

Studencka Pracownia Inżynierii Oprogramowania  
Instytut Informatyki Uniwersytetu Wrocławskiego

Piotr Gunia, Paweł Sikora

# Faza konstrukcji systemu

## Fast Schedule

Wrocław, 4 stycznia 2021

Wersja 0.3

Data	Numer wersji	Opis	Autor
20.12.2020	0.1	Utworzenie dokumentu	Piotr Gunia
28.12.2020	0.2	Uzupełnienie treści	Paweł Sikora
04.01.2021	0.3	Korekta dokumentu	Piotr Gunia

## **Spis treści**

### **1. Testy wymagań funkcjonalnych**

- a. Dodawanie pracownika
- b. Dodanie kompetencji pracownika
- c. Wprowadzanie dyspozycyjności

### **2. Testy wymagań niefunkcjonalnych**

- a. Intuicyjny i prosty interfejs
- b. Interfejs dostosowany do różnych wielkości wyświetlaczy
- c. Niezawodne działanie systemu
- d. Bezpieczeństwo danych
- e. Wielojęzyczność

### **3. Plan beta testowania**

### **4. Plan zarządzania jakością**

### **5. Dokładniejszy plan wykonania produktu**

### **6. Ocena zgodności ze specyfikacją**

# 1. Testy wymagań funkcjonalnych

## a) Test 1

Nazwa: Dodawanie pracownika

Inicjator: Użytkownik (Pracodawca)

Liczba testów: >1000

Główny scenariusz:

1. Pracodawca zatrudnia nowego pracownika i chce wprowadzić go do bazy
2. Pracodawca otwiera zakładkę "Pracownicy"
3. Pracodawca wybiera opcję "Dodaj nowego"
4. Pracodawca wypełnia wyświetlony formularz
5. Pracodawca zatwierdza wprowadzone dane

Krok	Opis	Oczekiwany rezultat
1	Otwarcie formularza	Na wyświetlaczu użytkownika zostaje otwarty formularz do wypełnienia.
2	Wprowadzanie danych	W formularzu widoczne są wszystkie podane dane, jeżeli są one poprawne przycisk "Zatwierdź" staje się aktywny.
3	Zatwierdzenie wprowadzonych danych	Formularz zostaje zamknięty, dane, podane przez użytkownika, są zapisywane w bazie danych.

## a) Test 2

Nazwa: Dodanie kompetencji pracownika

Inicjator: Użytkownik (Pracodawca)

Liczba testów: >1000

Główny scenariusz:

1. Kompetencje pracownika ulegają zmianie i trzeba je zaktualizować w systemie
2. Pracodawca otwiera zakładkę "Pracownicy"
3. Pracodawca wybiera opcję "Kompetencje"
4. Pracodawca wybiera pracownika, którego kompetencje będą edytowane
5. Pracodawca wypełnia wyświetlony formularz
6. Pracodawca zatwierdza wprowadzone zmiany

Krok	Opis	Oczekiwany rezultat
1	Otwarcie spisu pracowników	Na wyświetlaczu przedstawiona zostaje lista pracowników z ich aktualnymi kompetencjami.
2	Otwarcie formularza	Na wyświetlaczu użytkownika zostaje otwarty formularz do wypełnienia.
3	Wprowadzanie danych	W formularzu widoczne są wszystkie podane dane, jeżeli są one poprawne przycisk "Zatwierdź" staje się aktywny.
4	Zatwierdzenie wprowadzonych danych	Formularz zostaje zamknięty, dane, podane przez użytkownika, są

		zapisywane w bazie danych.
--	--	----------------------------

## a) Test 3

Nazwa: Wprowadzanie dyspozycyjności

Inicjator: Użytkownik (Menedżer)

Liczba testów: >3000

Główny scenariusz:

1. Na koniec miesiąca trzeba uzupełnić dyspozycyjność pracowników w celu ułożenia grafiku na kolejny miesiąc
2. Menadżer Otwiera okno "Dyspozycja"
3. Menadżer wybiera interesujący go dzień i pracownika
4. Menadżer naciska "+" i podaje godziny, w których dany pracownik jest dyspozycyjny

Krok	Opis	Oczekiwany rezultat
1	Otwarcie okna "Dyspozycja"	Na ekranie użytkownika otwiera się widok kolejnego miesiąca.
2	Wybranie dnia i pracownika	Wybrany dzień zostaje podświetlony, a imię i nazwisko pracownika pojawia się w polu wyboru.
3	Dodanie godzin dyspozycyjności	W widoku kalendarza w danym dniu wyświetlają się wybrane godziny, a po wybraniu tego dnia pojawia się informacja o pracownikach

## 2. Testy wymagań niefunkcjonalnych

Pomiary mające na celu sprawdzenie czy system spełnia wymagania niefunkcjonalne na podstawie normy ISO/IEC 9126:

### a) Efektywność

Aby aplikacja była efektywna powinna być w stanie utworzyć grafik w ciągu 1 sekundy przy użyciu łącza o przepustowości 100Mb/s. Efektywność będziemy liczyli wzorem (1):

$$X = T_{\text{mean}} / TX_{\text{mean}} \quad (1)$$

Gdzie:

$T_{\text{mean}} \rightarrow \sum(T_i) / N$

$TX_{\text{mean}} \rightarrow$  oczekiwany średni czas (1s)

$T_i \rightarrow$  czas i-tej próby

$N \rightarrow$  łączna ilość prób (1000 prób)

### b) Bezpieczeństwo danych

Bezpieczeństwo przechowywanych danych oraz transferu oceniać będzie specjalista z zakresu IT Security. Aby aplikacja była bezpieczna nie mogą zaistnieć sytuacje umożliwiające przechwycenie danych przez osoby niepożądane. Jedyną osobą mającą dostęp do danych z bazy jest administrator.

