# **Załącznik nr 5 do OPZ**

## **Wymagania Dotyczące Dokumentacji Systemu**

Wymagania dotyczące zawartości Dokumentacji - specyfikacja zakresu informacyjnego Dokumentacji Aplikacji.

Podręcznik użytkownika zewnętrznego (PUZ)

Wymagania:

Dokumentacja powinna zostać uzupełniona elementy identyfikujące dokument,

Dokumentacja powinna zostać uzupełniona o zasady świadczenia wsparcia technicznego dla tej grupy użytkowników

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Identyfikator dokumentu | Informacje na temat: wersji dokumentu, historii zmian, nazwy i wersji Systemu Aplikacyjnego, do którego się odnosi. |
| Opis ogólny | Ogólnie opis do czego służą Aplikacje, zasady nawigacji pomiędzy poszczególnymi komponentami Aplikacji oraz generalne zasady współpracy z aplikacją oraz zasady świadczenia wsparcia technicznego. |
| Opis funkcji | Opis ról i ich uprawnień dla tej klasy użytkowników, opis funkcjonalności oraz interfejsu użytkownika dla tej klasy użytkowników, zasady walidacji pól, opis raportów i zestawień. |
| Zasady komunikacji | Opis komunikacji za pomocą e-mail; |

Podręcznik użytkownika wewnętrznego (PUW)

Wymagania:

Dokumentacja powinna zostać uzupełniona o elementy identyfikujące dokument.

Dokumentacja powinna zostać uzupełniona o zasady świadczenia wsparcia technicznego dla tej grupy użytkowników.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Identyfikator dokumentu | Informacje na temat: wersji dokumentu, historii zmian, nazwy i wersja Systemu Aplikacyjnego, do którego się odnosi. |
| Opis ogólny | Ogólnie opis do czego służą Aplikacje, zasady nawigacji pomiędzy poszczególnymi komponentami Aplikacji oraz generalne zasady współpracy z aplikacją oraz zasady świadczenia wsparcia technicznego |
| Opis funkcji | Opis ról i ich uprawnień dla tej klasy użytkowników, opis funkcjonalności oraz interfejsu użytkownika dla tej klasy użytkowników, zasady walidacji pól, opis raportów i zestawień. |
| Zasady komunikacji | Opis komunikacji za pomocą e-mail; |

Podręcznik administratora merytorycznego (PAM)

Wymagania:

Dokumentacja powinna zostać uzupełniona elementy identyfikujące dokument,

Dokumentacja powinna zostać uzupełniona o zasady świadczenia wsparcia technicznego dla tej grupy użytkowników.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Identyfikator dokumentu | Informacje na temat: wersji dokumentu, historii zmian, nazwy i wersja Aplikacji, do którego się odnosi. |
| Opis ogólny | Ogólnie opis do czego służy Aplikacji, zasady nawigacji pomiędzy poszczególnymi komponentami Aplikacji oraz generalne zasady współpracy z aplikacją oraz zasady świadczenia wsparcia technicznego |
| Opis funkcji | Opis ról i ich uprawnień dla tej klasy użytkowników, opis funkcjonalności oraz interfejsu użytkownika dla tej klasy użytkowników, zasady walidacji pól, opis raportów i zestawień. |
| Zasady komunikacji | Opis komunikacji za pomocą e-mail; |

Podręcznik Administratora Systemu (PAS)

Wymagania:

Podręcznik PAS obejmuje wszystkie czynności administracyjne związane z Aplikacjami.

W szczególności PAS zawiera opis zadań administratora Aplikacji, w sposób umożliwiający Zamawiającemu ich realizację bez udziału Wykonawcy:

instrukcje konfiguracji i administracji dostarczonych Aplikacji,

instrukcje postępowania w przypadkach szczególnych oraz awarii, w tym odtworzenia Aplikacji,

opisy komunikatów o błędach w Aplikacjach i procedury rozwiązania takich sytuacji,

dokumenty wymagane przez ustawę z dnia 29 sierpnia 1997 r. o ochronie danych osobowych i przepisy wykonawcze do tej ustawy.

Podręcznik jest zgodny z elementami Aplikacji, które opisuje.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wykaz instrukcji i odpowiedzialności | Pełna lista instrukcji wraz z określeniem zespołu odpowiedzialnego za wykonywanie i przestrzeganie danej instrukcji w zakresie administrowania Aplikacjami. |
| Określenie wersji | Określenie wersji Aplikacji / Produktów, których dotyczy wraz z wersjami systemu operacyjnego/bazy danych/serwerów aplikacyjnych itp. |
| Wytyczne do planu eksploatacji | Wytyczne do Planu Eksploatacji Aplikacji sugerujące np. harmonogram wykonywanych okresowo instrukcji. |
| Instrukcje administratorskie | Opis instrukcji obsługi wszystkich elementów Aplikacji (w tym uwzględniające mechanizmy bezpieczeństwa przetwarzania danych) niezbędnych dla eksploatacji i utrzymania Aplikacji.  Instrukcje administracyjne powinny dotyczyć co najmniej:  oprogramowania (wraz z obsługą danych),  wymaganej konfiguracji infrastruktury programowo – sprzętowej.  Opis powinien zawierać informacje nt. zachowania się w przypadku wystąpienia awarii i konieczności odtworzenia Aplikacji. |
| Komunikaty systemu | Opis komunikatów systemu (np. występujących w logach czy wyświetlanych na ekranie), w tym komunikatów o błędach – wraz ze szczegółowym wyjaśnieniem ich znaczenia. |
| Wymogi prawne | Opis dokumentów wymaganych przez ustawę o ochronie danych osobowych – w kontekście administrowania Aplikacjami. |

Podręcznik eksploatacji Aplikacji (PES)

Wymagania:

Podręcznik PES pokrywa pełny zakres procedur eksploatacyjnych Aplikacji.

Podręcznik, poprzez wprowadzane zasady eksploatacji, optymalizuje wykorzystanie zasobów (organizacyjnych i technologicznych) pod kątem spełnienia wymagań eksploatacyjnych systemu.

Opis działań dotyczących eksploatacji systemu umożliwia Zamawiającemu ich realizację bez udziału Wykonawcy.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Role i ich zakresy odpowiedzialności | Wykaz ról pełnionych w przez osoby w realizacji zadań eksploatacyjnych. |
| Cykliczne zadania eksploatacyjne | Szczegółowy wykaz cyklicznych zadań eksploatacyjnych wraz z pełnym opisem:  Nazwa zadania.  Wykaz ról uczestniczących w realizacji zadania, również jeśli rola występuje wyłącznie w czynnościach opcjonalnych.  Określenie kiedy zadanie jest wykonywane.  Określenie momentu zakończenia zadania – np. poprzez określenie czasu trwania lub czasu zakończenia.  Czynności wykonywane w ramach zadania, z określeniem:  roli, która wykonuje daną czynność,  wykorzystywanych komponentów oprogramowania. |
| Jednorazowe zadania eksploatacyjne | Określenie zasad zlecania jednorazowych zadań eksploatacyjnych oraz szablonu zlecenia zadania jednorazowego. |
| Opis stanowisk pracy | Pełna charakterystyka stanowiska pracy Użytkownika Aplikacji:  Opis przeznaczenia danego stanowiska pracy.  Wymagany sprzęt stanowiska pracy, np. minimalna konfiguracja komputera dla stanowiska pracy.  Wymagane oprogramowanie stanowiska pracy wspierające pracę systemu np. system operacyjny, przeglądarka, oprogramowanie biurowe, etc.).  Wymagane wsparcie telekomunikacyjne (sieć telefoniczna, LAN, karta GPRS, etc.).  Wymagane materiały eksploatacyjne.  Wymagania charakteryzujące bezpieczne użytkowanie systemu informatycznego, np. umiejscowienie stanowiska, prace w pomieszczeniu o ograniczonym dostępie.  Wymagane doświadczenie/umiejętności administratora Aplikacji.  Inne wymagania specyficzne dla Aplikacji. |

