



# PROJEKT INŻYNIERSKI

## Edytor graficzny systemów rozmytych dla języka Python

ID projektu - 46300

Opiekun projektu - dr inż. Jerzy Dembski

Dokument nr 2: Scrum: Backlog produktu.

### Streszczenie projektu:

Celem projektu jest tworzenie edytora graficznego systemu rozmytego z wykorzystaniem dowolnej biblioteki Pythona (np. pygame, opencv, opengl) pozwalającego na tworzenie i kształtowanie zbiorów rozmytych, definiowanie reguł rozmytych, wizualizację działania systemu dla zadanych wartości wejściowych oraz uczenie systemu na podstawie danych uczących metodą ANFIS wraz z przedstawieniem systemu w postaci wielowarstwowego modelu neuronowego do dalszego uczenia. System powinien pozwalać też na zapis i odczyt systemu rozmytego z pliku tekstowego, jak również przedstawienia go jako funkcji przetwarzającej dane wejściowe.

### Streszczenie dokumentu:

Celem dokumentu jest przybliżenie projektu. Ma on na celu opisanie jego praktycznych zastosowań poprzez przedstawienie scenariuszy użycia, persony modelowych użytkowników końcowych oraz backlogu produktu. Ma również na celu określenie kryteriów akceptacji oraz definicji ukończenia.

Wersja:	1.3
Data wydania:	03.04.2025
Redaktor:	Adam Zarzycki
Współautorzy:	Filip Wesołowski, Julian Kulikowski
Etap/zadanie:	2
Nazwa pliku:	ScrumBP_v1.3.pdf
Liczba stron:	18



## Historia zmian

Wersja	Data	Opis zmiany
1.0	27.03.2025	Dodanie punktów 1 oraz 2.1
1.1	28.03.2025	Dodanie punktów 2.2 oraz 3.1
1.2	01.04.2025	Uzupełnienie punktu 3. i dodanie punktu 4.
1.3	03.04.2025	Dodanie punktów 5. i 6.





## SPIS TREŚCI

<b>1. O projekcie i produkcji</b>	<b>4</b>
<b>2. Persony użytkowników</b>	<b>5</b>
2.1. Student Politechniki Gdańskiej	5
2.2. Pracownik Katedry Inteligentnych Systemów Interaktywnych	6
<b>3. Scenariusze użycia</b>	<b>9</b>
3.1. Skromne początki	9
3.2. EUREKA!	9
3.3. Delikatne wprowadzenie	10
<b>4. Backlog produktu</b>	<b>12</b>
4.1. Stan backlogu produktu	12
4.2. Opis backlogu produktu	13
4.2.1. Identyfikator	13
4.2.2. Story Points	13
4.2.3. Priorytet	13
4.2.4. Sortowanie backlogu produktu	13
<b>5. Kryteria akceptacji</b>	<b>14</b>
5.1. Kryteria akceptacji dla wybranych elementów backlogu	14
5.1.1. Jakość podobna do wersji matlabowej	14
5.1.2. Edycja wykresów poprzez przeciąganie na ekranie	15
5.1.3. Eksport systemu rozmytego do pliku	16
5.1.4. Wnioskowanie Mamdaniego i Sugeno	16
5.2. Ogólne kryteria akceptacji/gotowości do wdrożenia produktu	17
<b>6. Definicja ukończenia (DoD)</b>	<b>18</b>

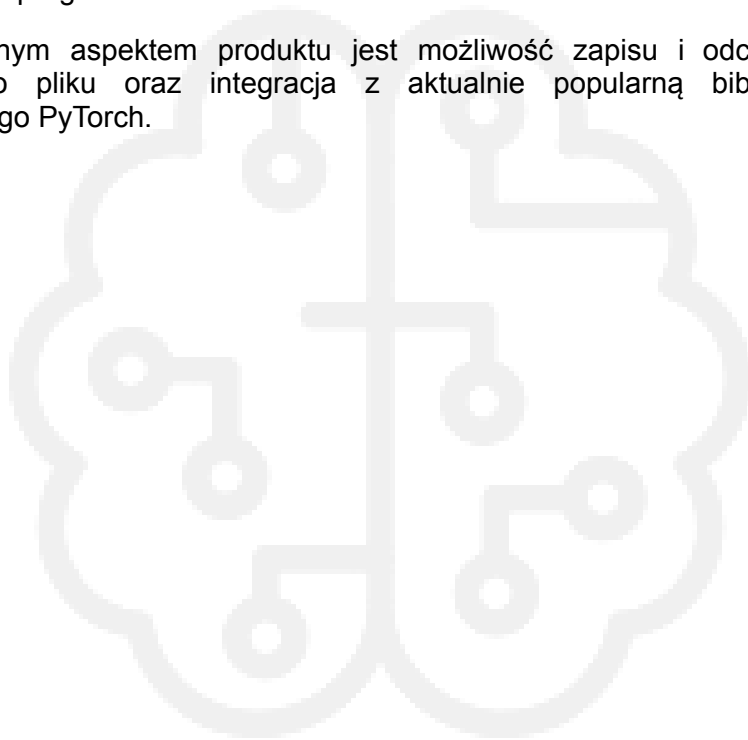


# 1. O projekcie i produkcji

Celem końcowym projektu jest utworzenie aplikacji umożliwiającej wizualizację zbiorów rozmytych za pomocą interfejsu graficznego w języku Python. Ważnym aspektem produktu jest udostępnienie darmowej alternatywy dla programu Fuzzy Logic Designer operującego w płatnym środowisku MATLAB.

