Dokumentacja testowa

Jak ją tworzyć i jakie elementy powinna zawierać?

Dokumentacja testowa

Dokumentacja wytworzona podczas procesu testowego jest niezbędna do ewentualnego odtworzenia danego etapu testów czy przekazania wiedzy nowym członkom zespołu. Stanowi też swego rodzaju środek komunikacji – nie zawsze cały zespół projektowy przebywa w jednej siedzibie. Wszelkie ustalenia i zmiany są umieszczone we wskazanych lokalizacjach (takich jak miejsce na dysku firmowym lub dedykowane narzędzie, do którego prowadzi wskazany link). Dokumentacja powinna być stworzona według wzorców przyjętych i obowiązujących w danej organizacji.

Dokumentacja testowa - przykład

Wyobraź sobie, że przystępujesz do projektu w trakcie jego trwania jako tester bądź programista. Nikt nie ma czasu wprowadzać nowej osoby w każdy aspekt czynności, które będzie wykonywać, dlatego podstawowym źródłem wiedzy będzie wszelkiego rodzaju dokumentacja. To dzięki niej wiadomo, jak zgłaszać błędy oraz do jakich narzędzi potrzeba dostępu.

Plan testów

Plan testów

Plan testów to dokument określający strategię, która zostanie wykorzystana do weryfikacji, czy oprogramowanie zostało wytworzone zgodnie ze specyfikacją i ustalonymi wymaganiami.

Plan testów jest tworzony na etapie planowania procesu testowego. Ten dokument jest aktywny przez cały czas trwania projektu. Powinien być każdorazowo modyfikowany i uzupełniany o nowe informacje. Wszystkie osoby zaangażowane w projekt muszą mieć dostęp do dokumentów.

Plan testów - odpowiedzialność

Tego typu dokumentację zwykle tworzy lider zespołu testowego, bądź kierownik działu testów. Jednak zdarza się, że takie zadanie otrzymuje po prostu najbardziej doświadczony specjalista ds. testów. Oczywiście prace nad takim dokumentem są bardzo rozległe i będziemy potrzebować wielu szczegółowych informacji, dlatego zwykle w tworzenie takiego dokumentu jest zaangażowanych kilka osób.

Schemat przykładowego planu testów

I Cel testów

W tym punkcie należy opisać cel stworzenia dokumentu a także wskazać projekt, którego dotyczy. Celem działań testowych może być np. dostarczenie informacji dla interesariuszy, którzy na ich podstawie będą mogli świadomie podjąć decyzję o odbiorze produktu, czy wypuszczeniu / niewypuszczeniu oprogramowania na rynek.

II Przedmiot i zakres

Należy wskazać, co jest celem testów oraz wymienić elementy, które będą testowane. W tym punkcie może znaleźć się np. opis informujący o tym, że zostaną sprawdzone interfejsy i nowe funkcjonalności, procesy biznesowe.

III Terminologia

Zawiera wykaz definicji pojęć użytych przy tworzeniu dokumentów.

IV Obowiązki, odpowiedzialności uprawnienia

Zawiera opis ról wraz z zakresem obowiązków przydzielonych do osób, które biorą udział w tworzeniu specyfikacji funkcjonalnej. W ramach poszczególnych etapów testów do realizacji zadań przydzielone zostały następujące osoby:

Przydział osób do realizacji testów na poszczególnych etapach

ID	lmię i nazwisko	Rola/ odpowiedzialność/ zakres zadań
1	Anna Kowalska	Project manager - zarządzanie projektem, kontakt z dostawcą zewnętrznym oprogramowania
2	Krzysztof Nowak	Analityk biznesowy - analiza błędów, przygotowanie danych do testów
3	Jan Konieczny	Koordynator testów - zarządzanie testami, informowanie zespołu o niedostępności środowiska

V Opis postępowania

- 1. Informacje wstępne
 - a. dokumenty powiązane

Należy wskazać wszystkie dokumenty, które są związane z projektem (harmonogram, zestawienie wymagań dotyczące danych, zestawienie wymagań dotyczących środowiska itp.)

ID	Dokument	Data ostatniej modyfikacji	Wersja dokumentu	Autor dokumentu
1	Specyfikacja wymagań funkcjonalnych	01.03.2020	v2	Anna Kowalska
2	Plik z danymi do importu	07.03.2020	v3	Krzysztof Nowak
3	Harmonogram prac	09.03.2020	v4	Anna Kowalska

b. założenia i ograniczenia

Należy wymienić jakie typy testów będą przeprowadzane np. weryfikacji gotowości kodu, konfiguracja oprogramowania, migracji danych, regresji, akceptacyjne, po wdrożeniu na produkcję (smoke testy). Trzeba też opisać, jak będą realizowane zależności od innych projektów. Należy także wskazać na powiązania z komponentami, które mogą mieć wpływ na testowany projekt np. wspólne środowisko testowe.

