

Eski Monolithlerde Refactoring



Güven Atbakan

CTO @Moneo

[linkedin.com/in/guvenatbakan](https://www.linkedin.com/in/guvenatbakan) • twitter.com/GAtbakan



moneo

Ürün ve yazılım geliştirme şirketi

KOMMUNITY

 ZORLU

KARACA®

 www.
kitapYurdu.com

 vispera

 GENÇ SİGORTA

 transiyzi

düğün.com

 İSBAK | İSTANBUL
METROPOLİTAN
MÜHİMETŞİFİLLİĞİ

 NETLOG
LOJİSTİK

 conectohub

 TEKNASYON


nettunett.
DÜJÜL İLETİŞİM PLATFORMU


anneyesen
.com

D C E Y


akrep.com
aklını sev, cebini koru


yeni
BEN
her deneyimden perfe yapar


PENGUEN
KİTABEVI


Abilitypool
yeterlikte paylaşı

 dynorender



Hangi Projelerden Bahsediyorum?

- Uzun yıllardır aynı codebase içerisinde geliştirilen projeler
- (Genellikle) Frontend ve Backend iç içe olan projeler
- Core ekibi dağılmış veya hiç olmamış projeler
- İşin nasıl çalıştığını sadece developerin bildiği projeler
- Yük problemi çıktığında yeni sunucu eklenen projeler

Neden refactoringe ihtiyaç oluyor?

- İş geliştirme süreleri uzuyor
- İş geliştirme sonrası hatalar ortaya çıkıyor
- Business logic bilgisi azalıyor
 - ↳ Biz bu işi niye geliştirmiştik?
 - ↳ Bu butona basınca ne oluyordu? - kod takip edilebilir değil
- Kod eski kalıyor
 - ↳ İşin geliştirmesi zorlaşıyor
 - ↳ Eski kod yeni takım arkadaşları tarafından sevilmiyor
 - ↳ Yazılım ekibi, geliştirme süreçlerinden mutsuz oluyor

Refactoring kültürelidir

- Sadece kod değil, kod ile birlikte iş süreçlerinin de değişmesi gerekebilir
- Baskı altında refactoring yapmak zordur.
 - ↳ Yöneticilerin ikna edilmesi gerekir ki baskı minimuma ınsın
- Gerçekten ihtiyaç oluşmalıdır



Refactoring daha iyi
kod yazmanızı sağlar

Sıfırdan yazalım...

Refactoring, kodun ürettiği sonucu değiştirmeden, kodun yapısını ve içeriğini düzenlemeye sürecidir.

Rewrite





Durum Analizi

Mevcut Hatalar

- Mevcutta yaşanan kod hatalarının tespiti yapılmalı
- Kolay kazanım: Bugları çözerken refactoring planlamak



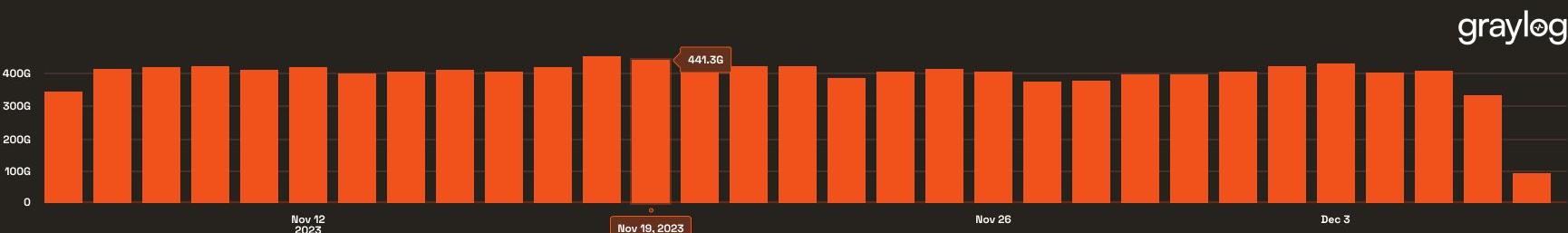
bugsnag



Logging & Observability

Ölçemediğin şeyi yönetemezsin

- En çok kullanılan sayfalar
- Kullanıcıların aldıkları hatalar (4xx)
- Yapılan requestler ve response bilgileri (Pil Dikkat!)
- Önemli gördüğünüz herşey
 - ↳ Kullanıcıların başarısız login denemeleri
 - ↳ Authorization hataları