Specyfikacja wymagań do dokumentu „Specyfikacja Wymagań Biznesowych (SWB)” - wymagania:

Podstawowym pojęciem związanym ze specyfikacją wymagań biznesowych będzie scenariusz biznesowy wspierany przez System. Scenariusz biznesowy należy rozumieć jako logicznie wydzielone i przynoszące określone rezultaty działanie biznesu. Opisywane w dokumentacji scenariusze biznesowe są w całości bądź w części wspierane przez Aplikację. SWB musi enumeratywnie wymienić scenariusze, które wspiera system,

Z każdym scenariuszem biznesowym powinien być związany model dziedziny (może być jeden wspólny dla wszystkich). Diagramy dziedziny opisujące poszczególne scenariusze nie mogą zawierać sprzecznych informacji. Obiekty umieszczone na poszczególnych diagramach dziedziny muszą pochodzić ze wspólnej puli obiektów biznesowych,

Każdy scenariusz biznesowy musi być związany z przynajmniej jednym aktorem biznesowym oraz być dokumentowany za pomocą diagramu aktywności lub notacji BPMN,

Każdy obszar biznesowy musi być powiązany z listą rezultatów, które produkuje i z listą zasobów z których korzysta,

Każdy obszar biznesowy musi zostać powiązany z regułami i/lub wymaganiami biznesowymi. Na liście wymagań biznesowych muszą się znaleźć wymagania dotyczące dostępności, specyficzne wymagania z dziedziny bezpieczeństwa (np. dot. danych osobowych) oraz wymaganej wydajności Systemu.

Wymagany zakres informacyjny SWB:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Lista Scenariuszy Biznesowych | Każdy scenariusz biznesowy musi posiadać:  Unikalną nazwę,  Szczegółowy werbalny opis  Model dziedziny,  Diagram aktywności lub BPMN wraz z powiązanymi Aktorami Biznesowymi  Powiązana Listę wymagań biznesowych, |
| Model dziedziny | Opis obiektów biznesowych modelowanej dziedziny i relacje występujące pomiędzy nimi. Model dziedziny musi być przedstawiony zarówno w formie tekstowej (katalog obiektów wraz z ich opisem), jak i graficznej (tj. diagram modelu dziedziny). |
| Aktorzy biznesowi | Wykaz aktorów biznesowych powiązanych z obszarem.  Każdy aktor zostanie opisany przez:  identyfikator aktora,  opis aktora. |
| Wymagania biznesowe | Wykaz wymagań biznesowych powiązanych ze scenariuszem biznesowym. Wymagania muszą być opisane następującymi atrybutami:  identyfikator wymagania,  nazwa wymagania,  szczegółowa treść wymagania,  wersja wymagania,  data jego dodania do Aplikacji,  autor - imię i nazwisko autora, autorów,  uwagi (opcjonalne),  załączniki plikowe (opcjonalne). |

Specyfikacja wymagań do dokumentu „Specyfikacja Wymagań Systemu Informatycznego” (SWSI) - wymagania:

Wymagania funkcjonalne i poza funkcjonalne muszą bezpośrednio wiązać się z konkretnym wymaganiami biznesowymi (SWB). Związki muszą być utrzymywane w Repozytorium Architektury,

Każde wymaganie funkcjonalne musi być powiązane z konkretnym przypadkiem użycia,

Wymagania te muszą być w pełni rozliczalne tj. muszą być mapowane na przypadki testowe – z opisem zakresu i sposobu realizacji każdego z rozważanych wymagań. Związki wymagania funkcjonalne, przypadki użycia oraz przypadki testowe muszą być utrzymywane jawnie w Repozytorium Architektury,

Dokument SWSI stanowi kluczowy raport z modelu wymagań,

Rozróżnienie wymagań funkcjonalnych i poza funkcjonalnych będzie realizowane poprzez stosowną konwencję nazewniczą.

Zakres informacyjny dokumentu prezentowany jest z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Ogólny opis systemu | Ogólny opis głównych czynników wpływających na kształt Aplikacji i jego wymagania, w ramach którego musi być przedstawiony:  Kontekst (środowisko), w jakim będzie działał system.  Charakterystykę użytkowników systemu (aktorów) w formie dekompozycji aktorów biznesowych wyspecyfikowanych w SWB.  Uwarunkowania, jakie zostały uwzględnione podczas tworzenia systemu.  Założenia i zależności np. dotyczące wykonalności technicznej systemu. |
| Kontekst wymagań na system informatyczny | Opisanie sposobu wynikania (ang. traceability) kluczowych wymagań na Aplikacje z wymagań biznesowych oraz innych czynników sterujących. |
| Wymagania  funkcjonalne | Wykaz wymagań funkcjonalnych na Aplikacje.  Każdy przypadek użycia musi być przedstawiony za pomocą  następujących atrybutów:  identyfikator,  nazwa wymagania,  opis wymagania,  źródło pochodzenia danego wymagania,  status wymagania określający jego cykl życia,  zgłaszający – podmiot bądź osoba, która zgłosiła wymaganie,  wersja wymagania, data jego powstawania i data jego ostatniej modyfikacji. |
| Wymagania  Pozafunkcjonalne | Wykaz wymagań poza funkcjonalnych na system informatyczny.  Muszą być uwzględnione co najmniej następujące rodzaje wymagań poza funkcjonalnych:  wymagania na niezawodność Aplikacji,  wymagania wydajnościowe na Aplikacje,  wymagania dotyczące architektury Aplikacji (tj. konstrukcji systemu),  wymagania dotyczące bezpieczeństwa Aplikacji (w szczególności wymagania dotyczące rozliczalności – zakresu informacyjnego danych zapisywanych do dzienników usług i dzienników danych Aplikacji) oraz dotyczące ochrony przetwarzanych danych (w szczególności danych osobowych),  wymagania dotyczące użyteczności (usability) Aplikacji,  wymagania dotyczące Oprogramowania Systemowego (np. platform systemowych, platform baz danych, platform aplikacji, itp.) i urządzeń którymi dysponuje Zamawiający z punktu widzenia Aplikacji,  wymagania dotyczące interfejsów, które muszą posiadać Aplikacje(zarówno zewnętrznych, jak i wewnętrznych) – np. użytkownika, sprzętowe, programowe, sieciowe,  wymagania wynikające z wymaganych do stosowania standardów informatycznych i biznesowych Zamawiającego,  wymagania związane z warunkami licencjonowania, prawami autorskimi, patentowymi itp. lub ograniczenia związane z użytkowaniem, które musi spełnić system,  wymagania odnoszące się do wspierania bądź utrzymywania zbudowanego systemu, w tym standardy pisania kodu, konwencje nazewnictwa, biblioteki klas, dostęp serwisowy i narzędzia serwisowe. |

Specyfikacja wymagań do dokumentu „Opis Architektury Rozwiązania” (OAR) - wymagania:

Należy przygotować dokument OAR zgodnie z następującym szablonem:

