



# CodeInsight.ai

Czyli Jak Wszechstronnie Zarządzać Kodem

GOLEM SCRIBE

Robert Komar

Jakub Sobolewski  
Bartosz Kozakiewicz

Wojciech Maciejewski

Igor Obrępalski  
OpenAI GPT4

# Nasze obserwacje

”LLMy są wykorzystywane do bardzo trudnych zadań, ale wciąż nie znajdują zastosowania w wielu najprostszych codziennych czynnościach”

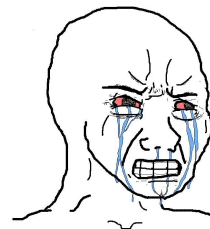
”Pisanie commit message jest świetnym przykładem tego problemu, to prosta czynność, ale bolesna dla programisty, ponieważ wymaga zmiany swojego kontekstu pracy. **Context switch jest kosztowny!**”

# Sugerowanie *commit message*

```
def division(a, b):  
+   if b == 0:  
+       return 0  
    return a / b
```

```
git commit -m "minor fix"
```

```
git commit -m "added handle division by zero in division function"
```



# Nasze obserwacje

”Popularne narzędzia jak Copilot Chat i GPT nie posiadają odpowiedniego kontekstu, kiedy sugerują nam poprawki w kodzie i niekoniecznie generują sensowne rzeczy. Nie działają kompleksowo!”

”Trend: dawanie odpowiedniego kontekstu LLMowi jest kluczowe!”

# Nasze rozwiązanie - 3 filary



## Repozytorium

- Graf wiedzy
- Eksploracja repozytorium
- Analiza zmian



## Developer

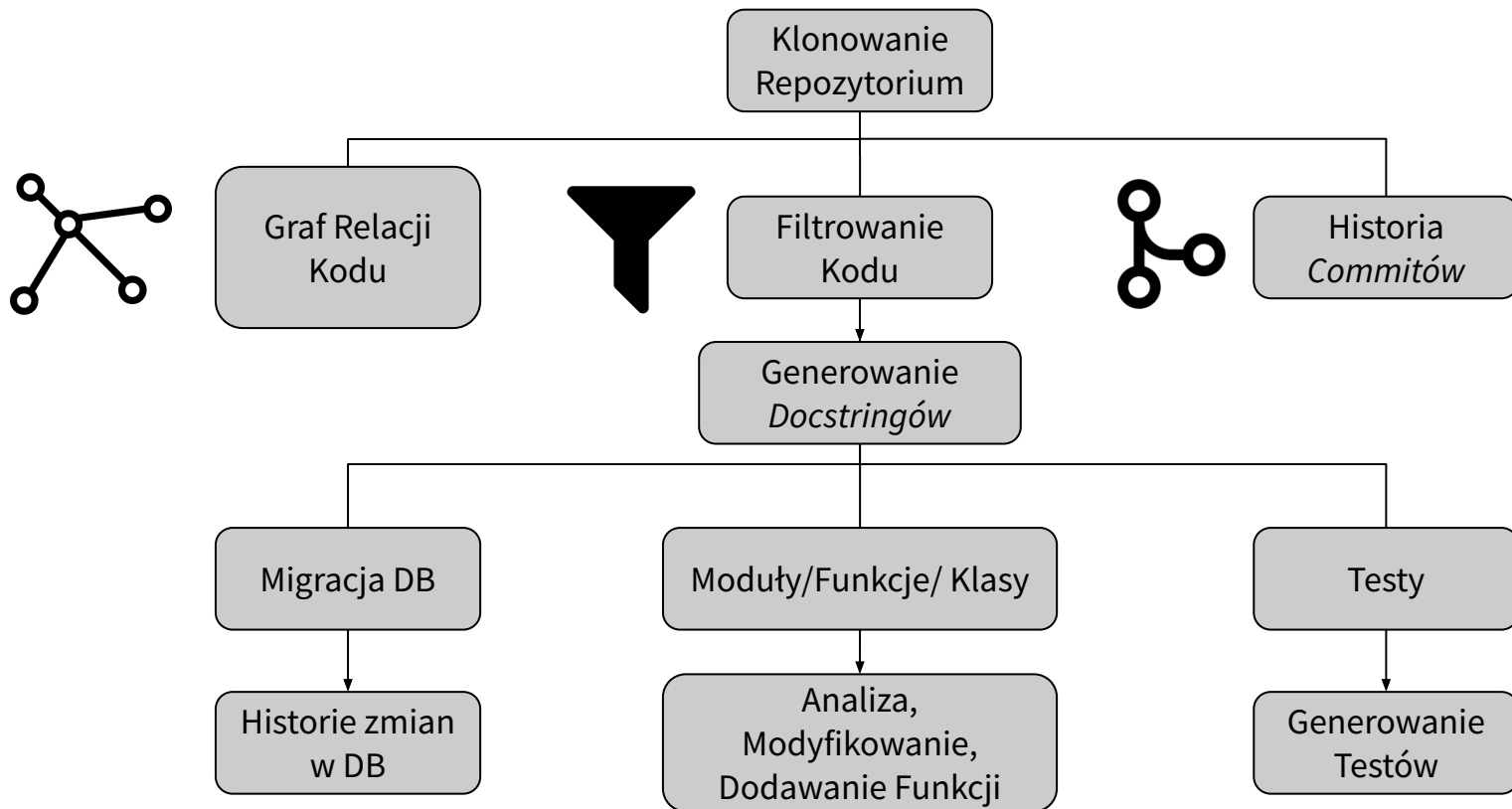
- Nowe funkcje
- Naprawa błędów
- Refaktoryzacja
- Zrozumienie działania



