

# Dokument wymagań projektowych

**Nazwa projektu: „Ja nie Taki Ork”**

**Autorzy: Zofia Dobrowolska, Maciej Grochowski, Katarzyna Makohon**

**Data: 18.11.2020**

## 0. Wersje dokumentu

2.2 – Dostosowanie dokumentu do uwag promotora.

2.1 – Dostosowanie dokumentu do uwag promotora.

2.0 – Adaptacja do nowego formatu dokumentu.

## 1. Elementy składowe projektu (produkty projektu)

Wraz z rozwojem oprogramowania do klientów będą dostarczane kolejne wersje aplikacji poszerzające system o kolejne zdefiniowane funkcjonalności. Ostatecznie klienci otrzymają następujące produkty:

### Elementy programistyczne:

#### Semestr I

- Podstawowa aplikacja internetowa oparta o framework React;
- Podstawowa aplikacja serwerowa napisana w języku Java;
- Instancja bazy danych oparta o silnik MySQL.

#### Semestr II

- Rozbudowana aplikacja internetowa oparta o framework React;
- Rozbudowana aplikacja serwerowa napisana w języku Java.

### Elementy nieprogramistyczne:

#### Semestr I

- specyfikacja funkcjonalna;
- prototyp wykonany z użyciem narzędzia Adobe XD;
- Opracowany schemat generowania losowej postaci;

## Semestr II

- Instrukcja obsługi dla użytkownika;
- Instrukcja obsługi dla administratora;
- Diagram generowania postaci;
- Repozytorium kodu wraz z przygotowaniem projektu Open Source.

## 2. Granice projektu

Ograniczenia wynikające ze złożoności prezentowanych danych:

- Brak wersji mobilnej – Z racji na ilość i złożoność prezentowanych danych zespół deweloperski uznał, że wytworzenie czytelnego interfejsu mobilnego i/lub stworzenie aplikacji mobilnej jest niemożliwe. Aby komfortowo używać aplikacji wymagany jest ekran o rozdzielczości co najmniej 1024x768;
- Brak wsparcia dla innych systemów - System nie wspomaga gier RPG nieużywających mechaniki i świata gry Warhammer Fantasy Roleplay (w drugiej edycji). Rozbieżności pomiędzy poszczególnymi systemami jak i wersjami jednego systemu uniemożliwiają łatwą przenaszalność wytworzonego kodu. Wiązałoby się to ze zmianą znacznej części logiki aplikacji.

Ograniczenia nałożone przez bilans kosztów i zysków:

- Brak implementacji mechaniki rzucania kostkami – Mechanika ta jest zbędna w procesie tworzenia postaci, na czym ma skupiać się system. W opinii zespołu deweloperskiego – jak i w opinii klientów - funkcja byłaby zbędna.
- Brak implementacji mapy, po której poruszają się postaci — Mechanika ta jest zbędna w procesie tworzenia postaci, na czym ma skupiać się system. W opinii zespołu deweloperskiego – jak i w opinii klientów - funkcja byłaby zbędna.
- Brak funkcjonalności wgrywania własnych zdjęć/rysunków postaci – Funkcja nie daje użytkownikowi wymiernej korzyści, a jest czasochłonna w implementacji. Wymaga również znacznych zasobów pamięci by przechowywać obrazki. Zespół deweloperski nie może również zapewnić, że po zaimplementowaniu takiej funkcjonalności nie będzie ona wykorzystywana przez graczy, aby łamać prawa autorskie np. upubliczniając objęte tym prawem prace innych.

Ograniczenia wynikające z założeń aplikacji:

- Platforma obsługuje jedynie język polski. – Z analiz przeprowadzonych przez zespół deweloperski znaczna część graczy Warhammera Fantasy Role Play w edycji drugiej

pochodzi z Polski. Zwiększenie więc ilości wersji językowych nie przyczyniłoby się, prawdopodobnie, do zwiększenia liczby użytkowników aplikacji. Dodatkowo planowane wprowadzenie historii tworzonych przez graczy wymuszałoby podziały pomiędzy wersje językowe co kłóci się z przyjętą przez nas koncepcją aplikacji w której dane mają być ogólnodostępne.

