



PROJEKT INŻYNIERSKI

Edytor graficzny systemów rozmytych dla języka Python

ID projektu - 46300

Opiekun projektu - dr inż. Jerzy Dembski

Dokument nr 6: Dobór i adaptacja metodyki

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 określenie metodyki, która będzie implementowana podczas wykonywania projektu. Przeprowadzona zostanie analiza konkretnych elementów projektu mających wpływ na wybór, a także ocena samej metodyki pod względem elementów niezbędnych, tych które nie będą implementowane oraz rozszerzeń. Celem końcowym jest wskazanie konkretnej, dostosowanej do potrzeb metodyki.

Wersja:	1.2
Data wydania:	04.06.2025
Redaktor:	Adam Zarzycki
Współautorzy:	Filip Wesołowski, Julian Kulikowski
Etap/zadanie:	6



Nazwa pliku:	Dobór_i_Adaptacja_Metodyki_v1.2.docx
Liczba stron:	13





Historia zmian

Wersja	Data	Opis zmiany
1.0	03.06.2025	Dodanie punktu 3
1.1	04.06.2025	Dodanie punktów 1. i 2.
1.2	05.06.2025	Dodanie punktów 4. i 5.





SPIS TREŚCI

1. O projekcie i produkcji	5
2. Ocena według modelu uproszczonego	6
2.1. Ocena wg 5 kryteriów	6
2.1.1. Rozmiar	6
2.1.2. Krytyczność	6
2.1.3. Dynamika	6
2.1.4. Osoby	6
2.1.5. Kultura	6
2.2. Dopasowanie na podstawie ocen	7
2.3. Wizualizacja oceny	7
3. Ocena według zaadoptowanego modelu pełnego	8
3.1. Zastosowanie	8
3.1.1. Główne Cele	8
3.1.2. Środowisko	8
3.2. Zarządzanie	8
3.2.1. Komunikacja	8
3.3. Techniczne	9
3.3.1. Wymagania	9
3.3.2. Wytwarzanie	9
3.4. Osoby	9
3.4.1. Klient	9
3.4.2. Kultura	10
4. Model dostarczania produktu końcowego projektu	11
4.1. Dlaczego przyrostowy	11
4.2. Sugerowana metodyka	11
5. Metodyka i jej adaptacja	12
5.1. Uzasadnienie wyboru metodyki zwinnej	12
5.2. Elementy projektu niepasujące do klasycznego Scrum/Kanban	12
5.3. Proponowane adaptacje metodyki zwinnej	12
5.4. Jak adaptacja kompensuje niedopasowania	13



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. Ocena według modelu uproszczonego

2.1. Ocena wg 5 kryteriów

2.1.1. Rozmiar

- Ostateczna ocena - 3 osoby,
- Uzasadnienie - w zespole są 3 osoby,
- Lepiej dopasowana metodyka - zwinna.

2.1.2. Krytyczność

- Ostateczna ocena - straty pomiędzy "kluczowymi funduszami" a "niekluczowymi funduszami" (z przewagą "kluczowych"),
- Uzasadnienie - porażka projektu skutkuje niską oceną z projektu inżynierskiego (wysoce wpływającego na ostateczną średnią ocen ze studiów), lub, w skrajnym przypadku, niezaliczeniem projektu i brakiem możliwości przystąpienia do egzaminu inżynierskiego. Nie jest to pełna "kluczowość", ze względu na możliwość występowania drobnych błędów w ostatecznym rozwiązaniu,
- Lepiej dopasowana metodyka - podobnie, z drobną przewagą klasycznej.

2.1.3. Dynamika

- Ostateczna ocena - 2% zmienionych wymagań / miesiąc,
- Uzasadnienie - projekt jest bardzo stabilny pod względem wymagań, dobrze zdefiniowany i zrozumiały, jak również niezależny od zmian czynników zewnętrznych. Zlecniodawca nie przewiduje znaczących zmian wymagań w najbliższej przyszłości,
- Lepiej dopasowana metodyka - klasyczna.

2.1.4. Osoby

- Ostateczna ocena - 10% poziomu 1B, 30% poziomu 2 i 3,
- Uzasadnienie - własna ocena poziomu członków zespołu projektowego, wynikająca z doświadczenia w realizacji podobnych środowiskowo projektów,
- Lepiej dopasowana metodyka - zwinna.

2.1.5. Kultura

- Ostateczna ocena - 66% skłonnych do chaosu vs. porządku,
- Uzasadnienie - na przeprowadzonym spotkaniu, 2 osoby określiły się jako



skłonne do chaosu, a jedna - do porządku,

- Lepiej dopasowana metodyka - zwinna.