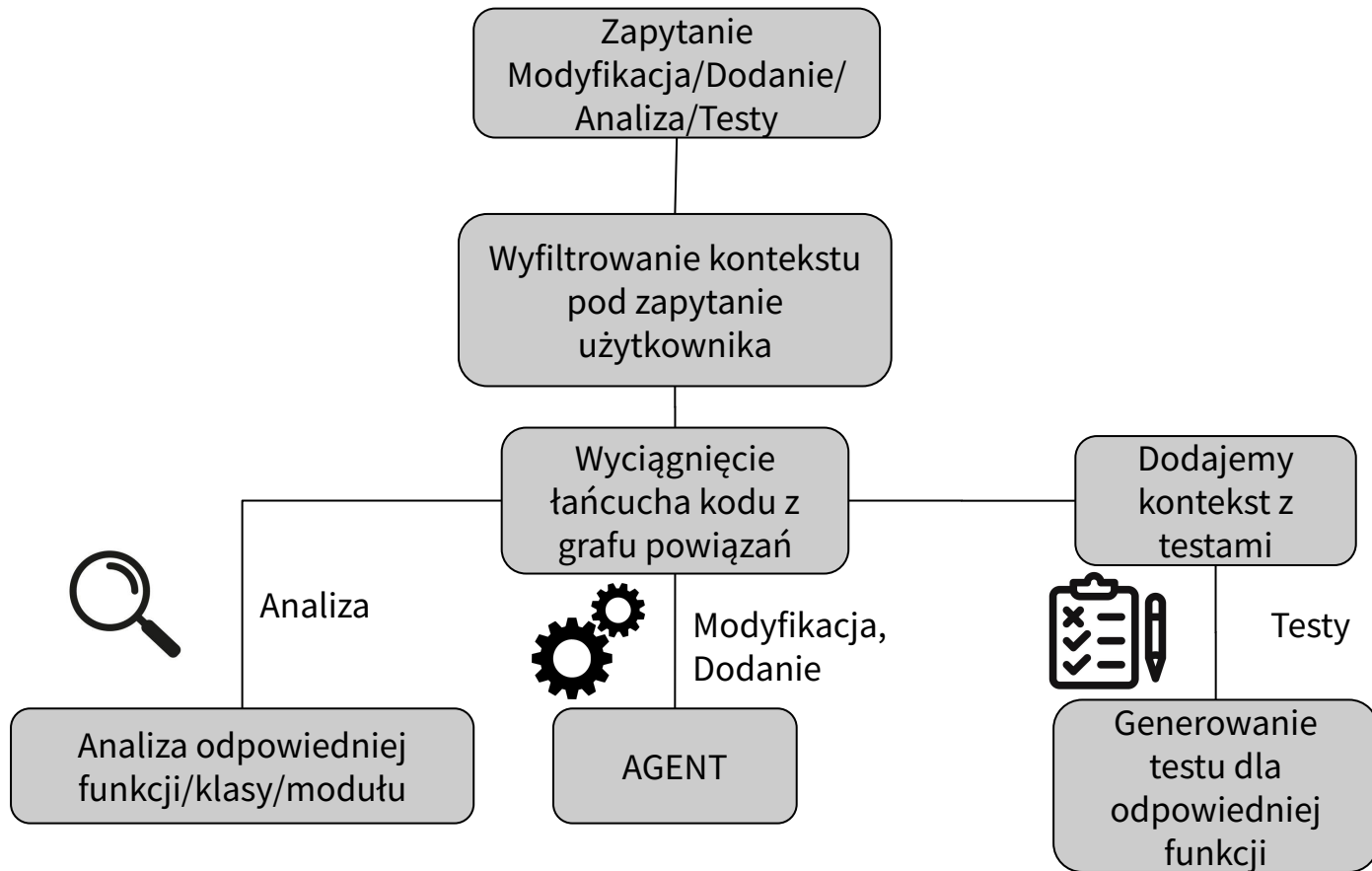
## Menedżer

- Generacja dokumentacji
- Automatyczne przypisywanie epiki
- Integracja Jira <-> Git

# Jak to działa?



# Logika funkcjonalności



# Graf wywoływania funkcji

Repository Path

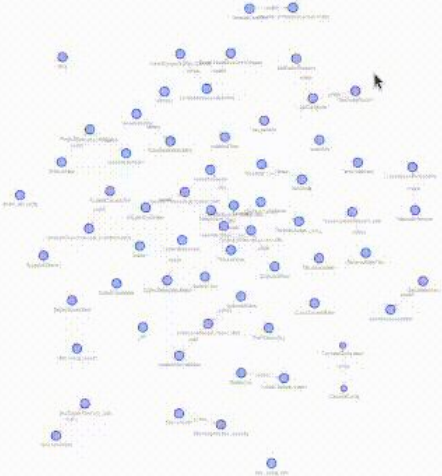
/home/laszcz/projects/bluesoft/hack/LLM

Select Module

corecode

Generate Graph

## Code Usage Graph



## Module Details

Class: Migration

[Ask LLM](#)

Docstring: None

```
class Migration(migrations.Migration):
    dependencies = [('corecode', '0001_initial')]
    operations = [migrations.RunPython(default_site
```

Function: default\_site\_config

[Ask LLM](#)

Docstring: Default site configurations

```
def default_site_config(apps, schema_editor):
    """Default site configurations"""
    User.objects.create_superuser('admin', 'admin@
    Config = apps.get_model('corecode', 'SiteConfig
    Config.objects.bulk_create([Config(key='school
    ), Config(key='school_slogan', value='A gre
    'school_address', value='Lagos, Nigeria')])
    Session = apps.get_model('corecode', 'Academic
    Session.objects.bulk_create([Session(name='201
    Term = apps.get_model('corecode', 'AcademicTerm
    Term.objects.bulk_create([Term(name='1st Term',
    name='2nd Term', current=False), Term(name=
    Subject = apps.get_model('corecode', 'Subject')
    Subject.objects.bulk_create([Subject(name='Math
    'English')])