Najważniejszym zadaniem produktu jest stworzenie prostego, przystępnego interfejsu graficznego, umożliwiającego użycie aktualnych implementacji logiki rozmytej. Użytkownik powinien być w stanie określać oraz edytować reguły rozmyte, a następnie obserwować ich efekt oraz podejmowanie decyzji w czasie rzeczywistym. W przeciwieństwie do aktualnych rozwiązań korzystających z wiersza poleceń, naszym priorytetem jest obserwowalność wyników, kontrolowalność kolejnych kroków oraz przystępność dla niedoświadczonych użytkowników. Powinien ich naprowadzać oraz uczyć w naturalny sposób, nawet jeżeli nigdy wcześniej nie pracowali z podobnymi programami.

Innym ważnym aspektem produktu jest możliwość zapisu i odczytu aktualnego projektu do pliku oraz integracja z aktualnie popularną biblioteką uczenia maszynowego PyTorch.





## 2. Persony użytkowników

### 2.1. Student Politechniki Gdańskiej



**Imię i nazwisko:** Adrian Kaczmarek

**Edukacja:** Studia pierwszego stopnia

**Stanowisko:** Student

**Branża:** Informatyka

**Wiek:** 21

**Używane technologie:** Windows 11, PyCharm

#### Historia:

Adrian jest studentem Politechniki Gdańskiej na kierunku informatyka pierwszego stopnia, aktualnie znajduje się na czwartym semestrze. Najbardziej komfortowo czuje się programując w języku programowania Python, ponieważ w liceum był to jedyny język, którego się uczył. Niedawno na wykładzie dowiedział się o zagadnieniu logiki rozmytej i jej zastosowań w uczeniu maszynowym, jako że jest to obszar, w którym chciałby pracować w przyszłości, a także jego zainteresowań, postanowił zgłębić temat na własną rękę.

Adrian nie ma wcześniejszego doświadczenia z logiką rozmytą oraz zbiorami rozmytymi, nigdy wcześniej nie korzystał z żadnego oprogramowania, w którym mógłby stworzyć własny projekt. Nie ma także szczególnego doświadczenia z bibliotekami wywoływanymi z poziomu wiersza poleceń, do tej pory zazwyczaj pracował za pomocą zintegrowanego środowiska programistycznego PyCharm.

Mieszka daleko za Gdańskiem, przeznacza dużą część swojego dnia na dojazdy na Politechnikę, a następnie powrót do domu. Co za tym idzie zazwyczaj na własne projekty może poświęcić czas spędzany w komunikacji miejskiej, pomiędzy zajęciami dydaktycznymi na politechnice lub późnymi wieczorami w domu. Mimo to jest zdeterminowany, aby poznawać nowe technologie i ich zastosowania.

#### Cele:

- Poznanie podstaw logiki rozmytej oraz zbiorów rozmytych,
- Tworzenie własnych projektów prywatnych we własnym zakresie,
- Wykorzystanie zbiorów rozmytych jako zbiorów danych dla sztucznej inteligencji,
- Zdobycie wiedzy, którą będzie mógł wykorzystać w przyszłości,
- Możliwość tworzenia bardziej skomplikowanych projektów w tym samym oprogramowaniu.

#### Obawy:



- Dostępne oprogramowanie jest płatne lub trudnodostępne,
- Oprogramowanie jest ukierunkowane na osoby, które miały wcześniej doświadczenie ze zbiorami rozmytymi,
- Oprogramowanie jest niekompatybilne z biblioteką PyThor,
- Oprogramowanie będzie ukrywać informacje, w tym informacje o implementacji, utrudniając naukę,
- Interfejs użytkownika będzie nieintuicyjny prowadząc to niezamierzonych działań bez możliwości cofnięcia błędu.

### Potrzeby:

- znalezienie przystępnego, intuicyjnego oprogramowania, która pozwala na naukę obsługi zbiorów rozmytych krok po kroku,
- znalezienie oprogramowania kompatybilnego z językiem Python,
- znalezienie oprogramowania oferującego interfejs graficzny,
- znalezienie oprogramowania, które nie wymaga zakupu licencji.

### Mile widziane:

- skróty klawiszowe przyspieszające pracę,
- poradnik i zbiór przykładów dotyczących poszczególnych funkcji,
- możliwość załadowania projektów z oraz do środowiska MATLAB.