### c) Przeność

Aplikacja musi działać bez zarzutów zarówno na smartfonach jak i komputerach osobistych i przenośnych. Testowanie tego wymagania będzie oparte na ankietach wypełnianych przez betatesterów, które pokażą poziom satysfakcji z używania aplikacji. W teście będzie brało udział 50 użytkowników smartfonów, 50 komputerów stacjonarnych i 100 użytkowników laptopów.

#### **d) Dostępność**

Aplikacja powinna być dostępna przez co najmniej 99,5% czasu działania. Pozostały czas powinien wystarczyć na naprawienie nieoczekiwanych błędów, wprowadzanie dużych aktualizacji wymagających restartu serwerów oraz odzyskanie sprawności serwerów po niespodziewanej utracie zasilania. Dostępność będzie liczona wzorem (2):

$$X = T_a / T_t \quad (2)$$

Gdzie:

$T_a$  -> czas działania aplikacji

$T_t$  -> czas od pierwszego uruchomienia aplikacji

Planujemy osiągnąć  $X > 99,5\%$

#### **e) Wdrożenie aplikacji**

Aplikacja zostanie wdrożona poprawnie, jeżeli etap wdrożenia zakończy się w planowanym terminie.

#### **f) Użyteczność**

Aplikacja będzie użyteczna, jeżeli będzie generowała mało błędów. Jeżeli system nie będzie generował dużej ilości błędów będzie to oznaczało, że jest on intuicyjny i łatwy w obsłudze. Użyteczność będziemy określać na podstawie wzoru (3):

$$X = T / N \quad (3)$$

Gdzie:

$T$  -> czas działania

$N$  -> liczba błędów użytkownika

### 3. Plan beta testowania

Testy beta będą podzielone na dwie fazy. Pierwsza zamknięta będzie przeprowadzana przez naszych testerów, druga przez realnych użytkowników

#### **a) Pierwsza faza (zamknięta)**

Testy będą prowadzone przez zatrudnionych przez nas testerów w kontrolowanych warunkach. Testy będą prowadzone głównie pod kątem błędów bezpieczeństwa przy użyci fikcyjnych danych osobowych. Po tej fazie chcemy zapewnić użytkownikom bezpieczeństwo danych. Pierwszą fazę planujemy zakończyć po dwóch tygodniach.

#### **b) Druga faza (otwarta)**

Testy beta będą odbywały się poprzez udostępnienie potencjalnym przyszłym użytkownikom darmowej wersji beta aplikacji. Firmom biorącym udział w testach zagwarantujemy darmowy dostęp do aplikacji na rok od zakończenia testów. W ten sposób planujemy zachęcić co najmniej 30 firm, które w przyszłości będą korzystać z naszej aplikacji. Przez trzy miesiące będą one korzystały z aplikacji, wypełniając ankiety dotyczące satysfakcji z używania naszej aplikacji, raportując ewentualne błędy. Faza ta ma potrwać dwa miesiące, a jej głównym celem jest badanie satysfakcji użytkownika, użyteczności oraz efektywności aplikacji.

### 4. Plan zarządzania jakością

- a) Testowanie aplikacji przez cały okres jej życia
- b) Dodatkowe testy najsłabiej funkcjonujących elementów
- c) Badanie satysfakcji użytkowników poprzez krótkie ankiety
- d) Audyt zewnętrzny oprogramowania po zakończeniu pierwszej fazy testów beta



## 5. Dokładniejszy plan wykonania

- a) Określenie wymagań (2 tygodnie)
- b) Ogólny projekt aplikacji (2 tygodnie)
- c) Iteracje (3 miesiące):
  - 1. Wybór podzbioru funkcji
  - 2. Szczegółowy projekt
  - 3. Implementacja
  - 4. Testy
  - 5. Dostarczenie zrealizowanej części systemu
- d) Pierwsza faza beta testów i poprawki (2 tygodnie)
- e) Druga faza beta testów (2 miesiące)
- f) Szkolenia i wdrażanie (2 tygodnie)
- f) Publikacja oprogramowania
- g) Utrzymanie oprogramowania (nieokreślone)

W sumie 7 miesięcy

## 6. Ocena zgodności ze specyfikacją

Ze stworzonych dokumentów wynika, że projekt jest niemal w całości zgodny ze sporządzoną specyfikacją. Wszystkie założone funkcje zostaną zawarte w projekcie, a wymagania spełnione. Różnice, które zauważamy to wyodrębnienie dwóch faz beta testów. Po przeanalizowaniu koniecznych testów zauważyliśmy, że nasz projekt prawdopodobnie przedłuży się o około miesiąc.