# **Załącznik nr 1** **do SWZ**

# **Opis przedmiotu zamówienia**

# **Zastosowane definicje.**

* 1. Strony nadają terminom używanym w dalszej treści Umowy następujące znaczenie:

|  |  |
| --- | --- |
| **Termin** | **Definicja** |
| **Analiza i Projekt** | Wynik realizacji I fazy usługi Modyfikacji i Rozwoju Systemu GW/EGW |
| **Analiza Wstępna** | Dostarczony w ramach realizacji Analizy i Projektu Produkt zawierający co najmniej: - harmonogram realizacji Modyfikacji zawierający terminy realizacji wszystkich zadań z nią związanych (m.in. analiza, implementacja, testowanie, szkolenia, wdrożenia). Harmonogram powinien umożliwić Zamawiającemu szczegółowe zarezerwowanie zasobów ludzkich wykorzystywanych na poszczególnych etapach procesów analizy, projektowania, implementacji, testowania i wdrażania Modyfikacji.  - wycenę realizacji opartej o szacunki realizacji wszystkich zadań i podzadań, o których mowa powyżej,  - wykaz obszarów Systemu, których dotyczy Modyfikacja wraz z opisem wpływu zmian na poszczególne obszary i funkcje Systemu,  - opis konsekwencji dla dostępności, wydajności, ergonomii, bezpieczeństwa technicznego Systemu oraz bezpieczeństwa w zakresie gromadzenia oraz przetwarzania danych osobowych (w postaci zgodnej z RODO analizy ryzyka) jakie spowoduje realizacja Modyfikacji.  - wykaz oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Obcego koniecznego do wykorzystania w ramach realizacji Modyfikacji. |
| **Awaria** | Wada inna niż Błąd i Usterka, powodująca całkowite zatrzymanie lub poważne zakłócenie pracy Systemu lub poszczególnych jego części, dla której nie ma alternatywnej metody wykonania danej operacji w Systemie, uniemożliwiająca korzystanie z funkcji Systemu przez jego Użytkowników tak jak było to możliwe przed wystąpieniem Awarii lub uniemożliwienie wywiązania się przez Zamawiającego z nałożonych na niego obowiązków/zadań wynikających z przepisów prawa, lub wysokiego ryzyka powstania sytuacji, w której nie będzie możliwe wywiązanie się przez Zamawiającego z nałożonych na niego obowiązków/zadań wynikających z przepisów prawa. |
| **Błąd** | Wada inna niż Awaria i Usterka, powodująca istotne zakłócenia pracy Systemu lub poszczególnych jego części, która jednak nie uniemożliwia Użytkownikom korzystania z funkcji Systemu, i nie stwarza ryzyka powstania sytuacji, w której nie będzie możliwe wywiązanie się przez Zamawiającego z nałożonych na niego obowiązków/zadań wynikających z przepisów prawa, polegająca w szczególności na ograniczeniu realizacji lub uciążliwości w realizacji co najmniej jednej z funkcji Systemu. |
| **Czas Naprawy** | Czas liczony od momentu dokonania Zgłoszenia Wady przez Zamawiającego do chwili udostępnienia Zamawiającemu Naprawy na Środowisku Produkcyjnym. |
| **Czas Obejścia** | Czas liczony od momentu dokonania Zgłoszenia Wady przez Zamawiającego do chwili dokonania Obejścia na Środowisku Produkcyjnym. |
| **Dokumentacja Systemu** | Wszelka dokumentacja opisująca System i Kody Źródłowe Systemu (w tym również zmiany oraz modyfikacje takiej dokumentacji), dotycząca aspektów technicznych, funkcjonalnych i użytkowych związanych z korzystaniem z Systemu, jego działaniem i rozwojem, w tym dokumentacja Systemu w wersji papierowej oraz elektronicznej. Dokumentacją Systemu jest istniejąca Dokumentacja Systemu, będąca w posiadaniu Zamawiającego na dzień podpisania Umowy, jak również Dokumentacja Systemu, którą Wykonawca zobowiązany jest zaktualizować, dostosować, wytworzyć i dostarczyć zgodnie z Umową. |
| **Dzień Roboczy** | Każdy dzień tygodnia od poniedziałku do piątku, za wyjątkiem dni ustawowo wolnych od pracy w Rzeczpospolitej Polskiej. |
| **Generator Wniosków (GW)** | Oprogramowanie informatyczne wspierające proces przygotowywania i skradania wniosków w ramach ogłaszanych Konkursów o zlecenie realizacji zadań, dokonywania wstępnej oceny złożonych wniosków, publikowania wyników oceny wniosków oraz dokonywania aktualizacji wniosków i składania propozycji aneksów do umowy o dofinansowanie na podstawie art. 36 ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych. |
| **Godziny Robocze** | Godziny od 8.00 do 17.00 w Dni Robocze. |
| **Informacje Poufne** | Wszelkie informacje, dokumenty oraz materiały dotyczące działalności jednej ze Stron, do których druga Strona Umowy uzyskała dostęp w związku z wykonywaniem niniejszej Umowy. Informacjami Poufnymi są w szczególności dane zawarte w dokumentach przekazywanych lub przetwarzanych za pośrednictwem Systemu, wszelkie informacje finansowe, organizacyjne, technologiczne, dane osobowe oraz inne informacje o działalności jednej ze Stron, które posiadają wartość gospodarczą lub zostały udostępnione drugiej Stronie z zastrzeżeniem poufności. |
| **Kierownik Projektu** | Osoba kontaktowa lub podejmująca decyzje dotyczące realizacji Umowy w ramach kompetencji przyznanych w Umowie, wyznaczona przez Zamawiającego/Wykonawcę, odpowiedzialna za prawidłowe wykonywanie zobowiązań wynikających z Umowy oraz bieżący przepływ informacji pomiędzy Stronami. |
| **Kody Źródłowe** | Zestaw plików w formie czytelnej dla człowieka zawierających nieskompilowany kod oprogramowania napisany języku programowania, wynikającym z przyjętej technologii rozwiązania normalnie używanej dla umożliwienia wprowadzania modyfikacji, (w tym również komentarze oraz kody proceduralne, takie jak skrypty w języku opisu prac i skrypty do sterowania kompilacją i instalowaniem), jak również dokumentacja niezbędna do użycia takiego kodu. |
| **Modyfikacja i Rozwój (MR)** | Wszelkie usługi polegające na wprowadzaniu zmian w Systemie, realizowane przez Wykonawcę w zakresie opisanym w Umowie. |
| **Naprawa** | Trwałe usunięcie Wady poprzez usuniecie przyczyn powstania Wady skutkujące przywróceniem pełnej sprawności Systemu, w tym również zakończenie innych działań naprawczych. |
| **Niedostępność Systemu** | Awaria Systemu lub obniżenie parametrów wydajnościowych Systemu, opisanych w Umowie i załącznikach do Umowy. |
| **Obejście** | Zapewnienie funkcjonowania Systemu poprzez zminimalizowanie uciążliwości Wady i doprowadzenie Systemu do działania bez usuwania przyczyny wystąpienia Wady. Obejście nie stanowi Naprawy, jednak pozwala korzystać nieprzerwanie z wszystkich funkcjonalności Systemu. |
| **Odbiór** | Czynności mające na celu potwierdzenie dostarczenia Zamawiającemu usług i Produktów, powstałych w wyniku zobowiązań wynikających z Umowy. |
| **Okno Serwisowe** | Czas pomiędzy godziną 20:00 a 07:00 przeznaczony na wykonywanie wszelkich niezbędnych prac serwisowych, przeglądów, aktualizacji Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu, a także wgrywania nowych wersji Systemu na Środowisko Produkcyjne i Środowiska Testowe. |
| **Oprogramowanie Standardowe / Oprogramowanie Obce** | Oprogramowanie dostarczone przez Wykonawcę, nie wytworzone w toku prac nad realizacją Systemu, stanowiące jego składnik, w tym system operacyjny, system zarządzania bazą danych, na którego użycie w procesach budowy, rozwoju, konfiguracji, instalacji lub użytkowania Systemu Zamawiający wyraził zgodę. Wykonawca powinien uzyskać zgodę Zamawiającego na użycie określonego Oprogramowania Standardowego / Obcego przed przystąpieniem do wszelkich prac, których efektem może być modyfikacja lub rekonfiguracja Systemu. |
| **Oprogramowanie Systemowe i Narzędziowe** | Oprogramowanie wykorzystywane na potrzeby Systemu, konieczne do poprawnego działania Systemu, inne niż Oprogramowanie Zamawiającego. Wykonawca powinien uzyskać zgodę Zamawiającego na użycie określonego Oprogramowania Systemowego i Narzędziowego przed przystąpieniem do wszelkich prac, których efektem może być modyfikacja lub rekonfiguracje Systemu. |
| **Oprogramowanie Zamawiającego** | Oprogramowanie aktualnie wykorzystywane na potrzeby Systemu, które zapewnia Zamawiający. |
| **Pakiet Aktualizacji** | Przygotowane do instalacji uaktualnienie Systemu: służące usunięciu nieprawidłowości, usprawnieniu pracy Systemu, wdrożeniu zmian. |
| **Podwykonawca** | Każdy podmiot inny niż: pracownik Wykonawcy, osoba fizyczna prowadząca działalność gospodarczą czy osoba fizyczna, z którą Wykonawca ma zawartą umowę cywilnoprawną (np. umowa o współpracy). |
| **Portal Serwisowy** | System informatyczny wykorzystywany przez Zamawiającego (np. Jira) służący do ewidencji i obsługi Zgłoszeń, wniosków i Zamówień zapewniający niezbędny poziom wymiany informacji pomiędzy Zamawiającym a Wykonawcą. |
| **Pracownik Zamawiającego**  **(Operator)** | Osoba korzystająca z Systemu. |
| **Produkt** | Wszelkie programy komputerowe, dokumentacja i inne utwory, które powstają w toku wykonywania Umowy w wyniku prac Wykonawcy, w tym także wszelkie modyfikacje i opracowania innych utworów, a także materiały i informacje niepodlegające ochronie prawa autorskiego, stworzone lub dostarczone Zamawiającemu przez Wykonawcę w wyniku wykonania zobowiązań wynikających z Umowy. |
| **Protokół Odbioru** | Dokument przedstawiony przez Wykonawcę i zaakceptowany przez Zamawiającego, potwierdzający prawidłowość i zakres wykonania konkretnych usług i Produktów. Wzory Protokołów Odbioru Usługi Asysty Technicznej i Konserwacji, Usługi Modyfikacji i Rozwoju, stanowią Załącznik nr 2 do Umowy. |
| **Przypadki Szczególne** | To takie, w których Użytkownik pomimo spełnienia wymogów określonych dla Systemu, dotyczących zainstalowanego środowiska oraz mimo wsparcia konsultantów, nie może skorzystać z dowolnej funkcji Systemu przewidzianej jako jedna z dostępnych możliwości. |
| **Pytanie** | Pytania dotyczące działania Systemu w ramach świadczenia Usługi Asysty Technicznej i Konserwacji. |
| **Repozytorium Projektu** | Narzędzie służące do rejestracji i rozliczania pracy osób realizujących Umowę po stronie Wykonawcy, środowisko skonfigurowane we wskazany przez Zamawiającego sposób, na wskazanej przez Zamawiającego infrastrukturze z wykorzystaniem wskazanego przez Zamawiającego środowiska systemu kontroli wersji (np. GIT), narzędziu typu case-tracker (np. JIRA, Microsoft Teams) lub systemie DMS (np. Sharepoint, Confluence). |
| **Roboczogodzina** | Jednostka miary pracochłonności wyrażająca normę ilościową pracy wykonanej przez jednego pracownika Wykonawcy w czasie jednej godziny. |
| **RTO** | Recovery Time Objective – czas niezbędny do przywrócenia Systemu po Awarii stanowiący sumę czasów naprawy Awarii z umowy hostingowej i Czasu Naprawy Awarii w ramach ATiK-u. |
| **RPO** | Recovery Point Objective – punkt w czasie, do którego jest przywrócony System po Awarii. |
| **SLA** | *Service Level Agreement* - Poziom świadczenia usług i sposób jego pomiaru, określony w Załączniku nr 5 do Opisu Przedmiotu Zamówienia |
| **Sprzęt** | Urządzenia, w szczególności sprzęt komputerowy i infrastruktura teleinformatyczna znajdująca się w posiadaniu Zamawiającego, na których działa System w okresie realizacji Umowy. |
| **System** | System GW/EGW, chyba że w treści Umowy wprost wskazano inaczej. |
| **System Ewidencji Godzin Wsparcia (System EGW)** | Oprogramowanie informatyczne wspierające proces gromadzenia i analizy danych o wsparciu udzielonym beneficjentom w ramach projektów realizowanych z dofinansowaniem PFRON na podstawie art. 36 ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych. |
| **System GW/EGW** | Oprogramowanie informatyczne, w skład którego wchodzi: kod źródłowy Generatora Wniosków i Systemu Ewidencji Godzin Wsparcia w postaci wykonywalnej, Oprogramowanie Standardowe/Obce, Oprogramowanie Systemowe i Narzędziowe niezbędne do prawidłowej pracy Systemu (systemy operacyjne, serwery aplikacji, bazy danych, szyny danych), infrastruktura sieciowa  i serwerowa, na której posadowione i użytkowane jest oprogramowanie (w tym: Środowisko Produkcyjne, Środowisko Testowe a także Środowisko Zapasowe aplikacji) oraz wszelka dokumentacja dotycząca wszelkich aspektów procesów budowy, rozwoju instalacji, odtwarzania, konfiguracji, użytkowania, rozwoju i utrzymania Systemu. |
| **Środowisko Deweloperskie** | Infrastruktura sprzętowo   * programowa Wykonawcy, która zapewnia Wykonawcy wykonywanie m.in. czynności: * wprowadzania zmian do Kodu Źródłowego Systemu; - tworzenia i uzupełniania Dokumentacji Systemu oraz Kodów Źródłowych; * wytwarzania wykonywalnej i instalacyjnej wersji Systemu dla Środowiska Testowego i Środowiska Produkcyjnego; * przeprowadzania testów realizowanych przez Wykonawcę w wersji instalacyjnej Systemu przed przystąpieniem do testów akceptacyjnych w Środowisku Testowym. |
| **Środowisko Produkcyjne** | Instancja Systemu działająca na infrastrukturze Zamawiającego wykorzystywana przez Użytkownika, na której przetwarzane są rzeczywiste dane, w tym dane osobowe i procesy. Środowisko Produkcyjne jest utrzymywane przez Wykonawcę w ramach Usługi Asysty Technicznej i Konserwacji |
| **Środowisko Preprodukcyjne** | Instancja Systemu działająca na infrastrukturze Zamawiającego. Środowisko informatyczne analogiczne do Środowiska Produkcyjnego w zakresie systemów operacyjnych, systemów bazodanowych oraz oprogramowania aplikacyjnego mogące się różnić od Środowiska Produkcyjnego mocą obliczeniową (liczba procesorów i RAM) oraz sposobem wirtualizacji, służące do testów wdrożeniowych dla użytkowników Zewnętrznych. Środowisko Preprodukcyjne jest  utrzymywane przez Wykonawcę w ramach Usługi Asysty Technicznej i Konserwacji |
| **Środowisko Zapasowe** | Instancja Systemu działająca na infrastrukturze Zamawiającego w lokalizacji innej niż Środowisko Produkcyjne, wykorzystywana przez Użytkownika w przypadku niedostępności Środowiska Produkcyjnego,  na której przetwarzane są rzeczywiste dane, w tym dane osobowe i procesy. Środowisko Zapasowe jest utrzymywane przez Wykonawcę w ramach Usługi Asysty Technicznej i Konserwacji. |
| **Środowisko Testowe** | Środowisko informatyczne Zamawiającego zapewniające pełne odwzorowanie warstwy funkcji Systemu posadowionego na Środowisku Produkcyjnym, analogiczne do Środowiska Produkcyjnego w zakresie systemów operacyjnych, systemów bazodanowych oraz oprogramowania aplikacyjnego mogące się różnić od Środowiska Produkcyjnego mocą obliczeniową (liczba procesorów i RAM) oraz sposobem wirtualizacji. |
| **Umowa** | Umowa zawarta między Zamawiającym, a Wykonawcą wraz ze wszystkimi aneksami i Załącznikami do Umowy. |
| **Usługa Asysty Technicznej i Konserwacji (ATiK)** | Wszelkie usługi związane z zapewnieniem bezawaryjnego działania Systemu, realizowane przez Wykonawcę w zakresie opisanym w Umowie. |
| **Usterka** | Wada niebędąca Awarią ani Błędem, powodująca zakłócenie pracy Systemu lub poszczególnych jego części mogąca mieć wpływ na jego funkcjonalność, natomiast nieograniczająca możliwości operacyjnych Systemu w sposób mogący mieć negatywny wpływ na jakość i terminowość realizacji zadań PFRON. |
| **Użytkownik** | Osoba korzystająca z Systemu lub jego poszczególnych części.  Użytkownicy są wewnętrzni (pracownicy PFRON) i użytkownicy zewnętrzni (użytkownicy Wnioskodawców). |
| **Wada** | Jakiekolwiek zaburzenie pracy Systemu objawiające się poprzez jego działanie w sposób odmienny od spodziewanego, przez co należy rozumieć między innymi:   * działanie odmienne od sposobu opisanego w Dokumentacji Systemu; * działanie odmienne od standardów lub zwyczajów wynikających z praktyki ustalonej w toku bieżącej eksploatacji i administracji Systemu; * działanie odmienne od sposobu ustalonego na mocy wszelkich innych dokumentów lub ustaleń Stron.   Wada może dotyczyć wszelkich możliwych nieprawidłowości w działaniu wszystkich komponentów Systemu, może dotyczyć jego dostępności, wydajności i reaktywności, cech mających wpływ na bezpieczeństwo i ciągłość działania, oraz wszystkich innych cech funkcjonalnych i pozafunkcjonalnych. Wady mogą mieć charakter Awarii, Błędu lub Usterki. |
| **WCAG** | Minimalne wymagania zapisane w złączniku do ustawy o dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych. |
| **Wnioskodawca** | Osoba fizyczna lub podmiot składająca wniosek w aplikacji Generator Wniosków. W przypadku podmiotu Wnioskodawcą nazywany jest podmiot w imieniu, którego składany jest wniosek. W imieniu podmiotu wniosek składa osoba lub osoby upoważnione do jego reprezentacji. |
| **Załącznik** | Każdy tekst, materiał graficzny lub też inny przedmiot, odnoszący się do treści głównego dokumentu, dołączony do niego w celu uzupełnienia, bądź uprawomocnienia jego treści. |
| **Zamówienie** | Przekazanie Wykonawcy zapotrzebowania na wykonanie określonych Produktów lub innych prac w ramach Modyfikacji i Rozwoju |
| **Zgłoszenie** | Przekazanie Wykonawcy zawiadomienia o Wadzie, złożenie pytań w ramach świadczenia Usługi Asysty Technicznej i Konserwacji oraz w okresie gwarancji. |

# **Ogólny opis zamówienia.**

* 1. Przedmiotem zamówienia jest świadczenie przez Wykonawcę na rzecz Zamawiającego:
     1. Usługi Asysty Technicznej i Konserwacji Systemu GW/EGW (dalej jako „ATiK”),

1. w ramach zamówienia gwarantowanego przez okres 24 miesięcy;
2. w ramach Opcji przez okres 24 miesięcy.
   * 1. Modyfikacji i Rozwoju Systemu GW/EGW w ramach maksymalnego limitu 10 000 Roboczogodzin w tym:
        1. w ramach zamówienia gwarantowanego 6 000 Roboczogodzin;
        2. w ramach zamówienia opcjonalnego 4 000 Roboczogodziny.

### **Gwarancja i rękojmia.**

Wykonawca udzieli Zamawiającemu gwarancji na okres 6 miesięcy liczonych od dnia zakończenia Umowy. Gwarancja wygasa przed upływem terminu wskazanego w zdaniu poprzednim w przypadku złożenia przez Zamawiającego Wykonawcy oświadczenia o przejęciu ATiK-u Systemu lub Modyfikacji i Rozwoju przez podmiot trzeci i zwolni Wykonawcę ze świadczenia usług gwarancyjnych. Gwarancja będzie świadczona z takimi samymi parametrami jak Usługa Asysty Technicznej i Konserwacji. Szczegóły dotyczące gwarancji i rękojmi zawierają postanowienia paragrafu 3 Umowy.

### **Prawa własności intelektualnej.**

Szczegóły i zasady dotyczące przeniesienia autorskich majątkowych prawa do Produktów oraz praw zależnych, a także udzielania i zapewniania licencji zawierają postanowienia paragrafu 8 Umowy.

### **Licencje.**

Wykonawca zobowiązuje się zapewnić Zamawiającemu licencje na korzystanie z Produktów, na warunkach i zasadach opisanych szczegółowo w paragrafie 8 Umowy.

### **Inne zobowiązania.**

Wykonawca zobowiązuje się wykonać inne zobowiązania na rzecz Zamawiającego, określone w Umowie i OPZ.

### **Szczegółowe zasady realizacji zobowiązań Wykonawcy.**

Niniejszy OPZ stanowi zestawienie ramowych wymagań niezbędnych do zrealizowania celu zamówienia. Lista wymagań zawarta w dokumencie stanowi opis zakresu zamówienia przedstawiony w sposób umożliwiający skalkulowanie wyceny przez Wykonawcę. Szczegółowe zasady realizacji zobowiązań Wykonawcy w ramach Przedmiotu Zamówienia, w tym zasady świadczenia usług/prac oraz kary umowne określa Umowa.

