

# Örnek Olaylar ile Gereksinim Analizi

YAZILIM MÜHENDİSLİĞİ

Muhammet Baykara



# Gereksinim Analizi Nedir?

Bilgisayar bilimlerinde, gereksinim analizi ya da gereksinim çözümlenme; çeşitli sistemlerin gerekliliklerini ve olası çelişkili durumlarını göz önüne alarak, yazılımı analiz etmek, belgelemek, doğrulamak ve yönetmek için yeni veya değiştirilmiş bir ürün üzerinde projenin ihtiyaçlarını, sistem gereksinimlerini ve koşullarını belirleyen görevleri kapsamaktadır.

# Gereksinim Analizinin Önemi

*Doğru analiz edilmemiş bir sistem en iyi şekilde tasarlansa bile yetersiz sistemdir*

Gereksinim analizi; bir sistem veya yazılım projesinin başarısı veya başarısızlığı açısından kritik önem taşır. İhtiyaçların belgelenmesi, uygulanabilir olması, ölçülebilir olması, test edilebilir formda olması, izlenebilir olması ve belirlenen işletmenin ihtiyaçlarına uygun olması gerekir.

# **Bir Projenin Gereksinim Analizinin Anatomisi**

MÜŞTERİ  
GEREKSİNİMLERİ  
NE UYGUNLUK

STANDART'  
LARA  
BAĞLILIK

İYİ  
BELGELENMİŞ  
SÜREÇLER

*IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğü bir gereksinimin müşteri tarafındaki rolünü şu şekilde tanımlar:*

Müşteri Gereksinimlerine Uygunluk, Bir sorunu çözmek veya bir amaca ulaşmak için kullanıcının ihtiyaç duyduğu bir koşul veya yetenektir.



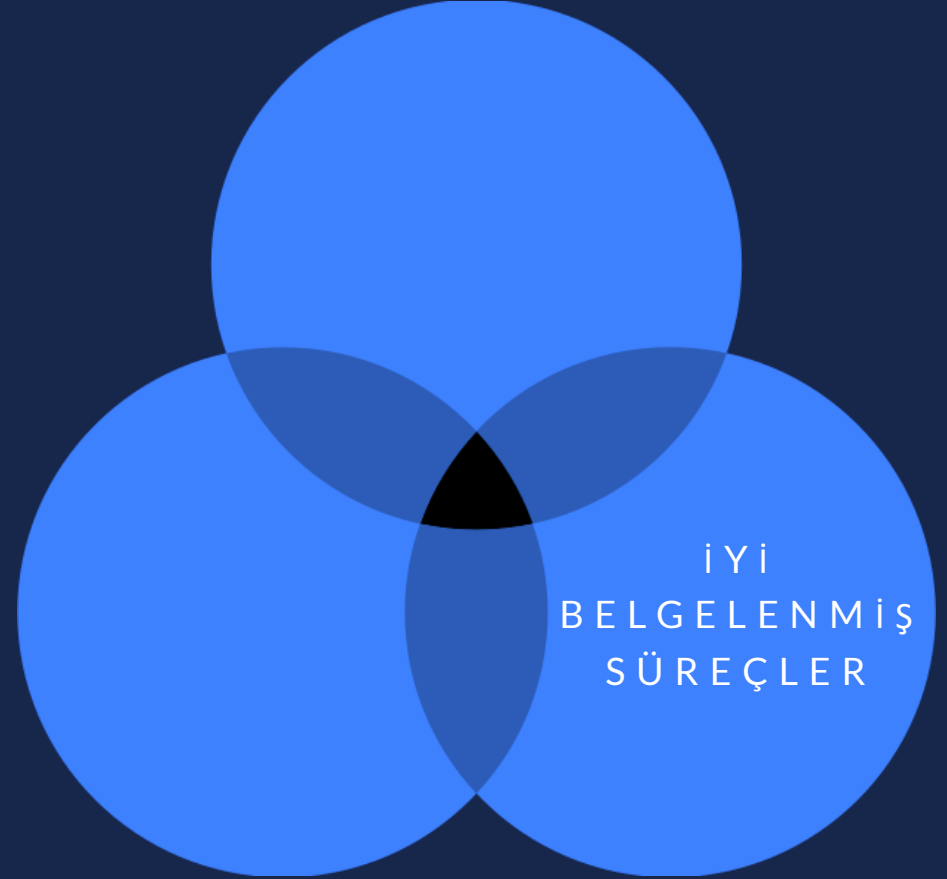
*IEEE Standart Yazılım Mühendisliđi Terminolojisi Sözlüđü bir gereksinimin standartlar tarafındaki rolünü řu řekilde tanımlar:*

Standart'lara bađlılık, bir sözleşmeyi, standardı, spesifikasyonu veya resmi olarak dayatılan başka bir belgeyi karşılamak için bir sistem veya sistem bileşeni tarafından karşılanması veya sahip olunması gereken bir koşul veya yetenektir.



