

# Inżynieria Oprogramowania

## II FAZA WYTWARZANIA OPROGRAMOWANIA - FAZA OPRACOWANIA -

# Hierarchia funkcji

1. Funkcje główne programu
  - 1.1. Start/Stop/Reset
    - 1.1.1. Kompilowanie programu użytkownika
    - 1.1.2. Uruchamianie programu użytkownika
    - 1.1.3. Zatrzymywanie wykonywania programu użytkownika
  - 1.2. Zarządzanie składowymi programu użytkownika
    - 1.2.1. Dodawanie składowych
      - 1.2.1.1. Wybór typu ("duszki", wyrażenia, pętle/instrukcje warunkowe/etc, interakcje(dźwięk), i/o, ruch, kontrola pisaka, zmienne)
      - 1.2.1.2. Nazywanie składowej
    - 1.2.2. Usuwanie składowych
    - 1.2.3. Przesuwanie elementów
    - 1.2.4. Łączenie w bloki
    - 1.2.5. Edycja wartości
    - 1.2.6. Edycja nazwy
    - 1.2.7. Cofnięcie ostatniej akcji
  - 1.3. Konwersja kodu obrazkowego na kod wykonywalny
    - 1.3.1. Wybór języka docelowego
    - 1.3.2. Zapis do pliku
2. Ustawienia
  - 2.1. Zarządzanie projektem
    - 2.1.1. Nazywanie projektu
    - 2.1.2. Wczytanie projektu
    - 2.1.3. Zapis projektu
  - 2.2. Kompilator
    - 2.2.1. Wybór flag kompilatora
  - 2.3. Wygląd programu
    - 2.3.1. Ustawienia skórki
    - 2.3.2. Rozmiar i rodzaj czcionek
    - 2.3.3. Kolory bloków
    - 2.3.4. Ustawienia wielkości bloków
3. Debugger
  - 3.1. Śledzenie zmiennych
  - 3.2. Zarządzanie Breakpointami
    - 3.2.1. Dodawanie Breakpointów
    - 3.2.2. Usuwanie Breakpointów
    - 3.2.3. Przesuwanie Breakpointów
  - 3.3. Edycja ustawień debuggera



## Krótki opis funkcji programu

### Funkcje główne programu

Główna funkcja, której użycie umożliwia tworzenia programów w języku ThickPic.

### Start/Stop/Reset

Odpowiada za włączanie napisanego wcześniej programu.

### Kompilowanie programu użytkownika

Odpowiada za skompilowanie programu użytkownika.

### Uruchamianie programu użytkownika

Pozwala na uruchomienie skompilowanego wcześniej programu.

### Zatrzymywanie wykonywania programu użytkownika

Umożliwia zatrzymanie aktualnie działającego programu.

### Zarządzanie składowymi programu użytkownika

Umożliwia wybór i edycję elementów składowych programu.

### Dodawanie składowych

Pozwala na wybór i nazwanie (w przypadku zmiennych) nowej składowej programu.

Wybór typu ("duszki", wyrażenia, pętle/instrukcje warunkowe/etc, interakcje(dźwięk), i/o, ruch, kontrola pisaka, zmienne). Określa typ składowej, jaką użytkownik chce dodać do programu

### Nazywanie składowej

Jeśli wybrana składowa jest zmienną, należy nadać jej nazwę.

### Usuwanie składowych

Pozwala na usunięcie składowej z kodu tworzonego programu.

### Przesuwanie elementów

Umożliwia przesuwanie/zmianę kolejności stworzonych wcześniej elementów składowych.

### Łączenie w bloki

Pozwala na łączenie niektórych ciągów instrukcji w bloki, w celu poprawy przejrzystości kodu.

### Edycja wartości

Zmienia wartość utworzonej wcześniej składowej programu.

### Edycja nazwy

Pozwala zmienić nazwę utworzonej wcześniej zmiennej.