zakres testów

Należy wskazać zakres testów zgodnie z dokumentacją projektowa i założeniami projektu.

rodzaje przeprowadzanych testów

Rodzaje przeprowadzanych testów

Rodzaj testów	Planowane	Uwagi
Smoke	TAK	Zespoły operacyjne wyznaczyły trzy osoby, które przeprowadzą testy w siedzibie firmy
Akceptacyjne	TAK	
Funkcjonalne	TAK	
Integracyjne	TAK	Należy skontaktować się z dostawcą oprogramowania i ustalić datę testów
Regresywne	TAK	
Systemowe	TAK	

c. dokumentacja realizowania testów

Potwierdzenia wyników testów zostaną umieszczone w oprogramowaniu np. Jira w postaci zrzutów ekranów ilustrujących poszczególne kroki scenariusza uzupełnionych tam, gdzie to możliwe, o identyfikator odpowiadający wykonywanemu testowi (numer zamówienia, identyfikator klienta itp. Do każdego scenariusza zakończonego ze statusem *zablokowany* zostanie dołączony e-mail lub inny dokument uzasadniający wykluczenie scenariusza z zakresu testów.

d. potrzeby szkoleniowe

Przykładowo warsztaty z architektami i kierownikami projektów. Dotyczy to wszystkich projektów w ramach ścieżki wdrożeniowej. Oprócz tego warto przeprowadzić warsztaty dla pracowników obsługi klienta, na których to warsztatach zaprezentujemy im nowe funkcje aplikacji, nowe interfejsy itp.

2. Ryzyka

Należy opisać potencjalne ryzyka:

- a. projektowe niedostępność środowiska, opóźnienie realizacji testów wynikające z nieterminowej realizacji działań zaplanowanych w harmonogramie (np. nie dostarczone na czas kod aplikacji, złej jakości dane testowe, niedostępność osób kluczowych dla testerów wynikająca z urlopów, zwolnień
- b. produktowe złej jakości kod, nieaktualna dokumentacja, luki analityczne, zmieniające się wymagania itp.
- c. bezpieczeństwo informacji wyciek danych osobowych, możliwość wejścia do panelu klienta bez podanie loginu i hasła, brak autoryzacji dostępu do danych wrażliwych.

- 3. Wymagania dotyczące danych, środowiska i narzędzi testowych
- a. wymagania dotyczące danych testowych dane klientów posiadających przynajmniej dwa aktywne produkty, dane anonimowe itp.
- b. wymagania dotyczące środowiska testowego środowisko powinno być zintegrowana z systemem głównym, baza danych musi być odświeżona na konkretny dzień, wymagana możliwość zmiany daty systemowej itp.
- c. wymagania dotyczące narzędzi testowych dostępność oprogramowania dla osób wyznaczonych do testów, generatory danych itp.

4. Harmonogram testów

Określamy (w stopniu możliwie najbardziej precyzyjnym i konkretnym) ogólny harmonogram testów, które zamierzamy przeprowadzić. Warto podzielić harmonogram na fazy oraz typy testów wraz z ich określonymi przedziałami czasowymi (datami). Pamiętajmy, aby przy określaniu terminów, dawać sobie lekki zapas czasu na mniej lub bardziej nieprzewidziane sytuacje, problemy i trudności.

Zadanie	Data rozpoczęcia	Data zakończenia	Odpowiedzialny	
Testy akceptacyjne	10.03.20200	20.03.2020	Jan Konieczny - koordynator testów	
Smoke testy	22.03.2020	22.03.2020	Anna Kowalska - project manager	

4. Raportowanie i komunikacja

a. produkty dostarczane w wyniku testów (testalia)

Nazwa produktu	Osoba odpowiedzialna		
Raport z testów akceptacyjnych	Jan Konieczny - koordynator testów		
Raport defektów	Jan Konieczny - koordynator testów		

b. plan raportowania i komunikacji

Należy opisać sposób raportowania oraz ustalony proces komunikacji np. realizacja testów będzie na bieżąco raportowania w narzędziu Jira

c. zespół projektowy

ID	lmię i Nazwisko	Rola	Sposób komunikacji
1	Anna Kowalska	Project Manager	poczta e-mail, MS teams, cykliczne spotkania - środa 10:00
2	Krzysztof Nowak	Analityk biznesowy	poczta e-mail, MS teams, cykliczne spotkania - środa 10:00
3	Jan Konieczny	Koordynator testów	poczta e-mail, MS teams, cykliczne spotkania - środa 10:00
4	Wojciech Marzec	Programista	poczta e-mail, MS teams, cykliczne spotkania - środa 10:00

6. Załączniki

Scenariusze testowe

Scenariusze testowe

W dobie ciągłego monitorowania wykonywanej pracy oraz pomiaru jej jakości scenariusze testowe są pewnego rodzaju dowodem, że podczas testów system zachował się w określony sposób. Taka wiedza jest niezbędna dla programistów, testerów i klientów.

