




ZARZĄDZANIE PROJEKTEM INFORMATYCZNYM

Aplikacja dla pracowników służby zdrowia
pomagająca raportować stan pacjentów

Spis treści

Karta projektu.....	3
Opis i cel projektu	3
Kierownik projektu i jego zadania	3
Pozostali interesariusze i ich zadania	4
Senior Backend Developer	4
Backend Developer	5
Junior Backend Developer	6
Test Developer	7
Web Application Security Specialist.....	8
Front-end Developer	9
UI Developer	10
Scrum Master	11
Wstępny zakres projektu	11
Harmonogram (kamienie milowe)	12
Etapy tworzenia aplikacji	12
Analiza i specyfikacja wymagań.....	12
Stworzenie planu projektu.....	13
Tworzenie aplikacji	13
Testy aplikacji	13
Wdrożenie aplikacji i szkolenie użytkowników.....	14
Analiza ryzyka.....	14
Niechęć pracowników służby zdrowia do nowych narzędzi	14
Przestarzałe urządzenia, które nie będą w stanie obsługiwać aplikacji	14
Stworzenie alternatywnej aplikacji przez konkurencję	14
Założenia	15
Ścieżki komunikacji i raportowanie	15
Raporty	16
Przegląd zasobów	16
Przegląd kosztów.....	17
Przepływ gotówkowy.....	19
Odchylenie kosztowe zadań	20
Przegląd kosztów zadań po realizacji pierwszego etapu.....	21

Diagram Ganta	22
Ścieżka krytyczna.....	23
Metoda szacowania kosztów w projekcie	23

	Karta projektu
Nazwa projektu	Aplikacja webowa do raportowania stanu pacjentów
Autor	Oliwia Strzelec
Data	1.11.2018r.
Sponsor	Ministerstwo Zdrowia
Opis projektu	Aplikacja dla pracowników specjalistycznej służby ochrony zdrowia psychicznego pomagająca raportować i monitorować stan pacjentów
Kierownik projektu	Jan Kowalski
Beneficjent	Szpitala kliniczne zajmujące się leczeniem oraz rehabilitacją zaburzeń i chorób psychicznych
Budżet	1 900 000 PLN
Planowana data rozpoczęcia	Wtorek, 1.01.2019r.
Planowana data zakończenia	Poniedziałek, 31.08.2020r.

Opis i cel projektu

Przedmiotem projektu jest aplikacja webowa dostępna na różne platformy dla pracowników służby zdrowia szczególnie związanych z leczeniem i rehabilitacją pacjentów z chorobami psychicznymi. Zadaniem aplikacji jest ułatwienie raportowania stanu pacjentów i wymiany informacji pomiędzy różnymi pracownikami, a w konsekwencji usprawnienie ich pracy oraz zwiększenie interakcji z pacjentami skutkujące poprawą ich zdrowia.

Kierownik projektu i jego zadania

Kierownik projektu wchodzi w interakcje z klientami, żeby zrozumieć wymagania biznesowe, oraz programistami w celu dostarczania początkowych modeli projektów, które należy zbudować. We współpracy z Senior Backend Developerem projektuje cały system w oparciu o otrzymane wymagania. W późniejszych etapach dokonuje ciągłej ewaluacji

produktu, tworzy plany projektowe i standardy dla projektu. Zapewnia pełne wsparcie techniczne projektu od momentu powstania, poprzez wydanie produktu, aż po udoskonalenia. Uczestniczy w konsultacjach z pracownikami służby zdrowia.

Pozostali interesariusze i ich zadania

Senior Backend Developer

Do podstawowych zadań Senior Backend Developera należy uczestniczenie w fazie projektowania aplikacji. Senior Backend Developer dokonuje wyboru architektury systemu i poszczególnych jego składników na wysokim poziomie, wyboru technologii implementacji każdego komponentu i połączeń między komponentami.

Ponadto Senior Backend Developer stale sprawdza kod, aby zapewnić jakość projektu i unikać złożoności. Do jego zadań należy także realizacja projektu, tworzenie oprogramowania aplikacji i nadzór nad pozostałymi programistami.

Wynagrodzenie	20 000 PLN	Doświadczenie	7 lat
Wymagania			
Java 8	★ ★ ★	Hibernate	★ ★ ★
Jenkins	★ ★ ★	SQL	★ ★ ★
Git	★ ★ ★	Maven	★ ★ ☆
Spring	★ ★ ★	Groovy	★ ★ ☆
Jira	★ ★ ★	Angular	★ ☆ ☆
JavaScript	★ ☆ ☆	Cloud Computing Services	★ ☆ ☆
Zadania			
Wdrażanie nowych rozwiązań		Projektowanie architektury aplikacji	
Rozwój bazy danych		Sprawdzanie jakości kodu	
Optymalizacja zapytań SQL			







Backend Developer

Backend Developer jest odpowiedzialny za logikę aplikacji po stronie serwera i integrację pracy programistów front-endowych. Definiuje i utrzymuje centralną bazę danych oraz zapewnia wysoką wydajność i szybkości reagowania na żądania z front-endu. Do zadań Backend Developera należy także pisanie serwisów internetowych i interfejsów API, które umożliwiają zarządzanie wymianą danych między serwerem a użytkownikami.

Wynagrodzenie	9 000 PLN	Doświadczenie	3 lata
Wymagania			
Java 8	★ ★ ★	Hibernate	★ ★ ★
SQL	★ ★ ★	Spring	★ ★ ★
Git	★ ★ ★	REST	★ ★ ☆
Jira	★ ☆ ☆	Maven	★ ★ ☆
Zadania			
Tworzenie oprogramowania	Pisanie zapytań SQL		
Pisanie testów jednostkowych	Wykonywanie testów manualnych		
Optymalizacja kodu aplikacji			

Junior Backend Developer

Junior Backend Developer pisze logikę aplikacji i testy jednostkowe. Implementuje nowe funkcjonalności zgodnie z dostarczonymi wymaganiami.

Wynagrodzenie	4 000 PLN	Doświadczenie	1 rok
Wymagania			
Java 8		Hibernate	
SQL		Spring	
Git		Maven	
Zadania			
Tworzenie oprogramowania		Wykonywanie testów manualnych	
Pisanie testów jednostkowych			
Pisanie zapytań SQL			

Test Developer

Test Developer projektuje i rozwija wszystkie aspekty automatycznego testowania wraz z tworzeniem, wykonywaniem i konserwacją zautomatyzowanych skryptów. Monitoruje pokrycie testowe i dostosowywać przypadki testowe w razie potrzeby, aby zapewnić maksymalną integrację. Mierzy skuteczność testowania i zgłasza wszelkie napotkane problemy.

Wynagrodzenie	13 000 PLN	Doświadczenie	5 lat
Wymagania			
Java 8	★ ★ ★	TDD	★ ★ ★
Git	★ ★ ★	SQL	★ ★ ★
Selenium	★ ★ ☆	Maven	★ ★ ☆
REST	★ ★ ☆	Jenkins	★ ☆ ☆
Jira	★ ☆ ☆	Cloud Computing Services	★ ☆ ☆
Zadania			
Planowanie i wykonywanie testów	Automatyzowanie testów		
Tworzenie oprogramowania	Rekonfiguracja i zestawianie środowiska testowego		
Współpraca z zespołem produktowym przy analizie i naprawianiu błędów	Tworzenie testów integracyjnych		
Nadzór nad jakością	Wykonywanie testów manualnych		

Web Application Security Specialist

Specjalista ds. bezpieczeństwa aplikacji dokonuje oceny bezpieczeństwa oprogramowania, przeprowadza testy bezpieczeństwa, uczestniczy w przeglądach kodu i współpracuje z programistami, aby zapewnić odpowiednie zabezpieczenia oprogramowania. Specjalista jest odpowiedzialny za kierowanie rozwojem, wdrażaniem i utrzymaniem inicjatyw w zakresie bezpieczeństwa aplikacji.

Wynagrodzenie	13 000 PLN	Doświadczenie	5 lat
Wymagania			
Java 8	★ ★ ★	Protokoły TCP/IP	★ ★ ★
Git	★ ★ ★	Spring Security	★ ★ ★
Maven	★ ★ ☆	Jira	★ ☆ ☆
Jenkins	★ ☆ ☆		
Zadania			
Wdrażanie systemu bezpieczeństwa w aplikacji	Przeprowadzanie testów bezpieczeństwa aplikacji		
Tworzenie oprogramowania			
Reagowanie na zagrożenia			

Front-end Developer

Front-end Developer programuje aplikację od strony klienta. Implementuje funkcjonalności umożliwiające odbieranie informacji od użytkownika, przetwarza je i wysyła do serwera za pomocą API wystawionego przez Backend Developerów. Front-end Developer wdraża rozwiązania zaproponowane przez UI Developera, sprawia, że interfejs użytkownika działa poprawnie i spełnia swoje funkcje. Zapewnia responsywność aplikacji.

Wynagrodzenie	10 000 PLN	Doświadczenie	3 lata
Wymagania			
JavaScript	★ ★ ★	CSS3	★ ★ ★
Java	★ ★ ☆	React	★ ★ ☆
HTML 5	★ ★ ★	Webpack	★ ★ ☆
Sass	★ ★ ☆	TypeScript	★ ★ ☆
Zadania			
Tworzenie oprogramowania	Obsługiwanie zapytań użytkowników		
Stylowanie aplikacji	Tworzenie aplikacji na podstawie projektów graficznych		
Tworzenie zapytań do backendu			

UI Developer

UI Developer projektuje interfejs użytkownika, czyli każdy szczegół, który użytkownik może kontrolować lub współdziałać z nim (ekran, klawiatura, mysz), na różnych urządzeniach, np. komputerach stacjonarnych czy tabletach. Projektuje grafiki wektorowe skalowalne na różnych urządzeniach oraz atrakcyjny i wygodny interfejs aplikacji w oparciu o wymagania projektowe.