### **Zobowiązanie do stosowania regulacji wewnętrznych PFRON.**

Wykonawca zobowiązany jest do stosowania regulacji wewnętrznych PFRON w zakresie utrzymania i rozwoju systemów informatycznych PFRON. Dokumenty zawierające regulacje wewnętrzne PFRON zostaną przekazane Wykonawcy po zawarciu Umowy.

# **Informacje dotyczące Systemu GW/EGW.**

* 1. **Informacje ogólne o Systemie.**

Państwowy Fundusz Rehabilitacji Osób Niepełnosprawnych (PFRON) użytkuje Aplikacje wspierające realizację procesu zlecania zadań w ramach art. 36 ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych (Dz. U. z 2018 r. poz. 511, z późn. zm.).

* + 1. Generator Wniosków (GW) - to aplikacja służąca do przygotowywania i składania wniosków o dofinansowanie przez PFRON projektów realizowanych przez organizacje pozarządowe, dokonywania wstępnej oceny złożonych wniosków, publikowania wyników oceny wniosków oraz dokonywania aktualizacji wniosków i składania propozycji aneksów do umowy o dofinansowanie oraz służąca do przekazywania korespondencji do Wnioskodawców.
    2. System Ewidencji Godzin Wsparcia (System EGW) - to aplikacja służąca do zbierania i analizy danych o wsparciu udzielonym beneficjentom w ramach projektów realizowanych przez wnioskodawców. System umożliwia gromadzenie danych o wsparciu udzielonym beneficjentom ostatecznym projektów dofinansowanych przez PFRON. Wsparcie ewidencjonowane jest przez organizacje pozarządowe realizujące projekty, których celem jest wejście osób niepełnosprawnych (beneficjentów projektu) na rynek pracy lub zwiększenie samodzielności osób niepełnosprawnych.
    3. Elementy i funkcje Systemu:
       1. Generator Wniosków.

Składa się następujących modułów:

* + 1. Moduł użytkownika – przeznaczony dla wnioskodawców składających wnioski   
       o dofinansowanie projektów przez PFRON.
    2. Moduł oceny wniosków – przeznaczony dla ekspertów dokonujących oceny.
    3. Moduł administratora – obsługa aplikacji przez PFRON.
    4. Moduł użytkownika umożliwia:
       1. wypełnianie i składanie wniosków o dofinansowanie realizacji projektów w trybie konkursowym,
       2. eksport wniosków do formatu PDF,
       3. aktualizację wniosków na kolejne okresy finansowania i składanie propozycji aneksów,
       4. automatyczną weryfikację formalną wniosku przed złożeniem,
       5. uzyskiwanie informacji dotyczących procedowania wniosku przez PFRON.
    5. Moduł oceny wniosków umożliwia:

1. dokonywanie oceny merytorycznej wniosków,
2. eksport karty oceny do pliku PDF.
   * 1. Moduł administratora umożliwia:
3. nadawanie wnioskom statusów odpowiadających poszczególnym krokom procedury (etapom życia wniosku) wywołującym określone reakcje aplikacji, np. wprowadzenie wniosku w tryb edycji w celu wprowadzenia zmian,
4. wysłanie pism w ramach realizowanych umów do Wnioskodawców,
   * + 1. wskazywanie realizatora umowy – oddziału PFRON,
       2. przygotowywanie list rankingowych i zestawień,
       3. eksport wybranych informacji z bazy danych do pliku \*.mdb.
       4. System Ewidencji Godzin Wsparcia.

Składa się z następujących modułów:

1. Moduł użytkownika – przeznaczony dla projektodawców (Wnioskodawców, posiadających umowę z PFRON) dokumentujących realizację projektów dofinansowanych przez PFRON,
2. Moduł administratora – obsługa Systemu EGW przez PFRON,
3. Moduł użytkownika umożliwia:
   * + 1. edycję danych projektodawcy,
       2. rejestrację i edycję danych beneficjentów ostatecznych projektów,
       3. rejestrację i edycję danych pracowników projektów,
       4. zarządzanie kontami użytkowników projektodawcy – tworzenie, edycja, aktywacja i dezaktywacja, nadawanie uprawnień (trzy poziomy uprawnień, macierz uprawnień szczegółowych dla najniższego poziomu),
       5. rejestrację i edycję wsparcia beneficjentów – podstawowa operacja wykonywana cyklicznie w Systemie EGW. Atrybutami wsparcia są data zdarzenia, czas trwania, forma wsparcia, pracownicy udzielający i beneficjenci otrzymujący. Operacja polega na wskazaniu zasobów biorących udział oraz podaniu daty i czasu. Szczegóły przedstawione są na schemacie poniżej.
       6. wykonywanie raportów z rejestrowanych danych oraz ich eksport do pliku .xls,
       7. porównywanie wartości wskaźników planowanych we wniosku (importowanych z generatora wniosków) z wartościami wskaźników osiągniętymi na podstawie danych rejestrowanych w Systemie EGW, 8. eksport bazy danych projektu do pliku \*.mdb.

4. Moduł administratora umożliwia:

* + - 1. wykonywanie wszelkich operacji dostępnych dla użytkowników,
      2. zarządzanie uprawnieniami wszystkich użytkowników,
      3. publikowanie plików pomocy oraz tworzenie i edycję pomocy kontekstowej dla poszczególnych ekranów,
      4. tworzenie statystyk z danych gromadzonych w Systemie EGW,

5. eksport wybranych informacji z bazy danych do pliku \*.mdb.

Dostęp do informacji przez poszczególnych użytkowników Aplikacji przedstawia poniższy schemat:

Obraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

* 1. Opis wymagań na infrastrukturę.
     1. Parametry techniczne instalacji produkcyjnej i testowej:

Tabela nr 1. Minimalne wymagania - stan obecny

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Generator Wniosków** | **System EGW** | **System EGW (WCAG 2.0) \*** |  |  |
|  |  |
| **Kod napisany w  PHP** | 1 406 342 | 1 068 561 | 1 562 538 |  |  |
| **Kod napisany w HTML** | 608 022 (html) + | 31 650 (html) + | 610 722 (html) + |  |  |
| 262 458 (phtml) + | 29 284 (phtml) + | 45 522 (phtml) + |  |  |
| 192 492 (tpl) | 0 (tpl) | 28 142  (tpl) |  |  |
| **Arkusze stylów CSS** | 1 101 222 | 6 964 | 1 097 419 |  |  |
| **JS** | 1 450 033 | 47 065 | 1 318 122 |  |  |
| **XML** | 278 159 | 240 967 | 247 214 |  |  |
| **YAML** | 4184 (yml) + 30 449 (yaml) | 3897(yml) | 814 (yml) |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | **01-02-2018** | **22-04-2021** | **10.04.2024** |  |  |
| Liczba użytkowników zarejestrowanych w Generatorze Wniosków | 2 204 | **3 023** | 3 711 |  |  |
| Liczba użytkowników zarejestrowanych w Systemie EGW | 2 477 | **4 530** | 5 245 |  |  |
| Liczba wnioskodawców zarejestrowanych w Generatorze Wniosków | 1 752 | 2 179 | 2 592 |  |  |
| Liczba wniosków utworzonych w Generatorze Wniosków | 7 151 | 10 274 | 13 293 |  |  |
| Liczba wszystkich wersji wniosków w Generatorze Wniosków | 21 469 | 30 759 | 41 933 |  |  |
| Liczba projektodawców zarejestrowanych w Systemie EGW | 601 | 655 | 801 |  |  |
| Liczba projektów obsługiwanych przez System EGW | 2 002 | 2 736 | 4 525 |  |  |
| Liczba beneficjentów zarejestrowanych w Systemie EGW (nieunikatowych) | 209 063 | 252 302 | 312 682 |  |  |
| Liczba pracowników projektodawców | 34 917 | 50 104 | 69 470 |  |  |
| zarejestrowanych w Systemie EGW (nieunikatowych) |  |  |
| Liczba wszystkich zarejestrowanych rekordów wsparcia | 14 094 100 | 22 257 347 | 34 250 687 |  |  |
| Wielkość bazy danych Generatora Wniosków w MB | 2 723 | 11 900 MB | 41 984 MB |  |  |
| Wielkość bazy danych Systemu EGW w MB | 19 456 | 38 000 MB | 63 488 MB |  |  |
| Maksymalna liczba logowań do Generatora Wniosków odnotowanych w ciągu jednej godziny | 1 260 | 1 260 | 1 260 (2017-10-13) |  |  |
| Maksymalna liczba logowań do Systemu EGW odnotowanych w ciągu jednej godziny | 262 | 262 | 322 (2019-04-29) |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **Węzeł** | **Typ serwera** | **Parametry techniczne maszyny wirtualnej** | **Oprogramowanie** |  |  |
| **Środowisko Produkcyjne** | Serwer bazodanowy | CPU: 8 vCore 2,5 GHz | OS: Debian 11 |  |  |
| RAM: 32 GB | PostgreSQL 9.4 |  |  |
| Pojemność dyskowa: 690 GB |  |  |  |
|  |  |  |  |
| Serwer aplikacyjny | CPU: 16 vCore 2,5 GHz | OS: Windows Server 2022 |  |  |
| RAM: 64 GB |  |  |
| Serwer Webowy | Pojemność dyskowa: 900 GB | Serwer http IIS w wersji 10 |  |  |
| **Środowisko Testowe** | Serwer bazodanowy | CPU: 8 vCore 2,2 GHz | OS: Debian 11 |  |  |
| RAM: 32 GB | PostgreSQL 9.4 |  |  |
| Pojemność dyskowa: 1690 GB |  |  |  |
| Serwer aplikacyjny | CPU: 8 vCore 2,2 GHz | OS: Windows Server 2022 |  |  |
| RAM: 32 GB |  |  |
| Serwer Webowy | Pojemność dyskowa: 900 GB | Serwer http IIS w wersji 10 |  |  |

System GW/EGW posiada Środowisko Zapasowe, które jest znajduje się w infrastrukturze innej niż Środowisko Produkcyjne i na którym została skonfigurowana replikacja IIS i baz danych PostgreSQL między środowiskiem produkcyjnym oraz testowym EGW/GW.

# **Wymagania funkcjonalne.**

### **Wymagania dotyczące Usługi Asysty Technicznej i Konserwacji.**

### **Wymagania ogólne.**

W ramach Usług Asysty Technicznej i Konserwacji Wykonawca zobowiązany jest do:

1. Zapewnienia ciągłości działania Systemu przez 24 godziny 7 dni w tygodniu 365 dni w roku („24/7/365”) przez cały okres obowiązywania Umowy z wyłączeniem Okna Serwisowego, pod warunkiem, że w ramach Okna Serwisowego realizowane są prace serwisowe wymagające wyłączenia Systemu lub powodujące tymczasową niedostępność Systemu i poszczególnych jego funkcjonalności.
2. Utrzymania i administracji Sytemu w tym Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego.
3. Utrzymania wartości parametrów związanych z Usługą Asysty Technicznej i Konserwacji na warunkach opisanych w Załączniku nr 5 „Poziom świadczenia usług SLA” do Opisu Przedmiotu Zamówienia.
4. Zapewnienia utrzymania parametrów wydajnościowych Systemu na poziomie określonym w Załączniku nr 1 do Opisu Przedmiotu Zamówienia, pod warunkiem, że w tym czasie nie są prowadzone Prace Serwisowe.
5. Zapewnienia wysokiego poziomu bezpieczeństwa Systemu i danych w nim przetwarzanych, między innymi poprzez instalowanie poprawek bezpieczeństwa dla Systemu, w tym do Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego w terminie 3 Dni Roboczych od dnia wydania ich przez producenta, wprowadzanie zmian konfiguracyjnych w Systemie, mających na celu zwiększenie poziomu bezpieczeństwa, zapewnienia zgodności z wymaganiami ujętymi w rozporządzeniu KRI, oraz dokumentach wewnętrznych Funduszu - Polityce Bezpieczeństwa Teleinformatycznego, Polityce Przetwarzania Danych Osobowych i Polityce Bezpieczeństwa Informacji. W szczególnych przypadkach, Zamawiający dopuszcza możliwość wydłużenia terminu wskazanego w zdaniu poprzednim, pod warunkiem przedstawienia przez Wykonawcę uzasadnienia. Na zmianę terminu musi wyrazić zgodę Zamawiający. Jeżeli realizacja ww. dostosowania Systemu będzie wymagała jego czasowego wyłączenia, wówczas na ten czas zawieszany jest ATK-01.
6. Przyjmowania Zgłoszeń i Naprawy Wad Systemu.
7. Usuwania Wad Systemu wszystkich kategorii zgodnie z wymaganiami opisanymi w pkt 4.1.2 Opisu Przedmiotu Zamówienia.
8. Wydawania rekomendacji dotyczących przeprowadzania zmian, aktualizacji i modernizacji Systemu.
9. Realizacja Zgłoszeń dotyczących zaleceń powstałych w wyniku audytu bezpieczeństwa teleinformatycznego. Jeżeli realizacja w/w zaleceń będzie wymagała czasowego wyłączenia Systemu, wówczas na ten czas zawieszany jest ATK-01.
10. Zapewnienia stałej opieki co najmniej jednego konsultanta do wsparcia przy rozwiązywaniu bieżących problemów związanych z funkcjonowaniem Systemu.
11. Bieżącej aktualizacji Dokumentacji Systemu oraz Kodów Źródłowych Systemu, przechowywanych w Repozytorium Projektu, zgodnie z wymaganiami opisanymi w Załączniku nr 3 do Opisu Przedmiotu Zamówienia. Wykonawca ma obowiązek wraz z Protokołem Odbioru usługi dostarczyć zaktualizowaną Dokumentację Systemu i Kody Źródłowe oraz wskazać zmiany, jakie zostały wprowadzone w ramach Usługi Asysty Technicznej i Konserwacji w okresie, za który przedstawia Protokół Odbioru.
12. Zrealizowania raz na kwartał przeglądu Kodów Źródłowych i Dokumentacji Systemu zgodnie z wymaganiami opisanymi w Załączniku nr 3 do Opisu Przedmiotu Zamówienia.
13. Realizacji Zgłoszeń dotyczących zaleceń powstałych w wyniku audytu WCAG oraz dostosowanie Systemu do wymagań opisanych w Załączniku nr 2 do Opisu Przedmiotu Zamówienia, przez cały okres trwania Umowy. Jeżeli realizacja w/w zaleceń będzie wymagała czasowego wyłączenia Systemu, wówczas na ten czas zawieszany jest ATK-01.
14. Aktualizacji warstw Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego nie później niż miesiąc po udostępnieniu przez producentów danego oprogramowania nowej, stabilnej jego wersji po wcześniejszym uzgodnieniu z Zamawiający i w terminie na jaki wyrazi zgodę Zamawiający. Wyżej wymieniony termin może zostać w szczególnych przypadkach zmieniony przez Zamawiającego na dłuższy. W przypadku krytycznych poprawek bezpieczeństwa wymaga się ich niezwłocznej instalacji. Wymóg nie dotyczy aktualizacji, do których instalacji konieczne będzie poniesienie przez Wykonawcę dodatkowych kosztów z tytułu zakupu licencji – wówczas koszty i decyzję o instalacji ponosi Zamawiający. Na czas instalacji ww. poprawek zawieszone jest ATK-01.
15. Instalowania na Środowisku Produkcyjnym i Środowisku Testowym, w czasie Okna Serwisowego, o ile Strony nie uzgodnią inaczej, Pakietów Aktualizacyjnych usuwających Wady mając na uwadze, że na czas instalacji zawieszone jest ATK-01. W przypadku błędów związanych z bezpieczeństwem Systemu, termin instalacji Pakietu Aktualizacyjnego musi zostać uzgodniony niezwłocznie. Instalacja takiego Pakietu może być wykonana poza Oknem Serwisowym.
16. Instalacji Pakietu Aktualizacji w ramach Rozwoju, z zastrzeżeniem wymagań dotyczących Oprogramowania Obcego lub Narzędziowego, realizowana będzie w terminie uzgodnionym z Zamawiającym, w czasie Okna Serwisowego, o ile Strony nie uzgodnią inaczej.
17. Usługi migracji Systemu realizowanej w opisany poniżej sposób:  
    1. Zamawiający może zgłosić Wykonawcy usługę migracji Systemu nie więcej niż 4 razy w czasie trwania umowy.  
    2. Zamawiający zgłasza usługę migracji jak każde zadanie w ramach usługi ATiK w systemie do obsługi zgłoszeń (np. JIRA).  
    3. Wykonawca po zleceniu przez Zamawiającego usługi migracji w systemie JIRA w pierwszej kolejności definiuje harmonogram prac oraz wymagania na nową infrastrukturę techniczną i sieciową. Zamawiający dopuszcza realizację migracji poprzez przeniesienie maszyn wirtualnych z jednego centrum przetwarzania danych do drugiego.   
    4. Po otrzymaniu niezbędnych danych od Wykonawcy i akceptacji ich przez Zamawiającego, Wykonawca rozpoczyna proces migracji.  
    5. Zamawiający, w przypadku utworzenia nowych maszyn, przekazuje Wykonawcy dostępy do nowo skonfigurowanej infrastruktury.  
    6. Dalsza obsługa Zgłoszenia dotyczącego usługi migracji jest realizowana jak każde zlecenie w ramach usługi ATiK opisanej w pkt 4.1.2 poniżej.
18. W Dni Robocze w Godzinach Roboczych Wykonawca musi realizować usługi od ATK-01 do ATK-14. Ponadto, Wykonawca na każde żądanie Zamawiającego zobowiązany jest do realizacji usługi opisanej w ATK-15, ATK-16.
19. W Dni Robocze pomiędzy Godzinami Roboczymi a Oknem Serwisowym Wykonawca musi realizować usługi od ATK-01, ATK-02, ATK-03, ATK-4, ATK-05, ATK-06. Ponadto, Wykonawca na każde żądanie Zamawiającego zobowiązany jest do realizacji usługi opisanej w ATK-15, ATK-16.
20. W Dni Robocze w Oknie Serwisowym Wykonawca musi realizować usługi ATK-01, ATK-02, ATK-03, ATK-4, ATK-05, ATK-06, ATK-10, ATK-15, ATK-16.
21. W Dni Świąteczne i Ustawowo Wolne Od Pracy Wykonawca musi realizować usługi ATK-01, ATK-02, ATK-03, ATK-04, ATK-05, ATK-06.
22. W Dni Świąteczne i Ustawowo Wolne od pracy oraz w Oknie Serwisowym Wykonawca musi realizować usługi: ATK-01, ATK-02, ATK-03, ATK-04, ATK-05, ATK-06, ATK-10, ATK-15,   
    ATK-16.

