**Plan projektu**

**Psi Hotel**

Spis treści

[1. Wstęp 4](#_Toc5173)

[2. Cel projektu 4](#_Toc31419)

[3. Struktura organizacyjna 4](#_Toc19417)

[4. Role i odpowiedzialność 4](#_Toc26722)

[5. Harmonogram projektu 5](#_Toc18763)

[6. Kosztorys projektu 6](#_Toc4211)

[7. Zasoby ludzkie projektu 6](#_Toc5381)

[8. Standardy i narzędzia w projekcie 6](#_Toc29217)

[9. Procesy zarządzania 6](#_Toc28547)

[9.1 Plan zarządzania konfiguracją 6](#_Toc31148)

[9.1.1. Wstęp 6](#_Toc21235)

[9.1.2. Organizacja dokumentu 7](#_Toc9163)

[9.1.3. Harmonogram projektu 7](#_Toc3942)

[9.1.4. Identyfikacja konfiguracji 7](#_Toc5534)

[9.1.5. Zarządzanie interfejsami 8](#_Toc15755)

[9.1.6. Zasady rejestracji statusów konfiguracji 8](#_Toc5370)

[9.1.7. Zasady kontroli konfiguracji 8](#_Toc25656)

[9.1.8. Zarządzanie podwykonawcami 8](#_Toc27724)

[9.2 Plan zarządzania ryzykiem 8](#_Toc20306)

[9.2.1. Zakres dokumentu 8](#_Toc24172)

[9.2.2. Ogólne zasady zarządzania ryzykiem projektu 8](#_Toc19149)

[9.2.3. Organizacja procesu zarządzania ryzykiem 8](#_Toc12810)

[9.2.3.1. Role w procesie zarządzania ryzykiem 8](#_Toc19168)

[9.2.3.2. Zakres odpowiedzialności dla ról 9](#_Toc22953)

[9.2.3.4. Opis procesu i procedur zarządzania ryzykiem 9](#_Toc4767)

[9.2.4. Ocena ryzyka 11](#_Toc9277)

[9.2.4.1. Identyfikacja ryzyka 11](#_Toc13460)

[9.2.4.1.1. Referencje do formularzy kontrolnych\* 12](#_Toc998)

[9.2.4.1.2. Terminologia 12](#_Toc25885)

[9.2.4.2. Analiza ryzyka 12](#_Toc9059)

[9.2.5. Minimalizacja ryzyka 12](#_Toc5254)

[9.2.6. Kontrola ryzyka 12](#_Toc2090)

[9.2.6.1. Zasady kontroli ryzyka, definicja miar ryzyka 12](#_Toc14149)

[9.2.6.2. Metody pozyskiwania ryzyk 13](#_Toc11859)

[9.2.6.3. Harmonogram kontroli i raportowania 13](#_Toc5309)

[9.2.7. Komunikacja, obieg dokumentów w procesie RM 13](#_Toc13487)

[9.2.8. Raportowanie analizy ryzyka 13](#_Toc31092)

[9.2.9. Dokumentacja ryzyka 13](#_Toc8901)

[9.2.9.1. Lista dokumentów zarządzania ryzykiem 13](#_Toc20387)

[9.2.9.2. Zasady kontroli wersji i dostępu do dokumentów 13](#_Toc24459)

[9.3 Plan zarządzania testami 13](#_Toc31101)

[9.4 Plan zarządzania jakością 15](#_Toc26388)

[9.4.1. Cel dokumentu 15](#_Toc25269)

[9.4.2. Zakres dokumentu 15](#_Toc32399)

[9.4.3. Charakterystyka projektu i budowanego systemu 15](#_Toc11001)

[9.4.4. Identyfikacja obszaru QA 16](#_Toc24899)

[9.4.5. Standardy i procedury QA 16](#_Toc6527)

[9.4.6. Relacja do innych planów projektu 16](#_Toc30320)

[9.4.7. Referencje 16](#_Toc18911)

[9.4.8. QA w strukturze organizacyjnej projektu 17](#_Toc24383)

[9.4.9. Zasoby QA 17](#_Toc17608)

[9.4.10. Zadania QA 17](#_Toc25892)

[9.4.11. Zakres odpowiedzialności za procesy QA 17](#_Toc2901)

[9.4.12. Harmonogram 18](#_Toc21603)

[9.4.13 Dokumentacja 18](#_Toc4618)

[9.4.15. Testy 18](#_Toc27535)

[9.4.16. Zasady rejestracji i raportowania o błędach 19](#_Toc20677)

[9.4.17. Zarządzanie ryzykiem 19](#_Toc4161)

[9.5 Zarządzanie zmianami 20](#_Toc30501)