Wynagrodzenie	7 000 PLN	Doświadczenie	3 lat
Wymagania			
Axure RP	★ ★ ★	CSS3	★ ★ ★
JavaScript	★ ★ ☆	Adobe	★ ★ ☆
HTML 5	★ ★ ★	Sass	★ ★ ☆
Zadania			
Projektowanie interfejsu graficznego aplikacji	Pisanie oprogramowania		
Tworzenie grafik koncepcyjnych	Stylowanie aplikacji		
Tworzenie elementów interfejsu graficznego			

Scrum Master

Scrum Master dba o przestrzeganie zasad ustalonych przez zespół, zapewnienie mu wsparcia, usuwanie przeszkód, rozwiązywanie codziennych problemów oraz zapewnianie odpowiednich warunków działania. Pracuje nad zwinnością, transparentnością i skuteczną komunikacją i samoorganizacją w zespole. Codziennie organizuje 15-minutowe spotkania, tzw. „daily scrum”, co dwa tygodnie „sprint review”, dzięki czemu programiści mają okazję spotkać się z Kierownikiem projektu i informować go o zakończonych etapach pracy lub napotkanych problemach.

Wynagrodzenie	9 000 PLN	Doświadczenie	3 lata
Wymagania			
Jira	★ ★ ★	Java	★ ★ ☆
Confluence	★ ★ ★	Git	★ ★ ☆
Slack	★ ☆ ☆	Scrum	★ ★ ★
Agile	★ ★ ★		
Zadania			
Organizowanie spotkań	Upewnianie się, że deweloperzy rozumieją swoje zadania		
Usprawnianie pracy zespołu	Dbanie o terminową realizację zadań		
Planowanie sprintów			

Wstępny zakres projektu

Wstępny zakres projektu obejmuje stworzenie w pełni funkcjonalnej aplikacji do raportowania stanu pacjentów, z różnymi profilami użytkowników dla odpowiednich służb zdrowia, np. pielęgniarzy, psychiatrów, psychologów. Ze stroną domową, medyczną bazą danych, możliwością zarządzania wspólnym kalendarzem, planowania terapii, sporządzania statystyk i prowadzenia szczegółowej dokumentacji medycznej dla pacjentów.

Harmonogram (kamienie milowe)

Poniżej przedstawiono harmonogram uwzględniający kamienie milowe. Szczegółowy harmonogram wraz z WBS załączono na końcu Projektu.

Nazwa zadania	Czas trwania	Rozpoczęcie	Zakończenie
1. Analiza i specyfikacja wymagań	113 dn	śro, 02.01.19	czw, 13.06.19
1.1. Określenie wymagań	62 dn	śro, 02.01.19	czw, 28.03.19
1.1.1. Tworzenie zespołu	12 dn	śro, 02.01.19	czw, 17.01.19
1.1.2. Konsultacje z udziałem pracowników medycznych	32 dn	pią, 18.01.19	pon, 04.03.19
1.1.3. Wybór metodyki zarządzania projektem	18 dn	wto, 05.03.19	czw, 28.03.19
1.2. Określenie celów projektowych	51 dn	pią, 29.03.19	czw, 13.06.19
2. Stworzenie planu projektu	22 dn	pią, 14.06.19	wto, 16.07.19
2.1. Planowanie	22 dn	pią, 14.06.19	wto, 16.07.19
2.1.1. Funkcjonalne	20 dn	pią, 14.06.19	pią, 12.07.19
2.1.2. Graficzne	22 dn	pią, 14.06.19	wto, 16.07.19
2.1.3. Techniczne	20 dn	pią, 14.06.19	pią, 12.07.19
3. Tworzenie aplikacji	368 dn	wto, 01.01.19	pon, 22.06.20
3.1. Etap 1	163 dn	wto, 01.01.19	pon, 26.08.19
3.2. Etap 2	42 dn	pią, 23.08.19	pon, 21.10.19
3.3. Etap 3	37 dn	pią, 04.10.19	śro, 27.11.19
3.4. Etap 4	39 dn	pią, 15.11.19	śro, 15.01.20
3.5. Etap 5	49 dn	czw, 02.01.20	śro, 11.03.20
3.6. Etap 6	70 dn	czw, 12.03.20	pon, 22.06.20
4. Wdrożenie aplikacji	50 dn	wto, 23.06.20	pon, 31.08.20
4.1. Recenzja poszerzona	40 dn	wto, 23.06.20	pon, 17.08.20
4.2. Instalacja na serwerze	5 dn	wto, 18.08.20	pon, 24.08.20
4.3. Szkolenie użytkowników	10 dn	wto, 18.08.20	pon, 31.08.20
4.4. Sporządzenie dokumentacji	5 dn	wto, 18.08.20	pon, 24.08.20

Etapy tworzenia aplikacji

Analiza i specyfikacja wymagań

Pierwszym etapem tworzenia aplikacji będzie poznanie oczekiwań Klienta oraz zaproponowanie własnych, dostosowanych do nich rozwiązań. W celu dogłębnego zrozumienia potrzeb przyszłych użytkowników odbędą się konsultacje z udziałem pracowników medycznych oraz Kierownika Projektu, Senior Backend Developera i UI Developera. Konsultacje mają za zadanie wypracowanie kompromisu pomiędzy oczekiwaniami klienta, a rozwiązaniem możliwym do dostarczenia przy założonym czasie i budżecie. Etap skończy się sporządzeniem dokumentu specyfikacji zawierającego wszystkie funkcjonalności, które ma dostarczyć gotowy produkt.

Stworzenie planu projektu

Kiedy rodzaj i zadania aplikacji będą już znane, rozpocznie się drugi etap projektu, czyli poszukiwanie najlepszych rozwiązań spełniających wymagania. Przeprowadzona zostanie analiza rynku i dostępnych technologii. Na tym etapie zespół stworzy konkretny plan działań, rozpisujący implementację aplikacji krok po kroku z uwzględnieniem interwałów czasowych. Etap skończy się dostarczeniem planu projektu do Klienta, dzięki czemu uzyska się pewność, że ewaluacja postępów będzie następowała w sposób zaakceptowany przez zespół.

Tworzenie aplikacji

Tworzenie aplikacji składa się z 5 zasadniczych etapów przeznaczonych na implementację nowych funkcjonalności oraz etapu 6 przeznaczonego wyłącznie na testowanie aplikacji oraz naprawę defektów. Każdy z etapów implementacji podzielony jest na część front-endową, back-endową, testową i część odpowiedzialną za wdrażanie systemu bezpieczeństwa.

Testy aplikacji

Aplikacja testowana jest na każdym etapie dostarczania funkcjonalności. Cały system testowany jest pod względem poprawności i wydajności. Sprawdza się, czy wszystkie krytyczne przypadki użycia są spełnione. Testy jednostkowe i testy integracyjne są wykonywane na każdym etapie powstawania aplikacji, w celu sprawdzenia czy nowe funkcjonalności nie wprowadzają defektów.

Podczas każdego sprintu należy przetestować aplikację na różnych płaszczyznach. Przeprowadza się testy:

- funkcjonalności – sprawdzają, czy wszystkie na dany etap funkcjonalności działają,
- użyteczności – sprawdzają, czy funkcjonalności są proste do zrozumienia i intuicyjne dla zwykłego użytkownika,
- wyglądu – czy po wprowadzeniu nowych funkcjonalności wszystkie widoki wyświetlają się poprawnie,
- wydajności – szczególny nacisk na testowanie wydajności kładziony jest w późniejszych sprintach, sprawdzają responsywność aplikacji, szybkość wykonywania operacji, wykorzystanie zasobów,
- regresyjne – testowanie działania funkcjonalności z poprzednich sprintów,
- uwzględniające różne platformy i urządzenia – na wielu rozmiarach ekranów, na wielu wersjach systemów, na które aplikacja jest przeznaczona,

System bez błędów może zostać przekazany użytkownikom w celu sprawdzenia, czy aplikacja spełnia oczekiwania użytkowników.

Wdrożenie aplikacji i szkolenie użytkowników

Aplikacja, która pomyślnie przeszła testy specjalistów, przechodzi ostatni etap testowania, tzw. recenzję poszerzoną. Aplikacja w fazie beta zostaje udostępniona ograniczonej liczbie potencjalnych użytkowników końcowych, którzy będą raportować działanie aplikacji. Aplikacja, która pomyślnie przejdzie recenzję poszerzoną i zostanie zaakceptowana przez klienta, zostaje zainstalowana na serwerze. Równolegle odbywa się szkolenie użytkowników.

Analiza ryzyka

Niechęć pracowników służby zdrowia do nowych narzędzi

Największym zagrożeniem dla aplikacji jest niechęć do nowych technologii ze strony potencjalnych użytkowników, czyli pracowników służby zdrowia. Pomimo tego, że pracownicy sektora medycznego z uwagi na charakter swojego zawodu przez całe życie muszą się rozwijać i zazwyczaj są na bieżąco z możliwościami współczesnej technologii, zdarzają się osoby, które z powodu uprzedzeń lub trudności w opanowaniu nowych umiejętności, stronią od nowoczesnych rozwiązań. Środkiem zaradczym na taką postawę będą szkolenia dla pracowników medycznych przeprowadzane na etapie wdrażania aplikacji. Na szkoleniach szczególną uwagę poświęci się osobom nie mającym wprawy w obsługiwaniu aplikacji webowych, a także osobom odpowiedzialnym za przyszłe doszkalanie nowych pracowników. Ponadto udostępnione zostaną instrukcje użytkowania aplikacji oraz szczegółowa dokumentacja.

Przestarzałe urządzenia, które nie będą w stanie obsługiwać aplikacji

Aplikacja zostanie stworzona również z myślą o starszych urządzeniach, jednak użytkowanie jej na urządzeniach o małej rozdzielczości może sprawiać trudności. Trudności w obsłudze aplikacji mogą zniechęcić użytkowników lub pogłębić uprzedzenia do nowoczesnych rozwiązań.

Stworzenie alternatywnej aplikacji przez konkurencję

Pomimo tego, że aplikacja powstaje na zlecenie Ministerstwa Zdrowia, a co za tym idzie – istnieje realne zapotrzebowanie na taki produkt – może się okazać, że na rynku dostępne będzie rozwiązanie o podobnych możliwościach, o ugruntowanej pozycji i przyjaźniejsze dla użytkownika.

