

DOMAIN DRIVEN DESIGN

Günümüz yazılım dünyasında iş süreçleri her geçen gün daha da karmaşılaşıyor. Bu karmaşıklık, yazılım projelerinin başarısını doğrudan etkileyen en önemli faktörlerden biri haline geldi. Peki, bu karmaşıklığı yönetmek, iş gereksinimlerini doğru anlamak ve esnek, sürdürülebilir yazılımlar geliştirmek mümkün mü? Cevabımız evet: Domain-Driven Design (DDD) ile!

Bu eğitim, yazılım geliştirme süreçlerinde karşılaşılan karmaşıklığı aşmak ve iş domain'ine odaklanarak daha anlamlı, anlaşılır ve bakımı kolay yazılımlar üretmek isteyen herkes için tasarlandı. DDD, sadece bir teknik veya desen seti olmaktan öte, iş ve yazılım ekipleri arasında ortak bir dil oluşturmayı, iş süreçlerinin temelini oluşturan domain bilgisini yazılım koduna doğru bir şekilde yansıtmayı hedefler.

DOMAIN DRIVEN DESIGN

KURUMSAL ÖZEL EĞİTİM

Bu konuyu istediğiniz gibi özelleştirebilirsiniz. Kullandığınız teknolojiye özel uygulamalar ile anlatılmasını, projenize uygun içerik haline getirilmesini veya belirli bir departmanınızın ihtiyaçlarına göre şekillendirilmesini talep edebilirsiniz.

DOMAIN DRIVEN DESIGN EĞİTİMİ

- **İş Ortaklarınızla Ortak Dil Geliştirme:** Ubiquitous Language (Ortak Dil) kavramını öğrenerek iş birimleri ile daha etkili iletişim kuracak, yanlış anlaşılmaların önüne geçeceksiniz.
- **Karmaşık Domain'leri Modelleme:** Agregalar, Varlıklar (Entities), Değer Nesneleri (Value Objects), Domain Servisleri ve Domain Olayları gibi temel DDD yapı taşlarını tanıyarak karmaşık iş kurallarını kodunuza doğru bir şekilde yansıtacaksınız.
- **Esnek ve Ölçeklenebilir Mimariler Oluşturma:** Katmanlı Mimari, Hexagonal Mimari (Port & Adapters), Command-Query Responsibility Segregation (CQRS) ve Event Sourcing gibi mimari desenlerin DDD ile nasıl entegre edileceğini keşfedeceksiniz.
- **Mikroservis Mimarilerinde DDD:** Sınırlanmış Bağlamlar (Bounded Contexts) ve Bağlam Haritaları (Context Maps) ile mikroservis tabanlı sistemlerde iş domain'lerini nasıl ayıracağınızı ve entegre edeceğinizi öğreneceksiniz.
- **Temiz Kod ve Bakım Kolaylığı:** Domain bilgisinin kodunuza doğru bir şekilde yansmasıyla daha anlaşılır, test edilebilir ve bakımı kolay bir kod tabanı oluşturacaksınız.
- **Stratejik ve Taktiksel Tasarım:** DDD'nin hem büyük resme odaklanan stratejik tasarım prensiplerini hem de günlük kodlamada kullanılan taktiksel tasarım desenlerini derinlemesine anlayacaksınız.





Modül 1: DDD ye Giriş ve Temel Yapı Taşları

DDD nin Amacı, Prensipleri ve Faydaları:

- DDD nin ne olduğu ve neden kullanılması gerektiği.
- İş odaklı yazılım geliştirme prensipleri.
- DDD nin proje başarısına etkileri ve sağladığı faydalar (karmaşıklık yönetimi, esneklik, sürdürülebilirlik).

Karmaşık İş Alanlarını Yönetme Yaklaşımı:

- Büyük ve karmaşık etki alanlarının nasıl parçalara ayrılacağı.
- Etki alanı bilgisinin yazılım tasarımına nasıl yön verdiği.

Anlamanın ve Niyetin Kodda İfade Edilmesi:

- İş gereksinimlerinin kodlama seviyesine nasıl yansıtılacağı.
- Anlaşılır ve ifade gücü yüksek kod yazma pratikleri.

Etki Alanı (Domain) ve Alt Alanlar (Subdomains):

- İş probleminin tanımlanması ve ana etki alanının belirlenmesi.
- Alt alanlara ayırma stratejileri ve neden gerektiği.

