

# **Gereksinim Mühendisliđi**

**22.12.2022**

# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# KAVRAMLAR

## Paydaş (Stakeholder):



- Doğrudan ya da dolaylı olarak proje dahil olan,
- Proje çıktılarını etkileyebilecek veya proje çıktılarından etkilenebilecek

**Kişi, Grup, Şirket vb. unsurlardır.**



# KAVRAMLAR

## PAYDAŞLAR

### İÇ PAYDAŞLAR

Proje Yöneticisi  
Proje Ekibi

### ARADAKİ PAYDAŞLAR

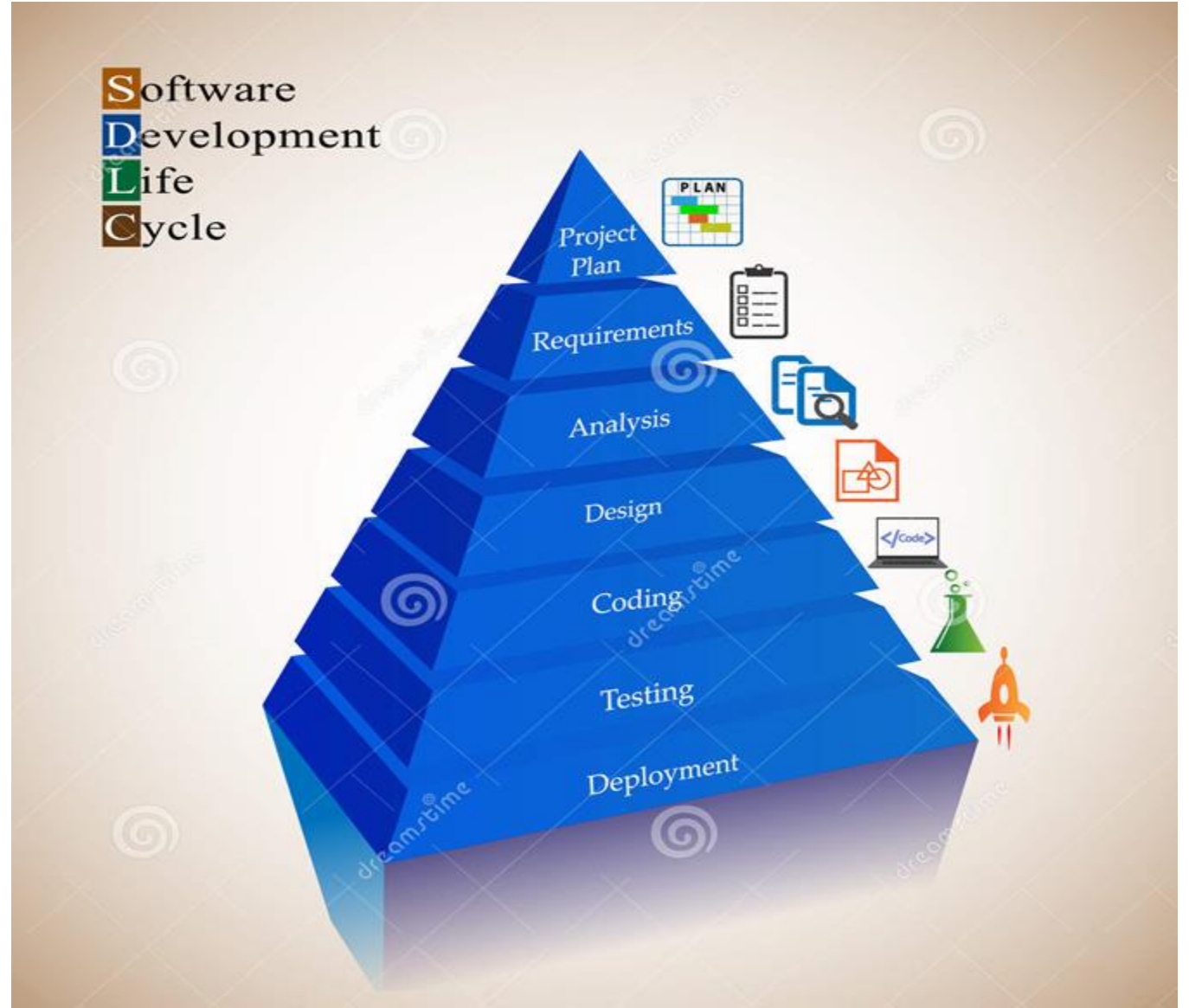
Müşteri / Kullanıcı  
Finans Sağlayıcı  
Şirket (Yüklenici)

### DIŞ PAYDAŞLAR

Hükümet  
(Government)  
Sağlayıcı (Supplier)

# KAVRAMLAR

## YAZILIM YAŞAM DÖNGÜSÜ



# KAVRAMLAR

## Yaşam Döngüsü Modelleri

Şelale  
(Waterfall)

V-shaped Model

Spiral Model

İteratif

İteratif Arttırımlı

Çevik (Agile)

# KAVRAMLAR

## Yaşam Döngüsü Modelleri

Şelale  
(Waterfall)

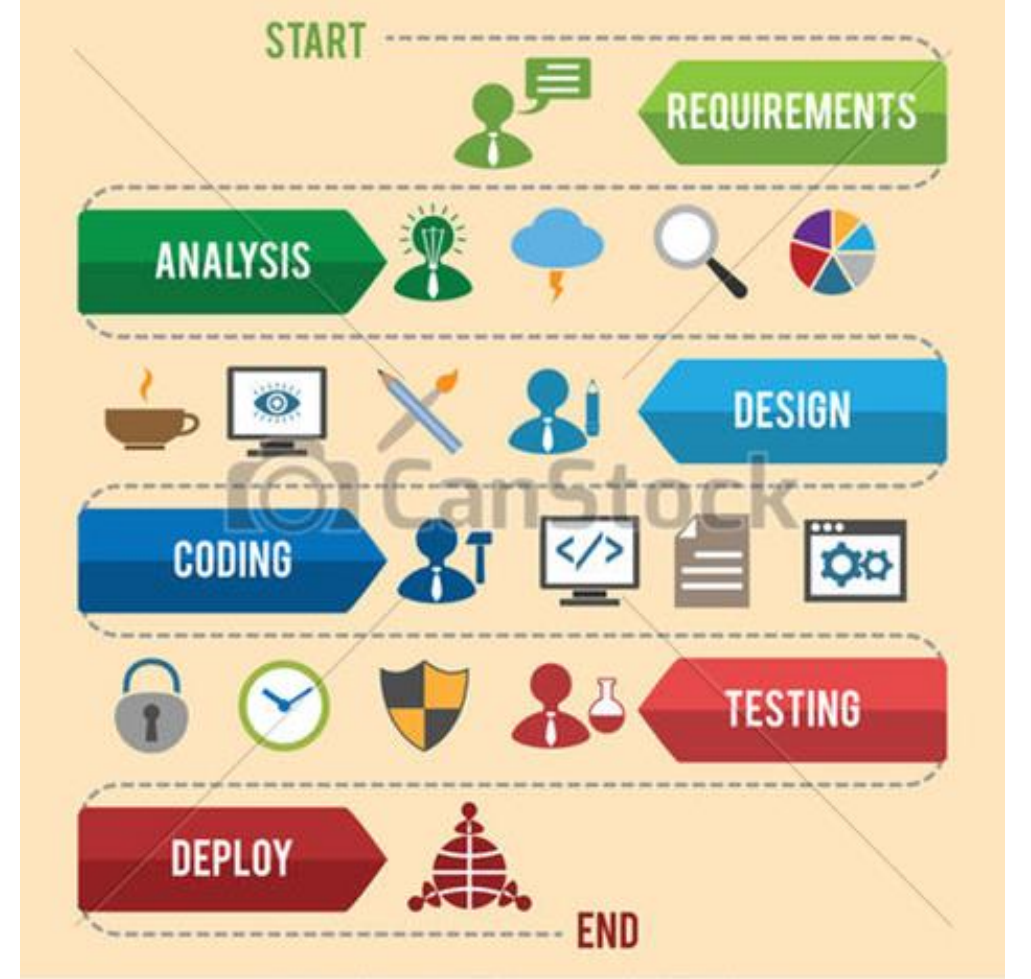
V-shaped Model

Spiral Model

İteratif

İteratif Arttırımlı

Çevik (Agile)

