sages

Course code: **J/DDD**

Course title: **Domain Driven Design (DDD) in the process of application development illustrated by the example of JEE solutions**

Days: 3

Description:

Course intended for:

The training is intended for programmers, designers and architects of JEE systems, wishing to get familiar with the methods of design and implementation of JEE application solutions using the DDD approach.

Course objective:

The training objective is to get the participants familiar with the alternative mode of development of JEE applications in relation to the most popular approach (anemic data model)

Key issues:

- Architecture and layers of JEE applications
- What is DDD and is this approach really worth using
- DDD and anemic data model – advantages and disadvantages of both
- Project implementation methodologies and their impact on the final version of the system
- Persistence of the data model and the potential problem with ValueObject mapping in the database

Course strengths:

The training is not focused exclusively on presentation of DDD-related concepts. One of the most significant components is presentation of DDD in comparison with the most popular approach to data model implementation, of the so-called anemic data model. Moreover, an increasing number of projects is being implemented in accordance with the agile approach – therefore, it is reasonable to provide some information on consequences of selection of a

given project implementation methodology and the actual architecture, developed in the process (taking into account the so-called "heavy methodologies" as well). In general, there is no single perfect solution. Participation in the training, however, will allow the participants to view the design and building of JEE solutions from an entirely different perspective and to find out when and where a given solution is optimum and when different approaches are worth considering. An exemplary application will be built on the basis of JSF 2, EJB 3 and JPA (Hibernate). On request, the training can be conducted on the basis of a different set of methodologies.

Requirements:

The training participants are required to be familiar with Java programming (to be learned at the J/JP course), to know the basic issues associated with object-oriented programming (to be learned at the UML/BASE course) and the basic concepts associated with JEE application programming (to be learned at J/EE6 course), in particular, to be familiar with JSF 2, EJB 3 and JPA (Hibernate).

Course parameters:

3*8 hours (3*7 net hours) of lectures and workshops.

Group size: no more than 8-10 participants.

Course curriculum:

1. JEE application architecture – a review of the most significant concepts and assumptions
    - Layered JEE applications – different perspectives
        - Three layer architecture (client, application, database)
        - Layers in JEE applications – application architecture (data model, presentation layer, logic layer, database access layer)
    - Anemic data model
    - Alternative for the anemic data model – Domain Driven Design
    - Object orientation vs. data model
        - Low Coupling
        - Cohesion

- Project implementation methodology and its impact on the solution building process
    - Heavy methodologies, e.g. Prince2 and the „Waterfall" approach
    - The so-called agile methodologies

2. Which model is dominant in practice?
    - Strengths and weaknesses of the anemic data model
    - Strengths and weaknesses of DDD
    - Selection of the data model and key assumptions concerning the architecture in practice
        - Are only technical issues significant?
        - What should be perceived as important?

3. DDD - Introduction
    - A different way of thinking, technologies are of secondary significance
    - Key terms
        - Ubiquitous Language
        - Entities
        - Value Objects
        - Aggregates
        - Repositories
        - Domain Services
        - Factories
        - Bounded Context
    - Application development in the DDD model
        - Application development phase

- Differences in approach to the modeling phase, solution building depending on the architectural assumptions made and the project methodology

- Application testing phase
  - DDD and Test Driven Design (TDD)
  - DDD and Behavior Driven Design (BDD)
  - Testing phase in the context of the selected solution architecture and project implementation methodology
- Maintenance and development phase
  - Advantages of using DDD from the BEGINNING of the project
    - Dependence of the maintenance and development phase on the previous phases
    - DDD and testability
    - DDD and expandability (e.g. Open-Close Principle)
    - Compliance of implementation with the expectations of the client

4. Exemplary application implementation according to DDD approach
   - Data model implementation using Entities, Value Objects and Aggregates
     - Difference between Entity and ValueObject
     - Mapping of Entity and ValueObject at the database level
     - Use of Aggregates
   - Repositories in the context of communication with the database
     - JPA and repositories
     - Dependency Injection
   - Implementation of the logic layer

        - The place and role of the services layer in the context of the fact that logic is implemented at the level of business/ domain objects (e.g. Application/Domain/Infrastructure Services)
        - Statelessness of services
    - Interface layer implementation and putting the project together
5. DDD – support through available design patterns
    - Builder
    - State Machine
    - Chain of Responsibility
    - Supple Design

Summary