```



# Modyfikacja kodu

Repository Path


/home/laszcz/projects/bluesoftack/LLM

Select Module

corecode

Generate Graph

Code Usage Graph



Module Details

Class: Migration

Ask LLM

Docstring: None

```
class Migration(migrations.Migration):
    dependencies = [('corecode', '0001_initial')]
    operations = [migrations.RunPython(default_site_config)]
```

Function: default\_site\_config

Ask LLM

Docstring: Default site configurations

```
def default_site_config(apps, schema_editor):
    """Default site configurations"""
    User.objects.create_superuser('admin', 'admin@schoolapp.com', 'admin123')
    Config = apps.get_model('corecode', 'SiteConfig')
    Config.objects.bulk_create([Config(key='school_name', value='My School'),
                               Config(key='school_slogan', value='A great school'),
                               Config(key='school_address', value='Lagos, Nigeria')])
    Session = apps.get_model('corecode', 'AcademicSession')
    Session.objects.bulk_create([Session(name='2019/2020', current=True)])
    Term = apps.get_model('corecode', 'AcademicTerm')
    Term.objects.bulk_create([Term(name='1st Term', current=True), Term(
        name='2nd Term', current=False), Term(name='3rd Term', current=False)])
    Subject = apps.get_model('corecode', 'Subject')
    Subject.objects.bulk_create([Subject(name='Mathematics'), Subject(name='English')])
```