## 2.2. Pracownik Katedry Inteligentnych Systemów Interaktywnych



**Imię i nazwisko:** Jan Maćczak

**Edukacja:** Doktor inżynier

**Stanowisko:** Profesor politechniki

**Branża:** Informatyka

**Wiek:** 45

**Używane technologie:** Ubuntu, VS Code

### Historia:



Profesor Maćczak od piętnastu lat naucza na Politechnice Gdańskiej, jego głównym obszarem pracy i zainteresowań jest sztuczna inteligencja. Na bieżąco śledzi trendy i nowinki na jej temat w świecie informatycznym, w ostatnich latach zauważył szczególnie szybki i nagły rozwój tej technologii. Zmusiło go to do refleksji na temat przedmiotów, których naucza i, mając na uwadze dobro swoich studentów, doszedł do wniosku, iż program, którego naucza jest nieaktualny. Studenci aktualnie nie są uczeni na temat logiki rozmytej oraz jej niuansów w uczących zbiorach wejściowych. Po rozmowach ze studentami również dowiedział się, że te kwestie, są dla nich mniej jak technologia i bardziej jak magia.

Mając dobrobyt swoich studentów na uwadze postanowił uaktualnić program studiów i wprowadzić laboratorium uczące zbiorów rozmytych jako preludium do sztucznej inteligencji. W chwili obecnej poszukuje narzędzi, które nadawałyby się do tego celu. Jako osoba mająca doświadczenie z tą technologią oraz wielki fan Linuxa, osobiście nie przeszkadza mu korzystanie z implementacji polegających na wierszu poleceń, martwi go jednak ile studentów by wynieśli z tego laboratorium. Z drugiej strony jedyną implementacją graficzną jaką do tej pory znalazł jest implementacja w środowisku MATLAB. Profesor Maćczak martwi się kwestią licencji oraz mocno preferuje inicjatywy Open Source, ale nawet jemu trudno zaprzeczyć zaletę tego rozwiązania.

Oprócz tego poza godzinami zajęć profesor Maćczak wykonuje własne projekty naukowe korzystające z logiki rozmytej. Zazwyczaj wykorzystuje ją w celu uczenia maszynowego, na przykład definiuje reguły rozmyte do nauki sztucznej inteligencji gry w Tetrisa. Nie przeszkadza mu korzystanie z implementacji w wierszu poleceń, aczkolwiek czasami trudno jest wykryć błąd w rozumowaniu. Sprawia to, że nie zostaje on znaleziony, aż jest za późno i należy powtórzyć cały proces. Czasami pragnie, aby istniało narzędzie pozwalające przyjrzeć się procesowi rozumowania krok po kroku.

### **Cele:**

- Wprowadzenie zajęć dydaktycznych z logiki rozmytej,
- Tworzenie zbiorów danych dla sztucznej inteligencji,
- Promowanie narzędzia alternatywnego dla technologii MATLAB,
- Łatwiejsze wykrywanie ewentualnych błędów.

### **Obawy:**

- Ukierunkowanie na interfejs graficzny i osoby niedoświadczone spowolni pracę,
- Brak ważnych skrótów klawiszowych,
- Brak funkcjonalności lub inna implementacja funkcji, do których przywykł,
- Oprogramowanie nie będzie nadawać się do zajęć dydaktycznych.

### **Potrzeby:**

- Ukierunkowanie na studentów,
- Oprogramowanie pomaga w nauce,
- Kompatybilność z biblioteką PyTorch,



- Oferowanie skrótów klawiszowych ułatwiających pracę dla doświadczonych użytkowników,
- Wsparcie odczytu danych z jego starszych projektów,
- Odwzorowanie funkcji znanych mu z innych programów.

**Mile widziane:**

- Program będący inicjatywą Open Source,
- Wsparcie dla starszych wersji systemów operacyjnych,
- Materiały uczące i poradniki dla studentów.







## 3. Scenariusze użycia

### 3.1. Skromne początki

Adrian Kaczmarek chciałby móc powiedzieć, że zaliczenie trzeciego etapu projektu z Metod Numerycznych zakończyło koszmar, jakim było wykorzystywanie oprogramowania MATLAB przez ostatnie cztery miesiące. Niefortunnym zrządzeniem losu jednak Adrian zainteresował się tematyką zbiorów rozmytych, a jedyną platformą oferującą dostateczne możliwości w tym polu był właśnie on. Jednak pewnego dnia, po zwierzeniu się ze swojej bolączki doktorowi Maćczakowi, ten zaoferował mu dostęp do nowego edytora graficznego w języku Python, który stworzyli podopieczni jego kolegi z katedry w ramach projektu inżynierskiego. Życie Adriana zmieniło się nie do poznania.

Po powrocie do domu, Adrian uruchomił otrzymany program. Od razu w oczy rzucił mu się estetyczny, minimalistyczny interfejs, jak również stonowana szata graficzna. Będąc nowym użytkownikiem systemu, Adrian wybrał opcję pomocy. System od razu wyświetlił mu najważniejsze informacje dotyczące poruszania się po programie. Na podstawie uzyskanych informacji studentowi udało się znaleźć kreator nowego projektu. Adriana zaskoczyło, jak proste jest utworzenie nowego projektu - wpisał na prośbę kreatora pożądany typ wnioskowania (na razie Mamdaniego, myślał, Sugeno zajmę się później), zmienne i reguły startowe, po czym zatwierdził swój wybór.