### **Zasady obsługi Zgłoszeń.**

1. Zgłoszenie dokonywane jest za pośrednictwem Portalu Serwisowego przez upoważnionych Pracowników Wykonawcy oraz Zamawiającego.
2. Wszystkie Zgłoszenia muszą być przez Strony rejestrowane i prezentowane w Portalu Serwisowym, w sposób pozwalający na archiwizację danych o czasie i treści Zgłoszeń oraz Obejścia i Naprawy Wad.
3. Jeżeli Wada została wykryta przez Wykonawcę, Wykonawca niezwłocznie poinformuje Zamawiającego o wystąpieniu Wady, zarejestruje Zgłoszenie, nada Wadzie odpowiednią kategorię oraz przystąpi do działań zmierzających do usunięcia Wady, z tym zastrzeżeniem, że ostateczna decyzja dotycząca uznania kategorii Wady należy do Zamawiającego.
4. Zgłoszenie Wady musi zawierać:
5. opis funkcjonalności Systemu, której dotyczy Wada;
6. opis zauważonych nieprawidłowości w działaniu Systemu, jeśli jest to możliwe, ilustrowanych zrzutami ekranów Systemu oraz krótkim scenariuszem sposobu uzyskania nieprawidłowości;
7. kategorię Wady.
8. W przypadku, gdy Zgłoszenie zostanie uznane przez Wykonawcę za niezasadne lub w przypadku uznania, iż Zamawiający w sposób nieprawidłowy określił kategorię Wady, Wykonawca zobowiązany jest do poinformowania Zamawiającego o wyniku analizy Zgłoszenia, przy czym ostateczna decyzja, co do realizacji oraz co do kwalifikacji określonej Wady należy do Zamawiającego.
9. Przyjmuje się, że do skutecznego Zgłoszenia Wady dochodzi z chwilą zarejestrowania Wady w Portalu Serwisowym i zaadresowanie jej do Wykonawcy.
10. W wyjątkowych sytuacjach, gdy Portal Serwisowy jest niedostępny, Zamawiający dopuszcza możliwość przekazania Zgłoszenia drogą telefoniczną lub mailową, na adres wskazany do komunikacji pomiędzy Stronami oraz w ten sam sposób zatwierdzenie Zgłoszenia i jego dalsze procedowanie. W chwili przywrócenia dostępności Portalu Serwisowego, Wykonawca jest zobowiązany do niezwłocznego uzupełnienia Zgłoszenia w Portalu Serwisowym. W sytuacji opisanej w zdaniu pierwszym, przyjmuje się, że do skutecznego Zgłoszenia Wady dochodzi z chwilą przekazania Wykonawcy Zgłoszenia drogą telefoniczną lub mailową, na adres wskazany do komunikacji pomiędzy Stronami.
11. Po otrzymaniu Zgłoszenia Wykonawca potwierdzi istnienie i kategorię Wady oraz przystąpi do jej Naprawy.
12. Jeśli Wykonawca stwierdzi w trakcie działań naprawczych, że dla dokonania usunięcia Wady niezbędne jest podjęcie przez Zamawiającego określonych czynności lub uzyskania dodatkowych wyjaśnień od Zamawiającego, Wykonawca niezwłocznie zwróci się do Zamawiającego z żądaniem wykonania odpowiednich działań. Czas na dokonanie odpowiednich działań przez Zamawiającego nie będzie wliczany do Czasu Naprawy Wady.
13. Usunięcie Wady nie może prowadzić do naruszenia struktur i integralności danych, do utraty danych lub wpływać negatywnie na funkcjonowanie Systemu lub innych składników infrastruktury Zamawiającego. Wykonawca zobowiązuje się również do usunięcia Wad w sposób zapobiegający utracie jakichkolwiek danych. W przypadku, gdy wykonanie usługi wiąże się z ryzykiem utraty danych, Wykonawca zobowiązany jest poinformować o tym Zamawiającego przed przystąpieniem do usunięcia Wady.
14. Usunięcie Wady nie może powodować braku zgodności z zaleceniami WCAG opisanymi w Załączniku nr 2 do Opisu Przedmiotu Zamówienia.
15. Wykonawca przed zainstalowaniem Pakietu Aktualizacji na Środowisku Testowym wykona testy wewnętrzne zgodnie z Załącznikiem nr 4 do Opisu Przedmiotu Zamówienia.
16. Zainstalowanie przez Wykonawcę Pakietu Aktualizacji usuwającego Wadę na Środowisku Testowym uznaje się za zgłoszenie przez Wykonawcę gotowości do Odbioru Pakietu Aktualizacji. Zainstalowanie przez Wykonawcę Pakietu Aktualizacji usuwającego Wadę na Środowisku Produkcyjnym może się odbyć wyłącznie za zgodą Zamawiającego. W wyjątkowych sytuacjach za zgodą Zamawiającego Wykonawca może zainstalować Pakiet Aktualizacji bezpośrednio na Środowisku Produkcyjnym.
17. Po zgłoszeniu gotowości Odbioru Pakietu Aktualizacji Zamawiający przystąpi niezwłocznie do jego weryfikacji.
18. Zamawiający ma prawo do weryfikacji należytego wykonania usługi dowolną metodą. Zamawiający ma w szczególności prawo przeprowadzić testy za pomocą samodzielnie zdefiniowanych scenariuszy testowych lub przez zaangażowanie podmiotu trzeciego działającego w imieniu Zamawiającego.
19. W przypadku, gdy Pakiet Aktualizacji nie usunie zgłoszonej Wady lub spowoduje pojawienie się nowej Wady w Systemie, Zgłoszenie uznaje się za niezakończone.
20. Do Czasu Naprawy Zgłoszenia nie są wliczane okresy potwierdzania przez Zamawiającego skuteczności dostarczonych poprawek oraz za zgodą Zamawiającego czas pomiędzy odbiorem przez Zamawiającego Pakietu Aktualizacji na Środowisku Testowym a zainstalowaniem Pakietu Aktualizacji na Środowisku Produkcyjnym.
21. Wykonawca zobowiązany jest do zainstalowania Pakietu Aktualizacji najpóźniej w najbliższym Oknie Serwisowym po dokonaniu odbioru przez Zamawiającego Pakietu Aktualizacji, chyba że Zamawiający postanowi inaczej.
22. Jeżeli Wykonawca nie dokona Naprawy / Obejścia w terminach, o których mowa w Załączniku nr 5 „Poziom świadczenia usług SLA” Opisu Przedmiotu Zamówienia, Zamawiający może:
23. zawiadamiając uprzednio Wykonawcę, usunąć Wadę we własnym zakresie lub powierzyć jej usunięcie innemu podmiotowi trzeciemu na koszt Wykonawcy, co nie spowoduje utraty przysługujących Zamawiającemu uprawnień z tytułu gwarancji - przy czym koszty poniesione przez Zamawiającego przy usunięciu Wady będą potrącone z wynagrodzenia przysługującego Wykonawcy lub z zabezpieczenia należytego wykonania przedmiotu Umowy;
24. obciążyć Wykonawcę karą umowną na zasadach opisanych w przyszłej Umowie.
25. Zakończenie instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym kończy obsługę Zgłoszenia.
26. Zamknięcie Zgłoszenia w Portalu Serwisowym dokonywane jest po instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym przez upoważnionych Pracowników Zamawiającego wskazanych w przyszłej Umowie.
27. Wykonawca zobowiązany jest do uzupełnienia Zgłoszenia w Portalu Serwisowym o informacje na temat przyczyn wystąpienia Wady oraz szczegółowego opisu sposobu jej usunięcia z Systemu. Zamawiający dopiero po uzyskaniu powyższych informacji przystąpi do zamknięcia Zgłoszenia.
28. W przypadku nieuzupełnienia Zgłoszenia o wymagane w punkcie ATK-44 informacje Zamawiający nie podpisze Protokołu Odbioru Usługi Asysty Technicznej i Konserwacji za dany okres rozliczeniowy.
29. W ciągu 10 Dni Roboczych od zakończenia obsługi każdego zgłoszenia dotyczącego Wady, aktualizacji Systemu, czy konsultacji - Wykonawca dostarczy Zamawiającemu w formie elektronicznej zaktualizowaną, wersję Kodów Źródłowych Systemu oraz wersję Dokumentacji Systemu aktualizującą jej części lub całość zgodnie z zasadami opisanymi w załączniku nr 5 Opisu Przedmiotu Zamówienia. Zmiany wprowadzane przez Wykonawcę do Dokumentacji Systemu będą oznaczone wyraźnie oraz w sposób umożliwiający Zamawiającemu ich zidentyfikowanie oraz wyszukanie w tekście, w szczególności poprzez zastosowanie trybu śledzenia zmian. Wykonawca będzie aktualizował Dokumentację Systemu oraz Kody Źródłowe w ramach wynagrodzenia, o którym mowa w paragrafie 9 ust.1 Umowy, oraz w sposób umożliwiający weryfikację przez Zamawiającego wprowadzonych zmian i na zasadach opisanych w Załączniku nr 5 do Opisu Przedmiotu Zamówienia.
30. W przypadku stwierdzenia niespójności pomiędzy Dokumentacją Systemu a działaniem Systemu Wykonawca zobowiązany jest do usunięcia niespójności w terminie 4 Dni Roboczych od pozyskania informacji.
31. Wykonawca zobowiązany jest sporządzać comiesięczny Raport, którego szablon stanowi załącznik do Protokołu Odbioru ATiK-u. Zamawiający zastrzega sobie prawo do modyfikacji Raportu w trakcie obowiązywania Umowy, na co Wykonawca wyraża zgodę.

### **Zasady udzielania stałych konsultacji.**

1. Konsultacje zgłaszane są w formie Pytań za pośrednictwem Portalu Serwisowego przez upoważnionych Pracowników Zamawiającego wskazanych w Umowie.
2. W wyjątkowych sytuacjach, gdy Portal Serwisowy jest niedostępny, Zamawiający dopuszcza możliwość przekazania Pytań drogą telefoniczną lub mailową, na adres wskazany do komunikacji pomiędzy Stronami oraz w ten sam sposób zatwierdzenie Pytań i ich dalsze procedowanie. W chwili przywrócenia dostępności Portalu Serwisowego, Wykonawca jest zobowiązany do niezwłocznego uzupełnienia Pytań w Portalu Serwisowym.
3. Konsultacje udzielane są za pośrednictwem Portalu Serwisowego przez upoważnionych Pracowników Wykonawcy wskazanych w przyszłej Umowie.
4. Wszystkie materiały z konsultacji muszą być przez Strony rejestrowane i prezentowane w Portalu Serwisowym w sposób pozwalający na archiwizację danych o czasie i treści konsultacji (zapytań i odpowiedzi).
5. Przyjmuje się, że do skutecznego zawiadomienia dochodzi z chwilą zarejestrowania i zaadresowania Zgłoszenia Pytania w Portalu Serwisowym.
6. Jeżeli Wykonawca nie będzie w stanie udzielić odpowiedzi w czasie określonym w Załączniku nr 5 -„Poziom świadczenia usług SLA” do Opisu Przedmiotu Zamówienia, jest zobowiązany powiadomić o tym fakcie Zamawiającego, z którym zostanie ustalony nowy termin udzielenia odpowiedzi.
7. Jeżeli udzielenie odpowiedzi będzie wymagało przez Wykonawcę kontaktu z podmiotem trzecim (Użytkownikiem zewnętrznym), w szczególności za pośrednictwem poczty elektronicznej, telefonicznie, Wykonawca niezwłocznie poinformuje o tym fakcie Zamawiającego i uzyska jego zgodę.
8. W ramach udzielonych odpowiedzi dotyczących Przypadków Szczególnych, Wykonawca opracuje i udostępni Zamawiającemu instrukcję opisującą rozwiązanie danego Przypadku Szczególnego.

### **Zasady aktualizacji Systemu.**

1. Aktualizacja Systemu realizowana jest dla: nowych wersji Systemu wytworzonych w związku ze zmianami Sprzętu i Oprogramowania Systemowego i Narzędziowego; nowych wersji lub uaktualnień Systemu lub jego poszczególnych części w ramach wersji głównej Systemu lub części Systemu, utworzonych z własnej inicjatywy przez Wykonawcę z uwzględnieniem zapisów poniżej, jako kolejne wersje Systemu lub części Systemu, zawierające usprawnienia w porównaniu z poprzednimi wersjami Sytemu lub części Sytemu; dostosowania Systemu do bezwzględnie obowiązujących przepisów prawa wpływających na sposób funkcjonowania oraz funkcjonalności Systemu, w tym również określających minimalne wymagania techniczne dla systemów informatycznych eksploatowanych przez Zamawiającego.
2. Jeżeli Wykonawca opracuje samodzielnie, niezależnie od zobowiązań wynikających z zamówienia jakiekolwiek aktualizacje polegające na uaktualnieniu Systemu, służące do usunięcia stwierdzonych nieprawidłowości pracy Systemu, dodania nowych funkcjonalności lub uwzględnienia zmian w przepisach prawa - Wykonawca zobowiązany jest niezwłocznie do poinformowania Zamawiającego o fakcie opracowania powyższych uaktualnień oraz ich przedstawienia. Wykonawca zobowiązany jest również poinformować Zamawiającego o ewentualnych skutkach zainstalowania Pakietu Aktualizacji, w szczególności ich wpływie na sposób jego funkcjonowania oraz sposób korzystania z Systemu.
3. Zasady aktualizacji Systemu obejmują również aktualizację Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego.
4. Aktualizacja Systemu przez Wykonawcę obejmuje w szczególności:
5. przygotowanie i uzgodnienie z Zamawiającym planu wdrożenia wersji Systemu, aby Zamawiający z odpowiednim wyprzedzeniem mógł poinformować Użytkowników wewnętrznych i zewnętrznych o przerwie w działaniu Systemu i planowanym zakresie aktualizacji;
6. dostarczenie aktualizacji;
7. instalację aktualizacji na Środowiskach Testowych;
8. instalację aktualizacji na Środowisku Produkcyjnym;
9. testy Systemu na Środowisku Produkcyjnym, Środowiskach Testowych;
10. wsparcie przy uruchamianiu Systemu na wyżej wymienionych środowiskach;
11. aktualizacje Dokumentacji Systemu oraz Kodów Źródłowych w formie elektronicznej;
12. podniesienie numeru wersji Systemu.

### **Zasady zapewnienia kontroli i ciągłości działania Systemu oraz okresowych przeglądów.**

1. W ramach przedmiotu zamówienia Wykonawca będzie realizował prace związane z utrzymaniem, konserwacją, administracją i aktualizacją systemów operacyjnych oraz oprogramowania firm trzecich (w tym w szczególności silników baz danych, serwerów aplikacyjnych oraz bibliotek programistycznych i narzędzi), które wykorzystywane są do prawidłowego działania oprogramowania dziedzinowego podlegające Usłudze ATiK-u, a w szczególności będzie realizował prace związane z:
2. monitorowaniem prawidłowości działania ww. systemów oraz oprogramowania firm trzecich. W przypadku zidentyfikowania niedostatecznej ilości zasobów Wykonawca zwróci się do Zamawiającego z wnioskiem o przydzielenie dodatkowych zasobów wraz ze wskazaniem ilości oraz określeniem powodu powstania ww. zapotrzebowania. Jeśli wskazane zasoby będą dostępne, Zamawiający przydzieli zasoby w terminie nie dłuższym niż 10 Dni Roboczych od prawidłowo przedłożonego zapotrzebowania. Za prawidłowo złożone zapotrzebowanie Zamawiający rozumie przekazanie za pośrednictwem kanału komunikacyjnego wskazanego w przyszłej Umowie informacji zawierających parametr podlegający zmianie oraz powód zmiany (muszą one zawierać się w zamkniętym katalogu parametrów konfiguracyjnych maszyn wirtualnych właściwym dla ww. wirtualizatora). O zakończeniu realizacji wniosku Zamawiający poinformuje Wykonawcę w sposób analogiczny do ww. Po przydzieleniu przez Zamawiającego dodatkowych zasobów w celu ich skutecznego wykorzystania Wykonawca dokona czynności rekonfiguracyjnych po stronie Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu. W/w czynności realizowane przez Wykonawcę muszą zostać zrealizowane w terminie nie dłuższym niż 10 Dni Roboczych od momentu poinformowania Wykonawcy o dostępności dodatkowych zasobów;
3. uaktualnianiem Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu do wersji aktualnie wspieranej. Przez uaktualnienie do wersji aktualnie wspieranych Zamawiający rozumie czynności związane z podniesieniem wersji Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu oraz wykonanie testów na Środowiskach Testowych i Produkcyjnym do wersji stabilnych posiadających aktualne wsparcie producenta tzn. posiadających możliwość pobierania i aktualizowania oprogramowania ze stron lub z repozytoriów udostępnianych przez producenta oraz wprowadzania wszystkich zalecanych przez producenta uaktualnień, w szczególności uaktualnień dotyczących zabezpieczeń;
4. instalowaniem poprawek i łat bezpieczeństwa dla Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu;
5. Zarządzaniem konfiguracją poszczególnych elementów Systemu oraz wersji Oprogramowania Systemowego i Narzędziowego, Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego w celu optymalizowania działania i zapewnienia ciągłości działania;
6. administrowaniem Oprogramowaniem Systemowym i Narzędziowym oraz Oprogramowaniem Standardowym/Obcym, Oprogramowaniem Zamawiającego oraz Systemem, w tym w szczególności dostosowywanie ww. oprogramowania w zakresie zapewniania oczekiwanego poziomu optymalizacji działania wyżej wskazanego oprogramowania;
7. analizowaniem oraz przygotowanie wytycznych w zakresie możliwości rozwojowych, realizacji zmian technologicznych mających na celu optymalizację pracy Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego, Oprogramowania Zamawiającego oraz Systemu z jednoznacznym wskazaniem możliwości migracji do wskazanych przez Zamawiającego rozwiązań, w tym w szczególności opis czynności do wykonania, przewidywaną pracochłonność oraz potencjalne występujące ryzyka;
8. administrowaniem certyfikatami służącymi do integracji Systemu z innymi systemami zewnętrznymi i wewnętrznymi.
9. okresowym przeglądem kopii zapasowych. Wykonawca, w cyklach 6 miesięcznych będzie przeprowadzał przegląd kopii zapasowych Systemu polegający na testowym odtworzeniu Systemu z kopii zapasowych na środowisko wskazane przez Zamawiającego. Odtworzeniu podlegać będzie zarówno serwery aplikacyjne jak i serwery bazodanowe wraz z danymi.
10. Wykonawca, w terminie 10 Dni Roboczych liczonych od następnego dnia od przekazania Wykonawcy dostępu, zweryfikuje konfigurację i działanie obecnie wykorzystywanego u Zamawiającego narzędzia do monitorowania zasobów (np. Zabbix) a następnie w ciągu 20 Dni Roboczych na podstawie paramentów i agentów w istniejącym Zabbixie utworzy nowy projekt dedykowany Systemowi GW/EGW, którym będzie zarządzał w ramach usługi ATiK. W przypadku zmiany narzędzia na nowe, Wykonawca w terminie 20 Dni Roboczych skonfiguruje projekt dedykowany Systemowi GW/EGW. Wykonawca odpowiedzialny będzie za aktualizację narzędzia, jego utrzymanie, konfigurację agentów i parametrów. Wykonawca zobowiązany będzie do przedstawienia raportu z wykonanych prac.
11. Wykonawca określi wszystkie parametry konfiguracyjne polityk archiwizacji danych Oprogramowania objętego ATiK-iem umożliwiających odtworzenie danych i uruchomienie wszystkich komponentów Oprogramowania. Dostarczone parametry konfiguracyjne muszą uwzględniać minimalizację parametrów RPO (Recovery Point Objective) oraz RTO (Recovery Time Objective). Na podstawie uzyskanych informacji Zamawiający przygotuje nowe lub zmodyfikuje istniejące zadania archiwizacyjne, a Wykonawca zweryfikuje i potwierdzi poprawność ich konfiguracji oraz działania. W/w określenie parametrów nastąpi nie później niż w ciągu 20 Dni Roboczych liczonych od następnego dnia od przekazania Wykonawcy dostępu do Systemu.
12. Wykonawca zobowiązany jest do okresowego analizowania i weryfikowania prawidłowości działania wszystkich zadań archiwizacyjnych. Czynności te winny być prowadzone nie rzadziej niż raz na trzy miesiące lub po każdej zmianie/modyfikacji konfiguracji polityk archiwizacji danych. Na wniosek oraz w porozumieniu z Wykonawcą, Zamawiający wskaże termin przeprowadzenia w/w prac. Nie może być on jednak dłuższy niż 21 dni od zgłoszenia przez Wykonawcę gotowości do dokonania w/w czynności. Każda weryfikacja musi zostać potwierdzona obustronnie zawartym protokołem odbioru bez uwag. Zamawiający w terminie 5 Dni Roboczych od otrzymania protokołu zaakceptuje go lub zgłosi uwagi. W terminie do 14 dni kalendarzowych Wykonawca zobligowany jest do usunięcia przyczyn powstania uwag wskazanych w Protokole Odbioru. Po usunięciu przyczyn powstania uwag proces odbioru zostanie powtórzony. Zamawiający dopuszcza dwukrotne powtórzenie czynności odbiorowych.
13. Wykonawca zobowiązany jest do przeprowadzania okresowych testów procedur odzyskiwania Systemu w tym testów scenariuszy “Disaster recovery”. Czynności te winny być prowadzone nie rzadziej niż raz na sześć miesięcy lub po każdej zmianie/modyfikacji konfiguracji polityk archiwizacji danych. Na wniosek oraz w porozumieniu z Wykonawcą Zamawiający wskaże termin przeprowadzenia w/w prac. Nie może być on jednak dłuższy niż 21 dni od zgłoszenia przez Wykonawcę gotowości do dokonania w/w czynności. Każda weryfikacja musi zostać potwierdzona obustronnie zawartym protokołem odbioru bez uwag. Zamawiający w terminie do 5 Dni Roboczych od otrzymania protokołu zaakceptuje go lub zgłosi Uwagi. W terminie do 14 dni kalendarzowych Wykonawca zobligowany jest do usunięcia przyczyn powstania uwag wskazanych w protokole odbioru. Po usunięciu przyczyn powstania uwag proces odbioru zostanie powtórzony. Zamawiający dopuszcza dwukrotne powtórzenie czynności odbiorowych. W przypadku uznania, że procedury odzyskiwania Systemu po awarii lub scenariusze “Disaster recovery” są niekompletne, Wykonawca zobowiązany jest do uzupełnienia wyżej wymienionych dokumentów w terminie 3 miesięcy do dnia podpisania Umowy, w ramach usługi ATiK.

### **Wymagania dotyczące Rozwoju.**

### **Wymagania Ogólne.**

W ramach Modyfikacji i Rozwoju Systemu Wykonawca zobowiązany jest do:

1. Opracowywania i wdrażania nowych funkcjonalności Systemu oraz dokonywania wszelkich innych zmian w Systemie w zakresie wskazanym przez Zamawiającego, w tym wynikających ze zmian przepisów prawa, zaleceń audytorów, kontrolerów, zmieniających się wymogów technologicznych oraz optymalizacji procesów biznesowych.
2. Dokonywania zmian w Systemie na potrzeby integracji z innymi systemami wykorzystywanymi przez Zamawiającego.
3. Utrzymania wartości parametrów związanych z Rozwojem na warunkach opisanych w Załączniku nr 5 „Poziom świadczenia usług SLA” do Opisu Przedmiotu Zamówienia.