Założenia

Zaangażowanie konsultantów zewnętrznych w postaci pracowników szpitali klinicznych na różnych stanowiskach: pielęgniarzy, psychologów, psychiatrów. Z uwagi na duże zróżnicowanie obowiązków wyżej wymienionych pracowników konsultacje pozwolą na kompleksowe poznanie oczekiwań względem aplikacji.

Ścieżki komunikacji i raportowanie

Zespół pracujący nad projektem znajduje się w jednej przestrzeni biurowej, co ułatwia komunikację, zwłaszcza pomiędzy programistami, których zadania łączą wzajemne zależności. Codziennie o ustalonej wcześniej – odpowiadającej ogółowi – porze Scrum Master organizował będzie daily scrum, czyli spotkanie mające na celu wyjaśnienie zespołowi, nad czym aktualnie pracują poszczególni członkowie zespołu, co osiągnęli poprzedniego dnia, oraz jakie trudności napotkali. Dzięki temu łatwo jest nadzorować, na jakim etapie znajduje się projekt, i czy dostarczanie produktu ze zwłoką przez jednych uczestników zespołu, wpływa na pozostałych. W daily scrum nierzadko uczestniczy także Kierownik projektu, dzięki czemu posiada najbardziej bieżące informacje o stanie projektu oraz może w razie konieczności reagować i wpływać na zmianę priorytetów. Podobne zadanie przypada Senior Backend Developerowi oraz Scrum Masterowi, którzy reagują i pomagają w momencie, kiedy członek zespołu napotka trudność, która w znaczący sposób opóźnia jego pracę.

Najważniejszymi spotkaniami, które mają za zadanie koordynować pracę całego zespołu są sprint review, odbywające się po zakończonym sprincie – czyli dwutygodniowym etapie pracy. Podczas sprint review członkowie zespołu informują o swoich postępach i stopniu realizacji zadań przeznaczonych na aktualny okres.

Wszystkie aktywności podejmowane przez członków zespołu i trudności, które napotykają, są jawne. Postępy najmłodszych programistów monitorowane są przez bardziej doświadczonych developerów. Do zadań regularnych programistów należy dbanie o terminową realizację zadań przez młodszych programistów, jak również raportowanie Senior Backend Developerowi oraz Kierownikowi projektu o stanie prac. Do zadań Senior Backend Developera oraz Kierownika projektu należą z kolei: nadzór i usprawnianie pracy pozostałych programistów oraz dbanie o informowanie Klientów o postępach i przekazywanie informacji pomiędzy obiema stronami.

Podobnie Scrum Master dba o obieg informacji w zespole, działając z polecenia Kierownika projektu w momencie, kiedy z uwagi na inne zobowiązania Kierownik projektu nie może samodzielnie dbać o rozwój projektu.

Raporty

Przegląd zasobów

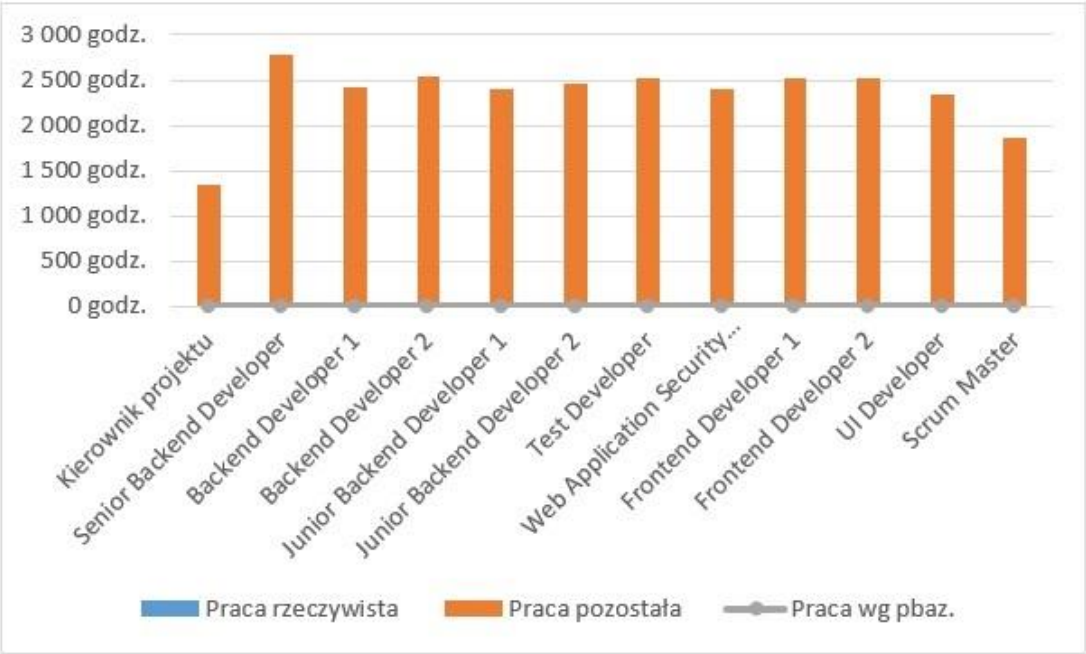
STAN ZASOBÓW

Pozostała praca wszystkich zasobów pracy.

Nazwa	Rozpoczęcie	Zakończenie	Praca pozostała
Kierownik projektu	czw, 03.01.19	pon, 24.08.20	1 344 godz.
Senior Backend Developer	śro, 02.01.19	pon, 24.08.20	2 784 godz.
Backend Developer 1	śro, 02.01.19	pon, 24.08.20	2 424 godz.
Backend Developer 2	śro, 02.01.19	pon, 24.08.20	2 548 godz.
Junior Backend Developer 1	śro, 02.01.19	pon, 24.08.20	2 400 godz.
Junior Backend Developer 2	śro, 02.01.19	pon, 24.08.20	2 456 godz.
Test Developer	śro, 02.01.19	pon, 24.08.20	2 512 godz.
Web Application Security Specialist	śro, 02.01.19	pon, 24.08.20	2 392 godz.
Frontend Developer 1	śro, 02.01.19	pon, 31.08.20	2 516 godz.
Frontend Developer 2	śro, 02.01.19	pon, 31.08.20	2 516 godz.
UI Developer	śro, 02.01.19	pon, 31.08.20	2 340 godz.
Scrum Master	śro, 02.01.19	pon, 31.08.20	1 868 godz.

STATYSTYKI ZASOBÓW

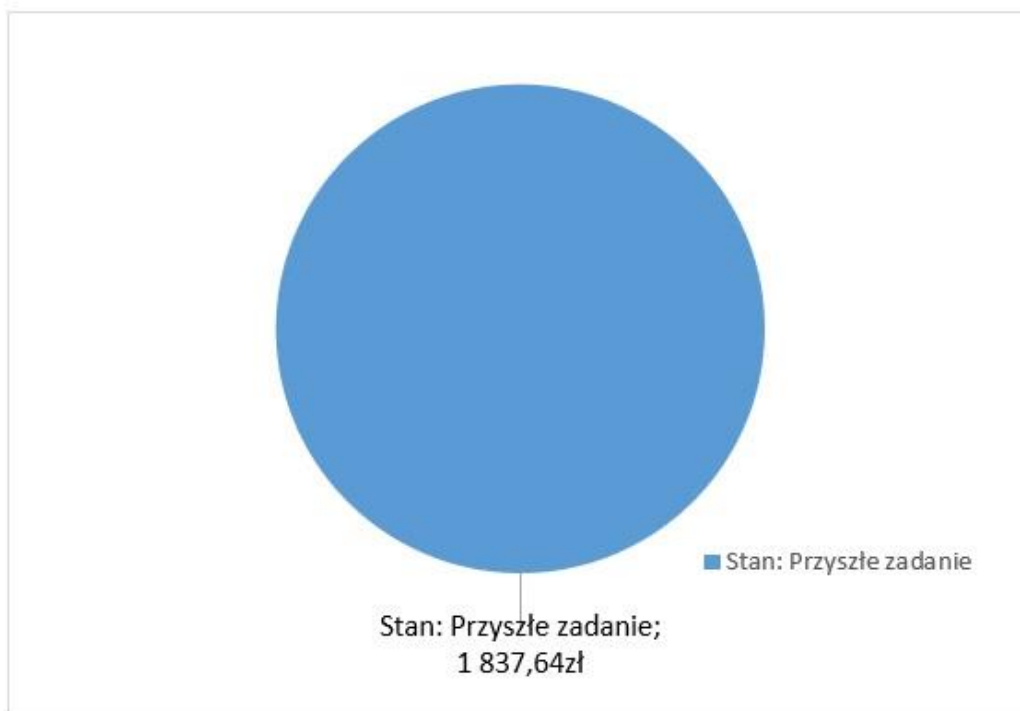
Stan pracy dla wszystkich zasobów pracy.