Database

- Backuplardan emin olun
- Veritabanını analiz edin
 - ↳ Slow query log (ekmek kapısı)
 - ↳ General log
 - ↳ Error log
 - ↳ Tablo yapıları, boyutları
- Mevcut DB'yi localde çalışılabilir hale getirin (PII Dikkat!)
- Dev/Staging/Prod ortamlarının eşit olduğundan emin olun
- Kullanılmıyorsa projeye migration dahil edin

Öncelikle Mevcut Projeyi En İyi Haline Getirin

Linters & CI/CD

PHPStan

psalm

phan

php-cs-fixer

php-parallel-lint

pnp-var-dump-checker

PHP_CodeSniffer

SonarQube

Queue

- Uzun süren ve asenkron olabilecek işlemleri tespit edin
 - ↳ Email Gönderimleri
 - ↳ 3. parti servis istekleri
 - ↳ İsteği etkilemeyen db güncellemleri



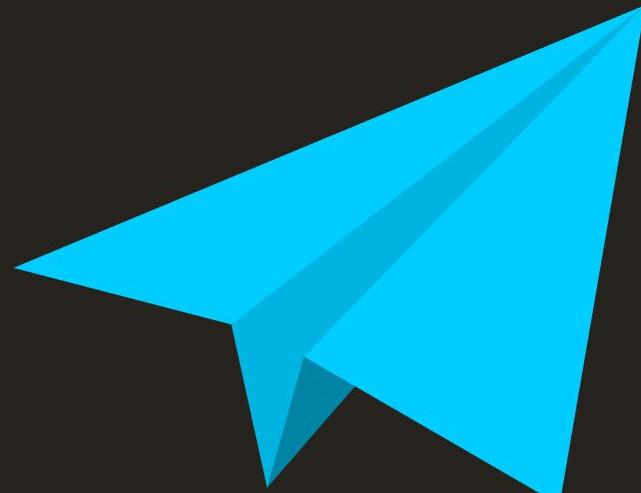
Symfony Messenger
(<https://symfony.com/doc/current/messenger.html>)



PHP Enqueue
(<https://php-enqueue.github.io/>)

Deployments

- Deployment yöntemi güncellenmeli
 - ↳ FTP?
- Hazır değiştirmişken zero downtime yapılmalı
- Kolay rollback yapılabilmeli
- Multi server deployment için hazırlanmalı



<https://deployer.org/>

FORCE



Envoyer

Testing



Microservice?

Refactoring bir methodu/özelliği baştan yazmak
değildir. Baştan yazılabılır, okunabilir,
test edilebilir hale getirmektir.