# Pisanie testów

Repository Path

/home/laszcz/projects/bluesoftack/LLM

Select Module

corecode

Generate Graph

Deploy

```
def __str__(self):  
    return self.name
```

Class: StudentClass

Ask LLM

Docstring: None

```
class StudentClass(models.Model):  
    name = models.CharField(max_length=200, unique=True)  
  
    class Meta:  
        verbose_name = 'Class'  
        verbose_name_plural = 'Classes'  
        ordering = ['name']  
  
    def __str__(self):  
        return self.name
```

Function: str

Ask LLM

Docstring: None

```
def __str__(self):  
    return self.name
```

File Path: /home/laszcz/projects/bluesoftack/LLM-Dec-Hackathon/tests/test\_repo/apps/corecode/models.py

Class: SiteConfig

# Wyszukiwanie podatności

Repository Path

/home/laszer/projects/bluesofthack/LLM

Select Module

corecode

Generate Graph

Deploy

return self.name

Class: StudentClass

Ask LLM

Docstring: None

```
class StudentClass(models.Model):  
    name = models.CharField(max_length=200, unique=True)  
  
    class Meta:  
        verbose_name = 'Classs'  
        verbose_name_plural = 'Classes'  
        ordering = ['name']  
  
    def __str__(self):  
        return self.name
```

Function: str

Ask LLM

Docstring: None

```
def __str__(self):  
    return self.name
```

File Path:

/home/laszer/projects/bluesofthack/LLM-Dec-Hackathon/tests/test\_repo/apps/corecode/models.py

Class: SiteConfig



# Time Lapse najczęściej zmienianych funkcji

## Function Evolution Over Time

Select a function:

all\_results\_view (2 modifications)

Select a point in time

2020-05-25T18:04:51+01:00

2020-05-25T18:04:51+01:00

2021-07-17T12:52:15+01:00

```
def all_results_view(request):
    results = Result.objects.filter(
        session=request.current_session, term=request.current_term)
    bulk = []

    for result in results:
        test_total = 0
        exam_total = 0
        subjects = []
        for subject in result.subjects:
            if subject.student == result.student:
                subjects.append(subject)
                test_total += subject.test_score
                exam_total += subject.exam_score

        bulk[result.student.id] = {
            "student": result.student,
            "subjects": subjects,
            "test_total": test_total,
            "exam_total": exam_total,
            "total_total": test_total + exam_total
        }

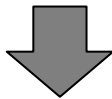
    context = {
        "results": bulk
    }

    return render(request, 'result/all_results.html', context)
```

# Naprawa błędów

```
def division(a, b):  
    return a / b
```

ZeroDivisionError: division by zero



```
def division(a, b):  
    if b == 0:  
        return 0  
    return a / b
```



