

Nazwa i akronim projektu: Szukanie ekstremów funkcji różnymi algorytmami / ESS (Extrema System Search)	Zleceniodawca: Projekt własny	Zleceńbiorca: PG, WFTiMS, zespół projektowy IO nr 001
Numer zlecenia: PG-WFTiMS-IO-2010-001	Kierownik projektu: Michał Jagielski	Opiekun projektu: Prof. dr hab. Józef E. Sienkiewicz

Specyfikacja Wymagań Systemowych (SWS)	Nr wersji: 1.0.2
Odpowiedzialny za dokument: Michał Jagielski	Data pierwszego sporządzenia: 08.05.2018
	Data ostatniej aktualizacji: 10.05.2018

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.0.1	wstępna wersja	całość	Michał Jagielski	08.05.2018
1.0.2	Poprawiona wersja	1,2,4	Michał Jagielski	10.05.2018

1 Wprowadzenie – o dokumencie

1.1 Cel i zakres dokumentu

Dokument specyfikacji wymagań systemowych jest podsumowaniem fazy analizy. Głównym zadaniem jest postawienie wymagań zapewniających realizację wyznaczonych celów. Głównym udziałowcem jest zespół pracujący nad projektem, opiekun projektu, zlecniodawca systemu informatycznego, a także przyszli odbiorcy oprogramowania czyli użytkownicy i firmy zewnętrzne. Do udziałowców pośrednich przede wszystkim należą firmy dostarczające oprogramowanie niezbędne w realizacji projektu tj.: Adobe Systems oraz Microsoft Corporation..

1.2 Odbiorcy

Prof. dr hab. Józef E. Sienkiewicz, prof. zw. PG, Michał Jagielski, Krzysztof Bądkowski, Maciej Dąbrowski, Paul Bugaj, Michał Fladziński, studenci

1.3 Terminologia

-ekstremum funkcji- maksymalna lub minimalna wartość funkcji.

-pochodna funkcji- miara szybkości zmian wartości funkcji względem zmian jej argumentów.

-algorytm heurystyczny- metoda znajdowania rozwiązań, dla której nie ma gwarancji znalezienia rozwiązania optymalnego, a często nawet prawidłowego.

-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)

-interpolacja wielomianowa - metoda numeryczna polegająca na przybliżaniu funkcji za pomocą wielomianów.

-deweloper- osoba, która tworzy programy komputerowe w pewnym języku programowania

-wykres funkcji – graficzne przedstawienie funkcji.

-interfejs użytkownika – (środowisko graficzne)określenie sposobu prezentacji informacji przez komputer oraz interakcji z użytkownikiem.

-alokacja pamięci (dealokacja pamięci) – odpowiednio przydział i zwolnienie ciągłego obszaru pamięci.

2 Klienci i ich potrzeby

2.1 Klienci wewnętrzni

Kierownik - Osoba odpowiedzialna za wszystkich pracowników i za projekt. Główna rola szefów polega na upewnieniu się, że terminy są spełnione, a pracownicy wykonują swoją pracę.

Deweloper - Główną rolą programistów jest skuteczne kodowanie programu do potrzeb firmy. Programiści mają kluczowe znaczenie dla naszego projektu, ponieważ nasz cały projekt opiera się na stworzeniu skutecznego programu, z którego będą korzystać inni ludzie.

Analitik Biznesowy - Główną rolą naszego działu finansowego jest upewnienie się, że pracujemy w ramach naszego budżetu i efektywne zarządzanie naszym kapitałem, który otrzymujemy od naszych sponsorów. Dział finansowy jest również odpowiedzialny za obliczanie naszych kosztów i prognozowanie naszych przyszłych zysków oraz zarządzanie dochodami z reklam i donacji.