0 212 660 76 23

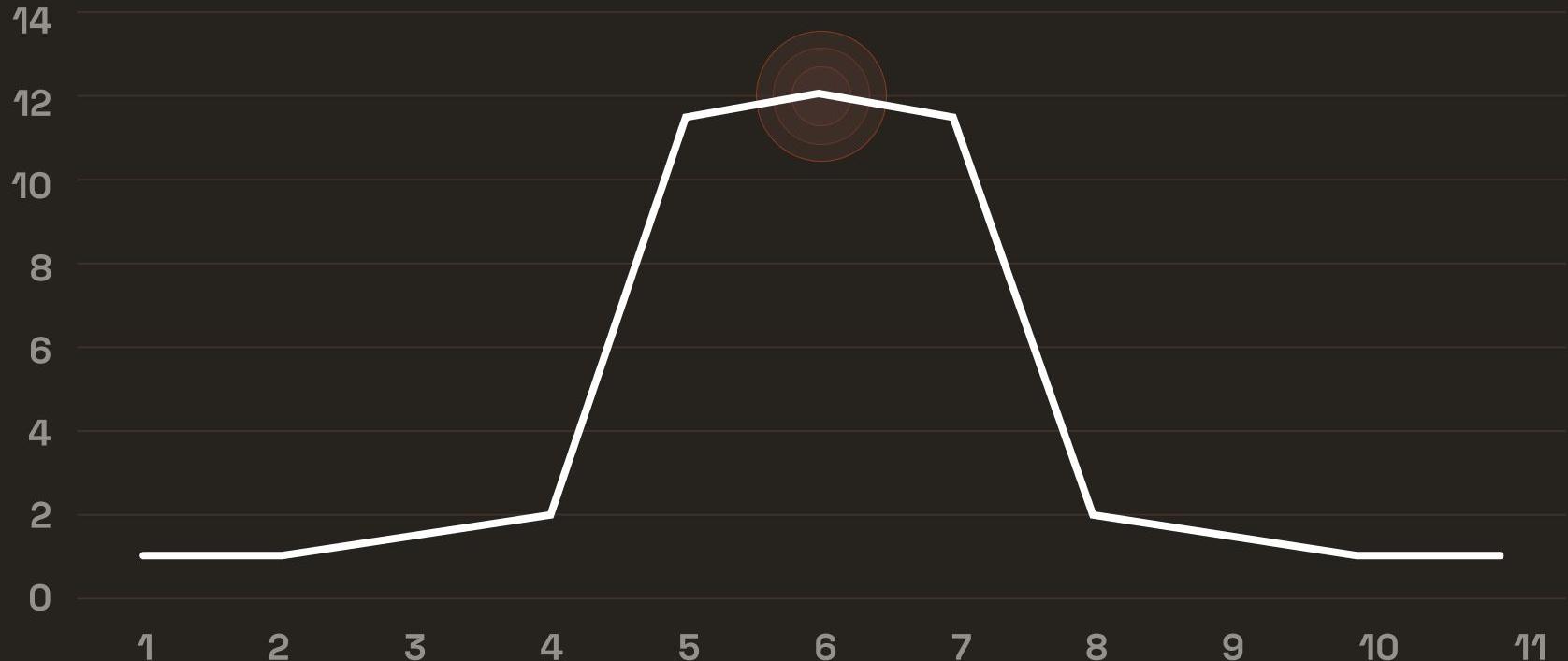
SÜREYYA HANIM'IN
MUHTEŞEM DEĞİŞİMİ...

Düzgün Kod Nedir?

- Genel kabul görmüş Design Patternleri uygulayan
- Geliştirilebilir
- Kolay okunabilir
- Testlerle desteklenmiş
- Hata takibi ve performans ölçümü yapılabilen

