

SZKOLENIE ZAAWANSOWANE

Architektura systemów dla analityków

J/ARCH/ANA

Czas trwania: 3 dni

Współpraca z architektem i utrzymanie architektury przez analityka

Cele szkolenia

.....

- Celem szkolenia jest nabycie wiedzy koniecznej do wykrywania zagrożeń architektonicznych, dzięki czemu analityk biznesowy już w trakcie wywiadów z klientem będzie w stanie dokonywać pierwszych założeń architektonicznych, co z kolei pozwoli na zgłębienie kluczowych aspektów. Aby to osiągnąć konieczne jest zrozumienie nie tylko rozwiązań architektonicznych, ale również celów do osiągnięcia. Jest to wiedza przydatna nie tylko podczas tworzenia nowych systemów, gdyż zrozumienie aktualnej budowy systemu i osiąganych dzięki temu parametrów systemowych ułatwi ustalanie szczegółów podczas rozwoju istniejących systemów
- Poznanie wybranych technologii wspierających złożone architektury z podziałem na warstwy, w tym specyfikacje z JEE. Uczestnicy poznają główne cechy wybranych technologii oraz klasy rozwiązań, co ułatwia dalsze poszukiwania, gdyby ograniczenia obecnej architektury wymagały innego środowiska
- Omówienie diagramów architektonicznych w UML na poziomie wystarczającym do zrozumienia budowy systemu. Ponieważ adresatami szkolenia są analitycy, a nie architekci, jedynie zapoznajemy się z modelami architektonicznymi i nie rozwijamy umiejętności ich tworzenia (zajmujemy się tym w wersji pięciodniowej dla architektów)

Zalety

.....

- Duży nacisk kładziony jest na osiągnięcie wysokiej świadomości konsekwencji doboru poszczególnych rozwiązań, technologii, wzorców czy innych decyzji architektonicznych
- Podczas ćwiczeń budowana jest umiejętność doboru rozwiązań w warunkach nieklarownych wizji systemu czy dużej ilości założeń. Rozwiązania rozpatrujemy z poziomu konsekwencji (wady i zalety) dla poszczególnych parametrów systemu

Dla kogo?

.....

- Szkolenie przeznaczone jest dla analityków biznesowych chcących zapoznać się z architekturą systemów, w celu zrozumienia stosowanych rozwiązań oraz usprawnienia komunikacji z architektami. Pozwala to lepiej definiować wymagania niefunkcjonalne oraz wykrywać aspekty



istotne dla architektury (ograniczenia, ryzyka, potencjalne rozwiązania)

Wymagania

- Wprowadzamy do zagadnień architektury od podstaw, zatem brak wymagań wstępnych



Program

1. Podstawy Architektury

- a. Czym jest architektura
- b. Cele tworzenia architektury
- c. Kim jest architekt i jaką pełni rolę
- d. Proces architektoniczny
- e. Dokumentacja architektoniczna
- f. Zarządzanie ryzykiem

2. Parametry systemowe

- a. Czym są parametry systemowe
- b. Jak poprawnie definiować wymagania нефункционалне
- 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
- e. Priorytetyzacja parametrów systemowych

3. 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 i CAP Theorem

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

4. Architektura warstwy klienta i prezentacji

- Przechowywanie sesji
- Podział klientów: gruby, cienki, RIA
- Technologie klienta grubego: Swing, SWT, RCP
- Technologie klienta cienkiego
 - HTML Statyczny i dynamiczny
 - MVC na przykładzie JSF
 - SPA/SPI (Angular, React, Vue)
 - AJAX i jego wsparcie (Java Script, Prototype, Ajax4JSF, PrimeFaces)
 - WebSocket

5. Architektura warstwy biznesowej

- Przetwarzanie rozproszone
- Komunikacja zdalna a lokalna
- Protokoły komunikacyjne:
 - CORBA i IIOP,
 - Web Services SOAP i REST,
 - Sockets (własny protokół),
 - RMI, EJB i RMI-IIOP,
 - Inne rozwiązania na rynku

6. Architektura warstwy integracji i zasobów

- Technologie utrwalania danych:
 - Bazy relacyjne, hierarchiczne, NoSQL
- Komunikacja asynchroniczna
- Systemy legacy
- Screen Scrapping

7. Odczyt architektury opisanej w UML

- Diagram komponentów
- Diagram wdrożenia

8. Zaawansowane aspekty architektury w UML

- Instancyjne diagramy wdrożenia
- Niskopoziomowe diagramy wdrożenia
- Szablony architektoniczne
- Model wdrożenia na diagramach wdrożenia (artefakty)
- Diagram pakietów