### Cofnięcie ostatniej akcji

Pozwala na cofnięcie ostatniej zmiany w programie użytkownika.

### Konwersja kodu obrazkowego na kod wykonywalny

Umożliwia zmianę programu w języku ThickPic na jeden z języków dostępnych w programie.

### Wybór języka docelowego

Pozwala na wybór języka.

### Zapis do pliku

Pozwala na zapisanie wyniku konwersji do zewnętrznego pliku.

### Ustawienia

Funkcja odpowiedzialna za zmianę ustawień programu.

### Zarządzanie projektem

Umożliwia tworzenie/wczytanie/zapisanie projektu.

### Nazwa projektu

Pozwala zmienić nazwę stworzonego już projektu.

### Wczytanie projektu

Odpowiada za wczytanie projektu z pliku.

### Zapis projektu

Pozwala na zapis projektu do pliku.

### Kompilator

Umożliwia wybór flag oraz innych opcji, pod którymi będzie działał kompilator. Także flag docelowego języka konwersji.

## Wybór flag kompilatora

Zmienia flagi kompilatora i sprawdza ich kompatybilność.

## Wygląd programu

Pozwala na zmianę szaty graficznej programu.

## Ustawienia skórki

Dostosowuje kolor tła oraz paneli menu.

## Rozmiar i rodzaj czcionek

Odpowiada za zmianę czcionki, którą napisany jest interfejs ( w pewnym ograniczonym zakresie ).

## Kolory bloków

Pozwala na zmianę koloru elementów składowych programu użytkownika.

## Ustawienia wielkości bloków

Umożliwia zmianę domyślnej wielkości bloków w programie użytkownika.

## Debugger

Odpowiada za debugowanie programu użytkownika.

## Śledzenie zmiennych

Dodawanie i usuwanie zmiennych do śledzenia podczas debugowania.

## Zarządzanie Breakpointami

Pozwala dodawać, usuwać i edytować breakpointy.

## Dodawanie Breakpointów

Umożliwia dodanie nowego breakpointu.

## Usuwanie Breakpointów

Pozwala na usunięcie istniejącego breakpointa.

## Przesuwanie Breakpointów

Pozwala na przeciągnięcie breakpointa w inne miejsce w programie użytkownika.

## Edycja ustawień debuggera

Umożliwia zmianę ustawień działania debuggera.

## Przypadki użycia

Nazwa Przypadku : Rozpoczęcie pisania programu użytkownika.

Aktor : Użytkownik.

Warunki początkowe : Włączenie programu.

Zdarzenie inicjujące : Stworzenie lub załadowanie projektu.

Przebieg przypadku : 1. Po włączeniu programu należy wybrać opcję projekt z menu głównego.  
2. Po przejściu do menu projektu należy wczytać lub utworzyć nowy projekt.  
3. W przypadku tworzenia nowego projektu należy go nazwać.

Przebiegi alternatywne : Zamiast wybierać opcji z menu, możliwe jest wykorzystanie skrótu klawiszowego.

Sytuacje wyjątkowe : W przypadku nazwaniu nowego projektu nazwą, która jest już zajęta program poprosi o podjęcie decyzji, czy nadpisać istniejący projekt.

Warunki końcowe : Program tworzy nową kartę projektu, w której możliwe jest tworzenie programu użytkownika.

Funkcje użyte w przypadku : Zarządzanie projektem  
Nazywanie projektu  
Wczytanie projektu

Nazwa Przypadku : Zapisanie projektu.

Aktor : Użytkownik.

Warunki początkowe : Posiadanie aktywnego programu użytkownika.

Zdarzenie inicjujące : Wybranie opcji zapisania projektu z menu lub użycie skrótu klawiszowego.

Przebieg przypadku : 1. Po wybraniu opcji następuje zapisanie projektu.