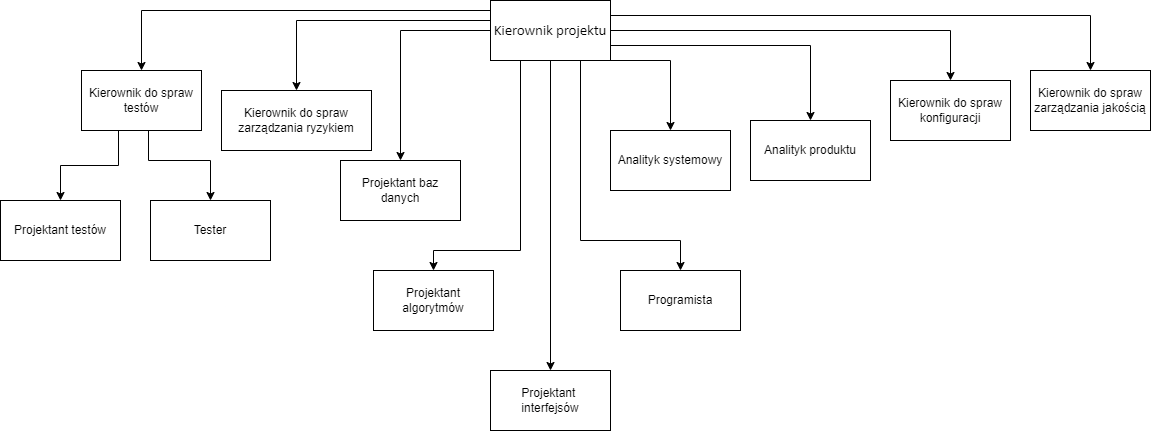
# 1. Wstęp

nie dotyczy

# 2. Cel projektu

Celem projektu jest stworzenie aplikacji pomagającej prowadzić hotel dla psów. Aplikacja ma za zadanie przechowywać informacje o dziennej aktywności każdego z psów przebywających w hotelu, pomóc planować wizyty i zarządzać informacjami o psach. Projekt ma za zadanie przechowywać w sposób zorganizowany dane, które są niezbędne do wykonywania działań w danym hotelu. Dzięki zestawieniu wszelkich niezbędnych informacji, pracownicy wykonujący określone zadanie, mają możliwość sprawdzenia koniecznych informacji o danym zwierzaku.

# 3. Struktura organizacyjna



# 4. Role i odpowiedzialność

**Kierownik projektu** - przygotowuje plan oraz kosztorys projektu, oraz zatwierdza poszczególne etapy wytwarzania aplikacji;

**Kierownik do spraw testów** - Zatwierdza i zakańcza fazę testów;

**Kierownik do spraw zarządzania ryzykiem** - przygotowuje plan zarządzania ryzykiem;

**Kierownik do spraw zarządzania jakością** - przygotowuje plan zarządzania jakością;

**Kierownik do spraw konfiguracji** - przygotowuje plan zarządzania konfiguracją;

**Analityk produktu** - przygotowuje specyfikację wymagań funkcjonalnych i niefunkcjonalnych;

**Analityk systemowy** - przygotowuje analizę specyfikacji wymagań;

**Projektant baz danych** - przygotowuje projekt bazy danych;

**Projektant interfejsów** - przygotowuje projekty interfejsów;

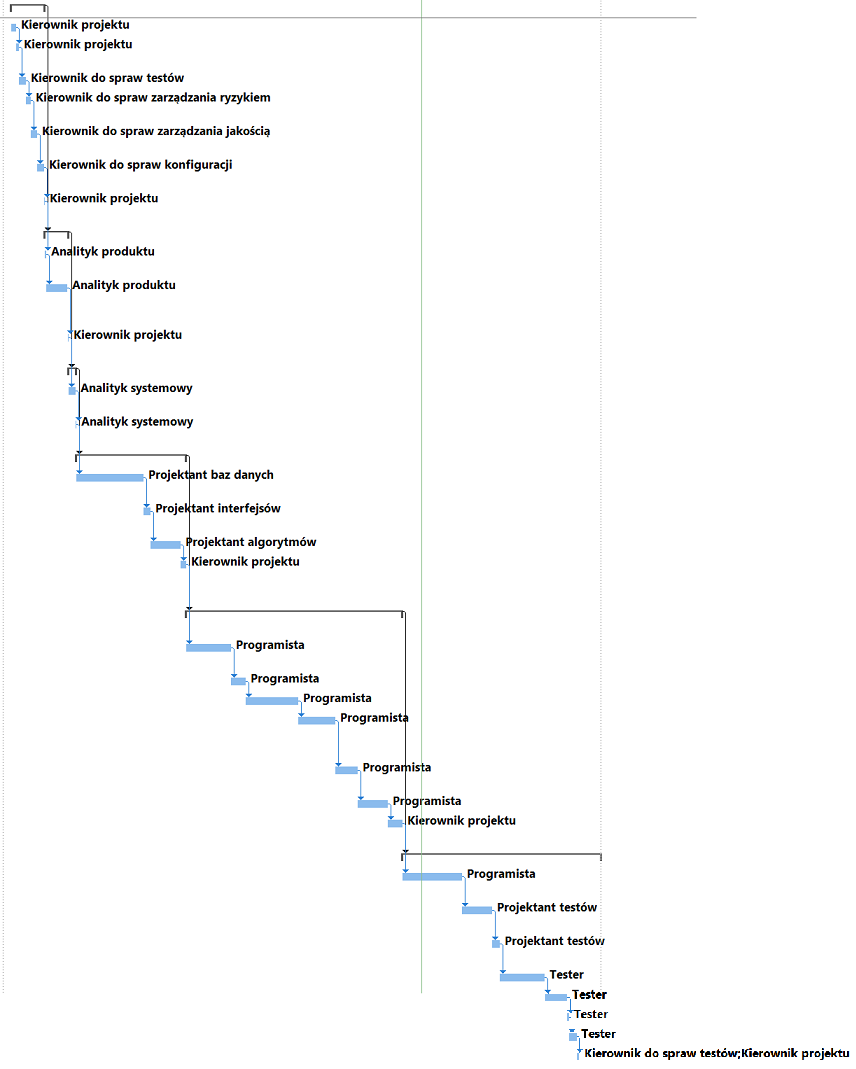
**Projektant algorytmów** - przygotowuje projekty algorytmów;

**Programista** - opracowuje podstawowy kod aplikacji, opracowuje bazę danych i formatki. Przygotowuje oraz implementuje metody dla aplikacji. Łączy aplikacje z bazą danych, oraz opisuje aplikację;