Scenariusz testowy – zbiór przypadków testowych, czyli kroków służących do sprawdzenia danej funkcjonalności systemu.

Scenariusze testowe - przykład

Funkcjonalność: logowanie do systemu

W tej funkcjonalności należy utworzyć osobne przypadki dla poprawnego i niepoprawnego logowania (tzn. przy użyciu poprawnych i błędnych danych).

Logowanie przy użyciu błędnych danych

Logowanie przy użyciu poprawnych danych



Scenariusz testowy

Scenariusze testowe

Scenariusz testowe mogą być wykonywane przez osoby, które nie miały styczności z pracą w dziale IT. Podczas testów akceptacyjnych często o wykonanie pewnych czynności testowych są proszone osoby pracujące w działach operacyjnych (np. obsługa klienta, sprzedaż), które nie mają wiedzy o projekcie ani umiejętności technicznych. Im dokładniejszy scenariusz, tym mniej pytań od takich osób. Jeżeli testy będą wykonywane przez osoby z wewnątrz organizacji, możemy sobie pozwolić na przypadki wysokiego poziomu. Zaoszczędzimy wtedy czas na szczegółowym rozpisywaniu każdego z kroków.

Nazwa	Przypadek niskiego poziomu	Przypadek wysokiego poziomu
Tytuł	Logowanie przy użyciu poprawnych danych	Poprawne logowanie
Cel testu	Celem testu jest sprawdzenie poprawności działania mechanizmu logowania	Weryfikacja mechanizmu logowania
Warunki wstępne	 Podano prawidłowy login i hasło dla użytkownika testowego Aplikacja została uruchomiona i znajdujemy się na ekranie użytkownika 	 Podany jest prawidłowy login i hasło dla użytkownika testowego Aplikacja została uruchomiona i znajdujemy się na ekranie użytkownika

Nazwa	Przypadek niskiego poziomu	Przypadek wysokiego poziomu
Kroki do wykonania	 Wprowadź niepoprawny login i prawidłowe hasło, po czym kliknij <i>Zaloguj</i> Wprowadź prawidłowy login i nie wprowadzaj hasła, po czym kliknij <i>Zaloguj</i> Wprowadź prawidłowy login i nieprawidłowe hasło, po czym kliknij <i>Zaloguj</i> Wprowadź prawidłowy login i prawidłowe hasło, Wprowadź prawidłowy login i nieprawidłowe hasło, po czym kliknij <i>Zaloguj</i> 	Zaloguj się do systemu (zastosuj różne warianty prawidłowych i nieprawidłowych danych)

Nazwa	Przypadek niskiego poziomu	Przypadek wysokiego poziomu
Oczekiwany wynik	 Pojawia się komunikat o wprowadzonym nieistniejącym loginie, wyświetlana jest prośba o ponowne wpisanie loginu Pojawia się komunikat informujący o brakujących danych, po jego wyświetlaniu powinno nastąpić przejście do pola, w których wprowadza się hasło Pojawia się komunikat informujący o podaniu nieprawidłowego hasła, zostaje także wyświetlona prośba o ponowne wpisanie hasła Użytkownik zostaje pomyślnie zalogowany 	 Błędny login, logowanie nie powiodło się Logowanie nie powiodło się, wprowadź hasło Błędne hasło, logowanie nie powiodło się Następuje prawidłowe zalogowanie do aplikacji

Nazwa	Przypadek niskiego poziomu	Przypadek wysokiego poziomu
Środowisko	Próby logowania należy wykonać w:	Próby należy wykonać w środowiskach dedykowanych dla projektu

Warto zwrócić uwagę na to że, przypadki wysokiego poziomu są bardziej ogólne i nie wymieniają szczegółowo wszystkich sytuacji, jakie mogą nastąpić. Przypadki te realizują osoby, które posiadają niezbędną wiedzę i są świadome czynności, które doprowadzą do pomyślnego ukończenia testów.

Lista kontrolna

Lista kontrolna

Lista kontrolna (ang. checklist) to narzędzie umożliwiające kontrolę poprawności i/lub stopień ukończenia danego przedsięwzięcia. Lista składa się z serii pytań lub zagadnień dotyczących projektu. Skuteczność listy zależy od jej złożoności. Im bardziej szczegółowe są zagadnienia lub pytania, tym większa jest skuteczność. Czas poświęcony na sporządzenie szczegółowej listy kontrolnej może uchronić przed zbędnymi i czasochłonnymi pracami na późniejszym etapie projektu.

