

Dokument wymagań projektowych

Nazwa projektu: ...

Autorzy: ...

Data: ...

0. Wersje dokumentu

W przypadku zmian w dokumencie, należy określić datę i zakres wprowadzonych zmian.

1. Elementy składowe projektu (produkty projektu)

Należy jednoznacznie i precyzyjnie określić **wszystkie** powstałe w ramach projektu elementy programistyczne i udokumentowane nieprogramistyczne elementy projektu. **Przykładami** elementów programistycznych są:

- aplikacja mobilna dla systemów Android 25+,
- aplikacja serwerowa w języku Python oparta o framework Django,
- instancja serwera bazy danych oparta o silnik MSSQL.

Przykładami elementów nieprogramistycznych są choćby:

- specyfikacja funkcjonalna,
- prototyp wykonany z użyciem narzędzia Adobe XD,
- dokumentacja użytkownika (w formie dokumentu, systemu pomocy, samouczka),
- strona promocyjna projektu
- raporty z wykonanych testów użyteczności,
- mechaniczny robot powstały w związku z realizacją projektu i jego schemat budowy
- udokumentowana analiza porównawcza wydajności technologii, uzasadniająca podjęte decyzje technologiczne.

Elementy składowe projektu należy zaprezentować w podziale na semestry prac, zgodnie z planowanymi terminami ich ukończenia.

2. Granice projektu

W tym paragrafie należy dokonać krytycznej analizy pełnego zakresu funkcjonalności oraz oczekiwań klienta. W szczególności należy określić (z uzasadnieniem), jakie warunki funkcjonalne i niefunkcjonalne są oczekiwane przez klienta, ale nie zostaną wdrożone w trakcie realizacji projektu. Należy wskazać zarówno możliwości, które będą dostępne w systemie, jak i te nie obecne, ze szczególnym uwzględnieniem sytuacji wątpliwych/konfliktowych (granicy systemu). **Przykłady** pytań, które mogą być pomocne przy analizie to:

- Dlaczego aplikacja mobilna działa tylko z systemami Android 25+?
- Dlaczego zdecydowano się na wykonanie aplikacji webowej, chociaż aplikacja mobilna wydawałaby się bardziej odpowiednia?
- Dlaczego niektóre operacje na danych nie mogą zostać wykonane za pośrednictwem aplikacji? (np. możliwość dodawania użytkowników jedynie z poziomu bazy danych)
- Dlaczego nie zdecydowano się na integrację z zewnętrznymi systemami (np. systemami autoryzacji, systemami pośredniczącymi w płatnościach)?
- Dlaczego niektóre funkcjonalności nie zostały (lub nie zostaną) wdrożone, chociaż nie wymagają istotnego nakładu prac?

3. Lista wymagań funkcjonalnych

W tym paragrafie należy szczegółowo określić listę wymagań funkcjonalnych dla całości realizowanego projektu (zakres ten może wykraczać poza ramy zajęć). Można w tym celu wykorzystać user stories albo skrócone przypadki użycia (use cases). Przygotowana lista powinna pokrywać cały zakres funkcjonalności oczekiwanych od projektu, a nie jedynie wybrane z nich. Wymagania powinny być ponumerowane, aby móc się do nich odwoływać w dalszych sekcjach.

4. Lista wymagań niefunkcjonalnych

W tym paragrafie należy szczegółowo określić listę wymagań niefunkcjonalnych dla całości realizowanego projektu. Lista ta może być oparta na udokumentowanym opisie wymagań klienta/udokumentowanych badaniach na grupie docelowej oraz ograniczeniach. Wymagania powinny być ponumerowane, aby móc się do nich odwoływać w dalszych sekcjach.

5. Kryteria akceptacji projektu dla I semestru prac

W tym paragrafie należy określić proces akceptacji produktów projektu po pierwszym semestrze prac, a także kryteria, które pozwolą na rozstrzygnięcie czy produkty projektu są akceptowalne.

W przypadku wymagań funkcjonalnych, niefunkcjonalnych oraz wskaźników wdrożeniowych należy odwołać się do tych określonych w sekcjach 3, 4 oraz 6 (**wskazując jednak słownie ich znaczenie dla zwiększenia czytelności**).

Kryteria akceptacji projektu należy zaprezentować w dwóch kategoriach:

- **wymagane** – kryteria wymagane muszą być spełnione, aby produkt mógł być odebrany przez klienta (np. dostarczenie prototypu, wdrożenie odpowiednich funkcjonalności, pozytywnie zakończone testy obciążeniowe, zrozumienie procesów zachodzących u klienta), bez ich spełnienia projekt nie może zostać dopuszczony do obrony.
- **oczekiwane** – kryteria oczekiwane przez klienta, pozwalające na dalszy rozwój systemu i na jego potencjalne wdrożenie (np. integracja z wewnętrznym systemem autoryzacji klienta, wdrożenie odpowiednich funkcjonalności).