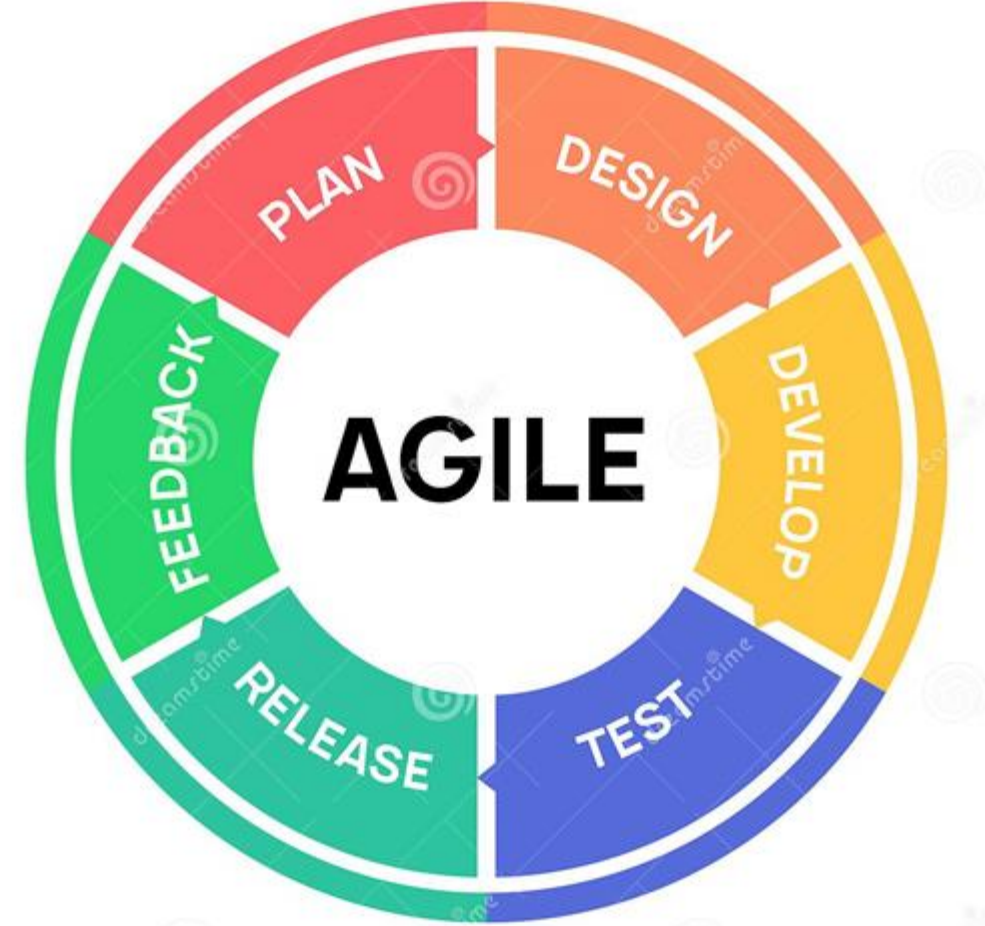


# KAVRAMLAR

## Yaşam Döngüsü Modelleri

Şelale  
(Waterfall)  
V-shaped Model  
Spiral Model

İteratif  
İteratif Arttırımlı  
Çevik (Agile)





# İçerik

- Kavramlar
- **Paydaş Gereksinimleri Nasıl Toplanır?**
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# Paydař Gereksinimleri Nasıl Toplanır?



# Ortaya ıkarma (Elicitation) Teknikleri

Müşteri ile  
Toplantı

Müşteri  
Ortamında  
Gözlem

Doküman  
Analizi

Beyin Fırtınası

Kullanım  
Durumu  
Senaryosu

Kullanıcı  
Hikayeleri

# Ortaya ıkarma (Elicitation) Teknikleri

Tersine  
Mühendislik

Arayüz  
Analizi

Süreç  
Modelleme

Anket

Prototip

Simülasyon

# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- **Gereksinimlerin Dokümantasyonu**
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# Gereksinimlerin Dokümantasyonu

- Metodolojilere göre deęişiklik gösterebilir
  - Çevik
  - Dięer
- Standartlar
  - 830-1998 - IEEE Recommended Practice for Software Requirements Specifications
- Modeller
  - Capability Maturity Model Integration (CMMI)



**CMMI<sup>®</sup> Institute**

# Gereksinimlerin Dokümantasyonu

**Yazılım Gereksinimleri  
Dokümanı**

**İzlenebilirlik Matrisi**

**Paydaş Gereksinimleri  
Dokümanı**

**Paydaş Gereksinimleri  
Listesi**

# Yazılım Gereksinimleri Dokümanı

Amaç, Kapsam, Tanımlar, Kısaltmalar

Operasyonel Konseptler

Varsayımlar Bağımlılıklar

Tasarım Kısıtları

Kullanıcı Grupları

Fonksiyonel Gereksinimler

Fonksiyonel Olmayan Gereksinimler

İzlenebilirlik



# Operasyonel Konseptler (Concepts of Operation)

IEEE Std 1362™-1998 (R2007)  
(Incorporates  
IEEE Std 1362a-1998)

Bir sistemin operasyonel (işletme) özelliklerini son kullanıcının bakış açısından açıklayan, kullanıcı odaklı bir dokümantasyondur.

## IEEE Guide for Information Technology—System Definition— Concept of Operations (ConOps) Document

Sponsor  
**Software Engineering Standards Committee  
of the  
IEEE Computer Society**

Approved 19 March 1998  
Reaffirmed 5 December 2007

**IEEE-SA Standards Board**

**Abstract:** The format and contents of a concept of operations (ConOps) document are described. A ConOps is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint. The ConOps document is used to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements (for example, training, facilities, staffing, and maintenance). It is used to describe the user organization(s), mission(s), and organizational objectives from an integrated systems point of view.

# Operasyonel Konseptler (Concepts of Operation)(devamı)

## Canlı Ortam

- *Proje canlıya alındıktan sonrası tarif edilir.*
- *Canlı ortam iç ve dış faktörleriyle birlikte tanımlanır.*

## • Organizasyon Yapısı

- Projeyi canlı ortamda yürütecek organizasyon yapısı
- Rol, sorumluluk ve çalışma alanı gibi kavramlar anlatılır.

## • İş Süreçleri

- İş akışları tanımlanır.
- Kullanıcılar ile sistem arasındaki ilişkileri içeren iş süreçleri tanımlanır.
- Senaryo şeklinde anlatılabilir.
- Aktivite diyagramları kullanılabilir

# Operasyonel Konseptler (Concepts of Operation) (devamı)

## • İşletme Politikası ve İş Kuralları

- Tanımlanmış iş akışlarının,
  - Başlaması
  - Alternatif akışların gerçekleştirilmesi
  - Akışların sonlandırılması gibi durumlarla ilgili politika ve iş kuralları verilir.