Rozdział dotyczący Perspektywy Biznesu,

Rozdział dotyczący Perspektywy Aplikacji,

Rozdział dotyczący Perspektywy Infrastruktury

Zakłada się hierarchiczność dokumentowania architektury. OAR jest dokumentem o najwyższym poziomie zagregowania, zawierający definicje pojęć występujących we wszystkich obszarach (biznesu, aplikacji i infrastruktury). W OAR powinny znajdować się informacje, które są następnie dekomponowane w innych częściach dokumentacji,

Zakres informacyjny dokumentu ma być prezentowany z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Sposób dokumentowania architektury | Opis przyjętego sposobu dokumentowania architektury Aplikacji – w szczególności objaśnienie wykorzystanych perspektyw wraz z charakterystyką ich zawartości. |
| Syntetyczne przedstawienie architektury rozwiązania | Syntetyczny opis architektury Aplikacji przedstawiający za pomocą diagramu następujące warstwy:  warstwę biznesową złożoną ze scenariuszy biznesowych i głównych aktorów,  warstwę modułów i ich zasadniczych komponentów,  warstwę wdrożenia obejmującą lokalizacje oraz główne węzły. |
| Kluczowe wymagania architektoniczne | Wykaz istotnych wymagań funkcjonalnych i poza funkcjonalnych mających istotny wpływ na obecny kształt architektury Aplikacji np. dotyczących bezpieczeństwa, dostępności, wolumetrii danych, integracji z systemami zewnętrznymi.  Każde wymaganie zostanie opisane przez:  identyfikator wymagania,  opis wymagania. |
| Kluczowe decyzje architektoniczne | Wykaz istotnych decyzji kształtujących obecną architekturę Aplikacji. Decyzje architektoniczne mogą dotyczyć np.: aspektów funkcjonalnych oraz poza funkcjonalnych, takich jak bezpieczeństwo systemu, modyfikowalność, niezawodność, przenaszalność.  Każda decyzja zostanie opisana przez:  identyfikator decyzji,  opis słowny decyzji. |
| Wzorce architektoniczne | Opis wzorców architektonicznych stosowanych w oprogramowaniu Aplikacji (np. wzorce projektowe GoF, wzorce integracyjne, wzorce szczególne dla Aplikacji).  Każdy wzorzec architektoniczny zostanie opisany przez:  identyfikator wzorca,  opis wzorca (lub jego nazwę, jeśli jest to wzorzec powszechnie znany). |
| Perspektywa biznesu | W rozdziale musi zostać umieszczony diagram pojęciowy (nomenklatura biznesu) wraz ze szczegółowym opisem pojęć.  Muszą być wymienione wszystkie scenariusze biznesowe. Każdy scenariusz biznesowy musi zawierać kilkuzdaniowy opis uzupełniony o informacje jakie jest jego zadanie i odsyłacz do SWB. Opisy scenariuszy muszą zostać uzupełnione o hierarchię aktorów (UML) powiązanych z aktorami biznesowymi/rolami pełnionymi w systemie. |
| Perspektywa Aplikacji | W rozdziale musi zostać przedstawiona:  Struktura aplikacji,  Zachowanie aplikacji,  Perspektywa danych aplikacji,  Usługi infrastruktury dla aplikacji |
| Struktura aplikacji  Zakłada się hierarchiczność struktury opisu Aplikacji tj. Aplikacja składająca się z  Podsystemów / Modułów, składających się z komponentów o różnym poziomie zagnieżdżenia.  W rozdziale musi się znaleźć opis struktury logicznej Aplikacji w kontekście jej Podsystemów / Modułów i komponentów składowych, Każdy z modułów Aplikacji powinien być skrótowo opisany i zawierać odsyłacz do PMS. Dla każdego komponentu należy podać co najmniej jego nazwę, realizowane funkcjonalności i ogólną definicję interfejsów, które odpowiadają za wzajemne relacje pomiędzy komponentami Aplikacji oraz z innymi systemami. Należy również odnieść się do kwestii architektonicznych dotyczących sposobów integracji komponentów Aplikacji z zewnętrznymi komponentami i systemami. Opis musi być uzupełniony odpowiednimi diagramami pakietów / komponentów.  Opis struktury aplikacji musi być uzupełniony o diagram prezentujący  warstwy aplikacji (tj. prezentacji, logiki biznesowej, baz danych) oraz wzajemnych powiązań pomiędzy poszczególnymi warstwami i elementami. Opis musi zostać przedstawiony w postaci tekstowej i zostać uzupełniony adekwatnym diagramem. |
| Zachowanie aplikacji  Prezentacja działania Aplikacji w kontekście głównych scenariuszy biznesowych, stanowiących istotną, centralną funkcjonalność Aplikacji lub mających znaczący wpływ na architekturę całej Aplikacji.  Prezentacji należy dokonać za pomocą diagramów aktywności lub diagramów sekwencji. Prezentacja musi pokazywać związki z modułami/podsystemami i podstawowymi rezultatami uzyskiwanymi przez konkretny scenariusz biznesowy. |
| Perspektywa danych aplikacji  Opis danych z których korzysta Aplikacja, w szczególności:  Modelu klas projektowych odpowiedzialnych za warstwę danych (zawierającego główne atrybuty i podstawowe metody) będącego dekompozycją modelu/modeli dziedziny aplikacji. Model musi zostać uzupełniony o opisy każdego z atrybutów/metod,  Opis mapowania klas projektowych odpowiadających za warstwę danych na fizyczny model danych wraz z odpowiednimi odsyłaczami do opisów konkretnych tabel, czy procedur wbudowanych,  Precyzyjny opis kwestii zapewnienia transakcyjności, współbieżności, rozproszenia, tzn. jakie mechanizmy są wykorzystywane, jakie elementy oprogramowania je realizują itd.  Należy wymienić kluczowe wymagania i decyzje architektoniczne mające wpływ na perspektywę danych aplikacji. |
| Usługi infrastruktury dla aplikacji  Należy wymienić samodzielnie funkcjonujące elementy Aplikacji (moduły/podsystemy oraz samodzielne komponenty) funkcjonujące pod kontrolą oprogramowania systemowego i połączyć je z nazwanymi usługami. Każda nazwana usługa powiązana z konkretnym modułem/podsystemem oraz samodzielnym komponentem musi zawierać listę parametrów w formie wymagań:  Zapotrzebowania na moc obliczeniową,  Zapotrzebowania na pamięć dyskową,  Usługi sieciowe,  Wymagania na dostępność.  Należy wymienić kluczowe wymagania i decyzje architektoniczne mające wpływ na perspektywę fizyczną. |
| Zestawienie infrastruktury programowej i sprzętowej | Charakterystyka niezbędnej infrastruktury programowej i sprzętowej wykorzystywanej przez oprogramowanie Aplikacji.  Zestawienie infrastruktury programowej obejmuje oprogramowanie systemowe (np. system operacyjny, serwery aplikacji, oprogramowanie integracyjne, oprogramowanie baz danych itp.).  Zestawienie infrastruktury programowej i sprzętowej należy wykonać w formie diagramów wdrożenia. Każdy elementów umieszczonych na diagramie musi posiadać aktualny opis, który musi być przechowywany w Repozytorium Architektury.  Wykonawca zapewni narzędzie do wydruku w formie tabelarycznej zestawień:  Zestawienie oprogramowania :  nazwę oprogramowania, typ, wersję, producenta,  liczbę i rodzaj wykorzystywanych licencji.  Zestawienie sprzętu:  nazwę urządzenia, typ, wersję, producenta,  liczbę wykorzystywanych urządzeń danego typu |

Specyfikacja wymagań do dokumentu „Projekt Modułu /Podsystemu Systemu” (PMS) - wymagania:

PMS musi być dokumentem składających z osobnych tomów z których każdy opisuje jeden moduł /podsystem Aplikacji

Identyfikowany model danych w ogólnym przypadku obejmuje model danych Aplikacji (bazujący na modelu obiektów biznesowych) oraz model danych jego interoperacyjności (semantycznej, ew. technicznej dla danych) z systemami zewnętrznymi.

