

ISO/IEC/IEEE 12207:2026 Systems and software engineering, Software life cycle processes (wydanie 2., kwiecień 2026)

Struktura procesowa cyklu życia oprogramowania

63 wymagań. Dla tej normy nie istnieje akredytowany certyfikat. Stan na 07/2026.

Norma ISO/IEC/IEEE 12207 definiuje strukturę 30 procesów obejmujących cały cykl życia oprogramowania, od zawarcia umowy przez rozwój i eksploatację aż po wycofanie z użytku. Nie istnieje akredytowany system certyfikacji: norma wprost traktuje zgodność jako samodeklarację w trzech wariantach (pełna zgodność z wynikami, pełna zgodność z zadaniami lub zgodność dostosowana z udokumentowanym zapisem dostosowania), dlatego wynikiem oceny jest deklaracja zgodności, a nie certyfikat. Organizacje, które potrzebują zewnętrznej oceny, zlecają ocenę procesów opartą na serii ISO/IEC 33000, która dostarcza profil zdolności procesowej w formie raportu.

Firma

Data

Opracował

6.1 Procesy umowne (Agreement Processes)

6

6.1.1

Proces Pozyskiwania jest wdrożony: identyfikowana jest potrzeba oprogramowania lub usług pozyskiwanych zewnętrznie, wybierani są dostawcy, a umowa jest zawierana i monitorowana.

Dokument

Jak to wykazać: Dowody obejmują dokumentację przetargową lub wymagania, porównania ofert, podpisane umowy lub zamówienia oraz protokoły odbioru. W małych zespołach wystarczy krótkie zgłoszenie zakupowe (ticket) w systemie do śledzenia zadań na każdy zakup, obejmujące potrzebę, porównanie alternatyw oraz zatwierdzenie.

Otwarte

W toku

Spełnione

Nie dotyczy

Dowód / uzasadnienie

6.1.1.2

Osiągane są wyniki pozyskiwania: definiowane są wymagania dotyczące pozyskiwanego elementu, umowa zostaje zawarta, a jej wykonanie zostaje potwierdzone.

Jak to wykazać: Kryteria odbioru w umowie lub w zgłoszeniu, wraz z testem odbiorczym lub protokołem odbioru. W przypadku zależności open source dowodem odbioru jest plik blokujący wersje (lockfile) z udokumentowanym kryterium wyboru oraz skan licencji.

Otwarte

W toku

Spełnione

Nie dotyczy

Dowód / uzasadnienie

6.1.1.3

Realizowane są działania pozyskiwania: identyfikacja potrzeby, wybór dostawcy, zawarcie umowy, monitorowanie postępu i odbiór są przeprowadzane w sposób identyfikowalny.

Jak to wykazać: Zapisy z systemu do śledzenia zadań oraz z folderu umów, uzupełnione raportami statusu dostawców. Dla małych organizacji wystarczy lista dostawców z datą ostatniej oceny.

Otwarte

W toku

Spełnione

Nie dotyczy

Dowód / uzasadnienie

6.1.2

Proces Dostawy jest wdrożony: organizacja analizuje zapytania, składa ofertę, zawiera umowę, dostarcza produkt lub usługę i formalnie zamyka umowę.

Dokument

Jak to wykazać: Oferty, potwierdzenia zamówień, dokumenty dostawy i informacje o wydaniu (release notes), protokoły przekazania oraz faktury. Informacje o wydaniu dla każdej wersji są najbardziej praktycznym dowodem dostawy dla zespołów programistycznych.

Otwarte

W toku

Spełnione

Nie dotyczy

Dowód / uzasadnienie

6.1.2.2

Osiągane są wyniki dostawy: zakres dostawy jest uzgodniony, produkt zostaje przekazany, a odpowiedzialność za niego zostaje uregulowana.

Jak to wykazać: Podpisane artefakty wydania z numerem wersji, dziennikiem zmian oraz wiadomością przekazania do klienta. Dodatkowo ustalenia dotyczące wsparcia i gwarancji zawarte w warunkach umowy.

Otwarte

W toku

Spełnione

Nie dotyczy

Dowód / uzasadnienie

6.1.2.3 Realizowane są działania dostawy: oceniana jest wykonalność, realizowana jest dostawa, a umowa zostaje zamknięta po jej wypełnieniu.

Jak to wykazać: Zapis oceny wykonalności lub szacunku nakładu pracy w ofercie, dziennik wdrożenia z potoku CI jako dowód dostawy oraz krótki zapis zamknięcia dla każdego zamówienia.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2 Organizacyjne procesy wspierające projekty (Organizational Project-Enabling Processes)

9

6.2.1 Zarządzanie Modelem Cyklu Życia jest wdrożone: organizacja definiuje, utrzymuje i udostępnia modele cyklu życia, procesy i procedury wykorzystywane przez projekty.

[Dokument](#)

Jak to wykazać: Związy opis procesu w repozytorium, na przykład plik CONTRIBUTING.md wraz z jednostronicowym opisem przepływu pracy zawierającym model gałęzi (branching model), definicję ukończenia (definition of done) oraz rytm wydań. Zmiany w nim można datować dzięki historii Git.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.1.2 Osiągane są wyniki: stosowane modele cyklu życia są zdefiniowane, projekty są ich świadome, a ich skuteczność jest regularnie weryfikowana.

Jak to wykazać: Notatki z retrospektyw lub wpis z corocznego przeglądu w repozytorium, wskazujący powód dostosowania przepływu pracy. Dowodem skuteczności są pull requesty dotyczące dokumentów procesowych.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.2 Zarządzanie Infrastrukturą jest wdrożone: infrastruktura potrzebna do rozwoju, testowania i eksploatacji jest definiowana, zapewniana i utrzymywana.

Jak to wykazać: Infrastruktura jako kod w repozytorium (Terraform, Docker Compose, konfiguracja runnerów CI) wraz z listą używanych usług i ich właścicielami. Dla małych zespołów wystarczy jedna strona z opisem środowisk, dostępów i zasad tworzenia kopii zapasowych.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.2.3 Realizowane są działania infrastrukturalne: identyfikowane są potrzeby, zasoby są pozyskiwane, eksploatowane i wycofywane, gdy nie są już potrzebne.

