**Załącznik nr 1 do SWZ**

## **Opis przedmiotu zamówienia**

**Spis treści:**

[**Opis przedmiotu zamówienia 1**](#_Toc130289567)

[**1. Zastosowane definicje. 4**](#_Toc130289568)

[**2. Ogólny opis zamówienia. 13**](#_Toc130289569)

[**2.1. Przedmiot zamówienia. 13**](#_Toc130289570)

[**2.2. Gwarancja i rękojmia. 13**](#_Toc130289571)

[**2.3. Prawa własności intelektualnej. 13**](#_Toc130289572)

[**2.4. Licencje. 13**](#_Toc130289573)

[**2.5. Inne zobowiązania. 13**](#_Toc130289574)

[**2.6. Szczegółowe zasady realizacji zobowiązań Wykonawcy. 14**](#_Toc130289575)

[**2.7. Zobowiązanie do stosowania regulacji wewnętrznych PFRON. 14**](#_Toc130289576)

[**3. Informacje dotyczące Systemu e-PFRON2. 14**](#_Toc130289577)

[**3.1 Dziedzina Systemu. 14**](#_Toc130289578)

[**3.1. Elementy i funkcje Systemu 15**](#_Toc130289579)

[**3.2. Architektura Systemu 18**](#_Toc130289580)

[**4. Wymagania funkcjonalne. 31**](#_Toc130289581)

[**4.1 Wymagania dotyczące Usługi Asysty Technicznej i Konserwacji. 31**](#_Toc130289582)

[**4.1.1 Wymagania ogólne. 31**](#_Toc130289583)

[**4.1.2 Zasady obsługi Zgłoszeń. 34**](#_Toc130289584)

[**4.1.3 Zasady udzielania stałych konsultacji. 37**](#_Toc130289585)

[**4.1.4 Zasady aktualizacji Systemu. 38**](#_Toc130289586)

[**4.1.5 Zasady zapewnienia kontroli i ciągłości działania Systemu oraz okresowych przeglądów. 39**](#_Toc130289587)

[**4.2 Wymagania dotyczące Rozwoju. 43**](#_Toc130289588)

[**4.2.1 Wymagania Ogólne. 43**](#_Toc130289589)

[**4.2.2 Zasady realizacji Rozwoju. 43**](#_Toc130289590)

[**4.2.3 Etapy realizacji Rozwoju Systemu 44**](#_Toc130289591)

[**Etap I 45**](#_Toc130289592)

[**Etap II 46**](#_Toc130289593)

[**5. Wymagania wydajnościowe i niezawodnościowe. 49**](#_Toc130289594)

[**6. Wymagania w zakresie WCAG. 49**](#_Toc130289595)

[**7. Wymagania dla Dokumentacji. 49**](#_Toc130289596)

[**8. Wymagania dotyczące testów. 49**](#_Toc130289597)

[**9. Poziom świadczenia usług SLA. 49**](#_Toc130289598)

[**Załącznik nr 1: Wymagania wydajnościowe. 50**](#_Toc130289599)

[**Załącznik nr 2: Wymagania WCAG 2.1 51**](#_Toc130289600)

[**Załącznik nr 3: Wymagania dotyczące Dokumentacji. 67**](#_Toc130289644)

[**Załącznik nr 4: Wymagania dotyczące testów. 79**](#_Toc130289659)

[**Załącznik nr 5: Poziom świadczenia usług (SLA). 80**](#_Toc130289660)

