

WIMU - projekty

1. Informacje ogólne

1.1 Projekt z WIMU może dotyczyć:

- eksperymentalnych badań naukowych lub odtworzenia eksperymentu z istniejącego artykułu z dziedziny
- wytworzenia nowego narzędzia, frameworku, biblioteki - w szczególności wspomagającego eksperymenty lub ułatwiającego/umożliwiającego analizę, wizualizację, ewaluację danych audio
- rozbudowy istniejącego narzędzia, w szczególności z poprzedniej edycji WIMU

1.2 Projekt z WIMU musi cechować się:

- wysoką jakością kodu
- reprodukowalnością
- rzetelnością
- zastosowaniem nietrywialnych narzędzi

1.3 Projekt opracowywany jest w zespołach 3 os. Do projektu należy dostarczyć dokumentację, instrukcję użytkowania oraz adekwatne do tematu projektu testy. Oczekiwane są regularne kontrybucje do projektu od całego zespołu.

1.4 Pierwszym etapem projektu jest dostarczenie pisemnego **design proposal** („projekt projektu”). Design proposal zawiera:

- harmonogram projektu z podziałem na tygodnie i planowanymi postępami projektu per tydzień
- bibliografię
- planowany zakres eksperymentów
- planowaną funkcjonalność programu
- planowany stack technologiczny

Design proposal może zostać dostarczony w formacie .md albo .pdf. Harmonogram należy rozplanować i opisać wg dat. W harmonogramie projektu należy również uwzględnić potencjalny czas potrzebny na trenowanie modeli i przewidywane zapotrzebowanie na zasoby obliczeniowe.

Termin dostarczenia design proposal: 3. tydzień zajęć

1.5 W ciągu 3 tygodni od dostarczenia design proposal należy zademonstrować **funkcjonalny prototyp projektu** lub **postęp analizy literaturowej i konfigurację środowiska eksperymentalnego**.

1.6 Brak design proposal, prototypu, instrukcji użytkowania, dokumentacji lub adekwatnych do projektu testów skutkuje niezaliczeniem projektu.

1.7 Pracę nad projektem, ewentualne odchylenia od harmonogramu, pośrednie wyniki eksperymentów itp. należy na bieżąco dokumentować w prosty sposób (np. krótkie notatki w .md w repozytorium, issue tracker itp.) tak, aby być w stanie zreferować bieżące postępy na zasadzie „weekly standup”.

1.8 Oddając projekt należy dostarczyć 3-5min filmik demonstrujący projekt „w pigułce”.

2. Szczegóły technologiczne

Ogólne wymagania technologiczne:

- autoformatter (np. black), linter (np. flake8, ruff)
- środowisko wirtualne (np. venv, conda, pipenv, poetry)
- oskryptowane budowanie, testowanie, uruchamianie aplikacji (np. make, just, tradycyjne skrypty powłoki, skrypty w Pythonie - argparse, typer, click)
- dokumentacja
- instrukcja użytkownika
- styl zgodny z PEP8 (ze zwiększonym limitem długości linii)

Wskazówki: projekty AI / data science / eksperymentalne:

- niemutowalność oryginalnych danych
- rzetelna struktura projektu (np. cookiecutter, copier)
- śledzenie eksperymentów (np. Tensorboard / W&B)
- konfiguracja odseparowana od kodu wykonawczego
- eksport kodu z notatników do kodu bibliotecznego
- korzystanie z frameworków odpowiednich dla danego zagadnienia (np. Lightning, Huggingface)

Wskazówki: nowe frameworki, narzędzia, aplikacje:

- warstwowa struktura aplikacji
- mechanizmy zbierania logów i monitorowania aplikacji (np. logging, loguru, Sentry)
- konteneryzacja
- semantic versioning <https://semver.org/>
- budowa paczki (np. pdm, poetry, można też przez setup.py)
- mkdocs, sphinx
- tety automatyczne na co najmniej kilku wersjach interpretera (np. tox)
- dokładny stack technologiczny - do uzgodnienia w zależności od realizowanego zagadnienia