Jak to wykazać: Historia zmian konfiguracji infrastruktury w Git, panele monitorowania oraz zgłoszenia dotyczące zmian środowisk. Wycofane środowiska są dokumentowane w zgłoszeniu wraz z datą.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.3 Zarządzanie Portfelem jest wdrożone: projekty są inicjowane, priorytetyzowane, zasobowane, na bieżąco oceniane i kończone, gdy przestają przynosić wartość.

Jak to wykazać: Mapa drogowa produktu (roadmap) lub tablica portfela z priorytetami i datami decyzji, wraz z zapisami zatrzymanych inicjatyw. Dla małych organizacji wystarczy dokument mapy drogowej z kwartalnym przeglądem.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.4 Zarządzanie Zasobami Ludzkimi jest wdrożone: określone są wymagane kompetencje, personel jest kwalifikowany, a wiedza jest budowana i zachowywana.

Jak to wykazać: Macierz kompetencji dla każdej roli, zapisy szkoleń i certyfikatów, lista kontrolna wdrożenia nowego pracownika (onboarding). W małych zespołach wystarczy tabela umiejętności pokazująca luki i planowane szkolenia wraz z dowodami odbytych szkoleń.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2.5 Zarządzanie Jakością jest wdrożone: definiowane są cele jakościowe, polityka jakości i odpowiedzialności, a osiągnięcie celów jest oceniane.[Dokument](#)

Jak to wykazać: Krótka polityka jakości w repozytorium, zdefiniowane cele jakościowe (pokrycie testami, gęstość defektów, czas reakcji na incydenty) oraz odpowiednie metryki z CI i systemu do śledzenia zadań.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.2.5.2 Osiągane są wyniki zarządzania jakością: cele jakościowe są mierzalne, odchylenia są wykrywane, a działania korygujące są monitorowane.

Jak to wykazać: Raporty trendów z potoku CI (pokrycie, nieudane kompilacje), statystyki defektów z systemu do śledzenia zadań oraz zgłoszenia wynikających z nich działań wraz z datą zakończenia.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.2.6 Zarządzanie Wiedzą jest wdrożone: wiedza, umiejętności i zasoby wielokrotnego użytku są gromadzone, udostępniane i utrzymywane w dostępności dla przyszłych projektów.

Jak to wykazać: Wewnętrzna baza wiedzy lub wiki, rejestry decyzji architektonicznych (ADR) w repozytorium, biblioteki wielokrotnego użytku z dokumentacją. Mocnym dowodem są notatki pociędyentalne (postmortem).

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3 Techniczne procesy zarządzania (Technical Management Processes)

16

6.3.1 Planowanie Projektu jest wdrożone: planowane są zakres, produkty do dostarczenia, harmonogram, nakład pracy, role i interfejsy, a plan jest na bieżąco aktualizowany.[Dokument](#)

Jak to wykazać: Plan projektu lub backlog z kamieniami milowymi, szacunkami i osobami odpowiedzialnymi. Prowadzona tablica sprintu lub kamieni milowych w systemie do śledzenia zadań spełnia wymóg, o ile zakres i daty są widoczne.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.1.2 Osiągane są wyniki planowania: zadania i zasoby są przydzielone, role są wyznaczone, a plany są uzgodnione z osobami zaangażowanymi.

Jak to wykazać: Przypisane zgłoszenia z osobą odpowiedzialną i wersją docelową, wraz z przeglądem kamieni milowych. Założenia dotyczące pojemności zespołu są krótko odnotowane w zgłoszeniu planistycznym.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.1.3 Realizowane są działania planistyczne: definiowany jest zakres projektu, opracowywane są plany, a przy każdej zmianie są one aktualizowane.

Jak to wykazać: Historia wersji dokumentu planu lub mapy drogowej w Git oraz komentarze zmian przy kamieniach milowych. Dzięki temu każdą aktualizację można datować.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.2 Ocena i Kontrola Projektu jest wdrożona: postęp jest oceniany względem planu, odchylenia są wykrywane, a działania kontrolne są podejmowane.[Dokument](#)

Jak to wykazać: Raporty statusu, przeglądy sprintów, metryki burndown lub przepustowości z systemu do śledzenia zadań oraz zapisy decyzji podjętych w przypadku wystąpienia odchyleń. Wystarczy krótka cotygodniowa notatka o statusie.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.3 Zarządzanie Decyzjami jest wdrożone: decyzje są przygotowywane w sposób ustrukturyzowany, alternatywy są oceniane, a wybór jest uzasadniany i rejestrowany.[Dokument](#)

Jak to wykazać: Rejestry decyzji architektonicznych (ADR) w repozytorium, obejmujące kontekst, opcje, decyzję i konsekwencje. Dla mniejszych decyzji wystarczy uzasadniony komentarz w pull requestie.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.3.2 Osiągane są wyniki zarządzania decyzjami: potrzeba podjęcia decyzji jest rozpoznawana, alternatywy są oceniane, a podjęta decyzja wraz z jej uzasadnieniem jest identyfikowalna.

Jak to wykazać: Folder ADR z kolejną numeracją i statusem (zaproponowana, zaakceptowana, zastąpiona). Historia Git dokumentuje datę i autora.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.4 Zarządzanie Ryzykiem jest wdrożone: ryzyka są identyfikowane, analizowane, obsługiwane i na bieżąco monitorowane.[Dokument](#)

Jak to wykazać: Rejestr ryzyk z prawdopodobieństwem, wpływem, sposobem postępowania i osobą odpowiedzialną. Ryzyka techniczne można śledzić jako zgłoszenia z etykietą ryzyka, wraz z wynikami skanów zależności i podatności.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.4.2 Osiągane są wyniki zarządzania ryzykiem: ryzyka są priorytetyzowane, definiowane są progi działania, a skuteczność postępowania z ryzykiem jest monitorowana.

Jak to wykazać: Historia aktualizacji rejestru ryzyk, zamknięte zgłoszenia ryzyk wraz z uzasadnieniem oraz raporty z automatycznego skanowania zależności pokazujące datę usunięcia problemu.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.5 Zarządzanie Konfiguracją jest wdrożone: elementy konfiguracji są identyfikowane, definiowane są linie bazowe, zmiany są kontrolowane, a stan każdego elementu można w dowolnym momencie ustalić.[Dokument](#)

Jak to wykazać: Git jako system kontroli wersji z tagami dla wydań, chronionymi gałęziami głównymi, plikami blokującymi wersje zależności (lockfiles) oraz zestawieniem składników oprogramowania (SBOM) dla każdego wydania. Historia tagów stanowi zapis linii bazowych.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.5.2 Osiągane są wyniki zarządzania konfiguracją: linie bazowe są ustanawiane, zmiany są identyfikowalne, a wydania są odtwarzalne.