Przebiegi alternatywne : brak

Sytuacje wyjątkowe : brak  
Warunki końcowe : Zapisanie projektu użytkownika.  
Funkcje użyte w przypadku : Zarządzanie projektem  
Zapisanie projektu

Nazwa Przypadku : Kompilowanie programu.  
Aktor : Użytkownik.  
Warunki początkowe : Utworzenie lub wczytanie projektu.  
Zdarzenie inicjujące : Uruchomienie kompilowania programu użytkownika z poziomu menu lub przy użyciu skrótu klawiszowego.  
Przebieg przypadku : 1. Po uruchomieniu programu użytkownika program użytkownika jest kompilowany.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku błędów kompilacji, program wyświetli odpowiedni komunikat w polu konsoli programu.  
Warunki końcowe : Program kompiluje program użytkownika  
Funkcje użyte w przypadku : Start/Stop/Reset  
Kompilowanie programu użytkownika

Nazwa Przypadku : Uruchomienie programu.  
Aktor : Użytkownik.  
Warunki początkowe : Pomyślna kompilacja programu lub wczytanie wcześniej skompilowanego projektu.  
Zdarzenie inicjujące : Uruchomienie programu użytkownika z poziomu menu lub przy użyciu skrótu klawiszowego.  
Przebieg przypadku : 1. Po uruchomieniu program wykonuje skompilowany program użytkownika.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : brak  
Warunki końcowe : Program użytkownika zostaje uruchomiony.  
Funkcje użyte w przypadku : Start/Stop/Reset  
Uruchamianie programu użytkownika

Nazwa Przypadku : Zatrzymanie działania programu.  
Aktor : Użytkownik.  
Warunki początkowe : Program użytkownika jest w trakcie wykonywania się.  
Zdarzenie inicjujące : Zatrzymanie działającego programu użytkownika z poziomu menu lub przy użyciu skrótu klawiszowego.  
Przebieg przypadku : 1. Po włączeniu funkcji zatrzymania program zatrzymuje wykonywanie programu użytkownika.  
2. Program wyświetla w konsoli ostatnią wykonaną przez program instrukcję.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : brak  
Warunki końcowe : Program użytkownika zostaje zatrzymany.  
Funkcje użyte w przypadku : Start/Stop/Reset  
Zatrzymywanie wykonywania programu użytkownika

Nazwa Przypadku : Dodanie składowej programu użytkownika.  
Aktor : Użytkownik.  
Warunki początkowe : Rozpoczęcie pisania programu użytkownika.  
Zdarzenie inicjujące : Interakcja z menu wyboru składowej programu.  
Przebieg przypadku : 1. Wybieranie typu składowej.  
2. Wybranie składowej.

