

Program szkolenia:

AI-Assisted Domain-driven Design i archetypów modeli biznesowych

Informacje:

Nazwa:	AI-Assisted Domain-driven Design i archetypów modeli biznesowych
Kod:	ddd-ddd
Kategoria:	Domain Driven Design i Event Storming
Odbiorcy:	architekci, developerzy, liderzy zespołów, liderzy techniczni, Product Owners
Czas trwania:	3 dni
Forma:	60% warsztaty/40% wykłady

Współczesne podejście do pracy z AI pokazuje jednoznacznie, że proste „vibecoding” nie działa w projektach, których złożoność wykracza poza prototypy i zabawki. Generatywna AI potrafi błyskawicznie pisać kod, lecz działa skutecznie tylko w środowisku, które jest dla niej zrozumiałe i ograniczone — w ramach precyzyjnie zdefiniowanych modułów, czytelnych granic domeny i poprawnie zaprojektowanej architektury.

Dlatego kluczowe stają się dziś umiejętności, które pozwalają wykorzystać AI jako realny boost pracy inżynierskiej:

- modularyzacja, umożliwiająca delegowanie zadań do agentów w małych, jednoznacznych kontekstach,
- weryfikacja i krytyczna ocena wyników AI, oparta o dobre praktyki inżynierskie,
- projektowanie autonomicznych modeli, API i warstw testów, które stanowią granice, w których AI może działać stabilnie,
- znajomość archetypów modeli biznesowych, pozwalająca składać systemy jak z klocków i prototypować nowe rozwiązania w krótkim czasie — przy silnym wsparciu generatywnej AI.

AI (jeszcze) nie myśli modularyzacyjnie, nie zaprojektuje architektury i nie nada spójności systemowi. Jej moc ujawnia się dopiero wtedy, gdy człowiek świadomie ograniczy kontekst i zdefiniuje ramy, w których agent może operować. Rolą inżyniera jest zaplanować strukturę, nazwać granice, ustalić kontrakty — a dopiero potem delegować implementację. W takim modelu AI staje się potężnym multiplikatorem produktywności.

Pracujemy z narzędziami takimi jak Claude Code, aby tworzyć precyzyjne instrukcje systemowe i kontrolować pracę AI poprzez specyfikacje, zamiast jednorazowych promptów. Uczestnicy ucą się budować własne agent-driven rozwiązania do implementacji modułów, testowania, analizy domeny czy code review.

Po stronie technicznej generowanie kodu na szkoleniu opiera się na znanych wzorcach taktycznych i lokalnych architekturach warstwowych, portach i adapterach oraz na separacji odpowiedzialności i technikach modularyzacji — tworząc środowisko idealne do kontrolowanego użycia AI. Po stronie biznesowej uczestnicy pracują ze sprawdzonymi archetypami domenowymi, takimi jak Product, Pricing, Accounting/Billing, Party czy archetypy grafowe. To one tworzą wspólny „język biznesowy”, w którym AI może efektywnie współpracować z inżynierem oraz umożliwiają prototypowanie systemów w godzinę, a nie w miesiące.

BO·TT·EGA

IT minds

Szkolenie koncentruje się na praktycznym projektowaniu systemów, które są zarówno AI-friendly, jak i agent-friendly. Uczy nowoczesnych praktyk inżynierskich, które pozwalają efektywnie, świadomie i bezpiecznie wykorzystywać AI w dużym zakresie procesu tworzenia oprogramowania — od eksplorowania domeny, przez architekturę modułarną i implementację autonomicznych modeli, aż po testowanie, weryfikację i szybkie budowanie funkcjonalnych modułów z użyciem archetypów biznesowych i agentowych workflowów.

To połączenie wzorców technicznych, archetypów biznesowych, podejścia spec-driven i zaawansowanych agentów Claude sprawia, że AI staje się produkcyjnym narzędziem o realnej mocy — a nie tylko ciekawostką czy gadżetem dla programistów.