Jak to wykazać: Odtwarzalna kompilacja z tagu Git przez potok CI, wraz z dziennikami kompilacji i rejestrem artefaktów z sumami kontrolnymi.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.3.5.3 Realizowane są działania zarządzania konfiguracją: wnioski o zmianę są oceniane, zatwierdzane lub odrzucane, a ich wdrożenie jest weryfikowane.

Jak to wykazać: Historia pull requestów z obowiązkowym przeglądem i powiązaniem zgłoszeniem oraz reguły scalania w repozytorium (ochrona gałęzi) jako dowód kontroli zatwierdzania.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.3.6 Zarządzanie Informacją jest wdrożone: określone są informacje, które mają być tworzone i przechowywane, a następnie są one utrzymywane, chronione i udostępniane osobom uprawnionym.** [Dokument](#)
- Jak to wykazać: Koncepcja zarządzania zapisami obejmująca miejsce przechowywania, właściciela, okres przechowywania i prawa dostępu. Uprawnienia repozytorium oraz dzienniki kopii zapasowych dokumentują ochronę i dostępność.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.3.7 Proces Pomiaru jest wdrożony: określone są potrzeby informacyjne, definiowane są odpowiednie miary, które są zbierane, analizowane i wykorzystywane do podejmowania decyzji.**
- Jak to wykazać: Metryki z CI i systemu do śledzenia zadań, takie jak czas realizacji (lead time), wskaźnik defektów, pokrycie testami czy częstotliwość wdrożeń. Istotna jest udokumentowana droga od miary do decyzji, na przykład w zapisie retrospektywy.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.3.7.2 Osiągane są wyniki pomiaru: zebrane dane są prawidłowe, analiza jest udokumentowana, a wyniki są komunikowane.**
- Jak to wykazać: Automatycznie generowane raporty potoku ze znacznikiem czasu oraz regularne podsumowanie metryk publikowane na kanale zespołu lub w wiki.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.3.8 Zapewnienie Jakości jest wdrożone: produkty i procesy są niezależnie sprawdzane względem zdefiniowanych kryteriów, a odchylenia są rejestrowane i usuwane.** [Dokument](#)
- Jak to wykazać: Obowiązkowy przegląd na zasadzie czterech oczu w pull requeście, uruchamianie lintera i testów jako bramka scalania oraz rejestrowanie wykrytych odchyłeń jako zgłoszeń. Mały zespół osiąga niezależność dzięki rotacyjnym recenzentom, a nie dedykowanemu działowi QA.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.3.8.2 Osiągane są wyniki zapewnienia jakości: oceniana jest zgodność procesu i zgodność produktu, a działania korygujące są monitorowane aż do zamknięcia.**
- Jak to wykazać: Komentarze z przeglądu wraz ze statusem rozwiązania, zamknięte zgłoszenia błędów odwołujące się do commita naprawczego oraz zielony przebieg potoku dla tagu wydania.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4 Procesy techniczne (Technical Processes)

26

- 6.4.1 Przeprowadzana jest Analiza Biznesowa lub Analiza Misji: określone są problem do rozwiązania, przestrzeń możliwych rozwiązań oraz ograniczenia przedsięwzięcia.**
- Jak to wykazać: Wizja produktu, opis problemu lub uzasadnienie biznesowe (business case) w wiki, uzupełnione analizą rynku lub wywiadami z użytkownikami. Dla małych przedsięwzięć wystarczy epik z opisem problemu i celem w systemie do śledzenia zadań.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

- 6.4.2 Przeprowadzane jest Definiowanie Potrzeb i Wymagań Interesariuszy: identyfikowani są interesariusze, pozyskiwane są ich potrzeby, które są przekształcane w weryfikowalne wymagania interesariuszy.** [Dokument](#)
- Jak to wykazać: Lista interesariuszy, notatki z wywiadów, historyjki użytkownika (user stories) z kryteriami akceptacji. Kryteria akceptacji w zgłoszeniu stanowią weryfikowalną formę wymagania.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.2.2 Osiągane są wyniki: wymagania interesariuszy są uzgodnione, priorytetyzowane i identyfikowalne aż do leżącej u ich podstaw potrzeby.

Jak to wykazać: Priorytetyzowany backlog łączący historyjkę z epikiem oraz komentarz zatwierdzenia klienta w zgłoszeniu.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.3 Przeprowadzane jest Definiowanie Wymagań Systemowych: techniczne wymagania dla systemu oprogramowania są wyprowadzane z wymagań interesariuszy, z uwzględnieniem cech jakościowych i ograniczeń.

[Dokument](#)

Jak to wykazać: Dokument wymagań lub ustrukturyzowane zgłoszenia obejmujące wymagania funkcjonalne i нефункционалне (wydajność, bezpieczeństwo, dostępność). Powiązanie każdego wymagania z przypadkiem testowym czyni weryfikowalność jednoznaczną.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.3.2 Osiągane są wyniki definiowania wymagań: wymagania są jednoznaczne, wolne od sprzeczności, weryfikowalne i identyfikowalne aż do wymagań interesariuszy.

Jak to wykazać: Macierz identyfikowalności (traceability matrix) lub alternatywnie powiązania zgłoszeń od historyjki przez test do commita. Zautomatyzowane testy akceptacyjne stanowią jednocześnie dowód weryfikowalności.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.4 Przeprowadzane jest Definiowanie Architektury Systemu: definiowana i oceniana względem wymagań jest architektura z widokami, blokami składowymi i interfejsami.

[Dokument](#)

Jak to wykazać: Przegląd architektury w repozytorium (na przykład diagramy arc42 lub C4 jako obraz wraz z opisem tekstowym), wraz z rejestrami decyzji architektonicznych zawierającymi ocenione alternatywy.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.4.2 Osiągane są wyniki definiowania architektury systemu: architektura jest oceniana względem wymagań, alternatywy są rozważane, a wybór jest uzasadniany.

Jak to wykazać: ADR rozważający opcje i konsekwencje, wraz z wynikami przeglądu architektury w pull requestie lub w protokole spotkania.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.5 Przeprowadzane jest Definiowanie Projektu: istnieją projekty bloków architektonicznych wystarczające do implementacji, w tym model danych i kontrakty interfejsów.