Adrian zdziwił się, gdy przed oczami ukazał mu się wykres bardzo podobny do tego w MATLABie, i to już po parunastu sekundach (korzystając z MATLABa, zdążyłby zaparzyć sobie herbatę). Na początku myślał, że musi się to wiązać z pewną utratą jakości względem pierwowzoru, ale nie - wykresy były tak samo interaktywne i łatwe w ręcznej manipulacji jak wcześniej.

Adrian postanowił przeprowadzić ostateczny test - czy program ten jest w stanie załadować system rozmyty, który z największym trudem rozwijał w MATLABie przez ostatnie sześć miesięcy. W tym celu wybrał opcję importowania, po czym załączył swój system w formacie .zip. Jaka była jego radość, gdy zobaczył znajome wykresy i zmienne w nowo utworzonym oknie projektu.

Następnego dnia w drodze na uczelnię Adrian siedział pochłonięty w programie, ucząc się skrótów klawiszowych do menu i, oddając cześć niepisanej tradycji, zmieniając domyślną szatę graficzną na tryb ciemny. Uniósł na chwilę głowę, chcąc sprawdzić w telefonie, ile kilometrów dzieli go jeszcze od Politechniki Gdańskiej, i jakież było jego zdziwienie, gdy zobaczył znajomy dach starego budynku wydziału ETI na horyzoncie. Spanikowany, wybrał opcję eksportowania projektu, nie wierząc jednak, że zdąży ocalić dokonane przez siebie zmiany. Gdy jednak w wybranej lokalizacji pojawił się nowy plik zanim musiał wyjść z autobusu, Adrian już wiedział, że jego znajomość z MATLABem dobiegła końca.

### 3.2. EUREKA!

Doktor inżynier Jan Maćczak był człowiekiem, który nie znosił stania w miejscu. Gdy tylko zauważył, że jego program nauczania lub wykorzystywaną technologię można poprawić, od razu budził się w nim zew innowacyjności. Właśnie z tego powodu dr Maćczak przez długi czas uważał nauczanie systemów rozmytych za swoje osobiste nemezis - na rynku brak było lepszych alternatyw dla MATLABa, z którym styczność wywoływała u większości studentów gniew prawie tak palący jak komputer po godzinie jego wykorzystywania. Pewnego dnia, jego kolega z katedry udostępnił mu program napisany przez jego



studentów w ramach pracy inżynierskiej. Dr Maćczak niezwłocznie przystąpił do testów.

Dr Maćczak, jako zaawansowany użytkownik MATLABowego oprogramowania do systemów rozmytych, z drobną satysfakcją przyjął fakt, że interfejs programu jest bardzo podobny do jego starszego odpowiednika. To oraz jego wcześniejsza wiedza pozwoliły mu szybko utworzyć nowy projekt bez wykorzystania kreatora. Już w widoku projektu, wykładowca zdefiniował nowe zmienne rozmyte i reguły wnioskowania, jak również wybrał pożądany typ wnioskowania i typ wykresu funkcji przynależności (gaussowski, jako że wykresy trójkątne, trapezowe i dzwonowe nie mają znanego matematyka w swojej nazwie).

Już miał uznać testowany program za miłą, lecz jednak mało potrzebną mu w dydaktyce aplikację, gdy, przeglądając opcje dostępne w szybkim menu programu, natrafił na "animuj przebieg wnioskowania". Nie wierząc własnym oczom, doktor Maćczak wybrał tę opcję, po czym patrzył w zachwycie jak po każdym jego kliknięciu, w osobnym oknie po kolei podświetlane były obecnie wykonywane reguły rozmyte, w czasie rzeczywistym modyfikujące wykres funkcji dopasowania wynikowego. Podglądając wartości zmiennych na kolejnych etapach wykonania wnioskowania, doktor pomyślał z jaką łatwością jego studenci będą teraz w stanie zrozumieć, co właściwie zrobili przez 90 minut na laboratorium, i że wreszcie nie będzie musiał spędzać 3 dodatkowych godzin na poprawianie prostych błędów na egzaminie praktycznym.

Nazajutrz, korzystając z mniejszego obciążenia zajęciami, doktor Maćczak postanowił wypróbować możliwości oprogramowania w zakresie integracji jego starszych projektów. Wybrał opcję importowania projektu, załadował jednego z potencjalnych kandydatów na przeciążenie systemu, i ku swej uldze ujrzał przed sobą poprawnie wyświetlające się wykresy i zmienne. Wtem powiadomienie w jego telefonie przypomniało mu o przerzuceniu pewnego systemu rozmytego na sieć neuronową, które zobowiązał się zrealizować do jutra dla swojego kolegi z katedry. Zażenowany takim obrotem spraw, doktor Maćczak z rezygnacją przejechał myszką nad ikonkę udostępniającego taką funkcję MATLABa, gdy w jego głowie zaświtała pewna nadzieja. W pasku wyszukiwania programu wpisał hasło, które, jak sądził, powinno znaleźć to, czego szukał, i faktycznie - program zaproponował mu opcję "wyeksportuj system rozmyty do sieci neuronowej". Uradowany, doktor Maćczak wybrał tę opcję, by po paru minutach i jednym mailu znów móc zagłębić się w świat systemów rozmytych.

