

Specyfikacja wymagań

Software Requirements Specification

Wersja

Wersja	Data	Autor	Zakres zmian
0.1.0	2022-10-31	Artur Karczmarczyk	Inicjalizacja szablonu dokumentu

Spis treści

Wersja.....	1
Spis treści.....	1
1. Wprowadzenie.....	2
1.1 Cel	2
1.2 Przyjęte konwencje	2
1.3 Zakres projektu	2
1.4 Odwołania.....	2
2. Opis ogólny	2
2.1 Perspektywa produktu.....	2
2.2 Klasy i charakterystyki użytkowników.....	2
2.3 Środowisko działania	3
2.4 Ograniczenia projektowe i wykonawcze	3
2.5 Założenia i zależności	3
3. Funkcjonalności systemu	3
3.x Funkcjonalność XYZ.....	3
3.x.1 Opis	3
3.x.2 Wymagania funkcjonalne	3
4. Wymagania dotyczące danych.....	3
4.1 Logiczny model danych	3
4.2 Raporty	3
4.3 Pozyskiwanie, integralność przechowywanie i usuwanie danych	4
5. Wymagania dotyczące interfejsu zewnętrznego.....	4
5.1 Interfejsy użytkownika	4
5.2 Interfejsy programowe	4
5.3 Interfejsy sprzętowe	4
5.4 Interfejsy komunikacyjne	4
6. Cechy jakości.....	5
6.1 Użyteczność	5
6.2 Wydajność	5
6.3 Zabezpieczenia.....	5

6.4 Bezpieczeństwo użytkowania	5
6.x Inne	5
7. Wymagania internacjonalizacji i lokalizacji	5
8. Inne wymagania	5
Suplement A: Słownictwo	5
Suplement B: Modele analityczne	5

1. Wprowadzenie

We wprowadzeniu znajduje się przegląd, który ma pomóc czytelnikowi zrozumieć, jak zorganizowany jest ten dokument i jak z niego korzystać.

1.1 Cel

Zidentyfikuj produkt lub aplikację, których wymagania są określone w tym dokumencie, w tym numer wersji lub wydania.

Jeśli ten SRS dotyczy tylko części złożonego systemu, zidentyfikuj tę część lub podsystem.

Opisz odbiorców, dla których ten dokument jest przeznaczony (programiści, kierownicy projektów, marketing, użytkownicy, testerzy, dział wdrożeń itp.)

1.2 Przyjęte konwencje

Opisz tu przyjęte standardy i konwencje, znaczenie formatowania.

1.3 Zakres projektu

Podaj krótki opis planowanego oprogramowania i jego przeznaczenie. Powiąż oprogramowanie z celami użytkownika lub firmy oraz celami biznesowymi.

Jeśli dokument wizji / zakresu istnieje osobno, należy go tutaj przywołać, zamiast powielać jego treść.

W przypadku projektów iteracyjnych, SRS powinien zawierać własne określenie zakresu jako podzbiór długoterminowej strategii rozwoju produktu.

Można podać ogólne podsumowanie głównych funkcji produktu.

1.4 Odwołania

Odwołania do wszystkich przydatnych dokumentów, do których odwołuje się SRS.

2. Opis ogólny

Ta sekcja przedstawia ogólny przegląd produktu i środowiska, w którym będzie używany, przewidywanych użytkowników oraz znanych ograniczeń, założeń i zależności

2.1 Perspektywa produktu

Opisz kontekst produktu. Czy to nowy produkt, czy kolejny fragment większego systemu? W drugim przypadku, należy określić, w jaki sposób oprogramowanie odnosi się do całego systemu i zidentyfikować główne interfejsy. Należy rozważyć użycie diagramów wizualnych do ilustracji tych zależności.

2.2 Klasy i charakterystyki użytkowników

Należy zidentyfikować i scharakteryzować przewidywane klasy użytkowników końcowych tego produktu.

Opisy klas użytkowników mogą być używane wielokrotnie. Jeśli dostępny jest katalog głównych klas użytkowników, można zamiast powielać treści wstawić tutaj odwołanie do zewnętrznych dokumentów.

2.3 Środowisko działania

Należy opisać środowisko, w którym będzie działać oprogramowanie, w tym platformę sprzętową, systemy operacyjne, geograficzne lokalizacje użytkowników, serwerów i baz danych (NA, EU, APAC) itp.

Należy wymienić wszelkie inne składniki oprogramowania lub aplikacje, z którymi system musi współpracować / współistnieć.

2.4 Ograniczenia projektowe i wykonawcze

W niektórych przypadkach język programowania lub biblioteki do wykorzystania są z góry określone. Należy opisać wszelkie czynniki, które ograniczą opcje dostępne dla programistów, oraz uzasadnienie każdego ograniczenia.

