BO·TT·EGA
IT minds

Program szkolenia:

Domain Driven Design - projektowanie modeli złożonych domen (część 1)

Informacje:

Nazwa:	**Domain Driven Design - projektowanie modeli złożonych domen (część 1)**
Kod:	**ddd-workshop-DDD**
Kategoria:	Warsztaty eksperckie DDD
Odbiorcy:	developerzy, analitycy, architekci
Czas trwania:	3 dni
Forma:	60% wykłady / 40% warsztaty

Autorski program oparty na 11 latach doświadczenia w stosowaniu i nauczaniu DDD. Warsztat symuluje krytyczne etapy w pracy nad projektem, w którym mamy do czynienia z nietrywialnymi wymaganiami, kilkoma zespołami i wieloma interesariuszami biznesowymi. Zakładamy też, że odpowiedni model da nam przewagę konkurencyjną.

Zakres

- techniki z poziomu Wzorców Taktycznych: Building Blocks DDD wraz z najlepszymi praktykami oraz elementami rozszerzonymi.
- techniki z poziomu Wzorców Strategicznych: Domain Distillation, Bounded Context Mapping,
- poznasz praktyczne podejścia i sposoby prowadzenia sesji modelowania z wykorzystaniem Event Stormingu.

Forma

Pierwszego dnia zaczynamy od Event Stormingu procesowego aby odkryć pod-domeny.
Następnie tworzymy mapę kontekstów, z których dzięki destylacji wyłaniają się generyczne archetypy modeli biznesowych. Na tej podstawie podejmujemy strategiczne decyzje na poziomie współpracy zespołów i izolacji modeli.
Kończymy z projektem krytycznych komponentów ilustrujących problemy techniczne: integracja, optimistic locking, skalowanie.

Drugiego dnia pochylamy się nad wybranymi kontekstami aby stworzyć model taktyczny z wykorzystaniem building blocks. Skupiamy się na granicy agregatów i politykach.

Trzeciego dnia zmieniamy wymagania aby sprawdzić jak zareaguje nasz model. Dodajemy również modelowanie funkcyjne dla dynamicznie zmieniających się reguł.

Formuła warsztatu to kolejne omawianie technik i ich demonstracja przez trenera, po czym uczestnicy w podgrupach samodzielnie przechodzą tą samą ścieżkę na innych wymaganiach. W zadaniach są ukryte pułapki odzwierciedlające realne problemy projektowe, które omawiamy wspólnie po każdym etapie.

Korzyści płynące z wykorzystania DDD

- Techniki i wzorce strategiczne, których celem jest rozwiązanie typowych problemów organizacyjnych

BO·TT·EGA
IT minds

    - efektywne techniki prowadzenia sesji modelowania z uczestnictwem Ekspertów Domenowych
    - skupienie wysiłku na Core Domain: inwestycja najlepszych ludzi i technik DDD jedynie w krytyczne moduły
    - techniki separacji osobnych domen (Bounded Context) wyznaczanych przez granice wiedzy Ekspertów Domenowych, redukcja ryzyka związanego z tworzeniem monolitów
    - techniki integracji modułów otwartych na zmiany i skalowanie wydajnościowe
    - strategie współpracy zespołów (w tym w modelu outsourcing) pracujących nad osobnymi modułami
- Techniki i wzorce taktyczne, których celem jest rozwiązanie typowych problemów technicznych i jakościowych
    - język wzorców dla tworzenia modelu - Building Blocks
    - architektury odpowiednie do tworzenia rozszerzalnych systemów
    - podejścia wspierające i ułatwiające testowanie automatyczne
- Płynna integracja ze Scrum dzięki podejściu Modeling Whirlpool

Projekt referencyjny

Sprawdź naszą implementację przykładowego projektu DDD+CqRS: Sample Projects.

Implementacja DDD

Techniki implementacji DDD (architektura aplikacyjna i systemowa, wykrzyknienie IoC i ORM) są omawiane na szkoleniu DDD-implementacja, które powinno nastąpić w drugiej kolejności, po szkoleniu z zakresu modelowania.