**Projektant testów** - Przygotowuje przypadki testowe i środowisko testowe;

**Tester** - Wykonuje wszystkie cykle testowe, pisze raporty z testów, przygotowuje testy akceptacyjne i bazę błędów.

# 5. Harmonogram projektu



Harmonogram projektu w pliku Harmonogram\_projektu\_Hotel\_ver3\_28.03.2022.mpp

# 6. Kosztorys projektu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lp | Rola | Nakład pracy [h] | Koszt na godzinę [zł/h]] | Koszt ogólny [zł] |
| 1 | Kierownik projektu | 32 | 350 | 8 000 |
| 2 | Kierownik do spraw testów | 10 | 220 | 2 200 |
| 3 | Kierownik do spraw zarządzania ryzykiem | 5 | 220 | 1 100 |
| 4 | Kierownik do spraw zarządzania jakością | 5 | 220 | 1 100 |
| 5 | Kierownik do spraw konfiguracji | 7 | 220 | 1 540 |
| 6 | Analityk produktu | 9 | 280 | 2 520 |
| 7 | Analityk systemowy | 9 | 290 | 2 610 |
| 8 | Projektant baz danych | 16 | 270 | 4 320 |
| 9 | Programista | 152 | 150 | 22 800 |
| 10 | Projektant testów | 24 | 250 | 6000 |
| 11 | Tester | 72 | 75 | 5 400 |

# Zasoby ludzkie projektu

|  |  |
| --- | --- |
| Kierownik projektu - 1 osoba | Analityk systemowy - 1 osoba |
| Kierownik do spraw testów - 1 osoba | Projektant baz danych - 1 osoba |
| Kierownik do spraw zarządzania ryzykiem - 1 osoba | Programista - 2 osoby |
| Kierownik do spraw zarządzania jakością - 1 osoba | Projektant testów - 1 osoba |
| Kierownik do spraw konfiguracji - 1 osoba | Tester - 1 osoba |
| Analityk produktu - 1 osoba |  |

# Standardy i narzędzia w projekcie

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Kompilator | Język programowania | Framework | Edytor tekstu | Arkusz kalkulacyjny | Baza danych | Modelowanie bazy danych |
| Microsoft Visual Studio 2022 | C# | .Net 6.0 | WPS Office dokumenty format .docs | WPS Ofiice arkusz kalkulacyjny format .xls | Microsoft SQL Server Management Studio 2019 | Oracle Data base modeler |

# 9. Procesy zarządzania

## 9.1 Plan zarządzania konfiguracją

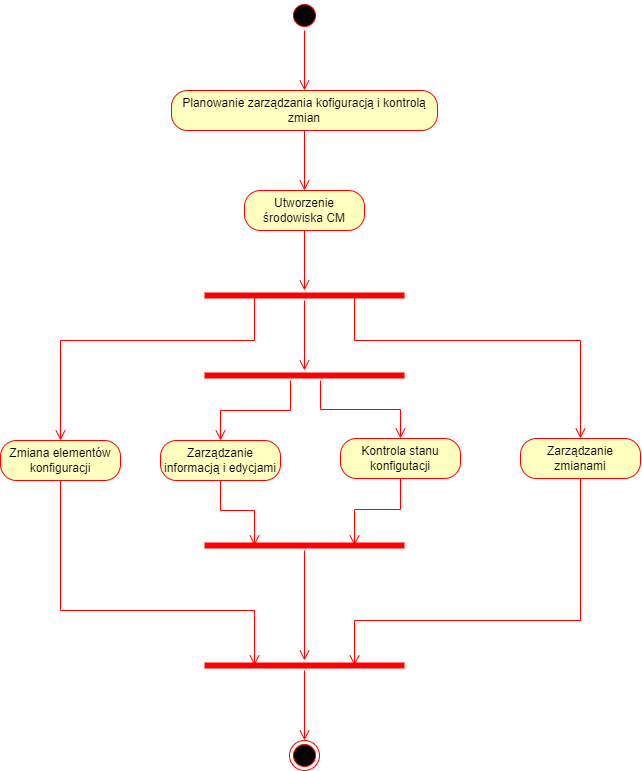
### 9.1.1. Wstęp

**Kontrola konfiguracji dotyczy**

* kodów źródłowych tworzonego oprogramowania,
* plików binarnych tworzonego oprogramowania,
* danych na których operują tworzone programy,
* dokumentacji tworzonego oprogramowania (specyfikacji, instrukcji instalacyjnych i konfiguracyjnych,
* dokumentacji użytkownika),
* dokumentacji związanej z prowadzeniem projektu (planów, raportów, notatek ze spotkań).

### 9.1.2. Organizacja dokumentu

Proces zarządzania konfiguracją i zmianami



### 9.1.3. Harmonogram projektu

Harmonogram projektu: Harmonogram\_projektu ver.2022.03.28 v.3.mpp

### 9.1.4. Identyfikacja konfiguracji

Kolejne dokumenty wersjonowane są za pomocą daty. Kolejne wersje aplikacji wersjonowane są liczbami wymiernymi według zasady +0.1 dla kolejnej wersji tzn. po wersji 1.0 mamy wersje 1, jeśli kolejna wersja wprowadza wiele zmian i jest obszerna zaokrąglamy w górę numer wersji do liczby całkowitej tzn po wersji 1.3 mamy wersje 2.0.

### 9.1.5. Zarządzanie interfejsami

Nie dotyczy

### 9.1.6. Zasady rejestracji statusów konfiguracji

Nie dotyczy

### 9.1.7. Zasady kontroli konfiguracji

Nie dotyczy

### 9.1.8. Zarządzanie podwykonawcami

Brak podwykonawców.

## 9.2 Plan zarządzania ryzykiem

### 9.2.1. Zakres dokumentu

Dokument zawiera zasady zarządzania ryzykiem, organizacje procesów zarządzania ryzykiem, opisuje role i ich odpowiedzialność w takim procesie. Opisuje ryzyka, ich wagę, minimalizacje, metody pozyskiwania i kontroli.