## • İşletme Kısıtları

- Canlıya alındıktan sonraki operasyonel süreçler ile ilgili kısıtlar anlatılır

## • İşletme Yöntemleri

- *Beklenmeyen durumlarda ortaya konulacak işletme yöntemleri anlatılır.*
- *Felaket, kaza gibi durumlar için sistemin olağan yapısı dışındaki*
  - *Manuel işletme yöntemleri*
  - *Felaket kurtarma yöntemleri gibi durumlar anlatılır.*

# Operasyonel Konseptler (Concepts of Operation) (devamı)

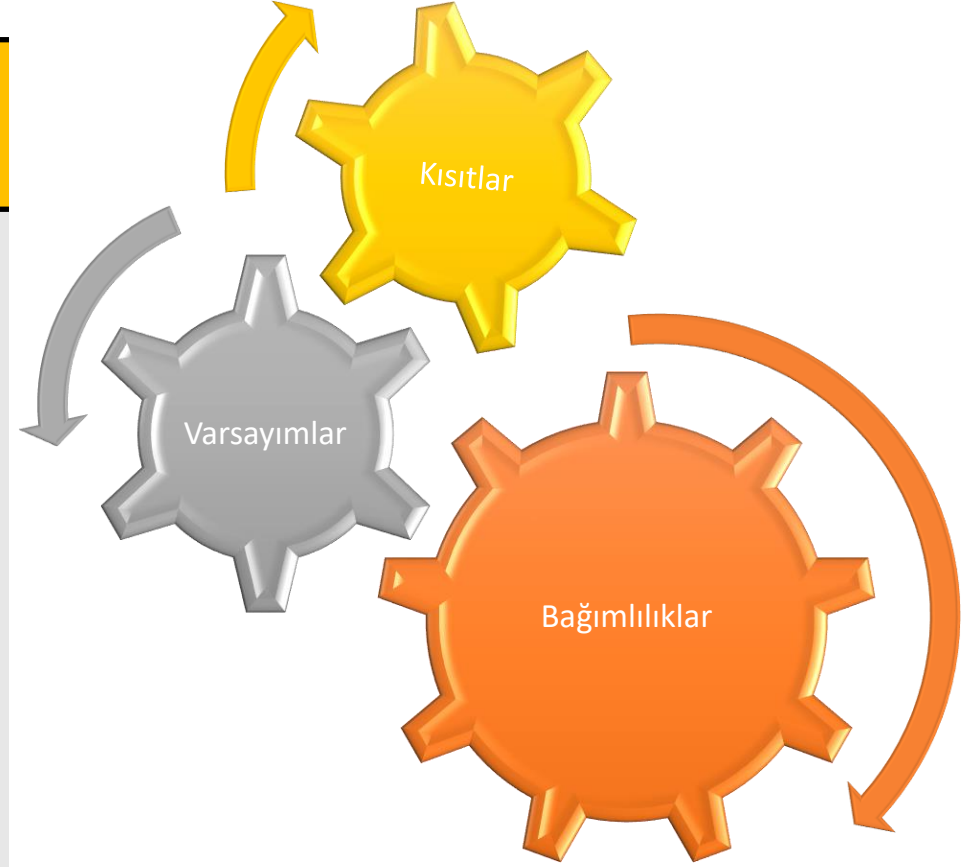
- Ayrı bir doküman olarak oluşturmak istenirse

Title page
Revision chart
Preface
Table of contents
List of figures
List of tables
1. Scope
1.1 Identification
1.2 Document overview
1.3 System overview
2. Referenced documents
3. Current system or situation
3.1 Background, objectives, and scope
3.2 Operational policies and constraints
3.3 Description of the current system or situation
3.4 Modes of operation for the current system or situation
3.5 User classes and other involved personnel
3.6 Support environment
4. Justification for and nature of changes
4.1 Justification of changes
4.2 Description of desired changes
4.3 Priorities among changes
4.4 Changes considered but not included
5. Concepts for the proposed system
5.1 Background, objectives, and scope
5.2 Operational policies and constraints
5.3 Description of the proposed system
5.4 Modes of operation
5.5 User classes and other involved personnel
5.6 Support environment
6. Operational scenarios
7. Summary of impacts
7.1 Operational impacts
7.2 Organizational impacts
7.3 Impacts during development
8. Analysis of the proposed system
8.1 Summary of improvements
8.2 Disadvantages and limitations
8.3 Alternatives and trade-offs considered
9. Notes
Appendices
Glossary

# Varsayımlar Bağımlılıklar

## ÖRNEKLER

- Sistemin parmak izi doğrulama yapması için kullanıcının parmak izi okuyup doğrulama yapabilen bir mobil cihaza (akıllı cep telefonu, tablet, bilgisayar) sahip olması gerekmektedir.
- Kullanıcıların e-devlet üzerinden doğrulanacağı varsayılmıştır.
- Sistemin e-imza ile imzalama yeteneği olması gerektiği için e-imza donanımlarına bağımlılığı bulunmaktadır.



# Tasarım Kısıtları

- Tasarım sürecini kısıtlayacak gereksinimler anlatılır.



CONSTRAINT

## ÖRNEKLER

- Sistem, internete kapalı bir ortamda çalışmalıdır.
- Tüm kullanıcı girişleri e-imza ile yapılmalıdır.
- Sistem için geliştirilecek ekranlarda pop-up kullanılmamalıdır.
- Saklama Prosedürleri (Store Procedure) kullanılmamalıdır.

# Kullanıcı Grupları

- Kullanıcı grupları tanımlanır.
- Temel düzeyde ne yaptıklarının anlatılır.



## Kullanıcı Grubu

## Açıklama

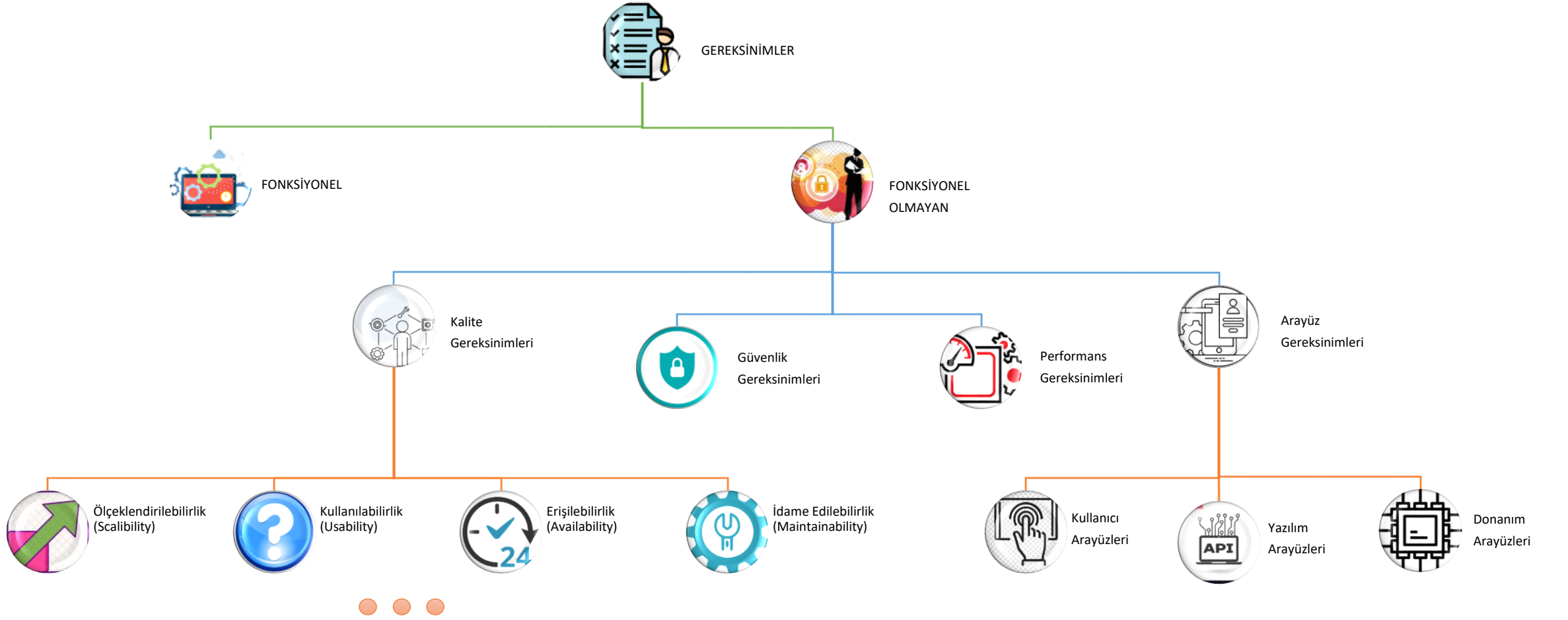
Borsa Yöneticisi

İşlem İptali

Seans içerisinde veya seans sonrasında emir ve pozisyon girişi

Seans içerisinde veya seans sonrasında emir ve işlem iptali/düzeltilmesi

...





