

## Program szkolenia:

# AI w pracy z kodem Legacy

### Informacje:

|                      |                                             |
|----------------------|---------------------------------------------|
| <b>Nazwa:</b>        | <b>AI w pracy z kodem Legacy</b>            |
| <b>Kod:</b>          | <b>ai-legacy</b>                            |
| <b>Kategoria:</b>    | Refaktoryzacja kodu legacy                  |
| <b>Odbiorcy:</b>     | liderzy techniczni, architekci, developerzy |
| <b>Czas trwania:</b> | 3 dni                                       |
| <b>Forma:</b>        | warsztaty                                   |

Zrozumienie istniejących systemów, odkrywanie ukrytej logiki i bezpieczna modernizacja oprogramowania z pomocą sztucznej inteligencji. Rozbudowane zależności, nieaktualna dokumentacja, niejawne reguły biznesowe i decyzje projektowe podejmowane wiele lat temu sprawiają, że nawet pozornie niewielka zmiana może prowadzić do trudnych do przewidzenia konsekwencji.

Narzędzia AI otwierają nowe możliwości pracy z takim oprogramowaniem. Potrafią wspierać analizę dużych baz kodu, odkrywanie zależności, rekonstruowanie logiki biznesowej, tworzenie testów i projektowanie alternatywnych modeli. Jednak samo zadawanie pytań asystentowi AI nie wystarczy. Aby skutecznie pracować z legacy, potrzebujemy uporządkowanego podejścia, które pozwala odróżniać fakty od hipotez, weryfikować wnioski generowane przez AI i kontrolować ryzyko wprowadzanych zmian.

Podczas szkolenia pokażemy, jak wykorzystać AI nie tylko do szybszego czytania i modyfikowania kodu, lecz przede wszystkim do systematycznego odkrywania rzeczywistej struktury, zachowania i znaczenia istniejącego systemu. Nauczmy się budować mapę wiedzy o aplikacji, identyfikować ukryte reguły i decyzje projektowe, rozpoznawać uniwersalne wzorce problemów oraz znajdować lepsze reprezentacje modeli biznesowych.

Szczególną uwagę poświęcimy bezpiecznej transformacji systemów legacy. Zamiast ryzykownego przepisywania całych aplikacji będziemy pracować nad podejściem opartym na równoległych modelach, porównywaniu ich zachowania i stopniowej migracji odpowiedzialności. Przyjrzymy się również temu, jak AI może wspierać analizę danych historycznych i projektowanie migracji między starymi a nowymi strukturami.

Celem szkolenia jest rozwinięcie umiejętności świadomej, opartej na dowodach pracy z systemami legacy, w której AI staje się narzędziem eksploracji, modelowania i weryfikacji, a nie jedynie generatorem kolejnych fragmentów kodu.

### Czego nauczysz się podczas szkolenia?

- Budować środowisko pracy z AI (AI Harness) wspierające eksplorację kodu, analizę zależności, dokumentowanie odkryć i kontrolowane wprowadzanie zmian.
- Tworzyć mapę wiedzy o istniejącym systemie, identyfikując kluczowe komponenty, modele, przepływy i reguły oraz rozróżniając fakty, hipotezy i obszary wymagające dalszej analizy.
- Rozpoznawać archetypy problemów ukryte w kodzie legacy, takie jak grafy, kolejki, modele rozliczeniowe czy problemy optymalizacyjne, i wykorzystywać je do poszukiwania właściwych rozwiązań.
- Odkrywać rzeczywiste zachowania systemu oraz tworzyć wspierane przez AI testy charakterystyczne, które zabezpieczają istniejącą funkcjonalność przed niezamierzonymi zmianami.

## BO•TT•EGA

IT minds

- Rekonstruować reguły biznesowe i decyzje projektowe, oddzielając istotną logikę domenową od przypadkowej złożoności technicznej.
- Identyfikować nieadekwatne abstrakcje i projektować alternatywne reprezentacje modeli, które lepiej oddają rzeczywistą naturę rozwiązywanego problemu.
- Projektować bezpieczną modernizację z wykorzystaniem Parallel Models, budując nowy model obok istniejącego, porównując ich wyniki i stopniowo przenosząc odpowiedzialność.
- Wspierać migrację danych narzędziami AI, odkrywając ukrytą semantykę danych historycznych, projektując transformacje i weryfikując ich poprawność.

