

Domain Driven Design (DDD) Eğitimi

Süre: 5 Gün

Açıklama

Domain Driven Design (DDD) Eğitimi, karmaşık iş problemlerini doğru şekilde analiz etmek, domain modelleri oluşturmak ve bu modelleri sürdürülebilir yazılım çözümlerine dönüştürmek isteyen teknik ekipler için hazırlanmıştır. Beş günlük online eğitimde Domain-Driven Design yaklaşımının stratejik ve taktiksel tasarım bileşenleri, gerçek dünya senaryoları ve canlı kodlama çalışmalarıyla ele alınır.

Domain-Driven Design yalnızca bir kodlama tekniği değildir. DDD; yazılım ekipleri ile iş birimleri arasında ortak bir dil oluşturmayı, karmaşık iş alanlarını yönetilebilir parçalara ayırmayı ve yazılım mimarisini iş gereksinimleriyle uyumlu hale getirmeyi amaçlayan bir problem çözme ve yazılım tasarım yaklaşımıdır.

Katılımcılar; problem space ve solution space ayrımından subdomain analizine, Ubiquitous Language kullanımından Bounded Context ve Context Mapping çalışmalarına kadar stratejik DDD kavramlarını öğrenir. Entity, Value Object, Aggregate, Domain Service, Domain Event, Factory ve Repository gibi taktiksel tasarım bileşenleri ise uygulamalı örneklerle koda aktarılır.

Bu Eğitimde Neler Öğreneceksiniz?

- Domain-Driven Design yaklaşımını ve temel amaçlarını açıklamayı
- Karmaşık iş problemlerini domain odaklı şekilde analiz etmeyi
- Problem space ve solution space arasındaki farkı belirlemeyi
- Core, Supporting ve Generic Subdomain yapılarını ayırt etmeyi
- Ubiquitous Language oluşturarak teknik ve iş ekipleri arasında ortak dil kurmayı
- Bounded Context sınırlarını belirlemeyi
- Context Mapping ile farklı bağlamlar arasındaki ilişkileri modellemeyi
- Event Storming yaklaşımını domain keşfi amacıyla kullanmayı
- DDD'nin taktiksel tasarım bileşenlerini doğru şekilde uygulamayı
- Oluşturulan domain modelini C# ile çalışan koda dönüştürmeyi

Kimler Katılmalı?

- Orta ve ileri düzey yazılım geliştiriciler
- Backend geliştiriciler
- Yazılım ve çözüm mimarları

- Teknik liderler ve takım liderleri
- İş analistleri ile birlikte domain modelleme süreçlerinde çalışan teknik ekipler
- Mikroservis mimarisi geliştiren veya uygulama modernizasyonu yürüten ekipler
- Karmaşık iş kurallarına sahip kurumsal uygulamalar geliştiren profesyoneller

Eğitim İçeriği

1. Domain-Driven Design'a Giriş

- Domain-Driven Design nedir?
- DDD yaklaşımının amacı ve temel prensipleri
- DDD'nin çözmeye çalıştığı yazılım geliştirme problemleri
- DDD ile ilgili yaygın yanlış anlamalar
- DDD'den çekinmek: Karmaşıklık ve öğrenme bariyerleri
- İyi, kötü ve etkili yazılım tasarımı
- DDD'nin yazılım geliştirme yaşam döngüsündeki yeri

2. Domain ve Domain Modelleme

- Domain nedir?
- İş problemi ile yazılım modeli arasındaki ilişki
- Domain bilgisinin keşfedilmesi
- Domain uzmanları ve yazılım ekipleri arasındaki iş birliği
- Domain Modeling yaklaşımı
- Domain modelinin sürekli geliştirilmesi
- Model ile kod arasındaki ilişkinin korunması

3. Problem Space ve Solution Space

- Problem Space nedir?
- Solution Space nedir?
- İş problemini teknik çözümden ayırma
- İş ihtiyaçlarının doğru anlaşılması
- Çözüm üretmeden önce domainin modellenmesi
- Problem ve çözüm alanları arasındaki ilişkinin kurulması

4. Subdomain Analizi

- Subdomain nedir?
- Karmaşık domainlerin alt alanlara ayrılması
- Core Domain
- Supporting Subdomain
- Generic Subdomain

- Subdomainlerin iş değeri açısından önceliklendirilmesi
- Kurumsal yetkinlikler ile subdomainler arasındaki ilişki

5. Ubiquitous Language

- Ubiquitous Language nedir?
- İş birimleri ve teknik ekipler arasında ortak dil oluşturma
- Domain kavramlarının açık ve tutarlı şekilde tanımlanması
- Ortak dilin analiz, dokümantasyon ve kod içinde kullanılması
- Belirsiz ve çelişkili kavramların tespit edilmesi
- Ubiquitous Language'in domain modeliyle birlikte geliştirilmesi