# FONKSİYONEL GEREKSİNİMLER

Yazılımdan beklenen,

- Girdisi
- Çıktısı
- İşlevi

olan her şey.



“Müşterinin, yazılımdan beklediği tüm fonksiyonallite”

# FONKSİYONEL GEREKSİNİMLER

## ÖRNEK:

- Yatırımcılar, sisteme emir girebilmelidir.
- Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
- Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.

# FONKSİYONEL GEREKSİNİMLER

Yatırımcılar, sisteme emir girebilmelidir.

**Girdi:** Emir parametreleri

**Çıktı:** Sisteme girilmiş emrin kendisi, sisteme etkisi ve sistemde bıraktığı izler.

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

- Arayüz Gereksinimleri
  - Kullanıcı Arayüzleri
  - Yazılım Arayüzleri
  - Donanım Arayüzleri



“Kullanıcının etkileşimde  
bulunduğu arayüzler”



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

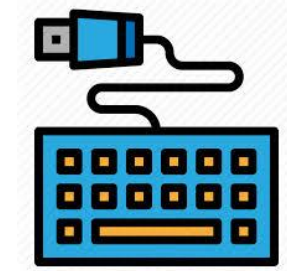
- Arayüz Gereksinimleri
  - Kullanıcı Arayüzleri
  - Yazılım Arayüzleri
  - Donanım Arayüzleri

“Geliştirilecek olan sistemin diğer yazılımsal sistemlerle etkileşimde bulunduğu arayüzler”



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

- Arayüz Gereksinimleri
  - Kullanıcı Arayüzleri
  - Yazılım Arayüzleri
  - Donanım Arayüzleri

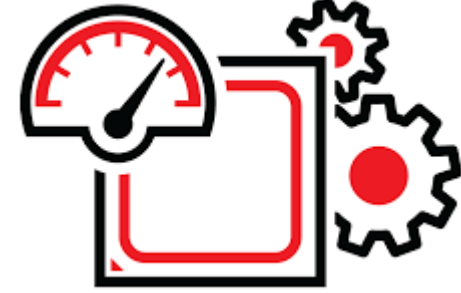


# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Performans Gereksinimleri

NASIL OLMALI?  
Ölçülebilir, Test Edilebilir ve Makul

- Aynı anda kaç kullanıcıya hizmet edeceği (1000, 100000 kullanıcı)
- Bir işlemi ne kadar sürede yapacağı (Saniye, Milisaniye, ...)



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Güvenlik Gereksinimleri

Kriptografik  
Yöntemler

Yetkilendirme  
Politikası

Veri  
Bütünlüğü

Veri Gizliliği



### ÖRNEK:

- *Sistem, kişisel veri kapsamına giren verileri veritabanında maskeleyerek saklayacaktır.*
- *Sistem, sadece izin verilmiş IP'ler üzerinden istem alacaktır.*
- *Sistem, her gün veritabanının özetini (Hash) hesaplayarak kaydedecektir.*



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Ölçeklendirilebilirlik (Scalability)

*Sistemin,*

- *Artan iş yükünün yönetilebilmesi*
- *Büyümeyi karşılayabilmek*

*için genişletilebilmesi ile ilgili gereksinimlerdir.*



**ÖRNEK:**

- *Sistem, yeni müşterilerin yıllık %10'luk büyümesini destekleyebilecektir.*
- *Sistem, işlem sayısında yıllık %15'lik bir büyümeyi destekleyebilecektir.*
- *Sistem, yeni bir marka eklenmesine olanak sağlayacaktır.*
- *Sistemi kullanan tüm markalar aynı sistem ve süreçleri kullanabilecektir.*

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Erişilebilirlik (Availability)

*Herhangi bir zamanda, tanımlanmış koşullar altında kullanıldığında beklenen fonksiyonu gerçekleştirilme olasılığı ile ilgili olan gereksinimlerdir.*



### ÖRNEK:

- *Sistem, Pazartesi'den Cumartesi'ye 08:00 - 19:00 saatleri arasında normal kullanım için hazır olacaktır.*
- *Sistem, her gün 22:00 - 02:00 saatleri arasında sistem bakımı için kullanılabilir.*
- *Sistem, Birleşik Krallık'taki bankaların kapanış döneminde ve tüm resmi tatillerde normal kullanım veya sistem bakımı amacıyla kullanılamayacaktır.*

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## İdame Edilebilirlik (Maintainability)

*Teslim sonrası ortaya çıkabilecek olan hatalardan kaynaklı*

- *Yetersizliklerin teşhis edilebilmesi*
- *Bu yetersizliklerin giderilebilmesi yeteneđi ile ilgili olan gereksinimlerdir.*



### ÖRNEK:

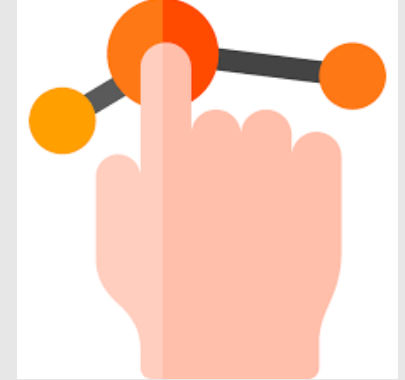
- *Bir sistem arızasının ardından sistemi geri yüklemek için ortalama süre (MTRS) 10 dakikadan fazla olmamalıdır.*
- *Bir sistem arızası durumunda 2 sn içinde yedek sunuculara geçiş yapılacaktır.*

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Kullanılabilirlik (Usability)

*Sistemin, kullanıcı tarafından*

- *Anlaşılabilmesi*
  - *Öğrenilebilmesi*
  - *Kullanılabilmesi yeteneği*
- ile ilgili gereksinimlerdir.*



**ÖRNEK:**

- *Sistem, görme engelli kullanıcıların kullanımına imkan sunacaktır.*
- *Kullanıcıların sistemi kolay kullanabilmesi için kullanıcı arayüzlerinde ilgili arayüze yönelik kullanım kılavuzu ihtiva edecektir.*

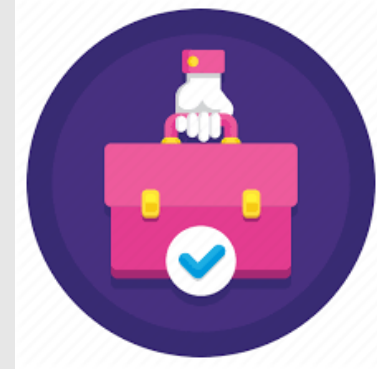
# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Taşınabilirlik (Portability)

*Bir ortamdan farklı bir ortama taşınabilmesi yeteneđi ile ilgili olan gereksinimlerdir.*

### ÖRNEK:

- *Sistem, veritabanı çeşidinden bağımsız olarak çalışabilmelidir.*
- *Windows 10'da çalışan bir sistem, davranışında ve performansında herhangi bir deđişiklik olmadan Windows 11'de çalışabilmelidir.*
- *Sistem, cep telefonu, tablet ve tarayıcılardan açıldığında ekran boyutuna adaptif olmalıdır.*



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Uyarlanabilirlik (Adaptability)

Kullanıcı ihtiyaçlarını karşılayabilmek için

- Kullanıcı tarafından özelleştirilebilmesi (Favoriler gibi)
- Sistem veya yazılımın kendisini kullanıcı için yapılandırabilmesi yeteneği ile ilgili olan gereksinimlerdir.



