

## SZKOLENIE ZAAWANSOWANE

---

# Wzorce projektowe i architektoniczne, architektura aplikacji dla projektantów

ARCH/PRO

Czas trwania: 5 dni (40h)

Architektura z punktu widzenia projektanta, czyli jak ją zrozumieć i zachować wyznaczone cele przy projektowaniu

### Cele szkolenia

---

- Celem szkolenia jest zdobycie umiejętności projektowania przy wybranych wzorcach architektonicznych z użyciem wzorców projektowych. Innymi słowy uwzględnienie w projekcie ograniczeń architektonicznych narzuconych przez model architektury (zawarte w nim warstwy, stosy technologiczne, rozmieszczenie komponentów i protokoły komunikacji)
- Na szkoleniu uczestnik poznaje parametry systemowe oraz wzorce architektoniczne od strony ich struktury i osiągania określonych celów (wyrażonych we wspomnianych parametrach systemowych)
- Uczymy się struktury i modelowania w UML na poziomie architektonicznym, a następnie przechodzimy do poziomu projektowego
- Na poziomie projektowym uwzględniamy ograniczenia architektury przy modelowaniu przypadków użycia, jak również uszczegóławiamy architekturę na poziomie diagramów architektonicznych
- Aby to osiągnąć poznajemy wspomniane parametry systemowe, wybrane diagramy UML oraz same wzorce architektoniczne i projektowe
- Szkolenie pomija wiele aspektów niezbędnych architektom do podjęcia decyzji architektonicznych, gdyż te obszary omawiane są na szkoleniu dedykowanym dla architektów

### Zalety

---

- Jakościowe spojrzenie na użyte wzorce
- Umiejętne użycie wzorców z zachowaniem ograniczeń architektonicznych
- Zrozumienie architektury
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretnie umiejętności - w ramach każdego szkolenia rozwijamy praktyczne umiejętności związane z daną technologią i tematyką
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how



## Dla kogo?

---

- Głównymi adresatami szkolenia są projektanci oraz osoby planujące wykonywać czynności projektowe
- Szkolenie adresowane jest do osób, które chciałyby zapoznać się z wzorcami architektonicznymi oraz projektowymi od strony aspektów projektowych
- Za przydatne uznają je osoby, które muszą tworzyć projekt dla wybranej już architektury. Dowiedzą się nie tylko, na co zwracać uwagę przy konkretnych wzorcach architektonicznych, ale poznają też kluczowe przesłanki doboru konkretnej architektury. Dzięki temu nauczą się jak stworzyć projekt, który nie będzie kłócił się z założeniami architektonicznymi i pozwoli na osiągnięcie założonych poziomów parametrów systemowych

## Wymagania

---

- Szkolenie wprowadza do zagadnień projektowych od podstaw. Zarówno w zakresie zrozumienia cech poznawanych wzorców architektonicznych, modelowania w UML, jak i kluczowych aspektów modelowania na poziomie projektowym
- W związku z tym szkolenie nie posiada wymagań wstępnych stawianych uczestnikom aczkolwiek mile widziana jest znajomość JEE, gdyż czas szkolenia nie pozwala na pełne omówienie ograniczeń zawartych tam technologii, a część ćwiczeń odwołuje się do nich

## Program

---

1. Parametry systemowe
  - a. Czym są parametry systemowe
  - b. Jak poprawnie definiować wymagania niefunkcjonalne
  - c. Parametry systemowe
    - Usability, Security, Performance
    - Scalability, Availability, Reliability
    - Extensibility, Reusability, Portability, Flexibility
    - Realizability, Planability, Testability
    - Maintainability, Serviceability, Manageability
  - d. Wymiary systemu i ich wpływ na parametry systemu
2. Wzorce architektoniczne
  - a. Wprowadzenie do wzorców
  - b. Stable Dependency Principle
  - c. Wzorce podziału odpowiedzialności
    - MVC i SPA, Desktop, Client-Server
    - Web-centric, Application-centric, Enterprise
    - Architektura wielowarstwowa (Layers Pattern)
  - d. Wzorce infrastruktury
    - Redundancja Ścieżek, Skalowanie poziome i pionowe
    - Load Balancing, Reverse Proxy, Cloud
    - Clustering, HA, Failover
  - e. Wzorce EAI (Enterprise Application Integration)



-MOM, SOA, ESB

f. Microservices i technologie

-Microservices a SOA

-Microservices a monolit

-Client-side a Server-side service discovery

-DevOps i Continuous Delivery/Deployment

-Zalety i problemy microservices

-On Premise(s), IaaS, CaaS, PaaS, FaaS, SaaS

-Microservices i skalowanie 3D

-Jak tworzyć architekturę microservices

-Polyglot Persistence

-CQRS i Event Sourcing jako wsparcie microservices

-Niespójne dane, czyli ACID kontra BASE

g. Słów kilka o szablonach: Microservices Patterns, POSA, PEAA, Core J2EE, DDD, EIP

3. Modelowanie architektury w UML

a. Diagram komponentów

b. Diagram wdrożenia

4. Zaawansowane aspekty modelowania architektury w UML (opcjonalne)

a. Instancyjne diagramy wdrożenia

b. Niskopoziomowe diagramy wdrożenia

c. Szablony architektoniczne

d. Model wdrożenia na diagramach wdrożenia (artefakty)

e. Diagram pakietów

5. Przejście z architektury do projektu

a. Warstwy i komponenty a realizacja projektu

b. Warstwy i komponenty a model projektowy

c. Uwzględnienie ograniczeń architektury w projekcie

6. Modelowanie projektu w UML - wybrane diagramy

a. Diagram klas

-Klasa i jej elementy

-Klasy i metody abstrakcyjne

-Interfejs

-Relacje

-Realizacja

-Dziedziczenie

-Zależność i wybrane stereotypy

-Asocjacja i jej cechy

-Agregacja i kompozycja

b. Diagram sekwencji

-Linia życia

-Rodzaje komunikatów

-Synchroniczny

-Asynchroniczny

-Zwrotny



- Tworzenie i niszczenie obiektu

- Blokki złożone: alt, break, loop, par

## 7. Wzorce projektowe a architektura

a. Jak wzorce projektowe wpływają na architekturę

b. Wybrane wzorce GoF

- Factory Method (flexibility)
- Abstract Factory (reliability, flexibility)
- Builder (reliability, flexibility)
- Prototype (performance)
- Singleton (performance)
- Façade (performance, flexibility)
- Command (flexibility)
- Strategy (flexibility)
- Adapter (flexibility)
- Mediator (flexibility)
- Observer (performance, flexibility)
- Memento (reliability)
- Proxy (flexibility, performance)
- Decorator (flexibility)

c. Wybrane wzorce Core J2EE

- Warstwa Prezentacji
  - Intercepting Filter (flexibility)
  - Context Object (maintenance, flexibility)
  - Service To Worker (flexibility)
- Warstwa Biznesowa
  - Business Delegate (maintenance)
  - Service Locator (maintenance)
  - Session Façade (performance, flexibility)
  - Transfer Object (performance)
  - Application Service (flexibility, maintenance)
  - Business Object (flexibility, maintenance)
- Warstwa Integracji
  - DAO (flexibility)