# **Zastosowane definicje.**

|  |  |
| --- | --- |
| **Termin** | **Definicja** |
| **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 ich części, która jednak nie uniemożliwia Użytkownikom korzystania z podstawowych funkcji Systemu, polegająca w szczególności na ograniczeniu realizacji lub uciążliwości w realizacji co najmniej jednej z funkcji Systemu. |
| **Czas Obejścia** | Czas podawany w Godzinach Roboczych, liczony od momentu dokonania przez Zamawiającego Zgłoszenia Wady w Portalu Serwisowym do chwili dokonania Obejścia na Środowisku Produkcyjnym. |
| **Czas Naprawy** | Czas podawany w Godzinach Roboczych, liczony od momentu dokonania Zgłoszenia Wady przez Zamawiającego w Portalu Serwisowym do chwili udostępnienia Zamawiającemu Naprawy 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, wytwarzać i dostarczać zgodnie z Umową. |
| **Dzień Roboczy** | Każdy dzień tygodnia od poniedziałku do piątku, za wyjątkiem dni ustawowo wolnych od pracy w Rzeczpospolitej Polskiej. |
| **Godziny Robocze** | Godziny od 6:00 do 18:00 w Dni Robocze. |
| **Kierownik Projektu Strony (Zamawiającego/Wykonawcy)** | Osoba 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 (zarządzająca operacyjnie projektem). |
| **Kody Źródłowe Systemu (Kod Źródłowy)** | Zestaw plików zawierających nieskompilowany kod oprogramowania napisany w języku programowania, wynikającym z przyjętej technologii rozwiązania oraz w formie czytelnej dla człowieka, 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 oraz niestandardowe biblioteki wykorzystywane przy produkcji Systemu oraz opis działania bibliotek ogólnie dostępnych wykorzystywanych przy produkcji Systemu), jak również Dokumentacja niezbędna do użycia takiego kodu. Utrzymywane w systemie kontroli wersji GIT. |
| **Moduł Komunikacyjny(API)** | Interfejs realizujący wymianę danych pomiędzy Systemami e-PFRON2 i NEO. PFRON posiada niewyłączną, nieograniczoną czasowo i terytorialnie licencję na korzystanie i modyfikacje tego modułu. Wymiana danych odbywa się z wykorzystaniem WebService’ów wyspecyfikowanych w języku WSDL.  Obsługa Modułu Komunikacyjnego realizowana jest z wykorzystaniem Systemu NEO, który w procesie komunikacji między systemami NEO  i e-PFRON2 jest stroną aktywną – nawiązuje połączenie. |
| **Modyfikacje  i Rozwój (Rozwój)** | Wszelkie prace polegające na wprowadzaniu zmian w Systemie, realizowane przez Wykonawcę według zakresu opisanego w Zleceniu w sposób i na warunkach opisanych w Umowie wraz załącznikami. |
| **Naprawa** | Trwałe usunięcie Wady poprzez usunięcie przyczyn powstania Wady skutkujące przywróceniem pełnej sprawności Systemu oraz przywrócenia utraconych w wyniku wady danych, w tym również zakończenie innych działań naprawczych. |
| **Niedostępność Systemu** | Awaria Systemu, lub obniżenie parametrów wydajnościowych Systemu, w szczególności brak możliwości pracy w Systemie przez Użytkowników. |
| **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. |
| **Obszar Operatora**  **(Konsola Operatora)** | Zakres funkcjonalny wraz z zestawem obiektów Systemu, do których posiadają dostęp Pracownicy Zamawiającego, i za pomocą których realizowane są czynności w Systemie. |
| **Obszar Pracodawcy** | Zakres funkcjonalny wraz z zestawem obiektów Systemu, do którego posiada dostęp Pracodawca, i za pomocą których realizowane są czynności w Systemie. |
| **Odbiór** | Weryfikacja prawidłowości wykonania Pakietów Aktualizacji lub Produktów lub wszelkich prac zrealizowanych w ramach ATiK-u oraz Modyfikacji i Rozwoju. Odbiór pozytywny oznacza potwierdzenie prawidłowości wykonania prac/Produktów/Pakietów Aktualizacji.  Dowodem dokonania Odbioru jest podpisany przez Strony Umowy bez zastrzeżeń odpowiedni Protokół Odbioru. |
| **Okno Serwisowe** | Czas w ciągu dnia pomiędzy godziną 21:00 a 06:00 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. |
| **Oprogramowanie Standardowe / Oprogramowanie Obce** | Wszelkie oprogramowanie obce firm trzecich, stanowiące składnik Systemu, na którego użycie w procesie budowy, rozwoju, konfiguracji, instalacji lub użytkowania Systemu, w tym system operacyjny, systemy zarządzania bazą danych, serwery aplikacyjne.  Wykonawca powinien uzyskać zgodę Zamawiającego na użycie nowego Oprogramowania Standardowego/Oprogramowania Obcego przed przystąpieniem do wszelkich prac, których efektem może być modyfikacja lub rekonfiguracja Systemu. |
| **Oprogramowanie Systemowe i Narzędziowe** | Oprogramowanie wykorzystywane na potrzeby Systemu, którego producentem nie jest Wykonawca, konieczne do poprawnego działania Systemu, inne niż Oprogramowanie Zamawiającego, w tym biblioteki programistyczne, narzędzia do zarządzania logami, narzędzia zarządzania klastrami.  Wykonawca powinien **uzyskać zgodę** Zamawiającego **na użycie nowego** Oprogramowania Systemowego i Narzędziowego przed przystąpieniem do wszelkich prac, których efektem może być modyfikacja lub rekonfiguracja Systemu. |
| **Oprogramowanie Zamawiającego** | Oprogramowanie wykorzystywane na potrzeby Systemu, które zapewnia Zamawiający, z uwzględnieniem aktualizacji tego oprogramowania dokonanych w trakcie trwania Umowy. |
| **Pakiet Aktualizacji** | Przygotowane do instalacji uaktualnienie Systemu, służące usunięciu nieprawidłowości lub usprawnieniu pracy Systemu wytworzone w wyniku realizacji ATiK-u lub Modyfikacji i Rozwoju. |
| **Pomyłka Operatora** | Błąd w Systemie, powstały na skutek działania Operatora (Użytkownika wewnętrznego), który nie może być usunięty z poziomu interfejsu Użytkownika. |
| **Portal Serwisowy** | System informatyczny udostępniony przez Zamawiającego służący do ewidencji i obsługi Zgłoszeń, Wniosków i Zleceń zapewniający niezbędny poziom wymiany informacji pomiędzy Zamawiającym a Wykonawcą (Jira). |
| **Pracodawca** | Użytkownik zewnętrzny Systemu. Osoba fizyczna, osoba prawna lub jednostka organizacyjna nieposiadająca osobowości prawnej zobowiązana do wpłat lub zwolniona z wpłat na PFRON lub wystawiająca informacje o obniżeniu wpłaty. |
| **Pracownik Zamawiającego (Operator)** | Osoba fizyczna lub osoba prowadząca jednoosobową działalność gospodarczą, świadcząca osobiście pracę na rzecz Zamawiającego na podstawie umowy o pracę lub umowy cywilnoprawnej (umowy o dzieło lub umowy zlecenia). |
| **Produkt** | Wszelkie programy komputerowe, Dokumentacja i inne utwory, które powstają w toku wykonywania Umowy w wyniku prac Wykonawcy, a także materiały i informacje niepodlegające ochronie prawa autorskiego, stworzone lub dostarczone Zamawiającemu przez Wykonawcę w wykonaniu zobowiązań wynikających z Umowy. |
| **Pytanie** | Pytania dotyczącego działania Systemu w ramach świadczenia Usługi Asysty Technicznej i Konserwacji. |
| **Protokół Odbioru** | Dokument sporządzany przez Wykonawcę i podpisany przez Strony, potwierdzający prawidłowość i zakres wykonania konkretnych Przedmiotu Umowy. Wzory Protokołów Odbioru Usługi Asysty Technicznej i Konserwacji, Rozwoju stanowią Załącznik do Umowy. |
| **Przypadki Szczególne** | To takie, w których Użytkownik pomimo instrukcji Użytkownika Systemu i wsparcia konsultantów nie może skorzystać z dowolnej funkcji Systemu. |
| **Raport** | Dokument przedstawiony przez Wykonawcę i podpisany przez Strony, potwierdzający prawidłowość i zakres wykonania Usługi Asysty Technicznej i Konserwacji. Raport stanowi załącznik do Protokołu Odbioru ATiK-u. |
| **Repozytorium Architektury** | Część Repozytorium Projektowego służące do przechowywania modelu architektury. Forma, zawartość oraz zasady prowadzenia zostały opisane w załączniku nr 3 do OPZ. Format plików musi być możliwy do poprawnego odczytania w narzędziu Sparx Enterprise Architekt w wersji co najmniej 12. Narzędzie Sparx Enterprise Architekt jest standardowym oprogramowaniem za pomocą, którego Zamawiający zarządza repozytoriami eksploatowanych przez siebie systemów informatycznych. |
| **Repozytorium Projektowe/Repozytorium Projektu** | Środowisko służące do przechowywania Dokumentacji Systemu, Kodu Źródłowego Systemu oraz do dokumentowania bieżących prac Wykonawcy. Forma, zawartość oraz zasady prowadzenia zostały opisane Załączniku nr 3 do Opisu Przedmiotu Zamówienia. |
| **Repozytorium Wymagań** | Część Repozytorium Projektowego służące do przechowywania wymagań funkcjonalnych i poza funkcjonalnych. Forma, zawartość oraz zasady prowadzenia zostały opisane w Załączniku nr 3 do Opisu Przedmiotu Zamówienia. Format plików musi być możliwy do poprawnego odczytania w narzędziu Sparx Enterprise Architekt w wersji co najmniej 12. Narzędzie Sparx Enterprise Architekt jest standardowym oprogramowaniem za pomocą, którego Zamawiający zarządza repozytoriami eksploatowanych przez siebie systemów informatycznych. |
| **Repozytorium Kodu Źródłowego** | Część Repozytorium Projektowego służące do przechowywania i aktualizacji Kodu Źródłowego Systemu. Repozytorium Kodu Źródłowego prowadzone jest przez Wykonawcę w ramach przygotowanego przez Zamawiającego projektu w oprogramowaniu GitLab. Oprogramowanie GitLab funkcjonuje w infrastrukturze Zamawiającego i jest zarządzane przez pracowników Zamawiającego. Wymagania dotyczące organizacji i prowadzenia Repozytorium Kodu Źródłowego zawiera Załącznik nr 3. |
| **Roboczogodzina** | Jednostka miary pracochłonności wyrażająca normę ilościową pracy wykonanej przez jednego pracownika Wykonawcy w czasie jednej godziny zegarowej. |
| **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)** | Warunki poziomu świadczenia ATiK-u i Rozwoju, a także sposobu ich pomiaru, określone 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 e-PFRON2 (System)** | Oprogramowanie informatyczne wspierające obsługę procesów związanych ze składaniem i odbieraniem deklaracji, informacji oraz innych dokumentów poprzez teletransmisję danych w formie dokumentów elektronicznych przez Pracodawców zobowiązanych do wpłat lub zwolnionych z wpłat na PFRON, na podstawie ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych. W skład Systemu wchodzi kod w postaci wykonywalnej, kody źródłowe Systemu, 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 oraz Środowisko Testowe) oraz dokumentacja dotycząca wszelkich aspektów procesów budowy, rozwoju instalacji, odtwarzania, konfiguracji, użytkowania, rozwoju i utrzymania Systemu. |
| **System NEO** | Oprogramowanie informatyczne wspierające realizację zadań w obszarze wpłat obowiązkowych, w zakresie obsługi Pracodawców zobowiązanych do dokonywania obowiązkowych wpłat na PFRON lub zwolnionych z tych wpłat, wynikających z ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych. |
| **Środowisko Deweloperskie** | Infrastruktura sprzętowo – programowa Wykonawcy, która zapewnia Wykonawcy wykonywanie następujących 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 Deweloperskie jest utrzymywane przez Wykonawcę w ramach Usługi Asysty Technicznej i Konserwacji. |
| **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 Testowe** | 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 wewnętrznych Zamawiającego i Wykonawcy. Środowisko Testowe jest utrzymywane przez Wykonawcę w ramach Usługi Asysty technicznej i Konserwacji. |
| **Umowa** | Umowa zawarta między Zamawiającym, a Wykonawcą wraz ze wszystkimi aneksami i Załącznikami do Umowy. |
| **Usługi Asysty Technicznej i Konserwacji (ATiK)** | Wszelkie usługi i prace realizowane w celu zapewnienia ciągłości działania, w tym przywrócenia działania Systemu, w tym przywrócenie sprawności Systemu po wystąpieniu Wady i wysokiego poziomu bezpieczeństwa Systemu zgodnie z wymogami określonymi w Umowie wraz z załącznikami. |
| **Usterka** | Wada niebędąca Awarią ani Błędem, powodująca zakłócenie pracy Systemu lub poszczególnych jego części mogąca mieć wpływ na jego funkcjonalność, natomiast nieograniczająca zdolności operacyjnych Systemu. |
| **Użytkownik** | Osoba korzystająca z Systemu lub jego poszczególnych części. |
| **Wada** | Jakiekolwiek zaburzenie pracy Systemu objawiające się poprzez jego działanie w sposób odmienny od ustalonego, 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 wydajności i reaktywności, cech mających wpływ na bezpieczeństwo i ciągłość działania, oraz wszystkich innych cech funkcjonalnych i poza funkcjonalnych. Wady mogą mieć typ: Awarii, Błędu lub Usterki. |
| **Wniosek** | Przekazanie Wykonawcy zapotrzebowania w ramach Rozwoju poprzez utworzenie Zadania w Portalu Serwisowym. Otrzymanie Wniosku obliguje Wykonawcę do wykonania etapu I – „analiza i wycena”, czyli przygotowania analizy wraz z wyceną i przedstawienia jej wyników Zamawiającemu. |
| **Wykonawca** | Podmiot, który ubiega się o wykonanie zamówienia, złoży ofertę na jego wykonanie lub zawrze z Zamawiającym Umowę w sprawie wykonania zamówienia a następnie ją realizuje. |
| **Zadanie** | Zadanie w Portalu Serwisowym służące do obsługi Zgłoszeń, Wniosków i Zleceń, w tym: zamieszczania wyników prac przez Wykonawcę, akceptacji wyników prac przez Zamawiającego. |
| **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. |
| **Zlecenie** | Przekazanie Wykonawcy zapotrzebowania na wykonanie określonych Produktów lub innych prac, poprzez utworzenie Zadania w Portalu Serwisowym, w ramach Rozwoju na podstawie analizy wykonanej przez Wykonawcę. |
| **Zgłoszenie** | Przekazanie Wykonawcy zawiadomienia o Wadzie, zaleceń audytów, w tym audytu bezpieczeństwa i WCAG, złożenie Pytania, poprzez utworzenie Zadania w Portalu Serwisowym, w ramach świadczenia Usług Asysty Technicznej i Konserwacji oraz w okresie gwarancji. W okresie gwarancji Zgłoszenie będzie dotyczyć jedynie zawiadomienia o Wadzie. |