Przegląd kosztów

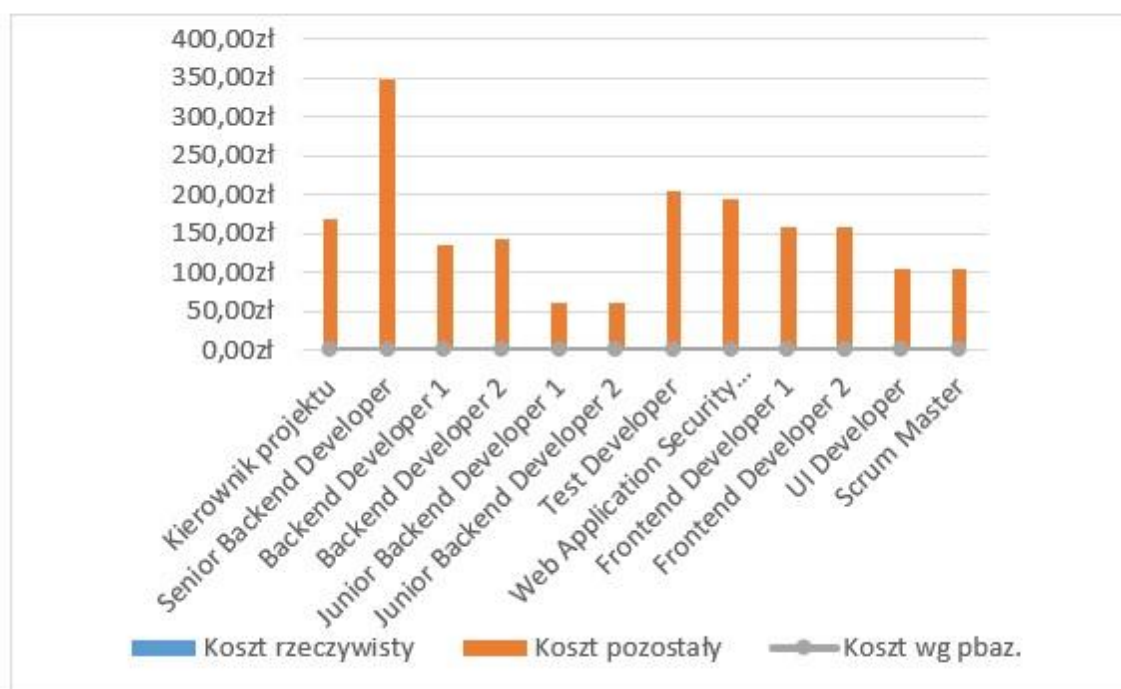
ROZKŁAD KOSZTÓW

Rozłożenie kosztów wśród zadań na podstawie ich stanu.



STAN KOSZTÓW

Stan kosztów zasobów pracy.



SZCZEGÓŁY KOSZTÓW

Szczegóły kosztów dotyczące wszystkich zasobów pracy.

Nazwa	Praca rzeczywista	Koszt rzeczywisty	Stawka standardowa
Kierownik projektu	0 godz.	0,00 zł	125,00 zł/godz.
Senior Backend Developer	0 godz.	0,00 zł	125,00 zł/godz.
Backend Developer 1	0 godz.	0,00 zł	56,00 zł/godz.
Backend Developer 2	0 godz.	0,00 zł	56,00 zł/godz.
Junior Backend Developer 1	0 godz.	0,00 zł	25,00 zł/godz.
Junior Backend Developer 2	0 godz.	0,00 zł	25,00 zł/godz.
Test Developer	0 godz.	0,00 zł	81,00 zł/godz.
Web Application Security Specialist	0 godz.	0,00 zł	81,00 zł/godz.
Frontend Developer 1	0 godz.	0,00 zł	63,00 zł/godz.
Frontend Developer 2	0 godz.	0,00 zł	63,00 zł/godz.
UI Developer	0 godz.	0,00 zł	44,00 zł/godz.
Scrum Master	0 godz.	0,00 zł	56,00 zł/godz.

SZCZEGÓŁY KOSZTÓW

Szczegóły kosztów dla wszystkich zadań najwyższego poziomu.

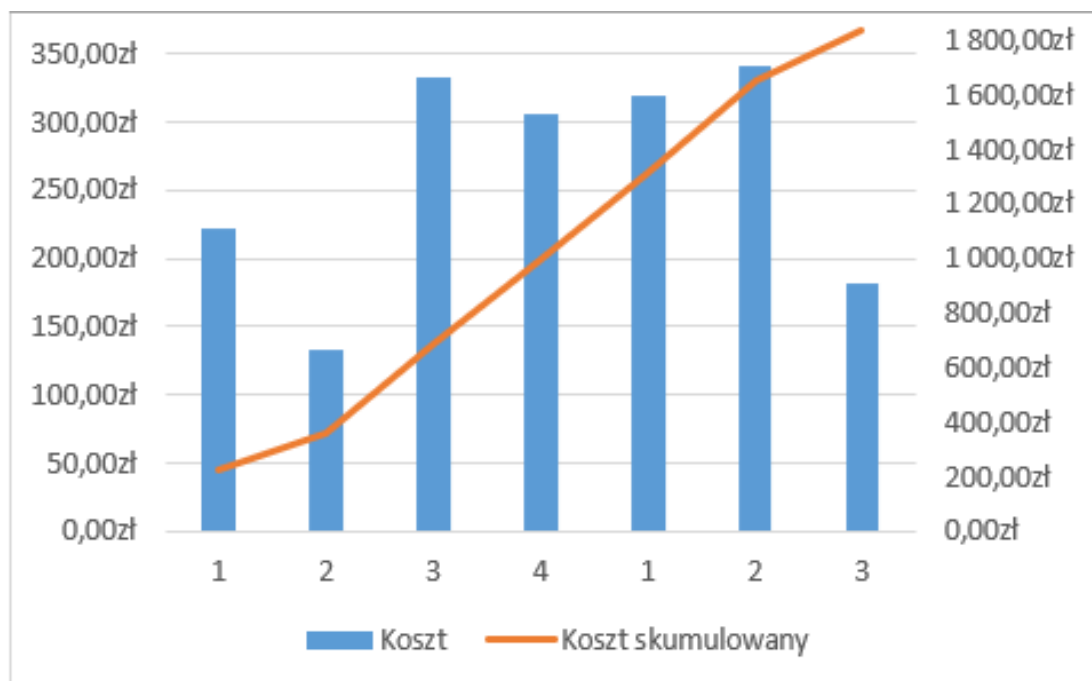
Nazwa	Koszt stały	Koszt rzeczywisty	Koszt pozostały	Koszt	Koszt wg pbaz.	Odchylenie kosztowe
1. Analiza i specyfikacja wymagań	0,00 zł	0,00 zł	291 600,00 zł	291 600,00 zł	0,00 zł	291 600,00 zł
2. Stworzenie planu projektu	0,00 zł	0,00 zł	130 720,00 zł	130 720,00 zł	0,00 zł	130 720,00 zł
3. Tworzenie aplikacji	0,00 zł	0,00 zł	1 220 840,00 zł	1 220 840,00 zł	0,00 zł	1 220 840,00 zł
4. Wdrożenie aplikacji	0,00 zł	0,00 zł	194 480,00 zł	194 480,00 zł	0,00 zł	194 480,00 zł

STAN KOSZTÓW

Stan kosztów dla zadań najwyższego poziomu.



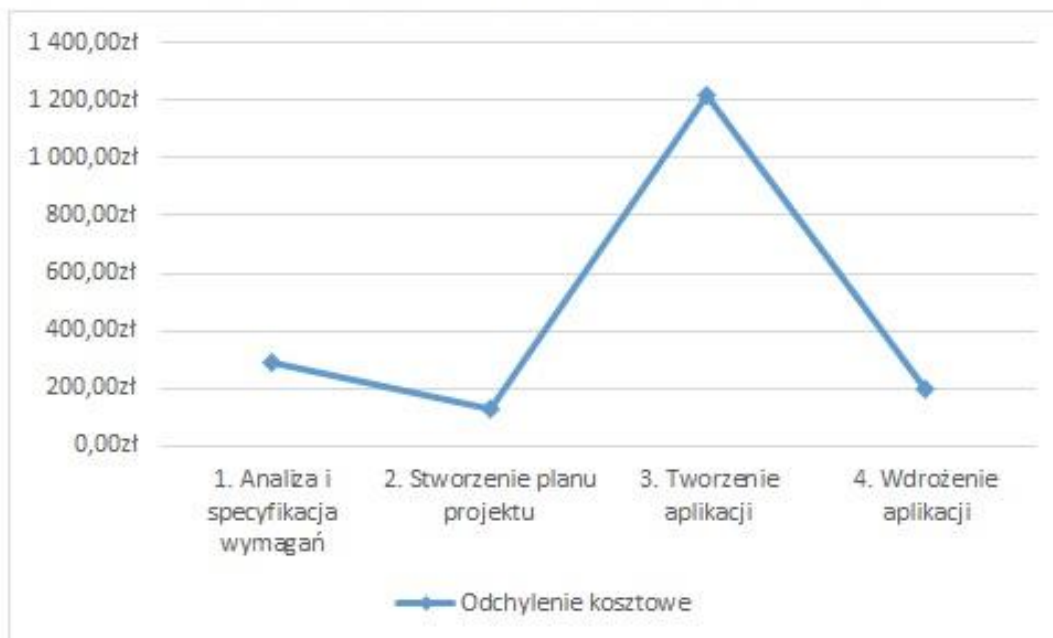
Przepływ gotówkowy



Odchylenie kosztowe zadań

ODCHYLENIE KOSZTOWE ZADAŃ

Odchylenie kosztowe dla wszystkich zadań najwyższego poziomu w projekcie.



W tabelce i na wykresie dotyczących przeglądu zasobów można zauważyć, że większość członków zespołu jest równomiernie obciążona pracą - wartość ta waha się w okolicach 2500h. W przypadku Kierownika projektu oraz Scrum Mastera ilość zaplanowanych zadań jest mniejsza, ponieważ oba stanowiska polegają bardziej na koordynowaniu, zarządzaniu ludźmi i usprawnianiu pracy całego zespołu, niż na zasadniczym tworzeniu produktu, przez co części ich zadań nie da się zaplanować i uwzględnić w harmonogramie.

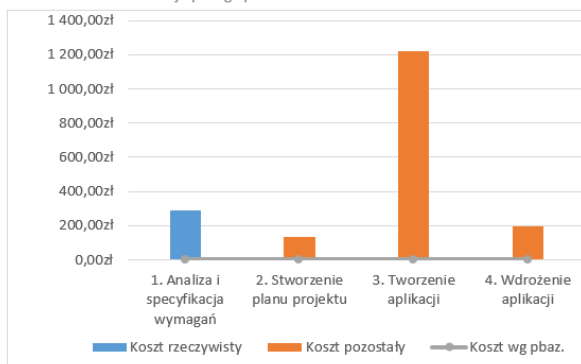
Na wykresach przepływu gotówkowego i odchylenia kosztowego zadań widać, że prawie cały koszt przedsięwzięcia skumulowany jest w części odpowiedzialnej za tworzenia aplikacji.