Zakres informacyjny dokumentu ma być prezentowany z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Opis techniczny | Wykaz standardów technicznych oraz podejść projektowych wykorzystywanych do implementacji komponentów oprogramowania w danym module Aplikacji. |
| Wykaz narzędzi wspomagających tworzenie systemów informatycznych | Wykaz narzędzi wspomagających tworzenie komponentów oprogramowania w postaci tabeli zawierającej:  nazwę danego narzędzia,  oznaczenie wykorzystywanej wersji danego narzędzia,  opis zastosowania – krótki opis celu wykorzystania danego narzędzia (w jakim celu jest wykorzystywane). |
| Koncepcja integracji | Opis integracji danego modułu z innymi modułami i systemami  Informatycznymi.  Dla każdej integracji pomiędzy modułami Aplikacji należy opisać:  Moduły, których dotyczy dana integracja. Moduły, których dotyczy integracja oraz zewnętrzne systemy informatyczne należy umieścić na odpowiednim diagramie uzupełniając je o opis interfejsów,  Cel integracji modułów (jakie zadanie spełnia dana integracja),  Charakterystyka i sposób wymiany danych pomiędzy modułami tzn. jakie informacje i za pomocą jakich mechanizmów je się wymienia. |
| Model komponentów modułu/podsystemu | Opis poszczególnych komponentów oprogramowania wchodzących w skład modułu. Zamieszczony opis musi prezentować ogólną informacje o ww. elementach oraz sposób ich wzajemnej interakcji.  Minimalny zakres informacji związanych z opisem modułu to:  opis komponentów wchodzących w skład modułu z podziałem na komponenty funkcjonalne (wymagania funkcjonalne specyficzne dla modułu), komponenty danych (implementacja warstwy danych) oraz komponenty biblioteczne (jeśli jest taka potrzeba),  opis powiązań pomiędzy komponentami (tj. identyfikacja interfejsów),  ogólny opis interfejsów:  identyfikator interfejsu;  nazwa interfejsu;  syntetyczny opis interfejsu;  rodzaj interfejsu (np. plikowy, WebAPI);  powiązanie modułu z konkretnymi przypadkami użycia oraz aktorami w formie diagramu przypadków użycia. |
| Logiczny model danych | Diagram obiektów biznesowych wykorzystywanych przez moduł wraz opisem.  Dekompozycja modelu danych przedstawionych w OAR, ograniczona do elementów wykorzystywanych przez moduł. Dekompozycje należy dokonać w formie szczegółowego opisu logicznego modelu danych uzupełnionego przez odpowiedni diagram.  Każdy element logicznego modelu danych musi zostać powiązany z elementami fizycznego modelu danych, które go implementują. |
| Fizyczny model danych | Opis tabel wchodzących w skład modelu uzupełnionych o opis relacji między nimi w tym zestawienie kluczy obcych ze wskazaniem nazwy klucza, kolumny referencji i tabeli referencji  Opis każdej tabeli i widoku należy uzupełnić o opis wierszy, kluczy głównych oraz indeksów ze wskazaniem nazwy, kolumn składowych z zachowaniem kolejności oraz unikalności indeksu.  Opis kodu działającego po stronie bazy danych, w tym wyzwalaczy, procedur, itd. |
| Model wdrożenia | Model wdrożenia dla danego modułu składających się ze stosownych diagramów wdrożenia opracowanych w języku UML.  Model wdrożenia opisuje rozmieszczenie komponentów i obiektów w węzłach (nodes, typu CPU, pamięć RAM, urządzenie/serwer itp.), jak również występujące tam (inne) artefakty oraz wzajemne połączenia sieciowe.  Do artefaktów należy zaliczyć skrypty, konfiguracje, pliki, kody źródłowe, fizyczne zbiory/bazy danych, komponenty przetwarzające itd. |

Specyfikacja wymagań do dokumentu „Specyfikacja Komponentu Oprogramowania \_ Funkcjonalny” (SKOF) - wymagania:

SKOF musi być dokumentem składających z osobnych tomów z których każdy opisuje jeden komponent funkcjonalny będącego rozwinięciem opisu modelu komponentów zamieszczonego w PMS,

Specyfikacji komponentu powinna wyjaśnić: jaka jest struktura wewnętrzna danego komponentu wraz z powiązaniami, interfejsy z otoczeniem i zakresy wymienianych informacji, zakres realizowanych wymagań systemowych,

Zakres informacyjny dokumentu prezentowany jest z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny dla komponentu funkcjonalnego:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Kontekst komponentu | Cel i zastosowanie komponentu. Ogólny opis komponentu oraz powiązań z jego otoczeniem, w tym interfejsów. Opis powinien posiadać odsyłacz/odsyłacze do PMS. |
| Skład komponentu | Model klas komponentu z wyróżnionymi stereotypami oraz opisanymi metodami i atrybutami. Model musi zostać uzupełniony o opisy poszczególnych klas, atrybutów i metod.  Opis musi zawierać specyfikację interfejsów danego komponentu. |
| Zakres realizowanych wymagań | Lista wymagań funkcjonalnych i poza funkcjonalnych uzupełniona o listę przypadków użycia powiązanych z komponentem. |
| Model danych | Opis logicznego i fizycznego modelu danych komponentu. Opis musi zawierać diagram modelu danych, jak i jego specyfikację.  Model musi być dekompozycją modelu danych zamieszczonego w PMS i musi zachowywać z nim spójność. |
| Zachowanie komponentu | Przedstawienie logiki działania komponentu w postaci diagramów sekwencji lub diagramów aktywności. |
| Specyfikacje algorytmów (w miarę potrzeb) | Opis złożonych algorytmów obejmujący zastosowane w komponentach algorytmy wyliczania wartości liczbowych lub logicznych. Algorytmy zostaną opisane za pomocą diagramów zachowania oraz opisu słownego.  Przedstawiony w dokumencie model danych powinien odpowiadać poziomem szczegółowości opisom algorytmów komponentu. |
| Odniesienie do wzorców projektowych | Wskazanie zastosowanych wzorców projektowych (odniesienie o OAR) wraz z podaniem miejsca zastosowania. |
| Zastosowane bibliotek | Zestawienie i ogólny opis zastosowanych bibliotek. |

Specyfikacja wymagań do dokumentu „Specyfikacja Komponentu Oprogramowania \_ Bazodanowy” (SKOB) - wymagania:

SKOB musi być dokumentem składającym się z osobnych tomów dla każdego modułu. Każdy tom musi opisywać wszystkie komponenty warstwy danych związane z konkretnym modułem,

Specyfikacji komponentu musi być rozwinięciem opisu modelu komponentów zamieszczonego w PMS,

Specyfikacji komponentu powinna wyjaśnić: jaka jest struktura wewnętrzna danego komponentu wraz z powiązaniami, interfejsy z otoczeniem i zakresy wymienianych informacji,

