Kod szkolenia: **ARCH/PRO**

Tytuł szkolenia: **Wzorce projektowe i architektoniczne, architektura aplikacji dla projektantów**

Dni: 5

Opis:

Adresaci szkolenia

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. Ze względu na to głównymi adresatami szkolenia są projektanci oraz osoby planujące wykonywać czynności projektowe.

Cel 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 wspominane 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.

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 żadnych 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).

Parametry szkolenia

5*7 godzin wykładów i warsztatów. Proporcje warsztatów różnią się w poszczególnych partiach materiału. Pierwszego dnia przeważa wykład, w pozostałe dni zdecydowanie przeważa część warsztatowa. Dzień pierwszy obejmuje parametry systemowe i wzorce architektoniczne - jest to fundament dalszej pracy i opiera się głównie na zrozumieniu kluczowych cech poszczególnych architektur z perspektywy osiągania parametrów systemowych. Drugiego dnia uczymy się modelowania architektury przy użyciu UML w narzędziu Enterprise Architect. Trzeciego dnia uczymy się projektowania z wyszczególnieniem różnicy między analizą i projektem (dla ułatwienia pracy projektantom, którym nie dostarczono modelu analitycznego). Czwartego dnia projektujemy z użyciem wzorców GOF. Piątego dnia projektujemy z użyciem wzorców Core J2EE.

Program szkolenia:

1. Parametry systemowe
    - Czym są parametry systemowe
    - Jak poprawnie definiować wymagania niefunkcjonalne
    - Parametry systemowe
        - Wygoda użytkownika (Usability)
        - Bezpieczeństwo (Security)
        - Wydajność (Performance)
            - Przepustowość (Throughput)
            - Czas odpowiedzi (Response Time)
            - Czas reakcji (Responsivness)
        - Dostępność (Availability)
        - Niezawodność (Reliability)
        - Skalowalność (Scalability)
        - Różne wymiary elastyczności systemu
            - Rozszerzalność (Extensibility)
            - Reużywalność (Reusability)
            - Przenaszalność (Portablity)
            - Elastyczność (Flexibility)
        - Realizowalność (Realizability)
        - Planowalność (Planability)
        - Testowalność (Testability)
        - Utrzymanie (Maintainability)
        - Serwisowalność (Serviceability)
        - Zarządzalność (Managebility)
    - Wymiary systemu
        - Wymiary związane z infrastrukturą
            - Pojemność (Capacity)
            - Redundantność/Replikacja (Redundancy)

- Modułowość (Modularity)
- Wpływ wymiarów na parametry systemu
- Inne wymiary systemu
  - Tolerancja (Tolerance)
  - Obciążenie (Workload)
  - Niejednorodność/Jednorodność (Homo/Heterogenity)
- Priorytety parametrów systemu
  - Skąd wynikają priorytety?
  - Problemy priorytetowania

2. Wzorce architektoniczne
   - Wprowadzenie do wzorców
     - Definicja wzorca
     - Cechy i zalety wzorców
     - Rodzaje wzorców
   - Wzorce architektoniczne
     - Problemy architektury komponentowej
     - Wzorce podziału odpowiedzialności
       - MVC (Model View Controll)
       - Web-centric
       - Application-centric
       - Enterprise
         - Wymagania systemów Enterprise
         - Architektura Enterprise w JEE
       - Architektura wielowarstwowa (Layers Pattern)
         - Architektura wielowarstwowa w JEE
     - Wzorce EAI (Enterprise Application Integration)
       - SOA (Service Oriented Architecture)
       - ESB (Enterprise Service Bus; Szyna Danych; Broker Integracyjny)
       - MOM (Message Oriented Middleware)
       - Microsercives
         - Założenia
         - Monolit a Microservices
         - DB w monolicie a w Microservices
         - Dlaczego Microservices?
         - Problemy z microsercives
     - Wzorce infrastruktury
       - Redundancja Ścieżek
       - Skalowanie pionowe
       - Skalowanie poziome (Replikacja)
       - Równoważenie obciążenia (Load Balancing)
       - Klastry (Clustering)
         - HA (High Availability)
         - Failover
       - Forward Proxy Cache
     - Wzorce blokowania zasobów

- Blokowanie optymistyczne (Optimistic Lock)
- Blokowanie pesymistyczne (Pessimistic Lock)
  - Blokada domyślna (Implicit Lock)
  - Blokowanie gruboziarniste (Coarse-Grained Lock)

- Inne wzorce architektoniczne
  - Dependency Injection
  - Dependency Inversion
  - Stable Dependency Principle
  - CQRS
  - Event Sourcing
  - Szablon wzorców POSA
  - Szablon wzorców PEAA
  - Szablon wzorców Core J2ee
  - Szablon wzorców EAI (wzorce JMS)

