Kod szkolenia: **ANDROID/TDD**

Tytuł szkolenia: **Test-Driven Development aplikacji Android**

Dni: 3

Opis:

Adresaci szkolenia:

Szkolenie adresowane jest do programistów Android, chcących podnieść swoje umiejętności tworzenia czystego, utrzymywalnego i testowalnego kodu.

Cel szkolenia:

Uczestnicy czują się pewnie stosując TDD w codziennej pracy swojego zespołu.

Jako że TDD jest trudne, szkolenie nie sprawi, że wszyscy staną się ekspertami. Jednak przeszkolenie całego zespołu daje dużą szansę, że zespół przyjmie tę praktykę.

W ramach szkolenia uczestnicy dowiedzą się czym różnią się poszczególne rodzaje testów. Poznają i przyswoją sobie cykl pracy TDD, nauczą się projektować oprogramowanie pod względem testowalności i tworzyć czytelny kod. Poznają biblioteki ułatwiające stosowanie TDD oraz umożliwiające testowanie na różnych poziomach.

Mocne strony szkolenia:

Szkolenie prowadzone jest w formie warsztatów. Uczestnicy przyswajają wiedzę w najskuteczniejszy możliwy sposób – praktykując TDD podczas serii ćwiczeń. Po szkoleniu czują i rozumieją, czym TDD jest i jak je zastosować w pracy.

Szkolenie adresuje tematy specyficzne dla tworzenia aplikacji androidowych. Uczestnicy dowiedzą się m. in., jak za pomocą TDD tworzyć Activities, jak komunikować się z serwisami REST.

Wymagania:

Od uczestników szkolenia wymagana jest umiejętność programowania na Androidzie w Javie.

Parametry szkolenia:

3*8 godzin (3*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Wielkość grupy: maks. 8-10 osób.

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: biuro@sages.com.pl web: www.sages.com.pl

Program szkolenia:

Podstawy

- dlaczego TDD może Ci pomóc
- definicja TDD
- stosowanie cyklu TDD Red-Green-Refactor
- rodzaje i poziomy testów: jednostkowe, integracyjne, akceptacyjne…

Testowanie jednostkowe

- co to jest test jednostkowy?
- możliwości JUnit
- pisanie czytelnych asercji przy użyciu biblioteki AssertJ i AssertJ Android
- testy parametryzowane z wykorzystaniem JUnitParams

Czytelne testy

- nazewnictwo testów
- struktura testów: given-when-then
- pisanie przejrzystych, utrzymywalnych testów
- organizacja testów

Testy współpracujących obiektów

- wykorzystanie dublerów testowych (Test Doubles) w celu izolacji klas testowanych
- różnice między mockami a stubami
- biblioteka Mockito - wygodny sposób generowania mocków w locie

Szybkie testy Aktywności z wykorzystaniem Robolectric

- czym jest Robolectric
- testowanie cyklu życia Aktywności
- symulowanie zachować użytkownika
- moduły rozszerzające Robolectric

Projektowanie obiektowe

- testability, czyli projektowanie pod kątem łatwego pisania testów
- zasady wspierające dobry design (SOLID principles, Inversion of Control, Dependency Injection, powiązania, spójność)

Dependency Injection na Androidzie

- koncepcja Dependency Injection
- DI bez dodatkowych bibliotek
- biblioteki DI na Androidzie

Refaktoryzacja

- najczęściej stosowane refaktoryzacje
- pełne wykorzystanie możliwości IDE - refaktoryzacje automatyczne, skróty klawiaturowe
- refaktoryzowanie małymi krokami przy przechodzących testach i kompilującym się kodzie

Test-drive'owanie kodu asynchronicznego

- sposoby testowania kodu asynchronicznego
- tworzenie test-driven klienta REST-owego API
- narzędzia pomocne w testowaniu klienta REST

Sages sp. z o.o., ul. Nowogrodzka 62c, 02-002 Warszawa
tel. +48 22 203 56 00 fax: +48 22 203 56 01
e-mail: biuro@sages.com.pl web: www.sages.com.pl