Zakres informacyjny dokumentu prezentowany jest z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny dla komponentu funkcjonalnego:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Kontekst komponentu | Cel i zastosowanie komponentu. Opis musi zostać uzupełniony o listę powiązanych z komponentem obiektów biznesowych. |
| Sposób użycia komponentu | Szczegółowy opis metod publicznych komponentu wraz ze szczegółowym opisem użycia przetwarzania transakcyjnego przez komponent. |
| Skład komponentu | Model klas komponentu z wyróżnionymi stereotypami oraz opisanymi metodami i atrybutami. Model musi zostać uzupełniony o opisy poszczególnych klas, atrybutów i metod.  Opis musi zawierać specyfikację metod publicznych/interfejsów danego komponentu. |
| Logiczny model danych | Odwołanie do dokumentu PMS i ewentualnie ograniczenie granicy danych do obiektów przetwarzanych przez komponent |
| Fizyczny model danych | Odwołanie do dokumentu PMS. |

Projekt Infrastruktury Systemu (PIS)

Wymagania:

Zawartość dokumentu powinna pozwalać na zidentyfikowanie pokrycia potrzeb infrastrukturalnych Aplikacji przez przydzielone do niego lub planowane do przydzielenia zasoby.

Należy oczekiwać wyspecyfikowania przynajmniej środowisk: deweloperskiego, szkoleniowego, testowego, produkcyjnego – wraz z określeniem ich skalowalności.

Zakres informacyjny dokumentu prezentowany jest z wykorzystaniem języka UML, w szczególności zawierać będzie elementy, relacje i diagramy zgodne z powyższym językiem.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Rodzaje środowisk Aplikacji | Charakterystyka środowisk (deweloperskiego, szkoleniowego, testowego, produkcyjnego) opisana poprzez zestaw atrybutów:  rodzaj środowiska,  użytkowników środowiska (zewnętrznych, wewnętrznych),  cel istnienia,  lokalizację środowiska (wskazanie ośrodka przetwarzania danych). |
| Architektura logiczna systemu | Specyfikacja i charakterystyka komponentów Aplikacji oraz schemat blokowy Aplikacji |
| Opis poszczególnych środowisk | Opis wymagań niefunkcjonalnych:  zasięg systemu [opis],  liczba użytkowników w Aplikacjach [szt.],  relacje z innymi systemami:  zasila systemy [opis]  jest zasilany z systemów [opis]  szacowany rozmiar danych w systemie [GB],  planowany przyrost danych w ciągu roku [GB],  wymagania systemu na platformę sprzętowo-systemową,  wymagania systemu na platformę bazodanową,  inne. |
| Architektura systemu | Opis i diagramy architektury dla danego środowiska, uwzględniające warstwy Aplikacji (prezentacji, aplikacji, DB, integracji, itp.) |
| Serwery aplikacyjne | Opis serwerów aplikacyjnych w danym środowisku – za pomocą zestawu atrybutów:  nazwa serwera aplikacyjnego,  przeznaczenie (m.in. lista wdrażanych komponentów),  producent, rodzaj i wersja,  rodzaj konfiguracji (np. klaster – liczba nodów / standalone),  opis konfiguracji serwera i jego parametry,  opis powiązań serwera aplikacyjnego z otoczeniem (np. bazą danych, LDAP). |
| Systemy baz danych | Opis baz danych w danym środowisku – za pomocą zestawu atrybutów:  nazwa bazy danych,  funkcja bazy danych,  producent, rodzaj i wersja RDBMS,  rodzaj konfiguracji (np. klaster – liczba nodów / standalone),  rozmiar bazy danych,  szacowany przyrost danych (w ciągu roku),  opis środowiska bazy danych i parametry bazy,  fizyczna struktura bazy danych na poziomie przestrzeni tabel,  opis powiązań bazy danych,  komunikacja aplikacji z bazą danych,  lista użytkowników bazy,  zadania automatyczne. |
| Urządzenia klienckie i peryferyjne | Rodzaje i liczba użytkowników Aplikacji oraz ich wymagania na komputery klienckie wraz z ich oprogramowaniem i urządzeniami peryferyjnymi |
| Koncepcja backup’u | Ogólny opis systemu backupu i archiwizacji oraz urządzeń i narzędzi przewidywanych do obsługi procesu backupu i archiwizacji (m.in. serwer backupowy, klienci, urządzenia przechowujące).  Opis:  architektury systemu backupu i archiwizacji,  oprogramowania do backupu i archiwizacji,  polityk backupu,  lista elementów Aplikacji objętych backup’em i archiwizacją. |
| Koncepcja Disaster Recovery | Opis koncepcji odtworzenia Aplikacji po awarii dla każdego ze środowisk Opis Architektury Rozwiązania Disaster Recovery |
| Wymagane licencje | Wykaz zbiorczy wszystkich potrzebnych do wdrożenia Aplikacji licencji dla wszystkich środowisk i we wszystkich ośrodkach przetwarzania danych scharakteryzowany poprzez informacje:  rodzaj licencji,  nazwa licencji,  sposób licencjonowania,  liczba,  okres wsparcia. |
| Załączniki | W ramach załączników do dokumentu należy przestawić listę i zawartość plików konfiguracyjnych. |

Plan Zarządzania Wymaganiami (PZW)

Wymagania:

PZW jest kluczowym dokumentem zarządczym definiującym proces zarządzania wymaganiami w ramach Umowy

Wykonawca musi określić i uzgodnić z Zamawiającym jakie narzędzia będą użyte do zarządzania wymaganiami i utrzymania powiązań z innymi repozytoriami czy rejestrami (np. z modelem architektury, z modelem przypadków testowych zamkniętych w scenariusze).

Rozróżnienie wymagań funkcjonalnych i pozafunkcjonalnych będzie realizowane poprzez stosowną konwencję nazewniczą.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Organizacja, zakres  odpowiedzialności i zasady komunikacji | Opis, kto będzie odpowiedzialny za poszczególne działania opisane w procesie zarządzania wymaganiami oraz charakterystyka zasad komunikacji. |
| Narzędzia | Opis narzędzi, które będą używane do realizacji zadań związanych z zarządzaniem wymaganiami podczas realizacji niniejszej Umowy. |
| Sformułowanie wymagań | Opis typów wymagań (funkcjonalnych i niefunkcjonalnych) wykorzystywanych w trakcie realizacji systemu informatycznego, jak i zdefiniowanie konwencji ich nazewnictwa, oznaczenia i numeracji. |
| Zatwierdzanie wymagań | Definicja zasad zatwierdzania zidentyfikowanych wymagań, a szczególnie zasad wprowadzania zmian w zaakceptowanych wymaganiach. |
| Kontrola wymagań | Dla każdego typu wymagania, należy wymienić dodatkowe reguły lub wskazówki niezbędne do zastosowania w celu śledzenia jego powiązań z innymi wymaganiami. Należy opisać wszystkie zasady, których należy się trzymać, np. każda zaakceptowana cecha oprogramowania musi być powiązana z przynajmniej jednym wymaganiem. |
| Atrybuty wymagań | Adekwatnie do konwencji przyjętej SWB |

Instrukcja Kompilacji i Tworzenia Pakietu Instalacji (IKTP)

Wymagania:

Instrukcja powinna umożliwić realizację budowy kodu źródłowego do postaci kodu wykonywalnego Aplikacji, a następnie jego instalację.