Zespół projektowy

L.p.	Imię i nazwisko	Rola	Potrzeba
1.	Michał Jagielski	Kierownik, Deweloper	Zarządzanie ekipą i nadzorowanie terminów

			<i>prac projektowych</i>
2.	<i>Paul Bugaj</i>	<i>Twórca rozwiązania, Deweloper</i>	<i>Zaplanowanie i wdrożenie odpowiednich modułów programistycznych, tworzenie aplikacji</i>
3.	<i>Maciej Dąbrowski</i>	<i>Analityk Biznesowy, Deweloper</i>	<i>Określenie potrzebnego budżetu i monitorowanie ewentualnych zysków / strat, tworzenie aplikacji</i>
4.	<i>Krzysztof Bądkowski</i>	<i>Doradca techniczny, Deweloper</i>	<i>Określanie czy dane zadanie jest wykonalne w określonym środowisku i czasie przy określonych zasobach ludzkich, czasu i sprzętu</i>
5.	<i>Michał Fladziński</i>	<i>Deweloper</i>	<i>Tworzenie aplikacji</i>

2.2 Klienci zewnętrzni

Indywidualny klient- Klient, który nie jest częścią organizacji lub klienta, który będzie korzystać z programu na własne potrzeby. Są to klienci korzystający z naszego programu na potrzeby domowe, osobiste hobby lub osobiste badania. Płacą subskrypcję na własny użytek.

Inne firmy lub organizacje- Są to klienci korzystający z naszego produktu w swoich firmach. W takim przypadku nasz program najprawdopodobniej zostanie użyty w połączeniu z innym programem utworzonym przez firmę klienta. Może być również wykorzystany do pomocy w badaniach prowadzonych przez firmę, w zależności od rodzaju wykonywanej pracy.

Szkoły i laboratoria badawcze- Klienci ci korzystaliby w przeważającej mierze z naszego produktu do celów naukowych i edukacyjnych. W szkołach programy mogą być wykorzystywane przez uczniów do nauki i jako narzędzie do ich edukacji. Może być również wykorzystywany przez profesorów do prowadzenia badań. W przypadku laboratoriów program ten może być używany podczas prowadzenia badań obejmujących funkcje i zerowanie funkcji.

3 Projekt w kontekście – udziałowcy i relacje z nimi

3.1 Udziałowcy

KARTA UDZIAŁOWCA	
Identyfikator:	UOB 01
Nazwa:	Grupa projektowa
Opis:	Osoby pracujące nad aplikacją: programowanie, testowanie, tworzenie dokumentacji.
Typ udziałowca:	ożywiony, bezpośredni
Punkt widzenia:	Techniczny, operator systemu
Reprezentant:	Michał Jagielski, Maciej Dąbrowski, Michał Fladziński, Paul Bugaj, Krzysztof Bądkowski
Ograniczenia:	Nieograniczony wpływ na treść wymagań funkcjonalnych systemu
Historia:	Cykliczne spotkania z członkami zespołu raz w tygodniu. Komunikowanie się przy pomocy Facebooka i komunikatorów głosowych - Skype, Discord
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UOB 02
Nazwa:	Zleceniodawca systemu
Opis:	Osoby pracujące nad aplikacją: programowanie, testowanie, tworzenie dokumentacji.
Typ udziałowca:	ożywiony, bezpośredni
Punkt widzenia:	Perspektywa techniczna, ekonomiczna, biznesowa, użytkownika systemu
Reprezentant:	Michał Jagielski, Maciej Dąbrowski, Michał Fladziński, Paul Bugaj, Krzysztof Bądkowski
Ograniczenia:	Nieograniczony wpływ na treść wymagań funkcjonalnych systemu
Historia:	Cykliczne spotkania z członkami zespołu raz w tygodniu. Komunikowanie się przy pomocy Facebooka i komunikatorów głosowych - Skype, Discord
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UOB 03
Nazwa:	Opiekun projektu
Opis:	Osoby pracujące nad aplikacją: programowanie, testowanie, tworzenie dokumentacji.
Typ udziałowca:	Ożywiony, bezpośredni
Punkt widzenia:	Perspektywa techniczna, biznesowa
Reprezentant:	prof. dr. hab. Józef Sienkiewicz
Ograniczenia:	Nieograniczony wpływ na treść wymagań funkcjonalnych systemu
Historia:	Cykliczne spotkania z członkami zespołu raz w tygodniu.

