować szacowanie testów w różnych modelach wytwarzania oprogramowania;
 - podrozdział 2.3 – Zarządzanie defektami: dostosowano do najnowszych wydań norm ISO i rozszerzono o wykorzystanie w podejściu zwinnym i w hybrydowym modelu wytwarzaniu oprogramowania.
- Dodano nowy rozdział „Zarządzanie zespołem”:
 - podrozdział 3.1 – Zespół testowy: główne tematy są takie same jak w sylabusie CTAL TM 2012; omówiono określanie indywidualnych umiejętności i tworzenie zespołów testowych;
 - podrozdział 3.2 – Relacje z interesariuszami: jest to dotychczasowy podrozdział sylabusu TM 2012 zatytułowany „Wartość biznesowa testowania”.
- Wprowadzono istotne zmiany i usunięto podrozdziały/rozdziały w sylabusie CTAL TM 2012:
 - usunięto podrozdział dotyczący testów rozproszonych, zewnętrznych i wewnętrznych;
 - usunięto podrozdział dotyczący zarządzania wdrażaniem norm branżowych;
 - usunięto rozdział dotyczący przeglądów;
 - usunięto sekcje dotyczące metod doskonalenia procesu testowego CTP i STEP;
 - usunięto sekcje dotyczące analizy testów, projektowania testów, implementacji testów i wykonywania testów.

8. Dodatek D – Słowa kluczowe specyficzne dla dziedziny

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 wysiłku 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 szacowania opierająca się na zbiorowej wiedzy członków zespołu polegająca na dokładnym szacowaniu.

9. Dodatek 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 drzewa usterek, 31	hybrydowy model wytwarzania oprogramowania, 22, 35, 58
analiza przyczyn i skutków awarii, 31	IDEAL, 37, 81
analiza przyczyny podstawowej, 39	identyfikacja ryzyka, 18, 27
analiza ryzyka, 18, 27, 29	identyfikacja ryzyka jakościowego, 27
analiza zagrożeń, 31	implementacja testów, 30
anomalia, 59	interesariusze, 68
atrybuty defektu, 58	interesariusze testów, 20
awaria, 55	ISO/IEC/IEEE 29119-2, 17
awaria wewnętrzna, 69	ISO/IEC/IEEE 29119-3, 32, 54, 59
awaria zewnętrzna, 69	iteracyjny model wytwarzania, 23
błąd szacowania, 53	komitet ds. zarządzania defektami, 56
bramka jakości, 26	kompetencje, 63
całkowity koszt jakości, 70	kontekst testowania, 19
cel testów, 18, 35	kontrola ryzyka, 27
Cel-Pytanie-Metryka, 39	koszt, 51
CMMI, 38	koszt ekspozycji, 31
cykl Deminga, 37	koszt jakości, 69
cykl życia narzędzia, 44	koszty narzędzia, 43
czynniki higieniczne, 68	kryteria wyjścia, 19
czynniki motywujące, 68	krzywa Boehma, 70
dane testowe, 35	łagodzenie ryzyka, 27
defekt, 19, 54, 55	lista kontrolna, 27
diagram przyczynowo-skutkowy, 39	macierz interesariuszy, 20
doskonalenie procesu testowego, 37, 38	metryka, 32, 39, 47, 48, 81
FMEA, 31	metryki defektów, 49
GQM, 39	metryki kosztów, 50
harmonogram testów, 18	

metryki narzędzia, 45	poziom testów, 17, 23, 33
metryki pokrycia, 50	PRAM, 31
metryki postępu testów, 50	prawdopodobieństwo ryzyka, 18, 28, 29
metryki ryzyka produktowego, 49	priorytetyzacja defektów, 58
model Tuckmana, 67	priorytetyzacja testów, 30
model wytwarzania oprogramowania, 17, 30, 34	priorytetyzacja w głąb, 30
monitorowanie ryzyka, 27	priorytetyzacja wszerek, 30
monitorowanie testów, 17, 25, 26, 30, 48	PRISMA, 31
nadzór nad testami, 17, 18, 25, 26, 30, 48	proces testowy, 17
narzędzia testowe, 41	projekt pilotażowy, 42
ocena, 66	projektowa strategia testów, 32
ocena ryzyka, 27	przeptyw pracy dotyczący defektu, 55, 78
ocena ryzyka jakościowego, 28	raport o defekcie, 55, 58
odchylenie standardowe, 53	raportowanie testów, 49
odsetek wykrytych defektów, 48	retrospektywa, 19, 27, 40
określenie ryzyka, 28	ROI, 43
organizacyjna strategia testowa, 17, 18, 34	ryzyko, 18
PDCA, 37	ryzyko jakościowe, 27, 29
plan poziomu testów, 35	ryzyko produktowe, 18, 27, 28, 48
plan testów, 17, 18, 25, 35	ryzyko projektowe, 27, 28
planowanie testów, 17, 25, 29	sekwencyjny model wytwarzania, 23
podejście "cały zespół", 40	sekwencyjny model wytwarzania oprogramowania, 55
podejście analityczne, 38	SMART, 35
podejście do testów, 18, 33	SST, 31
podstawa testów, 18	status defektu, 55
poker planistyczny, 81	sumaryczny raport z testów, 19
pokrycie kodu, 25, 48	szacowanie testów, 50
pokrycie podstawy testów, 18	średnia oszczędność na defekt, 70
pokrycie wymagań, 48	środowisko testowe, 19
polityka testów, 17	tablica wskaźników, 58
poziom ryzyka, 29	technika szacowania testów, 52

technika testowania, 20, 33	ukończenie testów, 17, 19
testalia, 17, 19	umiejętności, 63
testowanie akceptacyjne, 24	usprawnianie procesu testowego, 36
testowanie białoskrzynkowe, 25	wpływ ryzyka, 18, 28, 29
testowanie czarnoskrzynkowe, 25	wskaźnik, 39
testowanie dynamiczne, 54	wykonywanie testów, 30
testowanie funkcjonalne, 24	wykrycie, 69
testowanie integracji modułów, 23	wysiętek, 51, 71
testowanie integracji systemów, 24	zapobieganie defektom, 69
testowanie modułowe, 23	zarządzanie czynnościami testowymi, 15
testowanie нефunkcjonalne, 25	zarządzanie defektami, 54, 57
testowanie oparte na ryzyku, 26, 31	zarządzanie produktem, 46
testowanie statyczne, 54	zarządzanie ryzykiem, 18, 25
testowanie systemowe, 24	zarządzanie zespołem, 62
TMap, 54	zwinne wytwarzanie oprogramowania, 17, 58
TMMi®, 38	zwrot z inwestycji, 43
TPI Next®, 38	żądanie zmiany, 19
triaż, 56	
typ testów, 17, 24, 33	