Nazwa i akronim projektu: Extrema System Search- ESS	**Zleceniodawca:** *projekt własny*	**Zleceniobiorca:** PG, WFTiMS, zespół projektowy numer 1
Numer zlecenia: PG-WFTiMS-IO-2018-001	**Kierownik projektu:** Michał Jagielski	**Opiekun projektu:** dr inż. Marta Łabuda

Nazwa / kod dokumentu: **Harmonogram**	**Nr wersji:** *1.0*
Odpowiedzialny za dokument: *Michał Jagielski*	**Data pierwszego sporządzenia:** 05.10.2018
	Data ostatniej aktualizacji: *05.10.2018*

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0	Wersja wstępna.	*całość*	*Michał Jagielski*	05.10.2018

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

Celem tego dokumentu jest sprecyzowanie zadań niezbędnych do wykonania oraz zaplanowanie prac projektowych w zakresie: --tworzenia kodu źródłowego aplikacji
-oszacowanie ryzyka i podjęcie odpowiednich środków zapobiegawczych
-wyznaczenie terminów pracy przy poszczególnych zadaniach oraz osób za nie odpowiedzialnych
-oszacowanie czasu niezbędnego do wykonania poszczególnych zadań

1.2 Zakres dokumentu

Dokument precyzuje zadania do wykonania, przybliża proces wytwarzania oprogramowania oraz definiuje organizację projektu.

1.3 Odbiorcy

Opiekun projektu - dr. inż. Marta Łabuda
Członkowie zespołu : Michał Jagielski, Maciej Dąbrowski, Krzysztof Bądkowski, Paul Bugaj, Michał Fladziński

1.4 Terminologia

-ekstremum funkcji- maksymalna lub minimalna wartość jaką osiąga funkcja – miejsca gdzie pochodna funkcji jest równa 0,
*lokalne – miejsca, w których funkcja osiąga minimum albo maksimum ale nie jest to największa lub najmniejsza wartość biorąc pod uwagę całą funkcję
*globalne – miejsce/miejsca, w którym/których funkcja osiąga absolutne i jedyne minimum bądź maksimum
-pochodna funkcji- miara szybkości zmian wartości funkcji względem zmian jej argumentów.

-granica funkcji – wartość, do której obrazy danej funkcji zbliżają się nieograniczenie dla argumentów dostatecznie bliskich wybranemu punktowi
-algorytm heurystyczny- metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego, a często nawet prawidłowego. Rozwiązań tych używa się np. wtedy, gdy pełny algorytm jest z przyczyn technicznych zbyt kosztowny lub gdy jest nieznany (np. przy przewidywaniu pogody).
-metody numeryczne - są jedną z tych dziedzin matematyki stosowanej, których zastosowanie w praktyce jest powszechne. Wykorzystywane są wówczas, gdy badany problem nie ma w ogóle rozwiązania analitycznego (danego wzorami), lub korzystanie z takich rozwiązań jest uciążliwe ze względu na ich złożoność lub z innych powodów.

2 Proces wytwarzania

2.1 Strategia prowadzenia projektu

Projekt będzie wytwarzany wykorzystując **cykl przyrostowy.**

2.2 Opis etapów wytwarzania (prowadzenia projektu)

1. Określenie całości wymagań, wykonanie wstępnego, ogólnego projektu całości systemu
2. Wybór pewnego podzbioru funkcji systemu (funkcjonalności)
   - -moduł algorytmiczny „HeuristicAlgorithsm”
   - -moduł graficzny „Graphic”
   - -GUI
   - -moduł komunikacyjny „Communication”
3. Szczegółowy projekt oraz implementacja części systemu realizującej wybrane funkcje
4. Testowanie zrealizowanego fragmentu i dostarczenie go klientowi
   - -testowanie odbywania się na etapie powstawania poszczególnych modułów
5. Powtarzanie kolejnych etapów, aż do zakończenia implementacji całego systemu

3 Organizacja projektu

3.1 Zespół projektowy

Specyfikacja wymagań - 5 osób - analitycy
Obsługa i implementacja - 5 osoby - programiści API
Tworzenie interfejsu użytkownika - 3 osoby - projektanci, programiści GUI
Tworzenie funkcjonalności związanej z znajdywaniem ekstremów funkcji - 2 osoby – programiści oprogramowania
Tworzenie funkcjonalności związanej z wyświetlaniem funkcji w 2 i 3 wymiarach – 2 osoby – programiści graficzni
Testowanie i poprawy funkcjonalności - 2 osoby - testerzy, programiści
Nadzór projektu (zgodność ze przyjętym standardem pisania kodu) – 1 osoba – nadzorca projektu

Imię i nazwisko	***Rola w projekcie***
Michał Jagielski	Programista / Nadzorca projektu / Kierownik
Maciej Dąbrowski	Programista / Dokumentalista / Tester

Krzysztof Bądkowski	Programista / Grafik /
Paul Bugaj	Programista / Grafik / Tester
Michał Fladziński	Programista / Grafik

Powyższy podział ról przypisuje nadzór, nad poszczególnymi sekcjami, konkretnym członkom zespołu. Kierownik w porozumieniu z nadzorcą danej sekcji będzie rozdzielał szczegółowe zadania członkom zespołu.

