

Program szkolenia:

Mastering Cursor IDE: Przyspieszenie tempa wytwarzania oprogramowania i wzrost jakości kodu w skali Enterprise

Informacje:

Nazwa:	Mastering Cursor IDE: Przyspieszenie tempa wytwarzania oprogramowania i wzrost jakości kodu w skali Enterprise
Kod:	cursor-master
Kategoria:	programowanie
Czas trwania:	3 dni
Forma:	40% wykłady / 60% warsztaty

Przekształć Cursor IDE z inteligentnego edytora w pełnoprawnego partnera inżynierskiego. Wykorzystanie AI jedynie do autouzupełniania kodu to marnowanie potencjału tej technologii w środowisku Enterprise.

W trakcie tego intensywnego, 3-dniowego warsztatu pokażemy, jak przejść od indywidualnej pracy z "inteligentnym podpowiadaczem" do budowy spójnego, opartego na AI ekosystemu wytwórczego. Skupimy się na tym, jak skalować wiedzę w organizacji, wymuszać standardy jakości poprzez współdzielone reguły oraz jak integrować Cursorsa z wewnętrznymi narzędziami firmy.

W trakcie szkolenia:

- opanujesz zaawansowaną inżynierię kontekstu, ucząc się, jak precyzyjnie sterować modelem, aby rozumiał specyfikę Twojej domeny i unikał halucynacji
- zmusisz AI do precyzyjnego podążania za Waszą architekturą aplikacji, eliminując generyczne wzorce na rzecz rozwiązań ściśle zgodnych z Waszym stylem architektonicznym i decyzjami projektowymi
- wdrożysz mechanizmy "Team-Aware AI", konfigurując pliki .cursorsrules i indeksowanie codebase'u tak, aby każdy członek zespołu otrzymywał sugestie spójne z Waszymi standardami kodowania, architekturą i Definition of Done
- zbudujesz i wdrożysz własne integracje oparte na MCP, łącząc AI z wewnętrznymi systemami firmy, a dzięki definiowaniu dedykowanych Docs i Tools wyposażysz asystenta w unikalne zdolności specyficzne dla Waszych procesów biznesowych
- przećwiczysz kompletny zoptymalizowany proces AI-Assisted SDLC, od analizy wymagań i projektowania rozwiązania, implementacji, zachowania jakości kodu i zgodności ze wymaganiami, aż po automatyzację Code Review, merge i dokumentację

Zalety szkolenia:

- **Reverse Engineering i Wizualizacja**
Błyskawiczne zrozumienie "Big Picture" aplikacji i generowanie diagramów z kodu, z zachowaniem chirurgicznej precyzji tam, gdzie logika jest kluczowa.
- **Reverse Architecture**
Ekstrakcja "DNA" architektury z istniejącego projektu, aby AI przy tworzeniu nowych funkcji idealnie odwzorowywało Wasze wzorce i konwencje.
- **Techniki Specyfikacji z AI**
Transformacja niejasnych wymagań biznesowych w precyzyjną specyfikację techniczną oraz wykrywanie luk logicznych, zanim napiszesz pierwszą linię kodu.



- **Jakość bez kompromisów**

Generowanie kodu, o jakim zawsze marzyłeś (Clean Code, pełne testy itp.), w czasie krótszym niż tradycyjne pisanie rozwiązań "na szybko".

Szczegółowy program:

1. Specyfikowanie i Analiza Wymagań

- 1.1. Przekształcanie surowych materiałów biznesowych (transkrypty, schematy, dokumenty) w precyzyjny kontekst techniczny
- 1.2. Automatyzacja dekompozycji wymagań na atomowe zadania programistyczne – Composer / Agent
- 1.3. Projektowanie kontraktów API, generowanie makiet itp. – Composer / Agent
- 1.4. Wykrywanie nieścisłości logicznych i przypadków brzegowych na wczesnym etapie analizy
- 1.5. Eksploracja obecnego systemu i analiza wpływu zmian (Impact Analysis) przed rozpoczęciem prac

2. Architektura i Proces Wytwórczy

- 2.1. Demonstracja skuteczności instrukcji architektonicznych – ćwiczenia programistyczne na dostarczonym zestawie reguł (.cursorrules)
- 2.2. Wymuszanie ścisłej zgodności kodu z przyjętymi wzorcami projektowymi i Architektura
- 2.3. Optymalizacja podziału ról (Human-AI collaboration) w procesie generowania kodu
- 2.4. Automatyczne generowanie instrukcji architektonicznych dla istniejących projektów – Composer / Agent
- 2.5. Propagowanie i współdzielenie standardów architektonicznych w zespole (Team-Aware AI)

3. Reverse Engineering i Praca z Legacy

- 3.1. Automatyczne dokumentowanie zastanego kodu i odtwarzanie procesów biznesowych – Composer / Agent
- 3.2. Generowanie wizualizacji architektury i diagramów zależności – Composer / MCP Tool
- 3.3. Strategie bezpiecznej modernizacji (Strangler Fig) czy wewnątrz monolitu (Bubble Context)
- 3.4. Odtwarzanie i weryfikacja decyzji projektowych (Architecture Recovery) na bazie kodu źródłowego

4. Ekosystem Cursor IDE

- 4.1. Konfiguracja reguł projektowych w .cursorrules sterujących zachowaniem agentów

4.2. Wykorzystanie trybu Composer (Agent) do realizacji złożonych zadań wieloplikowych i domenowych

4.3. Projektowanie zaawansowanych instrukcji i dokumentacji (@Docs) dla obsługi złożonych procedur inżynierskich

4.4. Wyszukiwanie, konfiguracja i aspekty bezpieczeństwa narzędzi MCP (Model Context Protocol)

4.5. Tworzenie własnych serwerów MCP integrujących AI z infrastrukturą firmy (np. bazy danych, logi)

4.6. Strategie propagowania i współdzielenia konfiguracji w zespołach rozproszonych (Shared Rules)