2.2. Dopasowanie na podstawie ocen

- Kryteria skłaniające się ku metodyce zwinnej - 3,
- Kryteria skłaniające się ku metodyce klasycznej - 2,
- Ostateczny werdykt - metodyka zwinna.

2.3. Wizualizacja oceny

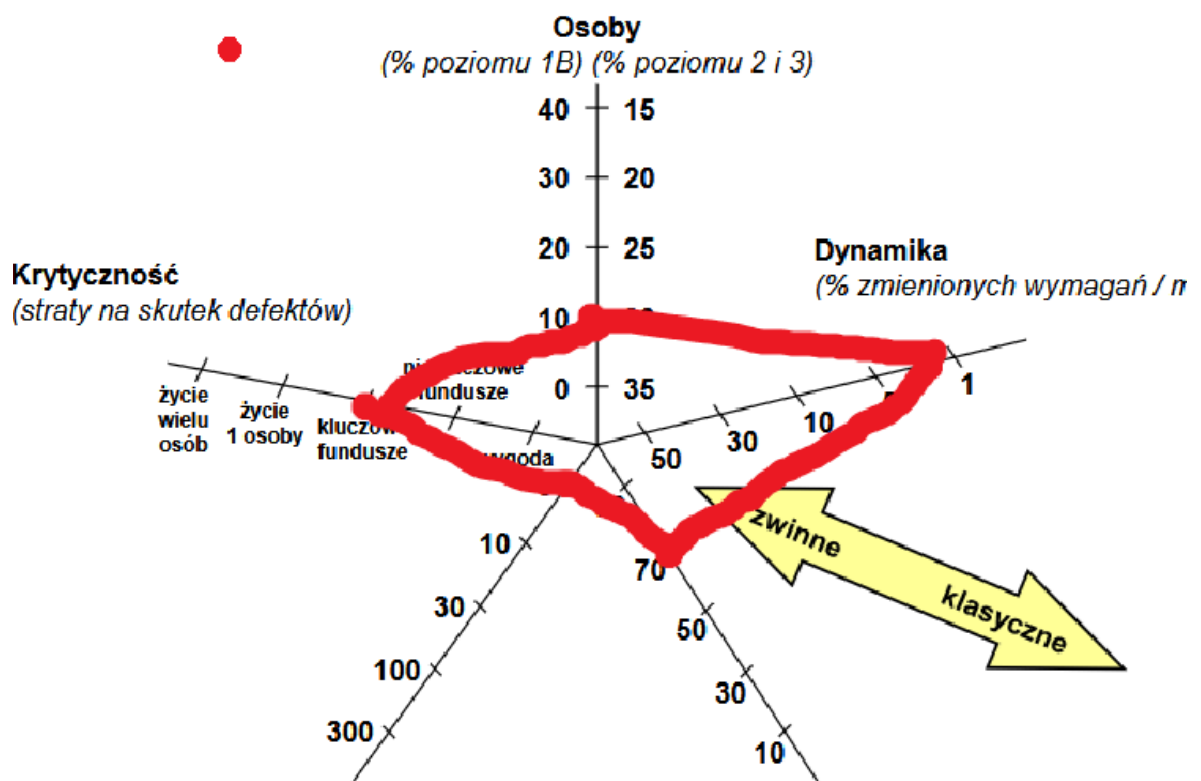


Fig. 1 - Wynikowy wykres 5 głównych kryteriów zwinne vs. klasyczne wg Boehma i Turnera



3. Ocena według zaadoptowanego modelu pełnego

3.1. Zastosowanie

3.1.1. Główne Cele

Głównym celem projektu jest możliwość szybkiego, inkrementalnego dostarczania produktu klientowi. Wysoko ceni się regularne spotkania z klientem, możliwość przedstawienia pracy, poddania jej ocenie oraz zadawanie pytań odnośnie zaimplementowanych funkcjonalności.

Elementy te pokrywają się z “własnym podwórkiem” metodyk zwinnych. W szczególności element szybkiego, częstego dostarczania inkrementalnych wersji produktu pokrywa się w dużo większym stopniu z metodami takie jak Kanban czy Scrum, niż Waterfall.

3.1.2. Środowisko

Otoczenie projektu jest niestabilne pod względem dostępności członków zespołu, często zdarzają się sytuacje wyjątkowe, które mają na nią wpływ. Przykłada się wysoką wagę zwraca się na posiadanie “produktu w rękach”, a mniejszą na jego późniejsze utrzymanie. Zakres osób, na które ma on wpływ jest wąski, gdyż logika rozmyta jest tematem niszowym. Jednocześnie zakłada się stosunkową stabilność wymagań biznesowych i brak drastycznych zmian.