Opisy powinny uwzględniać fakt, że Zamawiający nie ma wszystkich repozytoriów, narzędzi i wiedzy Wykonawcy, stąd ważne jest dokonanie transferu pełnej wiedzy i wskazanie dostępnych dla Zamawiającego metod i środków działania.

Budowa Aplikacji powinna być możliwa bez dostępu do Internetu (np. z wykorzystaniem wewnętrznych repozytoriów bibliotek).

Pakiet instalacyjny rozumiany jest zarówno jako pakiet aktualizacyjny czy usuwający wadę dla danej wersji Aplikacji, jak i pakiet pełnej instalacji Aplikacji. Dokument IKTP obejmuje wszystkie rodzaje instalacji (m.in. aktualizację i instalację pełną).

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Opis środowiska wytwórczego | Opis środowiska wytwórczego, w tym jego komponentów składowych oraz ich konfiguracji. |
| Opis przygotowania pakietu instalacji | Instrukcje związane z kompilacją i konsolidacją kodu (budową kodu) oraz wstępne prace konfiguracyjne, zmierzające do przygotowania pakietu instalacji systemu. |
| Opis instalacji i konfiguracji | Informacje niezbędne do przeprowadzenia pełnej instalacji i konfiguracji wszystkich elementów Aplikacji wymaganych od Wykonawcy. W przypadku zastosowania mechanizmów automatyzacji procesu instalacji, należy przedstawić również opis manualnego instalowania Aplikacji. Należy opisać instrukcje instalowania i wycofywania nowej wersji Aplikacji. |

Opis Wersji Oprogramowania (OWO)

Wymagania:

OWO jest elementem kontroli konfiguracji w procesie wytwarzania Aplikacji

Każda wersja (wydanie) ma określoną funkcję, poziom szczegółowości. Utrwalanie i archiwizowanie tych wersji ma tym większe znaczenie, im bardziej zbliżone są one do wydań głównych.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Nazwa wydania | Symboliczna nazwa wydawanego kodu. |
| Wersja - numer | Unikalne oznaczenie wersji, zgodne ze stosowaną konwencją i zasadami wersjonowania np. kod z liczb naturalnych w układzie „[numer główny].[numer dodatkowy].[numer wydania poprawkowego wersji rozwojowej]. |
| Znacznik czasu wydania | Stempel czasowy wydania – data i godzina z minutami. |
| Status rozwoju oprogramowania | Oznaczenie etapu według systematyki np.:  Wersja niestabilna – testowa:  Wersja robocza – wersja dostępna przeważnie tylko dla twórców, w której zaimplementowano algorytmy, definiowany jest interfejs użytkownika, dodaje się funkcje oprogramowania.  Wersja alfa – wersja sygnalizująca w określonym zakresie funkcjonalność oprogramowania, udostępniana dla zebrania uwag, propozycji zmian, sugestii (np. wersja prototypowa).  Wersja beta - wersja wydawana w celu przetestowania kodu w rożnych środowiskach i warunkach, aby zidentyfikować ewentualne błędy w działaniu oraz dla optymalizacji działania kodu.  Wersja RC (ang. release candidate) – kandydat do wydania stabilnego, wersja gotowa do celów produkcyjnych, ale publikowana jako testowa i bez gwarancji poprawności działania.  Wersja stabilna – oznaczenie wersji finalnej, produkcyjnej |
| Odnośniki do dokumentacji bazowej | Odnośniki do dokumentacji bazowej, której wynikiem jest dana wersja oprogramowania (m.in. SWB, OAR, SWSI).  Odniesienie do procedur instalacji z ewentualnymi uwagami dodatkowymi dotyczącymi danej wersji. |
| Komponenty składowe | Lista komponentów składowych i ich wersji. |
| Opis zmian wersji | Lista zmian oraz możliwych problemów. |

Skomentowany Kod Źródłowy (SKZ)

Wymagania:

SKZ składa się z właściwego skomentowanego Kodu Źródłowego oraz opisu tej dostawy w postaci dokumentu DKZ,

Jakość komentarzy ma bezpośrednie znaczenie dla jakości DKZ, z tego powodu Wykonawca jest odpowiedzialny za zapewnienie odpowiedniej zawartości informacyjnej.

Opis należy sporządzać zgodnie z konwencją i najlepszymi praktykami w tym zakresie, dostosowując się do stosowanego języka oraz dedykowanego narzędzia.

Jako roboczą regułę wyznaczającą jakość opisu jest, aby komentować kod w taki sposób, jakiego tworzący komentarz programista sam by oczekiwał - co do zakresu, podejścia, zawartości, szczegółowości, konsekwencji w stylu, spójności konwencji itd. Minimalne wymagania to:

Każda klasa (aplikacji, formularzy, raportów itd.) musi zawierać kilkuzdaniowy komentarz opisujący, jakiego rodzaju obiekty generuje i jaka jest ich semantyka,

Każdy atrybut każdej klasy musi zawierać komentarz opisujący jego znaczenie,

Każda metoda każdej klasy musi zawierać komentarz opisujący, do czego metoda służy, jakie ma parametry (co one oznaczają) oraz jaką wartość zwraca,

Każde wywołanie metody obiektu musi zawierać komentarz objaśniający czemu służy,

Każde wykonanie instrukcji SQL musi zawierać komentarz objaśniający czemu służy

Brak komentarzy lub ich niska jakość traktowana będzie jako wada jakościowa Kodu Źródłowego, która zostanie krytycznie oceniona w trakcie dokonywania odbioru przez Zamawiającego, ze wstępnym wskazaniem rezultatu jako uwaga istotna, uniemożliwiająca odbiór produktu.

SKZ będzie instalowany na środowisku wytwórczym Zamawiającego z zachowaniem zasad zarządzania konfiguracją.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wykaz kodów  źródłowych | Wykaz wszystkich Kodów Źródłowych koniecznych do generowania określonej wersji Aplikacji, modyfikacji oraz dalszego rozwoju zbudowanych komponentów oprogramowania Aplikacji. W zestaw kodów źródłowych zalicza się również wszelkie dodatkowe zasoby takie jak skrypty, dane konfiguracyjne, itp. |
| Charakterystyka zestawów kodów źródłowych | Charakterystyka zestawu kodów źródłowych opisana przez zestaw co najmniej następujących atrybutów:  identyfikator zestawu kodów źródłowych,  nazwa zestawu kodów źródłowych,  opis przeznaczenia zestawu kodów źródłowych,  opis formatu dostarczonego zestawu kodów źródłowych (np. archiwum ZIP),  lista technologii, wraz z wersją technologii, w których zostały wytworzone kody źródłowe danego zestawu,  wykaz środowisk programistycznych, wraz z wersją danego środowiska, w których zostały wytworzone kody źródłowe danego zestawu,  pełna charakterystyka zestawu:  katalogów,  plików,  skryptów,  dokumentacji Kodów Źródłowych. |
| Kod Źródłowy | Kod Źródłowy oprogramowania w zakresie dostawy (w postaci elektronicznej i edytowalnej z oznaczeniem wersji/wydania). |
| Instrukcje konfiguracji środowiska i utworzenia paczki instalacyjnej | Odesłanie do odpowiedniego dokumentu Instrukcja Kompilacji i Tworzenia Pakietu Instalacji (IKTP) dla opisywanej paczki instalacyjnej. |

Dokumentacja Kodu Źródłowego (DKZ)

Wymagania:

DKZ jest generowana na podstawie SKZ przy użyciu wybranego dedykowanego narzędzia i jest pozyskiwana z niego na podstawie odpowiednich znaczników wpisywanych w komentarze (o składni zgodnej z regułami narzędzia).