# Naprawa błędów — ulepszony pipeline



stackoverflow



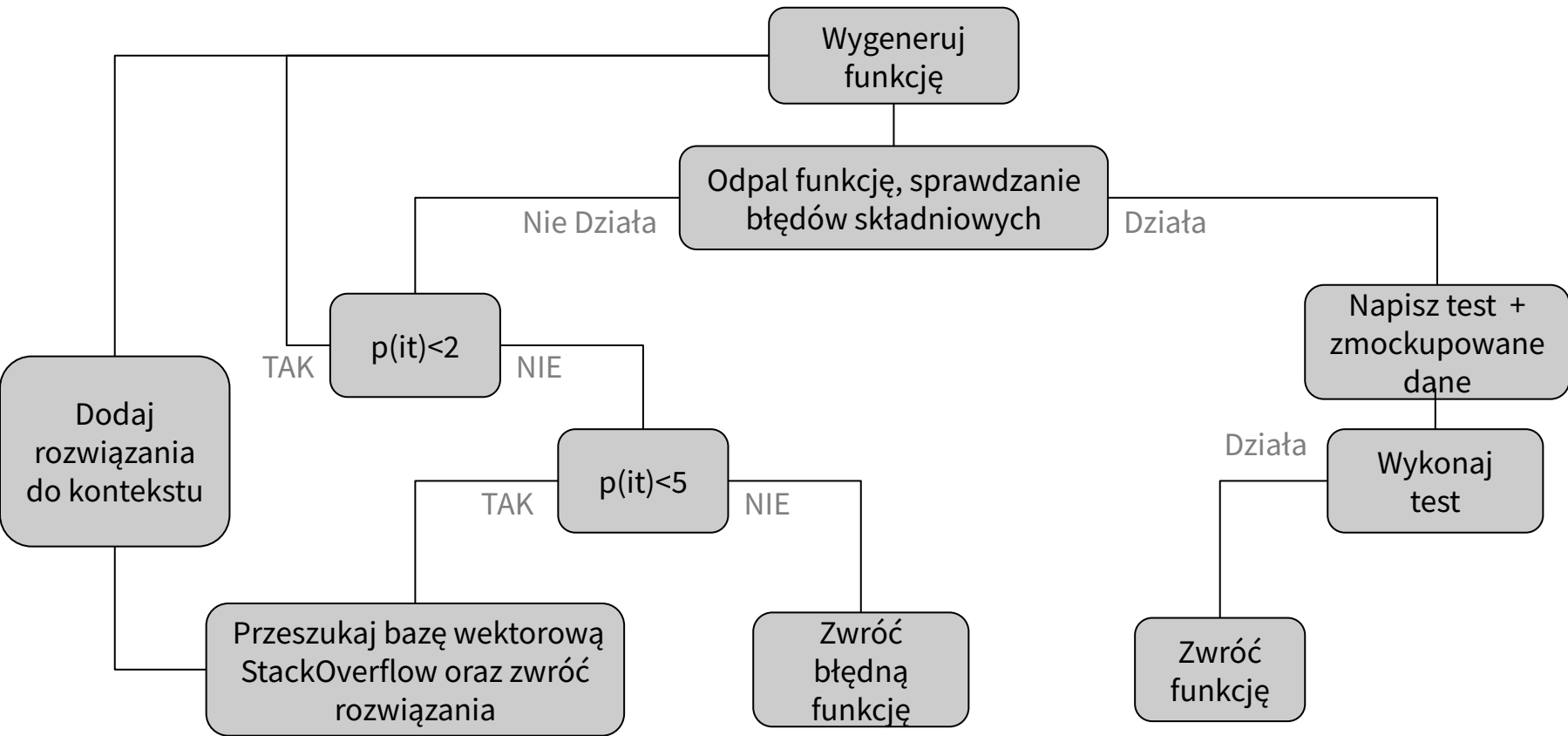
LangChain



Bug Fix!



# PRZYKŁADOWY AGENT GENEROWANIA KODU



# Narzędzie CLI - przykład

Modyfikacja istniejącego kodu bez potrzeby manualnej analizy istniejącej architektury — która bywa bardzo czasochłonna.

```
Wybierz jedną z możliwych kategorii: 1: Analiza, 2: Dodanie, 3: Testy: 4, Modyfikacja, 5: Szukanie podatności 6: Inne. Wpisz numer kategorii: 3
```

```
Wybrana kategoria to 3
```

```
Wchodzę w pipe modyfikacji, Wpisz swój prompt do chatu
```

```
Modify code in a way to add height row to student Model data with type of Number.
```

```
Function:
```

```
□
```

```
```python
class StudentModel:
    def __init__(self, name, age, grades, height):
        self.name = name
        self.age = age
        self.grades = grades
        self.height = height # Added height attribute with Number type

    # ... other methods and functionalities for the StudentModel class ...
```
```

```
End of Function
```

```
End of Function
```

```
*** & *** other methods and functionalities for the StudentModel class ***
```



# Łączenie menedżerów i programistów

- LLM + Code + Confluence =  



- Obszerna dokumentacja w uniwersalnym, narzuconym szablonie
- Automatyzacja czasochłonnego procesu tworzenia tasków

# Dlaczego my?

- Własny algorytm tworzący graf wywołań funkcji — graf wiedzy
- Wiele przydatnych modułów w jednym miejscu
- Łączy dwa światy: management i programistów
- Dojrzałe rozwiązanie od strony AI
  - ◆ LangChain
  - ◆ Zaawansowany prompting
  - ◆ Dodawanie kontekstu
- Deploy
- Biznesplan