# **Ogólny opis zamówienia.**

### **Przedmiotem zamówienia jest świadczenie przez Wykonawcę na rzecz Zamawiającego:**

1. Usługi Asysty Technicznej i Konserwacji Systemu e-PFRON2 (dalej jako „ATiK”), w tym:
   * + - w ramach zamówienia gwarantowanego - 24 miesiące;
       - w ramach opcji – kolejne maksymalnie 24 miesiące;
2. Realizacja modyfikacji i rozwoju Systemu e-PFRON2 (dalej jako „Rozwój”), w tym:

* w ramach zamówienia gwarantowanego – 17.000 Roboczogodzin;
* w ramach Opcji – 13.000 Roboczogodzin.

### **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 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 Umowy.

### **Prawa własności intelektualnej.**

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

### **Licencje.**

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

### **Inne zobowiązania.**

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

### **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 będzie określać 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 e-PFRON2.**

# **3.1 Dziedzina Systemu.**

Państwowy Fundusz Rehabilitacji Osób Niepełnosprawnych (PFRON) eksploatuje System aplikacyjny e-PFRON2, który wspiera realizację procesu obsługi wpłat obowiązkowych na podstawie ustawy z dnia 27 sierpnia 1997 r. o rehabilitacji zawodowej i społecznej oraz zatrudnianiu osób niepełnosprawnych. (Dz.U. z 2023 r., poz. 100, ze zm.).

System e-PFRON2 obsługuje procesy związane ze składaniem i odbieraniem deklaracji, informacji oraz innych dokumentów poprzez teletransmisję danych w formie dokumentu elektronicznego przez Pracodawców zobowiązanych do wpłat lub zwolnionych z wpłat na PFRON.

System musi być dostępny dla Użytkowników przez 24 godziny na dobę, 7 dni w tygodniu, 365/366 dni w roku.

W chwili obecnej PFRON zapewnia hosting Systemu.

PFRON posiada prawa majątkowe do kodu źródłowego i dokumentacji Systemu, w tym do wykonywania zależnego prawa autorskiego.

W Systemie e-PFRON2 używany jest aplet Szafir SDK służący do podpisywania dokumentów podpisem kwalifikowanym. PFRON posiada licencje na korzystanie z tego oprogramowania oraz wsparcie techniczne dla apletu. System e-PFRON2 umożliwia potwierdzanie podpisu profilem zaufanym zarejestrowanym w Systemie.

W postępowaniu przetargowym zostanie udostępniony Kod Źródłowy oraz Dokumentacja Systemu.

### **Elementy i funkcje Systemu**

System e-PFRON2 składa się z następujących podsystemów (oddzielnych aplikacji):

* Obszar Pracodawcy,
* Konsola Operatora PFRON.

W obszarze Pracodawcy System umożliwia:

* Zgłaszanie nowych Pracodawców.
* Zgłaszanie administratorów upoważnianych przez Pracodawcę m.in. do składania dokumentów w Systemie.
* Zgłaszanie aktualizacji danych ewidencyjnych.
* Przyznawanie Użytkownikom wymaganych uprawnień oraz ich zmianę.
* Generowanie certyfikatów PFRON służących do podpisywania dokumentów, rejestrację podpisów kwalifikowanych oraz profilu zaufanego.
* Tworzenie deklaracji i informacji zgodnych z obowiązującymi w tym zakresie aktami prawnymi.
* Importowanie informacji o kwocie obniżenia wpłat na PFRON przygotowanych w innych programach niż e-PFRON2.
* Podpisywanie dokumentów kwalifikowanym podpisem elektronicznym, albo podpisem potwierdzanym przez profil zaufany, albo certyfikatem PFRON.
* Pobieranie Urzędowego Poświadczenia Przedłożenia (UPP) dla złożonych dokumentów.
* Składanie dokumentów w formie elektronicznej.
* Przeglądanie i drukowanie wersji roboczych i złożonych deklaracji i informacji.
* Pobieranie złożonych przez Pracodawców dokumentów w formacie html.
* Generowanie danych do przelewu w formacie Elixir, Videotel oraz wpłaty gotówkowej.
* Realizację płatności internetowych na rzecz PFRON z wykorzystaniem systemu KIR Paybynet.
* Otrzymywanie i wysyłanie wiadomości pomiędzy Pracodawcą, a PFRON.
* Przeglądanie i drukowanie raportów ze złożonych deklaracji i informacji oraz z dokonanych wpłat.
* Zapoznanie się z informacją o wysokości salda z tytułu wpłat obowiązkowych.
* Przeglądanie danych ewidencyjnych Pracodawcy.
* Przeglądanie informacji o rachunkach bankowych PFRON.
* Rejestrowanie i przeglądanie operacji wykonywanych w Systemie.
* Pobieranie plików z instrukcją obsługi Systemu e-PFRON2.
* Łączenie ze stroną internetową PFRON i serwisami społecznościowymi ([www.pfron.org.pl](http://www.pfron.org.pl), youtube, facebook, twitter).