2.5 Założenia i zależności

Należy wymienić tutaj założenia dotyczące funkcjonalności systemu. Ponadto należy zidentyfikować wszelkie zależności budowanego produktu od czynników zewnętrznych lub komponentów pozostających poza jego kontrolą.

Przykładowo jeśli przed uruchomieniem produktu należy zainstalować system kontenerów Docker lub serwer web NGINX / Apache, należy taką zależność wskazać w tej sekcji.

3. Funkcjonalności systemu

Wymagania funkcjonalne można zorganizować według obszaru funkcjonalnego, przebiegu procesu, przypadku użycia, trybu działania, klasy użytkownika itp. Możliwe są również hierarchiczne kombinacje tych elementów, takie jak przypadki użycia w klasach użytkowników. Należy wybrać taką metodę organizacji, która ułatwi czytelnikom zrozumienie zamierzonych możliwości produktu.

3.x Funkcjonalność XYZ

Należy podać nazwę funkcji w kilku słowach, na przykład „3.1 Zarządzanie produktami”.

3.x.1 Opis

Należy podać krótki opis funkcjonalności i wskazać, czy ma ona wysoki, średni czy niski priorytet.

3.x.2 Wymagania funkcjonalne

Należy wymienić szczegółowe wymagania funkcjonalne związane z tą funkcjonalnością. Są to możliwości oprogramowania wymagane, aby użytkownik mógł wykonać dany przypadek użycia. Opisać, jak produkt powinien reagować na przewidywane błędy oraz nieprawidłowe dane wejściowe i zachowania. Każde wymaganie funkcjonalne powinno zostać unikalnie oznaczone.

4. Wymagania dotyczące danych

W tej sekcji szablonu należy opisać różne aspekty danych, które system będzie wykorzystywał jako dane wejściowe, przetwarzał w jakiś sposób lub produkował jako dane wyjściowe.

4.1 Logiczny model danych

Model danych jest wizualną reprezentacją obiektów danych i zbiorów, które system będzie przetwarzał, oraz relacji między nimi. Istnieje wiele notacji do modelowania danych, w tym diagramy ERD. To nie to samo, co implementacyjny model danych, który zostanie zrealizowany w postaci projektu bazy danych.

4.2 Raporty

Jeśli system będzie generować raporty, należy je tutaj zidentyfikować i opisać ich cechy. Jeśli raport musi być zgodny z określonym, wstępnie zdefiniowanym układem, można określić to tutaj jako ograniczenie lub podać przykład. Alternatywnie, można opisać zawartości raportu, kolejności sortowania, sumowaniu poziomów itd.

4.3 Pozyskiwanie, integralność przechowywanie i usuwanie danych

W stosownych przypadkach opisz, w jaki sposób dane są gromadzone i utrzymywane. Określić wszelkie wymagania dotyczące potrzeby ochrony integralności danych systemu. Zidentyfikować niezbędne techniki, takie jak kopie zapasowe, punkty kontrolne, kopie lustrzane.

5. Wymagania dotyczące interfejsu zewnętrznego

Ta sekcja zawiera informacje zapewniające prawidłową komunikację systemu z użytkownikami oraz z zewnętrznymi elementami sprzętu lub oprogramowania. Złożony system z wieloma mikroserwisami powinien tworzyć oddzielną specyfikację interfejsu lub specyfikację architektury systemu. Dokumentacja interfejsu może zawierać materiały z innych dokumentów poprzez odwołania.

5.1 Interfejsy użytkownika

Opisz logiczną charakterystykę każdego interfejsu użytkownika, którego potrzebuje system. Niektóre specyficzne cechy interfejsów użytkownika mogą pojawić się w 6.1 Użyteczność. Należy zwrócić uwagę na:

- Odniesienia do standardów interfejsu użytkownika, których należy przestrzegać
- Standardy dotyczące czcionek, ikon, etykiet przycisków, obrazów, schematów kolorów, sekwencji zakładek w polach, powszechnie używanych elementów sterujących, grafiki marki, informacji o prawach autorskich i prywatności itp.
- Ograniczenia rozmiaru ekranu, układu lub rozdzielczości
- Standardowe przyciski, funkcje lub łącza nawigacyjne, które pojawią się na każdym ekranie, na przykład przycisk pomocy
- Skróty klawiszowe
- Wyświetlanie wiadomości i konwencje budowy / składni komunikatów
- Wytyczne dotyczące walidacji danych (takie jak ograniczenia wartości wejściowych i kiedy sprawdzać zawartość pól)
- Standardy układu ułatwiające lokalizację (tłumaczenie) oprogramowania
- Udogodnienia dla użytkowników niedowidzących, daltonistów lub mających inne ograniczenia

5.2 Interfejsy programowe

Opisz połączenia między tym produktem a innymi składnikami oprogramowania (zidentyfikowanymi według nazwy i wersji), w tym innymi aplikacjami, bazami danych, systemami operacyjnymi, narzędziami, bibliotekami, witrynami internetowymi i zintegrowanymi komponentami komercyjnymi.