Zalety szkolenia:

- **AI zacznie działać dla mnie przewidywalnie**, bo nauczę się projektować moduły, warstwy i porty/adapters w sposób, który eliminuje halucynacje i chaos w generowanym kodzie.
- **Będę prototypować systemy w kilka godzin**, korzystając z archetypów domenowych (Product, Pricing, Billing, Party, modele grafowe), które AI świetnie rozumie i potrafi natychmiast implementować.
- **Zbuduję własne workflowy agentowe**, dzięki którym Claude będzie implementował moduły, pisał testy, analizował domenę i robił code review na podstawie specyfikacji CLAUDE.md — zamiast pojedynczych promptów.
- **Zacznę kontrolować AI zamiast zgadywać, co zrobi**, bo nauczę się krytycznie oceniać output, wykrywać błędy architektoniczne i domenowe oraz skutecznie poprawiać generowane rozwiązania.

Szczegółowy program:

1. Czym jest AI-assistance?

1.1. Jak wygląda model współpracy człowiek AI

2. Vibe coding nie działa – realia pracy z AI

2.1. Dlaczego vibe coding nie działa w projektach wykraczających poza prototypy

2.2. Jak złożoność systemu wpływa na zdolność AI do utrzymania spójności

2.3. Skracanie kontekstu jako fundament pracy z AI

2.4. Ćwiczenie praktyczne

3. Modularyzacja jako warunek bezpiecznej współpracy z AI

3.1. Dlaczego modularyzacja staje się kluczowym narzędziem w erze AI

3.2. Rola architekta w procesie AI-assisted: decyzje, których AI nie podejmuje

3.3. Jak szukać granic?

3.3.1. Pojedyncze Źródło Prawdy (SSOT)

3.3.2. Zdarzenia pivotujące

3.4. Ćwiczenie praktyczne

4. Implementacja modułu z AI

4.1. Pierwsze kroki z Claude

4.2. Specyfikacja pliku CLAUDE.md

4.3. Architektura warstwowa oraz porty i adaptory w pracy z AI

4.4. Typowe pułapki przy implementacji z AI

4.5. Zmniejszanie kontekstu

4.6. Faza planowania jako klucz

4.7. Typowe wzorce generowane przez AI: agregaty, serwisy aplikacyjne, Value Objects

4.8. Debuggowanie z użyciem AI

4.9. Ćwiczenie praktyczne

5. Strategie testowania z AI

5.1. Gdzie i jak wykorzystywać agentów testujących

5.2. Wykrywanie nieprzetestowanych scenariuszy i miejsc ryzyka

5.3. Jak unikać eksplozji testów generowanych przez AI

5.4. Co jest „obserwowalnym zachowaniem” z perspektywy AI

5.5. Weryfikacja wygenerowanych testów

5.6. Ćwiczenie praktyczne

6. Eksploracja domeny i archetypy biznesowe

6.1. Archetypowe wzorce domenowe — klucz do prototypowania systemów w godziny (AI działa najlepiej, gdy pracuje na znanych strukturach domenowych)

6.2. Konkretnie omawiane archetypy:

6.2.1. Product (katalog produktów, konfiguracje, warianty)

6.2.2. Pricing (cenniki, reguły wyceny, kalkulacje)

6.2.3. Accounting / Billing (płatności, księgowanie, rozliczenia)

6.2.4. Archetypy grafowe (relacje, sieci, zależności, polecenia)

6.2.5. Party (ludzie, organizacje, role, uprawnienia)

6.3. Pytania poszerzające domenę - myśl jak architekt produktowy

6.4. Ćwiczenie praktyczne: praca na gotowych modelach w kodzie — eksploracja, modyfikacja i rozszerzanie domeny z udziałem AI

7. Zaawansowany setup agentów oraz Spec-Driven

7.1. Agenci do analizy archetypów i eksploracji domeny

7.2. Agenci do testów i wykrywania brakujących scenariuszy

7.3. Agenci do code review i kontroli architektury

7.4. Agent „sceptyk”

8. Etyka i bezpieczeństwo danych

8.1. Odpowiedzialne korzystanie z AI

8.2. Bezpieczeństwo danych i kontekstu

9. Podstawy MCP