

JAK PISAĆ SCENARIUSZE TESTOWE?

Scenariusze testów funkcjonalnych, bo o nich mowa, są podstawowym narzędziem weryfikacji czy oprogramowanie działa prawidłowo i zgodnie ze specyfikacją. Oprócz tego scenariusze mogą i powinny być używane w czasie utrzymania aplikacji – do przeprowadzania testów regresyjnych.

Scenariusze testowe swoją formą przypominają rozpisane do postaci scenariuszy przypadki użycia aplikacji. Tak jak one mogą powstawać na etapie analitycznym (przed wdrożeniem) – wtedy stanowią uzupełnienie dokumentacji. Równie dobrze mogą powstawać przy samym końcu wdrożenia, opisywane już na bazie funkcjonującej wersji testowej systemu. Scenariusze zwykle są bardziej szczegółowe i „gęstsze” od przypadków użycia systemu, ponieważ ich zadaniem jest przygotowanie do weryfikacji wszystkich funkcji dostępnych w aplikacji. Dodatkowo scenariusze powinny obejmować różne warianty nawigacji i dostępu do funkcji.

Tak jak w każdym aspekcie – należy zachować zdrowy rozsądek przy tworzeniu materiałów do testów. Spisywanie wszystkich ścieżek zajmuje sporo czasu. Wydaje się być dobrym rozwiązaniem spisywanie scenariuszy na podstawie makiet interfejsu użytkownika i uwzględnianie tych elementów które są na nich wyróżnione.

Zakres testów

Bezwzględnie na samym początku musimy opisać jaki będzie zakres naszych testów. Powinniśmy wymienić wszystkie funkcje i przypadku użycia jakie będą testowane. Jeśli w etapie analitycznym nie powstała identyfikacja przypadków użycia to możemy dokonać jej dokonać w tym momencie – na podstawie projektu interfejsu użytkownika lub na podstawie działającej aplikacji. Ważne jest, aby granulacja była dość szczegółowa – ponieważ testy będą wtedy dokładniejsze. Należy jednak pamiętać, że każdy test będzie musiał być opisany a później też testowany. Nie ma sensu opisywać testów elementów które są statyczne lub nie wykonują żadnej logiki. Być może wystarczy zawrzeć kontrolę takich elementów w jednym scenariuszu.

Przykładowy zakres testów (wycinek) może wyglądać następująco:

- *SPU 1.0 – Rejestracja i logowanie*
- *SPU 2.0 – Strona Główna użytkownik zalogowany*
- *SPU 2.2 – Katalog towarów – Kategorie*
- *SPU 4.0 – Katalog towarów – lista towarów*
- *SPU 4.1 – Katalog towarów – lista bez zdjęć*
- *SPU 4.2 – Katalog towarów – Wyprzedaż*
- *SPU 4.3 – Bloczek towaru niedostępnego*
- *SPU 5.0 – Karta towaru*

Założenia

Oprócz listy funkcji które będą testowane koniecznie musimy wymienić nasze założenia, tzn:

- Adres wersji testowej gdzie testy będą przeprowadzone aby nie było wątpliwości w przypadku konieczności odtworzenia błędu,
- Wersję oprogramowania która jest testowana,
- Oprogramowanie klienckie (przeglądarka, wersja, system operacyjny) na jakim są wykonywane testy,

- Miejsce gdzie zostaną zgłoszone błędy (np. adres systemu ticketowego),
- Wymagania dla testera co do zgłoszenia błędu – ważne aby testerzy podawali co najmniej: adres wystąpienia błędu i numer scenariusza, kroki potrzebne do odtworzenia błędu, kategorię błędu (wg. umowy – błąd krytyczny, błąd normalny, błąd niski...) – ważne jest przypomnienie w tym miejscu jakie kryteria muszą spełniać błędy odpowiednich kategorii.

Dodatkowo konieczne jest opisanie ról jakimi powinni posługiwać się testerzy –jeśli aplikacja przewiduje różne poziomy dostępu.

Jak może wyglądać scenariusz?

Scenariusze swoją formą przypominają przypadki użycia. Jeśli takowe były tworzone w etapie analizy warto posiłkować się nimi jako bazą do utworzenia scenariuszy testowych.

Przypatrzmy się jak mogą wyglądać scenariusze testowe dotyczące ścieżki zakupowej w sklepie internetowym. Opisane zostały dwa przykładowe scenariusze. Sama ścieżka zakupowa składa się w tym przypadku z ok 10 scenariuszy testowych obejmujących kroki takie jak wybór płatności, podanie adresu dostawy (z błędami i poprawnego), realizacja płatności.