Przegląd kosztów zadań po realizacji pierwszego etapu

Koszt rzeczywisty	Koszt wg pbaz.	Koszt pozostały	Odchylenie kosztowe
291 600,00 zł	0,00 zł	1 546 040,00 zł	1 837 640,00 zł

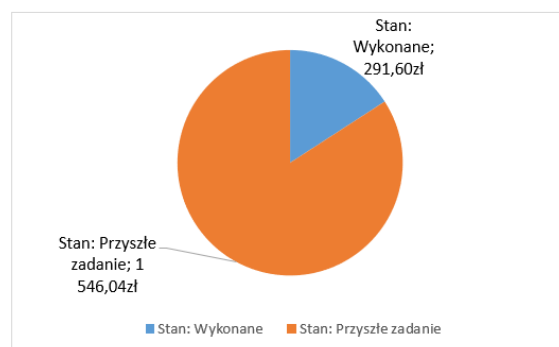
STAN KOSZTÓW

Stan kosztów dla zadań najwyższego poziomu.



ROZKŁAD KOSZTÓW

Rozłożenie kosztów wśród zadań na podstawie ich stanu.



STAN KOSZTÓW

Stan kosztów zasobów pracy.

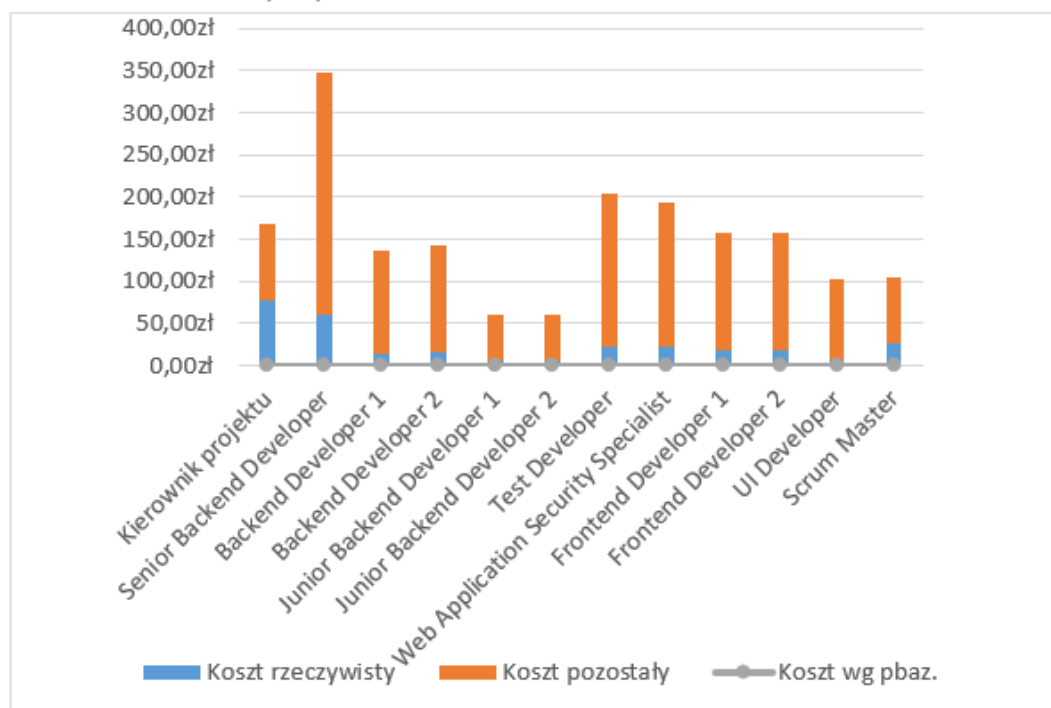
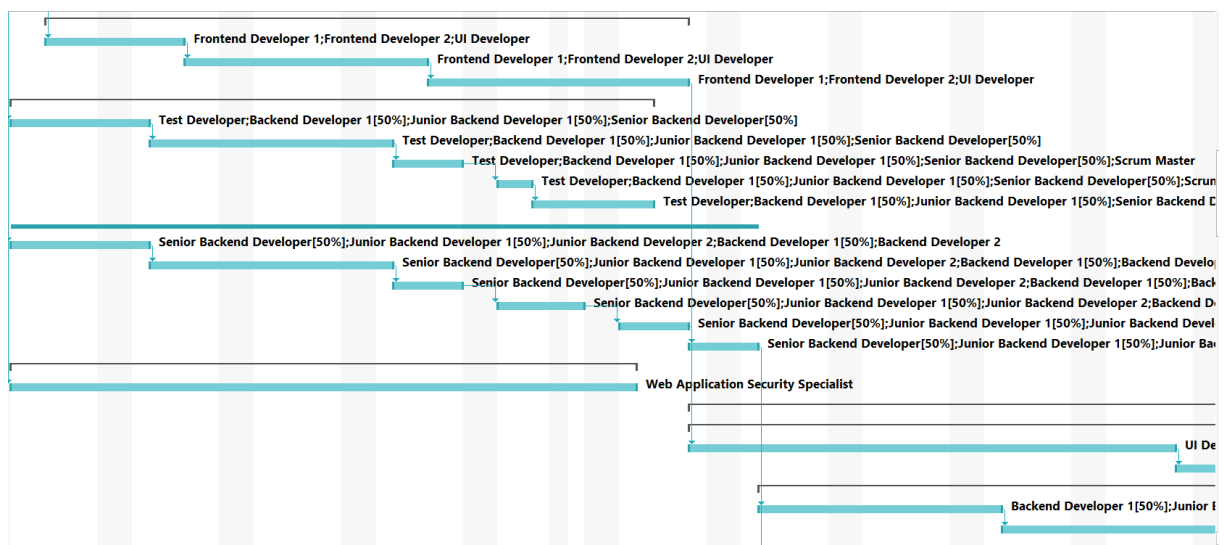
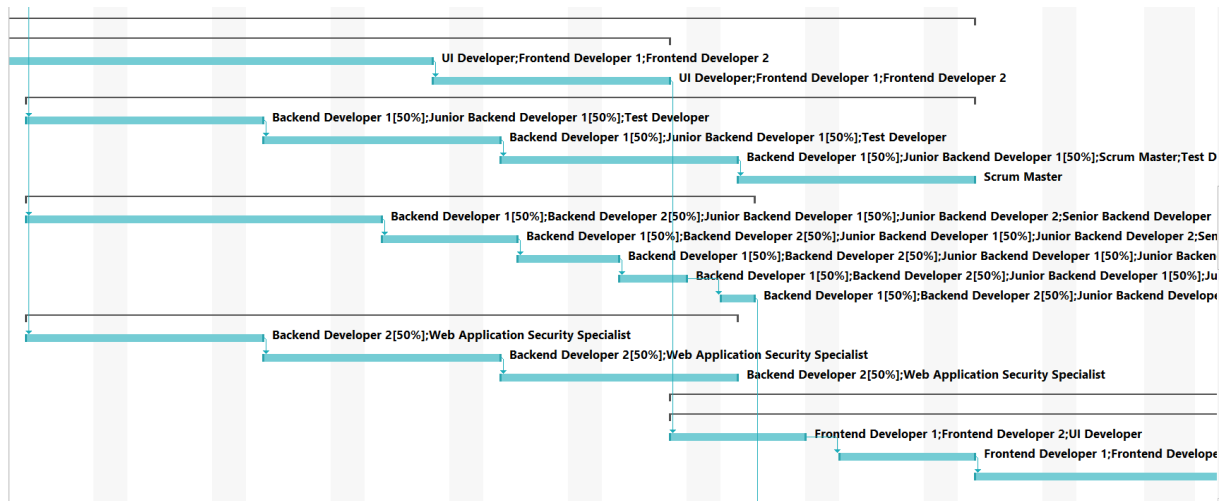


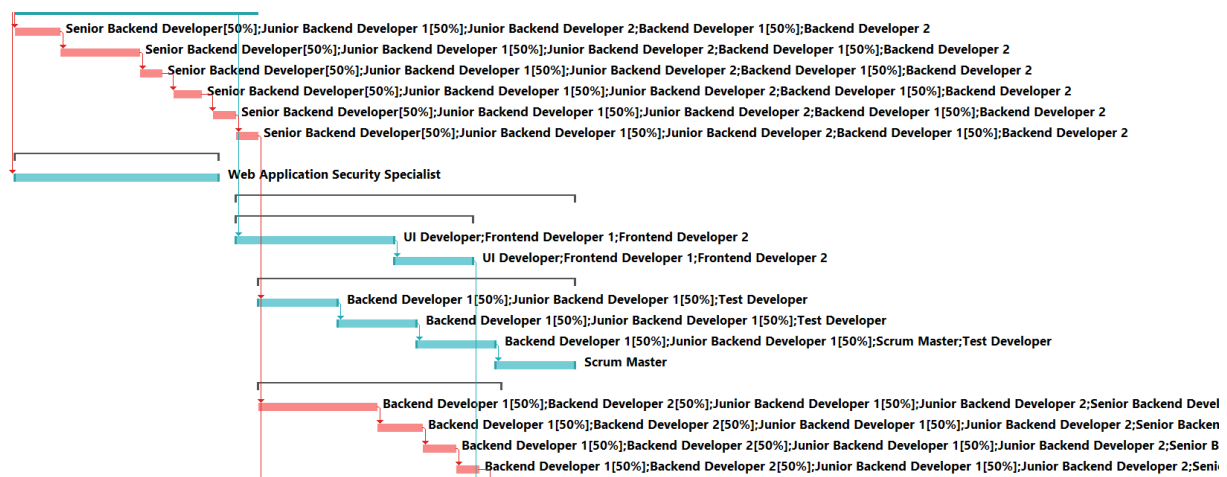
Diagram Gantt

Poniżej przedstawiono wycinki z diagramu Gantta – reprezentującego podział projektu na zadania i rozmieszczenie ich w czasie. Na wykresie można zauważyć, że niektóre zadania nie mogą się rozpocząć przed zakończeniem innych, podczas gdy pozostałe mogą przebiegać równoległe. Sytuację można zaobserwować na przykładzie implementacji front-endu, która jest uzależniona od pomyślnego zakończenia poprzedniego etapu front-endu, natomiast zadania back-endowe (oraz testowe) przebiegają niezależnie od postępów front-endu (ale są uzależnione od innych zadań back-endowych).