Wymagania:	----
-------------------	------

KARTA UDZIAŁOWCA	
Identyfikator:	UOB 04
Nazwa:	Użytkownik systemu
Opis:	Jednostka, która w przyszłości będzie korzystać z dostarczonych funkcjonalności określonego systemu
Typ udziałowca:	Ożywiony, bezpośredni
Punkt widzenia:	Perspektywa funkcjonalna
Reprezentant:	Społeczeństwo
Ograniczenia:	Nie zatwierdza poszczególnych etapów wykonania, nie ma wpływu na budowę systemu oraz na dostarczone funkcjonalności. Może jedynie korzystać z ostatecznej wersji systemu.
Historia:	----
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UOB 05
Nazwa:	Firmy zewnętrzne
Opis:	Firmy zewnętrzne zainteresowane rozwiązaniem proponowanym przez oprogramowanie ER
Typ udziałowca:	Ożywiony, bezpośredni
Punkt widzenia:	Perspektywa funkcjonalna
Reprezentant:	Firmy obecne na rynku
Ograniczenia:	Nie zatwierdza poszczególnych etapów wykonania, nie ma wpływu na budowę systemu oraz na dostarczone funkcjonalności. Może jedynie korzystać z ostatecznej wersji systemu.
Historia:	----
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UNP 01
Nazwa:	Microsoft Corporation
Opis:	Dostarcza środowisko programistyczne Microsoft Visual Studio wraz z pakietem do tworzenia aplikacji okienkowych w języku programistycznym C#, C++
Typ udziałowca:	Nieożywiony, pośredni
Punkt widzenia:	Deweloper środowiska programistycznego
Reprezentant:	Spółka Microsoft
Ograniczenia:	Nie wpływa na działanie algorytmów, zespół projektowy nie ma wpływu na środowisko programistyczne
Historia:	----
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UNP 02
Nazwa:	Adobe
Opis:	Dostarcza środowisko graficzne - Adobe Photoshop
Typ udziałowca:	Nieożywiony, pośredni
Punkt widzenia:	Deweloper środowiska graficznego
Reprezentant:	Spółka Adobe Systems
Ograniczenia:	----
Historia:	----
Wymagania:	----

KARTA UDZIAŁOWCA	
Identyfikator:	UNP 03
Nazwa:	Politechnika Gdańska
Opis:	Bierze udział w sprzedaży gotowego systemu
Typ udziałowca:	Nieożywiony, pośredni
Punkt widzenia:	Perspektywa ekonomiczna
Reprezentant:	prof. dr. hab. Józef Sienkiewicz
Ograniczenia:	----
Historia:	----
Wymagania:	----

4 Wymagania

4.1 Wymagania ogólne

KARTA WYMAGANIA

Identyfikator:	W01	Priorytet :	Pożądane
Nazwa:	Wymagania biznesowe		
Opis / uzasadnienie:	Aplikacja będzie bezpłatna, z możliwością donacji i nieinwazyjnymi reklamami. System zostanie opublikowany w ramach licencji Berkeley Software Distribution.		
Stabilność:	Musi zostać wybrany rodzaj licencji w przyszłości		
Udziałowiec:	Zespół projektowy.		
Odpowiedzialny:	Zespół projektowy.		
Historia zmian:	-----		

KARTA WYMAGANIA			
Identyfikator:	W02	Priorytet :	Ważny
Nazwa:	Ochrona kodu systemu operacyjnego		
Opis / uzasadnienie:	System powinien być zabezpieczony przed nieuprawnionym odczytem i modyfikowaniem danych oraz kodu.		
Stabilność:	To wymaganie nigdy nie ulegnie zmianie		
Udziałowiec:	Prawo Polskie		
Źródło:	Ustawy z dnia 29 sierpnia 1997 r. o ochronie praw osobowych		
Odpowiedzialny:	Zespół projektowy		
Wymagania powiązane:	F02		
Historia zmian:	-----		

KARTA WYMAGANIA			
Identyfikator:	W03	Priorytet :	Kluczowe
Nazwa:	Ograniczenie odpowiedzialności		
Opis / uzasadnienie:	Twórcy nie odpowiadają za niezgodne z prawem lub nieprawidłowe wykorzystanie i modyfikowanie stworzonego oprogramowania.		
Stabilność:	To wymaganie nigdy nie ulegnie zmianie		
Udziałowiec:	UOB 01, UOB 02, UOB 03, UOB 04		
Odpowiedzialny:	Kierownik zespołu - Michał Jagielski		
Historia zmian:	-----		

