BO·TT·EGA
IT minds

Program szkolenia:

Testy Kontraktowe na platformie Java i Pact

Informacje:

Nazwa:	**Testy Kontraktowe na platformie Java i Pact**
Kod:	**Arch-ct**
Kategoria:	Architektura systemów i aplikacji
Czas trwania:	3 dni
Forma:	50% wykłady / 50% warsztaty

Celem szkolenia jest nauka wykorzystania testów kontraktowych do zapełnienia luki w całości strategii testowania mikroserwisów aby zredukować koszt integracji i testów manualnych oraz automatycznych na środowiskach wdrożeniowych. Uczestnikami szkolenia są programiści mierzący się ze złożonością testowania integracji i bezpieczeństwa wdrożeń w architekturze mikroserwisowej, pragnący poszerzyć swoją wiedzę o sprawdzone strategie i wzorce. Warsztat systematyzuje rozbudowane zagadnienie, daje praktyczne wsparcie przy wdrażaniu i utrzymaniu testów kontraktowych.

Mnogość ruchomych elementów i skomplikowanie architektury mikroserwisów powoduje że znane dotąd strategie pracy z kodem i testami wymagają dedykowanego podejścia. Takiego które zagwarantuje wysoką jakość produktu i sprawną współpracę zespołów nie powiększając jednocześnie kosztu wytworzenia i utrzymania zarówno dla testów wewnątrz pojedynczego mikroserwisu jak i kontraktów oraz interakcji pomiędzy nimi.

Uczestnicy szkolenia:

- zrozumieją potrzeby i korzyści oraz konsekwencje wdrożenia testów kontraktowych
- poznają receptury i kryteria doboru zakresu oraz szczegółowości testowanego kontraktu
- rozpoznają wzorce i antywzorce spotykane na każdym poziomie testów kontraktowych

Zalety szkolenia:

- Zagadnienia architektury systemu i testów gwarantujących bezpieczeństwo wdrożeń
- Najlepsze wzorce i praktyki testów kontraktowych
- Aspekty strategii shift-left w testowaniu oprogramowania

BO·TT·EGA
IT minds

Szczegółowy program:

1. Testy kontraktowe w testowaniu mikroserwisów

- 1.1. Czym jest Architektura Mikroserwisów
  - 1.1.1. Odpowiedzialność serwisów i całego systemu
  - 1.1.2. Odniesienie poziomu testów do poziomu abstrakcji modelu C4
- 1.2. Złamanie kontraktu: Zyski i koszty rozpraszania
- 1.3. Odpowiedzialność zespołów za serwisy i jakość / testy
- 1.4. Wzorce integracji i kontraktów
  - 1.4.1. Komunikacja Synchroniczna i Asynchroniczna
  - 1.4.2. Różnicowanie rodzaju kontraktu między Producentem a konsumentem (OpenHost, Conformist, Consumer-Supplier)
  - 1.4.3. Różnicowanie wzorców komunikatów: Komendy, Zdarzenia i Kwerendy

2. Testy kontraktowe dla Shift-left: Redukcja kosztu testów

- 2.1. Miejsce testów kontraktowych w piramidzie testowania
- 2.2. Szybka pętla zwrotna
- 2.3. Uzasadnienie i koszt dla testów kontraktowych

3. Wachlarz narzędzi zapewniających zgodność kontraktów

- 3.1. Dedykowane narzędzia: Pact i Spring Cloud Contract
- 3.2. Pozostałe gwarantujące podzbiór funkcjonalności: SDK, JSON Schema, scalanie, stabilizowanie, Docker Compose

4. Wdrożenie w projekcie

- 4.1. Strategia pierwszego minimalnego testu kontraktowego
- 4.2. Weryfikacja zgodności kontraktu po obu stronach integracji

5. Publikowanie kontraktu i wyniku weryfikacji

- 5.1. Wykorzystanie brokera do publikacji i weryfikacji
- 5.2. Definicja kontraktów ‘In Progress’ i ‘Pending’

BOTTEGA IT minds

6. Rozbudowane testy kontraktowe

6.1. Testy komunikacji asynchronicznej

6.2. Test Fixtures zapewniające utrzymywalność testów kontraktowych

6.3. Zarządzanie stanem i parametrami

7. Włączenie testów kontraktowych w proces CI/CD

7.1. Automatyczna weryfikacja i publikacja kontraktów

7.2. Raportowanie wdrożeń na środowiska

7.3. Włączenie testów kontraktowych w proces Code Review

8. Can I deploy: Bezpieczeństwo wdrożeń

8.1. Can I merge: weryfikacja czy Pull Request nie złamie kontraktu

8.2. Can I deploy: czy na danym środowisku oczekiwany kontrakt jest spełniony

9. Wzorce i antywzorce w testowaniu kontraktów

9.1. Strategia testowania stanowego procesu

9.2. Wybór reprezentatywnych zapytań i odpowiedzi

9.3. Unikanie błędów w build pipeline u innych zespołów

9.4. ‘Nie obiecuj i nie oczekuj nic ponad to co w kontrakcie’