### ÖRNEK:

- Kullanıcılar, sık kullandıkları işlevlerin kısayollarını oluşturabilmelidir.
- Kullanıcılar, kullanıcı arayüzlerinde bulunan menü ve sekmeler için renk paletleri arasından seçim yapabilmelidir.
- Kullanıcılar, yazı fontunu küçültüp büyütebilmelidir.

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

## Tekrar Kullanılabilirlik (Reusability)

*Başka bir sistem veya yazılım geliştirme sürecinde tekrar kullanılabilmesi yeteneği ile ilgili gereksinimlerdir.*



### ÖRNEK:

- Elektronik imza modülü, ileride geliştirilebilecek sistemlere de kolayca entegre edilebilecek şekilde geliştirilmelidir.*
- Kimlik doğrulama modülü, ileride geliştirilebilecek sistemlere de kolayca entegre edilebilecek şekilde geliştirilmelidir.*
- Sistemin ürettiği anahtarlar, birlikte çalışabilirlik (interoperability) ilkesine uygun olarak geliştirilecektir.*

# Gereksinim Kriterleri

- Tekil tanımlayıcı (uniquely identified)
- Standart yapı
- Zorunluluk
- Doğrulanabilirlik (Verifiable)
- Tamlık, eksiksizlik (Complete)
- Tutarlılık (Consistent)
- Makul (Feasible)
- İzlenebilirlik (Traceable)
- Sadelik ve açıklık (Clear)
- Tasarımdan bağımsızlık (Design-independent)





## Tekil tanımlayıcı (uniquely identified)

- Her gereksinim için tek ve farklı bir tanımlayıcı olmalıdır.

### ÖRNEK:

GRK-1	Yatırımcılar, sisteme emir girebilmelidir.
GRK-2	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
GRK-3	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.

### ya da

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.

# Standart yapı

- Eylemler geniş veya gelecek zaman kipinde olmalıdır.
- Bütün gereksinimler için aynı zaman kipi kullanılmalıdır.
- Gereksinimlerde aktörler belirtilmiş olmalıdır.



## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirecektir. 😞

**SORUN NE? -> Geniş zaman ile başladığımız gelecek zaman kullanmamalıyız!**

# Standart yapı

- Eylemler geniş veya gelecek zaman kipinde olmalıdır.
- Bütün gereksinimler için aynı zaman kipi kullanılmalıdır.
- Gereksinimlerde aktörler belirtilmiş olmalıdır.



## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir. 😊

# Zorunluluk



- Gereksinimler sistem için zorunlu olmalıdır.
- Çıkarıldığında sistem için herhangi bir deęişiklik olmayan ifadeler gereksinim olarak deęerlendirilmez.

## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduęu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
YN-2	Borsa Yöneticisi, sistem sürüm bilgisinin üzerini tıklayabilmelidir. ☹️

# Zorunluluk



- Gereksinimler sistem için zorunlu olmalıdır.
- Çıkarıldığında sistem için herhangi bir deęişiklik olmayan ifadeler gereksinim olarak deęerlendirilmez.

## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduęu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
<del>YN-2</del>	<del>Borsa Yöneticisi, sistem sürüm bilgisinin üzerini tıklayabilmelidir.</del>

# Dođrulanabilirlik (Verifiable)



- Gereksinimler ölçülebilir ve dođrulanabilir olmalıdır.
- Gereksinimler tanımlanırken öznel olarak deđerlendirilebilecek ifadeler kullanılmamalıdır.
- Öznel ifade: hızlı, kolay, hemen hemen, çođunlukla, vb.

## KÖTÜ ÖRNEK:

PL-1	Yatırımcılar, sisteme kolay emir girebilmelidir.	☹️
YN-1	Borsa Yöneticisi, seansın açık olduđu saatleri sisteme çođunlukla tanımlayabilmelidir.	☹️
PL-2	Sistem, yatırımcılar tarafından girilmiř emirleri hızlı eşleřtirmelidir.	☹️

# Tamlık, eksiksizlik (Complete)

Gereksinimler, kendi içinde tam ve eksiksiz olmalıdır.

## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
YN-2	Borsa Yöneticisi, sistem sürüm bilgisinin üzerini tıklayabilmelidir. 😞



# Tamlık, eksiksizlik (Complete)

Gereksinimler, kendi içinde tam ve eksiksiz olmalıdır.

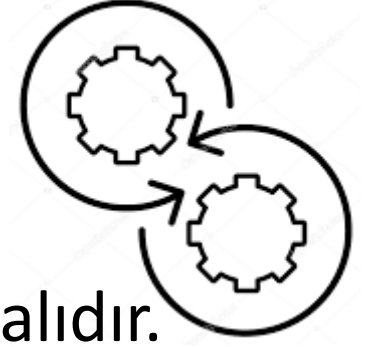
## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme tanımlayabilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
<del>YN-2</del>	<del>Borsa Yöneticisi, sistem sürüm bilgisinin üzerini tıklayabilmelidir.</del>





# Tutarlılık (Consistent)

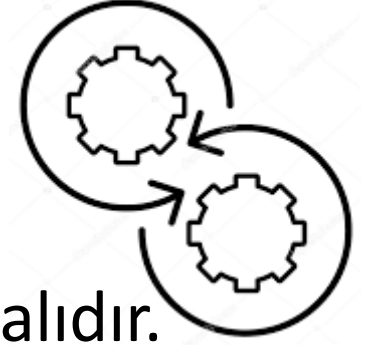


- Gereksinimler, diğer gereksinimler ve kaynaklarla tutarlı olmalıdır.

## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri güncelleyebilmelidir.
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
YN-2	Borsa Yöneticisi, seansın açık olduğu saatlere müdahale edememelidir. 😞

# Tutarlılık (Consistent)



- Gereksinimler, diğer gereksinimler ve kaynaklarla tutarlı olmalıdır.

## ÖRNEK:

PL-1	Yatırımcılar, sisteme emir girebilmelidir.
YN-1	<b>Borsa Yöneticisi, seansın açık olduğu saatleri güncelleyebilmelidir.</b>
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri eşleştirmelidir.
<del>YN-2</del>	<del>Borsa Yöneticisi, seansın açık olduğu saatlere müdahale edememelidir.</del>

# Makul (Feasible)

- Gereksinimler maliyet ile tutarlı olmalıdır.
- Gereksinimler takvim ile tutarlı olmalıdır.