4.2 Wymagania funkcjonalne

4.2.1 Funkcje/usługi

KARTA WYMAGANIA			
Identyfikator:	F01	Priorytet :	Kluczowe
Nazwa:	Prezentacja danych wyjściowych.		
Opis / uzasadnienie:	Aplikacja po wykonaniu funkcjonalności F02 przedstawia dla użytkownika dane uzyskane z wykonania algorytmu		
Dane wejściowe:	Dane wyjściowe		
Warunki początkowe:	---		
Warunki końcowe:	Aplikacja wyświetla prawidłowy wynik - ekstremum lub wynik i wykres funkcji jeśli jest to możliwe		
Sytuacje wyjątkowe:	Ekstremum nie zostało znalezione		
Efekty uboczne:	-----		
Czynności równoczesne:	----		
Stabilność:	To wymaganie nie może ulec zmianie.		
Udziałowiec:	UOB 01 i UOB 02		
Źródło:	Pierwotny projekt.		
Odpowiedzialny:	Michał Jagielski		
Wymagania powiązane:	F02, F03, IO02, N02,		
Historia zmian:	----		

KARTA WYMAGANIA			
Identyfikator:	F02	Priorytet :	Krytyczny
Nazwa:	Obliczanie ekstremów funkcji		
Opis / uzasadnienie:	System ma za zadanie przyjąć pewne dane wejściowe i przetworzyć je przy użyciu odpowiednich algorytmów zwracając poprawny wynik		
Dane wejściowe:	Wzór funkcji / pojedyncze punkty i przedział poszukiwania		
Warunki początkowe:	Wzór funkcji jest poprawny i nie zawiera niedozwolonych znaków		
Warunki końcowe:	Dane o poszczególnych ekstremach są obliczone.		
Sytuacje wyjątkowe:	Funkcja jest bardzo skomplikowana i nie jest ciągła w zadanym przedziale, niepoprawny wzór funkcji, lub błąd Framework'u.		

Efekty uboczne:	----
Czynności równoczesne:	Informowanie użytkownika o trwającym procesie, przetwarzanie danych przez różne algorytmy i porównywanie wyników w celu znalezienia wyniku najlepszego
Stabilność:	To wymaganie może ulec jedynie rozszerzeniu warunków początkowych.
Udziałowiec:	UOB 01 , UOB02
Źródło:	Pierwotny projekt.
Odpowiedzialny:	Michał Jagielski
Wymagania powiązane:	F01
Historia zmian:	----

KARTA WYMAGANIA			
Identyfikator:	F03	Priorytet :	Wysoki
Nazwa:	Funkcjonowanie jako aplikacja.		
Opis / uzasadnienie:	System powinien działać jako aplikacja okienkowa z specjalnym i intuicyjnym interfejsem graficznym		
Dane wejściowe:	---		
Warunki początkowe:	Użytkownik włącza aplikację poprzez otwarcie pliku wykonywalnego EXE		
Warunki końcowe:	Poprawnie działający program z pełnią funkcjonalności.		
Sytuacje wyjątkowe:	Brak bibliotek, nie wspierany system operacyjny, niewystarczające uprawnienia użytkownika, niewystarczające wymagania sprzętowe.		
Efekty uboczne:	---		
Czynności równoczesne:	Obliczanie, przetwarzanie danych, rysowanie wykresów		
Stabilność:	To wymaganie nie może ulec zmianie.		
Udziałowiec:	UOB 01 , UOB 02		
Źródło:	Pierwotny projekt.		
Odpowiedzialny:	Michał Jagielski		
Wymagania powiązane:	F01, N02, WŚ01		
Historia zmian:	---		