Następnego dnia, dr Maćczak został z nienacką poproszony o zastępstwo na wykładzie z przedmiotu Sztuczna Inteligencja, którego tematem miała być "logika pierwszego rzędu". Wiedząc, że jakiegokolwiek zainteresowanie studentów tematem może być najtrudniejszym zadaniem w jego karierze, doktor postanowił natychmiast zacząć przygotować się do wykładu. Informacja ta zastała go w środku poważnej edycji swojego systemu rozmytego, jednak dr Maćczak, za pomocą dogłębnie poznanych przez ostatnie dwa dni skrótów klawiszowych, szybko zapisał zmiany, wyeksportował system do pojedynczego katalogu i zamknął program, po czym wziął się do pracy.

### 3.3. Delikatne wprowadzenie

Profesor Maćczak jest znany na politechnice jako autorytet od wszystkiego dotyczącego systemów rozmytych. Uznaje się, że wie on wszystko co jest do wiedzenia w tej dziedzinie, od teorii po niuanse różnych implementacji. Nic dziwnego, więc że pewnego dnia na konsultacjach przyszedł do niego student z prośbą o wskazanie dobrego oprogramowania dla osób początkujących w tej dziedzinie.

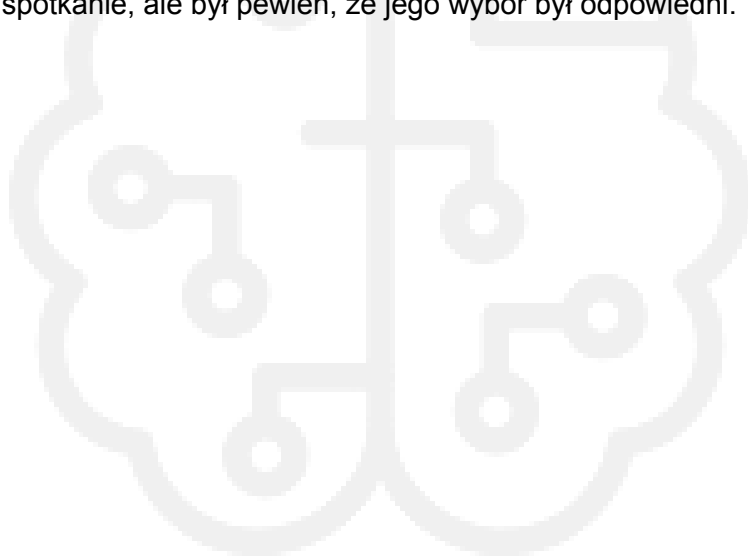
Doktor w ogóle nie musiał się zastanawiać o tym co by było najodpowiedniejsze dla  
Edytor graficzny systemów rozmytych dla języka Python



studenta drugiego roku informatyki. Kogoś kto nigdy wcześniej nie miał kontaktu z różnymi implementacjami, i jak wiedział z doświadczenia nabytym będąc profesorem na uczelni, potrzebował jak najwięcej pomocy. Sam od niedawna korzystał z implementacji, która te kwestie miała w szczególności na uwadze i pamiętał jak dobrze przeprowadzała przez wszystko co konieczne nowego użytkownika.

Szybko odnalazł charakterystyczną stronę główną projektu, dobrze opisaną i opatrzoną w łatwe do znalezienia odnośniki do wszystkich najważniejszych miejsc. Sprawnie przeszedł do zakładki z plikami do pobrania i wybrał odpowiedni plik instalacyjny. Podczas lat nauczania zauważył, że studenci stają się nerwowi, gdy muszą sami zbudować biblioteki czy programy od podstaw, więc prosta instalacja za pomocą instalatora jest dużą zaletą podczas polecania programu innym.

Następnie włączył swój komputer, aby zademonstrować najprostsze funkcjonalności programu. Co było najbardziej imponujące to jego szeroka możliwość personalizacji. Większość współczesnych implementacji logiki rozmytej operują na poziomie wiersza poleceń. W tego typu programach personalizacja nie istnieje, pracuje się w trybie tekstowym albo nie pracuje się w ogóle. Maćczak był świadom, że jego student na pewno szukał czegoś na własną rękę i przyszedł do niego tylko dlatego, że poszukiwania nie zakończyły się sukcesem. Profesor pokazał możliwość przesuwania rozszerzania oraz zważania wybranych okienek z informacjami, opcję dostosowania kolorów. Z zadowoleniem patrzył na rosnące zachwycenie studenta. Niestety profesor Maćczak musiał kończyć spotkanie, ale był pewien, że jego wybór był odpowiedni.