Jak to wykazać: Specyfikacje interfejsów (OpenAPI, schemat GraphQL, definicje typów), skrypty migracji bazy danych oraz notatki projektowe w pull requestie. Sam kod, wraz z jego typami i granicami modułów, stanowi akceptowalny dowód projektu, pod warunkiem że decyzje są udokumentowane.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.5.2 Osiągane są wyniki definiowania projektu: cechy projektu są identyfikowalne w odniesieniu do architektury, a wykonalność jest potwierdzona.

Jak to wykazać: Odwołania z pull requesta do ADR oraz do wymagania, a także prototypy lub eksperymenty rozpoznawcze (spikes) z notatką wyniku jako dowód wykonalności.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.6 Przeprowadzana jest Analiza Systemu: kwestie techniczne są badane metodami analitycznymi w celu uzasadnienia decyzji dotyczących architektury, projektu lub ryzyk.

Jak to wykazać: Pomiary obciążenia i wyniki testów porównawczych (benchmark), analizy kosztów lub kompromisów, notatki z modelowania zagrożeń. Wyniki powinny znajdować się przy powiązanym ADR jako krótka notatka analityczna.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.7 Przeprowadzana jest Implementacja: elementy oprogramowania są realizowane zgodnie z projektem i regułami kodowania, a wyniki są obejmowane kontrolą konfiguracji.

Jak to wykazać: Historia Git z commitami odwołującymi się do zgłoszeń, reguły lintera i formatowania jako bramka scalania oraz pomyślne kompilacje w potoku CI.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.7.3 Realizowane są działania implementacyjne: definiowane są standardy kodowania, realizacja jest recenzowana, a elementy są udostępniane do integracji.

Jak to wykazać: Pliki konfiguracyjne lintera i formatera w repozytorium, zaliczone przeglądy pull requestów oraz opublikowane artefakty kompilacji w rejestrze.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.8 Przeprowadzana jest Integracja: elementy oprogramowania są stopniowo składane w działający system, a interfejsy są sprawdzane.

Jak to wykazać: Przebiegi ciągłej integracji przy każdym scaleniu, testy integracyjne i testy kontraktowe interfejsów, wraz z dziennikami kompilacji ze znacznikami czasu.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.8.2 Osiągane są wyniki integracji: strategia integracji jest zdefiniowana, interfejsy są spójne, a defekty integracyjne są rejestrowane i usuwane.

Jak to wykazać: Definicja potoku w repozytorium jako strategia integracji, nieudane przebiegi zakończone zielonymi przebiegami jako dowód usunięcia defektu oraz zgłoszenia błędów wraz z commitem naprawczym.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.9 Przeprowadzana jest Weryfikacja: dla każdego określonego wymagania istnieje weryfikowalny, identyfikowalny dowód, że zbudowane oprogramowanie faktycznie je spełnia.[Dokument](#)

Jak to wykazać: Zautomatyzowane testy jednostkowe, integracyjne i regresyjne wraz z raportem testów z potoku CI, analiza statyczna oraz przeglądy kodu. Raport testów dla każdego tagu wydania jest centralnym dowodem.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.9.2 Osiągane są wyniki weryfikacji: kryteria weryfikacji są zdefiniowane, kontrole są wykonywane, a odchylenia są rejestrowane.

Jak to wykazać: Zdefiniowane progi akceptacji (na przykład minimalne pokrycie testami, brak krytycznych ustaleń), zarchiwizowane raporty testów oraz zgłoszenie dla każdego odchylenia.

Otwarte W toku Spełnione Nie dotyczy

[Dowód / uzasadnienie](#)

6.4.9.3 Realizowane są działania weryfikacyjne: zakres i metody weryfikacji są planowane, wykonywane, a wyniki są analizowane.

Jak to wykazać: Plan testów lub strategia testów w formie krótkiej strony w repozytorium, konfiguracja potoku z jego etapami weryfikacji oraz analiza w zatwierdzeniu wydania.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.10 Przeprowadzany jest Proces Przejścia (Transition): system jest instalowany w środowisku docelowym, przekazywany gotowy do użycia, a jego zdolność operacyjna jest potwierdzana. [Dokument](#)

Jak to wykazać: Instrukcja wdrożenia (runbook), dzienniki wdrożenia z potoku, testy dymne (smoke tests) po wdrożeniu oraz udokumentowany plan wycofania. Informacje o wydaniu dokumentują przekazanie użytkownikom i zespołowi operacyjnemu.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.11 Przeprowadzana jest Walidacja: w zamierzonych warunkach użytkowania wykazywane jest, że oprogramowanie faktycznie spełnia pierwotne potrzeby interesariuszy. [Dokument](#)

Jak to wykazać: Testy akceptacyjne z klientem, faza beta lub pilotażowa z informacją zwrotną od użytkowników oraz kryteria akceptacji w zgłoszeniu oznaczone jako spełnione. Dodatkowym wsparciem dla walidacji są dane o użytkowaniu z eksploatacji.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.11.2 Osiągane są wyniki walidacji: kryteria walidacji są zdefiniowane, walidacja odbywa się w reprezentatywnym środowisku, a odchylenia są rejestrowane.

Jak to wykazać: Protokół odbioru z datą i uczestnikami, wyniki ze środowiska staging lub pilotażowego oraz zgłoszenia informacji zwrotnej wraz ze statusem ich rozwiązywania.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.12 Proces Eksploatacji jest wdrożony: system jest eksploatowany zgodnie z przeznaczeniem, jest monitorowany, a użytkownicy otrzymują wsparcie.

Jak to wykazać: Monitorowanie i alertowanie ze zdefiniowanymi progami, zgłoszenia incydentów z czasami reakcji oraz skrzynka wsparcia lub kanał zgłoszeń. Dla małych organizacji wystarczy kilkustronicowy podręcznik eksploatacji.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.12.2 Osiągane są wyniki eksploatacji: strategia eksploatacji i jej działania są zdefiniowane, incydenty są rejestrowane, a wydajność jest oceniana względem oczekiwań.