System dla Użytkowników (Pracodawcy) dostępny jest pod adresem:

[**https://pracodawca.e-pfron.pl**](https://pracodawca.e-pfron.pl)

Korzystanie z Systemu e-PFRON2 przez Użytkowników jest możliwe pod warunkiem używania środowiska komputerowego spełniającego następujące wymagania:

Komputer z dostępem do sieci internetowej z zainstalowaną przeglądarką:

* Mozilla Firefox w wersji 72.x lub nowszej.
* Microsoft Edge w wersji 79.x lub nowszej.
* Google Chrome w wersji 79.x lub nowszej.
* Opera w wersji 66.x lub nowszej.
* Safari w wersji 13.x lub nowszej

W części dla PFRON - konsola Operatora umożliwia:

* Obsługę zgłoszeń nowych Pracodawców.
* Obsługę zgłoszeń administratorów Pracodawcy.
* Obsługę Użytkowników w zakresie przyznawania uprawnień, edycji oraz anonimizacji kont.
* Obsługę dokumentów elektronicznych złożonych przez Pracodawców.
* Obsługę dokumentów złożonych przez Pracodawców w formie dokumentów papierowych (wprowadzenie dokumentów przez Operatorów PFRON).
* Przeglądanie dokumentów roboczych wprowadzonych przez Użytkowników.
* Obsługę wiadomości odebranych i wysłanych pomiędzy Pracodawcą, a PFRON.
* Przesyłanie wiadomości do pracowników merytorycznych w PFRON oraz pobieranie i przesyłanie Użytkownikom odpowiedzi wiadomości otrzymanych z domeny pfron.org.pl drogą mailową.
* Obsługę certyfikatów.
* Przeglądanie listy zdarzeń.
* Przeglądanie płatności internetowych utworzonych z wykorzystaniem systemu KIR Paybynet.
* Obsługę Użytkowników wewnętrznych (Operatorów PFRON) w zakresie przyznawania uprawnień, edycji kont oraz ich anonimizacji.
* Dopisywania do kalendarza dni wolnych.
* Pobieranie plików z instrukcją operatora e-PFRON2.
* Łączenie ze stroną internetową i serwisami społecznościowymi PFRON ([www.pfron.org.pl](http://www.pfron.org.pl), youtube, facebook, twitter).

### **Architektura Systemu**

**Ogólny opis Systemu.**

**Warstwa podsystemów.**

Poniższy diagram prezentuje wszystkie podsystemy wchodzące w skład Systemu e-PFRON2 oraz ich podstawowe komponenty (na pierwszym poziomie zagnieżdżenia).



Rysunek 1 Ogólny schemat warstwy podsystemów

**Warstwa wdrożenia.**

Warstwa wdrożenia na ogólnym poziomie z wyszczególnieniem lokalizacji poszczególnych urządzeń wchodzących w skład Systemu e-PFRON2, a także stacji roboczych, została zaprezentowana na poniższym diagramie.

W kontekście lokalizacji poszczególnych elementów infrastruktury Systemu e-PFRON2 można wyróżnić następujące informacje:

* wszystkie urządzenia stanowiące infrastrukturę Systemu e-PFRON2 zlokalizowane są w serwerowni PFRON,
* stacje robocze: Operatorów PFRON znajdują się na terenie siedziby PFRON i uzyskują dostęp do zasobów Systemu za pośrednictwem sieci Intranet,
* stacje robocze Pracodawców mogą uzyskać dostęp do Systemu z dowolnego miejsca za pośrednictwem sieci Internet, przy czym mogą się łączyć bezpośrednio jedynie do strefy niezaufanej, w której umiejscowiony jest Load Balancer, filtrujący ruch z sieci oraz kierujący przepływem żądań do serwerów aplikacyjnych,
* stacje robocze administratorów uzyskują dostęp do Systemu wykorzystując połączenie tunelowe z serwerami aplikacyjnymi bądź serwerami bazodanowymi za pośrednictwem klienta VPN i przy pomocy zbioru protokołów IPSec.



Rysunek 2 Schemat lokalizacji urządzeń w warstwie wdrożenia

**Schemat rozmieszczenia komponentów.**

**Charakterystyka infrastruktury sprzętowej i programowej.**

Infrastrukturę sprzętową Systemu e-PFRON2 prezentuje poniższy diagram rozmieszczenia (*deployment diagram*).



Rysunek 3 e-PFRON2 - infrastruktura sprzętowa

**Load Balancer.**

Load Balancer kontroluje ruch generowany przez Pracodawców od strony segmentu publicznego sieci. Urządzenie realizuje następujące zadania:

* równoważenie ruchu http kierowanego do serwerów aplikacji,
* zapewnienie ciągłości pracy poprzez kierowanie ruchu do pozostałych serwerów w przypadku awarii jednego z nich,
* deszyfracja i szyfrowanie danych (https do http i odwrotnie),
* odciążenie serwerów aplikacji poprzez zastosowanie filtracji ruchu przychodzącego oraz mechanizmów cache’owania elementów stałych.

**Serwer aplikacji – konsola operatora.**

Każdy z serwerów aplikacyjnych modułu pracodawcy posiada identyczną infrastrukturę programową. Serwery aplikacyjne modułu pracodawcy stanowią elementy klastra serwerów aplikacji. Ruch generowany przez pracodawców przechodzi przez Load Balancer



Rysunek 4 Oprogramowanie - serwer aplikacji konsoli operatora

Na serwerach aplikacyjnych konsoli operatora zainstalowane zostało następujące oprogramowanie:

* system operacyjny RHEL 8.5,
* Warstwa backend aplikacji:
  + serwer webowy Nginx w wersji 1.21
  + środowisko PHP 8.1 wraz z frameworkiem programistycznym Symfony 5.4
  + oprogramowanie aplikacyjne:
    - własne: Konsola, Moduł komunikacyjny NEO, Moduł generacji dokumentów PDF i HTML
  + wykorzystanie Dockera/docker-compose do szybkiego rozstawiania kolejnych instancji backendu w razie konieczności a także zapewnienia spójności środowiska developerskiego, testowego i produkcyjnego
* Warstwa frontend aplikacji:
  + Angular w wersji 12.2
  + Bootstrap w wersji 4.5
* usługa OpenSSH.

**Serwer aplikacji Front – aplikacja pracodawcy (Serwer aplikacji modułu pracodawcy)**

Każdy z serwerów aplikacyjnych modułu pracodawcy posiada identyczną infrastrukturę programową. Serwery aplikacyjne modułu pracodawcy stanowią elementy klastra serwerów aplikacji. Ruch generowany przez pracodawców przechodzi przez Load Balancer



Rysunek 5 Oprogramowanie - serwer aplikacji modułu pracodawcy

Na serwerach aplikacyjnych modułu pracodawcy zainstalowane zostało następujące oprogramowanie:

* system operacyjny RHEL 8.0,
* Warstwa backend aplikacji:
  + serwer webowy Nginx w wersji 1.17.19
  + środowisko PHP 7.4 wraz z frameworkiem programistycznym Symfony 4.4
  + oprogramowanie aplikacyjne:
    - własne: Front, Moduł integracji z Węzłem Krajowym, Moduł generacji dokumentów PDF i HTML
    - obce: Szafir
  + wykorzystanie Dockera/docker-compose do szybkiego rozstawiania kolejnych instancji backendu w razie konieczności a także zapewnienia spójności środowiska developerskiego, testowego i produkcyjnego
* Warstwa frontend aplikacji:
  + Angular w wersji 9.0
  + Bootstrap w wersji 4.4
* usługa OpenSSH.

**Serwer aplikacji REDIS**

Serwer służący jako zaawansowany system cache w pamięci serwera.



Rysunek 6 Oprogramowanie – serwer aplikacji REDIS

Na serwerze aplikacji REDIS zainstalowane zostało następujące oprogramowanie:

* system operacyjny RHEL 8.0,
* usługa Redis jako zaawansowany system cache w pamięci serwera,
* usługa OpenSSH.

