

Program szkolenia:

Mikroserwisy .NET Core - Część I: Architektura aplikacji

Informacje:

Nazwa:	Mikroserwisy .NET Core - Część I: Architektura aplikacji
Kod:	NET-arch-ms1
Kategoria:	Architektura .NET
Odbiorcy:	DevOps, developerzy, architekci
Czas trwania:	3 dni
Forma:	30% wykłady / 70% ćwiczenia

Warsztaty wprowadzające w świat nowoczesnej architektury mikroserwisów z wykorzystaniem metodyki Event Storming oraz .NET Core.

Z nami odkryjesz domenę systemu wykorzystując Event Storming, a następnie utworzysz pierwsze mikroserwisy wraz z całą infrastrukturą.

Warsztaty mogą być połączone w kompleksowy program trwający łącznie 6 dni :

- Część pierwsza
 - Dzień 1: Event Storming - Jak odkrywać nieznaną? Określanie granic usług i dobieranie strategii integracji
 - Dzień 2 i 3: Architektura aplikacji - Wprowadzenie w architekturę mikroserwisów wraz z ich implementacją oraz integracją.
- Część druga
 - Dzień 4, 5 i 6: Architektura wdrożenia - Zaawansowane zagadnienia związane z testowaniem, budową, utrzymaniem i wdrożeniem mikroserwisów.

Zalety szkolenia:

- Analiza driverów architektonicznych z wykorzystaniem Event Storming i elementów Domain-driven Design
- Sprawdzone wzorce projektowe wraz z kontekstem poprawnego ich wykorzystania
- Architektura systemów rozproszonych

Szczegółowy program:

1. Event Storming - Jak odkrywać nieznane? Określanie granic usług i dobieranie strategii integracji.

1.1. Wstęp do Event Stormingu

1.1.1. Problem jaki chcemy rozwiązać

1.1.2. Stosowalność

1.1.3. Metodyka

1.1.3.1. Mechnika

1.1.3.2. Role i odpowiedzialność

1.2. Sesja procesowa - wydzielenie granic usług

1.2.1. Łagodne wejście w nomenklaturę poprzez stopniowe wprowadzanie notacji

1.2.2. Odkrycie procesów biznesowych

1.2.3. Wstępna destylacja Bounded Context

1.2.4. Określanie klasy problemu z jakim mamy do czynienia w każdym BC

1.2.4.1. Szacowanie ryzyk

1.2.4.2. Drivery architektoniczne

1.2.5. Odkrywanie ukrytych Bounded Context

1.2.5.1. Conway's Law vs SOA:Single Source of Truth

1.2.5.2. Destylacja dziedziny kontekstów tak aby były reużywalne

1.2.6. Opracowanie strategii integracji BC

1.2.6.1. Published Language

1.2.6.2. Open Host

1.2.6.3. Shared Kernel

1.2.6.4. Anticorruption Layer

1.2.6.5. Customer-Supplier

1.2.6.6. Conformist

1.2.7. Propozycja modułów technicznych na podstawie granic BC (jeden BC to potencjalnie kilka modułów)

1.2.7.1. Przygotowanie modułów do życia w izolacji jako microservices

1.2.7.2. Destylacja API

1.3. Sesja taktyczna - wewnętrzny model usług

1.3.1. Kryteria wyboru kontekstów, w których będziemy stosować DDD

1.3.2. Sesja ES z pogłębionym poszukiwaniem reguł domenowych

1.3.3. Określanie granic agregatów - reguły i heurystyki

1.3.3.1. Typowe problemy

1.3.3.1.1. Źle obrany korzeń

1.3.3.1.2. Zbyt duży agregat - brak kohezji

1.3.3.1.3. Mylenie obiektów biznesowych z widokami (projekcjami)

1.3.3.2. Najlepsze praktyki

1.4. Tematyka miękka

1.4.1. Nawyki kognitywne uczestników sesji - dobór stylu prowadzenia sesji do typów uczestników

1.4.2. Zadawanie pytań z intencją lepszego zrozumienia zamiast nękania

1.4.3. Dbanie i komfort emocjonalny nietechnicznych uczestników sesji

2. Fundamenty mikroserwisów

2.1. Stosowalność mikroserwisów

2.2. Poziomy architektury

2.3. Antywzorzec rozproszonego monolitu

3. DDD w praktyce

3.1. Agregaty i ich granice spójności

3.2. Repozytoria i serwisy aplikacji

3.3. Zdarzenia domenowe

4. Architektura aplikacji

4.1. Czysta architektura

4.2. Stworzenie RESTful API

4.3. CQS i CQRS

4.4. Warstwa infrastruktury

5. Architektura zorientowana na zdarzenia

5.1. Wzorce integracyjne

5.2. Broker wiadomości

5.3. Spójność danych

6. Integracja przez zdarzenia

6.1. Subskrypcja zdarzeń

6.2. Publikowanie zdarzeń

7. Transakcyjna obsługa wiadomości

7.1. Wzorzec Inbox

7.2. Wzorzec Outbox