Jak to wykazać: Metryki dostępności i błędów z monitorowania, dzienniki incydentów oraz analizy poincydentalne (postmortem) wraz z wynikającymi z nich działaniami.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.13 Proces Utrzymania jest wdrożony: poprawki defektów, adaptacje i usprawnienia są rejestrowane, oceniane, wdrażane i wydawane bez naruszania integralności systemu. [Dokument](#)

Jak to wykazać: Zgłoszenia błędów i zmian wraz z priorytetem i przyczyną źródłową, gałęzie poprawek awaryjnych (hotfix), testy regresyjne przed wydaniem oraz informacje o wydaniu dla każdej wersji utrzymaniowej. Regularne aktualizacje zależności są dokumentowane poprzez historię pliku blokującego wersje (lockfile).

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.13.2 Osiągane są wyniki utrzymania: zmiany są analizowane, oceniany jest ich wpływ, a skuteczność poprawki jest potwierdzana.

Jak to wykazać: Analiza wpływu w zgłoszeniu, uruchomienie testów regresyjnych po wprowadzeniu poprawki oraz potwierdzenie ze strony zgłaszającego. Aktualizacje bezpieczeństwa są dokumentowane poprzez raporty ze skanowania podatności pokazujące datę usunięcia problemu.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.14 Proces Wycofania z Eksploatacji jest wdrożony: wycofanie systemów lub ich części jest planowane, dane są migrowane lub usuwane, a zainteresowane strony są informowane.[Dokument](#)

Jak to wykazać: Plan wycofania z datami, dowody migracji lub usunięcia danych (w tym kopii zapasowych), zapis wyłączenia infrastruktury oraz zawiadomienie użytkowników lub klientów.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4.14.2 Osiągane są wyniki wycofania z eksploatacji: obowiązki dotyczące przechowywania są spełnione, środowiska są wyłączane, a odpowiedzialność za system się kończy.

Jak to wykazać: Archiwum danych objętych obowiązkiem przechowywania z podanym okresem przechowywania, zgłoszenie dotyczące wyłączenia zasobów wraz z datą oraz potwierdzenie usunięcia od dostawcy hostingu.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

A Zgodność i dostosowanie (Tailoring)**6****4.1 Zakres stosowania jest zdefiniowany (punkt 4.1, Postanowienia ogólne): ustala się, czy norma jest stosowana do projektu, organizacji czy umowy, oraz które procesy są objęte.**[Dokument](#)

Jak to wykazać: Krótkie oświadczenie o zakresie w dokumentacji procesowej wskazujące projekt, produkt oraz zastosowane grupy procesów. Wystarczy jedna strona w repozytorium.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.2 Deklarowany jest wybrany wariant zgodności: pełna zgodność ze wszystkimi wynikami deklarowanych procesów albo pełna zgodność ze wszystkimi zadaniami deklarowanych procesów.[Dokument](#)

Jak to wykazać: Deklaracja zgodności jako odrębny dokument, wskazujący wariant i odsyłający do dowodów dla każdego procesu. Powstaje w ten sposób samodeklaracja, a nie certyfikat.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.3 Jeżeli deklarowana jest zgodność dostosowana, dostosowanie jest dokumentowane: wskazuje się, które procesy, zadania lub wyniki zostały pominięte, zmodyfikowane lub dodane.[Dokument](#)

Jak to wykazać: Tabela dostosowania z kolumnami: proces, decyzja (przyjęty, dostosowany, pominięty), uzasadnienie i ryzyko. Dla małych zespołów ten dokument jest decydującym dowodem.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

5 Kluczowe koncepcje są stosowane w organizacji: związek między modelem cyklu życia, strukturą projektu a stosowanymi procesami jest przedstawiony w zrozumiały sposób.

Jak to wykazać: Strona przeglądowa odwzorowująca wybrane podejście (na przykład iteracyjne, ciągłe dostarczanie) na 30 procesów. Wystarczy diagram wraz z tabelą.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

A.1 Dostosowanie jest traktowane jako świadoma decyzja: dostosowanie jest dopuszczalne, ale stosuje się je wyłącznie tam, gdzie uzasadniają je ograniczenia przedsięwzięcia, i zawsze jest ono uzasadnione.[Dokument](#)

Jak to wykazać: Kolumna uzasadnienia w tabeli dostosowania odwołująca się do wielkości projektu, klasy ryzyka, sytuacji umownej lub wymagań prawnych. Ogólne usunięcia bez uzasadnienia nie stanowią ważnego dostosowania.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

A.2 Proces dostosowania jest przeprowadzany: identyfikowane są czynniki wpływające, zaangażowani są interesariusze, a decyzje dotyczące dostosowania są podejmowane i dokumentowane.[Dokument](#)

Jak to wykazać: Zapis lub zgłoszenie decyzji o dostosowaniu wskazujące zaangażowane role, wraz z wersjonowaną tabelą dostosowania w Git. Wpływ na ryzyka jest odnotowywany w rejestrze ryzyk.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

ISO/IEC 25010:2023, Inżynieria systemów i oprogramowania, SQuaRE, Model jakości produktu

Model jakości dla produktów oprogramowania

40 wymagań. Dla tej normy nie istnieje akredytowany certyfikat. Stan na 07/2026.

ISO/IEC 25010:2023 to model jakości, a nie norma wymagań z klauzulami podlegającymi audytowi, w związku z czym nie podlega certyfikacji akredytowanej. Model służy jako siatka pozwalająca kompletnie sformułować wymagania jakościowe: pomiar odbywa się zgodnie z ISO/IEC 25020, a ocena zgodnie z ISO/IEC 25040, a rezultatem jest możliwy do przesłania raport oceny, a nie certyfikat. Reklamowanie rzekomej certyfikacji ISO 25010 jest niedopuszczalne.

Firma	Data	Opracował

1 Przydatność funkcjonalna

3

1.1 Zestaw funkcji obejmuje wszystkie określone zadania i cele użytkowników, do których produkt jest wykorzystywany (kompletność funkcjonalna). Luki między udokumentowanymi wymaganiami a zaimplementowanymi funkcjami są znane i ocenione.

Jak to wykazać: Dowód pochodzi z macierzy identyfikowalności łączącej wymaganie z przypadkiem testowym, mierzalnej jako procentowe pokrycie wymagań. W małych zespołach wystarczy system śledzenia zgłoszeń z jedną etykietą na wymaganie oraz jednym testem akceptacyjnym na cel użytkownika.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

1.2 Dla zdefiniowanych danych wejściowych produkt dostarcza wyniki z wymaganym stopniem dokładności (poprawność funkcjonalna). Tolerancje i dopuszczalne odchylenia są określone.