**Serwer aplikacji UPO**

Serwer służący do komunikacji z urządzeniem HSM i generowania Urzędowych Poświadczeń Odbioru.



Rysunek 7 Oprogramowanie - serwer aplikacji UPO

Na serwerze aplikacyjnych UPO zainstalowane zostało następujące oprogramowanie:

* system operacyjny RHEL 7.5,
* środowisko OpenJDK-7-JRE,
* oprogramowanie aplikacyjne:
  + własne: moduł UPO (w technologii Java),
  + obce: Szafir, oprogramowania klienckie modułu HSM
* usługa OpenSSH.

**Serwer bazy danych.**

Każdy z serwerów baz danych posiada identyczną infrastrukturę programową. Serwery bazodanowe zostały połączone na zasadzie master-slave z replikacją danych metodą „streaming replication”, co pozwala utrzymywać serwer w trybie slave w gotowości wraz z uaktualnionymi bazami danych.

Infrastruktura programowa serwera bazy danych przedstawiona została na poniższym diagramie rozmieszczenia (*deployment diagram*).



Rysunek 8 Oprogramowanie – serwer bazy danych

Na serwerach baz danych zainstalowane zostało następujące oprogramowanie:

* system operacyjny Debian 7 64-bit,
* silnik bazy danych PostgreSQL w wersji 9.1 wraz z utworzoną bazą pracodawców i pracowników PFRON,
* usługa OpenSSH.

**Serwer RabbitMQ.**

Każdy z serwerów RabbitMQ posiada identyczną infrastrukturę programową. Infrastruktura programowa serwera RabbitMQ przedstawiona została na poniższym diagramie rozmieszczenia (*deployment diagram*).



Rysunek 9 Oprogramowanie – serwer bazy danych

Na serwerach RabbitMQ zainstalowane zostało następujące oprogramowanie:

* system operacyjny RHEL 8.5,
* usługa RabbitMQ 3.10.7

usługa OpenSSH.

**Zestawienie infrastruktury sprzętowej**

|  |  |  |
| --- | --- | --- |
| **Węzeł** | **Typ serwera** | **Parametry techniczne maszyny wirtualnej** |
| **Środowisko Produkcyjne** | 1 x Serwer bazodanowy | Liczba rdzeni przypisana do maszyny: 32 RAM: 16 GB Typ wirtualizacji: VMware vSphere Replikacja bazodanowa |
| 1 x Serwer bazodanowy | Liczba rdzeni przypisana do maszyny: 4 RAM: 16 GB Typ wirtualizacji: VMware vSphere Replikacja bazodanowa |
| 2 x Serwer aplikacyjny konsoli operatora | Liczba rdzeni przypisana do maszyny: 8 RAM: 16 GB Typ wirtualizacji: VMware vSphere |
| 3 x Serwer rabbitmq | Liczba rdzeni przypisana do maszyny: 8 RAM: 16 GB  Typ wirtualizacji: VMware vSphere |
| 1 x Serwer UPO | Liczba rdzeni przypisana do maszyny: 1 RAM: 2 GB Typ wirtualizacji: VMware vSphere |
| 2 x Serwer aplikacji Front – aplikacja pracodawcy | Liczba rdzeni przypisana do maszyny: 8  RAM: 16 GB  Typ wirtualizacji: VMware vSphere |
| 1 x Serwer REDIS | Liczba rdzeni przypisana do maszyny: 2  RAM: 8 GB  Typ wirtualizacji: VMware vSphere |
| **Środowisko Testowe** | 2 x Serwer bazodanowy | Liczba rdzeni przypisana do maszyny: 4 RAM: 8 GB Typ wirtualizacji: VMware vSphere Replikacja bazodanowa |
| 1 x Serwer aplikacyjny kosoli operatora | Liczba rdzeni przypisana do maszyny: 2 RAM: 8 GB Typ wirtualizacji: VMware vSphere |
| 3x Serwer rabbitmq | Liczba rdzeni przypisana do maszyny: 4 RAM: 4 GB Typ wirtualizacji: VMware vSphere |
| 1 x Serwer aplikacji Front – aplikacja pracodawcy | Liczba rdzeni przypisana do maszyny: 2  RAM: 8 GB  Typ wirtualizacji: VMware vSphere |
| 1 x Serwer REDIS | Liczba rdzeni przypisana do maszyny: 2  RAM: 8 GB  Typ wirtualizacji: VMware vSphere |

**Wykorzystywane wersje systemów linux:**

Debian 7, RedHat 7.5, Powered off CentOS, CentOS 6.9 i RedHat 8.0, RedHat 8.5, RedHat 8.6

**Przybliżona ilość linii kodu e-PFRON2 – część Operatora:**

|  |  |
| --- | --- |
| Kod napisany we framework’u Symfony (PHP) - backend | 92 010 |
| Kod napisany we framework’u Symfony (PHP) - API NEO | 2 677 |
| Kod napisany w środowisku Angular (HTML, Javascript, Typescript, CSS, json) - frontend | 214 522 |
| Kod napisany w języku JavaScript (moduł generacji dokumentów PDF i HTML) | 86 |

W skład części Operatora wchodzą: aplikacja REST API (PHP, Symfony), aplikacja frontend – (Angular), API NEO (PHP, Symfony) oraz moduł generacji dokumentów PDF i HTML (JavaScript).

**Przybliżona ilość linii kodu e-PFRON2 – część Pracodawcy:**

|  |  |
| --- | --- |
| Kod napisany we framework’u Symfony (PHP) - backend | 116 585 |
| Kod napisany w środowisku Angular (HTML, Javascript, Typescript, CSS, json) - frontend | 252 595 |
| Kod napisany w języku Java (moduł obsługi Węzła Krajowego) | 1 784 |
| Kod napisany w języku JavaScript (moduł generacji dokumentów PDF i HTML) | 86 |

W skład części Pracodawcy wchodzą: aplikacja REST API (PHP, Symfony), aplikacja frontend – (Angular), moduł integracji z Węzłem Krajowym (aplikacja Java) oraz moduł generacji dokumentów PDF i HTML (JavaScript).

**Dane statystyczne (wg stanu na 17.03.2023 r.):**

|  |  |
| --- | --- |
| Liczba wszystkich użytkowników zarejestrowanych e-PFRON2 | 329 754 |
| Liczba wszystkich dokumentów złożonych w e-PFRON2 | 12 661 775 |
| Liczba zgłoszeń administratorów zgłoszonych w e-PFRON2 | 171 216 |
| Liczba wszystkich zarejestrowanych metod uwierzytelniania | 398 602 |
| Liczba wiadomości wysłanych i odebranych od pracodawców | 563 648 |
| Liczba zapisów w rejestrze zdarzeń w e-PFRON2 | 101 779 840 |
| Liczba operatorów PFRON | 344 |
| Liczba dokumentów złożonych w lipcu 2022 | 118 720 |
| Liczba pracodawców, którzy złożyli dokumenty w lipcu 2022 | 56 425 |

# **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. Zamawiający dopuszcza 24 Roboczogodzin Niedostępności Systemu w miesiącu.
2. Współpracy z Wykonawcą świadczącym usługi hostingu w przypadkach dotyczących infrastruktury, na której posadowiony jest System.
3. Utrzymania i administracji Sytemu w tym Oprogramowania Systemowego i Narzędziowego oraz Oprogramowania Standardowego/Obcego.
4. Utrzymania wartości parametrów związanych z Usługą Asysty Technicznej i Konserwacji na warunkach opisanych w Załączniku nr 5 do Opisu Przedmiotu Zamówienia („Poziom świadczenia usług SLA”).
5. 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.
6. 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.
7. Przyjmowania Zgłoszeń i Naprawy Wad Systemu.
8. Usuwania Wad Systemu wszystkich kategorii zgodnie z wymaganiami opisanymi w pkt 4.1.2 Opisu Przedmiotu Zamówienia.
9. Wydawania rekomendacji dotyczących przeprowadzania zmian, aktualizacji i modernizacji Systemu.
10. 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.
11. Zapewnienia stałej opieki co najmniej jednego konsultanta do wsparcia przy rozwiązywaniu bieżących problemów związanych z funkcjonowaniem Systemu.
12. 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.
13. Zrealizowania raz na kwartał przeglądu Kodów Źródłowych i Dokumentacji Systemu zgodnie z wymaganiami opisanymi w załączniku nr 3 do Opisu Przedmiotu Zamówienia.
14. 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.
15. 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.
16. Instalowania na Środowisku Produkcyjnym, w czasie Okna Serwisowego, o ile Strony nie uzgodnią inaczej, Pakietów Aktualizacyjnych 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.
17. 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.
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-16, ATK-17.
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-16, ATK-17.
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, ATK -17.
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-14, ATK-16, ATK-17.

