SZKOLENIE ZAAWANSOWANE

Architektura systemów

J/ARCH

Czas trwania: 5 dni

* Tworzenie architektury na podstawie parametrów systemowych

Cele szkolenia

- Zdobycie 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 i na poziomie koncepcyjnym - niezależnym od języka, i na poziomie technologicznym - z oznaczeniem rozwiązań hetero i homogenicznych
- Sposób omawiania pozwala uczestnikom na odnajdywanie w przyszłości nowszych technologii danej klasy czy związanych z ich językiem rozwiązań homogenicznych (na szkoleniu przykłady homogeniczne podawane pod Java, ale ogromna przewaga wskazanych rozwiązań jest heterogeniczna)
- Kameralne grupy - szkolenia technologiczne prowadzimy w grupach liczących do 8 osób. Pozwala to na indywidualne podejście oraz aktywizację każdego uczestnika
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Konkretne 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, Portablity, 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
-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, EJB i RMI-IIOP
      -Inne rozwiązania na rynku
   e. Serwery aplikacji i kryteria wyboru
7. Architektura warstwy integracji i zasobów
   a. Technologie utrwalania danych:
      -Bazy relacyjne, hierarchiczne, NoSQL
   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, ActiveMQ, Kafka
      -Modele komunikacji w różnych implementacjach
      -Co uwzględnić przy wyborze message brokera?
   c. Systemy legacy
   d. Screen Scrapping
   e. Technologie scaffolding'owe

8. Wzorce projektowe a architektura
   a. Jak wzorce projektowe wpływają na architekturę
   b. Wybrane wzorce Core J2EE warstw prezentacji, biznesowej i integracji
   c. Wybrane wzorce GOF:
      -Factory Method, Abstract Factory,
      -Strategy, Proxy, Adapter,
      -Façade, Mediator, Observer
9. Modelowanie architektury w UML
   a. Diagram komponentów
   b. Diagram wdrożenia
10. Zaawansowane aspekty modelowania architektury w UML
    a. Instancyjne diagramy wdrożenia
    b. Niskopoziomowe diagramy wdrożenia
    c. Szablony architektoniczne
    d. Model wdrożenia na diagramach wdrożenia (artefakty)
    e. Diagram pakietów
11. 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
12. Bezpieczeństwo
    a. Zarządzanie bezpieczeństwem
    b. Mechanizmy bezpieczeństwa
    c. Serwery SSO (Single Sign On)
    d. Podstawowe rodzaje ataków
13. Transakcje
    a. ACID i BASE a CAP Theorem
    b. Wpływ transakcji na system
    c. Poziomy izolacji i skutki uboczne
    d. Konflikty zapisu danych
    e. Transakcje rozproszone
       -Transakcja lokalna a rozproszona
       -X/Open (TX, XA)
       -2PC a wydajność
       -JTA i alternatywy
       -Microservices bez 2PC (Saga i inne)
    f. Transakcje kompensacyjne
    g. Transakcje w WebService
    h. Kontrola obciążenia systemu transakcjami
14. Weryfikacja i ocena architektury
    a. Po co weryfikować?
    b. Zespół weryfikujący
    c. Techniki weryfikacji i oceny
    d. Proces weryfikacji

e. Raport z weryfikacji