Proje takvimini aşacak kapsama sahip gereksinimler



Proje maliyeti ile çözüm üretilmeyecek gereksinimler

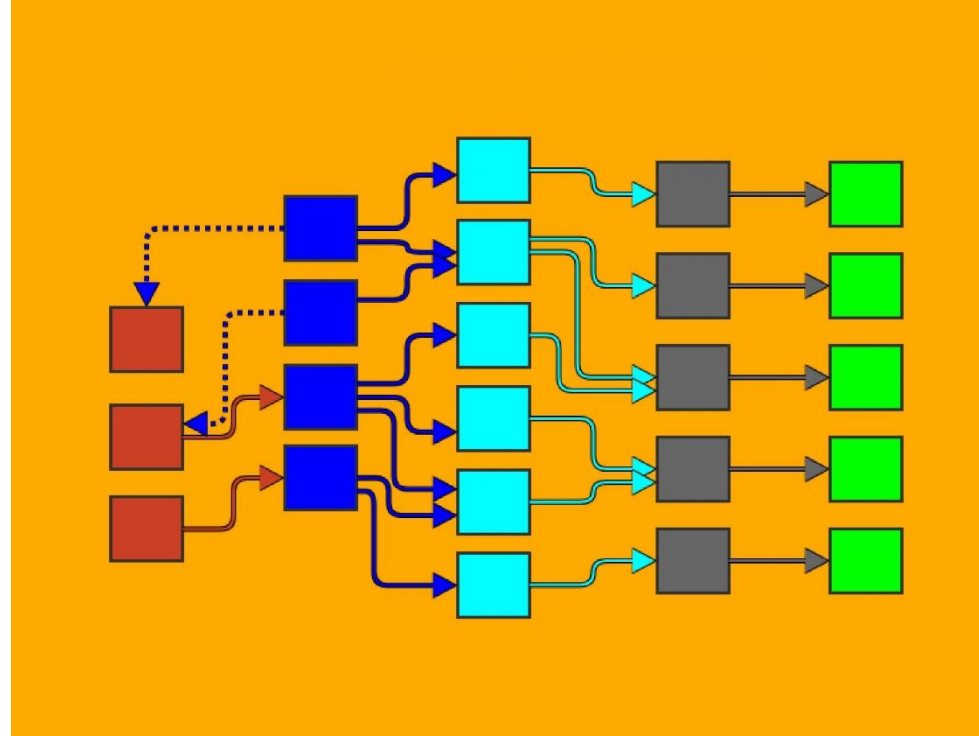


Varolan teknoloji ve bilim kaynaklarıyla gerçekleştirilemeyen gereksinimler

**UYGUN DEĞİLDİR!**

# İzlenebilirlik (Traceable)

- Gereksinimlerle diğer kaynaklar arasında izlenebilirlik sağlanabilir olmalıdır.
- Diğer Kaynaklar:
  - Teknik Şartname
  - Standartlar
  - Yönerge
  - Tasarım Dokümanı
  - Test Dokümanı



# Sadelik ve açıklık (Clear)



- Gereksinimler basit bir dille yazılmalıdır.
- Sistemin tek bir işlevini tanımlamalıdır.
- Gereksinimlerde muğlak ifadeler yer almamalıdır.

## KÖTÜ ÖRNEK:

PL-1	Yatırımcılar, sisteme bazı zamanlarda emir girebilmelidir.	☹️
YN-1	Borsa Yöneticisi, açık olunan saatleri sisteme tanımlayabilmelidir.	☹️
PL-2	Yatırımcılar, sisteme giriş yaparak sistemden raporlar alabileceklerdir.	😱



# Tasarımdan bağımsızlık (Design-independent)

- Gereksinimler tasarım detaylarına inmeyecek seviyede olmalıdır.
- Gereksinimlerde teknik detaylardan bahsedilmemelidir.

## KÖTÜ ÖRNEK:

PL-1	Yatırımcılar, sisteme text alanları kullanarak emir girebilmelidir.	😬
YN-1	Borsa Yöneticisi, seansın açık olduğu saatleri sisteme buton yardımıyla tanımlayabilmelidir.	😬
PL-2	Sistem, yatırımcılar tarafından girilmiş emirleri ekranda eşleştirmelidir.	😬

# Kaçınılması Gerekenler

- Belirsizlik (Vagueness)
- Fazla detaylı şartname (Over specification)
- Nesnellik (Subjectivity)
- Tekrarlar (Multiplicity)
- Açık olmayan anlamlardan (Unclear meaning)
- Kapalı anlamlardan (Implicit meaning)

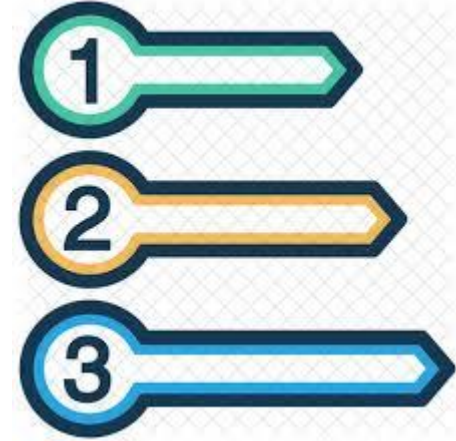
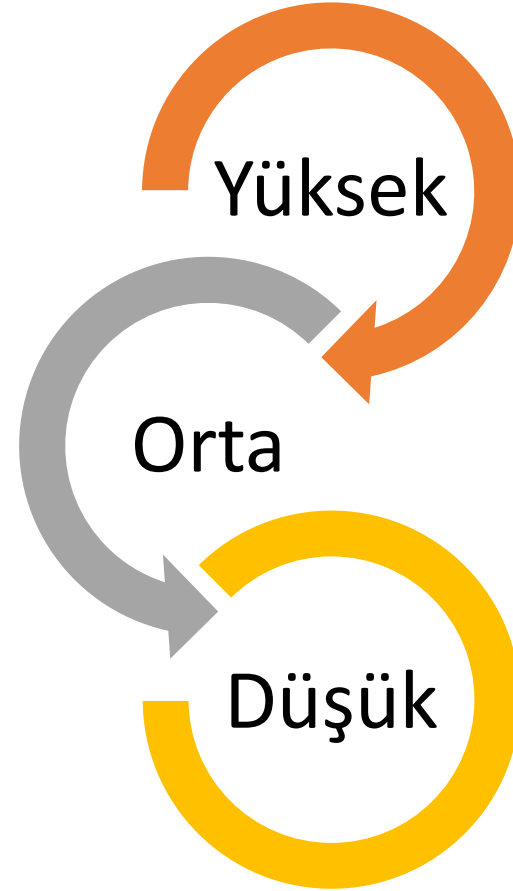
# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- **Paydaş Gereksinimlerinin Önceliklendirmesi**
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?



# Paydaş Gereksinimlerinin Önceliklendirmesi

- Önem
- Geliştirme Önceliği
- Zorluk
- Bağımlılık



# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- **Gereksinimlerin Geliştirilmesi**
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

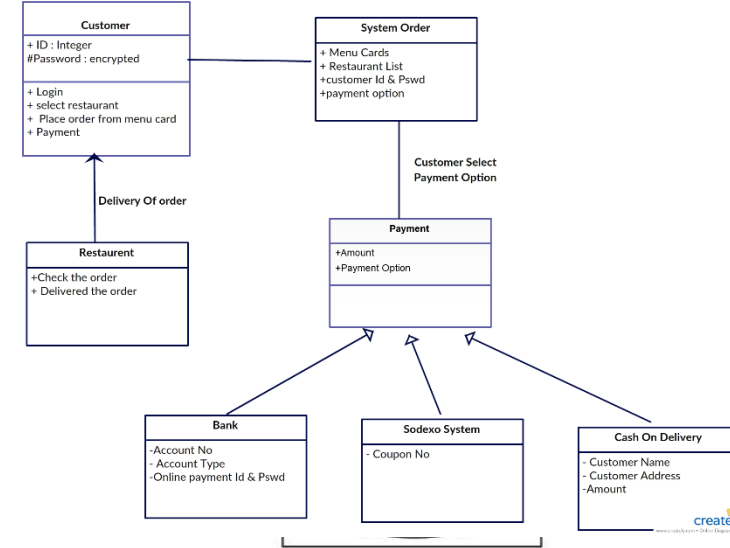
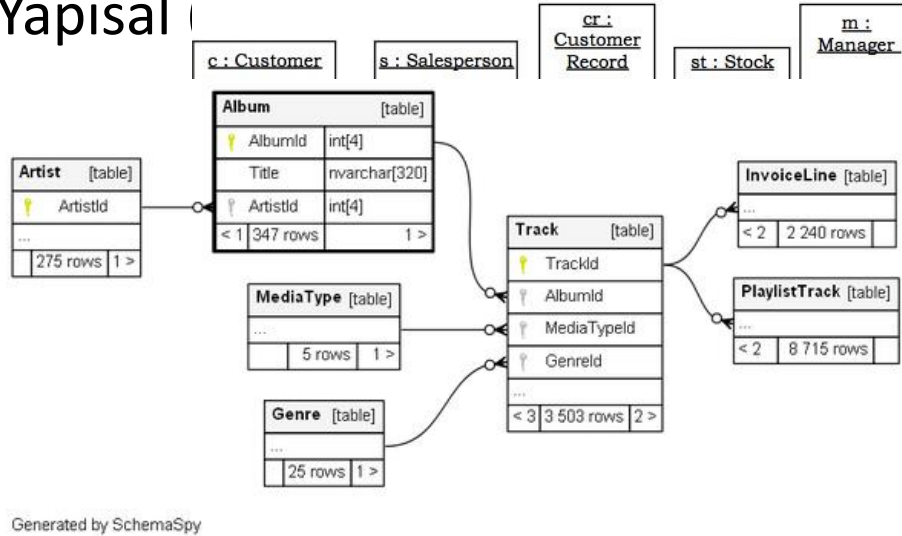
# Gereksinimlerin Geliştirilmesi