## 4. Backlog produktu

### 4.1. Stan backlogu produktu

Q -status:Backlog,Ready,"In progress","In review",Done					34	Discard	Save
Title	...	Story Points	≡ 1 2	...	PB Priority		
1  Cel Produktu - Stworzenie edytora graficznego systemów rozmytych dla języka Python, ofer... #5		100			1:MUST		
2  Animacja wykonywania reguł wnioskowania krok po kroku #10		40			1:MUST		
3  Eksport systemu rozmytego do pliku #4		20			1:MUST		
4  Import systemu rozmytego z pliku, w tym z pliku .zip #6		20			1:MUST		
5  Edycja wykresów poprzez przeciąganie na ekranie #9		13			1:MUST		
6  Modyfikacja wartości zmiennych w trakcie wykonywania wnioskowania #11		13			1:MUST		
7  Jakość podobna do wersji MATLABowej #26		13			1:MUST		
8  Wnioskowanie Mamdaniego i Sugeno #20		8			1:MUST		
9  Definiowanie zmiennych i reguł rozmytych #22		8			1:MUST		
10  Obsługa projektów tworzonych w MATLABie #29		8			1:MUST		
11  Zapisywanie zmian #15		5			1:MUST		
12  Tworzenie nowego projektu #28		3			1:MUST		
13  Wyświetlanie wykresów #18		2			1:MUST		
14  Wybór spośród 4 typów wykresów: dzwonowego, gaussowskiego, trójkątnego, trapezowego #19		2			1:MUST		
15  Kreator nowych projektów #21		13			2:SHOULD		
16  Wysoka wydajność importu projektu #30		13			2:SHOULD		
17  Szybki zapis do pliku #17		5			2:SHOULD		
18  Minimalizacja zmian interfejsu względem interfejsu MATLABowego #27		3			2:SHOULD		
19  Intuicyjne menu #37		3			2:SHOULD		
20  Możliwość ustawiania/przesuwania okien na ekranie #8		2			2:SHOULD		
21  Pomoc dla użytkownika #25		2			2:SHOULD		
22  Rozszerzanie/zwężanie okienek #33		2			2:SHOULD		
23  Tryb jasny/ciemny #7		1			2:SHOULD		
24  Dodanie skrótów klawiszowych #12		1			2:SHOULD		
25  Estetyczny, minimalistyczny interfejs #16		1			2:SHOULD		
26  Łatwa instalacja z pliku wykonywalnego #23		40			3:COULD		
27  Eksport systemu rozmytego do sieci neuronowej #13		20			3:COULD		
28  Import systemu rozmytego z sieci neuronowej #14		20			3:COULD		
29  Integracja z biblioteką PyTorch #35		20			3:COULD		
30  Zamykanie programu skrótem klawiszowym #31		2			3:COULD		
31  Samouczek do programu #36		2			3:COULD		
32  Stonowana szata graficzna #24		1			3:COULD		
33  Dostosowywanie kolorów interfejsu #32		1			3:COULD		
34  Automatyczna generacja reguł rozmytych #34		20			4:WONT		



## 4.2. Opis backlogu produktu

### 4.2.1. Identyfikator

Identyfikatorem elementu jest numer GitHub Issue (liczba po znaku '#').

### 4.2.2. Story Points

Story Points szacowane są zgodnie ze skalą M. Cohna. Są to liczby ze zbioru:

{1, 2, 3, 5, 8, 13, 20, 40, 100}

Im większa liczba, tym większa jest złożoność/trudność wykonania elementu.

### 4.2.3. Priorytet

Priorytet nadawany jest zgodnie z techniką MoSCoW:

- MUST - elementy kluczowe,
- SHOULD - elementy przydatne,
- COULD - elementy dodatkowe,
- WONT - elementy niepotrzebne.

### 4.2.4. Sortowanie backlogu produktu

Backlog produktu sortowany jest najpierw malejąco względem priorytetu danego elementu (w celu podkreślenia elementów najważniejszych z punktu widzenia akcjonariuszy). W tym celu do nazw priorytetów wynikających z techniki MoSCoW dodano przedrostki numeryczne:

- MUST -> 1:MUST,
- SHOULD -> 2:SHOULD,
- COULD -> 3:COULD,
- WONT -> 4:WONT.


Dodatkowo, wyniki tego sortowania są sortowane względem przypisanych im Story Points, również malejąco.



## 5. Kryteria akceptacji

### 5.1. Kryteria akceptacji dla wybranych elementów backlogu

#### 5.1.1. Jakość podobna do wersji matlabowej

 **Jakość podobna do wersji MATLABowej #26**  
IkeaSzark/RPI

**Interfejs użytkownika zawiera wszystkie kluczowe sekcje dostępne w MATLAB-ie:**

- **Edytor zmiennych lingwistycznych:**
  - Możliwość dodawania, edytowania i usuwania zmiennych rozmytych wejściowych i wyjściowych.
  - System powinien umożliwiać szybkie wprowadzanie nowych zmiennych, z opcją ustawienia zakresu wartości i typu funkcji przynależności.
- **Definiowanie funkcji przynależności:**
  - Użytkownik może wybierać spośród dostępnych funkcji przynależności, takich jak trójkątne, trapezowe, gaussowskie i inne.
  - Wybór funkcji przynależności odbywa się za pomocą intuicyjnego interfejsu, który umożliwia łatwą edycję parametrów funkcji.
- **Reguły rozmyte:**
  - Użytkownik ma możliwość definiowania reguł IF-THEN za pomocą edytora graficznego lub tabelarycznego.
  - Reguły są automatycznie synchronizowane z odpowiednimi zmiennymi i funkcjami przynależności.