Çekirdek (Core), Destekleyici (Supporting) ve Genel (Generic) Alt Alanların Tanımlanması:

- İşin kalbini oluşturan Çekirdek alanların belirlenmesi.
- Destekleyici ve Genel alanların ayırt edilmesi ve farklı yaklaşımlar.

Modül 2: Bağlam, Dil ve Temel Model Elemanları

Sınırlandırılmış Bağlam (Bounded Context):

- Modelin ve Ortak Dilin geçerli olduğu açık mantıksal sınırların belirlenmesi.
- Farklı bağlamlardaki aynı terimlerin farklı anlamlarını yönetme.
- Model çatışmalarını önleme ve çözme stratejileri.

Ortak Dil (Ubiquitous Language):

- İş uzmanları ve geliştiriciler arasında paylaşılan tutarlı ve kesin dilin önemi.
- Bu dilin doğrudan modelde ve kodda nasıl kullanıldığına dair örnekler.

Varlıklar (Entities) ve Değer Nesneleri (Value Objects):

- Nesnelerin kimliğe sahip olup olmamasına göre ayrımı.
- Varlıkların yaşam döngüsü yönetimi.
- Değer Nesnelerinin değişmezliği ve eşitlik karşılaştırmaları.

Modül 3: Topluluklar, Depolar ve Servis Katmanları

Topluluklar (Aggregates) ve Kökleri (Roots):

- Tutarlılık (invariants) sınırlarının belirlenmesi ve Topluluk Kavramı.
- Topluluk Kökü aracılığıyla erişim ve operasyonların yönetimi.
- Topluluk içindeki tutarlılık kurallarının (invariants) nasıl zorlandığı.

Depolar (Repositories):

- Topluluk Kökleri ne erişim sağlayan soyutlama katmanı.
- Veri kalıcılığı mekanizmalarını (veritabanı, ORM) etki alanından gizleme.
- Koleksiyon benzeri arayüzler aracılığıyla veri erişimi.

Etki Alanı Servisleri (Domain Services):

- Bir veya birden fazla Varlık veya Topluluğu ilgilendiren iş mantığı.
- Varlıkların veya Değer Nesnelerinin sorumluluğuna doğal olarak uymayan operasyonlar.

Uygulama Servisleri (Application Services):

- Kullanıcı arayüzü (UI) veya dış sistemlerle etkileşim için arayüz.
- Etki Alanı Servisleri ve Depoları kullanarak iş akışlarını (use cases) koordine etme.
- İşlemleri (transactions) başlatma ve yönetme rolü.

Katmanlı Mimari ve DDD:

- Sunum (Presentation), Uygulama (Application), Etki Alanı (Domain) ve Altyapı (Infrastructure) katmanlarının rolleri ve sorumlulukları.
- Bağımlılıkların doğru yönde (içe doğru) akmasını sağlama.

Modül 4: Mimari Desenler, CQRS ve Etki Alanı Olayları

Soğan (Onion) ve Temiz (Clean) Mimariler:

- Bağımlılıkların her zaman içe doğru, Etki Alanı modeline doğru akmasını sağlayan prensipler.
- Bu mimarilerin test edilebilirliği ve esnekliği nasıl artırdığı.

Komut Sorgu Sorumluluk Ayrımı (CQRS) Temelleri:

- Okuma (Query) ve Yazma (Command) operasyonlarının neden ve nasıl ayrıştırıldığı.
- Farklı operasyonlar için farklı model ve altyapı kullanma imkanı.
- CQRS nin performans ve ölçeklenebilirlik potansiyeli.

Etki Alanı Olayları (Domain Events):

- Etki Alanı içindeki önemli durum değişikliklerini bir "olay" olarak modelleme.
- Gevşek bağılılığı (loose coupling) teşvik etme mekanizması.
- Etki Alanı içinde veya farklı bağlamlarda yan etkileri tetiklemek için kullanımı.

Fabrikalar (Factories):

- Karmaşık Varlık veya Topluluk nesnelerinin yaratılma mantığını kapsülleme.
- Oluşturulan nesnelerin başlangıç durumlarının geçerliliğini garanti etme.

Spesifikasyonlar (Specifications):

- İş kurallarını veya filtreleme/seçim mantığını tekrar kullanılabilir nesneler olarak ifade etme.
- Depo sorgularında veya doğrulama süreçlerinde kullanımı.