6. Stratejik Domain-Driven Design

- Stratejik tasarımın amacı
- Büyük ve karmaşık sistemlerin ayrıştırılması
- Domain sınırlarının belirlenmesi
- İş yetenekleri ile yazılım sınırlarının ilişkilendirilmesi
- Stratejik kararların yazılım mimarisine etkisi

7. Bounded Context

- Bounded Context nedir?
- Model sınırlarının belirlenmesi
- Aynı kavramın farklı bağlamlardaki anlamları
- Bounded Context ile subdomain arasındaki ilişki
- Bağlam sınırlarının belirlenmesinde kullanılan ölçütler
- Yanlış veya belirsiz bağlam sınırlarının oluşturduğu problemler

8. Context Mapping

- Context Map nedir?
- Bounded Context'ler arasındaki ilişkilerin modellenmesi
- Ekip ve sistem sınırlarının değerlendirilmesi
- Bağlamlar arası veri ve model paylaşımı
- Entegrasyon noktalarının belirlenmesi
- Context Mapping sonuçlarının mimari kararlara aktarılması

9. Event Storming

- Event Storming yaklaşımı nedir?
- Domain keşfi ve iş süreçlerinin görünür hale getirilmesi
- Domain Event'lerin belirlenmesi
- Komutların, aktörlerin ve iş kurallarının incelenmesi

- İş birimleri ile teknik ekiplerin birlikte modelleme yapması
- Event Storming çıktılarının Bounded Context çalışmalarında kullanılması

10. Taktiksel Domain-Driven Design

- Taktiksel tasarımın amacı
- Domain modelinin kod seviyesinde ifade edilmesi
- İş kurallarının doğru katmanda konumlandırılması
- Zengin domain modeli yaklaşımı
- Taktiksel tasarım bileşenleri arasındaki ilişkiler

11. Entity ve Value Object

- Entity nedir?
- Kimlik ve yaşam döngüsü kavramları
- Value Object nedir?
- Değer üzerinden eşitlik
- Değişmezlik yaklaşımı
- Entity ve Value Object arasındaki farklar
- Doğru modelleme bileşeninin seçilmesi

12. Aggregate ve Aggregate Root

- Aggregate nedir?
- Aggregate Root kavramı
- İş kurallarının ve değişmezlerin korunması
- Aggregate sınırlarının belirlenmesi
- Aggregate'ler arasındaki ilişkilerin yönetilmesi
- Gereğinden büyük Aggregate tasarımlarından kaçınma
- İşlem ve tutarlılık sınırları

13. Domain Service ve Application Service

- Domain Service nedir?
- Bir Entity veya Value Object'e ait olmayan domain davranışları
- Application Service'in sorumlulukları
- Domain ve uygulama katmanlarının ayrılması
- İş kurallarının servisler içinde doğru şekilde konumlandırılması

14. Domain Event

- Domain Event nedir?
- Domain içinde gerçekleşen önemli değişikliklerin modellenmesi
- Domain Event'lerin oluşturulması ve yayımlanması

- Bileşenler arasındaki bağımlılığın azaltılması
- Domain Event'lerin iş süreçleriyle ilişkilendirilmesi

15. Factory, Repository ve Module

- Factory yaklaşımının kullanım amacı
- Karmaşık nesne oluşturma süreçlerinin yönetilmesi
- Repository nedir?
- Domain modeli ile veri erişim mekanizmasının ayrılması
- Repository sınırlarının belirlenmesi
- Module kullanımı ve modelin düzenlenmesi

16. DDD ve Clean Architecture

- Clean Architecture yaklaşımı
- Domain katmanının mimarideki konumu
- Bağımlılıkların iç katmanlara doğru yönlendirilmesi
- Domain, uygulama ve altyapı sorumluluklarının ayrılması
- Teknoloji bağımsız domain modeli geliştirme
- Test edilebilir ve sürdürülebilir yazılım tasarımı

17. Coding Workshop

- Gerçek dünya problemi üzerinden domain keşfi
- Problem Space ve Solution Space analizi
- Subdomainlerin ve Core Domain'in belirlenmesi
- Ubiquitous Language oluşturulması
- Bounded Context sınırlarının modellenmesi
- Stratejik ve taktiksel DDD kavramlarının uygulanması
- Domain modelinin C# ile koda aktarılması
- Entity, Value Object ve Aggregate tasarımlarının uygulanması
- Repository ve Domain Service bileşenlerinin geliştirilmesi
- Canlı kodlama ve çözüm değerlendirme çalışması

Ön Koşullar

- Katılımcıların en az orta düzey programlama bilgisine sahip olması beklenmektedir.
- Yaygın olarak kullanılan bir nesne yönelimli programlama dilini bilmek yeterlidir.
- Uygulamalı çalışmalar C# ile gerçekleştirileceği için temel C# bilgisi önerilir.
- Katılımcıların bilgisayarlarında Visual Studio veya Rider, Docker ve C# 8.0 ya da eğitim için bildirilen uyumlu sürüm kurulu olmalıdır.