**Układ interfejsu powinien być intuicyjny i podobny do MATLAB-a, z podziałem na:**

- **Lewy panel:**
  - Wyświetla listę zmiennych wejściowych i wyjściowych, umożliwiając szybkie przełączanie między nimi.
  - Panel powinien być nawigacyjny i szybki, umożliwiając użytkownikowi szybkie dodawanie, edytowanie lub usuwanie zmiennych.
- **Główna sekcja:**
  - Zawiera edytor funkcji przynależności, reguł oraz interaktywną wizualizację wyników.
  - Wizualizacja wyników odbywa się w czasie rzeczywistym, automatycznie aktualizując wykresy po każdej zmianie.
- **Dolny panel:**
  - Przeznaczony do testowania systemu, wprowadzania danych testowych oraz przeglądania wyników w czasie rzeczywistym.
  - System umożliwia uruchamianie symulacji z natychmiastowym wyświetlaniem wyników.

**Wydajność systemu:**

- Interfejs musi działać płynnie bez opóźnień przy edycji wykresów, zmiennych i reguł.
- **Wydajność:** System powinien działać bez opóźnień przy wprowadzaniu zmian, a czas reakcji na każde kliknięcie lub edycję nie powinien przekraczać 500 milisekund.





### 5.1.2. Edycja wykresów poprzez przeciąganie na ekranie

## Edycja wykresów poprzez przeciąganie na ekranie #9

Open IkeaSzark/RPI Public

ReadySetGet opened 2 days ago · edited by Julku11 Edits ▾ ...

**Płynne i responsywne przesuwanie punktów oraz całego wykresu:**

- System umożliwia użytkownikowi intuicyjne przesuwanie punktów i wykresu w czasie rzeczywistym.
- Przesuwanie punktów i wykresu odbywa się bez opóźnień, zapewniając płynne doświadczenie użytkownika.
- Zmiany są natychmiastowo widoczne, co pozwala na precyzyjne dostosowanie wykresu.

**Automatyczna aktualizacja wykresu podczas przeciągania:**

- W trakcie przeciągania punktów lub wykresu, system automatycznie przelicza i aktualizuje dane na wykresie.
- Aktualizacje wykresu są widoczne od razu, co pozwala użytkownikowi na bieżąco śledzić zmiany w danych i ich reprezentacji.

**Możliwość cofnięcia zmian (funkcja „Cofnij”) oraz ich przywrócenia (funkcja „Przywróć”):**

- Użytkownik może w dowolnym momencie cofnąć ostatnią zmianę (np. przesunięcie punktu, edycję danych).
- Funkcja „Cofnij” pozwala na przywrócenie poprzedniego stanu wykresu bez utraty danych.
- Funkcja „Przywróć” umożliwia ponowne zastosowanie cofniętej zmiany, co daje elastyczność w edycji wykresu.

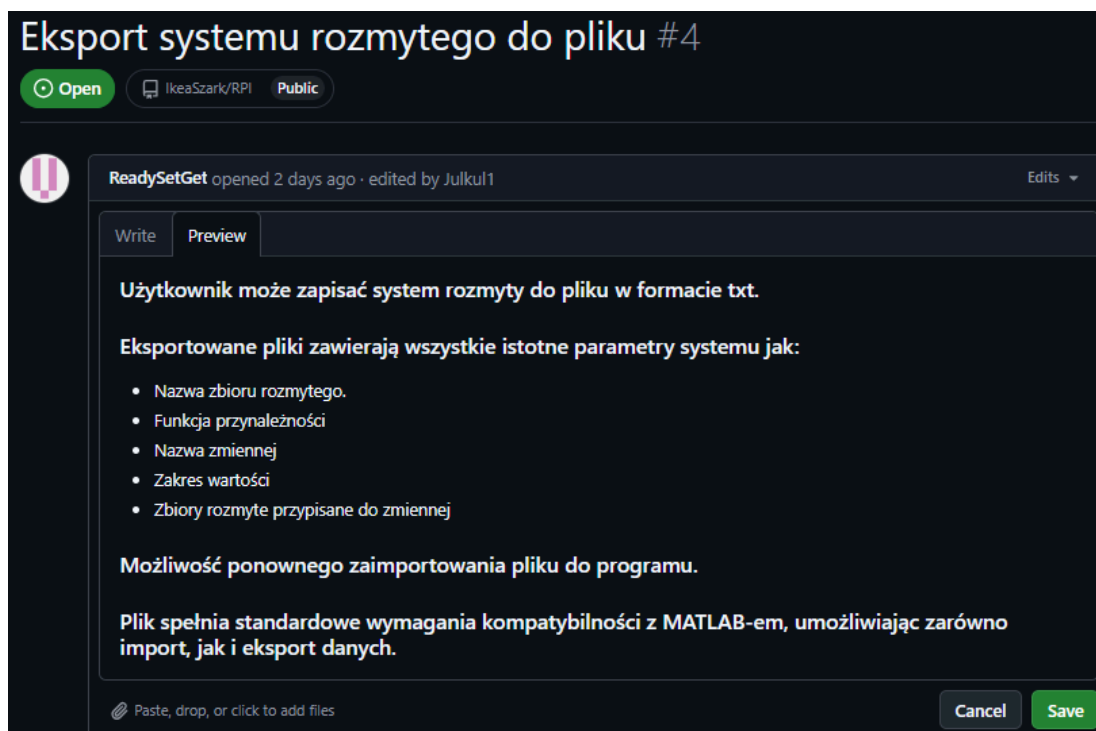
**Automatyczna aktualizacja parametrów wykresu podczas edycji:**