- System nie umożliwia szerokiej i szczegółowej edycji postaci po jej wytworzeniu bez tworzenia ich historii. - Postaci z założenia mają być ogólnodostępne. Nie możliwym jest więc wpływanie na statystyki już stworzonej postaci, gdyż jedna postać może być równocześnie używana w sesjach wielu graczy.
- System nie obsługuje paradoksów czasowych wynikających z konwencji gry dziejącej się w innych latach niż domyślne – Dopuszczamy tworzenie postaci żyjących w innych latach niż domyślne jednak nie zostaną one domyślnie wygenerowane przez system.

### 3. Lista wymagań funkcjonalnych (lista skróconych przypadków użycia)

*Słowniczek:*

**Bohater Niezależny** - Inaczej BN - Postać stworzona i odgrywana przez Mistrza Gry. Najczęściej drugoplanowe postacie występujące w grze i spójnie wypełniające świat.

**Bohater Gracza** - Inaczej BG - Postać stworzona i odgrywana przez Gracza. Najczęściej pierwszoplanowe postacie występujące w grze, główni bohaterowie opowiadanej historii.

**Postać** - byt będący opisem cech, historii, zdolności oraz różnorodnych szczegółów wybranego bohatera gry fabularnej. Jej lista cech: imię, nazwisko, płeć, profesja, lista poprzednich profesji, rasa, wiek(data urodzenia), wzrost, waga, kolor włosów, kolor oczu, znak gwiazdny(odpowiednik znaku zodiaku), miejsce urodzenia, miejsce pobytu, cechy charakteru, cechy wyglądu, dominujące emocje, wiara, walka wręcz, umiejętności strzeleckie, krzepa, odporność, zręczność, inteligencja, siłą woli, ogłada, żywotność(HP), maksymalna liczba ataków wykonywana w turze, magia, szybkość, przepowiednia (maksyma nadawana postaci tradycyjnie przy pierwszych urodzinach, zazwyczaj pesymistyczna), historie, umiejętności, zdolności

**GCP** - inaczej Główne Cechy Postaci. Lista: Imię, nazwisko, płeć, profesja, rasa, miejsce pobytu

**Miejsce** - Opisane nazwą oraz opisem konstrukt powiązany z wydarzeniami, będący częścią jednego lub więcej osad.

**Osada** - Miasto, miasteczko lub wieś istniejące w świecie gry.

**WFRP2** - Gra fabularna "Warhammer Fantasy Roleplay", edycja druga.

**Wydarzenie** - Historia, w której biorą znaczący udział przynajmniej dwie osoby, jednoznacznie i koniecznie powiązana z miejscem i czasem.

## *Lista Aktorów:*

**Gracz** - Osoba uczestnicząca w rozgrywce gry fabularnej WFRP2. Jej rola w grze to wykreowanie własnej postaci spójnej ze światem przedstawionym i w zgodzie z zasadami mechaniki gry oraz późniejsze odgrywanie jej w opowieści snutej przez Mistrza Gry.

**Mistrz Gry** - Inaczej MG - Osoba pełniąca rolę Mistrza gry podczas rozgrywki gry fabularnej WFRP2. Rola ta polega na prowadzeniu narracji, odgrywaniu BN, kontrolowaniu spójności świata przedstawionego oraz byciu sędzią w zgodzie z zasadami mechaniki gry.

**Użytkownik Systemu** - Inaczej US - Gracze jak i Mistrzowie Gry.

## 1. Generator

### 1.1 Wygenerowanie Częściowo Losowej Postaci

US chciałby stworzyć nową postać, aby móc z niej swobodnie korzystać w ramach sesji RPG. System pyta się go o poszczególne cechy, jakie powinna posiadać postać. US może wpisać je samodzielnie lub wygenerować losowo. Po akceptacji przez US postać powinna być widoczna dla wszystkich użytkowników.

### 1.2 Stworzenie predefiniowanej postaci

US chciałby zapamiętać utworzoną przez siebie postać. US wprowadza wszystkie parametry wymyślonej postaci. System zapisuje ją.