	3. Umieszczenie składowej w programie użytkownika
Przebiegi alternatywne	: W przypadku tworzenia zmiennej po kroku 3. należy wybrać unikalną nazwę.
Sytuacje wyjątkowe	: W przypadku kolizji nazw zmiennych program poprosi o wybranie innej nazwy dla zmiennej. W przypadku upuszczenia składowej poza okno programu tworzenie zostanie anulowane.
Warunki końcowe	: Utworzenie nowej składowej programu użytkownika lub anulowanie procesu.
Funkcje użyte w przypadku	: Zarządzanie składowymi programu użytkownika Dodawanie składowych Utworzenie składowej programu użytkownika. Wybór typu ("duszki", wyrażenia, pętle/instrukcje warunkowe/etc, interakcje(dźwięk), i/o, ruch, kontrola pisaka, zmienne) Nazywanie składowej Przesuwanie elementów
Nazwa Przypadku	: Usuwanie składowych programu użytkownika
Aktor	: Użytkownik.
Warunki początkowe	: Posiadanie aktywnego programu użytkownika z przynajmniej jedną składową.
Zdarzenie inicjujące	: Kliknięcie opcji usuwania składowej w menu, zaznaczenie jej i wciśnięcie przycisku Backspace albo Del lub przeciągnięcie składowej poza okno programu użytkownika.
Przebieg przypadku	: 1. Po wykonaniu zdarzenia inicjującego następuje usunięcie składowej wraz z składowymi podrzędnymi.
Przebiegi alternatywne	: brak
Sytuacje wyjątkowe	: brak
Warunki końcowe	: Składowa programu użytkownika zostaje usunięta.
Funkcje użyte w przypadku	: Zarządzanie składowymi programu użytkownika Usuwanie składowych
Nazwa Przypadku	: Przesuwanie składowych użytkownika
Aktor	: Użytkownik.
Warunki początkowe	: Posiadanie aktywnego programu użytkownika z przynajmniej dwoma składowymi.
Zdarzenie inicjujące	: Chwycenie składowej programu użytkownika za pomocą myszy.
Przebieg przypadku	: 1. Po chwyceniu składowej należy ją przeciągnąć za inną składową. 2. Po upuszczeniu składowej następuje przesunięcie składowej.
Przebiegi alternatywne	: brak
Sytuacje wyjątkowe	: W przypadku upuszczenia składowej poza okno programu użytkownika następuje jej usunięcie.
Warunki końcowe	: Przesunięcie lub usunięcie składowej.
Funkcje użyte w przypadku	: Zarządzanie składowymi programu użytkownika Przesuwanie elementów Usuwanie składowych
Nazwa Przypadku	: Łączenie składowych w bloki
Aktor	: Użytkownik.
Warunki początkowe	: Posiadanie aktywnego programu użytkownika z przynajmniej dwoma składowymi nie będących blokiem.
Zdarzenie inicjujące	: Zaznaczenie składowych i wybór opcji łączenia w blok.
Przebieg przypadku	: 1. Po wyborze opcji składowe zostaną połączone w blok.
Przebiegi alternatywne	: brak
Sytuacje wyjątkowe	: brak
Warunki końcowe	: Utworzenie bloku składowych programu użytkownika.

Funkcje użyte w przypadku : Zarządzanie składowymi programu użytkownika  
Łączenie w bloki

Nazwa Przypadku : Edycja wartości składowej użytkownika  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego programu użytkownika z przynajmniej jedną składową.  
Zdarzenie inicjujące : Podwójne kliknięcie w istniejącą składową programu użytkownika.  
Przebieg przypadku : 1. Po podwójnym kliknięciu w składową należy podać nową wartość składowej.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku nie podania żadnej wartości albo wartości błędnej, edycja zostanie cofnięta.  
Warunki końcowe : Zmiana wartości składowej albo pozostawienie jej bez zmian.  
Funkcje użyte w przypadku : Zarządzanie składowymi programu użytkownika  
Edycja wartości

Nazwa Przypadku : Edycja nazwy zmiennej.  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego programu użytkownika z przynajmniej jedną zmienną.  
Zdarzenie inicjujące : Wybranie opcji edycji nazwy z menu.  
Przebieg przypadku : 1. Po wybraniu opcji z menu należy podać nową nazwę zmiennej.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku kolizji nazw nazwa nie zostanie zmieniona.  
Warunki końcowe : Zmiana nazwy zmiennej albo pozostawienie jej bez zmian.  
Funkcje użyte w przypadku : Zarządzanie składowymi programu użytkownika  
Edycja nazwy

Nazwa Przypadku : Cofnięcie ostatniej akcji w programie użytkownika.  
Aktor : Użytkownik.  
Warunki początkowe : Wykonanie jakiejś nie cofniętej operacji na składowych w programie użytkownika.  
Zdarzenie inicjujące : Wybór opcji cofnięcia z menu lub użycie skrótu klawiszowego.  
Przebieg przypadku : 1. Po wybraniu opcji z menu ostatnia akcja nie będąca cofnięciem zostaje cofnięta.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : brak  
Warunki końcowe : Cofnięcie akcji w programie użytkownika.  
Funkcje użyte w przypadku : Zarządzanie składowymi programu użytkownika  
Cofnięcie ostatniej akcji