### Dla kogo jest to szkolenie?

Szkolenie skierowane jest do programistów, senior developerów, tech leadów oraz architektów oprogramowania, którzy pracują z istniejącymi systemami i odpowiadają za ich utrzymanie, rozwój lub modernizację. Szczególnie skorzystają z niego osoby, które:

- Pracują z dużymi, wieloletnimi bazami kodu o ograniczonej dokumentacji.
- Muszą rozwijać funkcjonalności w systemach o wysokim stopniu złożoności i licznych zależnościach.
- Planują refaktoryzację, modernizację architektury lub stopniowe zastępowanie starszych komponentów.
- Chcą wykorzystywać narzędzia AI do głębszej analizy i zrozumienia oprogramowania, a nie tylko do generowania kodu.
- Odpowiadają za bezpieczeństwo zmian w systemach krytycznych dla działania organizacji.

### Efekt szkolenia

Po szkoleniu będziesz dysponować uporządkowanym podejściem do pracy z legacy wspieranym przez AI: od odkrywania i dokumentowania wiedzy o systemie, przez rozpoznawanie jego rzeczywistych modeli i zachowań, aż po projektowanie i weryfikowanie bezpiecznej transformacji.

Zamiast traktować istniejący kod jako zbiór klas i metod wymagających kolejnych poprawek, nauczysz się analizować go jako reprezentację ukrytych reguł, decyzji i problemów, które można zrozumieć, zweryfikować i często rozwiązać w lepszy sposób.

To szkolenie pokazuje, jak wykorzystać AI, aby modernizacja legacy przestała polegać na zgadywaniu intencji poprzednich autorów i stała się kontrolowanym procesem odkrywania, modelowania i ewolucyjnego doskonalenia systemu.

### Zalety szkolenia:

- AI jako narzędzie rozumienia, nie tylko generowania kodu
- Od analizy kodu do odkrywania rzeczywistego problemu
- Weryfikacja zamiast bezkrytycznego zaufania AI
- Ewolucyjna modernizacja zamiast Big Bang Rewrite
- Połączenie pracy z kodem, architekturą i danymi

## Szczegółowy program:

**1. AI Harness – fundamenty pracy z legacy. Efektywna praca z legacy wymaga czegoś więcej niż uruchomienia asystenta AI w IDE. Potrzebujemy środowiska, które wspiera systematyczne odkrywanie kodu, gromadzenie wiedzy, testowanie hipotez i kontrolowanie zmian.**

1.1. Czym jest AI Harness i jaką rolę pełni w pracy z istniejącym oprogramowaniem.

1.2. Organizacja narzędzi, kontekstu i instrukcji dla agentów AI.

1.3. Strategie eksploracji dużych i słabo udokumentowanych baz kodu.

1.4. Wykorzystanie AI do analizy zależności i weryfikacji hipotez.

1.5. Dokumentowanie odkryć i utrzymywanie kontekstu między kolejnymi etapami pracy.

1.6. Mechanizmy kontroli i weryfikacji zmian proponowanych przez AI.

**2. Budowanie mapy świadomości systemu z AI. Zanim zaczniemy modyfikować system, musimy zrozumieć, co naprawdę o nim wiemy, czego jedynie się domyślamy i które obszary pozostają nieodkryte.**

2.1. Odkrywanie najważniejszych komponentów, odpowiedzialności i zależności.

2.2. Identyfikacja modeli domenowych, przepływów danych i scenariuszy biznesowych.

2.3. Rekonstruowanie architektury na podstawie kodu i obserwacji.

2.4. Budowanie mapy wiedzy o systemie z pomocą AI.

2.5. Klasyfikowanie ustaleń jako faktów, hipotez i niewiadomych.

2.6. Identyfikowanie obszarów największego ryzyka i planowanie dalszej eksploracji.

**3. Odkrywanie archetypów oprogramowania w legacy z AI. Wiele pozornie unikalnych systemów biznesowych realizuje znane problemy informatyczne i matematyczne. Ich rozpoznanie pozwala oderwać się od historycznej implementacji i znaleźć prostsze, bardziej adekwatne rozwiązania.**

3.1. Czym są archetypy oprogramowania i dlaczego warto ich szukać.

3.2. Oddzielanie istoty problemu od szczegółów biznesowych i technologicznych.

3.3. Rozpoznawanie struktur grafowych, kolejek i przepływów.

3.4. Odkrywanie problemów optymalizacyjnych, rozliczeniowych i alokacyjnych.

3.5. Wykorzystanie AI do formułowania i porównywania hipotez dotyczących archetypu.

3.6. Dobór algorytmów, struktur danych i wzorców do rzeczywistego charakteru problemu.

3.7. Unikanie odtwarzania przypadkowej złożoności istniejącej implementacji.

**4. Odkrywanie obserwowalnych zachowań z AI. Dokumentacja i kod źródłowy nie zawsze odzwierciedlają rzeczywiste zachowanie systemu. W legacy to właśnie aktualne działanie aplikacji często stanowi najważniejszy punkt odniesienia podczas modernizacji.**