Jakość DKZ wynika z zasad stosowanych przy realizacji produktu SKZ.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Treść dokumentacji Kodu Źródłowego | Dokumentacja kodu w formacie html, wygenerowana stosownym dla Aplikacji narzędziem (np. javadoc).  Dokumentacja powstanie na podstawie komentarzy zawartych w Kodzie Źródłowym i oznaczonych odpowiednimi znacznikami. |
| Konfiguracja narzędzia dokumentującego | Opis konfiguracji narzędzia dokumentującego, w tym jego ewentualne pliki konfiguracyjne w postaci załączników. |

Plan Testów Systemu (PTS)

Wymagania:

Zakłada się, że Plan Testów Systemu, ze względu na wagę tego etapu procesu wytwórczego, nie jest tylko dokumentem zarządczym, ale narzędziem dobrego przygotowania testów odbiorczych Aplikacji.

Szczególnie ważne są harmonogramy testów: zabezpieczenie odpowiednich zasobów, przygotowanie lokalizacji, zgromadzenie odpowiednich osób o odpowiednich kwalifikacjach i wiedzy.

Plan Testów wspiera proces, w którym rozliczane są wszystkie wymagania na Aplikacje w przypadkach testowych zawartych w przeprowadzanych scenariuszach testów. Dzięki wsparciu dodatkowych narzędzi, zarówno Wykonawca, jak i Zamawiający są w stanie systematycznie monitorować stan realizacji i reagować na odstępstwa i zmiany.

Opis oceny stosowany przy weryfikacji i akceptacji wyników testów będzie zgodny z praktykami stosowanymi w realizacji Usługi Modyfikacji i Rozwoju Aplikacji.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Przegląd planowanych testów | Przegląd zakresu testów:  Ogólny przegląd głównych planowanych testów.  Przegląd wyłączeń  Przegląd potencjalnych testów, które mogłyby zostać przeprowadzone, ale zostały z określonych przyczyn wyłączone z zakresu testów. Należy jasno wskazać, dlaczego dane testy (lub cała ich kategoria) zostały wyłączone z zakresu planu testów, podając uzasadnienie, np.:  testy te nie przyczyniają się do oceny zakresu funkcjonalności przewidzianego w ramach tego etapu kontraktu,  brak możliwości technicznych przeprowadzenia określonych testów. |
| Strategia testów | Przegląd zalecanej strategii analizowania, projektowania, tworzenia i wykonywania niezbędnych testów.  Pomiar postępów testów – Wskazanie jakie podejścia zostaną zastosowane do mierzenia postępów prac testowych.  Przeprowadzenie testów – Opisanie, w jaki sposób testy będą przeprowadzane. Charakterystyka obszarów ryzyka dla jakości produktu oraz rodzaje testów, jakie będą przeprowadzone, a także związane z tym techniki testowania. Przedstawienie sposobu prowadzenia testów dla każdej techniki: w jaki sposób zostanie przeprowadzone projektowanie, tworzenie i wykonywanie testów oraz kryteria weryfikacji przydatności i skuteczności danej techniki. |
| Harmonogram testów | Opisanie harmonogramu testów oraz omówienie kryteriów wykorzystywanych do oceny możliwości rozpoczęcia, zakończenia lub zawieszenia wykonywania testów. Harmonogram powinien uwzględniać szkolenia przewidziane na cele procesu testowania. |
| Proces testowania | Przedstawienie kolejności wykonywania testów, którego powinien trzymać się zespół testowy w realizacji planu testów. Przedstawienie poszczególnych zadań testowych. Opisanie wymaganej dokumentacji, powstającej w procesie testowania oraz charakterystyka produktów dodatkowych m.in.:  Testów regresji i pomocniczych skryptów testowych  Dodatkowych skryptów testowych - Mogą to być zestawy kodów źródłowych dla testów automatycznych w postaci skryptów lub odniesienie do repozytorium kodów źródłowych i skompilowanych plików wykonywalnych dla skryptów utrzymywanych przy użyciu wyspecjalizowanego narzędzia do testów automatycznych.  Wskazówek dla testerów - Wskazówki dla testerów obejmują szereg kategorii dokumentów, jak np. Katalogi wskazań testowych, zestawy najlepszych praktyk, wzorce testowe, itd. |
| Środowisko testowe | Charakterystyka zasobów innych niż ludzkie niezbędnych dla realizacji testów – tj.:  sprzęt,  oprogramowanie wspierające proces testowania,  oprogramowanie dodatkowe,  konfiguracja środowiska testowego. |
| Personel | Charakterystyka zasobów ludzkich niezbędnych dla realizacji testów, opisanie zakresu odpowiedzialności oraz poziom wiedzy i umiejętności oczekiwanych od tych zasobów. Należy wskazać, w którym momencie cyklu życia projektu potrzebne będą określone umiejętności i zasoby ludzkie i w jakiej liczbie. Ze względu na fakt, że zespół testowy często potrzebuje wsparcia ze strony członków innych zespołów, należy uwzględnić zaangażowanie i dostępność niezbędnych specjalistów. |
| Proces zarządzania | Opis procesów i procedur, które należy uruchomić w przypadku pojawienia się problemów związanych z „Planem testów systemu” i jego realizacją.  Określenie zasad zarządzania strategią mierzenia i oceny postępów testów.  Opis procesu oceny stosowany przy weryfikacji i akceptacji Raportów z Testów.  Opis sposobu raportowania i eskalowana problemów związanych z testami, oraz opis procesu ich rozwiązywania.  Opis procesu kontroli cyklu testów przez kierownictwo. |

Specyfikacja Przypadków i Scenariuszy Testowych (SPST)

Wymagania:

W przyjętej nomenklaturze zakłada się, że zestaw testów dla danego zakresu Aplikacji składa się ze scenariuszy testowych, a te z kolei z przypadków testowych.

Dany przypadek testowy jest związany z określonymi wymaganiami, dla których zawiera on opis sposobu i zakresu realizacji.

Przypadki testowe są także podstawą rozliczenia wymagań funkcjonalnych i pozafunkcjonalnych na Aplikacji, analogicznie – poprzez wskazanie zakresu i sposobu realizacji.

W celu zapewnienia powiązań przypadków testowych z wymaganiami Wykonawca zapewni przechowywanie przypadków testowych we wspólnym Repozytorium Wymagań, tak by możliwa była analiza spójności, raporty podsumowujące itd.

Planowanie zasobów podczas realizacji testów Aplikacji w SPST:

SPST przedstawia strategię podejścia do testów, w sposób, który optymalizuje wykorzystanie zasobów (osób, czasu, sprzętu, oprogramowanie itp.) po stronie Zamawiającego.

SPST jest przygotowany w ten sposób, że planowane zasoby po stronie Zamawiającego są realne do uzyskania.

Przeprowadzenie równolegle testów z wymaganym udziałem osób o tych samych kompetencjach po stronie Zamawiającego zostało przez niego zaakceptowane.

