

# Program szkolenia:

## Aplikacje iOS klasy enterprise

### Informacje:

<b>Nazwa:</b>	<b>Aplikacje iOS klasy enterprise</b>
<b>Kod:</b>	<b>ios-enterprise</b>
<b>Kategoria:</b>	iOS
<b>Odbiorcy:</b>	developerzy, architekci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	30% wykłady / 70% ćwiczenia

Duże projekty wymagają specjalnego podejścia do kodu. Wiele prostych z pozoru aplikacji kryje pod spodem dużą, skomplikowaną logikę. Czasem jeden kod służy do budowy wielu aplikacji. Praca z takimi projektami pod wieloma względami nie jest łatwa. Wprowadzanie nowych funkcjonalności, poprawianie błędów czy chociażby podstawowe testowanie przestaje być trywialną rzeczą a zamienia się w ciężką pracę.

### Cel szkolenia

Enterprise iOS Applications to szkolenie które ma na celu wyposażyć uczestników w wiedzę niezbędną do poprawnego projektowania skalowalnych, modularnych i testowalnych aplikacji na iOS. W programie szkolenia zawarte są zagadnienia zarówno z obszarów dotyczących architektury aplikacji, testowania na wielu poziomach jak i projektowania rozwiązań za pomocą metody zyskującej coraz większą popularność, tj. Event Stormingu.

Szkolenie podzielone na dwa moduły: projektowy i techniczny.

Moduł projektowy to już warsztat z Event Stormingu podczas którego uczestnicy przeanalizują przedstawione wymagania biznesowe i zaprojektują aplikację.

Moduł techniczny zawiera zagadnienia z zakresu różnych architektur aplikacji oraz sposobów ich testowania. Ma na celu uzupełnienie lub wyposażenie uczestników w wiedzę niezbędną do zaawansowanego projektowania modularnych i testowalnych aplikacji.

### Dla kogo

Warsztat jest skierowany dla programistów iOS mających już doświadczenie z platformą.

### Wymagane jest:

- znajomość zagadnień związanych z programowaniem w środowisku XCode
- znajomość języka Swift
- podstawy testowania (znajomość XCTest, uruchamianie testów)

Idealny uczestnik szkolenia powinien być w stanie samodzielnie napisać aplikację (interfejs, komunikacja przez internet, przechowywanie danych) i umieć przetestować w dowolny sposób wybrane przez siebie fragmenty kodu.

## Przebieg szkolenia

Szkolenie Enterprise iOS Applications trwa 3 dni. Wszystkie zagadnienia omawiane będą w ramach wirtualnego projektu. W trakcie szkolenia uczestnicy przeanalizują istniejący proces biznesowy. Zaprojektują nową funkcjonalność istniejącej aplikacji ucząc się przy tym jak zmodyfikować istniejący kod w taki sposób aby udało się osiągnąć dodatkowo kilka istotnych aspektów takich jak modularność, testy logiki biznesowej oraz interfejsu użytkownika.

Pierwszy dzień to warsztat Event Stormingowy. Uczestnicy szkolenia przeanalizują istniejący proces biznesowy oraz aplikację. Na podstawie przedstawionych wymagań biznesowych zaprojektują nowy moduł do istniejącej aplikacji który w kolejnych dniach szkolenia będą musieli napisać i zintegrować z istniejącą aplikacją. Wymagania skonstruowane są w taki sposób aby jak najbardziej wykorzystać Event Storming do analizy procesów biznesowych (Big Picture) na podstawie których zostanie zaprojektowana logika biznesowa aplikacji (Design Level). Tego dnia uczestnicy odkryją najlepszy sposób na projektowanie funkcjonalności aplikacji pozwalający ominąć silosy kompetencyjne w dużych firmach.