Design Pattern

— Experience



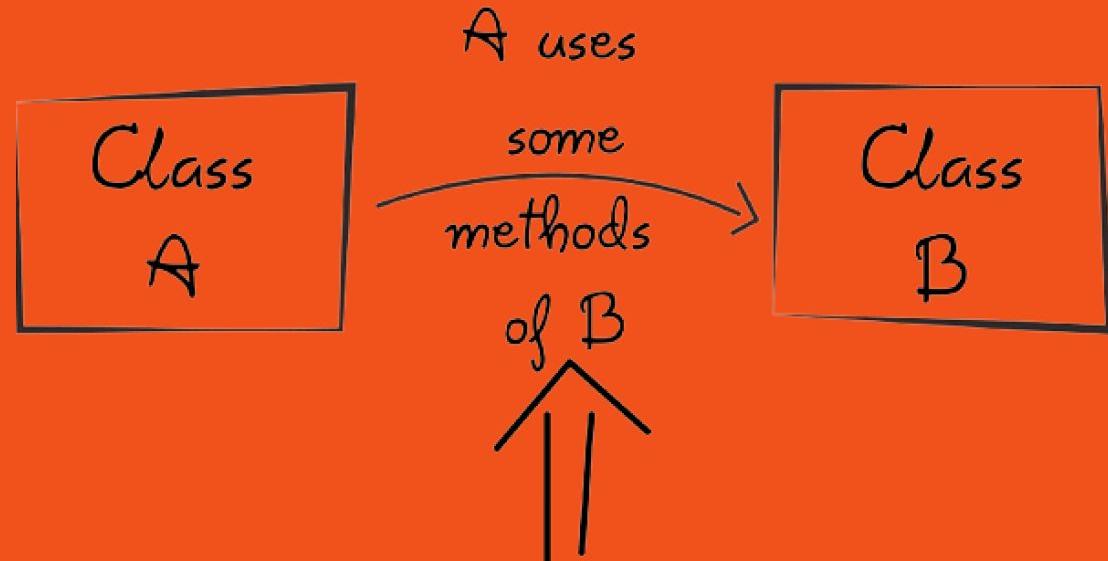
DIC & Facades

 Symfony
Pimple

 PHP-DI **7**

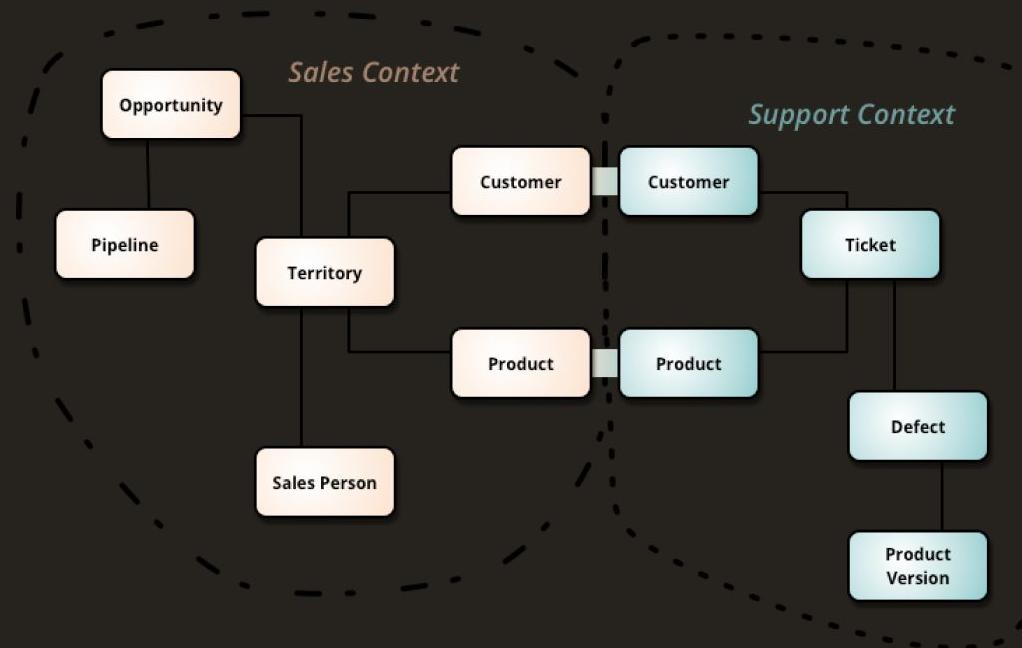
 Symfony
Dependency Injection

 Laravel



Its a dependency

Domain Driven Design



Kaynak: <https://martinfowler.com/bliki/BoundedContext.html>

Kod Tavsiyeleri

- Gob Object'lerden kaçının
 - ↳ Her işe özel sınıflar oluşturun
- Uzun fonksiyonlardan kaçının
 - ↳ Mümkün oldukça daha küçük fonksiyonlara bölün
 - ↳ Sorumluluğu dağıtın
 - ↳ Test edilebilirliği arttırın
 - ↳ Daha küçük birimler için daha kolay test yazılabilir
- Kod tekrarından kaçının
- Kod stili standartı oluşturun (PSR)
- Framework agnostic kod yazın
 - ↳ Bir frameworkü temel alabilirsiniz ama ona göbekten bağlanmayın

Katı Tip Tanımlamaları Kullanın



```
declare(strict_types=1);
```



```
function add(int $x, int $y): int
{
    return $x + $y;
}
```

Bağımlılıkları Azaltın

- Composer kullanımını iyi analiz edin
 - ↳ Gereksiz bağımlılıklardan kurtulun
- Kullandığınız servisleri azaltın
- Güncel sürümleri yakalamayı çalışın
 - ↳ PHP
 - ↳ Composer paketleri
 - ↳ Database
 - ↳ Distro

Dökümantasyon

- Gereksiz doc-block kullanmayın, gereksiz bilgi vermeyin
- Kodunuzu döküman olarak konumlandırın
- Not yazmaktan çekinmeyin
- Gerekli yerlere TODO bırakın, ama sebebini açıklayın
- Business logic bilgisini aktarın

Sıklıkla yapılan hatalar

- Refactoring esnasında yeni bir paradigma, framework vb. öğrenmeyein
 - ↳ Danışmanlık alabilirsiniz
- Tüm projeyi sıfırdan yazmayın (ikinci kez söylüyorum)
- Paydaşları mental olarak hazırlayın
- Ekibi büyütün
- Kısa sürede karar almayın
- Bilgiyi aktarın / Döküman oluşturun

Planlama



SORULAR?



Güven Atbakan
CTO @Moneo

izlediğiniz için
teşekkürler

linkedin.com/in/guvenatbakan • twitter.com/GAtbakan