Zalety szkolenia:

- Uświadomisz sobie kompetencje miękkie jakie powinien posiadać Modelarz
- Dowiesz się jak dzielić system na moduły i odkrywać Bounded Contexty
- Uświadomisz sobie decyzje architektoniczne, jakie musisz podjąć na każdym poziomie z C4 oraz poznasz sposoby zdobywania odpowiedzi na te pytania
- Poznasz różne podejścia do modelowania: obiektowe, funkcyjne, oparte o archetypy

BO·TT·EGA
IT minds

Szczegółowy program:

1. Zadanie wstępne - praca nad systemem, którego granice modułów są niepoprawne

- 1.1. Próba integracji modułów przy pomocy zdarzeń i komend
- 1.2. Analiza problemów
- 1.3. Analiza genezy
- 1.4. Wstęp do wizualizacji architektury systemów
  - 1.4.1. Podejście C4
  - 1.4.2. Drivery architektoniczne na każdym poziomie
  - 1.4.3. Pytania na jakie musimy sobie odpowiedzieć na każdym poziomie

2. Proces i techniki modelowania

- 2.1. Event Storming
  - 2.1.1. Notacja
  - 2.1.2. Fazy: rozbieżna eksploracja, analiza, synteza i krystalizacja modelu
  - 2.1.3. Facylitacja sesji - Tips and Tricks
- 2.2. Trzy poziomy Event Stormingu
  - 2.2.1. Big Picture - otoczenie systemu
    - 2.2.1.1. Notacje wspomagające
  - 2.2.2. Process Level
    - 2.2.2.1. Techniki odkrywania pod-domen biznesowych
    - 2.2.2.2. Techniki destylacji generycznych pod-domen
    - 2.2.2.3. Heurystyki proponowania Bounded Contextów
    - 2.2.2.4. Mapowanie Bounded Contextów
  - 2.2.3. Tactical design
    - 2.2.3.1. Określanie granic agregatów na podstawie spójności reguł biznesowych

BOTTEGA IT minds

- 2.3. Podejście "od procesu" vs podejście "od domeny" w kontekście modelu Cynefin
- 2.4. Wstęp do archetypów modeli biznesowych na przykładzie generycznych pod-domen

3. Wzorce Taktyczne - Building Blocks

- 3.1. Koncepcja języka Wzorców DDD
  - 3.1.1. Potrzeba większej ilości building blocks niż serwis i encja (procedura i struktura danych)
- 3.2. Encje
  - 3.2.1. Obiekty do który możemy odnieść się w formie "ten obiekt"
- 3.3. Agregaty
  - 3.3.1. Hermetyzacja i otwarcie na rozbudowę
  - 3.3.2. Strategie określania granicy agregatu
  - 3.3.3. Modelowanie niezmienników
  - 3.3.4. Techniki lingwistyczne
    - 3.3.4.1. Pełne zdania: podmiot.orzeczenie(dopełnienie, przydawka)
    - 3.3.4.2. Odwrócenie kolejności: Słowo-Znaczenie(Kontekst)-Reguły
- 3.4. Value objects
  - 3.4.1. Obiekty do który możemy odnieść się w formie "taki obiekt"
  - 3.4.2. Zwiększenie siły wyrazu
  - 3.4.3. Styl funkcyjny
- 3.5. Serwisy Domenowe
  - 3.5.1. Model procedur biznesowych
- 3.6. Repozytoria
  - 3.6.1. Abstrakcja magazynu danych
  - 3.6.2. Orientacja na model domenowy zamiast na model danych
- 3.7. Fabryki

BOTTEGA IT minds

BOTTEGA IT minds

- 5.1. Podział logiki na aplikacyjną i domenową
- 5.2. Podejście warstwowe - rozmieszczenie building blocks na warstwach
  - 5.2.1. Warstwa interfejsów (prezentacji)
  - 5.2.2. Warstwa logiki - rozwarstwienie na dwie warstwy logiki
    - 5.2.2.1. Logika aplikacyjna (API)
    - 5.2.2.2. Model tego CO system powinien robić (User Case lub User Story)
    - 5.2.2.3. Zapis Domain Story w formie czytelnej prozy
    - 5.2.2.4. Logika domenowa (Building Blocks DDD)
    - 5.2.2.5. Model tego JAK i DLACZEGO system powinien się tak zachowywać
  - 5.2.3. Warstwa infrastruktury
    - 5.2.3.1. Dostęp do danych
    - 5.2.3.2. Infrastruktura techniczna
- 5.3. Przełożenie Use Case/User Story na warstwę aplikacji
- 5.4. Przełożenie modelu biznesowego na building blocks warstwy domenowej

6. Architektura systemu

- 6.1. Dobór strategii integracji na podstawie wyników Event Stormingu procesowego
- 6.2. Zarys architektury integracji modułów
  - 6.2.1. Event Driven Architecture
  - 6.2.2. Microservices