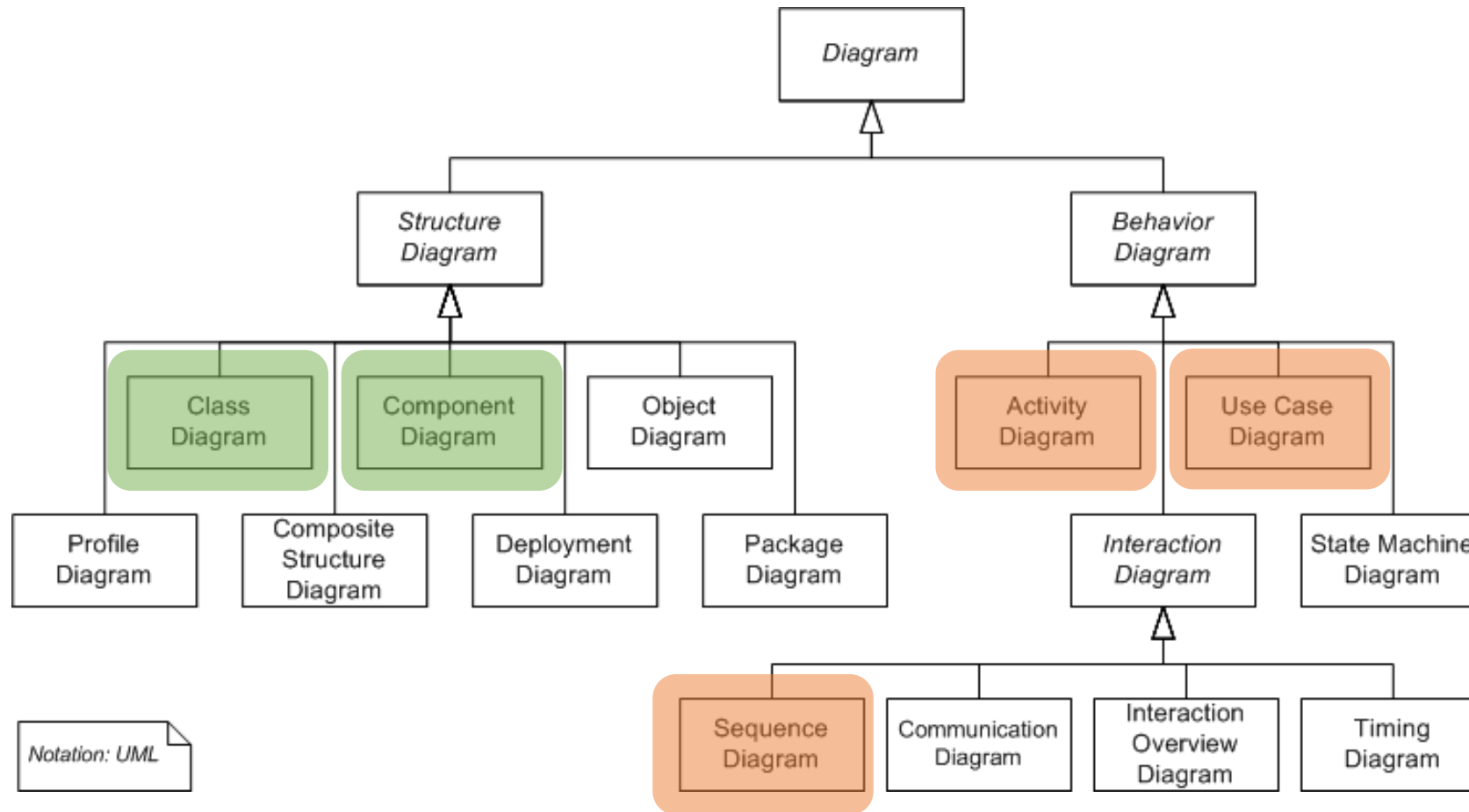
# UML (Unified Modelling Language)

Görsel ve metinsel gösterimlerle modelleme dili

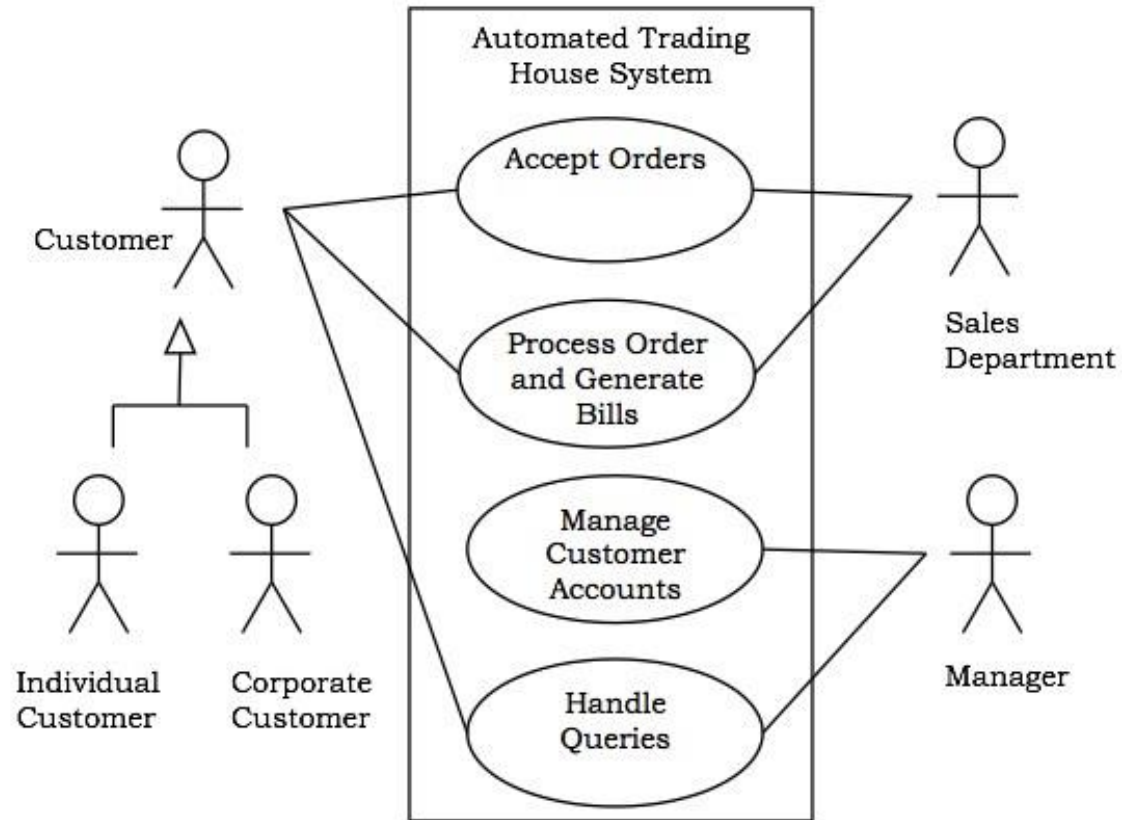
- Davranışsal (Behavioral) Diagramlar
- Yapısal (Structural) Diagramlar