*IEEE Standart Yazılım Mühendisliği Terminolojisi Sözlüğü bir gereksinimin belgeleme tarafındaki rolünü şu şekilde tanımlar:*

iyi belgelenmiş süreçler, Müşteri Gereksinimlerine Uygunluk ve Standartlara bağlı kalma gibi bir koşul veya yeteneğin doğru olarak belgelenmiş bir temsilidir.





# **Bir Bankacılık Uygulaması Üzerinden Gereksinim Analizi**

# Yazılım gereksinimi, sistemde uygulanması gereken işlevsel veya işlevsel olmayan bir ihtiyaçtır.

*İşlevsel Gereksinimler; Kullanıcıya doğrudan sunulan bir hizmeti ifade eden gereksinim türüdür.*

- Örneğin, bankacılık uygulaması bağlamında işlevsel gereksinim, müşteri "Bakiyeyi Görüntüle"yi seçtiğinde, en son hesap bakiyesine bakabilmeleri gerekir.

*İşlevsel Olmayan Gereksinimler: Kullanıcıya doğrudan sunulan hizmeti ifade etmeyen, yazılımın kalitesini ve doğru çalışmasını ele alan bir performans gereksinimidir.*

- Örneğin, işlevsel olmayan bir gereksinim, sistemin her sayfasının 5 saniye içinde kullanıcılara görünür olması gerektiğidir.

# Bankacılık Uygulaması için Gereksinim Tiplerinin Ayrıştırılması

## İŞ GEREKSİNİMLERİ

*Projelerden iş gerekçesinden alınan üst düzey gereksinimlerdir.*

## MİMARİ VE TASARIM GEREKSİNİMLERİ

*Bu gereksinimler, iş gereksinimlerinden daha ayrıntılıdır. İş gereksinimini uygulamak için gereken genel tasarımı belirler.*

## SİSTEM VE ENTEGRASYON GEREKSİNİMLERİ

*En alt seviyede olan sistem ve entegrasyon gereksinimleri, her gereksinimin ayrıntılı açıklamasıdır.*

# İŞ GEREKSİNİMLERİ:

Bir mobil bankacılık hizmet sistemi, Güneydoğu Asya'ya bankacılık hizmetleri sağlar. Hindistan için kararlaştırılan iş gereksinimi hesap özeti ve para transferi iken, Çin için hesap özeti ve fatura ödemesi bir iş gereksinimi olarak kararlaştırılır.

**Ülke**

-----

**Hindistan**

**Çin**

**İhtiyaç duyulan bankacılık hizmet veya servisleri**

-----

**Hesap Özeti ve Fon Transferi**

**Hesap Özeti ve Fatura Ödemesi**

# MİMARİ VE TASARIM GEREK SINİMLERİ:

Bir mobil bankacılık hizmet sisteminin fatura ödeme servisi için Mimari ve Tasarım Gereksinimleri Őu Őekildedir:

## Bankacılık Kullanım rneęi

---

### Fatura deme

## Gereksinim

---

- Bu kullanım rneęi, bir mŐterinin net bankacılıęına nasıl giriŐ yapabileceęini ve Fatura deme Aracını nasıl kullanabileceęini aıklar.
- MŐteri, kayıtlı fatura sahiplerinin denmemiŐ faturalarının bir panosunu grebilir. Bir fatura detayını ekleyebilir, deęiŐtirebilir ve silebilir. MŐteri, farklı faturalandırma iŐlemleri iin SMS, e-posta uyarıları yapılandırabilir. GemiŐ denmiŐ faturaların gemiŐini grebilir.
- Bu kullanım senaryosunu baŐlatan aktrler banka mŐterileri veya destek personelidir.