Biorąc pod uwagę powyższe kryteria projekt zawiera w sobie część elementów należących do metodyk zwinnych oraz część należących do metodyk klasycznych. Mimo to uznano, że kryterium środowisko w większym stopniu przynależy do metodyk agile.

3.2. Zarządzanie

3.2.1. Komunikacja

Zespół nie prowadzi jawnej, scentralizowanej dokumentacji projektowej, wymiana wiedzy odbywa się za pomocą niejawnych kanałów komunikacyjnych takie jak wiadomości grupowe czy prywatne. Spotkania planistyczne i przeglądowe stanowią bardzo ważny aspekt wymiany informacji. Samą podstawą wiedzy w zespole jest doświadczenie poszczególnych członków. Zespół nie planuje tworzyć formalnych raportów, o ile nie wystąpi sytuacja wyjątkowa, w której będzie to wymagane.

Projekt w tym aspekcie zdecydowanie należy do podwórka metodyk zwinnych. Brak formalnej dokumentacji i niejawna wymiana informacji są w szczególności elementami wyróżniającymi komunikację w projektach agile. Należy także zwrócić uwagę na aspekt jakim są osobiste spotkania dwukierunkowe, które stanowią podstawę wymiany wiedzy w zespole.



3.3. Techniczne

3.3.1. Wymagania

Podstawą wymagań projektowych, według których wytwarzany jest projekt są nieformalne historyjki utworzone na wzór informacji pozyskanych od interesariuszy. Zespół projektowy zebrał nieformalne informacje na temat tego jakie funkcjonalności projekt powinien udostępniać, w jaki sposób wyglądać i ogólne oczekiwanie działanie.

Na tej podstawie w celu skonkretyzowania wymagań utworzył osoby użytkowników końcowych oraz historyjki określające przypadki użycia. Z założenia mają one jedynie nakreślić wymagania projektu, a konkretne szczegóły implementacyjne uzgadniane będą z klientem podczas regularnych spotkań.

Podejście zespołu w całości mieści się w zakresie metodyk zwinnych. Wszystkie aspekty od nieformalności wymagań, przez budowanie person i historyjek, po spotkania z klientem w celu uzgodnienia szczegółów są dla nich charakterystyczne. Sposób traktowania wymagań przez zespół w bardzo wąsko pokrywa się z metodykami klasycznymi.

3.3.2. Wytwarzanie

Przed rozpoczęciem wytwarzania zespół poświęcił czas na zdefiniowanie wydolnej architektury. Postanowiono, że zamiast później spędzać środki na refaktoryzacji należy od początku ustalić standardy programowania, strukturę programu oraz umożliwić ponowne użycie komponentów. Zakłada się, że wymagania nie będą zmienne do tego stopnia, by miało to negatywny wpływ. Jednocześnie zakłada się względną prostotę projektu oraz wąski zakres pod względem testów i implementowanych funkcjonalności.

W tym wypadku projekt znajduje się w podwórku metodyk klasycznych. Przykłada się dużo większą wagę na odgórne projektowanie architektury, niż refaktoryzowanie projektu na bieżąco. Jedyne elementy pokrywające się z metodykami agile jest prostota projektu.

3.4. Osoby

3.4.1. Klient

Podstawą relacji z klientem są regularne, częste spotkania, na których przeprowadzane są przeglądy, dostarczane są kolejne iteracje produktu oraz specyfikowane kolejne wymagania. Zespół nie posiada formalnej umowy, ani odgórnej specyfikacji. Zakłada się wysokie zaufanie, zaangażowanie oraz wiedzę klienta.

W tym przypadku projekt w większym stopniu należy do metodyk zwinnych, niż klasycznych. Ważnym aspektem projektu jest bliska współpraca z klientem oraz budowanie zaufania poprzez dostarczanie przyrostu w iteracjach.



3.4.2. Kultura

Kultura w zespole opiera się na wzajemnym zaufaniu i szacunku członków. Każda osoba sama wyznacza sobie zadania, którymi chce się zająć, panuje duża swoboda, a dwóch z trzech członków przejawia skłonności do chaosu.

Powyższe elementy są typowe dla metodyk zwinnych, w szczególności swoboda i samodzielne wybieranie zadań są aspektami typowymi dla Scrum. Skłonności do chaosu członków zespołu kłócą się z założeniami kultury metodyk klasycznych.





4. Model dostarczania produktu końcowego projektu

Model dostarczania produktu w naszym projekcie będzie oparty na podejściu przyrostowym (iteracyjnym). Oznacza to, że produkt końcowy graficzny edytor systemów rozmytych będzie rozwijany i dostarczany etapami, gdzie każda kolejna wersja wnosi nowe funkcjonalności i poprawki. Takie podejście umożliwia bieżące testowanie oraz ocenę produktu przez interesariuszy i użytkowników końcowych, co pozwala na szybką reakcję na uwagi i poprawę jakości.

