Domain-Driven Design Essentials Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Domain-Driven Design (DDD) Eğitimi, karmaşık iş problemlerini analiz etmek, doğru domain modelleri oluşturmak ve bu modelleri sürdürülebilir yazılım çözümlerine dönüştürmek isteyen teknik ekipler için hazırlanmış, beş günlük kapsamlı bir online eğitim programıdır.

Domain-Driven Design, yalnızca kod geliştirme sırasında kullanılan bir teknik değil, yazılım ekipleri ile iş alanı uzmanlarının ortak bir anlayış geliştirmesini sağlayan bir tasarım ve problem çözme yaklaşımıdır. Yazılım modellerinin iş süreçleri ve kavramlarıyla uyumlu şekilde oluşturulmasını esas alan DDD, karmaşıklığın yönetilmesine ve değişen gereksinimlere uyum sağlayabilen sistemlerin geliştirilmesine yardımcı olur.

Eğitim kapsamında katılımcılar; Problem Space, Solution Space, Subdomain, Ubiquitous Language, Bounded Context ve Context Mapping gibi stratejik DDD kavramlarını inceleyeceklerdir. Ardından Entity, Value Object, Aggregate, Domain Service, Domain Event, Factory ve Repository gibi taktiksel tasarım bileşenlerinin yazılım geliştirme süreçlerinde nasıl kullanılacağını uygulamalı olarak öğreneceklerdir.

Program, gerçek bir iş probleminin analiz edilerek domain modelinin oluşturulduğu ve bu modelin C# kullanılarak çalışan bir yazılım yapısına dönüştürüldüğü uygulamalı kodlama atölyesiyle tamamlanacaktır.

Domain-Driven Design Kurumlara Ne Kazandırır?

Domain-Driven Design, teknik çözüm geliştirmeden önce iş probleminin doğru anlaşılmasını ve modellenmesini esas alır. İş kavramlarının doğru biçimde yazılım modellerine aktarılması, geliştiriciler, domain uzmanları ve iş paydaşları arasında daha etkili iletişim kurulmasını sağlar. Böylece mimari kararlar, ortak bir iş anlayışı üzerinden şekillendirilebilir.

DDD, özellikle karmaşık iş kuralları, birden fazla subdomain ve uzun geliştirme yaşam döngüsüne sahip kurumsal uygulamalarda önemli avantajlar sunar. Bununla birlikte her projede tüm DDD kalıplarının uygulanması zorunlu değildir. Kullanılacak yöntemlerin kapsamı, domain'in karmaşıklığına ve projenin gerçek ihtiyaçlarına göre belirlenmelidir.

Domain-Driven Design ve Mikroservis Mimarisi

Domain-Driven Design ile mikroservis mimarisi birbirinden farklı kavramlardır. DDD, iş alanının anlaşılmasına ve doğru model sınırlarının oluşturulmasına odaklanırken mikroservis mimarisi, uygulamaların bağımsız olarak geliştirilebilen ve dağıtılabilen servisler halinde yapılandırılmasını esas alır.

Subdomain analizi ve Bounded Context yaklaşımı, mikroservis sınırlarının belirlenmesinde yol gösterici olabilir. Ancak her Bounded Context'in ayrı bir mikroservise dönüştürülmesi gerekmez. Bu karar verilirken ekip yapısı, veri sahipliği, entegrasyon ihtiyaçları ve operasyonel yetkinlikler birlikte

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

değerlendirilmelidir.

Prerequisites

- Katılımcıların en az orta seviyede programlama bilgisine sahip olması beklenmektedir.
- Yaygın kullanılan nesne yönelimli programlama dillerinden birinde deneyim sahibi olmak yeterlidir.
- Uygulamalı çalışmalar C# ile gerçekleştirileceği için temel C# bilgisi önerilmektedir.
- Katılımcıların bilgisayarlarında Visual Studio veya Rider, Docker ve C# 8.0 ya da eğitim öncesinde belirtilen uyumlu sürümün kurulu olması gerekmektedir.