***Sami wykorzystujemy nasze rozwiązanie***

# Strona Biznesowa

- Produkt SaaS
- Kompleksowe zarządzanie kodem
- Ułatwia rozwój oprogramowania
- Największą wartością jest nasz potężny backend
- Klient sam może dostarczyć swojego *on-premise LLM*

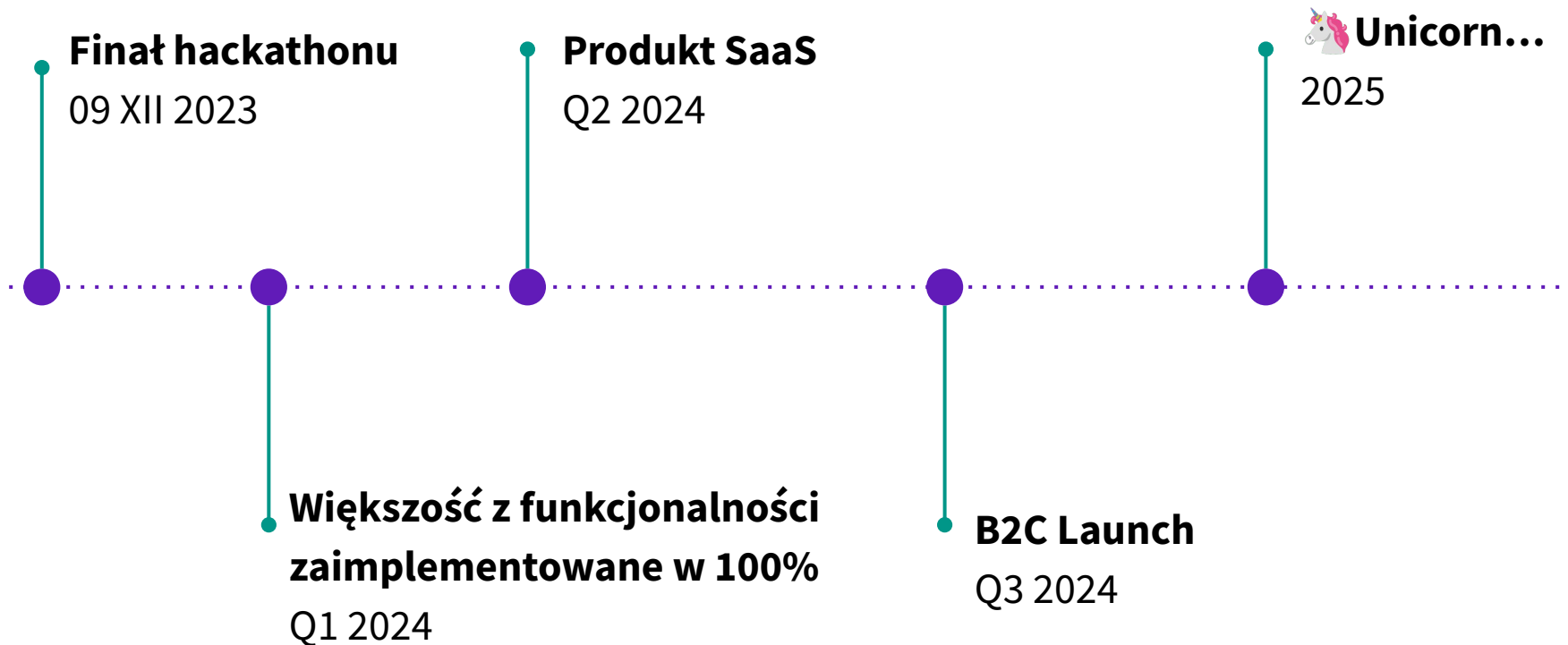
# Co Dalej?

- 🎉 Wiele z funkcjonalności zaimplementowane w 100% 🎉
- 👉 Proste dodawanie funkcjonalności 👉
- Opakowanie we wtyczkę do Chrome
- Sprzedaż produktu

Dalsze eksperymenty:

- group of experts [1]
- tipping the model [2]

# Co dalej?





**ADRES URL:**

<https://codeinsight.streamlit.app/>