### 1.3 Stworzenie losowej, wiarygodnej postaci

US chciałby stworzyć nową postać, aby móc z niej swobodnie korzystać w ramach sesji RPG. System przedstawia mu swoją kreację - nową postać zgodną z logiką gry WH2ED - i po akceptacji US zapisuje ją.

## 2. Przeglądanie postaci

### 2.1 Przeglądanie postaci

US chce przejrzeć listę dostępnych postaci, które mógłby wykorzystać w swojej rozgrywce. Komunikuje tę potrzebę Systemowi. Użytkownik chce mieć dostęp do podstawowych statystyk każdej z postaci (GCP). System udostępnia mu potrzebne dane i prezentuje je w czytelny sposób.

### 2.2 Sortowanie postaci

US chce móc sortować postaci, które może przeglądać w odpowiedniej kolejności, aby móc wybrać najdogodniejszą do umieszczenia w swoim RPG. US podaje cechę z GCP, po której chciałby szukać, a system zwraca mu posortowaną po wskazanym fragmencie listę postaci.

### 2.3 Filtrowanie postaci

US chciałby wyszukać postaci po konkretnych atrybutach i komunikuje tę potrzebę systemowi poprzez podanie konkretnej cechy z GCP oraz jej zakresu. System korzystając z zestawu filtrów zwraca mu skróconą listę postaci, których atrybuty mieszczą się w zadanych przedziałach.

### 2.4 Wyświetlanie wszystkich cech danej postaci.

US chciałby przyjrzeć się dokładniej postaci. W tym celu wybiera jedną z postaci dostępnych w bazie, a system w odpowiedzi zwraca mu dokładne dane dotyczące wszystkich cech wybranej postaci

### 3. Ułatwienia dla graczy tradycyjnych

#### 3.1 Ograniczanie cech postaci w widoku dla graczy.

Podczas sesji MG chciałby prezentować tylko część danych swoim graczom, aby nie odbierać im możliwości zgłębiania świata. W tym celu MG chciałby ograniczyć dane odnośnie NPC występujących w sesji. MG wybiera dane, które mają być wizualizowane. System przygotowuje je i generuje w formie, która zadowoli MG i pozwoli mu przekazać dane bezpośrednio graczom.

#### 3.2 Eksport cech postaci do Excela

MG chciałby mieć możliwość wyeksportowania danych (patrz [3.1](#)) do formatu odczytanego przez Excel. W tym celu system powinien przygotować wybrane przez MG dane i umożliwić mu pobranie ich.

#### 3.3 Wydruk wybranych postaci w formie kart postaci

Gracze, jak i MG, chcieliby korzystać z systemu bez konieczności używania prądu przy sesji RPG. Aby uzyskać retro-realizm chcieliby wydrukować "karty postaci" i móc rozdać je graczom. W tym celu prosi system o przygotowanie odpowiednich danych (patrz Przypadek użycia wyżej). System generuje potrzebne dane, tworzy szablon i umożliwia bezpośrednie drukowanie wybranych statystyk i postaci w formie "Kart postaci" lub całych tabel pomocnych mistrzowi Gry.

#### 3.4 Wydruk pustej karty postaci

Gracze grają w RPG bez użycia prądu sporządzają karty postaci samodzielnie. Gracz może poprosić system, aby stworzył i wydrukował puste karty (stylistycznie podobne do kart stworzonych w punkcie 3.3). System przygotowuje i pozwala na druk takiej karty.

#### 3.5. Motyw widoku listy postaci

Mistrz Gry chce zautomatyzować proces ograniczania cech postaci w widoku dla graczy (Patrz [3.1](#)). W tym celu samodzielnie definiuje indywidualne schematy widoku (wybiera informacje które mają być widoczne). Następnie może łatwo wybrać dany motyw i wyświetlić wszystkie dane w nim oznaczone.