Who Should Attend

- Orta ve ileri seviye yazılım geliştiriciler
- Backend geliştiricileri
- Yazılım ve çözüm mimarları
- Teknik liderler ve mühendislik ekip liderleri
- İş analistleri ve domain uzmanlarıyla birlikte çalışan teknik ekipler
- Mikroservis geliştiren veya kurumsal uygulamalarını modernize eden ekipler
- Karmaşık iş kurallarına sahip sistemler üzerinde çalışan yazılım profesyonelleri

Domain-Driven Design Essentials Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

Bu eğitim sonunda katılımcılar;

- Domain-Driven Design (DDD) yaklaşımını ve temel prensiplerini öğrenebilecek,
- İş problemlerini analiz ederek doğru domain modelleri oluşturabilecek,
- Problem Space, Solution Space ve Subdomain kavramlarını anlayabilecek,
- Ubiquitous Language ile iş ve yazılım ekipleri arasında ortak bir dil geliştirebilecek,
- Bounded Context ve Context Mapping yöntemlerini uygulayabilecek,
- Event Storming ile iş süreçlerini ve domain olaylarını analiz edebilecek,
- Entity, Value Object, Aggregate ve Domain Service yapılarını kullanabilecek,
- Domain Event, Factory ve Repository bileşenlerini uygulayabilecek,
- DDD prensiplerini Clean Architecture ve mikroservis tasarımında değerlendirebilecek,
- C# kullanarak sürdürülebilir ve test edilebilir domain modelleri geliştirebileceklerdir.

Outline

1. Domain-Driven Design'a Giriş

- Domain-Driven Design nedir?
- DDD yaklaşımının amacı ve temel ilkeleri
- DDD'nin çözüm sağlamayı hedeflediği yazılım geliştirme problemleri
- Domain-Driven Design hakkında yaygın yanlış anlamalar
- DDD'nin karmaşık veya zor algılanmasının nedenleri
- İyi, kötü ve etkili yazılım tasarımı yaklaşımları
- Yazılım geliştirme yaşam döngüsünde DDD'nin yeri

2. Domain Kavramını Anlamak

- İş alanı (Business Domain) nedir?
- İş problemleri ile yazılım modelleri arasındaki ilişki
- Domain bilgisinin keşfedilmesi
- Domain uzmanları ile yazılım ekipleri arasındaki iş birliği

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Domain modellemenin temel prensipleri
- Domain modelinin sürekli geliştirilmesi ve iyileştirilmesi
- Model ile kod arasındaki uyumun korunması

3. Problem Space ve Solution Space

- Problem Space kavramının anlaşılması
- Solution Space kavramının anlaşılması
- İş probleminin teknik çözümden ayrıştırılması
- İş gereksinimlerinin keşfedilmesi ve netleştirilmesi
- Teknik çözüm seçilmeden önce domain modelinin oluşturulması
- Problem Space analizinin Solution Space kararlarına yansıtılması

4. Subdomain Analizi

- Subdomain nedir?
- Karmaşık bir domain'in yönetilebilir alt alanlara ayrılması
- Core Domain kavramı
- Supporting Subdomain kavramı
- Generic Subdomain kavramı
- Subdomain'lerin iş değerine göre önceliklendirilmesi
- Organizasyonel yetkinlikler ile subdomain'ler arasındaki ilişkinin kurulması

5. Ubiquitous Language (Ortak Dil)

- Ubiquitous Language nedir?
- İş birimleri ve teknik ekipler arasında ortak bir dil geliştirilmesi
- Domain kavramlarının açık ve tutarlı biçimde tanımlanması
- Ortak dilin analiz, dokümantasyon ve kod içerisinde kullanılması
- Belirsiz veya çelişkili terimlerin belirlenmesi
- Ubiquitous Language'in domain modeliyle birlikte geliştirilmesi

6. Stratejik Domain-Driven Design

- Stratejik tasarım yaklaşımının amacı

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Büyük ve karmaşık sistemlerin daha küçük yapılara ayrılması
- Domain ve model sınırlarının belirlenmesi
- İş yetkinlikleri ile yazılım sınırlarının uyumlu hale getirilmesi
- Stratejik tasarım kararlarının yazılım mimarisine etkileri

7. Bounded Context

- Bounded Context nedir?
- Domain modelinin geçerli olduğu sınırların tanımlanması
- Aynı kavramın farklı bağlamlarda farklı anlamlar taşıması
- Bounded Context ve Subdomain arasındaki ilişki
- Doğru context sınırlarının belirlenmesinde kullanılan kriterler
- Belirsiz veya hatalı sınırların neden olduğu problemler