Jak to wykazać: Mierzalną wielkością jest gęstość defektów na tysiąc linii kodu lub na funkcję, potwierdzona testami jednostkowymi i integracyjnymi porównującymi wyniki oczekiwane z rzeczywistymi, a także statystykami błędów z systemu śledzenia zgłoszeń. Dla logiki obliczeniowej należy przechowywać referencyjne zestawy danych z oczekiwanymi wynikami.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

1.3 Dostarczone funkcje wspierają zadania użytkowników bez zbędnych kroków i są dopasowane do przewidzianego zastosowania (adekwatność funkcjonalna).

Jak to wykazać: Dowód pochodzi ze wskaźnika powodzenia zadań i liczby kroków na zadanie użytkownika w niewielkim teście użyteczności z pięcioma uczestnikami, rejestrowanym dla każdego zadania. Dane o użytkowaniu lub zgłoszenia do wsparcia ujawniające obejścia uzupełniają obraz.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

2 Wydajność

3

2.1 Czasy odpowiedzi, przetwarzania i przepustowości pozostają poniżej zdefiniowanych progów (zachowanie czasowe). Progi są określone dla typowych scenariuszy użytkowania.

Jak to wykazać: Mierzalne dzięki czasowi ładowania i opóźnieniu interakcji rejestrowanym za pomocą Lighthouse lub Web Vitals (LCP, INP), a także czasem odpowiedzi serwera z dzienników na poziomie percentyla 95. Progi dla poszczególnych scenariuszy należy zapisać w krótkim budżecie wydajnościowym.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

2.2 Zużycie czasu procesora, pamięci, przestrzeni dyskowej i przepustowości sieci pozostaje w zdefiniowanych granicach (wykorzystanie zasobów).

Jak to wykazać: Dowód pochodzi z wartości monitoringu CPU, RAM i wolumenu danych pod obciążeniem, a także z rozmiaru dostarczanego pakietu z raportu budowania. Dla małych firm wystarczy dziennik pomiarowy z jednego testu obciążeniowego odniesiony do limitów docelowego sprzętu.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

2.3 Maksymalna liczba równoczesnych użytkowników, transakcji lub rekordów jest znana i przestrzegana (pojemność).

Jak to wykazać: Dowód pochodzi z testu obciążeniowego (na przykład k6 lub JMeter) z udokumentowanym punktem nasycenia oraz punktem, w którym wskaźnik błędów zaczyna rosnąć. Wynik należy podsumować w krótkim raporcie pojemności wskazującym rzeczywisty limit i planowane zapotrzebowanie.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

3 Kompatybilność

2

3.1 Produkt działa we współdzielonym środowisku bez zakłócania innych produktów (współistnienie).

Jak to wykazać: Dowód pochodzi z testu przeprowadzonego w środowisku docelowym przy równoległym działaniu innego oprogramowania, rejestrującego konflikty portów, zużycie pamięci i dostęp do plików. Konteneryzacja z określonymi limitami zasobów jest praktycznym rozwiązaniem dla małych zespołów.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

3.2 Wymiana danych z innymi systemami odbywa się przez zdefiniowane interfejsy i formaty (interoperacyjność).

Jak to wykazać: Dowód pochodzi z wersjonowanego opisu interfejsu (OpenAPI lub plik schematu) oraz testów kontraktowych względem tego opisu. Formaty importu i eksportu należy zweryfikować na rzeczywistych plikach próbnych i zachować wyniki.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

4 Zdolność do interakcji

8

4.1 Użytkownicy potrafią bez zewnętrznej pomocy ocenić, czy produkt odpowiada ich potrzebom (rozpoznawalność adekwatności). Cel i korzyść są widoczne z poziomu interfejsu i tekstów poziomu wprowadzającego.

Jak to wykazać: Dowód pochodzi z krótkiego testu pierwszego wrażenia: uczestnicy opisują cel produktu po 30 sekundach, a wskaźnik trafności jest rejestrowany. Zrzuty ekranu strony startowej i ekranów wdrożeniowych wspierają wynik.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

4.2 Nowi użytkownicy osiągają podstawowe zadania w zdefiniowanym czasie i przy ograniczonej liczbie błędów (łatwość uczenia się).

Jak to wykazać: Mierzalne jako czas do pierwszego pomyślnie wykonanego podstawowego zadania oraz liczba nieudanych prób na zadanie, rejestrowane w teście z pięcioma nowymi użytkownikami. Wdrożenie, przewodnik po produkcie lub dane demonstracyjne należy udokumentować jako środki wspierające.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

4.3 Obsługa jest efektywna i kontrolowalna, a powtarzające się zadania można wykonać bez obejść (łatwość obsługi).

Jak to wykazać: Dowód pochodzi z analizy ścieżki kliknięć dla standardowego zadania (kroki docelowe wobec kroków rzeczywistych) oraz z obsługi wyłącznie za pomocą klawiatury, bez myszy. Dziennik testu użyteczności ze wskaźnikiem porzuceń dla każdego zadania uzupełnia dokumentację.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.4 Produkt zapobiega błędom obsługi lub czyni je odwracalnymi (ochrona przed błędami użytkownika). Krytyczne działania wymagają potwierdzenia lub można je cofnąć.

Jak to wykazać: Dowód pochodzi z przypadków testowych z nieprawidłowymi danymi wejściowymi i wartościami granicznymi, a także z potwierdzenia walidacji, okna dialogowego potwierdzenia i funkcji cofania. Błędne dane wejściowe z dziennika produkcyjnego należy ocenić i śledzić ich częstotliwość jako wskaźnik.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.5 Interakcja jest zaprojektowana tak, by być atrakcyjna i utrzymywać motywację użytkowników do kontynuowania zadania (zaangażowanie użytkownika).

Jak to wykazać: Mierzalne dzięki wskaźnikowi powrotów, czasowi trwania sesji i wskaźnikowi ukończenia rozpoczętych procesów pobranym z respektujących prywatność statystyk użytkownika. Krótka ankieta satysfakcji ze skalą oceny wystarcza jako uzupełnienie, także dla małych firm.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.6 Produkt jest użyteczny dla osób o różnych możliwościach, językach i tłach kulturowych (inkluzywność).

Jak to wykazać: Dowód pochodzi z weryfikacji zgodności z WCAG 2.2 poziom AA: współczynniki kontrastu, nawigacja klawiaturą, etykiety dla czytników ekranu, testowane za pomocą axe lub audytu dostępności Lighthouse oraz ręcznej próby. Pokrycie językowe i lokalizację należy wykazać kompletnością plików tłumaczeń.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.7 Użytkownicy otrzymują pomoc dopasowaną do zadania, gdy jej potrzebują, również w sytuacjach błędów i przypadkach brzegowych (pomoc dla użytkownika).