### **Zasady realizacji Modyfikacji i Rozwoju.**

1. Wykonawca nie może odmówić realizacji złożonego Zamówienia, poza przypadkami, gdy jego realizacja spowoduje przekroczenie limitu Roboczogodzin lub terminu realizacji Umowy.
2. Zamawiający informuje, iż na trzy tygodnie przed upływem terminu realizacji Umowy, Zamawiający nie planuje zlecania zmian modyfikujących System.
3. Zamawiający wymaga, aby Wykonawca przy realizacji prac w ramach Modyfikacji i Rozwoju dysponował zespołem projektowo-programowym, który może wykonać prace o zakresie nie mniejszym niż 500 Roboczogodzin w trakcie jednego miesiąca.
4. W przypadku, gdy do realizacji prac w ramach Modyfikacji i Rozwoju niezbędne jest użycie licencji, Wykonawca zobowiązany jest do wykorzystania licencji typu open source. Stosowanie płatnych licencji dopuszczalne jest wyłącznie w sytuacji braku odpowiedniej licencji typu open source. W takim przypadku Wykonawca udzieli Zamawiającemu lub zagwarantuje udzielenie na rzecz Zamawiającego przez podmioty trzecie, przenoszalnych, bezterminowych i niewyłącznych licencji na korzystanie z takiego Oprogramowania, zgodnie z postanowieniami Paragrafu 8 Umowy po udzieleniu przez Zamawiającego zgody na zastosowanie takiej licencji lub po dostarczeniu jej przez Zamawiającego. Koszt pozyskania licencji spoczywa na Wykonawcy. Zgoda Zamawiającego wymagana jest również w przypadku konieczności zastosowania oprogramowania open-source.
5. Zrealizowane prace nie mogą prowadzić do naruszenia struktur i integralności danych, do utraty danych lub wpływać negatywnie na funkcjonowanie Systemu lub innych składników infrastruktury Zamawiającego. W przypadku, gdy wykonanie prac wiąże się z ryzykiem utraty danych, Wykonawca zobowiązany jest poinformować o tym Zamawiającego przed przystąpieniem do realizacji prac w ramach Modyfikacji i Rozwoju.
6. W przypadku, gdy realizacja prac spowoduje pojawienie się Wady w Systemie, Wykonawca zobowiązany jest do wstrzymania prac w ramach Modyfikacji i Rozwoju, do czasu skutecznego usunięcia Wady.
7. Wykonawca zobowiązany jest do zapewnienia zgodności Produktów przekazywanych w ramach realizacji Modyfikacji i Rozwoju z zaleceniami WCAG zawartymi w Załączniku nr 2 do Opisu Przedmiotu Zamówienia.
8. Wszystkie Zamówienia oraz inne materiały z realizacji Modyfikacji i Rozwoju (w tym z testów) muszą być przez Strony rejestrowane i prezentowane w Portalu Serwisowym.

### **Etapy realizacji Rozwoju Systemu**

Procedura realizacji Modyfikacji i Rozwoju Systemu składa się z etapów:

* Etap I – analiza i projekt,
* Etap II – realizacja.

#### **Etap I (analiza i projekt)** - inicjowany jest przez Zamawiającego

1. Zamawiający tworzy Zamówienie w Portalu Serwisowym zawierające: krótki opis (koncepcje) procesu biznesowego, w miarę możliwości: funkcjonalności oraz zakres danych oraz inne informacje mogące mieć wpływ na realizację Zamówienia.
2. Wykonawca zobowiązany jest, by w terminie 10 dni od dnia zgłoszenia Zamówienia dostarczyć nieodpłatnie wynik realizacji Analizy Wstępnej. Akceptacja Analizy Wstępnej przez Zamawiającego warunkuje dalszą realizację Modyfikacji i Rozwoju.
3. Wykonawca zgodnie z harmonogramem przedstawionym w Analizie Wstępnej przedstawi Zamawiającemu projekt zmian w warstwach logiki biznesowej, danych, uprawnień, architektury fizycznej i logicznej systemu, konfiguracji komponentów sprzętowych, zawierający w szczególności:
4. opis dziedziny Systemu oraz specyfikację wymagań w obszarze funkcjonalnym i (poza funkcjonalnym), które będą przedmiotem prac programistycznych;
5. opis architektury Systemu po zmianach (głównie perspektywa biznesu, perspektywa logiczna, oraz perspektywa danych);
6. projekty wszystkich modułów, które będą przedmiotem prac. Minimalny zakres informacji to:

* opis elementów struktury (tj. komponentów, podmodułów itp.),
  + - opis głównych scenariuszy działania a w tym opis algorytmów po zmianach,
    - opis i makiety interfejsów po zmianach,
    - opis logicznego modelu danych wykorzystywanych w modułach,
    - opis modelu wdrożenia;

1. wycenę realizacji Etapu II w Roboczogodzinach z rozbiciem na poszczególne zadania składowe (podzadania) w podziale uzgodnionym z Zamawiającym;
2. zakres niezbędnego współdziałania Zamawiającego;
3. harmonogram realizacji prac;
4. informację o wpływie realizacji prac w ramach Rozwoju na integralność, wydajność oraz bezpieczeństwo Systemu;
5. wykaz niezbędnych licencji do uruchomienia zmian.
6. Wycena, o której mowa w pkt 14 powyżej musi zawierać, odrębnie dla każdej pozycji szacunkową liczbę Roboczogodzin niezbędną do przeprowadzenia:
7. prac analitycznych,
8. prac programistycznych,
9. zmian w Kodzie Źródłowym,
10. testów,
11. warsztatów z nowych funkcjonalności dla Użytkowników i przedstawicieli Zamawiającego.
12. Strony mogą ustalić inny termin dostarczenia Produktów Etapu I przez Wykonawcę.
13. Strony mogą ustalić inny zakres dla dokumentacji, o której mowa w pkt MR -14 dostarczanej w ramach Produktów Etapu I.
14. Dopuszcza się by w ramach Zamówienia na Modyfikację i Rozwój wytwarzane były inne produkty będące częścią składową Systemu lub powiązane z Systemem, nie będące oprogramowaniem. W takim przypadku Strony mogą ustalić indywidualny tryb i zakres realizacji Usługi.
15. Jeśli w ramach realizacji Modyfikacji i Rozwoju istnieje techniczna możliwość zastąpienia komercyjnego rozwiązania autorstwa podmiotu trzeciego lub Wykonawcy przez oprogramowanie otwarto-źródłowe, Wykonawca jest zobowiązany uwzględnić ten fakt w przedstawionym projekcie, o którym mowa powyżej, stwarzając Zamawiającemu możliwość podjęcia decyzji w zakresie doboru konkretnego rozwiązania.
16. Zamawiającemu przysługuje prawo weryfikacji i akceptacji sposobu oraz czasochłonności wykonania przez Wykonawcę prac, który został przedstawiony przez Wykonawcę. Ostateczna akceptacja wyceny czasochłonności prac należy do Zamawiającego. W przypadku zaistnienia różnicy zdań między Stronami dotyczącej wiarygodności przedstawionego rozliczeń i szacunków czasochłonności, Strony zobowiązują się do podporządkowania się opinii niezależnego od Stron biegłego i rozliczenia prac według podanych przez niego wskazań. Biegły zostanie wybrany przez Strony metodą zapewniającą bezstronność, a także będzie osobą posiadającą potwierdzoną certyfikatami wiedzę z zakresu wymiarowania przedsięwzięć informatycznych. Biegły będzie wybierany z listy osób wpisanych na listę rzeczoznawców Polskiego Towarzystwa Informatycznego, przy czym pod uwagę będą brane jedynie osoby wpisane na listę nie później niż w dniu publikacji ogłoszenia. Koszt sporządzenia opinii ponosi strona przeciwna tej, do racji której biegły się przychyli. Jeśli biegły nie przychyli się do racji żadnej ze stron, obie strony ponoszą koszt sporządzenia opinii po połowie.
17. Zamawiający zobowiązany jest do przekazania Wykonawcy informacji czy akceptuje, czy odrzuca przedstawiony przez Wykonawcę wynik Etapu I.
18. Strony mogą dokonywać zmian i uzupełnień do materiału w trybie roboczym. Wycena wykonania Modyfikacji i Rozwoju uzgodniona na koniec Etapu I będzie stanowić podstawę wyliczenia wynagrodzenia za wykonanie danej usługi.
19. Jeżeli Strony dokonają stosownych ustaleń przed rozpoczęciem realizacji Zamówienia, wycena Zamówienia (zadań składowych) może być aktualizowana w porozumieniu z Zamawiającym w miarę ustalania szczegółów realizacyjnych, które nie były znane lub nie zostały doprecyzowane w chwili zlecania realizacji Zamówienia. Ostateczna akceptacja wyceny czasochłonności prac należy do Zamawiającego.
20. Zamawiający ma prawo zrezygnować z realizacji Etapu II.
21. Niezależnie od realizacji etapu analizy i projektu, Wykonawca będzie zobowiązany by w terminie 10 dni od dnia zgłoszenia Zamówienia dostarczyć wynik realizacji Analizy Wstępnej. Akceptacja Analizy Wstępnej przez Zamawiającego warunkuje dalszą realizację Modyfikacji i Rozwoju.
22. Po zakończeniu Etapu I Zamawiający może złożyć Zamówienie o realizację Etapu II.
23. Poza Analizą Wstępną, produkty Etapu I podlegają odbiorom na zasadach opisanych w Umowie.

#### **Etap II (realizacja)** - inicjowany przez Zamawiającego.

1. Wykonawca przystępuje do realizacji Etapu II po otrzymaniu od Zamawiającego Zlecenia Etapu II.
2. W wyjątkowych sytuacjach podczas trwania etapu II (realizacja), ale przed przekazaniem Produktów tego etapu do Odbioru, Zamawiający może zgłosić wniosek o zmianę wymagań, Wykonawca zobowiązany jest uwzględnić zgłoszone we wskazanym czasie zmiany wymagań. Wykonawca przedstawi Zamawiającemu poprawkę do projektu zawierającą zgłoszone zmiany wymagań, wycenę ich wykonania oraz wpływ na harmonogram realizacji w terminie 5 Dni Roboczych – Strony mogą ustalić inny termin.
3. Wykonawca przeprowadza testy wewnętrzne zgodnie z wymaganiami opisanymi w Załączniku nr 4 do Opisu Przedmiotu Zamówienia na Środowisku Testowym według przygotowanych przez siebie scenariuszy testowych i potwierdza Zamawiającemu ich wykonanie poprzez wprowadzenie stosownej informacji do Portalu Serwisowego.
4. Po przeprowadzeniu testów wewnętrznych Wykonawca zgłasza Zamawiającemu gotowość do testów akceptacyjnych.
5. Poinformowanie Zamawiającego o gotowości do zainstalowania przez Wykonawcę Pakietu Aktualizacji na Środowisku Testowym uznaje się za zgłoszenie przez Wykonawcę gotowości do Odbioru realizowanego Zamówienia.
6. Po zgłoszeniu gotowości Odbioru Zamawiający przystąpi niezwłocznie do weryfikacji Pakietu Aktualizacji.
7. Zamawiający ma prawo do weryfikacji należytego wykonania Zlecenia dowolną metodą. Zamawiający ma prawo przeprowadzić testy za pomocą samodzielnie zdefiniowanych scenariuszy testowych.
8. Wykonawca ma obowiązek dostarczyć Zamawiającemu dokumenty, w tym raporty, scenariusze testowe wymagane w Załączniku nr 3 do OPZ, najpóźniej w momencie zgłoszenia Zamawiającemu przez Wykonawcę gotowości do Odbioru (patrz pkt MR-32).
9. Zamawiający w terminie określonym w harmonogramie, począwszy od daty zgłoszenia przez Wykonawcę gotowości do Odbioru, przeprowadzi testy akceptacyjne, w trakcie których zweryfikuje przedmiot Odbioru i przekaże Wykonawcy za pośrednictwem Portalu Serwisowego informację o zidentyfikowanych Wadach przedmiotu Odbioru.
10. Wady będą rejestrowane w Portalu Serwisowym przez Zamawiającego oraz będą im nadawane odpowiednie kategorie:
    * + 1. Krytyczny
        2. Ważny
        3. Zwykły
        4. Kosmetyczny
11. W przypadku, o którym mowa MR-37 pkt 1 Wykonawca zobowiązany jest do usunięcia Wady w przedmiocie Odbioru w terminie 1 Dnia Roboczego od dnia jej zgłoszenia przez Zamawiającego lub w innym terminie wskazanym przez Zamawiającego.
12. W przypadku, o którym mowa MR-37 pkt 2 Wykonawca zobowiązany jest do usunięcia Wady w przedmiocie Odbioru w terminie 2 Dni Roboczych od dnia jej zgłoszenia przez Zamawiającego lub w innym terminie wskazanym przez Zamawiającego
13. W przypadku, o którym mowa MR-37 pkt 3 Wykonawca zobowiązany jest do usunięcia Wady w przedmiocie Odbioru w terminie 3 Dni Roboczych od dnia jej zgłoszenia przez Zamawiającego lub w innym terminie wskazanym przez Zamawiającego
14. W przypadku, o którym mowa MR-37 pkt 4 Wykonawca zobowiązany jest do usunięcia Wady w przedmiocie Odbioru w terminie 4 Dni Roboczych od dnia jej zgłoszenia przez Zamawiającego lub w innym terminie wskazanym przez Zamawiającego
15. Testy akceptacyjne po stronie Zamawiającego będą trwały do usunięcia przez Wykonawcę wszystkich zgłoszonych Wad.
16. Jeżeli Wykonawca nie wykona usługi w terminach określonych w projekcie, o którym mowa w MR-14 powyżej, Zamawiający może:
17. wydłużyć termin wykonania usługi na pisemną prośbę Wykonawcy zawierającą uzasadnienie i zmiany harmonogramu;
18. obciążyć Wykonawcę karą umowną na zasadach opisanych w Umowie.
19. Po zakończeniu testów akceptacyjnych, Wykonawca ma obowiązek instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym w terminie 3 Dni Roboczych, chyba, że Zamawiający postanowi inaczej.
20. Nie później niż na 3 Dni Robocze przed Instalacją Pakietu Aktualizacji na Środowisku Produkcyjnym, Wykonawca dostarcza zaktualizowaną zgodnie z wymogami opisanymi w Załączniku nr 3 do Opisu Przedmiotu Zamówienia, kompletną zaktualizowaną Dokumentację Systemu. Przekazana zaktualizowana Dokumentacja Systemu musi zawierać wszelkie informacje pozwalające Zamawiającemu lub podmiotom wybranym przez Zamawiającego na samodzielne korzystanie z Produktów, a także na ich samodzielne utrzymywanie i rozwój.
21. Wykonawca nie później niż na 2 Dni Robocze przed Instalacją Pakietu Aktualizacji na Środowisku Produkcyjnym zobowiązany jest każdorazowo przeprowadzić warsztaty szkoleniowe on-line z nowych funkcjonalności dla Użytkowników i przedstawicieli Zamawiającego, jeśli Zlecenie zawiera takie zapotrzebowanie.
22. Instalacja Pakietu Aktualizacji na Środowisku Produkcyjnym realizowana będzie w czasie Okna Serwisowego, o ile Strony nie uzgodnią inaczej.
23. Zamawiający zastrzega sobie prawo rezygnacji z instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym.
24. Warunkiem zakończenia realizacji Zlecenia jest:
25. Pozytywny Odbiór Zlecenia;
26. zainstalowanie przez Wykonawcę Pakietu Aktualizacji na Środowisku Produkcyjnym.
27. dostarczenie Zamawiającemu przez Wykonawcę zaktualizowanej Dokumentacji Systemu (patrz pkt MR-46);
28. przeprowadzenie przez Wykonawcę warsztatów, o których mowa w pkt MR-47 (o ile Zlecenie obejmuje warsztaty).
29. Zakończenie realizacji Zlecenia potwierdzane jest poprzez zamknięcie Zadania w Portalu Serwisowym przez upoważnionego pracownika Zamawiającego wskazanego w Paragrafie 5 Umowy.
30. Zamknięcie Zadania w Portalu Serwisowym oznacza możliwość ujęcia Zlecenia w Protokole Odbioru Rozwoju, którego wzór stanowi Załącznik nr 2 do Umowy.
31. Podpisanie Protokołu Odbioru, o którym mowa w pkt MR-43 powyżej, przez Zamawiającego bez zastrzeżeń jest podstawą do wystawienia przez Wykonawcę faktury.
32. Z chwilą zainstalowania przez Wykonawcę Pakietu Aktualizacji na Środowisku Produkcyjnym Wykonawca obejmuje go Usługą Asysty Technicznej i Konserwacji oraz gwarancją, o której mowa w Paragrafie 3 Umowy bez zmiany wynagrodzenia przysługującego z tytułu realizacji Umowy.
33. Protokół Odbioru Zamówienia wykonanego w ramach Rozwoju zawierać będzie informację o liczbie Roboczogodzin, w ramach których Zamówienie zostało wykonane. Liczba Roboczogodzin wskazana w zaakceptowanym przez Zamawiającego Protokole Odbioru będzie podstawą do rozliczenia limitu Roboczogodzin na Rozwój określonego w Załączniku nr 2 do Umowy.
34. Zamawiający zastrzega sobie możliwość realizacji Modyfikacji i Rozwoju w trybie alternatywnym/zwinnym. Zakres tych prac, tryb i forma realizacji określane są na etapie zlecenia usługi. Rozliczanie pracochłonności i określanie wynagrodzenia prac z puli Roboczogodzin przewidzianych na realizację Modyfikacji i Rozwoju ma miejsce z dołu, w oparciu o przekazywane przez Wykonawcę na bieżąco codzienne raporty i inne gromadzone na bieżąco w Repozytorium Projektu wiarygodne dowody potwierdzające fakt świadczenia usług przez poszczególnych pracowników Wykonawcy.

Lista załączników do Opisu Przedmiotu Zamówienia:

# Załącznik nr 1 do OPZ - **Wymagania wydajnościowe i niezawodnościowe**.

# Załącznik nr 2 do OPZ - **Wymagania w zakresie WCAG 2.1**.

# Załącznik nr 3 do OPZ - **Wymagania dla Dokumentacji.**

1. Załącznik nr 4 do OPZ - Wymagania dotyczące testów.

# Załącznik nr 5 do OPZ - **Poziom świadczenia usług SLA**.

**Załącznik nr 1 do OPZ** - **Wymagania wydajnościowe i niezawodnościowe.**

1. Wykonawca zapewni ciągłe funkcjonowanie Systemu przy założeniu, że System działa w trybie ciągłym 24 godziny na dobę 7 dni w tygodniu 365 dni w roku, z uwzględnieniem jednoczesnego korzystania z Aplikacji 6 000 użytkowników w ciągu jednego dnia, z ustalonym punktem przesilenia w godzinach 9:00 - 15:00 oraz założeniu utrzymania minimum 1 000 sesji jednocześnie.
2. Czas reakcji Systemu na zatwierdzenie formularza nie przekroczy 2 sekund. Podany czas nie dotyczy czasu wyszukiwania danych, wysyłania plików oraz generowania i dostępu do raportów oraz innych czynności związanych z wykonywaniem bardzo złożonych operacji na danych, które nie są wykonywane w trakcie codziennej, rutynowej pracy z Systemem.
3. Zamawiający jest uprawniony do prowadzenia testów sprawdzających dotrzymanie parametrów wydajnościowych Systemu. Ze strony Zamawiającego zostanie użyte narzędzie Apache JMeter (<http://jmeter.apache.org>).
4. Wykonawca będzie prowadził działania prewencyjne mające na celu wydłużenie czasu bezawaryjnej pracy Systemu, w tym będzie wykonywał optymalizacje Systemu oraz przeglądy nie rzadziej niż raz na kwartał, a także na żądanie Zamawiającego.
5. W przypadku konieczności wykonania prac mających na celu optymalizację działania Systemu Wykonawca bezzwłocznie poinformuje Zamawiającego o zakresie prac jaki jest z tym związany.
6. Wszelkie planowane przerwy w działaniu Systemu związane z wykonywaniem optymalizacji muszą być uzgodnione z Zamawiającym.

## **Załącznik nr 2 do OPZ** - **Wymagania w zakresie WCAG 2.1**

System powinien być całkowicie dostępny cyfrowo dla Użytkowników z wszelkimi niepełnosprawnościami, dla seniorów i wszystkich innych użytkowników Internetu. Ze względu na rolę, jaką pełni PFRON, Serwis/system powinien być wzorcowy w zakresie dostępności.

Wymóg dostępności serwisu/systemu PFRON wynika z:

* Ustawy z dnia 4 kwietnia 2019 r. o dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych (Dz.U. 2019 poz. 848),
* Rozporządzenia Rady Ministrów z dnia 12 kwietnia 2012 r. w sprawie Krajowych Ram Interoperacyjności, minimalnych wymagań dla rejestrów publicznych i wymiany informacji w postaci elektronicznej oraz minimalnych wymagań dla systemów teleinformatycznych (Dz.U. 2017 poz. 2247).

Zgodnie z ustawą serwisy internetowe realizujących zadania publiczne muszą być zgodne z WCAG 2.1 na poziomie A i AA.

Zatem Wykonawco, jesteś zobowiązany do dostarczenia serwisu/systemu, który jest bezbłędny pod względem jakości kodu, zgodności z WCAG 2.1 i rzeczywistej dostępności dla wszelkich grup narażonych na wykluczenie cyfrowe.

## **1. Ogólne wymagania w zakresie dostępności cyfrowej.**

Twoim obowiązkiem jest zapewnić dostępność cyfrową serwisu/systemu na poziomie WCAG 2.1 A oraz AA, zgodnie z załącznikiem nr 1 do Ustawy z dnia 4 kwietnia 2019 r. o dostępności cyfrowej stron internetowych i aplikacji mobilnych podmiotów publicznych (Dz.U. 2019 poz. 848).