- Parametry wykresu, takie jak skala osi, wartości punktów, etykiety i jednostki miary, są automatycznie dostosowywane do wprowadzanych zmian.
- W miarę edycji danych, system zapewnia spójność wizualną i logiczną wykresu.
- Aktualizacje są dokonywane w czasie rzeczywistym, bez potrzeby ręcznego odświeżania wykresu.

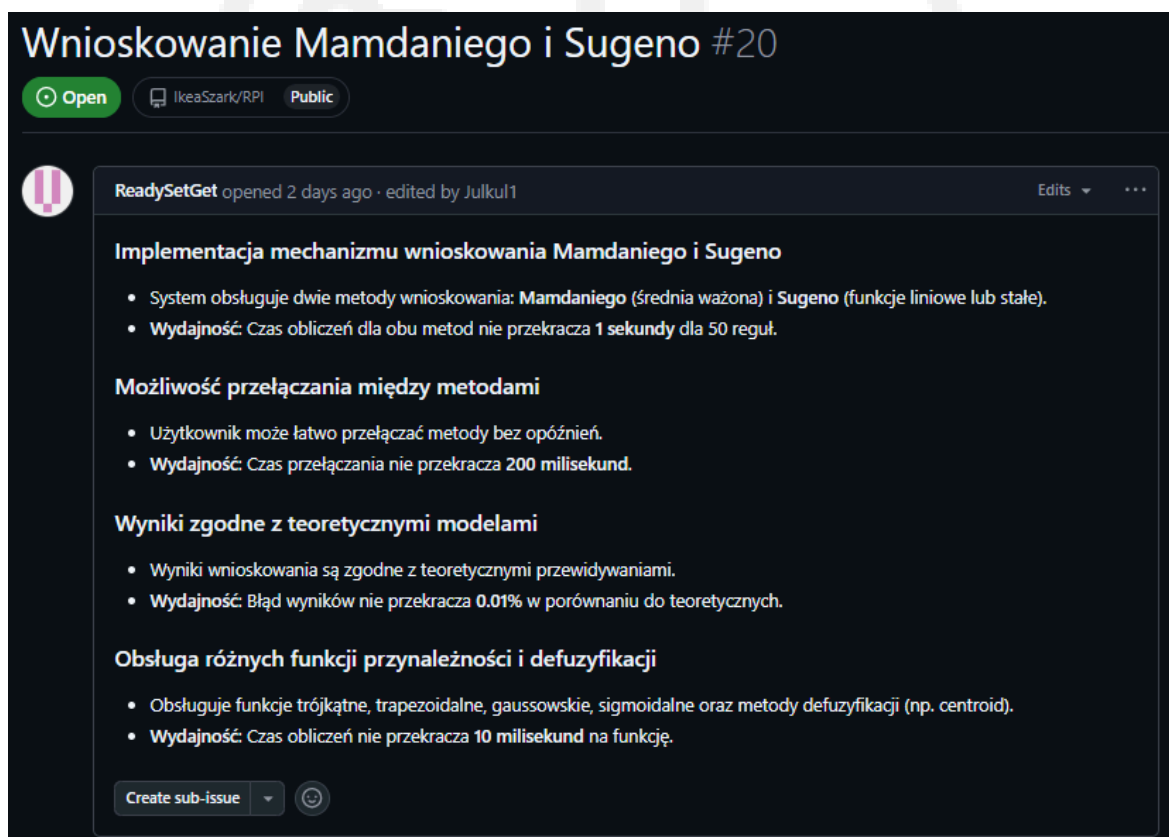
Create sub-issue ▾



### 5.1.3. Eksport systemu rozmytego do pliku



### 5.1.4. Wnioskowanie Mamdaniego i Sugeno







## **5.2. Ogólne kryteria akceptacji/gotowości do wdrożenia produktu**

- Wszystkie funkcjonalności są poprawnie zaimplementowane i przetestowane,
- Produkt przeszedł testy jednostkowe i integracyjne,
- Kod spełnia standardy jakości i jest udokumentowany,
- Brak krytycznych błędów uniemożliwiających działanie systemu,
- Użytkownik ma dostęp do instrukcji obsługi i dokumentacji technicznej,
- Produkt jest zgodny z wymaganiami biznesowymi i użytkowymi.





## 6. Definicja ukończenia (DoD)

- Wszystkie testy jednostkowe i integracyjne zakończyły się sukcesem,
- Funkcjonalność została zaprezentowana i zaakceptowana przez interesariuszy,
- System działa zgodnie z wymaganiami i nie zawiera błędów krytycznych,
- Dokumentacja została zaktualizowana zgodnie z nową funkcjonalnością,
- Wymagania нефunkcjonalne zostały przetestowane.