3.2 Infrastruktura techniczna

W ramach realizacji projektu użyte będą następuje narzędzia i technologie:
- komputery stacjonarne, laptopy
- Microsoft Office
- Discord, Skype
- Microsoft Visual Studio
- GitHub
- Google Drive
- Google Documents
- Microsoft .NET
- Gmail

3.3 Infrastruktura komunikacyjna

- spotkanie kontrolne grupy raz w tygodniu:
  środa, 10:00, PG, 115 GG
- spotkanie z opiekunem projektu raz w tygodniu:
  środa, 10:00, PG, 115 GG
- komunikacja skype
- komunikacja e-mailowa:
  adresy e-mail:
  - Michał Jagielski - michal77_1996@gmail.com
  - Maciej Dąbrowski - mdabrowski@task.gda.pl
  - Krzysztof Bądkowski - ka.badkowski@gmail.com
  - Paul Bugaj - bugajpaul@gmail.com
  - Michał Fladziński–michalfladzinski@gmail.com
- komunikacja Messenger
- komunikacja Discord

3.4 Infrastruktura dokumentacyjna (bezpośrednio dotycząca projektu)

- *SWS (Specyfikacja Wymagań Systemowych)* - zdefiniowanie udziałowców, wymagań na podstawie otoczenia projektu oraz analizy potrzeb klienta
- *AWS (Analiza i Wizja Systemu)* - przedstawienie celu projektu, zakresu funkcjonalności, modelu obiektowego oraz przypadków użycia systemu.
- *PS (Projekt Systemu)* - dokładny opis systemu, jego ogólny zarys, kosztorys oraz plan wdrażania na rynek konsumencki.
- "Raport z prac" będzie przygotowywany przez zespół projektowy lub jednego członka i zatwierdzany przez resztę zespołu.

3.5 Zarządzanie jakością w projekcie

- *sprawdzanie każdego dokumentu przez osobę, która nie jest autorem - osoba ta zweryfikuje czy dokument spełnia swoje założenia i czy jest zrozumiały*
- *inspekcja oprogramowania - weryfikacja wytworzonego oprogramowania przez członków grupy, by stwierdzić czy projekt spełnia założenia*
- *inspekcja kodu przez osobę, która nie jest autorem - osoba ta zweryfikuje styl kodowania, czy jest zrozumiały i zgodny z przyjętymi standardami pisania kod*
- *inspekcje kluczowych dokumentów - inspekcje te pomogą potwierdzić słuszność opisanych w dokumencie założeń dotyczących realizacji projektu*
- *przestrzeganie zasad kodowanaia (przyjętego przez grupę):*
  1. *funkcje powinny w miarę możliwości zwracać „true" jeśli zostały pomyślnie wykonane lub „false" jeśli doszło do błędu lub zwracać odpowiednie wyjątki*
  2. *wszelkie moduły powinny być ograniczone do wytyczonej przestrzeni nazw aby unikać problemów powstawania zmiennych i klas o tej samej nazwie*
  3. *moduły powinny być jak najprostsze w obsłudze dla pozostałych użytkowników – posiadać tylko niezbędne funkcjonalności*

3.6 Zarządzanie ryzykiem w projekcie

Czynnik ryzyka	***Prawdopodobieństwo wystąpienia*** (0-100%)	***Działania zapobiegawcze***	***Plany awaryjne(jeśli możliwe)***
Brak dobrej znajomości języka programowania C#	*20%*	*Dodatkowa nauka języka we własnym zakresie*	---
Brak współpracy niektórych członków zespołu	*70%*	*Całodobowy nadzór kierownika nad stanem zleconej pracy*	*Przydzielenie zadania innej osobie i zgłoszenie tego opiekunowi projektu*
Niemożliwość zaimplementowania pewnej funkcjonalności aplikacji	*20%*	*Przestrzeganie standardu pisania oprogramowania obiektowego*	*Ograniczenie tej funkcjonalności, znaleźnie funkcjonalności zastępczej lub całkowite jej usunięcie*
Zbyt mała ilość godzin na realizację całego projektu	*50%*	*ograniczenie funkcjonalności aplikacji do minimum*	*stworzenie „prototypu" aplikacji*
Utracenie części kodu w wyniku awarii sprzętu	*10%*	*umieszczanie projektu po każdej modyfikacji w repozytorium*	*cofnięcie projektu do ostatniego etapu zapisanego w repozytorium*

4 Harmonogram projektu

4.1 Ograniczenia czasowe na projekt

Na implementację projektu przeznaczymy 8 tygodni począwszy od 10.10.2018. Aplikacja zostanie oddana do oceny na ostatnich zajęciach laboratoryjnych, które odbędą się 12.12.2018.