Modül 5: Kalıcılık, Doğrulama ve DDD Uygulaması

Veri Kalıcılığı ve ORM (Entity Framework Core):

- .NET de popüler ORM olan Entity Framework Core ile DDD desenlerinin nasıl eşleştirileceği.
- Persistence Ignorance (Kalıcılıktan Habersiz Olma) hedefine ulaşma yolları.
- Varlıkların, Değer Nesnelerinin ve Toplulukların EF Core ile doğru şekilde eşlenmesi (mapping).

İşlem Yönetimi (Transaction Management):

- Topluluklar (Aggregates) genelinde veya birden fazla Topluluğu etkileyen operasyonlarda atomik işlem sağlama.
- Uygulama Servisleri katmanında işlem yönetimi prensipleri.

Etki Alanı İçi Doğrulama (In-Domain Validation):

- İş kurallarının Varlık ve Değer Nesnelerinin metotları ve yapılandırıcıları içinde uygulanması.
- Nesnelerin her zaman geçerli (valid) bir durumda olmasını sağlayan yaklaşımlar.

Depo Uygulama Desenleri:

- Jenerik Depoların avantaj ve dezavantajları vs. Topluluğa Özel Depoların kullanımı.
- Sorgu desenleri (örn. Linq kullanımı, Spesifikasyonlar ile sorgu oluşturma).
- Veritabanı bağlamının (DbContext) ve Unit of Work deseninin yönetimi.

Birim Testleri (Unit Tests) ve DDD:

- Etki Alanı Modelinin (Varlıklar, Değer Nesneleri, Topluluklar, Etki Alanı Servisleri) birim testlerinin yazılması.
- Uygulama Servislerinin iş akışı mantığının test edilmesi.
- Bağımlılıkların (Depolar, dış servisler) sahte (mock) veya taklit (stub) edilmesi teknikleri.



Modül 6: Sınırlar Arası Entegrasyon ve İleri Konulara Giriş

Bağlam Haritalama (Context Mapping):

- Farklı Sınırlandırılmış Bağlamlar arasındaki entegrasyon ilişkilerini (Customer/Supplier, Conformist, Partnership, Shared Kernel vb.) modelleme.
- Bu haritaların ekip organizasyonu ve entegrasyon stratejilerine etkisi.

Anti-Kirillik Katmanı (Anti-Corruption Layer - ACL):

- Dış veya "eski" sistemlerle entegrasyon sırasında kendi Etki Alanı modelini koruma ihtiyacı.
- Gelen ve giden veriler arasında çeviri (translation) rolü.

İleri Seviye CQRS Uygulamalarına Giriş:

- Ayrı okuma ve yazma veritabanı stratejileri (CQRSnin tam gücü).
- Materyalize görünümünün (Materialized Views) kullanımı.
- Dağıtık CQRS senaryolarına genel bakış.

Olay Kaynaklama (Event Sourcing) Giriş:

- Durumu, gerçekleşen olayların bir dizisi olarak saklama temel fikri.
- Olayların yeniden oynatılarak geçmişteki herhangi bir duruma ulaşma yeteneği.

Nihai Tutarlılık (Eventual Consistency) Giriş:

- Dağıtık sistemlerde (örn. farklı Bağlamlar arası) veri tutarlılığı modelleri.
- Etki Alanı Olayları ve mesajlaşma yoluyla nihai tutarlılığın sağlanması kavramı.

EĞİTİM SÜRESİ

- 3 Gün
- Ders Süresi: 50 dakika
- Eğitim Saati: 10:00 - 17:00

Eğitim formatında eğitimler 50 dakika + 10 dakika moladır. 12:00-13:00 saatleri arasında 1 saat yemek arasındaki verilir. Günde toplam 6 saat eğitim verilir. 3 günlük formatta 18 saat eğitim verilmektedir.

Eğitimler uzaktan eğitim formatında tasarlanmıştır. Her eğitim için teams linkleri gönderilir. Katılımcılar bu linklere girerek eğitimlere katılırlar. Ayrıca farklı remote çalışma araçları da eğitmen tarafından tüm katılımlara sunulur. Katılımcılar bu araçları kullanarak eğitimlere katılırlar.