Materiałami referencyjnym odnośnie spełnienia wytycznych WCAG 2.1 są:

1. [WCAG 2.1 (oficjalne tłumaczenie na język polski)](https://www.w3.org/Translations/WCAG21-pl/)
2. [Techniques for WCAG 2.1](https://www.w3.org/WAI/WCAG21/Techniques/#techniques) — jest to obszerny dokument on-line, który zawiera setki przydatnych fragmentów kodu i przykładów zastosowania kryteriów WCAG 2.1.  
   W trakcie projektowania elementów interfejsów (np. menu, nawigacja, okna modalne, formularze, nawigacja okruszkowa, tabele, karuzele itp.) powinieneś korzystać z wzorców projektowych i dobrych praktyk, opublikowanych na stronach:
   * 1. <https://www.w3.org/TR/wai-aria-practices/>
     2. [https://www.w3.org/WAI/tutorials/](https://www.w3.org/WAI/tutorials/menus/)

Wątpliwości dotyczące sposobów wdrażania dostępności cyfrowej będą rozstrzygane przez Zamawiającego na podstawie dokumentacji opracowanej przez [www.w3.org.](http://www.w3.org./)

## **2**. **Narzędzia wspierające budowę i testowanie dostępnych cyfrowo serwisów/systemów internetowych.**

Niżej wymienione narzędzia wspierają tworzenie dostępnych cyfrowo serwisów/systemów oraz umożliwiają wczesne wykrycie części problemów z obszaru dostępności cyfrowej. Pamiętaj jednak, że narzędzia automatyczne nie wykrywają wszystkich niezgodności z WCAG 2.1 – dlatego konieczna jest weryfikacja audytora WCAG.

Propozycja listy narzędzi:

* **NVDA** – czytnik ekranu <https://nvda.pl/>,
* **VoiceOver** - wbudowany w system Mac OS X mechanizm odczytywania komunikatów z ekranu - <https://www.apple.com/pl/voiceover/info/guide/_1121.html>,
* **WAVE** – narzędzie do wstępnej wizualnej ewaluacji zgodności strony z WCAG 2.1 <https://wave.webaim.org/>,
* **AXE Devtools** - narzędzie wspomagające badanie dostępności, generujące wstępną listę potencjalnych błędów <https://chrome.google.com/webstore/detail/axe-devtools-web-accessib/lhdoppojpmngadmnindnejefpokejbdd>,
* **ARC Toolkit** – rozszerzenie do przeglądarki Chrome, wspierające badanie kodu strony -[https://chrome.google.com/webstore/detail/arc‑toolkit/chdkkkccnlfncngelccgbgfmjebmkmce](https://chrome.google.com/webstore/detail/arc%1etoolkit/chdkkkccnlfncngelccgbgfmjebmkmce),
* **ANDI** – bookmarklet dla przeglądarki Chrome <https://www.ssa.gov/accessibility/andi/help/install.html>,
* **Colour Contrast Analyser** – narzędzie do weryfikowania kontrastu elementów strony <https://www.tpgi.com/color-contrast-checker/>,
* **HeadingsMap** – rozszerzenie pomagające określić strukturę oraz hierarchię nagłówków występujących na stronie <https://chrome.google.com/webstore/detail/headingsmap/flbjommegcjonpdmenkdiocclhjacmbi>,
* **Landmarks** – rozszerzenie pomagające określić punkty orientacyjne (tak zwane landmarki) występujące na stronie [https://chrome.google.com/webstore/detail/landmark‑navigation‑via‑k/ddpokpbjopmeeiiolheejjpkonlkklgp](https://chrome.google.com/webstore/detail/landmark%1enavigation%1evia%1ek/ddpokpbjopmeeiiolheejjpkonlkklgp)
* **Text Spacing** – narzędzie wspomagające symulację strony ze zwiększonymi odstępami w zakresie podanym w WCAG 2.1. <https://dylanb.github.io/bookmarklets.html>

## **3. Najważniejsze wymagania techniczne w zakresie dostępności (programistyczne).**

Poniższa część dokumentacji ma za zadanie zwrócić Twoją uwagę na kluczowe aspekty zapewnienia dostępności cyfrowej serwisu/systemu internetowego. Musisz wiedzieć, że są to tylko wytyczne wspierające Ciebie w realizacji kluczowych wytycznych WCAG 2.1, nie zaś pełna lista sposobów zapewnienia zgodności strony ze standardem WCAG 2.1.

### 3.1. Zgodność składni z walidatorem HTML.

Wszystkie strony serwisu/systemu muszą być bezbłędne pod względem jakości kodu HTML z walidatorem <https://validator.w3.org/nu/>.

W trakcie wdrożenia mogą się pojawić sytuacje, w których możemy zaakceptować błędy HTML. Muszą to być jednak uzasadnione i udokumentowane przypadki, związane z niestabilnością specyfikacji HTML5, które będą działać np. na rzecz dostępności.

**Uwaga:** Poza zgodnością z walidatorem samych szablonów serwisu/systemu, także treści zapisane przy użyciu edytora wizualnego WYSIWYG nie mogą powodować problemów. Dlatego edytor wizualny powinien generować prawidłowy kod HTML.

### 3.2. Jakość semantyczna kodu HTML.

Podstawowym warunkiem dostępności jest prawidłowe — adekwatne stosowanie znaczników HTML. Najprościej rzecz ujmując, serwis/system powinieneś realizować w pełnej zgodności ze [specyfikacją HTML5](https://www.w3.org/TR/html51/).

Przykłady poprawności semantycznej:

W ramach prac nad serwisem/systemem pamiętaj, że poszczególne elementy należy wykonać w określony sposób:

1. Linki za pomocą znacznika **<a>**, czyli natywnego semantycznego znacznika HTML. Jeśli jest to niemożliwe dopuszczalne są również niesemantyczne elementy **<div>** wraz z odpowiednią rolą **role=”link”**;
2. Nagłówki za pomocą znaczników **<h1>**..**.<h6>** (przy czym nagłówek **<h1>** winien występować tylko raz), czyli natywnego semantycznego znacznika HTML. Jeśli jest to niemożliwe dopuszczalne są również niesemantyczne elementy **<div>** wraz z odpowiednią rolą, na przykład dla nagłówka poziomu 1 **role=”heading” ARIA-level="1"**;
3. Przyciski za pomocą znaczników **<button>** lub **<input type="button">**, czyli natywnego semantycznego znacznika HTML. Jeśli jest to niemożliwe dopuszczalne są również niesemantyczne elementy **<div>** wraz z odpowiednią rolą **role=”button”**;
4. Listy za pomocą znaczników **<ul>**/**<ol>** i **<li>** dla poszczególnych elementów;
5. Rozwijane listy formularzy za pomocą znaczników **<select>**/**<option>**.

Przykłady błędów **semantycznych**:

Unikaj poniższych rozwiązań.

1. Link wykonany za pomocą **<span>** (oskryptowany JavaScript);
2. Nagłówek w formie **<p class="heading">**;
3. Lista rozwijana w formularzu, wykonana za pomocą znaczników listy **<ul>**/**<li>**.

### 3.3. Uzupełnienia semantyczne za pomocą ARIA.

Atrybuty ARIA muszą być uzupełnieniem semantyki HTML. To technologia przeznaczona przede wszystkim dla użytkowników czytników ekranu. Szczególnie ważne jest jej stosowanie w komponentach stron internetowych, które opierają się na rozbudowanej interakcji JavaScript.

Stosowanie atrybutów ARIA można podzielić na dwie części:

1. Uzupełnienie głównych bloków serwisu/systemu o punkty orientacyjne;
2. Dodatki do formularzy lub takich komponentów stron, jak karuzele, zakładki (**tabs**), menu rozwijane, bloki rozwijane, okna modalne, alerty, slidery.

Głównym źródłem informacji jak stosować ARIA powinna być dla Ciebie dokumentacja [Aria Techniques for WCAG 2.1](https://www.w3.org/WAI/WCAG21/Techniques/).

Niestety, nie jest to wystarczające źródło wiedzy. Nie ma jednego miejsca w Internecie zawierającego aktualną, pewną i gotową do stosowania wiedzę w zakresie ARIA.

Zastrzegamy sobie prawo do weryfikacji serwisu/systemu, w każdy dostępny sposób, pod względem zgodności ze specyfikacją ARIA w całym okresie obowiązywania Umowy. W przypadku stwierdzenia niezgodności ze specyfikacją ARIA będziesz zobowiązany do ich usunięcia na własny koszt w terminie wskazanym przez Zamawiającego.

### 3.4. Tytuły stron serwisu/systemu internetowego.

Wszystkie tytuły stron serwisu/systemu muszą być automatycznie generowane na podstawie informacji, które pozwolą użytkownikowi dowiedzieć się, co jest treścią danej strony.

Przykłady:

1. Strona główna serwisu/systemu powinna mieć tytuł — „Serwis informacyjny iPFRON+”.
2. Strona „Program Wsparcie Inicjatyw” powinna mieć tytuł — „Program Wsparcie Inicjatyw – Serwis informacyjny iPFRON+”.

Wszystkie strony mają mieć tytuł wg zasady - od szczegółu do ogółu.

Do uzgodnienia z Tobą pozostanie kwestia, ile elementów ścieżki ma być widocznych w tytule:

1. tytuł strony + nazwa serwisu lub
2. tytuł stron + nazwa działu + np. nazwa nadrzędnego działu + nazwa serwisu.

Zgodnie z Wymaganiami dla Systemu Zarządzania treścią (CMS), opis dodatkowych modułów i funkcjonalności CMS oraz CMS serwisu - Redaktor musi mieć możliwość indywidualnego definiowania zawartości atrybutu metatagu **title**, niezależnie od tytułu redakcyjnego.

### 3.5. Oznaczenie języka strony i treści.

Język naturalny treści na stronie powinieneś zawsze oznaczać odpowiednim atrybutem lang. W założeniu wszystkie strony serwisu/systemu będą miały atrybut **lang** o treści **"pl"** lub **“pl-PL”**.

Dodatkowo powinieneś zapewnić redaktorom serwisu w edytorze **WYSIWYG** możliwość oznaczenia takim atrybutem dowolnego ciągu znaków, tak by użytkownik korzystający z technologii asystujących mógł zorientować się, że treść jest w innym języku, niż domyślny język strony.

### 3.6. Nagłówki stałe.

W serwisie będą stałe bloki treści i bloki funkcjonalne. Powinieneś je oznaczyć nagłówkami na odpowiednim poziomie.

### 3.7. Nagłówki dla redaktorów.

Redaktorom powinieneś zapewnić możliwość ustawiania odpowiedniej struktury nagłówkowej stron. Nagłówki dostępne dla redaktora powinny się zawierać od **h2** do **h6**. Nagłówek **h1** powinien najtrafniej opisywać główną treść strony.

### 3.8. Linki.

W serwisie wszystkie linki powinny być zrozumiałe poza kontekstem tekstowym bądź wizualnym. W stałych częściach serwisu/systemu może oznaczać to potrzebę uzupełniania krótkich linków o treści uzupełniające. Linki powinny być uzupełniane przez treści niewidoczne dla użytkowników niekorzystających z czytników ekranu, na przykład za pomocą klasy **sr-only** czy **visually-hidden**.

Przykłady linków, które będzie można uzupełnić o dodatkową treść, to: zamknij, przewiń, następny, poprzedni, więcej, pobierz, pokaż wszystkie, itp.

### 3.9. Opisy alternatywne.

Wszystkie grafiki, które zamieścisz w szablonach za pomocą znacznika **<img>** powinny mieć atrybut **alt**.

1. W przypadku, gdy grafika nie będzie przekazywać żadnej treści (grafiki dekoracyjne), powinieneś je umieszczać za pomocą CSS, czyli stosując właściwość **background-image**. Inną metodą jest dodanie do **<img>** - pustego **alt**— zapis **alt** lub **alt=""**.
2. Jeśli grafika będzie przekazywać treść, atrybut **alt** powinieneś uzupełnić o adekwatny opis.
3. Jeśli grafika będzie linkiem, to w opisie alternatywnym powinieneś przekazywać funkcję linku, tak jakby to był link tekstowy lub zastosować opis **ARIA-label** lub ukrytą klasę   
   np. **<sr-only>** do opisu celu linku. Jeśli zastosujesz drugie rozwiązanie atrybut **alt** powinien być pusty.
4. Elementy, które zaimplementujesz za pomocą SVG powinny posiadać znacznik **<title>** **</title>**, w którym należy umieścić tekst alternatywny lub też dodać atrybut **ARIA-hidden=”true”**, jeśli ma to być grafika dekoracyjna.

### 3.10. Formularze — semantyka.

Budowa formularzy pod względem dostępności musi opierać się o dobre praktyki HTML5. Należy uwzględnić, że formularze mogą być używane przez osoby z niepełnosprawnością wzroku, niepełnosprawne ruchowo czy głucho-niewidome.

Powinieneś wiedzieć, jakie są popularne sposoby użycia formularzy, np. bez użycia myszki czy bez patrzenia na ekran.

W większości przypadków jako podstawy semantyki HTML dla formularzy rozumiemy:

1. użycie etykiet do wszystkich pól, etykiety mogą być ukryte lub widoczne,
2. zrozumiałość etykiet,
3. dostęp do wszelkich wskazówek bez konieczności patrzenia na ekran, np. za pośrednictwem czytnika ekranu (wskazówki, sugestie poprawy błędów, komunikaty błędów do obiektów formularzy powinny być powiązane semantycznie z tym obiektem, np. poprzez **ARIA-describedby**),
4. kolejność treści i pól formularzy wspierająca użyteczność i zrozumiałość,
5. zdefiniowanie atrybutu **“autocomplete”**,
6. zdefiniowanie wymagalności pól (**ARIA-required=”true/false” or required**)

### 3.11. Formularze — wsparcie użytkownika i informacja o błędach.

Większym wyzwaniem w przypadku formularzy jest właściwa dostępność informacji o tym, w jaki sposób wypełnić pola oraz informacje o błędach.

W tym przypadku kieruj się następującym podejściem:

1. wszystko, co możliwe, wykonaj za pomocą podstawowych elementów HTML + JavaScript — im dalej będzie sięgać wsteczna kompatybilność, tym lepiej,
2. jeśli formularz będzie tego wymagał, zastosuj atrybuty ARIA.

Kolejność w powyższym wypunktowaniu jest ważna - Użytkownicy mogą korzystać z przestarzałego oprogramowania. Dlatego zagwarantuj wsteczną kompatybilność w jak największym stopniu.

Nie rekomendujemy stosowania walidacji HTML. Prezentacja informacji w tym rozwiązaniu jest ograniczona czasem.

Przykłady poprawnych rozwiązań:

* Przykład z użyciem role="alert"  
   <https://www.upyoura11y.com/handling-form-errors/>
* Przykład wykorzystania aria-describedby:  
   <https://www.w3.org/WAI/WCAG21/Techniques/aria/ARIA1>

### 3.12. Tabele.

W przypadku tabel, kluczowe jest stosowanie odpowiedniej składni i semantyki HTML. Czytniki ekranu wspierają obsługę tabel bardzo dobrze.

Wskazówki, które pomogą Ci w tworzeniu dostępnych tabel znajdziesz na stronie <https://www.w3.org/WAI/tutorials/tables/>.

### 3.13. Działanie serwisu/systemu za pomocą klawiatury.

Prawidłowe zastosowanie semantyki HTML powinno gwarantować dostępność za pomocą klawiatury każdego aktywnego elementu na stronie – zadbaj o to na etapie wdrożenia, aby zapewnić bezbłędne działanie tej funkcjonalności.

Programiści muszą stosować zarządzanie fokusem przez JavaScript w taki sposób, aby nie stworzyć tzw. pułapki klawiaturowej. Taki błąd powoduje barierę dla użytkowników z niepełnosprawnością ruchu oraz korzystających z czytników ekranu.

### 3.14. Kolejność fokusu.

Fokus klawiatury powinien mieć kolejność wedle reguły od lewej do prawej i od góry do dołu. Na przykład, po przejściu fokusem menu głównego, powinien on trafić do głównego bloku treści lub lewej kolumny.

### 3.15. Ukrywanie treści.

W niektórych przypadkach, np. w linkach, może być konieczne stosowanie ukrytej treści. Takie rozwiązanie wspiera korzystanie z serwisu/systemu przez użytkowników z niepełnosprawnością wzroku.

Polecamy artykuły opisujące techniki ukrywania treści:

* <http://webaim.org/techniques/css/invisiblecontent>.
* <https://getbootstrap.com/docs/5.0/helpers/visually-hidden/>

Poza tymi obszarami, w których Wykonawca zaproponuje użycie techniki ukrywania, w ramach monitoringu wdrożenia, ekspert ds. dostępności pracujący w ramach zespołu projektowego będzie rekomendować miejsca, w których warto dodatkowo zastosować tę technikę.

### 3.16. Zabezpieczenie formularzy.

Zabezpiecz formularz w taki sposób, aby nie stwarzał barier dla użytkownika serwisu/systemu.

W takim przypadku musisz uważać z rozwiązaniami typu **CAPTCHA**. Tego typu zabezpieczenia najczęściej nie są w stanie zapewnić dostępności dla wszystkich odbiorców.

**W miarę możliwości filtrowanie spamu i działań niepożądanych pozostaw po stronie serwera lub wykonaj zabezpieczenia tak aby nie wymagały dodatkowego działania po stronie użytkownika.** Jeśli zaproponujesz rozwiązanie typu **CAPTCHA**, to będzie ono dokładnie testowane pod kątem dostępności dla wszystkich użytkowników.

Więcej informacji na temat rozwiązania **CAPTCHA** znajdziesz pod adresem:

<https://developers.google.com/recaptcha/docs/invisible>

### 3.17. Działanie filtrów / przeładowanie.

Wszelkie działania związane z przeładowaniem widoku takie jak:

1. filtrowanie,
2. sortowanie,
3. wyszukiwanie,

przetestuj z czytnikami ekranu. W takich sytuacjach kluczowy będzie komfort obsługi bezwzrokowej. Użytkownik powinien mieć pełną wiedzę na temat działania interfejsu i świadomość tego, że treść strony została zaktualizowana.

Przyjmij ogólną zasadę, że zmiany treści strony bez przeładowania stosujemy tylko w uzasadnionych sytuacjach.

W niektórych przypadkach, po zmianie przefiltrowania może być konieczna automatyczna zmiana tytułu strony **<title>**.

### 3.18. Działanie w trybie wysokiego kontrastu (WINDOWS).

Serwis/system powinien bezproblemowo działać w trybie wysokiego kontrastu Windows. Wykonawca powinien prowadzić takie testy na bieżąco w trakcie wdrożenia.

Typowe problemy w takim trybie mogą być związane z użyciem CSS-owego zastępowania tekstu grafiką. Dlatego w niektórych przypadkach zamiast użycia takiej techniki, będzie konieczne zastosowanie typowych linków graficznych **<a><img></a>**.

### 3.19. Skip linki.

Na każdej stronie serwisu powinien działać link „**Przejdź do wyszukiwania**”, „**Przejdź do głównej treści**” (jeżeli takie elementy występują), które pomagają przeskoczyć fokusem bezpośrednio do głównej funkcjonalności danej strony. Najczęściej będzie to oznaczać przeskoczenie nawigacji lub też innych powtarzających się elementów na stronie. Takie elementy zaprojektuj i skonsultuj z zespołem zleceniodawcy.

## **4. Inne wymagania techniczne.**

### 4.1. Szybkość działania serwisu/systemu.

Serwis/system powinien być maksymalnie zoptymalizowany do szybkiego działania. Lekkość serwisu wpływa pozytywnie na działanie z oprogramowaniem wspomagającym, takim jak np. czytnik ekranu. Takie działanie powoduje również komfortową obsługę w urządzeniach mobilnych. W ramach optymalizacji pod kątem szybkości działania trzeba będzie zwrócić uwagę na następujące kwestie:

1. brak nadmiarowego kodu HTML / CSS / JS,
2. nieobciążanie serwisu/systemu zbędnymi dodatkami JS,
3. dobrą optymalizację grafiki,
4. minimalizację liczby plików pobieranych wraz z unikalną stroną,
5. cache serwisu, który zminimalizuje zapytania do bazy danych.

### 4.2. Responsywność (RWD).

Przy budowaniu serwisu/systemu pamiętaj o urządzeniach mobilnych, które pełnią ważną rolę w odbiorze treści internetowych.

Serwis/system buduj w oparciu o najlepsze i aktualne praktyki tworzenia serwisów responsywnych.

Przygotuj wszystkie projekty graficzne z zastosowaniem skoków responsywnych szerokości w odniesieniu do typów urządzeń (standardów):

1. smartfon – z rozdzielczością 360x640 (wartość średnia) w wersji pionowej oraz poziomej (z uwzględnieniem wartości minimalnej 320 px),
2. tablet - z rozdzielczością 768x1024 w wersji pionowej oraz poziomej,
3. monitor komputerowy - z rozdzielczością 1366x768 (wartość średnia), z uwzględnieniem wartości minimalnej – 900 px oraz wartości maksymalnej - 1920 px.

Zwróć uwagę, aby obiekty nie zachodziły na siebie i nie przykrywały treści bądź funkcjonalności.

Przy projektowaniu widoków mobilnych uwzględnij minimalną wielkość fontów – 16 px. Jest to wartość ważna podczas analizy czytelności strony.

### 4.3. Możliwości edytora WYSIWYG.

W edytorze wizualnym poza standardowymi funkcjami, udostępnisz redaktorom kilka dodatkowych narzędzi. Dokonując wyboru **WYSIWYG**a trzeba będzie sprawdzić, czy rozwiązanie obsługuje dane funkcje natywnie, np. na podstawie pluginów.