4.2 Oszacowanie czasu realizacji poszczególnych etapów

1. Przygotowanie i zaprojektowanie modułu odpowiedzialnego za obliczanie ekstremów funkcji dowolnej liczby zmiennych przy użyciu algorytmów heurystycznych – 2tygodnie czyli 60h
    - Przygotowanie odpowiednich 6 dowolnych algorytów heurystycznych
    - Moduł ma zawierać funkcjonalności (metody) odpowiedzialne za:
        - ♦ Wykonanie jednej iteracji dla dowolnego z 6 algorytmów
        - ♦ Wykonanie pełnej pętli odnajdywania minimum/maksimum i zwrócenie znalezionego punktu
        - ♦ Wszelkie pozostałe funkcje powinny być prywatne – jeśli nie jest to możliwe powinny być ograniczone do minimum
2. Przygotowanie i zaprojektowanie modułu odpowiedzialnego za przyjmowanie i przetwarzanie „tablicy" punktów funkcji dowolnej liczby zmiennych i zamienianie ich na przybliżoną funkcje odpowiedniej liczby zmiennych – 2tydzien czyli 60osobogodzin
    - Przygotowanie bezpiecznego sposobu wczytywania plików z danymi
    - Moduł ma zawierać funkcjonalności (metody) odpowiedzialne za:
        - ♦ Wczytywanie puntków z plików tekstowych (.txt) lub excela (.xlsx) i przetworzenie ich na hardwerową tablice punktów
        - ♦ Zamianę wyżej wymienionej tablicy punków na wzór funkcji i zwrócenie jej w postaci łańcucha znaków(string) zrozumiałych dla użytkownika
        - ♦ Wszelkie pozostałe funkcje powinny być prywatne – jeśli nie jest to możliwe powinny być ograniczone do minimum
3. Stworzenie interfejsu użytkownika dostosowanego do innych modułów – 3tygodnie czyli 90 osobogodzin
    - Przygotowanie przyjaznego i intuicyjnego interfejsu użytkownika
    - GUI powinno być dostosowane do funkcji pozostałych modułów i prawidłowo z nimi współpracować
4. Przygotowanie i zaprojektowanie modułu odpowiedzialnego za wyświetlanie podanej przez użytkownika funkcji w postaci graficznej (jednej i dwóch zmiennych) – 2tygodnie czyli 60 osobogodzin
    - Podana funckja powinna być wyświetlana na podstawie podanej przez użytkownika funkcji w postaci łańcucha znaków i wyświetlana w dpowiednim przedziale
    - Moduł ma zawierać funkcjonalności (metody) odpowiedzialne za:
        - Wczytanie funkcji podanej w postaci łańcucha znaków i warunków ograniczających a następnie wyświetlenie jej w odpowiedniej kontrolce (panelu)
        - Podłączenie do modułu odpowiedniej kontrolki i przystosowanie jej w ten sposób do łatwego i szybkiego wyświetlania zadanej funkcji
        - Wszelkie pozostałe funkcje powinny być prywatne – jeśli nie jest to możliwe powinny być ograniczone do minimum
5. Sprawdzanie poprawności działania programu (testowanie) – 1tydzień czyli 30osobogodzin
6. Łączenie modułów w funkcjonalną aplikację – 2tygodnie czyli 60 osobogodzin
    - Połączenie wszystkich modułów do powstałego intefesju użytkownika w sposób, w którym będą one komunikowały się ze sobą bez błędów

Szacujemy, że prace nad poszczególnymi etapami zajmą nam 360 osobogodzin. Nasze obliczenia są obarczone znacznym ryzykiem, nie jesteśmy w stanie dokładnie przewidzieć ile czasu spędzimy nad danym etapem.

4.3 Przydzielenie odpowiedzialności i ścieżki krytyczne

L.p.	Szacowany czas wykonania	Data rozpoczęcia prac	Data zakończenia prac	Osoba odpowiedzialna	Osoby współodpowiedzialne	Plan awaryjny
1	2 tygodnie	10.10.2018	24.10.2018	Michał Jagielski	Krzysztof Bądkowski	-ograniczenie ekstremów do funkcji ograniczonej liczby zmiennych -wydłużenie czasu prac
2	2 tygodnie	17.10.2018	31.10.2018	Maciej Dąbrowski	Michał Fladziński	-ograniczenie działania modułu do minimum -wydłużenie czasu prac
3	3 tygodnie	07.11.2018	28.12.2018	Michał Fladziński	Maciej Dąbrowski	-wykorzystanie najprostrzych wzorców graficznych -konsultacja z resztą członków ekipy
4	2 tygodnie	31.10.2018	14.11.2018	Paul Bugaj	Michał Jagielski, Krzysztof Bądkowski	-uproszczenie wyświetlania funkcji (tylko od jednej zmiennej) -wydłużenie czasu pracy
5	1 tygodnie	05.12.2018	12.12.2018	Michał Jagielski	Michał Fladziński	-konsultacja z resztą zespołu -wyłączenie niedziałającego modułu lub ograniczenie jego funkcjonalności
6	2 tygodnie	21.11.2018	05.12.2018	Krzysztof Bądkowski	Paul Bugaj	-konsultacja z całym zespołem -wydłuzenie czasu prac -ograniczenie funkcjonalności modułów