Eğitim içeriğinde github ve codespace kullanılır. Katılımcılar bu platformlar üzerinden örnek projeler oluşturur ve eğitmenle birlikte eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Katılımcılar bu araçlarla eğitimlerde sorulan sorulara ve taleplere uygun içeriğe cevap verir. Eğitim yapay zeka destekli kendi kendine öğrenme formasyonu ile tasarlanmıştır. Katılımcılar eğitim boyunca kendi kendine öğrenme formasyonu ile eğitimlere katılırlar. Bu eğitim formatı sayesinde tüm katılımcılar gelecek tüm yaşamlarında kendilerini güncellemeye devam edebilecekler ve her türlü sorunun karşısında çözüm bulabilecekleri yeteneklere sahip olacaklardır.



DOMAIN DRIVEN DESIGN

DDD, yazılımın merkezine iş alanını (domain) ve onun karmaşıklığını koyan bir yazılım geliştirme metodolojisidir. Amacı, işin uzmanlarıyla (domain expert) yazılım geliştiriciler arasında ortak bir dil (Ubiquitous Language) oluşturarak, iş mantığını ve kurallarını yazılımın temel yapısına doğru ve açık bir şekilde yansıtmaktır.

- **Stratejik Tasarım (Strategic Design):** Büyük resmi görür ve iş alanını daha yönetilebilir, bağımsız parçalara (Bounded Contexts) ayırmanıza yardımcı olur. Bu, özellikle büyük ve karmaşık sistemlerde, her bir parçanın kendi iç tutarlılığını korurken diğer parçalarla nasıl etkileşim kuracağını belirlemek için önemlidir.
- **Taktiksel Tasarım (Tactical Design):** Stratejik tasarımda belirlenen sınırlar içinde, iş mantığını kodlamak için kullanılan detaylı desenler ve bloklardır. Varlıklar (Entities), Değer Nesneleri (Value Objects), Agregalar (Aggregates), Domain Servisleri (Domain Services) ve Domain Olayları (Domain Events) gibi yapılar, iş kurallarını açık ve modüler bir şekilde ifade etmenizi sağlar.

Sıfırdan Zirveye Domain Driven Design Neden Önemlidir?

- Karmaşıklığı Yönetme: Modern iş uygulamaları giderek daha karmaşık hale geliyor. DDD, bu karmaşıklığı yönetmek için güçlü prensipler ve desenler sunar. İş domain'ini küçük, anlaşılır parçalara bölerek ve bu parçalar arasındaki ilişkileri netleştirerek, geliştiricilerin daha odaklanmış çalışmasını sağlar.
- İş ile Yazılım Arasında Köprü Kurma: En büyük sorunlardan biri, iş birimleri ile teknik ekipler arasındaki iletişim boşluğudur. DDD'nin Ubiquitous Language kavramı, herkesin aynı terimleri kullanarak konuşmasını ve iş gereksinimlerinin doğru bir şekilde anlaşılmasını sağlar. Bu, yanlış anlamaları azaltır ve yazılımın gerçekten işin ihtiyacını karşılamasına yardımcı olur.
- Esneklik ve Sürdürülebilirlik: İş gereksinimleri sürekli değişir. DDD ile geliştirilen sistemler, iş domain'indeki değişikliklere karşı daha esnek ve adapte olabilir yapıdadır. Temiz ve modüler bir domain modeli, yeni özellik eklemeyi veya mevcut özellikleri değiştirmeyi çok daha kolay hale getirir, bu da yazılımın ömrünü uzatır ve bakım maliyetlerini düşürür.