4.1. Różnica między deklarowanym a rzeczywistym zachowaniem systemu.

4.2. Odkrywanie scenariuszy biznesowych na podstawie kodu, danych i punktów wejścia.

4.3. Identyfikowanie efektów ubocznych, wyjątków i przypadków brzegowych.

4.4. Wykorzystanie AI do projektowania eksperymentów i scenariuszy testowych.

4.5. Tworzenie testów charakterystycznych zabezpieczających istniejące zachowania.

4.6. Budowanie podstaw do późniejszego porównywania starego i nowego rozwiązania.

**5. Odkrywanie reguł, decyzji i rodzajów logiki z AI. W systemach legacy logika biznesowa, aplikacyjna i techniczna często jest ze sobą silnie wymieszana. Rozdzielenie tych warstw wymaga zrozumienia nie tylko tego, co robi kod, ale także dlaczego został napisany w określony sposób.**

5.1. Identyfikowanie jawnych i ukrytych reguł biznesowych.

5.2. Rekonstruowanie decyzji projektowych na podstawie istniejącej implementacji.

5.3. Rozróżnianie logiki domenowej, aplikacyjnej i infrastrukturalnej.

5.4. Odkrywanie niejawnych założeń, ograniczeń i zależności.

5.5. Wykorzystanie AI do analizy rozproszonych warunków i ścieżek decyzyjnych.

5.6. Oddzielanie istotnych reguł od historycznych obejść i przypadkowej złożoności.

5.7. Dokumentowanie odkrytej logiki jako podstawy dalszej transformacji.

**6. Odkrywanie i zmiana reprezentacji modelu z AI. Problemy z utrzymaniem systemu nie zawsze wynikają ze złej jakości kodu. Często źródłem trudności jest niewłaściwa reprezentacja problemu, która wymusza skomplikowane zależności i operacje.**

6.1. Rozpoznawanie sygnałów świadczących o nieadekwatnym modelu.

6.2. Analiza ograniczeń wynikających z aktualnych abstrakcji.

6.3. Wykorzystanie AI do proponowania alternatywnych reprezentacji problemu.

6.4. Modelowanie z użyciem grafów, agregatów i modeli matematycznych.

6.5. Porównywanie reprezentacji pod kątem prostoty, spójności i możliwości rozwoju.

6.6. Weryfikowanie nowych modeli względem odkrytych reguł i zachowań.

6.7. Projektowanie zmiany modelu bez utraty istotnej semantyki biznesowej.

**7. Parallel Models – bezpieczna transformacja legacy z AI. Przepisanie całego systemu od zera często wiąże się z wysokim ryzykiem, długim czasem realizacji i utratą części niejawnej wiedzy biznesowej. Alternatywą jest stopniowa transformacja oparta na współistnieniu starego i nowego modelu.**

7.1. Koncepcja Parallel Models i jej zastosowanie w modernizacji legacy.

7.2. Projektowanie nowego modelu obok istniejącej implementacji.

7.3. Synchronizowanie danych i operacji pomiędzy modelami.

7.4. Równoległe wykonywanie operacji i porównywanie wyników.

7.5. Wykorzystanie AI do analizy rozbieżności i identyfikowania ich przyczyn.

7.6. Stopniowe przenoszenie odpowiedzialności na nowy model.

7.7. Strategie ograniczania ryzyka, wycofywania zmian i kontrolowanego przełączania.

7.8. Weryfikacja zgodności funkcjonalnej podczas kolejnych etapów transformacji.

**8. Migracja danych wspierana przez AI. Transformacja modelu często wymaga również zmiany reprezentacji danych. W systemach legacy dane historyczne zawierają jednak niejawne reguły, wyjątki i znaczenia, które nie są opisane w schemacie bazy danych.**

8.1. Analiza struktury i jakości danych historycznych.

8.2. Odkrywanie niejawnej semantyki pól, relacji i wartości.

8.3. Identyfikowanie anomalii, wyjątków i historycznych konwencji.

8.4. Wykorzystanie AI do formułowania hipotez dotyczących znaczenia danych.

8.5. Projektowanie mapowań i transformacji pomiędzy starym a nowym modelem.

8.6. Walidacja poprawności, kompletności i spójności migrowanych danych.

8.7. Porównywanie wyników operacji na starym i nowym modelu.

8.8. Planowanie bezpiecznej, etapowej migracji i mechanizmów kontroli.