Wstępna rekomendacja na rozwiązanie WYSIWYG to [TinyMCE](https://www.tinymce.com/). W trakcie produkcji systemu trzeba będzie dokonać wyboru, która wersja edytora powinna być dostępna dla redaktorów strony internetowej lub aplikacji internetowej.

## **5. Szczegółowe wytyczne w zakresie dostępności (graficzne).**

### 5.1. Kontrast treści.

Kontrast między kolorem tekstu a kolorem jego tła, musi wynosić minimum 4,5:1 lub 3:1 dla większego tekstu (krój pisma powyżej 18 punktów).

Prostym narzędziem do analizy poziomu kontrastu jest [Colour Contrast Analyzer](https://www.paciellogroup.com/resources/contrastanalyser/).

W związku z wymogami dotyczącymi kontrastu, nie powinieneś stosować elementów prezentujących tekst na tle niejednorodnym, np. bezpośrednio na tle zdjęcia. Istnieje możliwość dodania takiego tekstu wraz z zastosowaniem atrybutu **CSS opacity** o wartości mniejszej niż 1.

Możesz stosować kolorystykę o mniejszym kontraście, ale tylko w zakresie elementów dekoracyjnych w serwisie. Kryterium kontrastu nie obejmuje logo serwisu.

### 5.2. Identyfikacja linków.

Linki tekstowe muszą być łatwe do odnalezienia przez wszystkich użytkowników serwisu.

Muszą odróżniać się od tekstu zarówno kolorem jak i podkreśleniem. Niedopuszczalne jest zastosowanie tylko koloru do wyróżnienia linku.

Podkreślenia użyj w projekcie graficznym wyłącznie do oznaczenia linków. To samo dotyczy koloru linków. Nie może być on powtórzony na żadnym elemencie nieklikalnym i musi spełniać wymogi wskazane w punkcie “Kontrast treści”.

Po oznaczeniu linku kursorem myszy (**hover**) podkreślenie linku powinno znikać, a kolor linku zmieniać się na kolor o wyższym wskaźniku kontrastu do tła, niż przy kolorze bazowym linku.

### 5.3. Formularze.

Wymóg widoczności dotyczy również formularzy stosowanych w serwisie/systemie. W szczególności odnosi się to do widoczności ramek pól, etykiet pól oraz przycisków.

Wszystkie elementy formularzy muszą spełniać wymóg kontrastu w stosunku do tła na poziomie przynajmniej 3:1.

Tak jak w przypadku linków, przyciski formularzy po oznaczeniu kursorem myszy bądź fokusem klawiatury muszą stawać się widoczne dla użytkowników (zwiększenie kontrastu między kolorem przycisku a kolorem tekstu przycisku).

Etykiety pól powinny być widoczne (w niektórych przypadkach mogą być ukryte, jednakże muszą być możliwe do przetworzenia przez narzędzia asystujące - na przykład **<label>** do elementu **<input>** wyszukiwarki) i prezentowane bezpośrednio obok pola. Etykiety powinny być programistycznie powiązane z polami formularzy za pomocą atrybutów **“for”** i **“id”**.

Dodatkowe informacje, które ułatwią użytkownikowi wypełnić formularz powinny być powiązane z elementem **<input>** za pomocą atrybutu **ARIA-labelledby**.

Informacje o błędach powinny być prezentowane tekstowo, bezpośrednio obok pól których dotyczą (dodatkowo powiązane z polem poprzez **ARIA-describedby**) oraz pod nagłówkiem rozpoczynającym blok z formularzem. Powinien istnieć jeden, ogólny komunikat informujący użytkownika o błędnym wypełnieniu formularza wraz z rolą alert <https://www.w3.org/TR/WCAG20-TECHS/ARIA19.htm>

### 5.4. Fokus klawiatury.

Cały serwis będzie umożliwiał nawigację za pomocą samej klawiatury.

Fokus klawiatury powinien mieć formę wzmocnioną w stosunku do fokusu domyślnego przeglądarki i być widoczny przy nawigacji za pomocą klawiatury w formie ramki, wokół wybranego elementu.

Kolor ramki fokusu dobierz do schematu kolorystycznego serwisu tak aby był dobrze widoczny na oznaczonym elemencie (minimalny kontrast – 3:1).

Przykład dobrze widocznego fokusu możesz zobaczyć w serwisie [www.pfron.org.pl](http://www.pfron.org.pl/) - wystarczy zacząć nawigację w serwisie/systemie za pomocą przycisku TAB.

Do rozróżnienia fokusa klawiatury i myszki możesz wykorzystać bibliotekę dostępną na stronie: <https://github.com/ten1seven/what-input>

### 5.5. Typografia.

Czcionki, których użyjesz w serwisie/systemie powinny być bezszeryfowe, o wysokim poziomie czytelności - także przy dużym powiększeniu. Przykładami tego typu czcionek są: Lato, Open Sans czy PT Sans.

Liczbę czcionek (kroju i wielkości) powinieneś ograniczyć w projekcie graficznym serwisu do niezbędnego minimum.

### 5.6. Spójna identyfikacja.

W ramach serwisu/systemu zaplanuj i zaprezentuj widoki tekstowych elementów semantycznych, takich jak:

1. nagłówek poziomu 1, każda strona powinna posiadać jeden nagłówek poziomu 1, pozostałe w odpowiedniej hierarchii, jeżeli treść jest wymagana.
2. lista numerowana (uporządkowana),
3. lista wypunktowana (nieuporządkowana),
4. listy obu typów wielokrotnie zagnieżdżone,
5. link,
6. tekst podstawowy,
7. tekst podstawowy wyróżniony,
8. przycisk (3 schematy dla różnych funkcjonalności),
9. listy rozwijane (**select**),
10. przyciski typu radio,
11. pola wyboru,
12. pole edycyjne.

Wielkość krojów pisma, których użyjesz w poszczególnych stylach powinna odpowiadać hierarchii tych stylów względem siebie. Dobrym rozwiązaniem jest, abyś przyjął zasadę, iż nagłówek poziomu 6 powinien być co najmniej wielkości kroju pisma podstawowego, tylko pogrubionego.

Minimalna wielkość kroju pisma, którą dopuszczamy w projekcie graficznym to 12 px., przy czym treść podstawowa powinna mieć wielkość minimum 16 px.

Dla treści nie powinieneś stosować formatowania wersalikami.

Odstępy między wierszami w akapitach powinieneś ustawić na co najmniej 1,3-1,5 wysokości linii, a odległość między akapitami powinna być przynajmniej 1,5 razy większa niż ta pomiędzy wierszami. W innym przypadku powinieneś zapewnić możliwość zmiany wielkości, bez utraty treści (np. za pomocą **1.4.12 Text Spacing** – narzędzie wspomagające symulację strony ze zwiększonymi odstępami w zakresie podanym w WCAG 2.1.).

W jednym wersie powinieneś zaprezentować do 85 znaków.

Nie justuj (równoczesne wyrównanie do lewej i prawej) żadnej treści w projekcie graficznym. Dopuszczamy tylko wyrównanie do lewej, a w uzasadnionych sytuacjach wyśrodkowanie tekstu.

Tam, gdzie to możliwe, treść prezentuj w formie tekstu, a nie grafiki tekstu. Do osiągnięcia pożądanego wyglądu użyj odpowiednich stylów CSS.

Spójna identyfikacja to nie tylko spójność użycia krojów pisma lub stylów. Rozumiemy to jako jednolitą implementację tych samych elementów na różnych podstronach. Przykładem jest ten sam opis logo serwisu/systemu we wszystkich miejscach, w których występuje bądź też elementu ukazującego podpowiedź przy wypełnianiu formularza (nie może raz być to “otwórz podpowiedź”, a za innym razem “pomoc”).

### 5.7. Tabele.

Pamiętaj, że tabele z danymi prezentowane w projekcie graficznym powinny posiadać wyraźnie odróżniające się od reszty komórek wersy/kolumny nagłówkowe. Prawidłowa implementacja jest kluczowa dla zrozumienia tabeli przez narzędzia asystujące.

Musisz zwrócić szczególną uwagę na informowanie technologii asystującej na temat stanu sortowania/filtrowania oraz ilości danych w tabeli.

### 5.8. Możliwość swobodnej zmiany wielkości widoku.

Pamiętaj, że koncepcja serwisu zakłada możliwość swobodnej zmiany wielkości strony (Ctrl + oraz Ctrl - ). Przy każdej szerokości ekranu/poziomie powiększenia (nie tylko przeznaczonej dla tabletów i smartfonów) wszystkie treści i funkcje serwisu powinny być czytelne. Projekt graficzny musi umożliwiać zaprogramowanie w ten sposób serwisu.

### 5.9. Elementy ruchome.

### Dopuszczamy elementy ruchome w serwisie, ale tylko w połączeniu z przyciskiem, który umożliwi użytkownikowi ich zatrzymanie i ponowne uruchomienie.

Żaden element serwisu/systemu nie może migać, jeśli czynność ta powtarza się więcej niż 3 razy na sekundę.

### 5.10. Elementy rozwijane.

Wszystkim elementom, które są rozwijane powinieneś przypisać atrybut **ARIA-expanded**. Jego wartość należy ustawić z poziomu JS (**true** albo **false**) - w zależności czy element jest zwinięty czy rozwinięty: **ARIA-expanded="true"** jeśli jest rozwinięty, **ARIA‑expanded="false”** jeśli jest zwinięty. Dzięki temu użytkownicy korzystający z aplikacji asystujących będą wiedzieli jaka jest aktualna struktura zamieszczonych informacji.

### 5.11. Multimedia.

Naszą rekomendacją odnośnie materiałów wideo jest ich prezentacja za pomocą standardowego odtwarzacza YouTube. Treści wideo powinny posiadać napisy i audiodeskrypcję. Projekt graficzny powinien uwzględniać zamieszczanie bezpośrednio pod materiałem wideo linku do transkrypcji tekstowej materiału, jeśli nie jest umieszczona bezpośrednio w filmie.

### 5.12. Elementy zmienne.

Wszelkie elementy, które zmieniają swoją wartość, dzięki działaniu jakiegoś mechanizmu (na przykład kalkulatora czy formularza), powinny mieć atrybut **ARIA-live**. Dzięki niemu użytkownik jest informowany o zmianie treści na stronie. Przykłady działania atrybutu znajdziesz na stronie <https://dequeuniversity.com/library/aria/liveregion-playground>

### 5.13. Kolorystyka serwisu/systemu.

Jedyne ograniczenia kolorystyczne w serwisie/systemie dotyczą logotypu Państwowego Funduszu Rehabilitacji Osób Niepełnosprawnych (patrz załącznik do wytycznych) oraz minimalnego kontrastu treści do tła oraz wymagania w 2.7. Spójna identyfikacja.

## **6. Zalecenia na poziomie AAA.**

Interfejs graficzny serwisu/systemu będzie zgodny z wytycznymi WCAG 2.1 poziomu A oraz AA. Dla wskazanych poniżej elementów interfejsu spełnione zostaną zalecenia na poziomie AAA:

1. 1.4.8 Prezentacja wizualna:
   1. szerokość nie przekracza 80 znaków, tekst nie jest wyjustowany,
   2. interlinia to przynajmniej 150%, a odstęp pomiędzy paragrafami 1.5 razy wartości interlinii,
   3. tekst powiększony do 200% nie wymaga przesuwania horyzontalnego.
2. 2.4.9 Cel łącza (z samego łącza): wymaganie opisaliśmy w punkcie [3.8 Linki](https://euc-word-edit.officeapps.live.com/we/wordeditorframe.aspx?ui=pl-pl&rs=pl-PL&wopisrc=https%3A%2F%2Fpfronwarszawa.sharepoint.com%2Fsites%2FPostpowaniaPZP-SOW%2F_vti_bin%2Fwopi.ashx%2Ffiles%2F5d81e8281ee74745a236b2da3e8433d8&wdenableroaming=1&mscc=1&hid=545501c5-c389-4b12-b031-96ea9e44371c.0&uih=teams&uiembed=1&wdlcid=pl-pl&jsapi=1&jsapiver=v2&corrid=058a4b79-059a-4081-acf7-43a4e29f06a6&usid=058a4b79-059a-4081-acf7-43a4e29f06a6&newsession=1&sftc=1&uihit=UnifiedUiHostTeams&muv=v1&accloop=1&sdr=6&scnd=1&sat=1&rat=1&sams=1&mtf=1&sfp=1&halh=1&hch=1&hmh=1&hsh=1&hwfh=1&hsth=1&sih=1&unh=1&onw=1&dchat=1&sc=%7B%22pmo%22%3A%22https%3A%2F%2Fwww.office.com%22%2C%22pmshare%22%3Atrue%7D&ctp=LeastProtected&rct=Medium&wdorigin=TEAMS-ELECTRON.teamsSdk.openFilePreview&wdhostclicktime=1667984849089&instantedit=1&wopicomplete=1&wdredirectionreason=Unified_SingleFlush#_Linki);
3. 2.5.5 Rozmiar celu dotykowego: wielkość kontrolki (poziom AAA). Wielkość obiektu, który trzeba dotknąć lub kliknąć myszą, musi być na tyle duża, by Użytkownik mógł łatwo trafić palcem lub kursorem myszy.

## **7. Dokumenty.**

Wszystkie dokumenty, które będziesz publikował w Systemie, muszą spełniać wymagania WCAG w odniesieniu do dokumentów cyfrowych (zalecenia w tym zakresie dostępne są na stronie W3C opisujące techniki WCAG dla PDF - <https://www.w3.org/TR/WCAG20-TECHS/pdf>).

Jesteś zobowiązany do każdorazowej adaptacji dokumentów dostarczanych przez Zamawiającego oraz prawidłowego (zgodnego z wytycznymi WCAG) przygotowania Dokumentacji Użytkownika.

Dokumentację Użytkownika przygotujesz zgodnie z zasadami prostego języka umieszczonymi w serwisie gov.pl (<https://www.gov.pl/web/sluzbacywilna/prosty-jezyk>).

## **Załącznik nr 3 do OPZ - Wymagania dla Dokumentacji.**

1. Wymagania Ogólne

1. W terminie 10 Dni Roboczych od dnia uzyskania dostępu do Repozytorium Wykonawca zapozna się z dokumentacją i sposobem organizacji i zarządzania Repozytorium Projektowego oraz przedstawi Zamawiającemu propozycję optymalizacji ww. Repozytorium. Zamawiający zastrzega sobie prawo do wyboru poszczególnych propozycji przedstawionych przez Wykonawcę.
2. Wykonawca w terminie 10 Dni Roboczych od dnia zaakceptowanej przez Zamawiającego propozycji optymalizacji Repozytorium Projektowego wprowadzi je do Repozytorium.
3. Wykonawca zobowiązuje się do prowadzenia Repozytorium Projektowego w oparciu o środowisko dostarczone przez Zamawiającego. 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 na przykład JIRA, narzędzia pracy grupowej na przykład Microsoft Teams, Sharepoint.
4. W Repozytorium Projektowym, w sposób szczególny będą wyróżniane aktualne wersje dokumentacji projektowej. Dokumenty projektowe będą zawierały historię zmian oraz dane identyfikacyjne, w tym numer wersji.
5. Wykonawca odpowiedzialny jest za sporządzanie notatek ze spotkań projektowych i umieszczanie ich w Repozytorium Projektowym.
6. W uzupełnieniu do dokumentacji w Repozytorium Projektowym, Wykonawca prowadzi i utrzymuje następujące repozytoria i bazy wchodzące w jego skład:
   1. „Dokumentacja użytkowa” zawierająca dokumentację dla użytkowników zewnętrznych (Wnioskodawcy), użytkowników wewnętrznych (PFRON) oraz administratora merytorycznego (PFRON) obejmująca:
      1. PUZ – podręcznik użytkownika zewnętrznego,
      2. PUW – podręcznik użytkownika wewnętrznego,
      3. PAM – podręcznik administratora merytorycznego.
   2. Architektura i schemat bazy danych – zawierające model architektury, obejmujący opis:
      1. PIS - Projekt Infrastruktury Systemu
      2. PMS – Projekt Modułu Systemu
      3. SKZ - Skomentowany Kod Źródłowy
      4. DKZ – Dokumentacja Kodu Źródłowego
      5. OAR - Opis Architektury Rozwiązania,
   3. Dokumentacja analityczna:
      1. SWB - Specyfikacja Wymagań Biznesowych
      2. SWSI - Specyfikacja Wymagań Systemu Informatycznego
      3. PTS - Plan Testów Systemu,
      4. SPST – Specyfikacja Przypadków i Scenariuszy Testowych
      5. RT - Raport z Testów
   4. Eksploatacja:
      1. IKTP – Instrukcja Kompilacji i Tworzenia Pakietu Instalacji,
      2. OWO – Opis Wersji Oprogramowania,
      3. PAS – Podręcznik Administratora Systemu,
      4. PES – Podręcznik Eksploatacji Systemu,
      5. PZW - Plan Zarządzania Wymaganiami,
   5. Dokumentacja Środowiska Zapasowego
   6. Dokumentacja API.

a) API EGW – SOW

b) API EGW/GW - iPFRON+

1. Dokumentacja Systemu GW/EGW obejmuje opisane poniżej kategorie dokumentów wraz z ich minimalnym zakresem informacyjnym.
2. Specyfikacja Wymagań Biznesowych (SWB).

Minimalny zakres informacyjny SWB:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| 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). |
| Obszary funkcjonalne | Wykaz obszarów funkcjonalnych, których realizacja jest wspierana przez System. Obszary funkcjonalne muszą być opisane następującymi atrybutami:   * identyfikator obszaru, * nazwa obszaru, * opis obszaru, * właściciel obszaru, * uczestnicy obszaru, * podstawa prawna powiązana z obszarem, * opis funkcji realizowanych w ramach danego obszaru, * powiązanie z obiektami biznesowymi związanymi z danym obszarem. |
| Aktorzy biznesowi | Wykaz aktorów biznesowych powiązanych z obszarem.  Każdy aktor zostanie opisany przez:   * identyfikator aktora, * opis aktora. |
| Wymagania biznesowe | Wykaz wymagań biznesowych powiązanych z obszarem funkcjonalnym i związanych z dalszą realizacją w Systemie. Wymagania muszą być opisane następującymi atrybutami:   * identyfikator wymagania, * nazwa wymagania, * realizowany cel (opcjonalne), * kryteria spełnienia (opcjonalne), * dokument źródłowy (opcjonalne), * szczegółowa treść wymagania, * źródło pochodzenia danego wymagania (opcjonalne), * wersja wymagania, * data jego dodania do Systemu (opcjonalne), * okres obowiązywania, * moduły, których dotyczy wymaganie; * status określających m.in. czy wymaganie jest otwarte, zaakceptowane, zrealizowane czy unieważnione, * autor - imię i nazwisko autora, autorów, * kategoria (Funkcjonalne, Pozafunkcjonalne), * priorytet (1 - Niski, 2 - Średni, 3 – Ważny), * etap realizacji: identyfikator umowy, nr zmiany, * słowa kluczowe, * uwagi (opcjonalne), * załączniki plikowe (opcjonalne). |
| Procedury realizacyjne, wytyczne, instrukcje | Wykaz dokumentów uwzględnionych przez Wykonawcę w toku projektowania Systemie. |