### 9.2.2. Ogólne zasady zarządzania ryzykiem projektu

Zasady zarządzania ryzykiem określają:

1) zakres podmiotowy i odpowiedzialność osób zaangażowanych w proces zarządzania ryzykiem,

2) sposób postępowania przy identyfikowaniu i analizie ryzyka,

3) sposób dokumentowania procesu analizy ryzyka,

4) sposób postępowania ze zidentyfikowanym ryzykiem,

5) sposób monitorowania i nadzoru nad procesem zarządzania ryzykiem.

### 9.2.3. Organizacja procesu zarządzania ryzykiem

Każdy proces zarządzania w projekcie wymaga analizy ryzyka

* analiza ryzyka we wszystkich fazach procesu wytwarzania oprogramowania (w fazie analizy i projektowania, implementacji, testów, instalacji),
* analiza ryzyka w procesie zarządzania konfiguracją,
* analiza ryzyka w procesie zarządzania jakością.
* analiza ryzyka w procesie zarządzania testami.

### 9.2.3.1. Role w procesie zarządzania ryzykiem

Procesem zarządzania ryzykiem zajmuje się kierownik do spraw zarządzania ryzykiem.

### 9.2.3.2. Zakres odpowiedzialności dla ról

Kierownik do spraw zarządzania ryzykiem jest odpowiedzialny za wszystko związane z zarządzaniem ryzykiem.

### 9.2.3.4. Opis procesu i procedur zarządzania ryzykiem

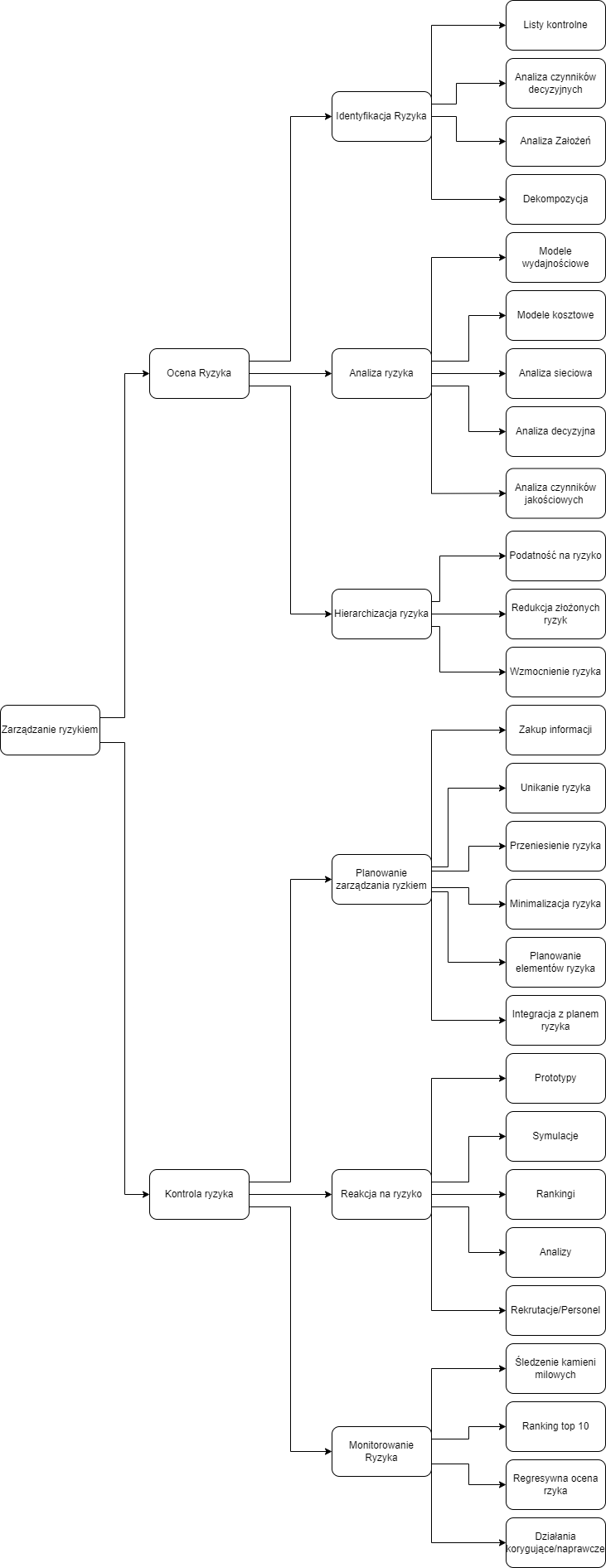
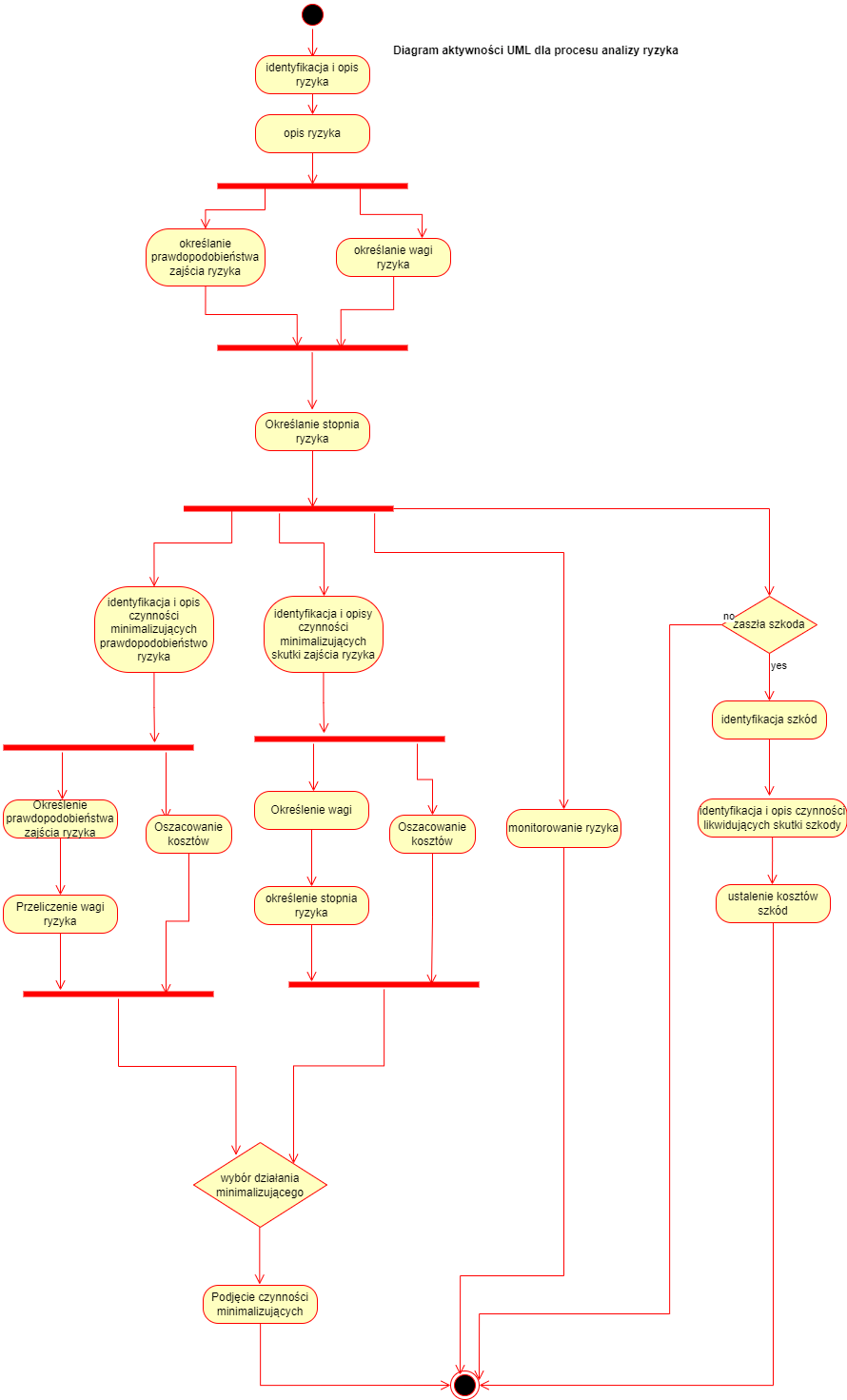
Diagram na podstawie Boehm, B.Software Risk Management Principleand Practices IEEE Software 