Drugiego dnia uczestnicy zapoznają się (lub uzupełnią wiedzę) z podstawowymi zagadnieniami związanymi z budowaniem aplikacji. Przyjrzymy się między innymi takim wzorcom jak MVC, MVVM czy MVP w kontekście aplikacji iOS z nastawieniem na poprawne testowanie poszczególnych obszarów (testy jednostkowe, funkcjonalne czy integracyjne). Tego dnia głównym zadaniem będzie przeprowadzenie refaktoru kodu istniejącej aplikacji w taki sposób aby możliwe było przetestowanie zadanych obszarów i wydzielenie niektórych jej funkcji do osobnych modułów.

Trzeciego dnia zadaniem grupy będzie napisanie nowego, zaprojektowanego wcześniej modułu aplikacji. Tego dnia przyjrzymy się dokładnie architekturze hexagonalnej i sposobie jej testowania. Poruszona zostanie również kwestia poprawnego testowania interfejsu użytkownika z naciskiem na zagadnienia związane z najczęściej spotykanymi problemami (np. jak dotrzeć do n-tego ekranu żeby móc go przetestować).

## Zalety szkolenia:

- Szeroki zakres wzorców budowania aplikacji z nastawieniem na testowanie
- Praktyczne porady przełożenia teorii na projekt
- Wprowadzenie do Event Stormingu

## Szczegółowy program:

### 1. Event Storming

#### 1.1. Wstęp do Event Stormingu

1.1.1. Problem jaki chcemy rozwiązać

1.1.2. Stosowalność

#### 1.2. Metodyka

1.2.1. Mechanika

1.2.2. Role i odpowiedzialność

#### 1.3. Sesja strategiczna (Big Picture) - wprowadzenie dla iOS developerów

1.3.1. Łagodne wprowadzenie w nomenklaturę poprzez zastosowanie uproszczonej notacji (no-DDD)

1.3.2. Odkrycie procesów biznesowych

1.3.3. Destylacja kontekstów

#### 1.4. Sesja taktyczna (Design Level) - wprowadzenie dla iOS developerów

1.4.1. Wprowadzenie w nomenklaturę z zastosowaniem uproszczonej notacji (no-DDD)

1.4.2. Projektowanie algorytmów na podstawie odkrytego/przeanalizowanego procesu biznesowego

#### 1.5. Praca z istniejącym kodem

1.5.1. Znajdowanie punktów styku z nowo projektowanymi rozwiązań z istniejącym kodem

#### 1.6. Jak przełożyć wynik Event Stormingu na projekt.

### 2. Budowa aplikacji iOS

#### 2.1. Przegląd podstawowych zagadnień.

#### 2.2. Wzorce architektoniczne

2.2.1. MVC

2.2.2. MVVM

2.2.3. MVP

2.3. Wzorce projektowe

2.3.1. Delegate

2.3.2. Data Source

2.4. Modularność

2.4.1. Dlaczego poprawna hermetyzacja jest tak ważna?

2.4.2. Architektura hexagonalna

2.4.2.1. Praktyczne przykłady zastosowań w iOS

### 3. Testowanie aplikacji

3.1. Jak pisać kod podatny na testowanie

3.1.1. Dobre praktyki: SOLID

3.1.2. Jak zmieniać istniejący kod do dającego się testować

3.1.3. Pułapki i typowe błędy

3.2. Zakres testów

3.2.1. Testowanie jednostkowe

3.2.2. Testowanie integracyjne

3.2.3. Testowanie funkcjonalne

3.3. Wybrane wzorce i techniki testowe - omówienie i przykłady zastosowania

3.3.1. Mock

3.3.2. Stub

3.3.3. Sut

3.3.4. Memory implementation

3.3.5. Test Driven Development

3.3.5.1. Cykl czerwony-zielony-refaktoring

3.3.5.2. Ewolucyjny rozwój kodu

3.3.5.3. Podstawowe techniki refaktoringu

3.4. Testowanie interfejsów użytkownika

3.4.1. Co testować?

3.4.2. Page Object Pattern

3.4.3. Wstrzykiwanie stanu.

3.4.4. Architektura pozwalająca na testy dowolnego ekranu aplikacji bez konieczności "dochodzenia" do niego.