#### 3.6 Predefiniowane motywy widoku listy postaci

Mistrz Gry podczas sesji chce przełączać się pomiędzy danymi na temat postaci. Wybiera jeden z predefiniowanych motywów widoku postaci (walka, odgrywanie). System wyświetla mu informacje domyślnie potrzebne lub przydatne w danej sytuacji.

#### 3.7 Opis sesji

MG chciałby na początku danej sesji RPG poinformować graczy o odstępstwach od ogólnie przyjętej konwencji WFRP2 jakie wprowadza do tej konkretnej rozgrywki. Podaje dokładną treść wiadomości, a System wyświetla ją informując wszystkich uczestników sesji.

#### 3.8 Dodawanie innych użytkowników do sesji

Gracz chciałby w łatwy sposób mieć dostęp do podglądu aktualnie prowadzonych i/lub granych sesji. Informuje on system do których sesji powinien być przypisany. System przypisuje go i umożliwia łatwy dostęp do wszystkich sesji w których uczestniczy gracz.

#### 3.9 Udostępnianie widoku sesji

MG chciałby udostępnić graczom opisy postaci, które już napotkali w swojej rozgrywce, aby mogli się do nich odnieść. Prosi system, aby ten pozwolił mu wizualnie zaprezentować te dane. System umożliwia mu to.

## 4. Funkcje społecznościowe

### 4.1. Blokowanie autora.

US chciałby mieć możliwość zablokowania twórczości (historii napisanych przez tego konkretnego użytkownika) danego użytkownika w tym celu podaje systemowi jego identyfikator a system ukrywa przed nim wszystkie historie i postaci stworzone przez użytkownika o danym identyfikatorze tak aby treści te nie pojawiały się kiedy ten konkretny gracz przegląda stworzone postaci lub historie.

### 4.2 Wyróżnianie autora

US chciałby mieć możliwość obserwowania historii stworzonych przez konkretnego gracza. Podaje jego identyfikator, a system wyróżnia historie tworzone przez tego użytkownika

### 4.3 Ocenianie historii

US chce mieć możliwość wyrażenia opinii na temat przeczytanej historii. Chciałby móc w jakiś sposób zobaczyć, te historie, które gracze ocenili pozytywnie. Chce aby system proponował mu je w pierwszej kolejności. W tym celu przekazuje systemowi która historia mu się podoba, a która nie, a system zlicza głosy wszystkich graczy lepiej pozycjonując historie przyjęte pozytywnie przez ogół graczy.

## 5. Historie

### 5.1 Tworzenie historii

US chciałby dodawać historie, które wymyślił dla danej postaci ubarwiając świat. W tym celu opisuje dokładnie historię, a system wyświetla ją jako jedną z dostępnych do odczytania dla tej konkretnej postaci.

### 5.2 Przeglądanie historii

US chciałby przeglądać historie napisane przez użytkowników aby zasięgnąć inspiracji do własnych sesji. Powiadamia System o tej chęci, a on w odpowiedzi prezentuje wizualnie.

### 5.3 Filtrowanie historii

US chciałby specyfikować wymagania przeglądanych historii. US podaje systemowi swoje wymagania (nazwę autora, postać lub datę) system zwraca historie o wskazanych parametrach i wyświetla je użytkownikowi.

### 5.4 Umieszczanie postaci wewnątrz historii innych postaci

US chce móc oznaczyć postać, która pojawiła się jako bohater poboczny opisywanej historii w tym celu podaje systemowi dane tej postaci, a system dba o to, aby w opublikowanej już opowieści istniał odnośnik do takiej oznaczonej postaci.

### 5.5 Dodawanie miejsc do historii

US chciałby mieć możliwość przeglądania historii, które działy się w konkretnej lokacji. W tym celu System, podczas tworzenia nowej historii proponuje możliwość dodania do niej miejsca. Tak oznaczone historie będą mogły zostać poddane filtrowaniu po miejscu.

### 5.6 Dodawanie alternatywnych historii bądź kontynuacji

US chciałby mieć możliwość odniesienia się do historii napisanych przez innych, bądź napisania dla danej postaci historii alternatywnej. W tym celu, pisząc swoją historię chciałby mieć możliwość oznaczenia, że odnosi się ona do wcześniejszej historii lub jest dla nich

alternatywą. System powinien to odnotowywać i pokazywać wszystkim zainteresowanym graczom informacje tak oznaczoną.