4.1. Dlaczego przyrostowy

- Projekt zakłada złożone funkcjonalności, jak wizualizacja zbiorów rozmytych, definiowanie reguł, uczenie systemu metodą ANFIS, co wymaga elastyczności i stopniowego wdrażania,
- Przyrostowość ułatwia identyfikację błędów i nieścisłości na wczesnym etapie,
- Pozwala na częstsze prezentacje postępów klientowi, budując zaufanie i umożliwiając doprecyzowanie wymagań w toku realizacji.

4.2. Sugerowana metodyka

Najbardziej odpowiednia będzie metodyka zwinna, zwłaszcza Scrum lub Kanban. Obie dobrze odpowiadają potrzebom projektu:

- Scrum oferuje jasno określone iteracje (sprinty), spotkania planistyczne i retrospektywy, co wspiera regularną komunikację i kontrolę jakości.
- Kanban zapewnia elastyczny przepływ pracy, pozwalając szybko reagować na zmiany i priorytety bez sztywnego podziału na sprinty.

W obu przypadkach priorytetem jest szybkie i częste dostarczanie wartościowego produktu, co jest kluczowe w naszym projekcie.



5. Metodyka i jej adaptacja

Bazując na analizie w punktach 2 i 3 oraz modelu dostarczania z punktu 4, rekomendowana metodyka bazowa to metodyka zwinna (Agile), z silnym wskazaniem na Scrum lub Kanban.

5.1. Uzasadnienie wyboru metodyki zwinnej

- Wysoka dynamika i nieformalność wymagań – zespół korzysta z historyjek użytkownika i person, co jest typowe dla Agile.
- Bliska i regularna współpraca z klientem – częste spotkania i iteracyjne dostarczanie działających wersji produktu.
- Kultura zespołu – samodzielność, swoboda wyboru zadań i skłonność do mniej formalnych struktur pasują do metodyk zwinnych.
- Komunikacja nieformalna – nieformalna wymiana informacji i brak scentralizowanej dokumentacji wpisują się w praktyki Agile.

5.2. Elementy projektu niepasujące do klasycznego Scrum/Kanban

- Odgórne projektowanie architektury
 - Projekt wymaga wstępnego zaprojektowania struktury i standardów kodu, co jest bardziej charakterystyczne dla metodyk klasycznych.
 - Agile zazwyczaj promuje ewolucję architektury przez iteracyjne refaktoryzacje, ale tutaj konieczne jest uniknięcie kosztownych przeróbek.
- Minimalna formalizacja dokumentacji
 - Brak dokumentacji może utrudnić utrzymanie i rozwój systemu, zwłaszcza przy zmienności zespołu i skomplikowanych algorytmach rozmytych.

5.3. Proponowane adaptacje metodyki zwinnej

- Faza wstępnego projektowania architektury

Przed rozpoczęciem iteracji należy przeprowadzić solidne, ale stosunkowo krótkie planowanie architektury. W tym etapie definiuje się:

- standardy kodowania,



- moduły i ich interfejsy,
- wytyczne do implementacji algorytmów rozmytych i integracji z bibliotekami (np. PyTorch).

Dzięki temu zespół minimalizuje ryzyko kosztownych zmian podczas późniejszego wytwarzania.

- Lekkie, ale systematyczne tworzenie dokumentacji
Wprowadzenie minimalnych artefaktów dokumentacyjnych, np.:
 - dokumentacja API modułów,
 - opis najważniejszych algorytmów i struktur danych,
 - zbiór najważniejszych decyzji projektowych (ang. decision logs).
Dokumentacja będzie tworzona i aktualizowana podczas sprintów, co zapewni jej aktualność i użyteczność bez nadmiernego formalizmu.
- Zachowanie elastyczności w realizacji i komunikacji
 - Pozostawienie swobody w doborze zadań przez członków zespołu.
 - Utrzymanie nieformalnej, szybkiej komunikacji poprzez spotkania, czaty i przeglądy kodu.

5.4. Jak adaptacja kompensuje niedopasowania

- Wstępne planowanie architektury zapobiega chaosowi i kosztownym refaktoryzacjom, co jest słabością czystych metodyk Agile w projektach wymagających stabilnej podstawy technicznej.
- Minimalna dokumentacja pozwala zachować kontrolę nad złożonym systemem rozmytym i ułatwia onboarding nowych członków zespołu.
- Utrzymanie zwinnej komunikacji i iteracyjnego podejścia gwarantuje elastyczność, szybkie reagowanie na zmiany i bliską współpracę z klientem.