# UML (Unified Modelling Language)



# Kullanım Durumu (Use Case) Diyagramları



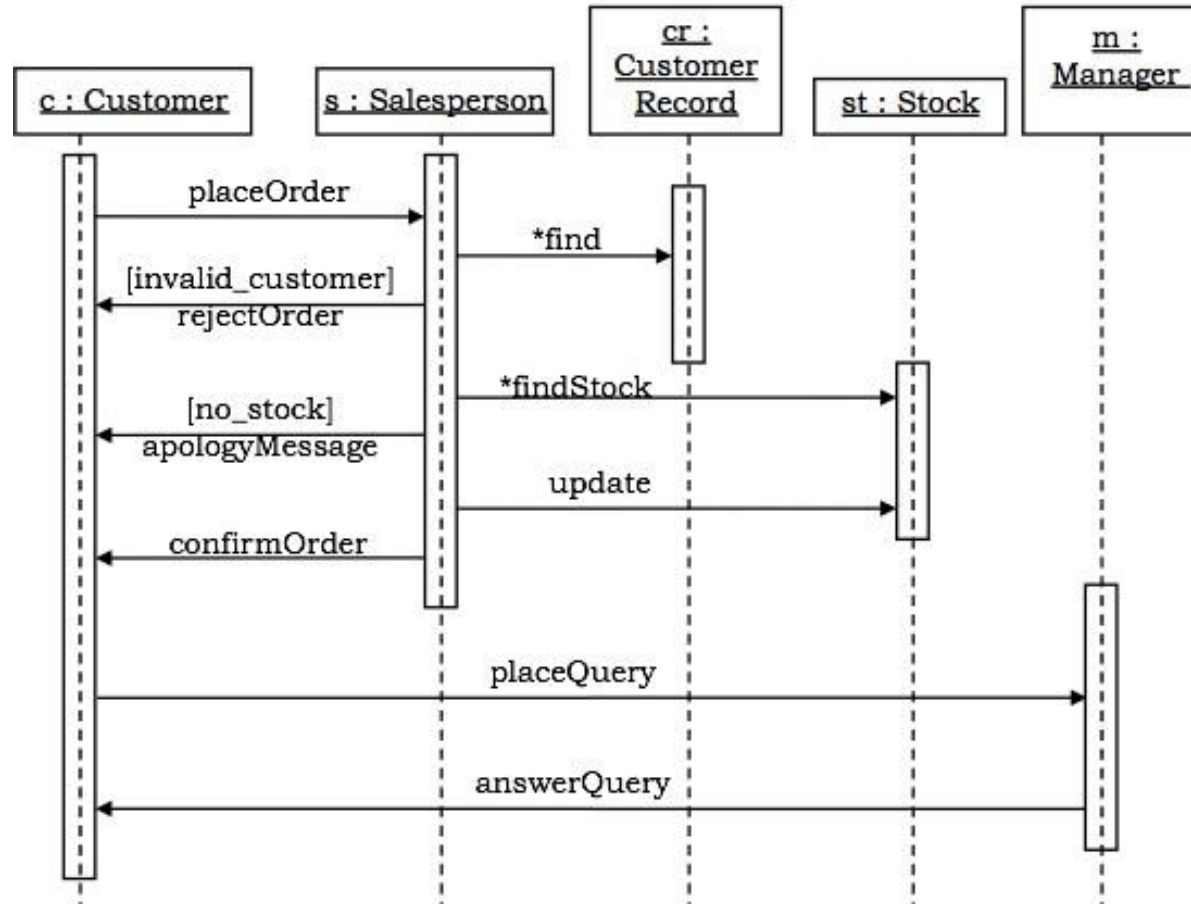
# Kullanım Durumu (Use Case) Senaryoları

Tanım
Aktörler
İlgili Gereksinim Maddeleri
Önkoşullar
Ana Senaryo
Alternatif Senaryo
İstisnai Durumlar
Son Durum
İlişkili İş Kuralları

## KDS 2 – Kullanım Senaryosunun Adı

Tanım	
Aktörler	
İlgili Gereksinim Maddeleri	
Önkoşullar	•
Ana Senaryo	1)
Alternatif Senaryo	
İstisnai Durumlar	
Son Durum	•
İlişkili İş Kuralları	•

# Dizi (Sequence) Diyagramı





# Kullanıcı Hikayeleri (User Story)

User Story Başlığı
Kısa Açıklaması
<Kullanıcı Grubu> olarak
... yapmak istiyorum
ki ... faydasını sağlayabileyim.
Kabul Kriterleri
... yaptığımda, ... şeklinde davranışın
Bağımlılıklar
Kısıtlar



# Diğer Gereksinim Geliştirme Teknikleri

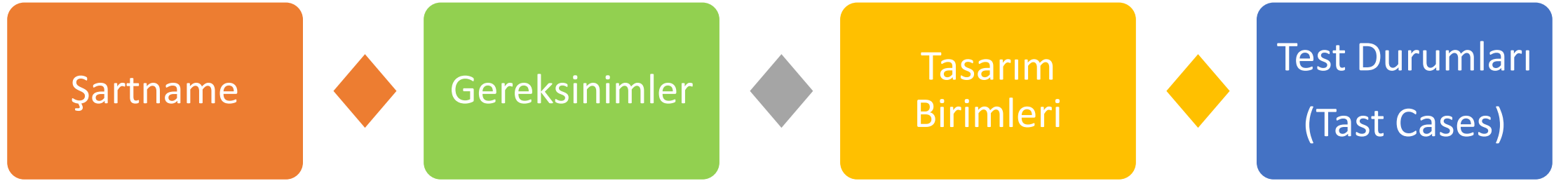
- SWOT (Strengths-Weaknesses-Opportunities-Threats) Analizi
- MOST (Mission, Objectives, Strategies, and Tactics) Analizi
- MoSCoW (Must Have, Should Have, Could Have and Won't Have)
- Zihin Haritalandırma
- Balık Kılıcı Metodu
- 5 Neden Metodu
- Süreç Modelleme
- Algoritmik Durum Makinaları
- ...

# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- **Gereksinimlerin İzlenmesi**
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# Gereksinimlerin İzlenmesi

## Çift Taraflı İzlenebilirlik



# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- **Gereksinimlerin Geçerlenmesi**
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# Gereksinimlerin Geerlenmesi

Taslak  
Ekranlar  
(Mock Up)

Storyboarding  
(Film Őeridi)

Simlasyon

Prototip  
OluŐturma

# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- **Değişiklik Yönetimi**
- Gereksinim Mühendisliği Nedir?

# Değişiklik Yönetimi

## DEĞİŞİKLİK İSTEKLERİ (Change Requests)

### Nedir?

Geçerlenmiş dokümanlara ya da ürünün kendine müşteri tarafından gelen DEĞİŞİKLİK isteği ya da YENİ ÖZELLİKLERdir.

Etki Analizi ve Konfigürasyon Yönetimi gerektirir.

Gerekirse  
“Değişiklik Kontrol Kurulu”  
kurulur.



# İçerik

- Kavramlar
- Paydaş Gereksinimleri Nasıl Toplanır?
- Gereksinimlerin Dokümantasyonu
- Paydaş Gereksinimlerinin Önceliklendirmesi
- Gereksinimlerin Geliştirilmesi
- Gereksinimlerin İzlenmesi
- Gereksinimlerin Geçerlenmesi
- Değişiklik Yönetimi
- Gereksinim Mühendisliği Nedir?

# Gereksinim Mühendisliđi Nedir?



Teşekkürler 😊