## 6. Profil Gracza

### 6.1 Wysyłanie wiadomości

US chciałby skontaktować się z innym użytkownikiem. W tym celu prosi system o umożliwienie mu przesyłania wiadomości. System realizuje prośbę w sposób preferowany przez odbiorcę.

### 6.2 Opis prowadzonych sesji

MG chciałby jakoś zakomunikować innym użytkownikom, że w prowadzonych u niego sesjach występują pewne odstępstwa od klasycznej konwencji WFRP2.

### 6.3 Profil gracza z listą stworzonych postaci i historii.

Użytkownik chciałby posiadać swoją wizytówkę na stronie po to aby wszyscy inni użytkownicy widzieli jego twory. W tym celu prosi system o katalogowanie informacji o samym użytkowniku (takich jak ksywka czy opis), dołączenie do tego listy stworzonych postaci oraz napisanych historii.

## 7. Miejsca

### 7.1 Dodawanie miejsca

US chciałby zdefiniować miejsce. Aby dodać je do historii jednej z postaci w aplikacji. System odpytuje użytkownika dokładnie o to miejsce i generuje takie, jak gracz sobie zażyczył.

### 7.2 Przeglądanie miejsc.

US chciałby mieć możliwość przeglądać miejsca stworzone przez wszystkich użytkowników. System pozwala mu na to z dokładnością do osady wybranej przez użytkownika i prezentuje w czytelnej formie.

## 4. Lista wymagań niefunkcjonalnych

- Udostępniony zostanie plik opisujący architekturę systemu.
- Udostępniony zostanie plik omawiający jak przygotować środowisko i skompilować projekt.
- Udostępniony zostanie instrukcja dla użytkownika omawiająca podstawowe funkcje systemu istotne z perspektywy użytkownika.
- Tworzone postaci będą zgodne z ogólną logiką świata Warhammera, co zostanie przetestowane przez klientów.
- Tworzone postaci będą zgodne z mechaniką Warhammer Fantasy Roleplay 2ed, co zostanie przetestowane przez klientów.

## 5. Mierzalne wskaźniki wdrożeniowe

Semestr I:

- Wdrożenie aplikacji i udostępnienie jej pod adresem internetowym [janietakiork.projektstudencki.pl](http://janietakiork.projektstudencki.pl);

Semestr II:

- Przetestowanie aplikacji pod kątem użytkowym przez 3 testerów (klientów końcowych) oraz zdobycie ich akceptacji;
- Przygotowanie środowiska produkcyjnego pod adresem internetowym `janietakiork.projektstudencki.pl`;
- 20 predefiniowanych postaci z osobnymi historiami;
- Udostępnienie kodu na GitHubie wraz z dokumentacją wymaganą do podjęcia pracy nad projektem.

## 6. Kryteria akceptacji projektu dla I semestru prac

- **Wymagane**
  - 1.1 Wygenerowanie częściowo losowej postaci;
  - 1.2 Stworzenie predefiniowanej postaci;
  - 2.1 Przeglądanie postaci;
  - 2.2 Sortowanie postaci;
  - 2.3 Filtrowanie postaci;
  - 2.4 Wyświetlanie wszystkich cech danej postaci.
- **Oczekiwane**
  - 1.3 Stworzenie losowej, wiarygodnej postaci;
  - 3.6 Predefiniowane motywy widoku listy postaci;
  - Przeprowadzenie testów użyteczności.
- **Planowane**
  - 3.1 Ograniczanie cech postaci w widoku dla graczy;
  - 3.2 Eksport cech postaci do Excela;
  - 3.3 Wydruk wybranych postaci w formie kart postaci;
  - 3.4 Wydruk pustej karty postaci.