4.2.2 Istniejąca infrastruktura

KARTA WYMAGANIA			
Identyfikator:	I 01	Priorytet :	Kluczowy
Nazwa:	Wprowadzanie danych		
Opis / uzasadnienie:	<p>Do poprawnego działania aplikacji użytkownik musi:</p> <p>dostarczyć wzór funkcji o odpowiednich warunkach:</p> <ul style="list-style-type: none"> -liczba niewiadomych nie może być większa niż podana, np. niedopuszczalne jest $f(x) = x*y$ -funkcja podana jest w poprawnym określonym przez aplikację formacie (błąd: $f[[x]] = [x];$) -funkcja nie jest dopuszczona z powodu zbyt dużej ilości zmiennych <p>dostarczyć punkty o odpowiednich warunkach:</p> <ul style="list-style-type: none"> -ilość punktów musi być większa niż 1 -punkty muszą być zgodne składniowo - jeśli wprowadzamy punkty o 2 współrzędnych niemożliwe jest podanie 1 punktu o 3 		
Dane wejściowe:	Poprawne dane, wprowadzone w dopuszczalny przez aplikację sposób		
Warunki początkowe:	Analiza wprowadzonych danych pod względem poprawności		
Warunki końcowe:	Zaakceptowanie podanych danych		
Sytuacje wyjątkowe:	Błąd przy analizie danych i prośba o ich poprawienie		
Efekty uboczne:	----		
Czynności równoczesne:	----		
Stabilność:	Wymagania dotyczące danych mogą ulec rozszerzeniu w przyszłości		
Udziałowiec:	UOB 01, UOB 02		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Pierwotny projekt		
Wymagania powiązane:	F 02 , N04		
Historia zmian:	----		

KARTA WYMAGANIA			
Identyfikator:	I 02	Priorytet :	Wysoki
Nazwa:	Wyświetlanie powiadomień o błędach		
Opis / uzasadnienie:	Użytkownik jest informowany o wystąpieniu sytuacji krytycznej, zgłoszeniu wyjątku lub dostarczeniu niepoprawnych danych		
Dane wejściowe:	----		
Warunki początkowe:	Wychwycenie informacji o ewentualnym błędzie		
Warunki końcowe:	Przedstawienie w sposób graficzny o błędzie w sposób zrozumiały dla użytkownika i umożliwiający jego ewentualne naprawienie		
Sytuacje wyjątkowe:	Pojawienie się nieprzewidzianego i nieopisanego błędu		
Efekty uboczne:	Awaryjne zamknięcie aplikacji		

Czynności równoczesne:	---
Stabilność:	<i>To wymaganie nie powinno ulec zmianie</i>
Udziałowiec:	<i>UOB 01 , UOB 02</i>
Źródło:	<i>Pierwotny projekt</i>
Odpowiedzialny:	<i>Zespół projektowy</i>
Wymagania powiązane:	<i>N 04</i>
Historia zmian:	----

4.2.3 Interfejs z otoczeniem

KARTA WYMAGANIA			
Identyfikator:	IO 01	Priorytet :	Wysoki
Nazwa:	<i>Obsługa aplikacji za pomocą myszki</i>		
Opis / uzasadnienie:	<i>Aplikacja powinna prawidłowo reagować na działania wykonywane za pomocą myszki np. wykonanie działania po kliknięciu w przycisk.</i>		
Dane wejściowe:	----		
Warunki początkowe:	----		
Warunki końcowe:	----		
Sytuacje wyjątkowe:	<i>Brak myszki</i>		
Efekty uboczne:	----		
Czynności równoczesne:	<i>obsługa klawiatury</i>		
Stabilność:	<i>Wymaganie nie ulegnie zmianie.</i>		
Udziałowiec:	<i>Zespół projektu</i>		
Źródło:	<i>Pierwotny projekt</i>		
Odpowiedzialny:	<i>Zespół projektu</i>		
Wymagania powiązane:	<i>F 01, F 02</i>		
Historia zmian:	----		