Nazwa Przypadku : Konwersja kodu obrazkowego na kod wykonywalny  
Aktor : Użytkownik.  
Warunki początkowe : Skompilowanie aktywnego programu użytkownika lub wczytanie wcześniej skompilowanego projektu.  
Zdarzenie inicjujące : Wybranie opcji kompilowania na inny język z menu.  
Przebieg przypadku : 1. Po wybraniu opcji należy wybrać język docelowy.  
2. Jeżeli chcemy zapisać wynik konwersji do pliku należy takowy podać. W przeciwnym wypadku wynik konwersji zostanie wyświetlony jako nowa karta w programie.  
Przebiegi alternatywne : brak.  
Sytuacje wyjątkowe : W przypadku podania nazwy istniejącego pliku zostaniemy zapytani, czy chcemy go nadpisać.  
Warunki końcowe : Zakończenie konwersji na wybrany język programowania.

Funkcje użyte w przypadku : Konwersja kodu obrazkowego na kod wykonywalny  
Wybór języka docelowego  
Zapis do pliku

Nazwa Przypadku : Ustawienie opcji kompilowania  
Aktor : Użytkownik.  
Warunki początkowe : Uruchomienie programu.  
Zdarzenie inicjujące : Kliknięcie menu wyboru opcji kompilatora.  
Przebieg przypadku : 1. Po wybraniu menu opcji kompilatora należy wybrać opcję którego z języków chcemy edytować.  
2. Należy wyedytować interesujące nas opcje  
3. Po zakończeniu wyboru należy kliknąć przycisk zastosuj.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku nie wciśnięcia przycisku zastosuj zmiany nie zostaną zaaplikowane.  
Warunki końcowe : Edycja opcji kompilowania.  
Funkcje użyte w przypadku : Kompilator  
Wybór flag kompilatora

Nazwa Przypadku : Zmiana szaty graficznej programu.  
Aktor : Użytkownik.  
Warunki początkowe : Włączenie programu.  
Zdarzenie inicjujące : Wybór edycji szaty graficznej.  
Przebieg przypadku : 1. Po wyborze menu edycji szaty graficznej należy wybrać interesujące nas opcje.  
2. Po zakończeniu wyboru należy kliknąć przycisk zastosuj.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku nie wciśnięcia przycisku zastosuj zmiany nie zostaną zaaplikowane.  
Warunki końcowe : Zmiana szaty graficznej lub anulowanie edycji.  
Funkcje użyte w przypadku : Wygląd programu  
Ustawienia skóry  
Rozmiar i rodzaj czcionek  
Kolory bloków  
Ustawienia wielkości bloków

Nazwa Przypadku : Włączenie debugowania programu użytkownika.  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego skompilowanego programu użytkownika.  
Zdarzenie inicjujące : Wybranie opcji debugowania z menu.  
Przebieg przypadku : 1. Po wybraniu opcji debugowania następuje włączenie programu użytkownika w trybie debugowania.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : brak  
Warunki końcowe : Program użytkownika zostaje włączony w trybie debugowania.  
Funkcje użyte w przypadku : Debugger

Nazwa Przypadku : Śledzenie zmiennych.  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego skompilowanego programu użytkownika.  
Zdarzenie inicjujące : Wybranie opcji dodania zmiennych do śledzenia.  
Przebieg przypadku : 1. Po wybraniu opcji zmienna zostaje dodana do listy zmiennych śledzonych lub



usunięta jeśli zostanie dodana drugi raz.

2. Przy włączeniu programu w trybie debugowania będzie można podejrzeć wartość zmiennej podczas wykonywania programu użytkownika.

Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : brak  
Warunki końcowe : Dodanie zmiennej do listy zmiennych śledzonych  
Funkcje użyte w przypadku : Debugger  
Śledzenie zmiennych

Nazwa Przypadku : Dodawanie breakpointów  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego skompilowanego programu użytkownika.  
Zdarzenie inicjujące : Wybór dodania breakpointa w menu.  
Przebieg przypadku : 1. Po wyborze opcji w menu należy wskazać miejsce w programie w którym ma zostać  
dodany breakpoint.

Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku kliknięcia w miejsce w którym istnieje breakpoint, program nie poczyni zmian.  
Warunki końcowe : Dodanie nowego breakpointa albo brak zmiany w przypadku wystąpienia sytuacji wyjątkowej  
Funkcje użyte w przypadku : Zarządzanie Breakpointami  
Dodawanie Breakpointów

Nazwa Przypadku : Usuwanie breakpointów  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego skompilowanego programu użytkownika z przynajmniej jednym breakpointem.  
Zdarzenie inicjujące : Zaznaczenie breakpointa i wybór opcji usunięcia go w menu.  
Przebieg przypadku : 1. Po wyborze opcji w menu breakpoint zostanie usunięty.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : bra.  
Warunki końcowe : Usunięcie breakpointa.  
Funkcje użyte w przypadku : Zarządzanie Breakpointami  
Usuwanie Breakpointów

Nazwa Przypadku : Przesuwanie breakpointów  
Aktor : Użytkownik.  
Warunki początkowe : Posiadanie aktywnego skompilowanego programu użytkownika z przynajmniej jednym breakpointem.  
Zdarzenie inicjujące : Zaznaczenie breakpointa i przeciągnięcie go w inne miejsce w programie użytkownika.  
Przebieg przypadku : 1. Po przeciągnięciu breakpointa na nową pozycję należy go upuścić.  
Przebiegi alternatywne : brak  
Sytuacje wyjątkowe : W przypadku Przesunięcia breakpointa na innych breakpoint lub poza okno programu użytkownika breakpoint zostanie usunięty.  
Warunki końcowe : Przesunięcie albo Usunięcie breakpointa.  
Funkcje użyte w przypadku : Zarządzanie Breakpointami  
Przesuwanie Breakpointów  
Usuwanie Breakpointów

Nazwa Przypadku	: Edycja ustawień debuggera.
Aktor	: Użytkownik.
Warunki początkowe	: Włączenie programu.
Zdarzenie inicjujące	: Wybór opcji zmiany ustawień debuggera w menu.
Przebieg przypadku	: 1. Po wyborze opcji zmiany ustawień należy wybrać interesujące nas ustawienia debugowania. 2. Po zakończeniu edycji ustawień należy kliknąć przycisk zastosuj.
Przebiegi alternatywne	: brak
Sytuacje wyjątkowe	: W przypadku nie wciśnięcia przycisku zastosuj zmiany nie zostaną zaaplikowane.
Warunki końcowe	: Zakończenie edycji opcji debugowania.
Funkcje użyte w przypadku	: Edycja ustawień debuggera

## Wymagania niefunkcjonalne

### 1. Czas działania

Określa maksymalny czas wykonania pojedynczej operacji. Głównie czas kompilacji/włączenia programu.

Wartość graniczna : 3 sek.

### 2. Rozmiar

Określa jaki maksymalny rozmiar może zajmować program wraz z wszystkimi plikami pomocniczymi.

Wartość graniczna : 500 MB

### 3. Wprowadzanie danych

Określa sposób wprowadzania danych do programu przez użytkownika.

Wartość : Klawiatura, mysz

### 4. Interakcja z użytkownikiem

Określa sposób w jaki program wyświetla swój stan oraz komunikuje się z użytkownikiem

Wartość : Ekran, głośnik

### 5. Bezpieczeństwo

Określa stopień bezpieczeństwa na włamania, wirusy itd.

Wartość : Wysoki standard bezpieczeństwa

### 6. Interfejsy komunikacyjne

Dostęp do sieci, protokoły, wydajność sieci, poziom abstrakcji protokołów komunikacyjnych, itd.