## 7. Kryteria akceptacji projektu dla II semestru prac

- **Wymagane**
  - Przeprowadzenie testów użyteczności;
  - Publikacja kodu wraz z instrukcją deweloperską w serwisie GitHub;
  - 3.7 Opis sesji;
  - 3.8 Dodawanie innych użytkowników do sesji;
  - 3.9 Udostępnianie widoku sesji;
  - 5.1 Tworzenie historii;
  - 5.2 Przeglądanie historii.
- **Oczekiwane**
  - 5.3 Filtrowanie historii;
  - 5.4 Umieszczanie postaci wewnątrz historii innych postaci;
  - 5.5 Dodawanie miejsc do historii;
  - 6.2 Opis prowadzonych sesji;
  - 6.3 Profil gracza z listą stworzonych postaci i historii;
  - Oddanie systemu do testów przez klientów.
- **Planowane**
  - 3.5. Motyw widoku listy postaci;



- 4.1. Blokowanie autora;
- 4.2 Wyróżnianie autora;
- 4.3 Ocenianie historii;
- 5.6 Dodawanie alternatywnych historii bądź kontynuacji;
- 6.1 Wysyłanie wiadomości;
- 7.1 Dodawanie miejsca;
- 7.2 Przeglądanie miejsc.

## 8. Organizacja pracy zespołu

### Omówienie członków zespołu:

Zofia Dobrowolska:

- Prace nad prototypem i UX;
- Prace programistyczne - głównie backend;
- Prace wdrożeniowe.

Maciej Grochowski:

- Pozyskanie klientów i kontakt z klientem;
- Pozyskanie danych zasilających aplikację;
- Prace programistyczne – full stack.

Katarzyna Makohon:

- Skryba podczas kontaktów z klientami;
- Dokumentacja i zarządzanie procesem.
- Prace programistyczne – głównie frontend.

### Omówienie kontaktów z klientem:

Cała komunikacja z klientami odbywa się na platformie Discord. W fazie zbierania wymagań z każdym z klientów osobno przeprowadzono rozmowy głosowe. Ich propozycje zostały spisane w formie Use Caseów i udostępnione im do wglądu. Klienci zostali poproszeni o przydzielenie wartości punktowej ustawiając tym samym funkcjonalności w listę priorytetów.

Obecnie klienci są konsultantami w sprawach niejednoznacznych. Są informowani o najnowszych wdrożeniach. Mogą wyrazić swoją opinię na temat zaimplementowanych funkcjonalności zarówno w formie głosowej jak i pisemnej.

Pod koniec pierwszego semestru przeprowadzili testy użyteczności. Pod koniec drugiego semestru planowane jest powtórzenie powyższych testów jak i przeprowadzenie testów akceptacyjnych.

### Omówienie metodyki:

Zespół zdecydował się na wytworzenie własnej metodyki w oparciu o manifest programowania zwinnego. Czerpiąc ze Scruma, Kanbanu jak i z wiedzy na temat tworzenia Use Caseów.

Główne cechy odróżniające naszą metodykę od Scruma:

- Wymagania spisane w formie skróconych przypadków użycia.

- Przyrosty mające długość miesiąca.
- Dwie osoby w zespole mają kontakt z klientem, a więc i rozdzielanie funkcji Product Owenera. Jedna rozmawia osoba, a druga zapisuje i doprecyzowuje słowa klienta umożliwiając płynną rozmowę z klientem i zwiększając ilość zebranych danych.
- Brak Scrum Mastera. Jego funkcję przejmuje jedna z osób Product Owenera.

Przedstawiony schemat pracy został wybrany na podstawie doświadczeń zespołu deweloperskiego w innych projektach, specyfiki samego zespołu jak i projektu jakiego się podjął. Rozważanie metodyk „czystych” doprowadziło do konkluzji, że zespół z racji swojego rozmiaru nie wymaga aż tak wyspecjalizowanych ról i narzędzi. Zauważono również, że rozbieżności funkcji Product Owenera przyczyni się do lepszego podziału kompetencji w zespole.

Wytworzony kod przechowywany jest w repozytorium publicznym w serwisie Github. Organizacja pracy, tworzenie backlogu, przydzielanie zadań są przeprowadzane w porozumieniu ze wszystkimi członkami zespołu, a efektu cotygodniowych rozmów widoczne są na platformie Jira.

