

## Program szkolenia:

# Architektura dla analityków

### Informacje:

<b>Nazwa:</b>	<b>Architektura dla analityków</b>
<b>Kod:</b>	<b>analiza-an</b>
<b>Kategoria:</b>	Analiza biznesowa
<b>Odbiorcy:</b>	analitycy
<b>Czas trwania:</b>	5 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Szkolenie adresowane jest do analityków biznesowych, którzy chcieliby zapoznać się z architekturą systemów, aby lepiej komunikować się z architektami oraz zrozumieć cele i rozwiązania architektoniczne. Tym samym wiedzieć nie tylko, jak lepiej definiować wymagania niefunkcjonalne, ale również jak wykrywać w trakcie analizy aspekty istotne dla architektury, potencjalne rozwiązania, ograniczenia, ryzyka. Szkolenie bardzo pomoże również analitykom, na których spadło utrzymanie istniejącej architektury.

Celem szkolenia jest zdobycie wiedzy niezbędnej do wykrywania zagrożeń architektonicznych, tak aby analityk biznesowy stał się pierwszą linią tworzenia założeń architektonicznych już podczas wywiadów z klientem. Materiał pozwala nie tylko zrozumieć cele i rozwiązania architektoniczne, ale także ułatwia ustalanie szczegółów z klientem podczas rozwoju istniejących systemów, dzięki zrozumieniu aktualnej budowy systemu i osiągniętych przez niego parametrów.

Za cel szkolenia obrano także wsparcie analityków, na których spadło utrzymanie istniejącej architektury. Szkolenie kładzie duży nacisk na osiągnięcie wysokiej świadomości konsekwencji związanych z doбором rozwiązań, technologii, wzorców i innych decyzji architektonicznych. W oparciu o tę świadomość ćwiczymy i budujemy umiejętność podejmowania i weryfikacji decyzji architektonicznych, poruszając się w realiach nieklarownych wizji systemu i dużej ilości założeń architektonicznych. Decyzje rozpatrywane są z punktu zalet i wad dla poszczególnych parametrów systemowych.

W ramach szkolenia omawiany jest również zakres popularnych technologii z podziałem na poszczególne warstwy oraz architektoniczne diagramy UML na poziomie wystarczającym do zrozumienia budowy systemu.

## Szczegółowy program:

### 1. Podstawy Architektury

- 1.1. Czym jest architektura
- 1.2. Kim jest architekt i jaką pełni rolę?
- 1.3. Zarządzanie ryzykiem

### 2. Parametry systemowe

- 2.1. Czym są parametry systemowe
- 2.2. Jak poprawnie definiować wymagania нефunkcjonalne
- 2.3. Opis parametrów systemowych:
  - 2.3.1. Wygoda użytkownika, Bezpieczeństwo, Wydajność,
  - 2.3.2. Dostępność, Niezawodność, Elastyczność,
  - 2.3.3. Realizowalność, Testowalność, Serwisowalność, Zarządzalność
- 2.4. Wymiary systemu
- 2.5. Priorytety parametrów systemowych

### 3. Wzorce architektoniczne

- 3.1. Wprowadzenie do wzorców
- 3.2. Stable Dependency Principle
- 3.3. Wzorce podziału odpowiedzialności: MVC a SPA, N-Tier i Layers
- 3.4. Wzorce infrastruktury:
  - 3.4.1. Skalowanie poziome i pionowe, Modularyzacja
  - 3.4.2. Load Balancing, Redundancja ścieżek, Klastry, Chmury
- 3.5. Wzorce EAI: Message Broker, SOA, ESB i mu podobne
- 3.6. Microservices
- 3.7. Przedstawienie wybranych szablonów wzorców

#### 4. Microservices z bliska

- 4.1. Jak ma się do SOA
- 4.2. A jak do Monolitu
- 4.3. Wady i zalety Microservices
- 4.4. Usługi w chmurze: IaaS, CaaS, Paas, FaaS, SaaS
- 4.5. Kiedy wybrać, kiedy nie, i jak zacząć
- 4.6. Dane w Microservices i podejście BASE
- 4.7. Wzorce pomocnicze: CQRS i Event Sourcing

#### 5. Architektura warstwy klienta i prezentacji

- 5.1. Przechowywanie sesji
- 5.2. Podział klientów
- 5.3. Technologie klienta grubego: biblioteki graficzne a RCP
- 5.4. Wsparcie klienta cienkiego: dla MVC, SPA (Angular, Vue, React)
- 5.5. Zaawansowane komunikacja z przeglądarką: AJAX, WebSocket

#### 6. Architektura warstwy biznesowej

- 6.1. Przetwarzanie rozproszone
- 6.2. Komunikacja zdalna a lokalna
- 6.3. Protokoły komunikacyjne:
  - 6.3.1. COBRA, SOAP, REST, GraphQL, Sockets, RMI, gRPC...

#### 7. Architektura warstwy integracji i zasobów

- 7.1. Technologie utrwalania danych:
  - 7.1.1. RDBMS, bazy hierarchiczne, NoSQL, BigData
- 7.2. Komunikacja asynchroniczna

#### 8. Przegląd architektury w UML

- 8.1. Diagram komponentów (component diagram)

## 8.2. Diagram wdrożenia (deployment diagram)