HEDEF KİTLE

- Bulut Mimarları (Cloud Architects): Karmaşık bulut çözümleri tasarlayan, altyapı stratejileri belirleyen ve teknoloji seçimleri yapan mimarlar için bu eğitim, tasarımlarını kanıtlanmış kalıplarla güçlendirerek daha dayanıklı, ölçeklenebilir ve maliyet etkin mimariler oluşturmalarına yardımcı olur.
- Yazılım Mimarları (Software Architects): Uygulama düzeyinde mimari kararlar alan ve yazılım sistemlerinin genel yapısını tasarlayan yazılım mimarları için bulut tasarım kalıpları, uygulamalarını bulutun dinamik yapısına uygun hale getirmelerini ve dağıtık sistemlerin zorluklarını aşmalarını sağlar.
- Kıdemli Yazılım Geliştiriciler: Bulut tabanlı uygulamalar geliştiren ve kodlama standartlarının ötesinde mimari düşünme becerilerini geliştirmek isteyen kıdemli geliştiriciler için bu eğitim, yazdıkları kodun daha büyük sistem içindeki rolünü anlamalarına ve daha sağlam bileşenler tasarlamalarına olanak tanır.
- DevOps ve Site Reliability Engineering (SRE) Uzmanları: Uygulama dağıtımı, operasyonları, izleme ve otomasyondan sorumlu DevOps ve SRE uzmanları için bulut tasarım kalıpları, operasyonel verimliliği artıracak, hata toleransını yükseltecek ve sistemlerin çalışma zamanı davranışlarını optimize edecek mimarileri anlamalarını ve uygulamalarını sağlar.
- Çözüm Mimarları ve Danışmanlar: Müşteri ihtiyaçlarına yönelik bulut tabanlı çözümler tasarlayan ve danışmanlık hizmeti veren profesyoneller için bu eğitim, farklı senaryolara uygun en iyi mimari yaklaşımları belirleme ve müşterilerine stratejik değer katma yetkinliği kazandırır.
- Bulut Bilişim Alanında Uzmanlaşmak İsteyen Herkes: Yukarıdaki unvanlara sahip olmasa da, temel bulut bilgisine sahip ve kariyerini bulut mimarisi ve ileri seviye bulut mühendisliği alanında şekillendirmek isteyen, problem çözmeye ve analitik düşünmeye hevesli tüm teknoloji profesyonelleri bu eğitime katılabilir.

"

Kurumsal size özel eğitimler hazırlıyoruz. Her eğitim yeni bir heyecan.

Vebende A.Ş.

Kurumsal Terzi Usulü Butik Eğitimler.

Size özel hazırlanan seminer, danışmanlık, eğitim ve hizmetlerimizle yüksek verim elde edin. Paranızı boşa harcamayın. Zaman çok değerli.

Her Eğitim Yeni Bir Heyecan

2000 yılından günümüze devam eden eğitim heyecanı. Uluslararası tecrübe, proje geliştirme deneyimleri, danışmanlıklar ve arge mühendislik deneyimlerimizi ülkemize sunmak için yeni bir konsept tasarladık. Sizi dinliyor, takımınıza uygun özel içerikler ile hazırlanmış eğitimler hazırlıyoruz. Her eğitim özel bir çalışma, içerik üretimi, uygulama örnekleri hazırlıkları, sunumlar hazırlamayı gerektiriyor. Aynı eğitimi yönetim kadrosuna farklı, teknik ve mimar ekiplerinize farklı içerikler ile hazırlıyoruz. Her eğitim yeni bir macera ve heyecan.



Bizimle İletişime Geçin

iletisim@vebende.com.tr

www.vebende.com.tr

İzmir - Türkiye

Kurumsal Eğitimler

www.vebende.com.tr





İletişime Geçin



Whatsapp 0542 5505704
0532 512 7811



iletisim@vebende.com.tr



www.vebende.com.tr



Adalet Mah. Manas Bulvarı
No:39 İç Kapı NO: 3107 Bayraklı
İZMİR



Sürekli Güncellenen Eğitimlerimizi Sitemizden
Takip Edin.

“

Kurumsal terzi usulu size özel
eğitimler ve uluslararası deneyime
sahip eğitmenler ile çalışmanın keyfi

”



Size Neler Sunuyoruz?

- Kitaptan okunan eğitimler sunmuyoruz. Sizin için özel hazırlanan içerikler ve uygulamalı eğitimler hazırlıyoruz.
- Her eğitim için sunumlar, materyaller, örnek senaryolar size özel hazırlıyoruz.
- Her eğitim için katılımcılara özel github repoları hazırlanıyor. Eğitimden sonra da katılımcılar hayat boyu kaynaklara ulaşmaya devam edebilsinler diye, **“katılımcılar ve eğitmenle birlikte yapılan tüm çalışmaları”** github repolarında saklıyoruz.



Misyonumuz

1

Ülkemizde dünyanın en güncel teknolojilerini kazandırmak. En güncel teknolojilerine hakim takımlarına katkı sağlamak.

2

Alışılmışın dışında en güncel teknolojileri deneyimli eğitmenler ve uygulamalar ile sunmak.

3

Siber güvenlik konusunda hassas çalışmalar sunarak, çağımızın büyük kabusu siber tehditler konusunda deneyimli bilgili takımların oluşmasına katkı sağlamak.