Ponadto na diagramie Gantta można zauważyć zasoby przypisane do poszczególnych zadań. W przypadku, kiedy jeden zasób (pracownik) w określonym czasie ma pracować nad więcej niż jednym zadaniem, jego zaangażowanie w poszczególne zadania można zmniejszyć do odpowiedniego ułamka (np. do 50 %). Wykres Gantta dla całego projektu wraz z uwzględnieniem ścieżki krytycznej zamieszczono na końcu projektu.



Ścieżka krytyczna



Na powyższym wycinku diagramu Gaanta uwzględniono ścieżkę krytyczną. Krytycznymi zadaniami dla projektu są zadania związane z implementacją back-endu, czyli z implementacją głównych funkcjonalności programu. Dzieje się tak, ponieważ zarówno zadania związane z testowaniem jak i zabezpieczaniem aplikacji są uzależnione od implementacji odpowiednich funkcjonalności.

Metoda szacowania kosztów w projekcie

Do oszacowania kosztów w projekcie posłużono się szacowaniem wstępującym czyli tzw. metodą bottom – up. Polega ona na szacowaniu poszczególnych etapów projektu, a następnie sumowaniu oszacowanych wartości w celu uzyskania łącznego kosztu projektu. Jest to metoda czasochłonna, jednak pozwala zaangażować i zmotywować do działania pracowników, którzy aktywnie uczestniczą w procesie budżetowania. Metoda ta pozwala uzyskać całkiem wiarygodne odwzorowanie przyszłych wydatków oraz uwzględnia wiedzę całego zespołu. Skuteczność metody bottom – up w dużej mierze zależy od dokładności, z jaką szacowane będą koszty poszczególnych zadań.

Przy planowaniu kosztów dobrym rozwiązaniem może się okazać wykorzystanie zarówno metody wstępującej, jak i metody przez analogię, tzw. top – down. W przypadku metody przez analogię korzysta się z doświadczenia zdobytego podczas poprzednich projektów do estymacji kosztów aktualnego projektu. Ta metoda pozwala skrócić czas budżetowania, ale angażuje tylko pracowników ze szczebla kierowniczego.

Poziom 1	Poziom 2	Poziom 3	Poziom 4
1. Analiza i specyfikacja wymagań	1.1. Określenie wymagań	1.1.1. Tworzenie zespołu	1.1.1.1. Zdefiniowanie ról w zespole
			1.1.1.2. Zidentyfikowanie wymaganych umiejętności
			1.1.1.3. Zidentyfikowanie zasobów
		1.1.2. Konsultacje z udziałem pracowników medycznych	1.1.2.1. Konsultacje z pracownikami medycznymi odnośnie ich oczekiwań
			1.1.2.2. Konsultacje w celu poznania różnych możliwości użytkowania aplikacji
			1.1.2.3. Konsultacje w celu zaprojektowania panelu użytkownika
			1.1.2.4. Konsultacje w celu zdefiniowania różnych profili użytkowników
			1.1.2.5. Badanie opinii przyszłych użytkowników
			1.1.3. Wybór metodyki zarządzania projektem
	1.1.3.2. Zdefiniowanie sposobu monitorowania problemów		
	1.1.3.3. Zdefiniowanie procedur kontroli wersji		
	1.2. Określenie celów projektów	1.2.1. Zdefiniowanie struktury projektu	1.2.1.1. Zdefiniowanie oczekiwań
			1.2.1.2. Określenie opłacalności
			1.2.1.3. Stworzenie struktury projektu
			1.2.1.4. Ustalenie celów projektowych
		1.2.2. Przygotowanie karty projektu	
		1.2.3. Uzyskanie akceptacji interesariuszy dla karty projektu	
		1.2.4. Opracowanie deklaracji zakresu projektu	
1.2.5. Przygotowanie WBS			
1.2.6.. Przygotowanie harmonogramu			
1.2.7. Uzyskanie akceptacji interesariuszy dla WBS i harmonogramu			
2. Stworzenie planu projektu	2.1. Planowanie	2.1.1. Funkcjonalne	2.1.1.1. Zdefiniowanie podstawowych funkcjonalności
			2.1.1.2. Zdefiniowanie zawartości rozszerzonej
			2.1.2. Graficzne
		2.1.2.2. Wybór technologii	
		2.1.2.3. Wybór przewodniego motywu	
		2.1.2.4. Wybór czcionki, kolorów	
		2.1.2.5. Zaprojektowanie przycisków	
		2.1.3. Techniczne	2.1.3.1. Wybór technologii i freamworków
			2.1.3.2. Wybór bazy danych
	2.1.3.3. Wybór infrastruktury hostingowej		
		2.1.3.4. Zdefiniowanie sposobów dbania o bezpieczeństwo	
		2.1.3.5. Przygotowanie specyfikacji	
		3.1. Etap 1	3.1.1. Front-end
	3.1.1.2. Implementacja strony domowej		
	3.1.1.3. Implementacja odnośników		
	3.1.2. Testowanie		3.1.2.1. Konfiguracja środowiska i narzędzi
			3.1.2.2. Implementacja freamworku testowego
			3.1.2.3. Zdefiniowanie kluczowych przypadków testowych
			3.1.2.4. Stworzenie materiałów treningowych
			3.1.2.5. Implementacja testów automatycznych
	3.1.3. Back-end - Implementacja modułu 1		3.1.3.1. Konfiguracja środowiska i narzędzi
		3.1.3.2. Implementacja funkcjonalności	
		3.1.3.3. Pisanie testów automatycznych	
		3.1.3.4. Zdefiniowanie przypadków testowych	
		3.1.3.5. Implementacja testów jednostkowych	
	3.1.3.6. Sprawdzenie pokrycia testowego		
	3.1.4. Bezpieczeństwo	3.1.4.1. Implementacja podstawowego systemu	
3.2. Etap 2		3.2.1. Front-end	3.2.1.1. Implementacja treści strony
			3.2.1.2. Implementacja panelu użytkownika
		3.2.2. Testowanie	3.2.2.1. Rozszerzanie frameworku testowego
			3.2.2.2. Implementacja testów automatycznych
			3.2.2.3. Przeprowadzenie testów regresyjnych
	3.2.2.4. Przeprowadzenie testów manualnych		
3.2.3. Back-end - Implementacja modułu 2	3.2.3.1. Implementacja funkcjonalności		
	3.2.3.2. Implementacja testów jednostkowych		
	3.2.3.3.Zdefiniowanie przypadków testowych		
	3.2.3.4. Naprawda defektów modułu 1		
	3.2.3.5. Sprawdzenie pokrycia testowego		
3.2.4. Bezpieczeństwo	3.2.4.1. Przeprowadzenie testów bezpieczeństwa		
	3.2.4.2. Implementacja zabezpieczeń przed atakami SQL		
	3.2.4.3. Implementacja zabezpieczeń przed atakami Cross-Site Scripting		

3. Tworzenie aplikacji	3.3. Etap 3	3.3.1. Front-end	3.3.1.1. Naprawa defektów etapu 1 i 2
			3.3.1.2. Implementacja motywu
			3.3.1.3. Zaprogramowanie formularzy
			3.3.1.4. Obsługa formularzy po stronie widoku
		3.3.2. Testowanie	3.3.2.1. Rozszerzanie frameworku testowego
			3.3.2.2. Implementacja testów automatycznych
			3.3.2.3. Przeprowadzenie testów regresyjnych
			3.3.2.4. Przeprowadzenie testów manualnych
		3.3.3. Back-end - Implementacja modułu 3	3.3.3.1. Implementacja funkcjonalności
			3.3.3.2. Implementacja testów jednostkowych
			3.3.3.3. Zdefiniowanie nowych przypadków testowych
			3.3.3.4. Naprawa defektów modułu 2
			3.3.3.5. Sprawdzenie pokrycia testowego i dopisanie koniecznych testów
		3.3.4. Bezpieczeństwo	3.3.4.1. Szyfrowanie aplikacji
			3.3.4.2. Zabezpieczenie na ataki MITM
			3.3.4.3. Przeprowadzenie testów bezpieczeństwa
	3.4. Etap 4	3.4.1. Front-end	3.4.1.1. Stylowanie widoków
			3.4.1.2. Naprawa defektów etapu 3
			3.4.1.3. Wdrożenie responsywności
		3.4.2. Testowanie	3.4.2.1. Implementacja testów automatycznych
			3.4.2.2. Przeprowadzenie testów regresyjnych
			3.4.2.3. Przeprowadzenie testów manualnych
		3.4.3. Back-end - Implementacja modułu 4	3.4.3.1. Implementacja funkcjonalności
			3.4.3.2. Implementacja testów jednostkowych
			3.4.3.3. Zdefiniowanie nowych przypadków testowych
			3.4.3.4. Naprawa defektów modułu 3
		3.4.4. Bezpieczeństwo	3.4.4.1. Przeprowadzenie testów bezpieczeństwa
			3.4.4.2. Przeprowadzenie ataku na aplikację w celu znalezienia słabości
			3.4.4.3. Naprawa odkrytych defektów
	3.5. Etap 5	3.5.1. Front-end	3.5.1.1. Stylowanie widoków
			3.5.1.2. Wdrożenie responsywności
			3.5.1.3. Naprawa defektów etapu 4
			3.5.1.4. Przeprowadzanie testów manualnych
		3.5.2. Testowanie	3.5.2.1. Implementacja testów automatycznych
			3.5.2.2. Przeprowadzenie testów regresyjnych
			3.5.2.3. Przeprowadzenie testów manualnych
		3.5.3. Back-end - Implementacja modułu 5	3.5.3.1. Implementacja funkcjonalności
			3.5.3.2. Implementacja testów jednostkowych
			3.5.3.3. Zdefiniowanie nowych przypadków testowych
			3.5.3.4. Naprawa defektów modułu 4
	3.6. Etap 6	3.6.1. Testowanie	3.5.4.1. Przeprowadzenie testów bezpieczeństwa
			3.6.1.1. Przeprowadzenie testów regresyjnych
			3.6.1.2. Przeprowadzenie testów manualnych
			3.6.1.3. Przeprowadzanie testów automatycznych
			3.6.1.4. Przeprowadzenie testów bezpieczeństwa
			3.6.1.5. Przeprowadzenie testów funkcjonalności
		3.6.2. Naprawa defektów	3.6.2.1. Naprawa defektów po stronie back-endu
			3.6.2.2. Naprawa defektów po stronie front-endu
			3.6.2.3. Naprawa defektów po stronie bezpieczeństwa aplikacji
4. Wdrożenie aplikacji	4.1. Recenzja poszerzona	4.1.1. Udostępnienie aplikacji w fazie beta	
		4.1.2. Analiza raportów fazy beta	
		4.1.3. Reprodukacja defektów	
		4.1.4. Naprawa błędów aplikacji	
	4.2. Instalacja na serwerze		
	4.3. Szkolenie użytkowników		
	4.4. Sporządzenie dokumentacji		