Przykład 1

P1	Wybór adresu dostawy
Scenariusz dotyczy	SPU 8.3
Cel testu	Testowanie poprawnego ustawiania adresu wysyłki użytkownika z odbiorem osobistym.
Sposób dostępu	Widok wywoływany z poziomu kroku drugiego przyciskiem „Zaloguj”. Użytkownik powinien mieć przydzielony limit kupiecki.
Scenariusz (kroki testowe)	
Akcje użytkownika	Odpowiedź systemu
1. Wybranie adresu zdefiniowanego wcześniej 3. Wybrać płatność (dowolna) 5. Zaznaczenie checkboxa „Określ dokładną datę odbioru”. 7. Wybór daty 9. Wybór godzin nocnych 11. Kliknąć przycisk „Dalej – wybór płatności”. 13. Wybrać płatności przy odbiorze i kliknięcie przycisku Dalej do podsumowania 15. Cofnięcie się do kroku Dostawy za pomocą przycisków wstecz i zmienić godziny odbioru na godziny nie nocne. Następnie przejście do podsumowania.	2. Wyświetlenie adresu 4. – 6. Rozwinięcie formularza z polem, do którego można wpisać datę odbioru oraz godziną. Użytkownik może wybrać datę dostarczenia towaru nie dłuższą niż 5 dni od momentu składania zamówienia poprzez użycie formularza wyboru daty wraz z kalendarzem. Dni, w które nie działa firma kurierska są zablokowane (nie można ich wybrać) z poziomu pojawiającego się tutaj kalendarza. Wyświetlany jest także komunikat o tym, że odbiór w godzinach nocnych będzie nałożony dodatkową opłatą. 8. – 10. – 12. Wyświetlenie kolejnego kroku – „Forma płatności”. 14. Wyświetlenie podsumowania gdzie została dodana opłata za usługę odbioru w godzinach nocnych 16. Wyświetlenie podsumowania bez opłaty za usługę odbioru w godzinach nocnych
Ocena testu	

Przykład 2

P2	Niepowodzenie płatności
Scenariusz dotyczy	SPU 8.4
Cel testu	Testowanie niepowodzenia metody płatności.

Sposób dostępu	Płatność nie została dostarczona. Sklep dostaje status o nieopłaconym zamówieniu.
Scenariusz (kroki testowe)	
Akcje użytkownika	Odpowiedź systemu
1 –	2. Aplikacja winna zapisać zamówienie ze statusem Zamówienie nieopłacone, zaprezentować zamówienie z przyciskiem „Powrót do płatności” który wywoła krok 4 ścieżki zamówienia z opcja wyboru płatności.
Ocena testu	

Opisane przypadki dotyczą ścieżki, w której oczekujemy prawidłowego rezultatu, ale także przypadku, w którym czekamy na błąd wygenerowany przez system obsługi płatności. Taka dokładna granulacja scenariuszy nie tylko daje nam dokładne wyniki testów ale w czasie rozwoju pozwala precyzyjnie testować tylko zmienione elementy.

Jak widzimy każdy scenariusz składa się z następujących elementów:

- **Scenariusz dotyczy** – wymieniony jest przypadek użycia zaprojektowany w czasie analizy. Dzięki odniesieniu wiemy na podstawie czego przypadek został stworzony (że nie jest wysany z palca) a także możemy w razie nieścisłości zawsze odnieść się do dokumentu analitycznego. Jeśli nie posiadamy opisu przypadków użycia należy podać jaka makieta lub który widok graficzny stał się podstawą do stworzenia opisu,
- **Sposób dostępu** – opisuje jak rozpocząć scenariusz testowy. W pierwszym przykładzie podana została droga w nawigacji sklepu internetowego. Drugi przypadek obejmuje błąd systemu płatności, który jest inicjatorem zdarzenia.
- **Scenariusz (kroki testowe)** – to najważniejszy element opisu scenariusza testowego. Kroki są podzielone na akcje użytkownika oraz na odpowiedzi systemu. Scenariusz przypomina budowę scenariusz filmowy w którym są uwzględnione wszystkie „kwestie” użytkownika oraz odpowiedzi systemu. Opis w formie dialogu w przejrzysty sposób pokazuje interakcje testera z systemem. Niepowodzenie testu może zostać momentalnie stwierdzone w momencie nieodpowiedniej odpowiedzi systemu lub niemożliwości wykonania akcji przez użytkownika
- **Ocena testu** – to pole pozostaje puste, do wypełnienia podczas przeprowadzania testów przez osobę odpowiedzialną.

Inne scenariusze testowe

Oprócz scenariuszy funkcjonalnych warto też spisać scenariusze dotyczące integracji z innymi systemami. Ich forma będzie inna, ponieważ często nie mamy do czynienia w tym przypadku z jakimkolwiek działaniem użytkownika (integracje automatyczne) oraz nie występuje interfejs użytkownika. Koniecznie należy zdefiniować jakie dane wejściowe zostaną wysłane, pod jaki adres, w jakim formacie. Następnie należy opisać w krokach wymagane działanie systemu z którym następuje integracja oraz warianty odpowiedzi wyjściowej.

Warto opisać jakie będzie działanie aplikacji w przypadku wystąpienia błędów, gdzie zostaną zalogowane i wyświetlone komunikaty błędów a także jaki będzie format wiadomości z informacjami o błędach.

W ten sposób wyczerpiemy większość przypadków jakie mogą wystąpić podczas eksploatacji systemu i damy sobie czas na zaprowadzenie odpowiedniej ich obsługi.