Diagram aktywności UML dla procesu zarządzania ryzyka



### 9.2.4. Ocena ryzyka

W skali 1-8, waga ryzyka określa przewidywana wielkość szkody, jaką może spowodować zajście niekorzystnego

zdarzenia.

Mała szkoda, wartość: 1.

Średnia szkoda, wartość: 2.

Poważna szkoda, wartość: 4.

Krytyczna szkoda, wartość: 8.

Prawdopodobieństwo ryzyka - prawdopodobieństwo zajścia niekorzystnego zjawiska.

Bardzo mało prawdopodobne (25%), wartość: 1.

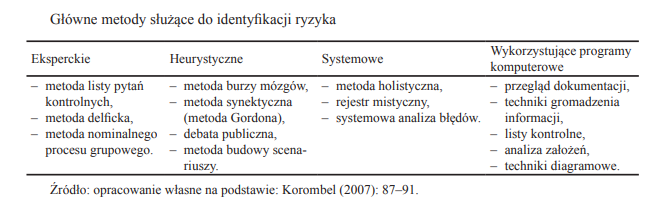
Mało prawdopodobne (50%), wartość: 2.

Prawdopodobne (75%), wartość: 3.

Bardzo prawdopodobne (100%), wartość: 4.

#### 9.2.4.1. Identyfikacja ryzyka

Identyfikacja jest to inwentaryzacja potencjalnych zagrożeń, zanim zaczną wywierać wpływ na realizację przedsięwzięcia informatycznego.



Lista ryzyk specyficznych dla aplikacji:

* Ryzyko w związku z zmianą zarządu
* Ryzyko w związku z powstaniem konkurencji
* Wyjdzie nowy framework, a używany będzie przestarzały
* Używane technologie staną się przestarzałe
* Ryzyko niewypłacalności firmy
* Ryzyko opóźnień w różnych fazach produkcji

##### 9.2.4.1.1. Referencje do formularzy kontrolnych\*

brak

##### 9.2.4.1.2. Terminologia

**Prawdopodobieństwem ryzyka** - jest prawdopodobieństwo zajścia niepożądanego zdarzenia w projekcie.

**Waga ryzyka (wielkość szkody)** - jest kosztem, miarą szkody jaka zajście ryzyka spowoduje w projekcie.

**Stopień ryzyka** - jest iloczyn prawdopodobieństwa ryzyka i wagi ryzyka (wielkości szkody).

#### 9.2.4.2. Analiza ryzyka

Szacowanie prawdopodobieństwa strat

* Metoda Delficka
* Oszacowanie przygotowane przez członków zespołu
* Metoda kasyna
* Wykorzystanie stopniowania określeń bardzo prawdopodobne, prawdopodobne, mało prawdopodobne, nieprawdopodobne

Szacowanie rozmiaru strat:

* Czas(dni), pieniądze
* Model Struktury Podziału Pracy (ang. Work Breakedown Structure)

### 9.2.5. Minimalizacja ryzyka

Minimalizacja ryzyka:

* Działania związane z minimalizacją wpływu lub prawdopodobieństwa wystąpienia czynnika ryzyka do akceptowalnego poziomu
* Powinna uwzględniać koszt podejmowanych czynności w kontekście prawdopodobieństwa ryzyka
* Wcześniej podjęte działania są bardziej efektywne i tańsze niż radzenie sobie z konsekwencjami

### 9.2.6. Kontrola ryzyka

Kontrola ryzyka to korygowanie odchyleń od przewidywanych rezultatów działań wynikających z planowania ryzyka.