SPST zawiera informacje dotyczące sposobu pomiaru postępu testów, co pozwala określić:

jaka część planowanych testów została wykonana,

ile testów (pracochłonność, czasochłonność, procent itp.) jest jeszcze do wykonania.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wykaz zestawów testów | Wykaz zestawów testów wykonywanych w ramach tworzenia Aplikacji oraz wykonywanych w celu jego odbioru, opisany za pomocą co najmniej następującego zestawu atrybutów:  identyfikator zestawu,  nazwa zestawu,  zespół, lub osoba odpowiedzialna za przeprowadzenie testów,  lista elementów (np. systemu oprogramowania, komponentów oprogramowania, platform programowych, urządzenia) Aplikacji podlegających kontroli w ramach danego zestawu testów, wraz z listą celów, jakie mają zostać osiągnięte poprzez przeprowadzenie danego testu,  opis przyjętego sposobu testowania, wraz z listą narzędzi wspierających testy (doszczegółowienie w „Planie testów systemu”). W przypadku zastosowania narzędzi wspierających należy tutaj wskazać instrukcje instalowania i konfigurowania danego narzędzia,  kryteria określające pozytywny wynik wykonania danego zestawu testów, w szczególności dla testów odbioru, powinny zostać umieszczone kryteria odbioru,  wykaz zasobów ludzkich koniecznych do przeprowadzenia testów,  wykaz zestawów danych testowych wymaganych do przeprowadzenia zestawu testów. |
| Scenariusze testowe | Opis wykorzystywanych do testów scenariuszy testowych. Każdy scenariusz testowy musi być opisany za pomocą następujących atrybutów:  identyfikator scenariusza,  nazwa scenariusza,  wykaz wymagań (np. identyfikator wymagania, identyfikator przypadku użycia), które są testowane w ramach danego scenariusza testowego,  opis danego scenariusza testowego,  wykaz warunków, jakie muszą być spełnione przed rozpoczęciem wykonania scenariusza testowego, włącznie ze wskazaniem specyficznych danych wejściowych dla danego scenariusza,  wykaz warunków, jakie muszą być spełnione po wykonaniu scenariusza testowego przykładowo stan Aplikacji, jaki musi zostać pozostawiony po wykonaniu scenariusza testowego,  waga danego scenariusza testowego określana ekspercko,  kryteria określające pozytywny rezultat danego scenariusza testowego. |
| Przypadki testowe | Opis wykorzystywanych do testów przypadków testowych (wraz z listą operacji), wykonywanych w ramach scenariusza testowego. Każdy scenariusz testowy musi być opisany za pomocą następujących atrybutów:  wykaz elementów (np. identyfikator wymagania, identyfikator przypadku użycia), które są testowane w ramach danego scenariusza testowego,  opis danego przypadku testowego,  wykaz warunków, jakie muszę być spełnione przed rozpoczęciem wykonania przypadku testowego,  opis operacji wykonywanych w ramach przypadku testowego (w tym: wprowadzane dane, oczekiwany rezultat oraz metodę oceny tego rezultatu). |
| Dane testowe | Opis danych testowych wykorzystywanych podczas realizacji testów, opisany za pomocą co najmniej następującego zbioru atrybutów:  opis przeznaczenia zestawu danych,  opis danego zestawu danych,  lokalizacja repozytorium, dokumentu lub pliku z danymi. |
| Konfiguracja środowiska testowego | Opis konfiguracji środowiska testowego, specyficzny dla zestawu testów, zawierający co najmniej następujące atrybuty:  nazwa środowiska testowego,  konfiguracja tych części Aplikacji, które będą podlegać testom w ramach danego środowiska testowego,  konfiguracja środowiska testowego w zakresie specyficznym dla zestawu testów,  wykaz zasobów sprzętowych wykorzystywanych do przeprowadzenia zestawu,  lista narzędzi wykorzystywanych do testowania, wraz z wykazem narzędzi przeznaczonych do automatyzacji testów (narzędzia, wersje, dodatkowe pakiety doinstalowywane do narzędzi), oraz wykaz paramentów konfiguracyjnych danych narzędzi. |
| Harmonogram testów | Szczegółowy harmonogram przeprowadzania testów Aplikacji (z uwzględnieniem etapu zaplanowania, przygotowania i realizacji testów). |
| Załączniki | Załącznikiem elektronicznym do SPST są dane testowe i dane umożliwiające testy m.in. pliki konfiguracyjne, skrypty, eksporty baz danych, dokumenty xml. |

Raport z Testów (RT)

Wymagania:

Raport z Testów ma wspierać Wykonawcę i Zamawiającego w weryfikacji poprawnego funkcjonowania Aplikacji i rozliczeniu wymagań na ten system.

Raport w jasny, czytelny sposób powinien podsumowywać, jaki zakres wymagań był testowany i jaki był wynik tego testowania, w szczególności obowiązkowo powinien nawiązywać do SPST.

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wynik testów | Przedstawienie końcowego wyniku testów, ich zakresu, formuły  realizacyjnej, wykorzystanych narzędzi testowych. |
| Podsumowanie testów | Syntetyczne podsumowanie, w szczególności przedstawienie informacji statystycznych, oraz porównanie wyników testów z ewentualnymi wcześniejszymi wynikami, przedstawienie ewentualnych rekomendacji, informacje o zrealizowanym i pominiętym zakresie testów. |
| Wskazanie na dokument z testów | Wskazanie na dokument testów będący podstawą przygotowania raportu z testów. |

Wymagania uzupełniające dotyczące prowadzonej Dokumentacji

Wykonawca prowadzi Repozytorium Projektowe zawierającego dokumentację projektu, w oparciu o środowisko dostarczone przez Zamawiającego. Repozytorium służy Służące dla celów rejestracji i rozliczania pracy osób realizujących Umowę po stronie Wykonawcy, środowisko zostanie skonfigurowane we wskazany przez Zamawiającego sposób, na wskazanej przez Zamawiającego infrastrukturze z wykorzystaniem wskazanego przez Zamawiającego środowiska systemu kontroli wersji (GIT), narzędziu typu case-tracker (JIRA, Microsoft Teams), lub systemie DMZ (Sharepoint, Confluence). Ponadto:

w Repozytorium Projektowym, w sposób szczególny wyróżniane są aktualne wersje dokumentacji projektowej;

dokumenty projektowe zawierają historię zmian oraz dane identyfikacyjne, w tym numer wersji;

Wykonawca odpowiedzialny jest za sporządzanie notatek ze spotkań projektowych i umieszczanie ich w Repozytorium Projektowym.

W uzupełnieniu do dokumentacji w Repozytorium Projektowym, Wykonawca prowadzi i utrzymuje następujące repozytoria i bazy wchodzące w jego skład:

Repozytorium Architektury – zawierające model architektury, utrzymywany według ustalonej po uruchomieniu Umowy definicji tego modelu (po ustaleniu tzw. Metamodelu), obejmującego przynajmniej:

strukturę modelu w narzędziu,

bibliotekę obiektów stosowaną do opisu architektury na poziomie biznesowym, logicznym (aplikacji i danych) oraz technicznym,

ustalone diagramy (widoki) architektury.

Repozytorium Architektury będzie m.in. służyć jako źródło do generowania części lub całości dokumentacji Aplikacji omawianej w niniejszym Załączniku. Repozytorium Architektury musi być prowadzone w narzędziu Sparx Enterprise Architect.

Repozytorium Wymagań – zawierające wszystkie wymagania funkcjonalne i pozafunkcjonalne, odnotowujące także informacji pomocnicze (stan realizacji wymagań itp.), mające więzy spójności (odniesienia) poszczególnych wymagań do elementów architektury w Repozytorium Architektury.

Repozytorium Wymagań prowadzone jest w narzędziu Sparx Enterprise Architect.

Repozytorium Kodów Źródłowych – zawierające wszystkie pakiety Kodów Źródłowych, w tym wydania archiwalne.