Lista kontrolna - przykład

Rejestracja konta użytkownika:

- utworzenie konta
- walidacja obligatoryjności pól
- zgody marketingowe
- autoryzacja SMS
- autoryzacja e-mail
- logowanie przy użyciu błędnych danych

- logowanie przy użyciu poprawnych danych
- edycja danych
- odzyskiwanie hasła
- dodanie karty płatniczej
- wysyłanie wiadomości e-mail
- usuwanie konta

Rejestr ryzyk

Rejestr ryzyk

Rejestr ryzyk jest dokumentem wytwarzanym we wczesnym stadium projektu, zazwyczaj dostarczają go osoby odpowiedzialne za zarządzanie projektem np. kierownicy.

Rejestr ma głównie na celu zwrócenie uwagi wszystkich zaangażowanych osób na czynniki, które mogą być zagrożeniem dla terminowej realizacji projektu, jak również dla jakości wytwarzanego produktu. Dokument pozwala przygotować się na potencjalne ryzyka oraz wcześnie na nie reagować.

Rejestr ryzyk - przykład

ID	Autor	Opis	Data zgłoszenia	Kategoria ryzyka	Wpływ ryzyka	Priorytet	Podjęte działania
1.	Jan Kowalski	Nieaktualna specyfikacja wymagań	10.03.2020	produktowe	duży	wysoki	Spotkanie z pomysłodawcami i ustalenie odpowiedzialności
2.	Anna Nowak	Realizacja projektu w sezonie urlopowym	05.03.2020	projektowe	duży	wysoki	Wyznaczenie obszarów, w których należy zapewnić dostępność pracowników

Raport błędów

Raport błędów

W czasie testów zgłaszanych jest wiele błędów - niektóre z nich są naprawiane od razu, inne zostają przełożone do realizacji w kolejnych wersjach programu. Raport błędów jest narzędziem o charakterze informacyjnym - pomaga ocenić jakość wytwarzanego rozwiązania i czas naprawy przez zespół programistów.

Taki raport powinien zawierać niezbędne informacje, które dadzą zainteresowanym pełny obraz aktualnej sytuacji, bez konieczności logowania się do narzędzia zarządzającego błędami.

Raport błędów

ID	Status	Aplikacja	Priorytet	Komponent	Data zgłoszenia	Osoba przypisana	Planowana data naprawy
ID-10 Błąd przy próbie logowania	otwarty	XYZ	wysoki	222	04.03.2020	Wojciech Marzec	???
ID-23 Błąd przy próbie uruchomienia aplikacji	do wyjaśnienia	XYZ	wysoki	222	05.03.2020	Adam Kwiecień	???
ID-25 Błąd w nazwie pola "Metoda płatności"	otwarty	XYZ	średni	111	05.03.202	Wojciech Marzec	???

1. Cel dokumentu

Należy krótko opisać cel raportu końcowego, np. informacje o statusie wykonanych prac, jakości wytworzonego oprogramowania, rekomendacje do wdrożenia.

2. Przedmiot i zakres dokumentu

Należy napisać dla jakich testów i jakiego projektu dotyczy raport.

Dokument obejmuje:

- podsumowanie wykonania scenariuszy testowych np. jaki tester ile miał przydzielonych scenariuszy i ile realnie wykonał
- podsumowanie zgłoszonych błędów, najlepiej w postaci tabelarycznej prezentując ich liczbę z podziałem na priorytety i statusy
- wyniki testów, czyli jakie scenariusze zostały wykonane z wynikiem pozytywnym / negatywnym / zablokowanym
- zakres testów należy podać wszystkie sprawdzone funkcjonalności, czyli co realnie zweryfikowano

- 3. Terminologia
- Zawiera wykaz skrótów i definicję używanych pojęć.
- 4. Obowiązki, odpowiedzialność i uprawnienia Zawiera opis obowiązków przypisanych do stanowisk, które biorą udział w tworzeniu raportu końcowe z testów
- 5. Opis postępowania
- dokumenty powiązane z raportem końcowym z testów np. plan testów, dokumentacja projektowa.

- podsumowanie wyników testów np. statystyki realizacji, zgłoszone błędy
 Warto wskazać funkcjonalności, które były najbardziej podatne na defekty.
 Przydatną informacją jest również średni czas realizacji zgłoszenia z podziałem na prorytety.
 - a) zespół realizujący testy (z odniesieniem do planu testów)
 - b) czynniki blokujące postęp testów np. niedostępność środowiska, brak zasobów ludzkich, braki sprzętu o określonych parametrach
 - c) rekomendacja dotycząca wdrożenia (na podstawie testów akceptacyjnych)

- szczegółowe wyniki testów
 Szczególną uwagę należy zwrócić na listę otwartych defektów i nierozwiązanych problemów, ocenę kryteriów zakończenia testów oraz ryzyka.
- 6. Wytworzone raporty i załączniki