#### 9.2.6.1. Zasady kontroli ryzyka, definicja miar ryzyka

Kontrolowanie ryzyka:

* Lista „top 10” (nazwa czynnika ryzyka, pozycja na liście, poprzednia pozycja na liście, liczba tygodni na liście, status rozwiązania dla czynnika ryzyka)
* Międzyfazowe przeglądy czynników ryzyka

**Miara ryzyka** – funkcja, która pozycji finansowej o niepewnej wartości przyszłej przypisuje współczynnik ryzyka, wyrażany przez liczbę rzeczywistą. Ryzyko rozumie się tu jako wysokość rezerw, jakie należy zgromadzić aby zabezpieczyć swoją pozycję.

#### 9.2.6.2. Metody pozyskiwania ryzyk

Metody pozyskiwania ryzyk:

* Wywiad środowiskowy
* Analiza rynkowa
* Analiza najnowszych technologi

#### 9.2.6.3. Harmonogram kontroli i raportowania

brak

### 9.2.7. Komunikacja, obieg dokumentów w procesie RM

brak

### 9.2.8. Raportowanie analizy ryzyka

Zapisanie ryzyka w bazie ryzyk:

Baza ryzyk ver.2022.05.30

### 9.2.9. Dokumentacja ryzyka

#### 9.2.9.1. Lista dokumentów zarządzania ryzykiem

Baza ryzyk ver.2022.05.30

#### 9.2.9.2. Zasady kontroli wersji i dostępu do dokumentów

Do dokumentów ma dostęp „kierownik do spraw zarządzania ryzykiem”.

## 9.3 Plan zarządzania testami

Do pisania testów przeznaczony jest specjalny formularz:

Formularz testu (szablon) ver.2022.05.27.docx

Sporządzono także bazę błędów, do której należy wprowadzić każdy znaleziony błąd:

Baza błędów(szablon) ver.2022.05.27.xlsx

Dla celów testowych określa się statusy błędów:

* **Nowy** (New). Nowo wykryty błąd.
* **Otwarty** (Open). Błąd przyjęty przez programistów do usunięcia (uznanie błędu).
* **Do wyjaśnienia** (To Clarify). Błąd niezrozumiały, źle opisany, niemożliwy do odtworzenia w środowisku programistów.
* **Odrzucony** (Rejected). Błąd odrzucony przez programistów.
* **Przydzielony** (Assigned). Znaleziony błąd został przydzielony konkretnemu programiście do usunięcia.
* **Usunięty** (Fixed). Błąd usunięty. Pakiety z usuniętym błędem zostały przesłane do instalacji w środowisku testowym.
* **Ponownie Otwarty** (Reopen). Poprawione pakiety zostały przetestowanie i błąd nadal występuje.
* **Zamknięty** (Closed). Poprawione pakiety zostały przetestowanie i błąd nie występuje.

Wagi wykrytych błędów

* **Modyfikacja** (Enhancement). Brakująca funkcjonalność nie opisana w specyfikacji funkcjonalnej systemu.
* **Niski** (Low). Błąd który nie ogranicza funkcjonalności systemu, np. błędy językowe.
* **Średni** (Medium). Brakująca funkcjonalność systemu.
* **Wysoki** (High). Brak podstawowej funkcjonalności w systemie.
* **Krytyczny** (Critical). Błąd powodujący awarie systemu (zawieszenie się systemu, nieoczekiwane zamkniecie się systemu, utratę danych, itp.).

Priorytety błędów

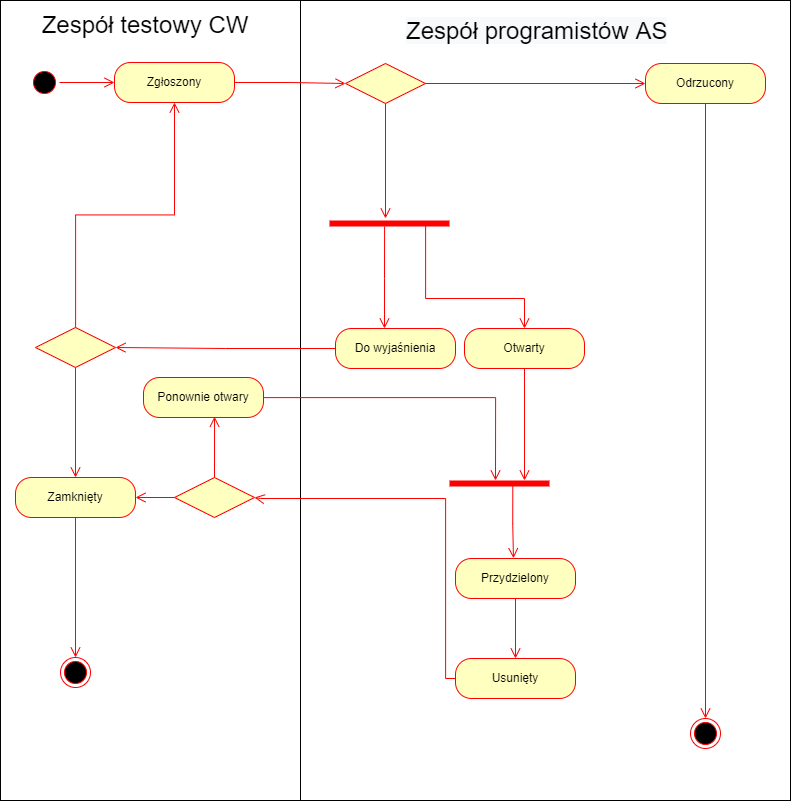
• **Niski** (Low),

• **Średni** (Medium),

• **Wysoki** (High),

• **Pilny** (Urgent). Błąd o tym priorytecie wymaga natychmiastowego usunięcia, np. ze względu na to, że wstrzymuje proces testowy.

