

SZKOLENIE ZAAWANSOWANE

Architektura systemów

J/ARCH

Czas trwania: 5 dni (40h)

* Tworzenie architektury na podstawie parametrów systemowych

Cele szkolenia

- Zdobyć wiedzy niezbędnej do tworzenia i weryfikacji architektury oraz umiejętności rozpatrywania potencjalnych rozwiązań z punktu widzenia parametrów systemowych
- Poznanie języka UML w zakresie modelowania architektury i umiejętności tworzenia modeli architektonicznych

Zalety

- Szkolenie kładzie duży nacisk na osiągnięcie wysokiej świadomości konsekwencji związanych z doborem rozwiązań, technologii, wzorców i innych decyzji architektonicznych
- Budujemy umiejętność podejmowania i weryfikacji decyzji architektonicznych, poruszając się w realiach nieklarownych wizji systemu i dużej ilości założeń architektonicznych
- Wzorce architektoniczne omawiane na poziomie koncepcyjnym (niezależnym od języka) i technologicznym (z oznaczeniem rozwiązań hetero i homogenicznych)
- Sposób omawiania pozwala uczestnikom na odnajdywanie w przyszłości nowszych technologii danej klasy czy związanych z danym językiem rozwiązań homogenicznych
- 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?

- Szkolenie jest odpowiednie dla programistów, projektantów, analityków, jak również dla architektów chcących usystematyzować wiedzę i wymienić doświadczenia w grupie
- Dla osób, które chciałyby zapoznać się z praktycznymi aspektami tworzenia architektury
- Dla wszystkich, którzy chcą otworzyć przed sobą nowe możliwości w zakresie realizacji zadań związanych z kompetencjami architekta
- Dla osób pragnących podejmować lepsze decyzje poprzez osiągnięcie wyższej świadomości konsekwencji płynących z dobranych rozwiązań

Wymagania



- 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 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
 - 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
 - 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
- Kiedy migrować do microservices

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

4. Prototypowanie

- a. Po co prototypować
- b. Prototyp Proof of Concept
- c. Prototyp ewolucyjny
- d. Antywzorzec Lava Flow

5. Architektura warstwy klienta i prezentacji

- a. Przechowywanie sesji
- b. Podział klientów: gruby, cienki, RIA
- c. Technologie klienta grubego: Swing, SWT, RCP
- d. 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

6. Architektura warstwy biznesowej

- a. Przetwarzanie rozproszone
- b. Komunikacja zdalna a lokalna
- c. Optymalizacja komunikacji sieciowej
- d. Protokoły komunikacyjne
 - CORBA i IIOP
 - Web Services SOAP i REST
 - GraphQL, Sockets (własny protokół)
 - RMI
 - gRPC i inne rozwiązania na rynku
- e. Serwery aplikacji i kryteria wyboru

7. Architektura warstwy integracji i zasobów

- a. Technologie utrwalania danych:
 - Bazy relacyjne, hierarchiczne, NoSQL, rozwiązania Big data
- b. Komunikacja asynchroniczna
 - Do czego przydatna
 - Jak zwrócić odpowiedź
 - API kontra protokół, czyli JMS a AMQP
 - Czym są wzorce EIP i jakie mają wsparcie
 - Przykładowe rozwiązania MOM: RabbitMQ, Redis, Kafka
 - Modele komunikacji w różnych implementacjach
 - Co uwzględnić przy wyborze message brokera?
- c. Systemy legacy
- d. Screen Scrapping
- e. Technologie scaffolding'owe



8. Modelowanie architektury w UML
 - a. Diagram komponentów
 - b. Diagram wdrożenia
9. Zaawansowane aspekty modelowania architektury w UML
 - a. Instancyjne diagramy wdrożenia
 - b. Niskopoziomowe diagramy wdrożenia
 - c. Szablony architektoniczne
 - d. Artefakty na diagramach wdrożenia
 - e. Diagram pakietów
10. 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
11. Weryfikacja i ocena architektury
 - a. Po co weryfikować?
 - b. Zespół weryfikujący
 - c. Techniki weryfikacji i oceny
 - d. Proces weryfikacji
 - e. Raport z weryfikacji