KARTA WYMAGANIA			
Identyfikator:	IO 02	Priorytet :	Wysoki
Nazwa:	<i>Obsługa aplikacji za pomocą klawiatury</i>		
Opis / uzasadnienie:	<i>Aplikacja powinna prawidłowo reagować na działania wykonywane za pomocą klawiatury np. wprowadzanie danych (funkcji, punktów)</i>		
Dane wejściowe:	----		
Warunki początkowe:	----		
Warunki końcowe:	----		
Sytuacje wyjątkowe:	<i>Brak klawiatury</i>		
Efekty uboczne:	----		
Czynności równoczesne:	<i>obsługa myszki</i>		
Stabilność:	<i>Wymaganie nie ulegnie zmianie.</i>		
Udziałowiec:	<i>Zespół projektu</i>		
Źródło:	<i>Pierwotny projekt</i>		
Odpowiedzialny:	<i>Zespół projektu</i>		
Wymagania powiązane:	<i>F 01, F 02</i>		
Historia zmian:	----		

4.3 Wymagania niefunkcjonalne

KARTA WYMAGANIA			
Identyfikator:	N 01	Priorytet :	Wysoki
Nazwa:	<i>Płynność i stabilność działania aplikacji</i>		
Opis / uzasadnienie:	<i>Czas wykonywania programu oraz jego użytkowanie powinno być płynne i jak najszybsze. Alokacja zasobów powinna być ograniczona do minimum, a wykorzystane zasoby powinny być zwolnione niezwłocznie po zamknięciu aplikacji.</i>		
Dane wejściowe:	----		
Warunki początkowe:	-----		
Warunki końcowe:	----		
Sytuacje wyjątkowe:	<i>Przepełnienie dostępnej pamięci</i>		
Efekty uboczne:	<i>Wyświetlenie informacji o braku pamięci</i>		
Czynności równoczesne:	----		
Stabilność:	<i>To wymaganie nie może ulec zmianie</i>		
Udziałowiec:	<i>Zespół projektu</i>		
Źródło:	<i>Pierwotny projekt</i>		
Odpowiedzialny:	<i>Zespół projektu</i>		
Wymagania powiązane:	-----		
Historia zmian:	----		

KARTA WYMAGANIA			
Identyfikator:	N 02	Priorytet :	Ważny
Nazwa:	Obsługa błędów		
Opis / uzasadnienie:	Użytkownik jest informowany o wystąpieniu sytuacji krytycznej, zgłoszeniu wyjątku lub dostarczeniu niepoprawnych danych		
Dane wejściowe:	----		
Warunki początkowe:	Oszacowanie z jakim błędem mamy doczynienia		
Warunki końcowe:	Wyświetlenie odpowiedniego komunikatu z opisem błędu		
Sytuacje wyjątkowe:	Nieznany błąd		
Efekty uboczne:	zamknięcie aplikacji }		
Czynności równoczesne:	----		
Stabilność:	To wymaganie nie może ulec zmianie		
Udziałowiec:	Zespół projektu		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Zespół projektu		
Wymagania powiązane:	IO 01, IO 02, I 02		
Historia zmian:	----		

KARTA WYMAGANIA			
Identyfikator:	N 03	Priorytet :	Ważny
Nazwa:	Intuicyjny interfejs użytkownika		
Opis / uzasadnienie:	Aplikacja nie powinna sprawiać problemów w jej użytkowaniu. Menu powinno być przejrzyste, niesprawiające problemów z obsługą. Każdy przycisk i funkcja powinna być dokładnie opisana w specjalnym podświetleniu pozwalającym pokazać pomoc do jego obsługi w postaci niezbędnych i krótkich informacji		
Dane wejściowe:	---		
Warunki początkowe:	---		
Warunki końcowe:	---		
Sytuacje wyjątkowe:	---		
Efekty uboczne:	----		
Czynności równoczesne:	----		
Stabilność:	To wymaganie nie może ulec zmianie		
Udziałowiec:	Zespół projektu		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Zespół projektu		
Wymagania powiązane:	----		
Historia zmian:	-----		