Diagram stanu obiektu błąd:



Testy zostaną przeprowadzone w trzech cyklach testowych.

Lista testów:

|  |  |  |  |
| --- | --- | --- | --- |
| Grupa Testów | Cykl 1 (23-28.05.2022) | Cykl 2 (02.06.2022) | Cykl 3 |
| Proces logowania | Blokada konta | Blokada konta |  |
| Komunikat o błędzie i logowanie po błędzie | Komunikat o błędzie i logowanie po błędzie |  |
| Poprawne Logowanie |  |  |
| Wizyty | dodawanie wizyty | dodawanie wizyty |  |
| edycja wizyty |  |  |
| usuwanie wizyty |  |  |

W drugim cyklu przeprowadzone zostaną testy, które w pierwszym cyklu wykryły jakiś błąd. W trzecim ostatnim cyklu przeprowadzone zostaną testy, które mimo naprawy, w drugim cyklu także zakończyły się niepowodzeniem.

## 9.4 Plan zarządzania jakością

### 9.4.1. Cel dokumentu

Celem danego dokumentu jest przedstawienie metod zarządzania jakością w projekcie oprogramowania dla hotelu dla psów.

### 9.4.2. Zakres dokumentu

Rozdział tego dokumentu zawiera szczegółowe opisy charakterystyki projektu oraz budowanego systemu, identyfikacje obszaru QA, standardy i procedury QA, relację do innych planów projektu, referencje. Następnie zostały przedstawione zasoby oraz zadania QA i zakres odpowiedzialności QA. Na samym końcu zostały przedstawione harmonogram, testy, standardy i procedury występujące w projekcie oraz zarządzanie ryzykiem.

### 9.4.3. Charakterystyka projektu i budowanego systemu

Oprogramowanie jest napisane w języku C# z wykorzystaniem frameworka .NET 6.0. Został on wykonany na zlecenie hotelu dla psów i ma zawierać następujące okna z możliwością wprowadzenia, edytowania, przeglądania oraz usuwania wierszy: informacje o psie, informacje o wizytach, informacje o aktywnościach. Dodatkowo oprogramowanie to ma obsługiwać połączenie z bazą danych w celu bezpiecznego oraz efektywnego przechowania danych na temat psów.

### 9.4.4. Identyfikacja obszaru QA

[Rozdział zawiera listę elementów tworzonego systemu (Computer Software Configuration Items) które będą podlegały procesom QA.]

### 9.4.5. Standardy i procedury QA

Standardy według jakich zostały wykonane procedury QA:

• IEEE 730-1998, IEEE Standard for Software Quality Assurance Plans.

• IEEE Std 730.1-1995, IEEE Guide for Software Quality Assurance Planning.

• IEEE Std 1298/A3563.1, Software Quality Management System.

• ISO/IEC 12207:1995 Standard for Information Technology - Software life cycle processes, IEEE Std 12207, 1998.

• CMU/SEI-93-TR-24, Capability Maturity Model for Software, Version 1.1, 1993.

• CMU/SEI-94-HB-01, Carnegie-Mellon University Software Engineering Institute, A Software Process Framework for the SEI Capability Maturity Model (CMM), 1994.

Procedury QA:

1. Analiza zapotrzebowania

2.Planowanie testów

3.Napisanie testów

4.Wykonanie testów oraz raportowanie błędów

5.Powtórzenie testów oraz wykonanie testów regresyjnych

6. Wykonanie testów użytkownika

### 9.4.6. Relacja do innych planów projektu

Nie dotyczy

### 9.4.7. Referencje

[ Rozdział zawiera listę elementów tworzonego systemu (Computer Software Configuration Items) które będą podlegały procesom QA.]

Oprogramowanie posiada następujące elementy:

* baza danych napisana w języku SQL w oprogramowaniu Microsoft SQL Server Manager
* aplikacja napisana w WPF w języku C#
* dokumentacja techniczna oprogramowana

### 9.4.8. QA w strukturze organizacyjnej projektu

[ Rozdział zawiera charakterystykę projektu i ogólny opis tworzonego systemu.]

Projekt jest oprogramowaniem bazo - danowym wspierającym działanie hotelu dla psów. Oprogramowanie ma za zadanie przechowywanie, edytowanie, dodawanie oraz usuwanie informacji na temat psów, aktywności związanych z psem oraz inforamcje na temat ich wizyt i właścicieli.

### 9.4.9. Zasoby QA

[… ]

### 9.4.10. Zadania QA

Przeglądanie specyfikacji jakościowych i technicznych dokumentów projektowych w celu zapewnienia terminowych i znaczących informacji zwrotnych

Tworzenie szczegółowych, kompleksowych i dobrze ustrukturyzowanych planów testów i przypadków testowych

Szacowanie, ustalanie priorytetów, planowanie i koordynowanie działań związanych z testowaniem jakości

### 9.4.11. Zakres odpowiedzialności za procesy QA

Przeglądanie wymagań, specyfikacji i dokumentów oprogramowania, aby zapewnić terminowe i znaczące informacje zwrotne

Tworzenie szczegółowy, kompleksowych i dobrze ustrukturyzowanych planów testów i przypadków testowych

Oszacowanie, ustalanie priorytetów, planowanie i koordynowanie działań testowych

Projektowanie, rozwijanie i wykonywanie skryptów automatyzacji za pomocą narzędzi open source

Identyfikacja, rejestracja, dokładne dokumentowanie oraz śledzenie błędów

Przeprowadzanie dokładnych testów regresji po usunięciu błędów

Opracowanie i zastosowanie procesów testowania nowych i istniejących produktów w celu zaspokojenia potrzeb klientów

Współpraca z zespołami wewnętrznymi (np. programistami i menedżerami produktu) w celu określenia wymagań systemowych

Monitorowanie wyników procesu debugowania

Badanie przyczyn niezgodnego oprogramowania i przeszkolenie użytkowników w zakresie wdrażania rozwiązań

Śledzenie metryki zapewniania jakości, takie jak gęstość defektów i liczba otwartych defektów

Bycie na bieżąco dzięki nowym narzędziom testowym i strategiom testowym

### 9.4.12. Harmonogram

Projekt jest oprogramowaniem bazodanowym mający na celu ułatwienie prowadzenia działalności usługowej czyli hotelu dla psów. Oprogramowanie powinno zawierać wszelkie niezbędne funkcjonalności dla szybkiego oraz czytelnego obsługiwania klientów.