### **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 odnośnie 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 Opisu Przedmiotu Zamówienia („Poziom świadczenia usług SLA”), 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 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 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. Po zamknięciu Zgłoszenia Wykonawca dostarcza zaktualizowaną Dokumentacje Systemu oraz zaktualizowaną wersje Kodów Źródłowych zgodnie z zasadami opisanymi w załączniku nr 3 Opisu Przedmiotu Zamówienia.
30. Wykonawca zobowiązany jest sporządzać comiesięczny Raport, którego szablon stanowi załącznik do Protokołu Odbioru ATiK-u (załącznik nr 3 do Umowy). 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 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 Pytań w Portalu Serwisowym.
6. Jeżeli Wykonawca nie będzie w stanie udzielić odpowiedzi w czasie określonym w załączniku nr 5 do Opisu Przedmiotu Zamówienia („Poziom świadczenia usług SLA”), jest zobowiązany powiadomić o tym fakcie Zamawiającego, z którym zostanie ustalony nowy termin udzielenia odpowiedzi.
7. 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 Środowisku Testowym;
8. instalację aktualizacji na Środowisku Produkcyjnym;
9. testy Systemu na Środowisku Produkcyjnym, Środowisku Testowym;
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 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 Produkcyjnym i Testowym 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 przygotowaniem 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 będą zarówno serwery aplikacyjne jak i serwery bazodanowe wraz z danymi
10. Wykonawca w terminie 10 Dni Roboczych od dnia zawarcia Umowy zweryfikuje konfigurację i działanie obecnie wykorzystywanego u Zamawiającego narzędzia do monitorowania zasobów (Zabbix) oraz w przypadku konieczności wprowadzenia zmian w konfiguracji powyższego narzędzia wykona jego rekonfigurację. Wykonawca zobowiązany jest do przedstawienia raportu z wykonanych prac.
11. Wykonawca określi wszystkie parametry konfiguracyjne polityk archiwizacji danych Oprogramowania objętego Usługą Utrzymania 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 15 Dni Roboczych od dnia zawarcia Umowy.
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 15 Dni Roboczych 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 10 Dni Roboczych 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 15 Dni Roboczych 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 10 Dni Roboczych 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 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 do Opisu Przedmiotu Zamówienia („Poziom świadczenia usług SLA”).

### **Zasady realizacji Rozwoju.**

1. Wykonawca nie może odmówić realizacji złożonego Wniosku i Zlecenia, poza przypadkami, gdy ich 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 Rozwoju dysponował zespołem projektowo-programowym, który może wykonać prace o zakresie nie mniejszym niż 1 000 Roboczogodzin w trakcie jednego miesiąca.
4. W przypadku, gdy do realizacji prac w ramach 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 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 Rozwoju.
6. W przypadku, gdy realizacja prac spowoduje pojawienie się Wady w Systemie, Wykonawca zobowiązany jest do wstrzymania prac w ramach Rozwoju, do czasu skutecznego usunięcia Wady.
7. Wykonawca zobowiązany jest do zapewnienia zgodności Produktów przekazywanych w ramach realizacji Rozwoju z zaleceniami WCAG zawartymi w załączniku nr 2 do Opisu Przedmiotu Zamówienia.
8. Wszystkie Wnioski i Zlecenia oraz inne materiały z realizacji Rozwoju (w tym z testów) muszą być przez Strony rejestrowane i prezentowane w Portalu Serwisowym.
9. W przypadku zaistnienia różnicy zdań między Stronami dotyczącej wiarygodności przedstawionej wyceny i szacunków czasochłonności, Strony zobowiązują się do podporządkowania się opinii niezależnego od Stron rzeczoznawcy i rozliczenia prac według podanych przez niego wskazań. Rzeczoznawca zostanie wybrany przez Strony metodą zapewniającą bezstronność, a także będzie osobą posiadającą potwierdzoną certyfikatami wiedzę z zakresu wymiarowania przedsięwzięć informatycznych. Rzeczoznawca 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 rzeczoznawca się przychyli. Jeśli rzeczoznawca nie przychyli się do racji żadnej ze stron, obie strony ponoszą koszt sporządzenia opinii po połowie. Skorzystanie przez Strony z opinii rzeczoznawcy nie wstrzymuje realizacji zleconej modyfikacji.

### **Etapy realizacji Rozwoju Systemu**

Procedura realizacji Rozwoju Systemu składa się z etapów:

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

#### **Etap I**

1. Zamawiający tworzy Zadanie w Portalu Serwisowym stanowiące Wniosek o dokonanie analizy, zawierający: podstawowe wymagania funkcjonalne i poza funkcjonalne oraz inne informacje mogące mieć wpływ na realizację Zadania.
2. Wykonawca w terminie do 10 Dni Roboczych od przekazania utworzonego Zadania, przedstawi Zamawiającemu w Portalu Serwisowym wynik analizy. Przy czym Strony mogą ustalić inny termin dostarczenia wyniku analizy przez Wykonawcę. Ostateczna decyzja w tym zakresie należy do Zamawiającego.
3. Analiza powinna zawierać w szczególności (w zależności od przedmiotu Zadania):
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 UMR-15 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. 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.
13. Zamawiający zobowiązany jest do przekazania Wykonawcy informacji czy akceptuje, czy odrzuca przedstawiony przez Wykonawcę wynik Etapu I.
14. Jeżeli Strony dokonają stosownych ustaleń przed rozpoczęciem realizacji Zadania, wycena Zadania (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 Zadania. Ostateczna akceptacja wyceny czasochłonności prac należy do Zamawiającego.
15. Zamawiający ma prawo zrezygnować z realizacji Etapu II. Realizacja Etapu I nie powoduje skutków finansowych dla Zamawiającego.
16. Końcowa wersja projektu realizacji Zadania zostaje uzgodniona w trybie roboczym. Wycena realizacji prac w ramach Rozwoju uzgodniona w końcowej wersji projektu dotyczącej realizacji Etapu II będzie stanowić podstawę wyliczenia wynagrodzenia za wykonanie Zadania.
17. W przypadku akceptacji projektu realizacji Zadania Zamawiający może złożyć Zlecenie o realizację Etapu II.