Na platformie Jira odbywa się również biletowanie zadań. Wyszczególniamy podział na 4 tablice.

- Do zrobienia – Zadania, które czekają na to, aby zajęły się nimi poszczególne osoby z zespołu.
- W toku – Zadania, które są w tym momencie robione. Każde zadanie w tym miejscu musi mieć już przypisaną osobę, która je wykonuje.
- Do weryfikacji – Osoba, która wykonała zadanie przesuwa je z „w toku” na „do weryfikacji”. Zadanie czeka na sprawdzenie przez innego członka zespołu. Jeżeli sprawdzenie to nie wykryje błędów przesuwa zadanie na gotowe. Jeżeli błędy zostają wykryte zadanie zostaje przeniesione na „w toku”, a osoba odpowiedzialna za zadanie powiadomiona jakie błędy należy poprawić.
- Gotowe – Sprawdzone zadania, bez wykrytych błędów.

Warto zauważyć, że zespół poza podziałem na tablicę stosuje również system epik pozwalający na przyporządkowanie zadań pod konkretny przypadek użycia.

## 9. Ryzyka projektowe

Z uwagi na specyfikę projektu głównym obszarem ryzyka, jaki dostrzegamy, jest ten związany ze zmiennymi nastrojami społecznymi.

Ryzyko o największym wpływie na projekt, ale równocześnie znikomej możliwości wystąpienia jest ryzyko zmiany licencji Games Workshop odnośnie do posiadanej przez nich marki Warhammer. Obecnie firma ta udostępnia świat jak i mechanikę, a także pozwala na tworzenie dodatków przez fanów pod warunkiem nieodpłatnego ich udostępnienia (<https://www.games-workshop.com/en-PL/Intellectual-Property-Policy>). Zmiana licencji na bardziej rygorystyczną wiązałaby się z porzuceniem produktu bądź próbą całkowitego przeniesienia go na inny system, co jest równoznaczne z przebudowaniem znacznej części aplikacji. Licencja ta nie zmieniła się jednak od ponad 15 lat, zatem ryzyko to jest raczej znikome.

Nie zaniedbywalnym czynnikiem są również sami fani, którzy mogą chcieć zmienić system na mniej skomplikowany, nowszy bądź bardziej popularny (obecnie Warhammer2ED jest najpopularniejszym systemem w Polsce, jednak na rynku jest już Warhammer3ED, a także Warhammer4ED wydany w 2018 roku). W przypadku dalszego rozwoju edycji czwartej (Trzecia nie

została ciepło przyjęta przez społeczność) istnieje szansa konwersji systemu, który z założenia jest dość podobnego do popularnej, drugiej edycji.

Istnieje również szansa, że “moda”, która nastąpiła na gry RPG w ostatnich latach wygaśnie, co również skutkuje zmniejszeniem liczby aktywnych graczy, a co za tym idzie użytkowników systemu.

Dostrzegamy również ryzyko wynikające ze straty zainteresowania klientów produktem. Nie jest to jednak szczególnie problematyczna kwestia, gdyż w takim przypadku można byłoby zaangażować innych przedstawicieli grupy fanów.

## 10. Kamienie milowe

### Faza I Kwiecień 2020 - Czerwiec 2020

- Zebranie wymagań i spisanie ich jako Use Case;
- Przygotowanie prototypu aplikacji;
- Utworzenie strony internetowej dostępnej pod określonym adresem;
- Umożliwienie przeglądania i tworzenia postaci( [UC 1.2](#) i [UC 2.1](#)).

### Faza II Listopad 2020 - Styczeń 2021

- Listopad - Zebranie uwag od klientów. Zdefiniowanie na nowo priorytetów;
- Grudzień - Umożliwienie użytkownikom tworzenia, przeglądania i filtrowania stworzonych historii ([UC 5.1](#), [5.2](#));
- Styczeń - Umożliwienie użytkownikom tworzenia i przeglądania manipulacji sesją RPG ([UC 3.7](#)), oddanie aplikacji, przeprowadzenie testów.