1. Opis Architektury Rozwiązania (OAR).

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Sposób dokumentowania architektury | Opis przyjętego sposobu dokumentowania architektury Systemu – w szczególności objaśnienie wykorzystanych perspektyw wraz z charakterystyką ich zawartości. Prezentowane jest objaśnienie zastosowanych perspektyw. |
| Syntetyczne przedstawienie architektury rozwiązania | Syntetyczny opis architektury Systemu przedstawiający za pomocą diagramu następujące warstwy:  warstwę biznesową złożoną z obszarów funkcjonalnych i głównych aktorów, warstwę modułów i ich zasadniczych komponentów,  warstwę wdrożenia obejmującą lokalizacje oraz główne węzły. |
| Kluczowe wymagania architektoniczne | Wykaz istotnych wymagań funkcjonalnych i pozafunkcjonalnych mających istotny wpływ na obecny kształt architektury Systemu 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 Systemu. Decyzje architektoniczne mogą dotyczyć np.: aspektów funkcjonalnych oraz pozafunkcjonalnych, 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 Systemu. Należy wskazać miejsce i specyfikę jego zastosowania w Systemie na ogólnym poziomie (tj. modułów).  Każdy wzorzec architektoniczny zostanie opisany przez:   * identyfikator wzorca, * opis wzorca (lub jego nazwę, jeśli jest to wzorzec powszechnie znany). |
| Perspektywa biznesu | Opis i przedstawienie na diagramie obszarów funkcjonalnych, aktorów z nimi powiązanych oraz obiektów biznesowych. |
| Perspektywa zachowania | Prezentacja działania Systemu za pomocą wybranych scenariuszy biznesowych, stanowiących istotną, centralną funkcjonalność Systemu lub mających znaczący wpływ na architekturę całego Systemu. Scenariusze biznesowe powinny być powiązane z obszarami funkcjonalnymi, modułami i ich komponentami.  Każdy scenariusz biznesowych zostanie opisany przez:   * identyfikator, * syntetyczny opis, * diagram zachowania. |
| Perspektywa logiczna | Opis struktury logicznej Systemu w kontekście jego modułów, podmodułów i komponentów składowych, warstw (np. prezentacji, logiki biznesowej, baz danych) oraz wzajemnych powiązań pomiędzy poszczególnymi warstwami i elementami. Każdy z modułów Systemu powinien być skrótowo opisany. Dla każdego podmodułu lub komponentu należy podać co najmniej jego nazwę, realizowane funkcjonalności i ogólną definicję interfejsów, które odpowiadają za wzajemne relacje pomiędzy komponentami Systemu oraz z innymi systemami. Należy również odnieść się do kwestii architektonicznych dotyczących sposobów integracji komponentów Systemu  z zewnętrznymi komponentami i systemami.  Opis zostanie przedstawiony w postaci tekstowej oraz graficznej  z wykorzystaniem m.in. diagramów komponentów. |
| Perspektywa danych | Opis sposobu przechowywania danych w Systemie,  w szczególności:   * modelu zasadniczych obiektów danych w Systemie w postaci diagramu klas oraz ich opis, * mapowania istotnych, nietrywialnych trwałych klas projektowych na model danych, w postaci tabeli z mapowaniem i diagramów klas, * części Systemu, które zostały zrealizowane bezpośrednio w bazie danych np. w postaci procedur, w postaci tabelarycznej wykazu procedur i powiązań z obiektami danych, * prezentacji kwestii zapewnienia transakcyjności, współbieżności, rozproszenia itp., w postaci opisu słownego przedstawiającego sposób zapewniania danego zagadnienia.   Należy wymienić kluczowe wymagania i decyzje architektoniczne mające wpływ na perspektywę danych. |
| Perspektywa fizyczna | Opis topologii poszczególnych elementów Systemu – zarówno sprzętowych, jak i programowych, w szczególności należy przedstawić:   * lokalizacje oraz elementy Systemu dla danej lokalizacji, * charakterystykę połączeń (w tym sieciowych) pomiędzy poszczególnymi elementami systemu informatycznego.   Perspektywa fizyczna zostanie opisana przez odrębny diagram wdrożenia dla każdej z lokalizacji.  Należy wymienić kluczowe wymagania i decyzje architektoniczne mające wpływ na perspektywę fizyczną.  Zamawiający dostarczy Wykonawcy informację o elementach perspektywy fizycznej pozostającej w gestii Zamawiającego lub podmiotów trzecich. |
| Zestawienie infrastruktury programowej i sprzętowej | Charakterystyka niezbędnej infrastruktury programowej i sprzętowej wykorzystywanej przez oprogramowanie Systemu.  Zestawienie infrastruktury programowej obejmuje oprogramowanie systemowe (np. system operacyjny, serwery aplikacji, oprogramowanie integracyjne, oprogramowanie baz danych itp.). Zestawienie zostanie opisane przez tabelę zawierającą:   * nazwę oprogramowania, typ, wersję, producenta, * liczbę i rodzaj wykorzystywanych licencji.   Zestawienie infrastruktury sprzętowej zostanie opisane przez tabelę zawierającą:   * nazwę urządzenia, typ, wersję, producenta, * liczbę wykorzystywanych urządzeń danego typu.   Zamawiający dostarczy Wykonawcy informację o elementach infrastruktury programowej i sprzętowej pozostającej w gestii Zamawiającego lub podmiotów trzecich z wykorzystaniem przygotowanych przez Wykonawcę arkuszy specyfikujących ww. infrastrukturę (tj. szablonu). |

1. Projekt Modułu Systemu (PMS)

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 Systemu. |
| 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 PFRON, jak i zewnętrznymi (w tym np. z platformą integracyjną, przez którą następuje komunikacja z innymi systemami).  Dla każdej integracji pomiędzy modułami Systemu należy opisać:   * Moduły, których dotyczy dana integracja, * Cel integracji modułów (jakie zadanie spełnia dana integracja), * Charakterystyka i sposób wymiany danych (np. komunikatów, plików). |
| Model komponentów oprogramowania | Opis poszczególnych podmodułów (o ile to zasadne) oraz komponentów oprogramowania. Zamieszczony opis prezentuje ogólną informacje o ww. elementach oraz sposób ich wzajemnej interakcji.  Zakres informacji powinien zostać dostosowany do konkretnego typu podmodułu czy komponentu oprogramowania. Poniżej został zamieszczony minimalny zakres informacji:   * opis elementów (tj. podmodułów, komponentów), * opis powiązań pomiędzy elementami (tj. identyfikacja interfejsów), * ogólny opis interfejsów: * identyfikator interfejsu; * nazwa interfejsu; * syntetyczny opis interfejsu; * rodzaj interfejsu (np. plikowy, WebAPI); * główne scenariusze działania, * powiązania elementów z funkcjami biznesowymi, w postaci realizacji zależności ww. funkcji do danego elementu bądź w innej formie ustalonej z Zamawiającym.   Zamawiający może wyrazić zgodę na inny zakres informacji. Informacje mogą zostać umieszczone w zewnętrznych dokumentach, będących załącznikami. |
| Logiczny model danych | Opis logicznego modelu danych wykorzystywany w ramach modułu. Opis zawiera diagram modelu danych, jak i jego specyfikację.  Model jest dalszą dekompozycją danych przedstawionych w OAR i zachowuje z nim spójność. |
| 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 można zaliczyć skrypty, konfiguracje, pliki, kody źródłowe, fizyczne zbiory/bazy danych, komponenty przetwarzające itd.  Diagramy rozlokowania komponentów powinny być tworzone zarówno na poziomie logicznym, jak i fizycznym.  Zamawiający dostarczy Wykonawcy informację o elementach obiektów w węzłach (nodes, typ CPU, RAM, urządzenia, serwery, itp.), jak również połączenia sieciowe w zakresie poszczególnych modułów Systemu z wykorzystaniem przygotowanych przez Wykonawcę arkuszy specyfikujących ww. infrastrukturę (tj. szablonu). |

1. Projekt Infrastruktury Systemu (PIS)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Rodzaje środowisk Systemu | 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 Systemu oraz schemat blokowy Systemu |
| Opis poszczególnych środowisk | Opis wymagań niefunkcjonalnych:   * zasięg systemu [opis], * liczba użytkowników w całym Systemie [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 Systemu (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 Systemu 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 Systemu objętych backup’em i archiwizacją. |
| Koncepcja Disaster Recovery | Opis koncepcji odtworzenia Systemu po awarii dla każdego ze środowisk Opis Architektury Rozwiązania Disaster Recovery |
| Wymagane licencje | Wykaz zbiorczy wszystkich potrzebnych do wdrożenia Systemu 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. |

1. Specyfikacja Wymagań Systemu Informatycznego (SWSI)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Ogólny opis systemu | Ogólny opis głównych czynników wpływających na kształt Systemu i jego wymagania, w ramach którego musi być przedstawiony:   * Kontekst (środowisko), w jakim będzie działał system. * Charakterystykę użytkowników systemu. * 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 System z wymagań biznesowych oraz innych czynników sterujących. |
| Wymagania  funkcjonalne | Wykaz wymagań funkcjonalnych na System.  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, * waga danego wymagania w kontekście Systemu (np. kluczowe, istotne, normalne, wspomagające), * stabilność danego wymagania w kontekście Systemu (np. wysoka, średnia, niska), * zgłaszający – podmiot bądź osoba, która zgłosiła wymaganie, * realizacja – wskazanie na wersję Systemu, w której wymaganie zostało odebrane, * część systemu, której dotyczy, * wersja wymagania, data jego powstawania i data jego ostatniej modyfikacji.   Oprócz opisu tekstowego należy przygotować diagram prezentujące wymagania.  Na powyższych diagramach przedstawione zostaną wzajemne relacje pomiędzy wymaganiami oraz powiązania z pozostałymi elementami modelu Repozytorium Architektury (np. komponenty, funkcje biznesowe). |
| Wymagania  niefunkcjonalne | Wykaz wymagań niefunkcjonalnych na system informatyczny.  Muszą być uwzględnione co najmniej następujące rodzaje wymagań niefunkcjonalnych:   * wymagania na niezawodność Systemu, * wymagania wydajnościowe na System, * wymagania dotyczące architektury Systemu (tj. konstrukcji systemu), * wymagania dotyczące bezpieczeństwa Systemu (w szczególności wymagania dotyczące rozliczalności – zakresu informacyjnego danych zapisywanych do dzienników usług i dzienników danych Systemu) oraz dotyczące ochrony przetwarzanych danych (w szczególności danych osobowych), * wymagania dotyczące użyteczności (usability) Systemu, * wymagania dotyczące Oprogramowania Systemowego (np. platform systemowych, platform baz danych, platform aplikacji, itp.) i urządzeń którymi dysponuje Zamawiający z punktu widzenia Systemu, * wymagania dotyczące interfejsów, które musi posiadać System (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 użytkowanie, 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. |
| Projekty ekranów | Projekty ekranów i diagramy nawigacji w postaci załącznika do dokumentu. Powiązanie ww. informacji z wymaganiami. |
| Projekt przetwarzania dokumentów | Projekt przetwarzania dokumentów w postaci załącznika do dokumentu.  Powiązanie ww. informacji z wymaganiami. |
| Załączniki | Ewentualne załączniki doprecyzowujący wymagania. |

1. Plan Zarządzania Wymaganiami (PZW)

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 | Dla każdego zdefiniowanego typu wymagań należy określić atrybuty, przy pomocy których będzie on opisywany. Występują co najmniej następujące atrybuty:   * identyfikator, * nazwa wymagania, * opis wymagania, * źródło pochodzenia danego wymagania, * status wymagania określający jego cykl życia, * waga danego wymagania w kontekście Systemu (np. kluczowe, istotne, normalne, wspomagające), * stabilność danego wymagania w kontekście systemu (np. wysoka, średnia, niska), * zgłaszający –podmiot bądź osoba, która zgłosiła wymaganie, * realizacja – wskazanie na wersję Systemu, w której wymaganie zostało odebrane, * część systemu, której dotyczy, * wersja wymagania, data jego powstawania i data jego ostatniej modyfikacji.   Atrybuty inne niż: identyfikator, nazwa, opis, status mogą za zgodą Zamawiającego posiadać wartości domyślne - brak określenia wartości oznacza, że przyjmuje się wartość domyślną. Można w ten sposób wskazywać jedynie odchylenia od typowej sytuacji. |
| Zarządzanie zmianą wymagań | Opis procesu, w ramach, którego propozycje nowych wymagań i zmian istniejących już wymagań są przedstawiane, przeglądane i oceniane.  Proces ten powinien obejmować działania dotyczące uzgadniania zmian wymagań z Zamawiającym. |

1. Instrukcja Kompilacji i Tworzenia Pakietu Instalacji (IKTP)

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 Systemu wymaganych od Wykonawcy. W przypadku zastosowania mechanizmów automatyzacji procesu instalacji, należy przedstawić również opis manualnego instalowania Systemu. Należy opisać instrukcje instalowania i wycofywania nowej wersji Systemu. |

1. Opis Wersji Oprogramowania (OWO)

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. |

1. Skomentowany Kod Źródłowy (SKZ)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wykaz kodów  źródłowych | Wykaz wszystkich Kodów Źródłowych wytworzonych w ramach realizacji Umowy i koniecznych do modyfikacji oraz dalszego rozwoju zbudowanych komponentów oprogramowania Systemu. 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. |

1. Dokumentacja Kodu Źródłowego (DKZ)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Treść dokumentacji Kodu Źródłowego | Dokumentacja kodu w formacie html, wygenerowana stosownym dla Systemu 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. |

1. Plan Testów Systemu (PTS)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Charakterystyka testowanego systemu | Opis tła Systemu wraz ze szczegółowym odniesieniem do istotnych implikacji na rzecz procesu testowania. Powinny zostać zawarte w syntetyczny sposób, takie informacje jak:   * Kluczowe problemy, jakie System ma rozwiązać. * Korzyści płynące z wdrożenia danego rozwiązania. * Planowana architektura rozwiązania. * Krótka historia projektu, w ramach którego realizowany jest System. * Lista produktów, które zostały zidentyfikowane jako przedmiot testów. |
| 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.  Dobór i uzasadnienie testów – Opis w jaki sposób zostanie przeprowadzony dobór testów przeznaczonych do włączenia do zakresu 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. |
| Ryzyka i zależności związane z testami | Wykaz ryzyk, które mogą mieć wpływ na efektywną realizację „Planu testów systemu”. Dla każdego z ryzyk należy określić strategię uniknięcia i mitygacji. Należy określić również prawdopodobieństwo wystąpienia ryzyka i wpływ ryzyka na proces testowania, o ile wystąpi.  Wykaz zależności wykrytych podczas tworzenia „Planu testów systemu”, które mogą mieć wpływ na jego efektywną realizację w przypadku nie wzięcia ich pod uwagę. Zazwyczaj odnoszą się one do działań na ścieżce krytycznej, współpracy z innymi zespołami bądź członkami zespołów niezwiązanych z projektem, planowania, współzależności innych planowanych zadań, dostępności innych elementów projektowych. |
| 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 dokumentacji potestowej oraz wyników testów (tj. Raportów z Testów).  Opis sposobu raportowania i eskalowana problemów związanych z testami, oraz opis procesu ich rozwiązywania.  Opis procesu kontroli cyklu testów przez kierownictwo. |

1. Specyfikacja Przypadków i Scenariuszy Testowych (SPST)

Minimalny zakres informacyjny:

|  |  |
| --- | --- |
| Element zakresu informacyjnego (np. rozdział) | Opis elementu zakresu informacyjnego |
| Wykaz zestawów testów | Wykaz zestawów testów wykonywanych w ramach tworzenia Systemu 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) Systemu 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 Systemu, 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 Systemu, 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 Systemu informatycznego (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. |

1. Raport z Testów (RT)

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. |

1. Podręcznik Administratora Systemu (PAS)

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 Systemem. |
| Określenie wersji | Określenie wersji Systemu / Produktów, których dotyczy wraz z wersjami systemu operacyjnego/bazy danych/serwerów aplikacyjnych itp. |
| Wytyczne do planu eksploatacji | Wytyczne do Planu Eksploatacji Systemu sugerujące np. harmonogram wykonywanych okresowo instrukcji. |
| Instrukcje administratorskie | Opis instrukcji obsługi wszystkich elementów Systemu (w tym uwzględniające mechanizmy bezpieczeństwa przetwarzania danych) niezbędnych dla eksploatacji i utrzymania Systemu.  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 Systemu. |
| Komunikaty systemu | Opis komunikatów Systemu wspierających jego administrowanie (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 Systemem. |

1. Podręcznik eksploatacji systemu (PES)

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 Systemu:   * 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 Systemu * Inne wymagania specyficzne dla Systemu. |

## **2. Organizacja, formatowanie, komentowanie i utrzymanie Kodu Źródłowego.**

# 1. Przechowywanie Kodu Źródłowego.

# 1.1 Repozytorium Kodu Źródłowego

Departament ds. Teleinformatyki prowadzi i nadzoruje Repozytorium Kodu Źródłowego. W przypadku projektów realizowanych przez firmy trzecie, pracownicy tych firm są odpowiedzialni za zarządzanie projektem i Kodem Źródłowym w repozytorium. W przypadku prac wykonywanych przez pracowników PFRON, taki obowiązek leży po stronie Funduszu. Repozytorium Kodu Źródłowego oparte jest na platformie GIT z wykorzystaniem interfejsu graficznego GitLab.

Zasady korzystania i prowadzenia repozytorium kodu źródłowego określają poniższe zapisy:

1. Każdy realizowany w PFRON projekt musi posiadać własną przestrzeń w systemie GitLab, tzw. projekt.
2. Projekt w GitLab musi mieć nazwę zgodną z nazwą projektu realizowanego w organizacji.
3. Kody źródłowe przekazywane są w formie zapewniającej kontrolę wersji.
4. Repozytorium kodu nie powinno być traktowane jako archiwum, wymagane jest ciągłe dostarczanie kolejnych wersji Kodu Źródłowego, zgodnie z procesem wytwórczym. Nie akceptowalna jest forma rzadkiego zatwierdzania commitów z dużą ilością linii Kodu Źródłowego.
5. W przypadku gdy, do aplikacji wykorzystane zostały Kody Źródłowe lub biblioteki innych dostawców a następnie zostały one zmodyfikowane na potrzeby projektu, bezwzględnie należy dodać do repozytorium kod wejściowy biblioteki lub modułu, a następnie wersjonować realizowane w nim zmiany.
6. Każdy commit powinien zawierać ogólny opis (jakiej funkcjonalności, pakietu dotyczy, do czego służy, dlaczego coś było modyfikowane - zmieniane) zmian oraz autora i wersję systemu, którego dotyczy.
7. Każdy commit powinien zawierać również informacje umożliwiające łatwe powiązanie poszczególnych aktualizacji Repozytorium Kodu Źródłowego z dokumentacją projektu, w tym dokumentacją zmian i dokumentacją Kodu Źródłowego.

### 1.2 Organizacja Repozytorium Kodu Źródłowego.

Struktura repozytorium powinna posiadać podział na moduły aplikacji, usługi integracyjne, konfiguracje i pliki specyficzne dla środowisk, strukturę bazy danych oraz obiekty bazodanowe, w tym pakiety, procedury, funkcje, wyzwalacze.

Strategia tworzenia gałęzi (ang. Branching Strategy) w narzędziu GitLab powinna być zgodna z zasadami GitFlow (<https://datasift.github.io/gitflow/IntroducingGitFlow.html>). Główną gałęzią musi być *master*. Bieżące prace rozwojowe powinny być prowadzone w oddzielnej gałęzi, na przykład o nazwie *develop*. Wytwarzanie pojedynczych nowych funkcjonalności w ramach prac rozwojowych odbywać się powinno w gałęziach *feature* (ang. feature branches). Prace programistyczne związane z usuwaniem błędów prowadzone są na osobnej gałęzi, na przykład *HotFIX*. Po zakończeniu prac rozwojowych lub utrzymaniowych i wdrożeniu zmian na środowisko produkcyjne danego systemu kod źródłowy z odpowiedniej gałęzi musi być połączony z gałęzią *master*.

# 2. Komentowanie Kodu Źródłowego.

## 2.1 Konwencja nazewnictwa.

Projekty realizowane w PFRON muszą posiadać opracowaną i stosowaną w ramach danego projektu konwencję nazewniczą. Konwencja musi zapewnić minimum:

1. Usystematyzowanie, uporządkowanie i ujednolicenie nazewnictwa w ramach danego projektu.
2. Umożliwiać łatwe rozróżnianie (po nazwie) typu zmiennej, stałej, kolumny w bazie, wartości zwracanej przez funkcję, metodę itp.
3. Nazwy mają być znaczące - informować o tym, do czego dany element jest wykorzystywany.
4. Konwencja powinna być opracowana i opisana w taki sposób, by programista pisząc kod nie miał wątpliwości jakich nazw ma używać.
5. Konwencja powinna uwzględniać instalacje testowe, tak aby nie wprowadzać chaosu pomiędzy np. nazwami/identyfikatorami elementów systemu dla instalacji testowej i produkcyjnej.

Opracowana konwencja nazewnicza musi uwzględniać minimum następujące elementy i twory programistyczne:

1. Wszystkie elementy Kodu Źródłowego, w tym pakiety, biblioteki, klasy, metody, pola klas, stałe, zmienne, funkcje, procedury itp.
2. Wszystkie składniki systemu baz danych, w tym nazwa baza danych, nazwy schematów, tabele, kolumny, funkcja, pakiet, wyzwalacz, tabele tymczasowe, zmienne itp.
3. Innych składników systemu, takich jak API, zmiennych formatu XML oraz JSON itp.

Nazwy obiektów programistycznych i bazodanowych, w tym nazwy zmiennych, metod, klas muszą być intuicyjne, jednoznaczne i napisane w języku polskim. W przypadku gdy nazwy będą zapisywane w języku angielskim, ich polskie odpowiedniki muszą być zapisywane w komentarzu związanym z danym obiektem programistycznym lub bazodanowym. W przypadku nazw klas, metod, zmiennych, funkcji, obiektów bazodanowych (tabele, kolumny, procedury, funkcje, zmienne itp.) należy obowiązkowo unikać nazw jednoliterowych oraz skrótów zrozumiałych w danym momencie wyłącznie dla programisty piszącego dany kod.

Wyjątkiem od powyższych zasad jest kod źródłowy bibliotek i frameworków wytworzonych przez firmy trzecie i wykorzystywanych w ramach danego projektu. W przypadku modyfikacji ww. bibliotek lub frameworków, zmiany wprowadzone do kodu źródłowego muszą spełniać już wymagania opisane w niniejszym dokumencie.

## 2.2 Formatowanie Kodu Źródłowego.

Dla każdego projektu należy zdefiniować formatowanie Kodu Źródłowego. Wszyscy, biorący udział w projekcie programiści muszą obligatoryjnie stosować jednolite formatowanie.

Kod źródłowy musi spełniać wymagania dotyczące kodu samo komentującego, powinien być sformatowany w sposób prosty, przejrzysty oraz jednolity.

Przykłady standardów formatowania dla Kodu Źródłowego:

* JAVA -Google Java Style Guide (<https://google.github.io/styleguide/javaguide.html>)
* PHP – PSR PHP Standard Recommendations (<https://www.php-fig.org/psr/>)
* Python – PEP8 (<https://www.python.org/dev/peps/pep-0008/>)
* PostgreSQL – Coding Standard for SQL and PL/SQL (<https://www.williamrobertson.net/documents/plsqlcodingstandards.html>)

## 2.3 Komentowanie Kodu Źródłowego.

Sposób komentowania i jakość samych komentarzy ma bezpośrednie znaczenie dla jakości Kodu Źródłowego danego systemu.