# Sistem ve Entegrasyon Gereksinimleri

Sistem ve Entegrasyon Gereksinimleri, günlük iş dilini gerçekten tanımlayan kullanıcı hikayeleri şeklinde olabilir. Geliştiricilerin kodlamaya başlayabilmeleri için gereksinimler bol miktarda ayrıntıda verilmiştir.

Burada bir Fatura ekleme gereksiniminin belirtileceği Fatura Ödeme modülü örneğinde:

## Fatura Ödeme Modülü

---

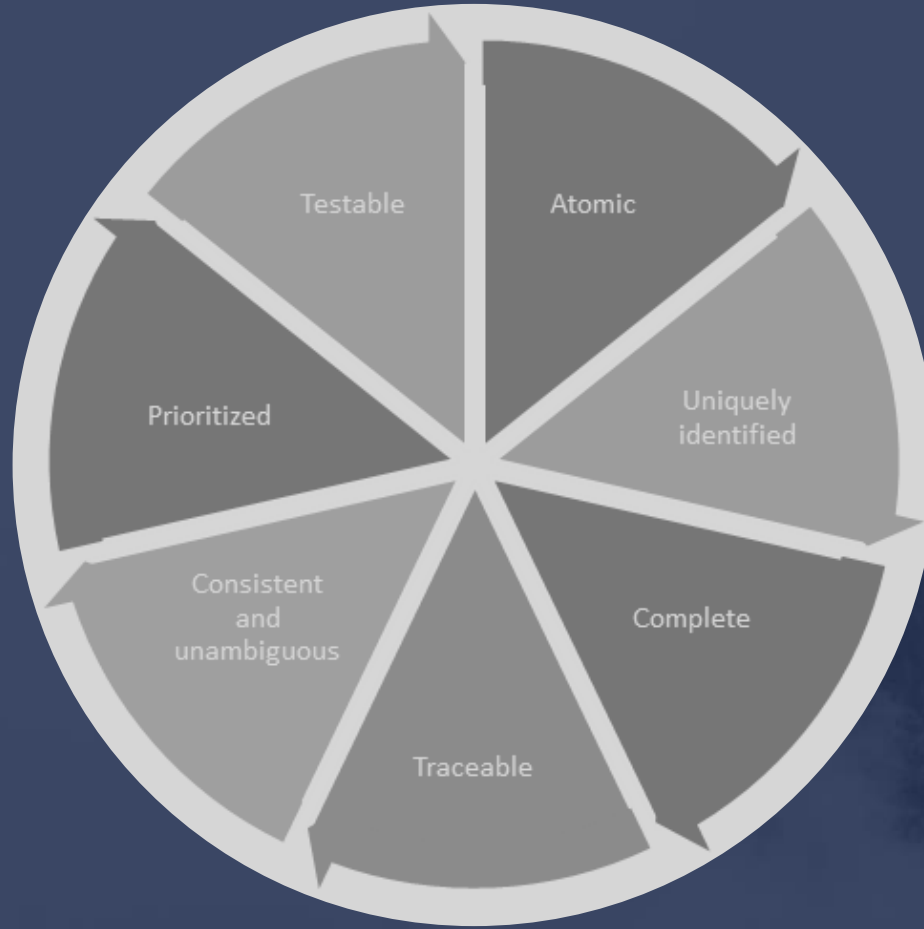
### Faturacı Ekle

## Gereksinim

---

- Hizmet Sağlayıcı Adı
- İlişki Müşteri Numarası
- Otomatik Ödemeler - Evet/Hayır
- Tüm Faturayı Öde - Evet/Hayır
- Otomatik Ödeme Limiti - Fatura belirtilen tutarın üzerindeyse ödeme yapmayın

# **Bir Öğrenci Ders Kayıt Yazılımı Üzerinden Gereksinim Kalitesinin Analizi**



# Gereksinim Analizi Kalitesi Nedir?

Gereksinimlerin belirli özellikleri sağlaması için ve doğruluğunun tespit edilebilmesi için belirlenmiş olan standart değerler kümesi üzerinden analiz edilmesine Gereksinim Analizi Kalitesi denir.