4.4 Wymagania dziedzinowe

KARTA WYMAGANIA			
Identyfikator:	WD 01	Priorytet :	Kluczowe
Nazwa:	Zgodność rezultatów z wartościami referencyjnymi		
Opis / uzasadnienie:	Wyniki algorytmu powinny być zbliżone z wynikami algorytmów obecnych na rynku.		
Dane wejściowe:	-----		
Warunki początkowe:	----		
Warunki końcowe:	----		
Sytuacje wyjątkowe:	---		
Efekty uboczne:	----		
Czynności równoczesne:	---		
Stabilność:	To wymaganie nie może ulec zmianie		
Udziałowiec:	OB 01		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Zespół projektu		
Wymagania powiązane:	F 02		
Historia zmian:	----		

4.5 Wymagania na środowisko docelowe

KARTA WYMAGANIA			
Identyfikator:	WŚ 01	Priorytet :	Ważne
Nazwa:	System operacyjny		
Opis / uzasadnienie:	Oprogramowanie powinno działać na systemie Windows 10 (x86 , x64)		
Dane wejściowe:	----		
Warunki początkowe:	Całkowicie sprawny system operacyjny z wszystkimi potrzebnymi bibliotekami		
Warunki końcowe:	Poprawne działanie aplikacji		

Sytuacje wyjątkowe:	Błędy związane z brakiem bibliotek lub błędy niezgodności systemowej
Efekty uboczne:	Aplikacja nie zostaje włączona
Czynności równoczesne:	Stabilne funkcjonowanie innych procesów systemowych
Stabilność:	Możliwe rozszerzenie na inne systemy
Udziałowiec:	----
Źródło:	Pierwotny projekt
Odpowiedzialny:	Zespół projektu
Wymagania powiązane:	----
Historia zmian:	-----

4.6 Wymagania dotyczące procesu wytwarzania

KARTA WYMAGANIA			
Identyfikator:	WPW 01	Priorytet :	Pożądane
Nazwa:	System kontroli wersji		
Opis / uzasadnienie:	Podczas wytwarzania oprogramowania w ramach systemu kontroli wersji będziemy korzystali z GIT (GitHub)		
Dane wejściowe:	-----		
Warunki początkowe:	Puste repozytorium		
Warunki końcowe:	Repozytorium zawierające wytworzone oprogramowanie, dokumentację oraz zestaw instrukcji użytkownika		
Sytuacje wyjątkowe:	-----		
Efekty uboczne:	-----		
Czynności równoczesne:	-----		
Stabilność:	-----		
Udziałowiec:	OB 01		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Zespół projektu		
Wymagania powiązane:	-----		
Historia zmian:	-----		

KARTA WYMAGANIA			
Identyfikator:	WPW 02	Priorytet :	Pożądane
Nazwa:	Sposób wytwarzania oprogramowania		
Opis / uzasadnienie:	Wytwarzanie prototypu oprogramowania będzie trwało od października 2017 do grudnia 2018. Oprogramowanie będzie produkowane z wykorzystaniem technik przedstawionych w innych dokumentach.		
Dane wejściowe:	Dokumentacja: RW, ZP, CASE, SWS, AWS, PS		
Warunki początkowe:	Kompletna dokumentacja		
Warunki końcowe:	Działający prototyp		
Sytuacje wyjątkowe:	Opóźnienia w harmonogramie, problemy personalne		
Efekty uboczne:	Opóźnienia w powstaniu aplikacji		
Czynności równoczesne:	Dokumentowanie wytwarzanego programu		
Stabilność:	Zmiany są możliwe. Mogą wynikać z problemów personalnych lub problemów z dotrzymywaniem terminów podanych w harmonogramie		
Udziałowiec:	OB 01, OB 02		
Źródło:	Pierwotny projekt		
Odpowiedzialny:	Zespół projektu		
Wymagania powiązane:	-----		
Historia zmian:	-----		

5 Kryteria akceptacji rozwiązania

-Aplikacja jest intuicyjna.

-Aplikacja poprawnie spełnia powierzone jej zadania (oblicza miejsca zerowe wybranymi algorytmami).

-Aplikacja pozbawiona jest błędów uniemożliwiających jej poprawne działanie.

-Aplikacja uruchamia się na każdej platformie.

Podsumowując aplikacja powinna działać i spełniać swoje założenia, a przy tym być przyjazna dla klienta. Wszystkie wymienione funkcjonalności powinny zostać zaimplementowane w określonym czasie i przekazane do oceny.