Wartość : Brak

## 7. Odporność na awarie

Określa konsekwencje błędów w oprogramowaniu, przerwy w zasilaniu, kopie zabezpieczające, częstotliwości składowania, dziennika zmian, itd.

Wartość : Automatyczny zapis aktywnych projektów w zewnętrznym pliku, usuwanym po wczytaniu projektów przy ponownym starcie programu. Wszystkie akcje użytkownika na składowych aktualnie aktywnego programu użytkownika są archiwizowane w celu umożliwienia cofnięcia ich.

## 8. Zasoby

Określa ograniczenia finansowe, ludzkie i materiałowe.

Wartość : Ludzkie : 40 osób + 10 możliwych do zatrudnienia w razie opóźnień.

Finansowe : 2 mln PLN

## 9. Skala czasowa

Określa ograniczenia na czas wykonania systemu, czas szkolenia, wdrożenia, itd.

Wartość : 14 miesięcy

## Zadanie II

### Projekt Architektury

Sprzęt, "MacBooki dla każdego!", ekrany prezentacyjne (do analizowania postępu, etc),

Oprogramowanie: MacOS + Unixowe systemy. Visual Studio (głównie C++), autorskie oprogramowanie do testowania oprogramowania i GIT (program do kontroli wersji).

Oprogramowanie służące do komunikacji między pracownikami: Slack.

Struktura logiczna: Projekt będzie pisany modularnie. Każdy moduł będzie od siebie w miarę możliwości niezależny oraz składał się będzie z klas. Nie będzie zagnieżdżonych modułów. Projekt nie będzie zawierał żadnej skomplikowanej bazy danych. Będziemy korzystać jedynie z systemów już zaimplementowanymi w naszej firmie, m.in do zarządzania postępami pracowników, jednak po stronie klienta takowej brak.

### Główne zasady kodowania (Style Guide)

Code: Kod będzie pisany ze standardem: Google C++

User Experience: System graficzny będzie oparty na standardzie FLAT.

### Identyfikacja ryzyka

- Konkurencja: Jak na razie na rynku nie widać jakiegokolwiek konkurencji, ale gdy się pojawi, Zmniejszymy czas wytworzenia oprogramowania poprzez przeniesienie mniej wydajnych pracowników do innych projektów lub zwiększymy ilość pracowników. Tak aby nasze oprogramowanie pojawiło się jak najszybciej na rynku.
- Zmiana kursu Euro: Nasz kapitał częściowo będzie składał się z waluty Euro, tak, aby podczas zmiany kursów, tymczasowe koszty nie uległy znacznym zmianom, nawet jeśli oznaczałoby to dodatkowy zysk dla firmy.
- Niewywiązywanie się z terminu dostarczenia oprogramowania. Będziemy starać się w jak najkrótszym czasie wywiązać się z owego terminu, jednak nie planujemy zwiększania kosztów tak drastycznie jak podczas pojawienia się konkurencji. Co wynika z dużego popytu na nasz produkt.

## Ocena zgodności tablicy koncepcyjnej ze specyfikacją wymagań

Nic nie wskazuje na to, że od stworzenia tablicy koncepcyjnej pojawiły się jakiegokolwiek nowe ryzyka, problemy czy wymagania. Wszystko tworzone jest zgodnie z tokiem przedstawionym w tablicy koncepcyjnej.

## Słowniczek pojęć

Program - Poprzez termin program rozumiemy wyrób naszej firmy - Programinator.

Program użytkownika - Jest to program napisany przez użytkownika przy użyciu Programinatora.

Aktywny program użytkownika - Jest to program użytkownika w otwartej do edycji karcie w naszym programie. (można mieć jednocześnie otwartych kilka kart z programami użytkownika ale edytować naraz - tylko jedną)

ThickPic - Język programowania, w którym przy pomocy składowych (czyli obrazków, kafelków przy pomocy których składamy program) tworzymy program użytkownika w Programinatorze.