Nazwa zadania	Czas trwania	Rozpoczęcie	Zakończenie
1. Analiza i specyfikacja wymagań	113 dn	śro, 02.01.19	czw, 13.06.19
1.1. Określenie wymagań	62 dn	śro, 02.01.19	czw, 28.03.19
1.1.1. Tworzenie zespołu	12 dn	śro, 02.01.19	czw, 17.01.19
1.1.1.1. Zdefiniowanie ról w zespole	4 dn	śro, 02.01.19	pon, 07.01.19
1.1.1.2. Zidentyfikowanie wymaganych umiejętności	4 dn	wto, 08.01.19	pią, 11.01.19
1.1.1.3. Zidentyfikowanie zasobów	4 dn	pon, 14.01.19	czw, 17.01.19
1.1.2. Konsultacje z udziałem pracowników medycznych	32 dn	pią, 18.01.19	pon, 04.03.19
1.1.2.1. Konsultacje z pracownikami medycznymi odnośnie ich oczekiwań	6 dn	pią, 18.01.19	pią, 25.01.19
1.1.2.2. Konsultacje w celu poznania różnych możliwości użytkowania aplikacji	6 dn	pon, 28.01.19	pon, 04.02.19
1.1.2.3. Konsultacje w celu zaprojektowania panelu użytkownika	6 dn	wto, 05.02.19	wto, 12.02.19
1.1.2.4. Konsultacje w celu zdefiniowania różnych profili użytkowników	6 dn	śro, 13.02.19	śro, 20.02.19
1.1.2.5. Badanie opinii przyszłych użytkowników	8 dn	czw, 21.02.19	pon, 04.03.19
1.1.3. Wybór metodyki zarządzania projektem	18 dn	wto, 05.03.19	czw, 28.03.19
1.1.3.1. Zdefiniowanie sposobu śledzenia postępów	8 dn	wto, 05.03.19	czw, 14.03.19
1.1.3.2. Zdefiniowanie sposobu monitorowania problemów	6 dn	pią, 15.03.19	pią, 22.03.19
1.1.3.3. Zdefiniowanie procedur kontroli wersji	4 dn	pon, 25.03.19	czw, 28.03.19
1.2. Określenie celów projektowych	51 dn	pią, 29.03.19	czw, 13.06.19
1.2.1. Zdefiniowanie struktury projektu	33 dn	pią, 29.03.19	pon, 20.05.19
1.2.1.1. Zdefiniowanie oczekiwań	10 dn	pią, 29.03.19	czw, 11.04.19
1.2.1.2. Określenie opłacalności	4 dn	pią, 12.04.19	śro, 17.04.19
1.2.1.3. Stworzenie struktury projektu	10 dn	czw, 18.04.19	śro, 08.05.19
1.2.1.4. Ustalenie celów projektowych	6 dn	czw, 09.05.19	pon, 20.05.19
1.2.2. Przygotowanie karty projektu	3 dn	wto, 21.05.19	pią, 24.05.19
1.2.3. Uzyskanie akceptacji interesariuszy dla karty projektu	2 dn	pon, 27.05.19	wto, 28.05.19
1.2.4. Opracowanie deklaracji zakresu projektu	2 dn	śro, 29.05.19	czw, 30.05.19
1.2.5. Przygotowanie WBS	2 dn	pią, 31.05.19	pon, 03.06.19
1.2.6.. Przygotowanie harmonogramu	3 dn	wto, 04.06.19	pią, 07.06.19
1.2.7. Uzyskanie akceptacji interesariuszy dla WBS i harmonogramu	3 dn	pon, 10.06.19	czw, 13.06.19
2. Stworzenie planu projektu	22 dn	pią, 14.06.19	wto, 16.07.19
2.1. Planowanie	22 dn	pią, 14.06.19	wto, 16.07.19
2.1.1. Funkcjonalne	20 dn	pią, 14.06.19	pią, 12.07.19
2.1.1.1. Zdefiniowanie podstawowych funkcjonalności	10 dn	pią, 14.06.19	pią, 28.06.19
2.1.1.2. Zdefiniowanie zawartości rozszerzonej	10 dn	pon, 01.07.19	pią, 12.07.19
2.1.2. Graficzne	22 dn	pią, 14.06.19	wto, 16.07.19
2.1.2.1. Przygotowanie projektu graficznego	4 dn	pią, 14.06.19	śro, 19.06.19
2.1.2.2. Wybór technologii	6 dn	pią, 21.06.19	pią, 28.06.19
2.1.2.3. Wybór przewodniego motywu	4 dn	pon, 01.07.19	czw, 04.07.19
2.1.2.4. Wybór czcionki, kolorów	2 dn	pią, 05.07.19	pon, 08.07.19
2.1.2.5. Zaprojektowanie przycisków	6 dn	wto, 09.07.19	wto, 16.07.19
2.1.3. Techniczne	20 dn	pią, 14.06.19	pią, 12.07.19
2.1.3.1. Wybór technologii i freamworków	6 dn	pią, 14.06.19	pon, 24.06.19
2.1.3.2. Wybór bazy danych	4 dn	wto, 25.06.19	pią, 28.06.19
2.1.3.3. Wybór infrastruktury hostingowej	4 dn	pon, 01.07.19	czw, 04.07.19
2.1.3.4. Zdefiniowanie sposobów dbania o bezpieczeństwo	4 dn	pią, 05.07.19	śro, 10.07.19
2.1.3.5. Przygotowanie specyfikacji	2 dn	czw, 11.07.19	pią, 12.07.19
3. Tworzenie aplikacji	368 dn	wto, 01.01.19	pon, 22.06.20
3.1. Etap 1	163 dn	wto, 01.01.19	pon, 26.08.19
3.1.1. Front-end	26 dn	śro, 17.07.19	czw, 22.08.19
3.1.1.1. Konfiguracja środowiska i narzędzi	6 dn	śro, 17.07.19	śro, 24.07.19
3.1.1.2. Implementacja strony domowej	10 dn	czw, 25.07.19	śro, 07.08.19
3.1.1.3. Implementacja odnośników	10 dn	czw, 08.08.19	czw, 22.08.19