#### **Etap II**

1. Wykonawca przystępuje do realizacji Etapu II po otrzymaniu od Zamawiającego Zlecenia Etapu II.
2. 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.
3. Po przeprowadzeniu testów wewnętrznych Wykonawca zgłasza Zamawiającemu gotowość do testów akceptacyjnych.
4. Poinformowanie Zamawiającego o gotowości do zainstalowania przez Wykonawcę Pakietu Aktualizacji na Środowisku Testowym uznaje się za zgłoszenie przez Wykonawcę gotowości do Odbioru realizowanego Zlecenia.
5. Po zgłoszeniu gotowości Odbioru Zamawiający przystąpi niezwłocznie do weryfikacji Pakietu Aktualizacji.
6. Zamawiający ma prawo do weryfikacji należytego wykonania Zlecenia dowolną metodą. Zamawiający ma prawo przeprowadzić testy za pomocą samodzielnie zdefiniowanych scenariuszy testowych.
7. Wykonawca ma obowiązek dostarczyć Zamawiającemu dokumenty, w tym raporty, scenariusze testowe wymagane w Załączniku nr 4 do OPZ, najpóźniej w momencie zgłoszenia Zamawiającemu przez Wykonawcę gotowości do Odbioru (patrz pkt UMR-26).
8. Po weryfikacji Pakietu Aktualizacji Zamawiający niezwłocznie potwierdzi wykonanie (Odbiór pozytywny) lub stwierdzi niewykonanie Zlecenia (Odbiór negatywny). W przypadku Odbioru negatywnego Produkt Zlecenia podlega dalszym pracom, do czasu jego należytego wykonania (patrz procedura odbiorowa określona w Umowie). Kara umowna z tytułu nienależytego wykonania przedmiotu Zlecenia zostanie naliczona Wykonawcy po przeprowadzeniu jednej iteracji odbiorowej.
9. Jeżeli Wykonawca nie wykona Zlecenia w terminie wskazanym w Zleceniu, Zamawiający może:
10. wydłużyć termin wykonania Zlecenia na pisemną prośbę Wykonawcy zawierającą uzasadnienie i zmianę terminu Zlecenia;
11. obciążyć Wykonawcę karą umowną na zasadach opisanych w Umowie.
12. Po zakończeniu testów akceptacyjnych, Wykonawca ma obowiązek instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym w terminie 3 Dni Roboczych, chyba, że Zamawiający postanowi inaczej.
13. 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 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.
14. 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.
15. Instalacja Pakietu Aktualizacji na Środowisku Produkcyjnym realizowana będzie w czasie Okna Serwisowego, o ile Strony nie uzgodnią inaczej.
16. Zamawiający zastrzega sobie prawo rezygnacji z instalacji Pakietu Aktualizacji na Środowisku Produkcyjnym.
17. Warunkiem zakończenia realizacji Zlecenia jest:
18. Pozytywny Odbiór Zlecenia;
19. zainstalowanie przez Wykonawcę Pakietu Aktualizacji na Środowisku Produkcyjnym;
20. dostarczenie Zamawiającemu przez Wykonawcę zaktualizowanej Dokumentacji Systemu (patrz pkt UMR-33);
21. przeprowadzenie przez Wykonawcę warsztatów, o których mowa w pkt UMR-34 (o ile Zlecenie obejmuje warsztaty).
22. Zakończenie realizacji Zlecenia potwierdzane jest poprzez zamknięcie Zadania w Portalu Serwisowym przez upoważnionego pracownika Zamawiającego wskazanego w Umowie.
23. Zamknięcie Zadania w Portalu Serwisowym oznacza możliwość ujęcia Zlecenia w Protokole Odbioru Rozwoju, którego wzór zawiera Załącznik do Umowy.
24. Podpisanie Protokołu Odbioru, o którym mowa w pkt UMR-39 powyżej, przez Zamawiającego bez zastrzeżeń jest podstawą do wystawienia przez Wykonawcę faktury.
25. 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.
26. Protokół Odbioru Zleceń wykonanych w ramach Rozwoju zawierać będzie informację o liczbie Roboczogodzin, w ramach których Zlecenia zostały wykonane. Liczba Roboczogodzin wskazana w zaakceptowanym przez Zamawiającego Protokole Odbioru będzie podstawą do rozliczenia limitu Roboczogodzin na Rozwoju określonego w niniejszej Umowie.
27. W szczególnych przypadkach za zgodą stron realizacje zlecenia może się odbyć w metodologii zwinnej na przykład Scrum lub Agile.

# **Wymagania wydajnościowe i niezawodnościowe.**

Szczegółowe wymagania zawiera Załącznik nr 1 do Opisu Przedmiotu Zamówienia.

# **Wymagania w zakresie WCAG.**

Szczegółowe wymagania zawiera Załącznik nr 2 do Opisu Przedmiotu Zamówienia.

# **Wymagania dla Dokumentacji.**

Szczegółowe wymagania zawiera Załącznik nr 3 do Opisu Przedmiotu Zamówienia.

# **Wymagania dotyczące testów.**

Szczegółowe wymagania zawiera Załącznik nr 4 do Opisu Przedmiotu Zamówienia.

# **Poziom świadczenia usług SLA.**

Szczegółowy opis zawiera Załącznik nr 5 do Opisu Przedmiotu Zamówienia.

## **Załącznik nr 1: Wymagania wydajnościowe.**

1. Wykonawca zapewni ciągłe funkcjonowanie Systemu przez 24 godziny na dobę, 7 dni w tygodniu, 365/366 dni w roku, przy założeniu, że w jednym czasie z Systemu korzysta 20 000 Użytkowników zewnętrznych i 100 Użytkowników wewnętrznych w ciągu jednej doby, z ustalonym punktem przesilenia w godzinach 9:00 – 15:00 oraz założeniu utrzymania minimum 2 500 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: Wymagania WCAG 2.1**

Serwis/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. 2023 poz. 82 ze zm.),
* 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. 2023 poz. 82 ze zm.).

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](https://www.w3.org/TR/WCAG20-TECHS/ARIA19.htm%20)

### 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: Wymagania dotyczące Dokumentacji.**

1. Wymagania Ogólne

1. W terminie 10 Dni Roboczych od dnia zawarcia Umowy, 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 zaakceptowania 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:
7. Dokumentacji Użytkownika,
8. Dokumentacji technicznej,
9. Dokumentacji administratora technicznego,
10. Dokumentacja Kodów Źródłowych.

Każda z części dokumentacji składa się z szeregu tomów opisujących poszczególne aspekty działania Systemu. I tak poszczególne części dokumentacji składają się z następujących tomów:

1. Dokumentacja Użytkownika:

* Instrukcja obsługi Systemu dla Operatora,
* Podręcznik obsługi Systemu dla Użytkownika (Pracodawcy).

1. Dokumentacja techniczna:

* Opis obszaru i procesów biznesowych,
* Opis systemu,
* Projekt techniczny - podsystem Front,
* Projekt techniczny - podsystem Konsola,
* Projekt techniczny - pozostałe podsystemy,
* Projekt techniczny infrastruktury,
* Formaty przelewów elektronicznych,
* Oprogramowanie zewnętrzne,
* Projekt koncepcyjny,
* Walidacje,
* Formularze,
* Macierz uprawnień.

1. Dokumentacja administratora technicznego:

* Instrukcja budowy systemu,
* Opis czynności administracyjnych,
* Opis parametrów konfiguracji,
* Opis systemu,
* Procedura instalacji,
* Procedura instalacji – Dokumentacja PostrgreSQL,
* Wykaz licencji – Warunki licencyjne SZAFIR,
* Wykaz licencji.

1. Dokumentacja kodu źródłowego – e-PFRON2 - Repozytorium Architektury (EA)
2. Repozytorium Architektury będzie m.in. służyć jako źródło do generowania części lub całości Dokumentacji Systemu omawianej w niniejszym Załączniku. Repozytorium Architektury musi być prowadzone w narzędziu Sparx Enterprise Architect.

## **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.

### 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 – 14 (OPZ).

Weryfikacja Kodu Źródłowego będzie prowadzona dla:

* 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.
* Dostaw Kodów Źródłowych realizowanych w ramach umów usług utrzymania i rozwoju, zgodnie z określonymi w umowie terminami.
* 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.

## 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: 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. Po zakończeniu testów Wykonawca zobowiązany jest do przedstawienia raportu z testów.

WT4 - Wykonawca zobowiązany jest do przeprowadzenia testów bezpieczeństwa na Środowisku Testowym. Po zakończeniu testów Wykonawca zobowiązany jest do przedstawienia raportu z testów.