**Gereksinimlerin kaliteli olabilmesi için, gereksinimlerinin standart bir kalitesini sağlamalıdır, farklı gereksinim kalitesi türleri şunları içerir:**

- **ATOMİK**
- **BENZERSİZ OLARAK TANIMLANMIŞ**
- **TAM**
- **TUTARLI VE AÇIK**
- **İZLENEBİLİR**
- **ÖNCELİKLİ**
- **TEST EDİLEBİLİR**

# ATOMİK

Sahip olduğunuz her bir gereksiniminiz atomik olmalıdır, bu da çok düşük düzeyde ayrıntılarda olması gerektiği anlamına gelir, bileşenlere ayrılmanın mümkün olmaması gerekir. Burada, Atomik ve benzersiz olarak tanımlanmış gereksinim seviyelerindeki gereksinimler için iki örneği göreceğiz.

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<ul style="list-style-type: none"><li>• Öğrenciler lisans ve lisansüstü derslere kayıt olabileceklerdir.</li></ul>	<ul style="list-style-type: none"><li>• Öğrenciler lisans derslerine kayıt olabilecekler</li><li>• Öğrenciler lisansüstü derslere kayıt olabilecekler</li></ul>

# Benzersiz olarak tanımlanmış

Benzer şekilde, bir sonraki gereksinim kalitesi, gereksinimlerin benzersiz bir şekilde tanımlanmış olup olmadığını kontrol etmektir, burada iki ayrı gereksinimimiz var, ancak her ikisinin de ID'si 1 olduğundan aynıdır. Bu nedenle, gereksinimlerimizi ID'ye atıfta bulunuyorsak, ancak her ikisi de aynı ID değeri olan 1'e sahip olduğundan, tam olarak hangi gereksinimden bahsettiğimiz açık değil. Böylece gereksinimlerimiz benzersiz id'lerle ayrılarak, benzersiz olarak tanımlanmalıdır, Bölüm 1 ders kayıtları olarak ayrılacak ve bu bölümünde iki koşulu olacaktır: 1.1 id lisans derslerine kayıt, 1.2 id ise lisansüstü derslere kayıttır.

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<b>1- Öğrenciler lisans derslerine kayıt olabilecekler</b> <b>1- Öğrenciler lisansüstü derslere kayıt yapabilecekler</b>	<b>1- Ders Kaydı</b> <b>1.1- Öğrenciler lisans derslerine kayıt olabilecekler</b> <b>1.2- Öğrenciler lisansüstü derslere kayıt olabilecekler</b>

# TAM

Ayrıca, her gereksinimin eksiksiz olması gerekir. Örneğin, burada kötü gereklilik, “profesör kullanıcı, kullanıcı adını, şifresini ve diğer ilgili bilgileri sağlayarak sisteme giriş yapacak” diyor. Burada diğer ilgili bilgiler net değildir, bu nedenle gerekliliği tamamlamak için diğer ilgili bilgiler iyi bir gereksinimle açıklanmalıdır.

## Kötü Gereksinim Örneği

- Bir profesör kullanıcı, kullanıcı adını, şifresini ve diğer ilgili bilgileri girerek sisteme giriş yapacaktır.

## İyi Gereksinim Örneği

- Bir profesör kullanıcı, kullanıcı adını, şifresini ve bölüm kodunu girerek sisteme giriş yapacaktır.

# TUTARLI VE AÇIK

Daha sonra her bir gereklilik tutarlı ve açık olmalıdır, yani burada örneğin “Bir öğrencinin ya lisans dersleri olacak ya da lisansüstü dersleri olacak, ancak her ikisi birden olmayacak” gereksinimlerimiz var, bu bir gerekliliktir, “Bazı dersler hem lisans hem de lisansüstü öğrencilere açık olmalıdır”. Bu iki durum birbiriyle çelişkili olduğundan şartın “Bir öğrencinin ya lisans dersleri ya da lisansüstü dersleri olacak ama her ikisi birden olmayacak” şeklinde iyi bir koşula dönüştürülmesi açıktır. Bu, her dersin lisans dersi veya lisansüstü ders olarak işaretleneceği anlamına gelir.