3.1.2. Testowanie	26 dn	pon, 15.07.19	wto, 20.08.19
3.1.2.1. Konfiguracja środowiska i narzędzi	6 dn	pon, 15.07.19	pon, 22.07.19
3.1.2.2. Implementacja frameworku testowego	10 dn	wto, 23.07.19	pon, 05.08.19
3.1.2.3. Zdefiniowanie kluczowych przypadków testowych	4 dn	wto, 06.08.19	pią, 09.08.19
3.1.2.4. Stworzenie materiałów treningowych	2 dn	pon, 12.08.19	wto, 13.08.19
3.1.2.5. Implementacja testów automatycznych	4 dn	śro, 14.08.19	wto, 20.08.19
3.1.3. Back-end - Implementacja modułu 1			
3.1.3.1. Konfiguracja środowiska i narzędzi	6 dn	pon, 15.07.19	pon, 22.07.19
3.1.3.2. Implementacja funkcjonalności	10 dn	wto, 23.07.19	pon, 05.08.19
3.1.3.3. Pisanie testów automatycznych	4 dn	wto, 06.08.19	pią, 09.08.19
3.1.3.4. Zdefiniowanie przypadków testowych	4 dn	pon, 12.08.19	pią, 16.08.19
3.1.3.5. Implementacja testów jednostkowych	4 dn	pon, 19.08.19	czw, 22.08.19
3.1.3.6. Sprawdzenie pokrycia testowego	2 dn	pią, 23.08.19	pon, 26.08.19
3.1.4. Bezpieczeństwo	25 dn	pon, 15.07.19	pon, 19.08.19
3.1.4.1. Implementacja podstawowego systemu	25 dn	pon, 15.07.19	pon, 19.08.19
3.2. Etap 2	42 dn	pią, 23.08.19	pon, 21.10.19
3.2.1. Front-end	30 dn	pią, 23.08.19	czw, 03.10.19
3.2.1.1. Implementacja treści strony	20 dn	pią, 23.08.19	czw, 19.09.19
3.2.1.2. Implementacja panelu użytkownika	10 dn	pią, 20.09.19	czw, 03.10.19
3.2.2. Testowanie	40 dn	wto, 27.08.19	pon, 21.10.19
3.2.2.1. Rozszerzanie frameworku testowego	10 dn	wto, 27.08.19	pon, 09.09.19
3.2.2.2. Implementacja testów automatycznych	10 dn	wto, 10.09.19	pon, 23.09.19
3.2.2.3. Przeprowadzenie testów regresyjnych	10 dn	wto, 24.09.19	pon, 07.10.19
3.2.2.4. Przeprowadzenie testów manualnych	10 dn	wto, 08.10.19	pon, 21.10.19
3.2.3. Back-end - Implementacja modułu 2	31 dn	wto, 27.08.19	wto, 08.10.19
3.2.3.1. Implementacja funkcjonalności	15 dn	wto, 27.08.19	pon, 16.09.19
3.2.3.2. Implementacja testów jednostkowych	6 dn	wto, 17.09.19	wto, 24.09.19
3.2.3.3. Zdefiniowanie przypadków testowych	4 dn	śro, 25.09.19	pon, 30.09.19
3.2.3.4. Naprawa defektów modułu 1	4 dn	wto, 01.10.19	pią, 04.10.19
3.2.3.5. Sprawdzenie pokrycia testowego	2 dn	pon, 07.10.19	wto, 08.10.19
3.2.4. Bezpieczeństwo	30 dn	wto, 27.08.19	pon, 07.10.19
3.2.4.1. Przeprowadzenie testów bezpieczeństwa	10 dn	wto, 27.08.19	pon, 09.09.19
3.2.4.2. Implementacja zabezpieczeń przed atakami SQL	10 dn	wto, 10.09.19	pon, 23.09.19
3.2.4.3. Implementacja zabezpieczeń przed atakami Cross-Site Scripting	10 dn	wto, 24.09.19	pon, 07.10.19
3.3. Etap 3	37 dn	pią, 04.10.19	śro, 27.11.19
3.3.1. Front-end	28 dn	pią, 04.10.19	czw, 14.11.19
3.3.1.1. Naprawa defektów etapu 1 i 2	6 dn	pią, 04.10.19	pią, 11.10.19
3.3.1.2. Implementacja motywu	6 dn	pon, 14.10.19	pon, 21.10.19
3.3.1.3. Zaprogramowanie formularzy	10 dn	wto, 22.10.19	wto, 05.11.19
3.3.1.4. Obsługa formularzy po stronie widoku	6 dn	śro, 06.11.19	czw, 14.11.19
3.3.2. Testowanie	31 dn	śro, 09.10.19	pią, 22.11.19
3.3.2.1. Rozszerzanie frameworku testowego	10 dn	śro, 09.10.19	wto, 22.10.19
3.3.2.2. Implementacja testów automatycznych	10 dn	śro, 23.10.19	śro, 06.11.19
3.3.2.3. Przeprowadzenie testów regresyjnych	5 dn	czw, 07.11.19	czw, 14.11.19
3.3.2.4. Przeprowadzenie testów manualnych	6 dn	pią, 15.11.19	pią, 22.11.19
3.3.3. Back-end - Implementacja modułu 3	34 dn	śro, 09.10.19	śro, 27.11.19
3.3.3.1. Implementacja funkcjonalności	15 dn	śro, 09.10.19	wto, 29.10.19
3.3.3.2. Implementacja testów jednostkowych	6 dn	śro, 30.10.19	czw, 07.11.19
3.3.3.3. Zdefiniowanie nowych przypadków testowych	5 dn	pią, 08.11.19	pią, 15.11.19
3.3.3.4. Naprawa defektów modułu 2	6 dn	pon, 18.11.19	pon, 25.11.19
3.3.3.5. Sprawdzenie pokrycia testowego i dopisanie koniecznych testów	2 dn	wto, 26.11.19	śro, 27.11.19
3.3.4. Bezpieczeństwo	30 dn	śro, 09.10.19	czw, 21.11.19
3.3.4.1. Szyfrowanie aplikacji	10 dn	śro, 09.10.19	wto, 22.10.19
3.3.4.2. Zabezpieczenie na ataki MITM	10 dn	śro, 23.10.19	śro, 06.11.19
3.3.4.3. Przeprowadzenie testów bezpieczeństwa	10 dn	czw, 07.11.19	czw, 21.11.19

3.4. Etap 4	39 dn	pią, 15.11.19	śro, 15.01.20
3.4.1. Front-end	30 dn	pią, 15.11.19	wto, 31.12.19
3.4.1.1. Stylowanie widoków	10 dn	pią, 15.11.19	czw, 28.11.19
3.4.1.2. Naprawa defektów etapu 3	10 dn	pią, 29.11.19	czw, 12.12.19
3.4.1.3. Wdrożenie responsywności	10 dn	pią, 13.12.19	wto, 31.12.19
3.4.2. Testowanie	30 dn	czw, 28.11.19	śro, 15.01.20
3.4.2.1. Implementacja testów automatycznych	10 dn	czw, 28.11.19	śro, 11.12.19
3.4.2.2. Przeprowadzenie testów regresyjnych	10 dn	czw, 12.12.19	pon, 30.12.19
3.4.2.3. Przeprowadzenie testów manualnych	10 dn	wto, 31.12.19	śro, 15.01.20
3.4.3. Back-end - Implementacja modułu 4	30 dn	czw, 28.11.19	śro, 15.01.20
3.4.3.1. Implementacja funkcjonalności	16 dn	czw, 28.11.19	czw, 19.12.19
3.4.3.2. Implementacja testów jednostkowych	4 dn	pią, 20.12.19	pon, 30.12.19
3.4.3.3. Zdefiniowanie nowych przypadków testowych	4 dn	wto, 31.12.19	wto, 07.01.20
3.4.3.4. Naprawa defektów modułu 3	6 dn	śro, 08.01.20	śro, 15.01.20
3.4.4. Bezpieczeństwo	30 dn	czw, 28.11.19	śro, 15.01.20
3.4.4.1. Przeprowadzenie testów bezpieczeństwa	10 dn	czw, 28.11.19	śro, 11.12.19
3.4.4.2. Przeprowadzenie ataku na aplikację w celu znalezienia słabości	10 dn	czw, 12.12.19	pon, 30.12.19
3.4.4.3. Naprawa odkrytych defektów	10 dn	wto, 31.12.19	śro, 15.01.20
3.5. Etap 5	49 dn	czw, 02.01.20	śro, 11.03.20
3.5.1. Front-end	40 dn	czw, 02.01.20	czw, 27.02.20
3.5.1.1. Stylowanie widoków	10 dn	czw, 02.01.20	czw, 16.01.20
3.5.1.2. Wdrożenie responsywności	10 dn	pią, 17.01.20	czw, 30.01.20
3.5.1.3. Naprawa defektów etapu 4	10 dn	pią, 31.01.20	czw, 13.02.20
3.5.1.4. Przeprowadzanie testów manualnych	10 dn	pią, 14.02.20	czw, 27.02.20
3.5.2. Testowanie	38 dn	czw, 16.01.20	pon, 09.03.20
3.5.2.1. Implementacja testów automatycznych	18 dn	czw, 16.01.20	pon, 10.02.20
3.5.2.2. Przeprowadzenie testów regresyjnych	10 dn	wto, 11.02.20	pon, 24.02.20
3.5.2.3. Przeprowadzenie testów manualnych	10 dn	wto, 25.02.20	pon, 09.03.20
3.5.3. Back-end - Implementacja modułu 5	40 dn	czw, 16.01.20	śro, 11.03.20
3.5.3.1. Implementacja funkcjonalności	18 dn	czw, 16.01.20	pon, 10.02.20
3.5.3.2. Implementacja testów jednostkowych	6 dn	wto, 11.02.20	wto, 18.02.20
3.5.3.3. Zdefiniowanie nowych przypadków testowych	4 dn	śro, 19.02.20	pon, 24.02.20
3.5.3.4. Naprawa defektów modułu 4	12 dn	wto, 25.02.20	śro, 11.03.20
3.5.4. Bezpieczeństwo	25 dn	czw, 16.01.20	śro, 19.02.20
3.5.4.1. Przeprowadzenie testów bezpieczeństwa	25 dn	czw, 16.01.20	śro, 19.02.20
3.6. Etap 6	70 dn	czw, 12.03.20	pon, 22.06.20
3.6.1. Testowanie	50 dn	czw, 12.03.20	pią, 22.05.20
3.6.1.1. Przeprowadzenie testów regresyjnych	10 dn	czw, 12.03.20	śro, 25.03.20
3.6.1.2. Przeprowadzenie testów manualnych	10 dn	czw, 26.03.20	śro, 08.04.20
3.6.1.3. Przeprowadzanie testów automatycznych	10 dn	czw, 09.04.20	czw, 23.04.20
3.6.1.4. Przeprowadzenie testów bezpieczeństwa	10 dn	pią, 24.04.20	pią, 08.05.20
3.6.1.5. Przeprowadzenie testów funkcjonalności	10 dn	pon, 11.05.20	pią, 22.05.20
3.6.2. Naprawa defektów	20 dn	pon, 25.05.20	pon, 22.06.20
3.6.2.1. Naprawa defektów po stronie back-endu	20 dn	pon, 25.05.20	pon, 22.06.20
3.6.2.2. Naprawa defektów po stronie front-endu	20 dn	pon, 25.05.20	pon, 22.06.20
3.6.2.3. Naprawa defektów po stronie bezpieczeństwa aplikacji	20 dn	pon, 25.05.20	pon, 22.06.20
4. Wdrożenie aplikacji	50 dn	wto, 23.06.20	pon, 31.08.20
4.1. Recenzja poszerzona	40 dn	wto, 23.06.20	pon, 17.08.20
4.1.1. Udostępnienie aplikacji w fazie beta	10 dn	wto, 23.06.20	pon, 06.07.20
4.1.2. Analiza raportów fazy beta	5 dn	wto, 07.07.20	pon, 13.07.20
4.1.3. Reprodukacja defektów	10 dn	wto, 14.07.20	pon, 27.07.20
4.1.4. Naprawa błędów aplikacji	15 dn	wto, 28.07.20	pon, 17.08.20
4.2. Instalacja na serwerze	5 dn	wto, 18.08.20	pon, 24.08.20
4.3. Szkolenie użytkowników	10 dn	wto, 18.08.20	pon, 31.08.20
4.4. Sporządzenie dokumentacji	5 dn	wto, 18.08.20	pon, 24.08.20