### 9.4.13 Dokumentacja

Następujące elementy powinny być udokumentowane

* zmiany w programie
* dodane nowej funkcjonalności programu
* dodanie nowych elementów niefunkcjonalnych
* zmiana struktury organizacyjnej bądź ilości zasobów ludzkich
* wszelkie zmiany związane z planem projektu
* działania korygujące w projekcie
* działania zapobiegawcze w projekcie
* audyty wewnętrzne
* nadzorowanie niezgodności

Pozostałe elementy wyżej nie wymienione powinny być przedstawione za pomocą innych, często już istniejących w systemie, dokumentów.

Celem procedury jest określenie sposobu wykonania danego działania, którego zakres wyznacza. Procedury są konieczne do realizacji polityki jakości poszczególnych komórek organizacyjnych w projekcie.

### 9.4.15. Testy

Ocena procesów fazy implementacji i testów (unit testów).

Weryfikacja czy

• zachowane są standardy kodowania i czy są przeprowadzane spotkania kontrolne (code review meetings),

• spotkania kontrolne są analizowane a wnioski z analizy są realizowane,

• testy komponentów (unit testy) są przeprowadzane zgodnie z procedurami,

• kod, dokumentacja fazy podlegają kontroli konfiguracji.

Kontrola procesów zarządzania testami, projektowania testów i testowania. Weryfikacja czy:

• proces planowania i projektowania testów jest realizowany zgodnie z procedurami,

• środowisko testowe jest przygotowane i administrowane zgodnie z procedurami,

• proces zarządzania błędami jest zgodny z procedurami,

• raporty z procesu testowania są generowane, dostarczane do odpowiednich osób i analizowane.

### 9.4.16. Zasady rejestracji i raportowania o błędach

Notka opisująca zaobserwowany błąd powinna zawierać informacje takie jak:

– Datę obserwacji, numer wersji na jakiej zaobserwowano błąd, commit itp. – coś co pozwala zidentyfikować wersję

– Typ oraz wersja systemu operacyjnego

– Opis błędu – czyli co zostało zaobserwowane

– Kroki do odtworzenia błędu – wypunktowane kroki

– Aktualny rezultat – czyli co aktualnie się dzieje po wykonanych krokach

– Spodziewany rezultat – czyli jaki powinien być rezultat po wykonanych krokach

Notka weryfikacyjna może zawierać następujące informacje:

– Data weryfikacji, wersja na jakiej przeprowadzana jest weryfikacja

– Typ i wersja przeglądarki lub systemu operacyjnego

– Kroki weryfikacji, bądź krótki opis tego co zostało przetestowane

– Podsumowanie zawierające wynik testów czy test przeszedł czy nie.

### 9.4.17. Zarządzanie ryzykiem

[Dla każdego etapu procesu tworzenia oprogramowania należy sporządzić formularze kontrolne. Lista formularzy kontrolnych.]

• planowania projektu (project planning, tracking and oversight process audit checklist),

• analizy systemowej (system requirements analysis process audit checklist),

• projektowania systemu (system design process audit checklist),

• analizy wymagań (software requirements analysis process audit checklist),

• modelownia architektury systemu (software design process audit checklist),

• implementacji i testów (software implementation and unit testing process audit checklist),

• integracji systemu i testów integracyjnych (unit integration and testing process audit checklist),

• testów i akceptacji systemu (csci integration testing, and system qualification process audit checklist),

• dostarczania systemu (end-item delivery process audit checklist),

• poprawy jakości oprogramowania, usuwania błędów w systemie (software corrective action process audit checklist),

• certyfikacji nośników danych (media certification process audit checklist),

• certyfikacji niedostarczanego oprogramowania (non deliverable software certification process audit checklist),

• przechowywania i przetwarzania danych (storage and handling process audit checklist),

• kontroli podwykonawców (subcontractor control process audit checklist),

• zarządzania konfiguracją (software configuration management process audit checklist),

• kontroli biblioteki projektu (software development library control process audit checklist),

• (non-developmental software process audit checklist).

## 9.5 Zarządzanie zmianami

Priorytety (priority) nadawane ‘żądaniu zmiany'

* niski (low)
* średni (medium)
* wysoki (high)
* pilny (urgent)

**Wagi (severity) nadawane ‘żądaniu zmiany'**

* mała zmiana (enhancement),
* średnia zmiana,
* poważna zmiana.

**Wagi okre**ś**lane s**ą **na podstawie ilo**ś**ci zasobów potrzebnych na implementacj**ę **zmiany**

* czas implementacji,
* liczbę programistów,
* ilość i czas trwania testów.

**Statusy nadawane ‘żądaniu zmiany'**

* dostarczony (submited),
* otwarty (open),
* przydzielony (assigned),
* przetestowany (tested).
* zamknięty (closed)
* do wyjaśnienia (to clarify)
* odrzucony (rejected).

**Poziomy ryzyka nadawane '**Żą**daniu zmiany'**

* niskie ryzyko (ryzyko akceptowalne),
* wysokie ryzyko,
* ryzyko nieakceptowalne.

Proces zarządzania zmianami

