

Program szkolenia:

Wprowadzanie testów automatycznych w z pozoru nietestowalnych projektach

Informacje:

Nazwa:	Wprowadzanie testów automatycznych w z pozoru nietestowalnych projektach
Kod:	craft-test-legacy
Kategoria:	Testowanie automatyczne
Odbiorcy:	testerzy, developerzy, architekci
Czas trwania:	3-4 dni
Forma:	20% wykłady, 80% ćwiczenia

Szkolenie przeznaczone jest dla zespołów, które potrzebują zobaczyć światło w tunelu.

W trakcie ćwiczeń warsztatowych testujemy i refaktoryjemy kod, którego złożoność odpowiada złożoności kodu z jakim zespół pracuje na co dzień.

Istnieje możliwość poprzedzenia szkolenia jednodniowym przeglądem kodu projektu klienta w celu oparcia na nim problemów warsztatowych.

Zalety szkolenia:

- Kompleksowe podejście łączące analizę, refaktoring i testowanie
- Sprawdzone techniki
- Event Storming jako technika wstecznej inżynierii

Szczegółowy program:

1. Testowanie kompletnej aplikacji w odizolowanym środowisku

- 1.1. Przygotowanie kontekstu testowania - start aplikacji (lub jej fragmentu) w izolacji
- 1.2. Przygotowanie danych oczekiwanych z innych serwisów
- 1.3. Wprowadzenie do JUnit i narzędzia podobne
- 1.4. Wprowadzenie do Mockito i narzędzia podobne
- 1.5. Wprowadzenie do Wire Mock i narzędzia podobne
- 1.6. Wybór optymalnych miejsc odizolowania aplikacji

2. Istotne zasady utrzymania testów

- 2.1. Typowe błędy i problemy
 - 2.1.1. Delikatne testy (fragile)
 - 2.1.2. Nieczytelne testy
 - 2.1.3. Wolne testy
 - 2.1.4. Testy niedeterministyczne
- 2.2. Budowanie biblioteki przykładowych danych testowych
 - 2.2.1. Stan bazy danych
 - 2.2.2. Danych oczekiwanych z innych serwisów
 - 2.2.3. Wzorce Object Mother i Test Data Builder
- 2.3. Enkapsulacja za pomocą wzorca Assert Object
- 2.4. Wzorzec Feature Object

3. Jak pozyskać pomoc testerów / analityków / biznesu w definiowaniu scenariuszy

- 3.1. Wprowadzenie do Gherkin i Cucumber for Java
- 3.2. Przygotowanie danych oczekiwanych z innych serwisów
- 3.3. Wykorzystanie biblioteki przykładowych danych testowych

4. Zwinny proces wytwarzania aplikacji z wczesnym uwzględnieniem testów akceptacyjnych i jednostkowych

4.1. Wydajne odkrywanie funkcjonalności i testów akceptacyjnych

4.1.1. Event Storming

4.2. Spisywanie testów akceptacyjnych w zespole krosfunkcyjnym

4.2.1. Specification by Examples i Acceptance Test Driven Development

4.3. Architektura aplikacji znacząco podnosząca testowalność

4.3.1. Porty i Adaptery, Walking Skeleton i Evolutionary Architecture

4.4. Refaktoring wspierający testowanie jednostkowe