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<ul style="list-style-type: none"><li>• Bir öğrencinin ya lisans dersleri ya da lisansüstü dersleri olur, ancak her ikisi birden olmaz. Bazı dersler hem lisans hem de lisansüstü öğrencilere açık olacaktır.</li></ul>	<ul style="list-style-type: none"><li>• Bir öğrenci ya lisans ya da yüksek lisans mezunu olacak, ancak ikisi birden değil</li></ul>

# İZLENEBİLİR

Her bir gereksinim izlenebilir olmalıdır çünkü zaten farklı gereksinim seviyeleri vardır, zaten en üstte iş gereksinimlerimiz olduğunu gördük ve ardından bir mimari ve tasarım gereksinimlerimiz ve ardından sistem entegrasyon gereksinimlerimiz var. Dolayısıyla Gereksinimlerin farklı seviyeler arasında izlenebilmesini sağlamak amacıyla her bir gereksinim belirli düzeylerde id eşlemelerine sahip olmalıdır. Bu kapsamda birinci gereksinimde id verilmemiş olduğundan belirtilen Gereksinimin takibi mümkün olmadığından ikinci gereksinimde id tanımı yapılarak belirtilen gereksinimin verilen id üzerinden izlenebilir olması sağlanmıştır

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<ul style="list-style-type: none"><li>• BRD req.ID ile eşlenmiş öğrenci bilgileri korunsun mu?</li></ul>	<ul style="list-style-type: none"><li>• Öğrenci bilgilerini koruyun-BRD req ID 4.1 ile eşleştirin</li></ul>

# ÖNCELİKLİ

Daha sonra her bir gereksinime öncelik verilmelidir, bu nedenle ekibin hangi gereksinimin önce uygulanabileceğini ve hangilerinin daha sonra yapılabileceğini gösteren bir kılavuzu vardır. Burada kötü önceliğin öğrenciyi kaydettirdiğini, kullanıcı bilgilerini koruduğunu ve her gereksinimin öncelik-1 verdiğini görebilirsiniz. Her şey aynı öncelikte olamaz, bu nedenle gereksinim önceliklendirilebilir. Bu nedenle, buradaki iyi gereksinim örneği, kayıt olan öğrenci ve kayıt kurslarına en yüksek öncelik 1 verilirken, kullanıcı bilgilerini koruma önceliği 2'nin altında gelir ve ardından rapor kartını öncelik-3'te görürüz.

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<ul style="list-style-type: none"><li>• Kayıtlı öğrenci-Öncelik 1</li><li>• Kullanıcı Bilgilerini Koru- Öncelik 1</li><li>• Kurslara kaydolun- Öncelik 1</li><li>• Rapor Kartını Görüntüle- Öncelik 1</li></ul>	<ul style="list-style-type: none"><li>• Kayıtlı öğrenci-Öncelik 1</li><li>• Kullanıcı Bilgilerini Koru- Öncelik 2</li><li>• Kurslara kaydolun- Öncelik 1</li><li>• Rapor Kartını Görüntüle- Öncelik 3</li></ul>

# TEST EDİLEBİLİR

Her gereksinim test edilebilir olmalıdır, burada kötü gereksinim "sistemin her sayfası kabul edilebilir bir zaman çerçevesinde yüklenecektir". Şimdi, bu gereksinimle ilgili iki sorun var, ilk olarak, her sayfanın, test çabalarını havaya uçuracak birçok sayfa olabileceği anlamına gelmesidir. Diğer sorun ise sayfanın kabul edilebilir bir zaman diliminde yükleneceğini söylemesi, şimdi kabul edilebilir zaman dilimi nedir? Kime göre kabul edilebilir. Bu nedenle, test edilemeyen argümanı, özellikle "öğrenciyi kaydet ve kurslara kaydol sayfaları" hakkında hangi sayfadan bahsettiğimizi söyleyen test edilebilir bir argümana dönüştürmeliyiz ve kabul edilebilir bir zaman çerçevesi de vermeliyiz, bu da 5 saniyedir.

<b>Kötü Gereksinim Örneği</b>	<b>İyi Gereksinim Örneği</b>
<ul style="list-style-type: none"><li>• Sistemin her sayfası kabul edilebilir bir zaman diliminde yüklenecektir.</li></ul>	<ul style="list-style-type: none"><li>• Sistemin öğrenci kaydı ve ders kaydı sayfaları 5 saniye içinde yüklenecektir.</li></ul>





## SONUÇ

Gereksinimlerin Analiz Edilmesi, bir projenin başlangıç safhasında ortaya çıkan hataları en aza indirdiğinden ve projenin ilerleyen safhalarında ortaya çıkacak hataları minimize ederek maliyeti azalttığından çok önemlidir.