Jak to wykazać: Dowód pochodzi z instrukcji, kontekstowych tekstów pomocy oraz zrozumiałych komunikatów o błędach sugerujących kolejne działanie. Wskaźnik: udział zgłoszeń do wsparcia, które istniejąca pomoc mogłaby rozwiązać, pobrany z systemu zgłoszeń.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

4.8 Interfejs objaśnia się samodzielnie: stany, skutki działań i wymagane dane wejściowe są rozpoznawalne bez wcześniejszej wiedzy (samoopisywalność).

Jak to wykazać: Dowód pochodzi z przeglądu interfejsu według listy kontrolnej: zrozumiałe etykiety, widoczny stan systemu, wskaźniki postępu i ładowania, jednostki przy polach liczbowych. Dziennik obserwacji z testu z użytkownikami wzmacnia dokumentację.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

5 Niezawodność**4****5.1 W normalnej eksploatacji produkt działa bez awarii spowodowanych błędami w oprogramowaniu (bezusterkowość).**

Jak to wykazać: Mierzalne dzięki gęstości defektów i średniemu czasowi między awariami (MTBF) z dzienników produkcyjnych, a także pokryciu testami jako wskaźnikowi zapobiegawczemu. Monitoring błędów, na przykład Sentry, dostarcza dane wejściowe przy niewielkim nakładzie pracy.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

5.2 Produkt jest osiągalny i użyteczny w uzgodnionych oknach czasowych (dostępność). Docelowy poziom dostępności jest określony.

Jak to wykazać: Dowód pochodzi z pomiaru czasu działania za pomocą zewnętrznego monitoringu, wyrażonego jako procentowa dostępność miesięczna i porównanego z uzgodnionym celem. Przerwy w działaniu wraz z czasem trwania i przyczyną należy rejestrować w prostym dzienniku zdarzeń.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

5.3 W obecności awarii sprzętu lub oprogramowania produkt pozostaje w zdefiniowanym, użytecznym stanie (tolerancja na błędy).

Jak to wykazać: Dowód pochodzi z przypadków testowych z symulowanymi awariami (brak sieci, niedostępna baza danych, brak odpowiedzi systemu zewnętrznego) oraz z udokumentowanego zachowania, na przykład trybu offline, ponawiania próby lub widoku zastępczego. Wyniki należy zarejestrować w dzienniku testów.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

5.4 Po awarii dane i stan operacyjny można przywrócić w zdefiniowanym czasie (odtwarzalność).

Jak to wykazać: Mierzalnymi wielkościami są docelowy czas przywrócenia (RTO) i maksymalna dopuszczalna utrata danych (RPO), potwierdzone faktycznie przeprowadzonym testem przywracania z datą i czasem trwania. Dla małych firm wystarczy półroczny test przywracania z krótkim dziennikiem.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6 Bezpieczeństwo**6****6.1 Dane są dostępne wyłącznie dla upoważnionych osób i systemów (poufność). Uprawnienia dostępu są przyznawane według zasady wiedzy koniecznej.**[Dokument](#)

Jak to wykazać: Dowód pochodzi z koncepcji ról i uprawnień, szyfrowania w trakcie transmisji i w spoczynku, a także testów potwierdzających, że dostęp jest odmawiany bez autoryzacji. Kontrola nagłówków i TLS aplikacji uzupełnia dokumentację.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.2 Nieautoryzowane zmiany w programach i danych są zapobiegane lub wykrywane (integralność).

Jak to wykazać: Dowód pochodzi z sum kontrolnych lub podpisów artefaktów, walidacji danych wejściowych oraz skanowania zależności (npm audit, Dependabot) wykazującego, że wykryte problemy zostały usunięte. Ograniczenia bazy danych i dzienniki migracji dodatkowo wspierają ten punkt.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.3 Wykonanych działań nie można później zaprzeczyć (niezaprzeczalność).[Dokument](#)

Jak to wykazać: Dowód pochodzi z odpornych na manipulację dzienników ze znacznikiem czasu i autorem, a także z podpisów, gdy transakcja ma znaczenie prawne. Dla małych firm wystarczy centralny, chroniony przed zapisem cel dziennika.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.4 Działania w systemie można jednoznacznie przypisać do odpowiedzialnej osoby lub usługi (rozliczalność).[Dokument](#)

Jak to wykazać: Dowód pochodzi z dzienników audytowych zawierających identyfikator użytkownika dla każdego działania istotnego dla bezpieczeństwa, połączonych z zakazem współdzielonych kont. Kontrola wrywkowa: prześledzić jedną transakcję od początku do końca w dzienniku i udokumentować wynik.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

6.5 Tożsamość użytkowników i systemów jest weryfikowana w sposób wiarygodny (autentyczność).

Jak to wykazać: Dowód pochodzi z metody uwierzytelniania, polityki haseł i uwierzytelniania dwuskładnikowego dla dostępu administracyjnego, a także z walidacji tokenów i certyfikatów. Przypadki testowe z wygasłymi i sfałszowanymi tokenami potwierdzają skuteczność.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

6.6 Produkt wytrzymuje ataki i pozostaje sprawny podczas ataku (odporność).

Jak to wykazać: Dowód pochodzi ze skanowania podatności, statycznej analizy kodu (SAST) oraz, gdy jest to proporcjonalne, testu penetracyjnego z listą ustaleń i statusem ich usunięcia. Dla małych firm praktycznym rozwiązaniem jest zautomatyzowane skanowanie w potoku wraz z krótkim zewnętrznym testem raz w roku.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

7 Utrzymywalność**5****7.1 Produkt składa się z rozdzielonych komponentów, których modyfikacja wpływa na możliwie najmniejszą liczbę innych komponentów (modularność).**

Jak to wykazać: Mierzalne dzięki sprzężeniu i spójności z analizy statycznej oraz grafom zależności (madge, dependency-cruiser) wykazującym brak cykli. Granice modułów należy uchwycić w krótkim szkicu architektury.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

7.2 Komponenty są zaprojektowane tak, aby można je było wykorzystać w więcej niż jednym kontekście (wielokrotna użyteczność).