## **Załącznik nr 5: 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** | 24 godziny na dobę, 7 dni w tygodniu 365/366 dni w roku  Okno Serwisowe w godzinach: 21.00 – 6.00. |
| **Czasy realizacji** | |  |  |  |  | | --- | --- | --- | --- | | **Lp.** | **Nazwa Wady** | **Czas Naprawy[[1]](#footnote-2)** | **Czas Obejścia[[2]](#footnote-3)** | | 1. | Awaria – w okresie 16-22 dnia każdego miesiąca | ……. godzin zegarowych | …… godzin zegarowych | | 2. | Awaria – poza okresem 16-22 dnia każdego miesiąca | ……….. godzin zegarowych | ……….. godzin zegarowych | | 3. | Błąd – w okresie 16-22 dnia każdego miesiąca | ……….. godzin zegarowych | …………. godzin zegarowych | | 4. | Błąd – poza okresem 16-22 dnia każdego miesiąca | …………. godzin zegarowych | ……………. godziny zegarowe | | 5. | Usterka – w okresie 16-22 dnia każdego miesiąca | 48 godzin zegarowych | Nie dotyczy | | 6. | Usterka – poza okresem 16-22 dnia każdego miesiąca | 72 godziny zegarowe | Nie dotyczy | | 7. | Odpowiedź na pytania w okresie 16-22 każdego miesiąca | 3 godziny zegarowe | Nie dotyczy | | 8. | Odpowiedź na pytania poza okresem 16-22 każdego miesiąca | 1 Dzień Roboczy | Nie dotyczy | | 9. | Pomyłka Operatora | 12 Godzin Roboczych | Nie dotyczy |   Definicje znajdują się w słowniku w pkt 1 Opisu Przedmiotu Zamówienia. |
| **Poziom dostępności usługi** | **RPDS** – **rzeczywisty poziom dostępności Systemu**  **RPDS ≥ 96,67%**  RPDS obliczany jest wg wzoru:  **(TD - ∑ TN) / TD \*100[%]**  Gdzie:  TD – uzgodniony czas dostępności usługi, wynikający z kalendarza dostępności usługi, po odjęciu uzgodnionych okien serwisowych [w godzinach].  TN – czas trwania Niedostępności usługi [w godzinach].  **Wyliczenie minimalnego progu RPDS:**  TN – czas trwania Niedostępności usługi - przyjmujemy dopuszczalną Niedostępność Systemu na 24 godziny w ciągu miesiąca.  ∑ TN = 24 godziny.  TD = 30 dni \* 24 godziny = 720 godzin (zgodnie z podanym kalendarzem dostępności usługi).  RPDS = (720 godzin - 24 godziny) / 720 godzin \* 100 =  **96,67%** |
|  |
| **Terminowość** | **PDTN** – **poziom dotrzymania terminów naprawy lub odpowiedzi**  **PDTN ≥ 96,59%**  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 – w okresie 16-22 dnia każdego miesiąca | **5** | **100,00** | | Awaria – poza okresem 16-22 dnia każdego miesiąca | **5** | **100,00** | | Błąd – w okresie 16-22 dnia każdego miesiąca | **3** | **96,00** | | Błąd – poza okresem 16-22 dnia każdego miesiąca | **3** | **96,00** | | Usterka – w okresie 16-22 dnia każdego miesiąca | **2** | **93,00** | | Usterka – poza okresem 16-22 dnia każdego miesiąca | **2** | **93,00** | | Odpowiedź na pytania w okresie 16-22 każdego miesiąca | **3** | **96,00** | | Odpowiedź na pytania poza okresem 16-22 każdego miesiąca | **2** | **93,00** | | Pomyłka Operatora | **2** | **93,00** |   **Wyliczenie minimalnego progu PDTN:**  PDTN = (5 \* 100 + 5 \* 100 + 3 \* 96 + 3 \* 96 + 2 \* 93 + 2 \* 93 + 3 \* 96 + 2 \* 93 + 2 \* 93) / (5 + 5 + 3 + 3 + 2 + 2 + 3 +2 + 2) = **96,59%** | |

|  |  |
| --- | --- |
| **Lista i częstotliwość raportów** | * Rzeczywisty poziom dostępności Systemu (RPDS) (kwartalnie -sporządzane przez Wykonawcę i przekazywane Zamawiającemu do 15 dni po zakończeniu kwartału). * Poziom dotrzymania terminów Naprawy (PDTN) (kwartalnie -sporządzane przez Wykonawcę i przekazywane Zamawiającemu do 15 dni po zakończeniu kwartału). |

1. **Modyfikacje i Rozwój Systemu e-PFRON2**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Czasy realizacji** | |  |  |  | | --- | --- | --- | | **Lp.** | **Nazwa** | **Czas realizacji usługi** | | 1. | Etap I | 10 Dni Roboczych | | 2. | Etap II | Ustalany indywidualnie | |
| **Wskaźniki:**  **WJCR oraz WJLB** | **WJCR – wskaźnik jakościowy czasu realizacji**  WJCR jest obliczany wg wzoru:  **WJCR =**  Gdzie:  TZi – rzeczywisty czas realizacji etapu (liczony: dla etap I - do dnia zgłoszenia przez Wykonawcę analizy do akceptacji Zamawiającego; dla etapu II - do dnia zgłoszenia przez Wykonawcę Produktu/prac do Odbioru Zamawiającemu. W przypadku, gdy prace realizowane przez Wykonawcę zostaną wykonane wcześniej niż w terminie ustalonym z Zamawiającym lub wymaganym w ramach danego etapu, to dla wyliczenia wskaźnika, Zamawiający przyjmie maksymalny, ustalony lub wymagany, czas realizacji danego Etapu. Powyższe ma jedynie zastosowanie do wyliczenia wskaźnika WJCR);  TWi – ustalony lub wymagany czas realizacji etapu;  i – każdy zakończony etap (dla wszystkich usług modyfikacji i rozwoju wykonanych lub rozpoczętych) w okresie, dla którego liczony jest wskaźnik;  n – liczba etapów;  WJCR będzie:  < 1 jeśli Wykonawca wykonuje zadania przed terminem,  = 1 jeśli wykonuje zadania w terminie,  > 1 jeśli się spóźnia z wykonaniem zadania.  **WJLB – wskaźnik jakościowy liczby błędów** (wykrytych podczas odbioru usług)  WJLB jest obliczany wg wzoru:  **WJLB =**  Gdzie:  LB – liczba błędów wykrytych podczas odbioru;  Rb – liczba Roboczogodzin związana z harmonogramem zlecenia usługi;  50 – oznacza, że badanie wskaźnika odbywa się na poziomie 50 Roboczogodzin.  WJLB będzie:  < 1 nie wystąpiły błędy podczas odbioru usługi (na poziomie 50 Roboczogodzin),  = 1 błąd - dopuszczalny (na poziomie 50 Roboczogodzin),  > 1 zbyt duża liczba błędów (na poziomie 50 Roboczogodzin). |
| **Lista i częstotliwość raportów** | **Okres Rozliczeniowy to jeden kwartał**.   * Poziom dotrzymania terminów usługi WJCR tj. wszystkich zakończonych etapów dotyczących wszystkich zleceń usługi w Okresie Rozliczeniowym  (kwartalnie -sporządzane przez Wykonawcę i przekazywane Zamawiającemu do 15 dni po zakończeniu kwartału). * Poziom wartości wskaźnika WJLB dla każdego zakończonego zlecenia usługi w Okresie Rozliczeniowym   (kwartalnie -sporządzane przez Wykonawcę i przekazywane Zamawiającemu do 15 dni po zakończeniu kwartału). |

1. Czas Naprawy zostanie dostosowany do oferty Wykonawcy. [↑](#footnote-ref-2)
2. Czas Obejścia zostanie dostosowany do oferty Wykonawcy – będzie stanowił 50% Czasu Naprawy zadeklarowanego przez Wykonawcę w Ofercie np. w przypadku zaoferowania Czas Naprawy Awarii w okresie 16-22 dnia każdego miesiąca w wymiarze 10 godzin zegarowych, Czas Obejścia dla takiej Awarii będzie wynosił 5 godzin zegarowych. [↑](#footnote-ref-3)