6. Mierzalne wskaźniki wdrożeniowe

W tym paragrafie należy określić w jaki sposób projekt zostanie wdrożony do użytku na koniec drugiego semestru. Ważne jest, aby wskaźniki były mierzalne w sposób obiektywny i wyrażone ilościowo co ma pozwolić określić stopnia ich realizacji. Spełnienie tych wskaźników jest oceniane w kryterium "Czy projekt został wdrożony?". **Przykładami** wskaźników wdrożeniowych mogą być:

- System zostanie udostępniony w domenie internetowej `mojprojekt.com` i będzie z niego korzystać co najmniej 70 użytkowników.
- System zostanie udostępniony w domenie internetowej `mojprojekt.com`, zostanie zasilony pozyskanymi i opracowanymi danymi wejściowymi obejmującymi 300 produktów w 20 kategoriach.
- System zostanie poddany testom obciążeniowym gwarantującym, że w przypadku, gdy z systemu będzie korzystać nie więcej niż 100 użytkowników jednocześnie, to maksymalny czas odpowiedzi serwera nie przekroczy 2 sekund.
- Na koniec drugiego semestru klient otrzyma system w wersji testowej beta i wykorzysta go do przeprowadzenia 25 rzeczywistych/symulowanych transakcji.
- W przeprowadzonych u klienta na 5 użytkownikach testach użyteczności, system osiągnie ocenę ogólną 80%.

7. Kryteria akceptacji projektu dla II semestru prac

Analogicznie do punktu 5. Sekcja ta jest **wymagana** już w pierwszym semestrze.

8. Organizacja pracy zespołu

W tym paragrafie należy wyczerpująco odpowiedzieć na następujące pytania:

- Jaki jest zakres prac poszczególnych członków zespołu, ze szczególnym uwzględnieniem prac programistycznych?
- Jakie role projektowe i deweloperskie przyjęli poszczególni członkowie zespołu?
- Kto zarządza komunikacją z klientem i w jaki sposób jest ona zorganizowana?
- Jaką metodykę pracy przyjął zespół? W jaki sposób dokonano wyboru tej metodyki? Czy rozważano inne metodyki i dlaczego je odrzucono?

- Jakie narzędzia wspomagające prace projektowe wykorzystuje zespół? W jaki sposób zespół zarządza kodami źródłowymi? W jaki sposób zarządza przebiegiem projektu? Czy wykorzystuje się narzędzia CI/CD? Czy stosowane narzędzia są ze sobą zintegrowane? Jeśli tak, to w jaki sposób? Należy też podać adresy dostępowe do tych narzędzi (jeśli jest taka możliwość).
- Jak wygląda diagram cyklu życia zadania w projekcie (od momentu powstania zadania do momentu uznania go za wykonane)?

9. Ryzyka projektowe

W tym paragrafie należy zdefiniować możliwie wszystkie ryzyka związane z realizacją projektu (w tym związane z wykorzystaniem technologii) oraz określić sposoby postępowania w przypadku ich spełnienia. Dla każdego ryzyka należy podać: jego opis, szansę jego zaistnienia, stopień wpływu na projekt. Na przykład:

- W projekcie planujemy wykorzystać model uczenia maszynowego xyz wykorzystywany do rozpoznawania twarzy. Ze względu na fakt, że model ten został wytrenowany na potrzeby rynku azjatyckiego, istnieje ryzyko, że jego skuteczność w przypadku zastosowania w projektach wdrażanych w Europie, może być nieakceptowalna. Jeżeli skuteczność rozpoznawania dla zbiorów testowych spadnie poniżej 75%, w dalszej kolejności sprawdzone zostaną modele abc, def oraz ghi, co zwiększy nakład pracy zespołu o około 20 roboczogodzin. Jeżeli skuteczność tych modeli również będzie niezadowalająca, konieczne będzie wytrenowanie własnego modelu, co zwiększy nakład prac zespołu o dodatkowych 120 roboczogodzin.
- Zaproponowana architektura nie reaguje na obciążenie zgodnie z oczekiwaniami. W szczególności, testy obciążeniowe wykazują, że niespełnione zostaną kryteria akceptacji w tym zakresie. W takiej sytuacji konieczne będzie wdrożenie mechanizmów skalowania w poziomie/w pionie, co zwiększy pracochłonność zespołu o 80 roboczogodzin.
- Planowana infrastruktura wdrożeniowa (np. Firebase) okazuje się zbyt droga na cele projektu inżynierskiego. W takiej sytuacji konieczne będzie przeniesienie projektu na infrastrukturę wydziałową, co zwiększy nakład pracy zespołu o 80 roboczogodzin.
- Klient/pomysłodawca projektu nie wykazuje wystarczającego zainteresowania projektem

10. Kamienie milowe

W tym paragrafie należy określić kluczowe kamienie milowe projektu, uwzględniając w szczególności kolejne wydania projektu wraz z planowanymi terminami ich opublikowania/przekazania klientowi.

Wymagane jest zaplanowanie przynajmniej 2 kamieni milowych na semestr. Kamień milowy projektu powinien być powiązany z przykładowo: wydaniem oprogramowania o nowych funkcjonalnościach, przeprowadzeniem testów (np. akceptacyjnych u klienta), publikacją systemu na Google Play, itp. Realizacja kamienia milowego powinna zostać udokumentowana przez zespół, kamień milowy musi zostać przedstawiony prowadzącemu zajęcia.