Jak to wykazać: Dowód pochodzi ze współdzielonych bibliotek lub komponentów z własnym wersjonowaniem i udokumentowanym interfejsem. Wskaźnik: udział ponownie wykorzystanych komponentów i poziom duplikacji kodu z analizy statycznej.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

7.3 Przyczyny awarii i wpływ planowanych zmian można ustalić efektywnie (analizowalność).

Jak to wykazać: Dowód pochodzi ze strukturyzowanego logowania, monitoringu błędów oraz czytelnych wskaźników kodu, takich jak złożoność cyklomatyczna z SonarQube lub ESLint. Wskaźnik: średni czas do ustalenia przyczyny źródłowej z systemu zgłoszeń.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

7.4 Zmiany można wprowadzać bez wprowadzania nowych defektów i bez obniżania jakości (modyfikowalność).

Jak to wykazać: Dowód pochodzi z testów regresyjnych w potoku oraz wskaźnika błędów wynikających ze zmian, czyli udziału defektów wprowadzonych przez zmiany. Obowiązkowy przegląd kodu i kontrola wersji ze śledzalnymi commitami wspierają ten punkt.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

7.5 Dla produktu i jego komponentów można ustalić i wykonać skuteczne kryteria testowe oraz testy (testowalność).

Jak to wykazać: Mierzalne dzięki pokryciu testami w procentach (linie i gałęzie) z raportu pokrycia oraz czasowi wykonania zestawu testów. Zautomatyzowane uruchamianie testów przy każdym commitcie stanowi dowód, że działa.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

8 Elastyczność

4

8.1 Produkt można dostosować do różnych lub zmieniających się środowisk bez konieczności przebudowy (zdolność adaptacji).

Jak to wykazać: Dowód pochodzi z konfiguracji przechowywanej poza kodem (zmienne środowiskowe, pliki konfiguracyjne) oraz z testów w kilku środowiskach docelowych, przeglądarkach lub systemach operacyjnych w ramach macierzy testowej. Wynik należy zapisać jako tabelę kompatybilności.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

8.2 Produkt obsługuje rosnące lub malejące wolumeny użytkowników i danych bez niedopuszczalnego spadku wydajności (skalowalność).

Jak to wykazać: Dowód pochodzi z testów obciążeniowych z rosnącą liczbą użytkowników, śledzących czas odpowiedzi i wskaźnik błędów, uzupełnionych strategią skalowania (poziomego lub pionowego). Wskaźnik: czas odpowiedzi przy pojedynczym, podwójnym i pięciokrotnym obciążeniu.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

8.3 Produkt można niezawodnie zainstalować, zaktualizować i usunąć w środowisku docelowym (instalowalność).

Jak to wykazać: Dowód pochodzi z powtarzalnej instrukcji instalacji oraz zautomatyzowanego testu instalacji na czystym systemie, mierzonego jako wskaźnik powodzenia i wymagany czas. Odinstalowanie i wycofanie zmian należy wykonać raz i zarejestrować wynik.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

8.4 Produkt może zastąpić inny produkt używany w tym samym celu przy zachowaniu danych i przepływów pracy (wymienialność).

Jak to wykazać: Dowód pochodzi z funkcji importu i eksportu w otwartych formatach oraz z testu migracji z rzeczywistymi danymi z poprzedniego systemu, sprawdzonego pod względem kompletności i wierności danych. Model danych i skrypty migracyjne należy utrzymywać pod kontrolą wersji.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

9 Bezpieczeństwo funkcjonalne

5

9.1 Bezpieczny zakres działania jest zdefiniowany, a produkt wykrywa i obsługuje każde odejście od tego zakresu (ograniczenie eksploatacyjne).

Jak to wykazać: Dowód pochodzi z udokumentowanych granic eksploatacyjnych (wartości progowe, dopuszczalne stany) oraz testów sprawdzających zachowanie poza tymi granicami. Jeśli oprogramowanie nie niesie żadnego potencjału szkody dla osób lub mienia, ten punkt zwykle nie ma zastosowania: taką klasyfikację należy uzasadnić i udokumentować, a nie pominąć bez wyjaśnienia.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

9.2 Zagrożenia, które mogą wynikać z produktu lub jego użycia, są identyfikowane i oceniane systematycznie (identyfikacja ryzyka).[Dokument](#)

Jak to wykazać: Dowód pochodzi z oceny zagrożeń i ryzyka wskazującej prawdopodobieństwo, dotykliwość i środek łagodzący dla każdego zagrożenia, praktycznej w formie tabeli w teczce jakości. Punkt nie ma zastosowania wyłącznie przy pisemnym uzasadnieniu, że nie istnieje potencjał szkody dla osób lub mienia.

Otwarte	W toku	Spełnione	Nie dotyczy
---------	--------	-----------	-------------

Dowód / uzasadnienie

9.3 W przypadku usterki produkt przechodzi w stan bezpieczny (bezpieczne zachowanie awaryjne).

Jak to wykazać: Dowód pochodzi z przypadków testowych z wymuszoną usterką oraz udokumentowanego przejścia do stanu bezpiecznego, na przykład wyłączenia, zablokowania lub powrotu do zweryfikowanej wartości domyślnej. Dla oprogramowania bez potencjału szkody ten punkt zwykle nie ma zastosowania, jednak uzasadnienie nadal powinno znaleźć się w ocenie.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

9.4 Produkt ostrzega o zbliżających się warunkach niebezpiecznych na tyle wcześniej, aby reakcja była możliwa (ostrzeżenie przed zagrożeniem).

Jak to wykazać: Dowód pochodzi z koncepcji alarmów i ostrzeżeń z progami, czasem wyprzedzenia i odbiorcami, a także z dziennika testów uruchomionych ostrzeżeń. Bez potencjału szkody ten punkt zwykle nie ma zastosowania: uzasadnienie musi zostać udokumentowane.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie

9.5 Interakcja z innymi systemami nie tworzy nowych zagrożeń (bezpieczna integracja).

Jak to wykazać: Dowód pochodzi z przeglądu ryzyka interfejsu oraz testów integracyjnych z połączonymi systemami, w tym zachowania w przypadku awarii systemu partnerskiego lub przesyłania przez niego błędnych danych. Dla oprogramowania bez potencjału szkody ten punkt zwykle nie ma zastosowania, co należy odnotować wraz z uzasadnieniem.

Otwarte W toku Spełnione Nie dotyczy

Dowód / uzasadnienie