Określić cel, formaty i treść komunikatów, danych i wartości kontrolnych wymienianych między komponentami oprogramowania.

Określ odwzorowania danych wejściowych i wyjściowych między systemami oraz wszelkie tłumaczenia, które muszą zostać wykonane, aby dane mogły zostać przesłane z jednego systemu do drugiego.

5.3 Interfejsy sprzętowe

Opisz charakterystykę każdego interfejsu między komponentami oprogramowania i komponentami sprzętowymi systemu, jeśli takie istnieją. Opis ten może obejmować obsługiwane typy urządzeń, dane i interakcje sterowania między oprogramowaniem a sprzętem oraz używane protokoły komunikacyjne. Dane wejściowe i wyjściowe, ich formaty, prawidłowe wartości lub zakresy oraz wszelkie problemy z synchronizacją, o których programiści powinni wiedzieć.

5.4 Interfejsy komunikacyjne

Określić wymagania dotyczące wszelkich funkcji komunikacyjnych, których będzie używać produkt, w tym poczty e-mail, przeglądarki internetowej, protokołów sieciowych i formularzy elektronicznych.

Zdefiniować odpowiednie formaty komunikatów. Określ zasady bezpieczeństwa i szyfrowania komunikacji, szybkości przesyłania danych, mechanizmy synchronizacji.

6. Cechy jakości

W tej sekcji określa się wymagania niefunkcjonalne inne niż ograniczenia, które są zapisane w sekcji 2.4 oraz 5.1.

Te wymagania jakościowe powinny być szczegółowe, ilościowe i weryfikowalne.

Należy wskazać względne priorytety różnych atrybutów, np. bezpieczeństwo ponad wydajnością.

6.1 Użyteczność

Wymagania dotyczące użyteczności dotyczą łatwości uczenia się, łatwości użytkowania, wydajności interakcji i dostępności. Określone tutaj wymagania dotyczące użyteczności pomogą projektantowi interfejsu użytkownika stworzyć optymalne user experience.

6.2 Wydajność

Określić wymagania dotyczące wydajności dla różnych operacji systemowych. Jeśli różne wymagania funkcjonalne lub cechy mają różne wymagania dotyczące wydajności, lepiej jest określić te cele wydajnościowe razem z odpowiednimi wymaganiami funkcjonalnymi, zamiast zbierać je w tej sekcji.

6.3 Zabezpieczenia

Określ wszelkie wymagania dotyczące kwestii bezpieczeństwa lub prywatności. Mogą one odnosić się do bezpieczeństwa fizycznego, danych lub oprogramowania.

6.4 Bezpieczeństwo użytkowania

Określ wymagania dotyczące możliwych szkód, które mogą wynikać z użytkowania produktu. Zdefiniuj wszelkie zabezpieczenia lub działania, które należy podjąć, a także potencjalnie niebezpieczne działania, którym należy zapobiec.

6.x Inne

Utwórz osobną sekcję w SRS dla każdego dodatkowego atrybutu jakości produktu, aby opisać cechy, które będą ważne zarówno dla klientów, jak i dla programistów.

7. Wymagania internacjonalizacji i lokalizacji

Wymagania dotyczące umiędzynarodowienia i lokalizacji zapewniają, że produkt będzie odpowiedni do użytku w innych krajach, kulturach i lokalizacjach geograficznych niż te, w których został stworzony. Przykładowo różnice walutowe; formatowanie dat, numerów, adresów czy numerów telefonów. Wymagania dotyczące umiędzynarodowienia i lokalizacji mogą być ponownie wykorzystywane w różnych projektach.

8. Inne wymagania

Opcjonalnie można zdefiniować dowolne inne wymagania nie zdefiniowane wcześniej w tym SRS. Przykładowo wymagania prawne, finansowe; wymagania instalacji produktu, jego konfiguracji, uruchomienia czy wyłączenia.

Suplement A: Słownictwo

Należy zdefiniować tutaj wszelkie skróty i specjalistyczne słownictwo używane w tym dokumencie.

Suplement B: Modele analityczne

W tej opcjonalnej sekcji należy umieścić wszelkie modele analityczne które pomogą czytelnikowi zrozumieć system. Przykładowo drzewa funkcjonalności, diagramy ERD, procesy BPMN. Czasem czytelniejsze może być umieszczenie tych diagramów bezpośrednio w treści SRSa.