8. Context Mapping

- Context Map nedir?
- Bounded Context'ler arasındaki ilişkilerin modellenmesi
- Ekip ve sistem sınırlarının değerlendirilmesi
- Farklı context'ler arasında veri ve model paylaşımı
- Entegrasyon noktalarının belirlenmesi
- Context Mapping sonuçlarının mimari kararlara dönüştürülmesi

9. Event Storming

- Event Storming nedir?
- İş alanının keşfedilmesinde Event Storming yönteminin kullanılması
- Domain Event'lerin belirlenmesi
- Command, Actor ve iş kurallarının analiz edilmesi
- Domain uzmanları ile teknik ekiplerin ortak çalışması
- Event Storming çıktılarından Bounded Context sınırlarının belirlenmesi

10. Taktiksel Domain-Driven Design

- Taktiksel tasarımın amacı

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Domain modelinin kod ile temsil edilmesi
- İş kurallarının uygun katmanlarda konumlandırılması
- Rich Domain Model prensipleri
- Taktiksel tasarım bileşenleri arasındaki ilişkiler
- Anemic Domain Model yaklaşımından kaçınma

11. Entity ve Value Object

- Entity nedir?
- Kimlik (Identity) ve yaşam döngüsü kavramları
- Value Object nedir?
- Değere göre eşitlik (Equality by Value)
- Değişmezlik (Immutability) prensipleri
- Entity ve Value Object arasındaki farklar
- Doğru modelleme bileşeninin seçilmesi

12. Aggregate ve Aggregate Root

- Aggregate nedir?
- Aggregate Root kavramı
- İş kurallarının ve değişmez koşulların (Invariants) korunması
- Aggregate sınırlarının tanımlanması
- Aggregate'ler arasındaki ilişkilerin yönetilmesi
- Gereğinden büyük Aggregate yapılarından kaçınma
- Transaction ve tutarlılık sınırlarının belirlenmesi

13. Domain Service ve Application Service

- Domain Service nedir?
- Entity veya Value Object içerisinde yer almayan domain davranışlarının modellenmesi
- Application Service sorumlulukları
- Domain ve uygulama katmanı sorumluluklarının ayrılması
- İş kurallarının doğru servislerde konumlandırılması

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

14. Domain Events

- Domain Event nedir?
- Domain içerisinde gerçekleşen önemli değişikliklerin modellenmesi
- Domain Event oluşturma ve yayınlama süreçleri
- Bileşenler arasındaki gereksiz bağımlılıkların azaltılması
- Domain Event'lerin iş süreçleriyle ilişkilendirilmesi

15. Factories, Repositories ve Modules

- Factory tasarım kalıbının amacı
- Karmaşık nesne oluşturma süreçlerinin yönetilmesi
- Repository nedir?
- Domain modelinin veri erişim mekanizmalarından ayrıştırılması
- Repository sınırlarının belirlenmesi
- Domain modelinin Modules kullanılarak düzenlenmesi

16. Domain-Driven Design ve Clean Architecture

- Clean Architecture yaklaşımına giriş
- Domain katmanının yazılım mimarisi içerisindeki konumu
- Bağımlılıkların Core Domain yönünde düzenlenmesi
- Domain, Application ve Infrastructure sorumluluklarının ayrılması
- Teknolojiden bağımsız domain modellerinin geliştirilmesi
- Bakımı kolay ve test edilebilir yazılım sistemlerinin oluşturulması

17. Uygulamalı Kodlama Atölyesi

- Gerçek bir iş problemi üzerinden domain keşfi
- Problem Space ve Solution Space analizi
- Subdomain'lerin ve Core Domain'in belirlenmesi
- Ubiquitous Language oluşturulması
- Bounded Context sınırlarının tanımlanması
- Stratejik ve taktiksel DDD prensiplerinin uygulanması

Domain-Driven Design Essentials Eğitimi

Süre: **5 Gün**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Domain modelinin C# koduna dönüştürülmesi
- Entity, Value Object ve Aggregate yapılarının geliştirilmesi
- Repository ve Domain Service bileşenlerinin uygulanması
- Canlı kodlama ve geliştirilen çözümün değerlendirilmesi