3. Wprowadzenie do UML
   - Czym jest modelowanie
   - Czym jest a czym nie jest UML
   - Rozwój UML
   - Podstawowe elementy UML
     - Podstawowe kwalifikatory
       - Klasa (Class)
       - Interfejs (Interface)
       - Obiekt (Object)
       - Przypadek Użycia (Use Case)
       - Komponent (Component)
       - Węzeł (Node)
     - Relacje (Relationships)
       - Asocjacja (Association)
       - Asocjacja (Association)
       - Zależność (Dependency)
       - Realizacja (Realization)
     - Diagramy (Diagrams)
     - Komentarze (Note)
     - Mechanizmy rozszerzenia
       - Stereotypy (Stereotype)
       - Etykiety (Tagged Values)
       - Ograniczenia (Constraints)
   - Diagram a model UML
   - Zastosowania UML
4. Modelowanie architektury w UML
   - Diagram komponentów (component diagram)
     - Komponent (component)
     - Komponenty zagnieżdżone
     - Interfejs (interface)
       - Interfejs wymagany (required interface)
       - Interfejs dostarczany (provided interface)

        - Złączenie (assembly)
    - Diagram wdrożenia (deployment diagrram)
        - Węzeł (node)
        - Łącze (communication path)
            - Łącze kierunkowe
            - Liczność łącza
        - Porty
        - Konektory
        - Realizacja komponentu
5. Zaawansowane aspekty modelowania architektury w UML
    - Zaawansowane elementy diagramu wdrożenia (deployment diagram)
        - Instancyjne diagramy wdrożenia
        - Niskopoziomowe diagramy wdrożenia
        - Szablony architektoniczne
        - Model wdrożenia na diagramach wdrożenia
            - Po co model wdrożenia
            - Artefakt
            - Stereotypy artefaktów
                - file
                - document
                - library
                - executable
                - script
                - source
            - Specyfikacja konfiguracji (deployment specification)
            - Relacje między artefaktami
                - Kompozycji (composition)
                - Zależności (dependency)
            - Instalacja artefaktów (deployment)
            - Manifestacja (manifestation)
    - Diagram pakietów (package diagram)
        - Pakiet
        - Zagnieżdżanie (nest, nesting)
        - Przestrzeń nazw
        - Importowanie (package import)
            - import
            - access
        - Łączenie (merge)
        - Diagramy pakietów i modelowanie warstw architektury
            - Przecięcie warstw i poziomów
    - Modelowanie w C4 (Context, Container, Component, Class)
6. Przejście z architektury do projektu
    - Warstwy i komponenty a realizacja projektu
    - Warstwy i komponenty a model projektowy
    - Uwzględnienie ograniczeń architektury w projekcie
        - Na modelu statycznym

        - Na modelu dynamicznym
7. Modelowanie projektu w UML - wybrane diagramy
    - Diagram klas (class diagram)
        - Klasa (class)
            - Elementy klasy (atrybuty, metody)
            - Widoczność (visibility)
            - Atrybuty i metody statyczne
        - Uogólnienie (generalization)
        - Klasy abstrakcyjne (abstract class)
            - Metody abstrakcyjne
        - Interfejs (interface)
        - Realizacja (realization)
        - Relacja zależności
            - Wybrane stereotypy zależności
                - instanciate
                - send
                - call
        - Asocjacja (association)
            - Cechy asocjacji
                - Nazwa asocjacji (name)
                - Rola (role)
                - Nawigowalność (navigablility)
                - Wielokrotność (multiplicity)
            - Asocjacja zwrotna i wielokrotna
            - Rodzaje asocjacji
                - Asocjacja (association)
                - Agregacja (aggregation)
                - Kompozycja (composition)
            - Klasa asocjacyjna (association class)
    - Diagram sekwencji (interaction diagram)
        - Linia życia (life line)
        - Komunikat (message)
        - Rodzaje komunikatów
            - Synchroniczny (synchronous message)
            - Asynchroniczny (asynchronous message)
            - Zwrotny (return message)
            - Utworzenie obiektu create
            - Zniszczenie obiektu destroy i destruction event
        - Wybrane bloki złożone (combined fragment)
            - Alternatywy (alternatives) – alt
            - Pętla (loop) – loop
            - Współbieżność (parallel) – par
        - Bloki złożone a notacja w UML 1.X
            - Dawniej alternatywy
            - Dawniej pętle (iteracja)
            - Dawniej współbieżność

8. Wzorce projektowe a architektura
    - Jak wzorce projektowe mogą wpływać na architekturę
    - 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)
        - Template Method (reliability, flexibility)
        - Bridge (reliability, flexibility)
        - Memento (reliability)
        - Visitor (flexibility)
        - Flyweight (memory performance)
        - Chain of responsibility (flexibility)
        - Proxy (flexibility, performance)
        - Decorator (flexibility)
    - 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)
            - Value List Handler (performance, scalability)
            - Application Service (flexibility, maintenance)
            - Business Object (flexibility, maintenance)
        - Warstwa Integracji
            - DAO (flexibility