Główna reguła, która musi być stosowana w przypadku konstruowania komentarzy do kodu źródłowego brzmi następująco: Należy komentować Kod Źródłowy w taki sposób, jakiego tworzący komentarz programista sam by oczekiwał - co do zakresu, podejścia, zawartości, szczegółowości, konsekwencji w stylu, spójności konwencji itd.

### 2.3.1. Minimalne wymagania dotyczące komentowania Kodu Źródłowego.

1. każda klasa (aplikacji, formularzy, raportów itd.) musi zawierać kilkuzdaniowy komentarz opisujący, jakiego rodzaju obiekty generuje i jaka jest ich semantyka,
2. każdy atrybut każdej klasy musi zawierać komentarz opisujący jego znaczenie,
3. każda metoda każdej klasy musi zawierać komentarz opisujący, do czego metoda służy, jakie ma parametry (co one oznaczają) oraz jaką wartość zwraca,
4. każde wywołanie metody obiektu musi zawierać komentarz objaśniający, czemu służy,
5. każde wykonanie instrukcji SQL musi zawierać komentarz objaśniający, czemu służy,
6. każda tabela oraz kolumna musi posiadać komentarz objaśniający jakie dane są przechowywane w danej tabeli lub kolumnie, jeśli sama nazwa nie posiada odpowiedniej informacji,
7. każdy obiekt bazodanowy, w tym, pakiet, funkcja, wyzwalacz itp. musi zawierać komentarz objaśniający, czemu służy.

Każdy obiekt programistyczny, taki jak pakiet, klasa, metoda, procedura, funkcja, pakiet bazodanowy, procedura bazodanowa, funkcja bazodanowa itp. zawiera opis nagłówkowy, zawierający przynajmniej poniższe informacje:

* autor,
* numer wersji obiektu,
* numer wersji systemu,
* data utworzenia i data ostatniej modyfikacji,
* lista i opis argumentów (jeśli takie posiada),
* opis zwracanej wartości (jeśli zwraca wartość) lub wyniku działania,
* krótki, ale wyczerpujący opis działania, słowny opis użytego algorytmu,
* zwracane nieobsłużone wyjątki (jeśli takie mogą się pojawić),
* ewentualnie odwołanie do dokumentacji systemu.

Komentarze wewnątrz pakietów, klas, procedur, funkcji, pakietów bazodanowych, procedur bazodanowych, funkcji bazodanowych itp. Muszą być umieszczone w przypadku, gdy:

* wyjaśnienie kodu, który nie jest oczywisty na pierwszy rzut oka,
* wyjaśnienie intencji, które ciężko ująć w kodzie,
* ostrzeżenie o konsekwencjach użycia danej funkcjonalności,
* wyjaśnienie niuansów procesów biznesowych, które realizuje program.

Komentarze Kodu Źródłowego należy uzupełniać o znaczniki wymagane przez narzędzia służące do automatycznego generowania dokumentacji Kodu Źródłowego wprost z plików źródłowych. W przypadku języka programowania PHP, komentarze powinny być opisane sposób pozwalający na wygenerowanie dokumentacji za pomocą narzędzia PHPDoc, phpDocumentor lub Doxygen. Dodatkowe wymagania dotyczące komentowania Kodu Źródłowego i znaczników interpretowanych przez dane narzędzie znajdują się w jego dokumentacji.

## 2.4 Dokumentacja Kodu Źródłowego.

Niezależnie od komentarzy znajdujących się w Kodzie Źródłowym i na tej podstawie wygenerowanej dokumentacji, wykonawcy realizujący projekty programistyczne w Funduszu zobligowani są do utworzenia, aktualizacji i prowadzenia dokumentacji kodu źródłowego. Dokumentacja, o której mowa powyżej musi zawierać:

1. wykaz (wraz z adresami w Git), wszystkich Kodów Źródłowych koniecznych do generowania określonej wersji systemu. Do Kodów Źródłowych zalicza się również wszelkie dodatkowe zasoby takie jak skrypty, dane konfiguracyjne, frameworki itp.,
2. listę technologii wraz z wersją technologii, w których zostały wytworzone Kody Źródłowe. Dokumentacja musi być powiązana z konkretną wersją/wydaniem sytemu,
3. wygenerowaną automatycznie na podstawie Kodu Źródłowego, dokumentację Kodu Źródłowego przy użyciu wybranego dedykowanego narzędzia (np. javadoc). Dokumentacja jest pozyskiwana na podstawie odpowiednich znaczników wpisywanych w komentarze (o składni zgodnej z regułami narzędzia),
4. instrukcję generowania kodu wynikowego i tworzenia wersji instalacyjnej z wersji wynikowej (skompilowanej),
5. instrukcję konfiguracji środowiska do generowania kodów wynikowych,
6. specyfikację środowiska sprzętowo-systemowego wymaganego do przeprowadzenia procedury generacji kodu wynikowego,
7. listę narzędzi do przygotowywania wersji instalacyjnych wytworzonego oprogramowania (wersji pełnej, aktualizacji, łat) wraz z dokumentacją użytkowania i licencjami, o ile są wymagane,
8. w przypadku, gdy został wykorzystany framework firm trzecich, dokumentacja kodu źródłowego musi zawierać pełną dokumentację frameworka oraz instrukcję użytkownika i dla programistów,
9. w przypadku wykorzystania własnych standardowych bibliotek lub frameworków przez wykonawców dokumentacja kodu źródłowego musi również zawierać dokumentację ww. elementów systemu.

# 3. Weryfikacja Kodu Źródłowego.

## 3.1 Weryfikacja Kodu Źródłowego – wewnętrzna.

Częstotliwość weryfikacji Kodów Źródłowych – wymaganie ATK – 12 (OPZ).

Weryfikacja Kodu Źródłowego będzie prowadzona dla:

1. Modyfikacji, wymuszających zmianę wersji systemu lub poszczególnych jego komponentów. Weryfikacja Kodu Źródłowego stanowić będzie część procedury odbioru modyfikacji i jej wynik końcowy ma wpływ na podpisanie lub nie protokołu odbioru.
2. Dostaw Kodów Źródłowych realizowanych w ramach umów usług utrzymania i rozwoju, zgodnie z określonymi w przyszłej umowie terminami.
3. Dostarczonego Kodu Źródłowego nowo wytworzonego systemu, w ramach procedury odbioru.

W każdym przypadku osoby odpowiedzialne za realizację Umowy ustalają harmonogram oraz niezbędne zasoby osobowe i sprzętowe do przeprowadzenia weryfikacji. Weryfikację Kodów Źródłowych przeprowadzają pracownicy Zamawiającego i Wykonawcy w formie warsztatów. Wykonawca ma obowiązek zaprezentować wszystkie zmiany wprowadzone w kodzie w ramach realizacji usług ATiK i Rozwoju, w okresie, którego przegląd dotyczy.

## 3.1.1. Zakres wewnętrznej weryfikacji Kodu Źródłowego.

W celu zweryfikowania zgodności Kodów Źródłowych z wymaganiami zawartymi w niniejszym dokumencie należy przeanalizować Kod Źródłowy pod kątem poniższych zagadnień.

|  |  |  |
| --- | --- | --- |
| Lp. | Kryterium weryfikacji | Czy jest spełnione |
| 1 | Czy Kod Źródłowy jest przechowywany w GitLab? | Tak/Nie/Nie dotyczy |
| 2 | Czy projekt w GitLab ma nazwę zgodną z nazwą projektu? | Tak/Nie/Nie dotyczy |
| 3 | Czy sposób przechowywania Kodów Źródłowych zapewnia możliwość kontroli wersji? | Tak/Nie/Nie dotyczy |
| 4 | Czy commity wykonywane są z odpowiednią częstotliwością i są odpowiednio opisane? | Tak/Nie/Nie dotyczy |
| 5 | Czy Kody Źródłowe lub biblioteki innych dostawców znajdują się w GitLab? | Tak/Nie/Nie dotyczy |
| 6 | Czy struktura repozytorium w GitLab jest odpowiednio przygotowana i adekwatna do projektu? | Tak/Nie/Nie dotyczy |
| 7 | Czy są tworzone i utrzymywane odpowiednie gałęzie w repozytroium GitLab? | Tak/Nie/Nie dotyczy |
| 8 | Czy projekt ma ustaloną konwencję nazewniczą? | Tak/Nie/Nie dotyczy |
| 9 | Czy konwencja nazewnicza jest stosowana w projekcie? | Tak/Nie/Nie dotyczy |
| 10 | Czy konwencja nazewnicza spełnia wymagania zawarte w dokumencie Standard komentowania Kodu Źródłowego? | Tak/Nie/Nie dotyczy |
| 11 | Czy w projekcie został zdefiniowany standard formatowania Kodu Źródłowego? | Tak/Nie/Nie dotyczy |
| 12 | Czy standard formatowania jest stosowany w projekcie? | Tak/Nie/Nie dotyczy |
| 13 | Czy kod źródłowy spełnia wymagania dotyczące samo komentującego? | Tak/Nie/Nie dotyczy |
| 14 | Czy komentarze zawarte w Kodzie Źródłowym spełniają minimalne wymagania zawarte w dokumencie Standard komentowania Kodu Źródłowego? | Tak/Nie/Nie dotyczy |
| 15 | Czy została wytworzona dokumentacja Kodu Źródłowego? | Tak/Nie/Nie dotyczy |
| 16 | Czy dokumentacja Kodu Źródłowego jest aktualna? | Tak/Nie/Nie dotyczy |
| 17 | Czy dokumentacja Kodu Źródłowego zawiera elementy wskazane w dokumencie Standard komentowania Kodu Źródłowego? | Tak/Nie/Nie dotyczy |

Tab. 1 Lista kontrolna dla weryfikacji Kodu Źródłowego.

## 3.2 Weryfikacja Kodu Źródłowego – audyt zewnętrzny.

Na wniosek Kierownika Projektu lub innej osoby decyzyjnej weryfikacja Kodu Źródłowego może być przeprowadzona przez podmiot zewnętrzny.

Zakres audytu zewnętrznego będzie obejmować następujące obszary:

1. Obszar Kodu Źródłowego:
   1. Inspekcja kodu (code review) i wykorzystanie obowiązujących praktyk;
   2. Wykorzystanie przyjętych standardów komentowania i formatowania Kodu Źródłowego;
   3. Wydajność Kodu Źródłowego i zapytań SQL;
   4. Podatność na ataki;
   5. Skalowalność Kodu Źródłowego;
   6. Stopień odporności Kodu Źródłowego na wprowadzanie zmian, w tym refaktoryzację kodu (refactoring);
   7. Zasięg i pokrycie testami automatycznymi;
   8. Wykorzystane wzorce projektowe i poprawność ich użycia;
   9. Optymalizacja i normalizacja bazy danych;
   10. Ocena długu technologicznego;
2. Obszar procesu wytwórczego i zagadnień projektowych
   1. Architektura aplikacji;
   2. Wykorzystywana w projekcie technologia;
   3. Poprawność wykorzystania frameworków i bilbiotek;
   4. Analiza potencjalnych kosztów wprowadzenia modyfikacji podczas fazy utrzymania i rozwoju systemu teleinformatycznego;
   5. Jakość przyjętego w projekcie procesu wytwórczego.

Powyższy zakres audytu zewnętrznego Kodu Źródłowego będzie dostosowywany do indywidualnych potrzeb w ramach każdego zlecenia.

Wynikiem audytu zewnętrznego Kodu Źródłowego będzie raport zawierający zidentyfikowane niezgodności, problemy oraz rekomendacje i zalecenia.

**Załącznik nr 4 do OPZ - Wymagania dotyczące testów.**

WT1 – Wykonawca zobowiązany jest do przeprowadzenia testów jednostkowych na Środowisku Developerskim. Po zakończeniu testów Wykonawca zobowiązany jest do przedstawienia raportu z testów wraz z logiem z narzędzia, za pomocą którego były przeprowadzane testy, potwierdzającym wykonanie i liczbę poprawnie i błędnie przeprowadzonych testów.

WT2 - Wykonawca zobowiązany jest do wykonywania testów funkcjonalnych na Środowisku Testowym. Po zakończeniu testów Wykonawca jest zobowiązany do przedstawienia raportu z testów wraz ze scenariuszami testowymi oraz dowodów przeprowadzenia wyżej wymienionych testów. Dowodami mogą być zrzuty ekranu, wyciąg z logów Systemu, wyciąg z informacji z bazy danych Systemu.

WT3 - Wykonawca zobowiązany jest do przeprowadzenia testów wydajnościowych na Środowisku Testowym, o ile zostaną zlecone przez Zamawiającego. Po zakończeniu testów Wykonawca zobowiązany jest do przedstawienia raportu z testów.

WT4 - Wykonawca zobowiązany jest do przeprowadzenia testów bezpieczeństwa na Środowisku Testowym, o ile zostaną zlecone przez Zamawiającego. Po zakończeniu testów Wykonawca zobowiązany jest do przedstawienia raportu z testów.

## **Załącznik nr 5 do OPZ - Poziom świadczenia usług (SLA).**

Wykonawca zobowiązuje się świadczyć Przedmiot Umowy z zachowaniem następujących parametrów SLA (*Service Level Agreement*):

1. **Usługa Asysty Technicznej i Konserwacji.**

|  |  |
| --- | --- |
| **Kalendarz świadczenia usługi** | Przez 24 godziny 7 dni w tygodniu 365 dni w roku („24/7/365”).  Okno serwisowe w godzinach: 20.00 – 7.00.  Przyjmowanie i obsługa: Dni Robocze w godzinach: 8:00 – 17:00. |
| **Czasy realizacji** | |  |  |  |  | | --- | --- | --- | --- | | **Lp.** | **Nazwa Wady** | **Czas Naprawy[[1]](#footnote-2)** | **Czas Obejścia** | | 1. | Awaria | …… godzin zegarowych | 4 godzin zegarowych | | 2. | Błąd | ….. Godziny Robocze | 9 Godzin Roboczych | | 3. | Usterka | ….. Godzin Roboczych | Nie dotyczy | | 4. | Pytanie | 1 Dzień Roboczy | Nie dotyczy |   Definicje znajdują się w słowniku w pkt 1 Opisu Przedmiotu Zamówienia. |
| **Poziom dostępności usługi** | **RPDS** – **rzeczywisty poziom dostępności Systemu dla Użytkowników**  **RPDS ≥ 95 %**  RPDS obliczany jest wg wzoru:  **(TD - ∑ TN) / TD x 100[%]**  Gdzie:  TD – uzgodniony czas dostępności usługi, wynikający z kalendarza świadczenia usługi (20 Dni Roboczych w miesiącu (30 dni w miesiącu) po 13 Godzin Roboczych).  TN – czas trwania niedostępności usługi – przyjmujemy dopuszczalnie łączny czas niedostępności usługi na poziomie 13 godzin w miesiącu. Czas niedostępności usługi mierzony jest w przedziale od 7:00 do 20:00 każdego Dnia Roboczego miesiąca.  **Wyliczenie minimalnego progu RPDS:**  TN = 13 godzin w miesiącu.  TD = 20 Dni Roboczych x 13 godzin = 260 godzin (zgodnie z podanym kalendarzem świadczenia usługi).  RPDS = (260 godzin - 13 godzin) / 260 godzin x 100 = **95 %**  Przykład 1:  W miesiącu nie wystąpiła niedostępność.  RPDS = (260 - 0) / 260 \*100% = 100%  Przykład 2  W miesiącu wystąpiła jedna Awaria – naprawiona w 10 godzin.  RPDS = (260 - 10) / 260\* 100% =96,15%.  Przykład 3  W miesiącu wystąpiły 2 Wady Systemu - Awaria naprawiona w 26  godzin + Awaria naprawiona w 6 godzin.  TN = 26 + 6 = 32  RPDS = (260- 32) /260\* 100% = 87,69%.  Przykład 1 i 2 nie powoduje możliwości naliczenia kary umownej. Przykład 3  gdzie RPDS=87,69% jest poniżej wymaganego poziomu 95% - może być  naliczona kara umowna zgodnie z tabelą zawartą w Umowie w paragrafie nr 11. |
|  |
| **Terminowość** | **PDTN** – **poziom dotrzymania terminów naprawy lub odpowiedzi**  **PDTN ≥ 5 %**  PDTN jest obliczany wg wzoru:  **Σ (Wx \* Px) / Σ Wx [%]**  Gdzie:  Px – wskaźnik dotrzymania terminów naprawy zgłoszeń serwisowych dla danej Wady lub odpowiedzi, obliczany wg wzoru: Ax / Bx \* 100 [%].  Ax – liczba zgłoszeń serwisowych danej Wady lub odpowiedzi, dla których w danym miesiącu kalendarzowym nie został przekroczony Czas Naprawy.  Bx – liczba wszystkich zgłoszeń serwisowych danej Wady lub odpowiedzi, zarejestrowanych w danym miesiącu kalendarzowym.  Wx – waga zgłoszenia serwisowego danej Wady lub odpowiedzi.  W poniższej tabeli znajdują się wartości Wx i Px dla poszczególnych rodzajów Wad lub odpowiedzi.   |  |  |  | | --- | --- | --- | | **Nazwa Wady** | **Wx** | **Px[%]** | | Awaria | **15** | **4** | | Błąd | **10** | **8** | | Usterka | **5** | **2** |   **Wyliczenie maksymalnego progu PDTN (zaokrąglonego do dwóch miejsc**  **po przecinku):**  **PDTN = (15 \* 4 + 10 \* 8 + 5\*2) / (15 + 10 + 5) = 5,00**  **Przykłady wyliczeń PDTN[[2]](#footnote-3):**  **Przykład 1**   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Nazwa Wady** | **Nr zdarzenia** | **Wymagany Czas Naprawy**  **Wcn** | **Faktyczny Czas Naprawy**  **Fcn** | **Czas przekroczenia SLA (kolumna 3 – 4) (Jeżeli przekroczenie nie nastąpiło pozostawiamy puste pole)** | **Px** | **Wx** | | Awaria | 1 | 8 | 13 | 5 | 5 | 15 | | Błąd | 1 | 18 | 19 | 1 | 3 | 10 | | 2 | 18 | 20 | 2 | | 3 | 18 | 17 |  |   **Zgodnie z wzorem:**  PDTN = (15 \* 5 + 10 \* 3) / (15+ 10) = **4,20**  Przykład 2   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Nazwa Wady** | **Nr zdarzenia** | **Wymagany Czas Naprawy**  **Wcn** | **Faktyczny Czas Naprawy**  **Fcn** | **Czas przekroczenia SLA (kolumna 3 – 4) (Jeżeli przekroczenie nie nastąpiło pozostawiamy puste pole)** | **Px** | **Wx** | | Awaria | 1 | 8 | 14 | 6 | 6 | 15 | | Błąd | 1 | 18 | 22 | 4 | 12 | 10 | | 2 | 18 | 26 | 8 | | 3 | 18 | 9 |  |   **Zgodnie z wzorem:**  PDTN = (15 \* 6 + 10 \* 12) / (15 + 10) = **8,40**  Przykład 3   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Nazwa Wady** | **Nr zdarzenia** | **Wymagany Czas Naprawy**  **Wcn** | **Faktyczny Czas Naprawy**  **Fcn** | **Czas przekroczenia SLA (kolumna 3 – 4) (Jeżeli przekroczenie nie nastąpiło pozostawiamy puste pole)** | **Px** | **Wx** | | Awaria | 1 | 8 | 9 | 1 | 1 | 15 | | Błąd | 2 | 18 | 20 | 2 | 12 | 10 | | 3 | 18 | 28 | 10 | | 4 | 18 | 16 |  | | Usterka | 5 | 36 | 48 | 10 | 10 | 5 |   **Zgodnie z wzorem:**  PDTN = (15 \* 1 + 10 \* 120 + 5\*10) / (15 + 10 + 5) = **6,17**  Przykład 1 nie powoduje możliwości naliczenia kary umownej.  Przykład 2 i 3 gdzie PDTN=**8,40** i PDTN=**6.17** jest powyżej wymaganego poziomu 5,00 - może być naliczona kara umowna zgodnie z tabelą zawartą w paragrafie 11 ust. 10.2 Umowy. |
| **Lista i częstotliwość raportów** | * Rzeczywisty poziom dostępności Systemu (RPDS) (miesięcznie). * Poziom dotrzymania terminów Naprawy (PDTN) (miesięcznie). |

1. **Usługa Modyfikacji i Rozwoju.**

|  |  |
| --- | --- |
| **Kalendarz świadczenia Rozwoju** | Przez 24 godziny 7 dni w tygodniu 365 dni w roku („24/7/365”).  Przyjmowanie i obsługa: Dni Robocze w godzinach: 8:00 – 17:00 |
| **Czasy realizacji** | |  |  |  | | --- | --- | --- | | **Lp.** | **Nazwa** | **Czas realizacji usługi** | | 1. | Rozwój - etap I (analiza wstępna) | 10 Dni Roboczych | | 2. | Rozwój etap I – analiza pełna | Ustalany indywidualnie | | 3. | Rozwój - etap II | Ustalany indywidualnie | |

1. Czas naprawy zostanie dostosowany do oferty Wykonawcy [↑](#footnote-ref-2)
2. Przykłady wyliczenia PDTN zostaną dostosowane do Czasu Naprawy zaoferowanego przez Wykonawcę. [↑](#footnote-ref-3)