

# Yazılım Gereksinimleri Mühendisliđi

2024 Bahar  
Ders Notu 1

# Gereksinimlerin Önemi

- ❑Günümüzün rekabetçi ve hızlı sektöründe geliştirime takımlarının karşı karşıya oldukları zorluklar
  - ❖Sektördeki üst düzey problemleri içermeyen bir ürün geliştirilmesi için, sistem ya da yazılıma yönelik ayrıntılı bir gereksinimler nasıl belirlenecektir?
- ❑Herhangi bir karmaşık sistemin, yazılımın veya ürünün geliştirilmesine ilişkin, tüm paydaşların ihtiyaçlarının net bir şekilde anlaşılmasının hedeflenmesi
  - ❖Gereksinimlerin yazılmasına yönelik en iyi uygulamaların derinlemesine incelenmesi
- ❑ Ürünün geliştirilmesi sürecinde doğruluğu ve hızı arttırmada bazı yazılım geliştirme araçlarından ve bunların otomasyonundan yararlanılması

# Etkili (Effective) Gereksinimler Niçin Önemlidir?

- Paydaşlar arasında daha net ve daha etkili iletişim için daha iyi gereksinimler
  - ❖ Kaliteden ödün vermemek,
  - ❖ Tüm organizasyonun daha şeffaf olması, daha az yeniden işleme ihtiyacı ve hızlandırılmış geliştirmeye doğru yönelme.
- Hem bir sanat hem de bir bilim olarak gereksinimlerin yazılması
  - ❖ Bağlama göre değişebilme özelliği
- İdeal olarak her bir bireysel kullanıcı, iş ortamı (business) ve fonksiyonel gereksinim için:
  - ❖ Tam (*complete*), doğru (*correct*), uygulanabilir (*feasible*), gerekli (*necessary*), öncelikli (*prioritized*), belirli (*unambiguous*), doğrulanabilir (*verifiable*), tutarlı (*consistent*), değiştirilebilir olma (*modifiable*) ve izlenebilme (*traceable*)

## A- Gereksinimleri Belirlemenin Ortak Zorlukları #2

### Çok Fazla Ayrıntı, Çok Az Yenilik

- ❑ Takımlara kendilerini aşırı kısıtlacak çok fazla ayrıntılar verilmesi
- ❑ Takımların yenilik yapma özgürlüğünün çok az olması
  - ❖ Yeniden kullanılan önceki ürün parçalarından kaynaklanması
  - ❖ Müşteri gereksinimlerinin son derece ayrıntılı olduğu müşteri RFP lerini (Request for Proposals) içeren kısıtlayıcı teknik özellikler
    - ✓ Yukarıdakiler bazen u gerekli olsa da genellikle yeniliğe yol açmaz.
- ❑ Sonuç ise tasarımcıların sorunu yaratıcı biçimde çözebilmelerinin engellenmesi ve #2 ile sadece söyleneni yapma modelinin benimsenmesi

# A - Gereksinimleri Belirlemenin Ortak Zorlukları #1

## Analist/Kullanıcı İletişimsizliği

- Analistlerin, gerek ürün ile ilgili sunacaklarından gerekse son kullanıcının ihtiyaçlarından emin olmaması
- Analistlerin bakış açısına göre ürünün ayrıntıları için bazen müşterinin verdiği bilginin yetersizliği ve birçok sorunun cevapsız kalması
- Analistlerin ya başarılı olmayacak bir tasarım gerçekleştirmesi ya da ürün ile ilgili net cevapları olmayan sorular sorması

## A- Gereksinimleri Belirlemenin Ortak Zorlukları #3

### Doğru Gereksinim, Yanlış Ürün

- ❑ Geliştirilen ürünün orijinal gereksinimleri karşılamaması
- ❑ Ekip problem bildirimini toplamada ve kısıtlamaları çözmede başarılı
  - ❖ Ürünün bir şekilde başlangıçta çözülmeye çalışılanın dışında sonuçlanması
- ❑ Tasarımdaki zorluklarla çeşitli ödünler verilmesi ve ürünle ilgili spesifikasyonların da (belirtilerin de) yavaş yavaş "kabul edilebilir" ürün kapsamı dışına çıkması
- ❑ #3 senaryosunda ekip 'bir şey' oluşturmaya odaklandıkça, sona yaklaştıkça doğru 'bir şeyin' oluşturulamaması

# B-Gereksinimler Yazılırken Zorlukların Üstesinden Gelinmesi

## Çözülecek Problemin Anlaşılması

- ❑ Tüm paydaşların uyumlu olması
- ❑ problem bildirimlerinin oluşturulması
  - ❖ Kime ve nasıl katkı sağlanacaktır?
    - ✓ Diğer bir ifade ile bağlam (context) nedir?
  - ❖ Problemler nelerdir?
  - ❖ Problemler ne sıklıkla karşılaşılmaktadır?
- ❑ Açıklamalarla takıma bağlam sağlanması
  - ❖ Ayrıntılı gereksinimlerin toplanmasında orijinal probleme veya yerine getirilecek gereksinimlere kadar uzanabilme
- ❑ Henry Ford örneği (otomobilin icadı) #1 örneği olarak inovasyona açık belirleyici

# B- Gereksinimler Yazılırken Zorlukların Üstesinden Gelinmesi

## Gereksinim Hiyerarşisi ve Sınıflandırmasının Tanımlanması

- Problem açık olarak tanımlandıktan sonra, daha ayrıntılı (kapsamlı) gereksinimlerin belirlenmesi
  - ❖ Nerede tutulur?
    - ✓ Excel sayfası olarak ya da gereksinimleri veritabanı şeklinde
- Gereksinimlerin yönetimde belge kullanmak geliştirme sürecinde büyük risk
- Formal bir gereksinim yönetimi platformu izlenebilirliği (traceability) sağlama özelliği
  - ❖ Açık izlenebilirlikle (Clear traceability), formal bir değişiklik yönetimi sürecini (change management process) sürdürülebilmesi olanağı;
  - ❖ etki analizinin (impact analysis) gerçekleşmesi;
  - ❖ Süreçle ilgili karşılıklı bağımlılıkların ortaya çıkarılması ile doğru karar vericilerin zamanında ve kolaylıkla projeye eklenmesi
- Gereksinim yönetiminin bir bütün olarak geliştirilebilmesi için, geliştirme takımının net bir hiyerarşi ve sınıflandırma oluşturması
- Takımların gereksinimler ile tasarım arasındaki farkı anlaması



# Gereksinimler (Requirements)

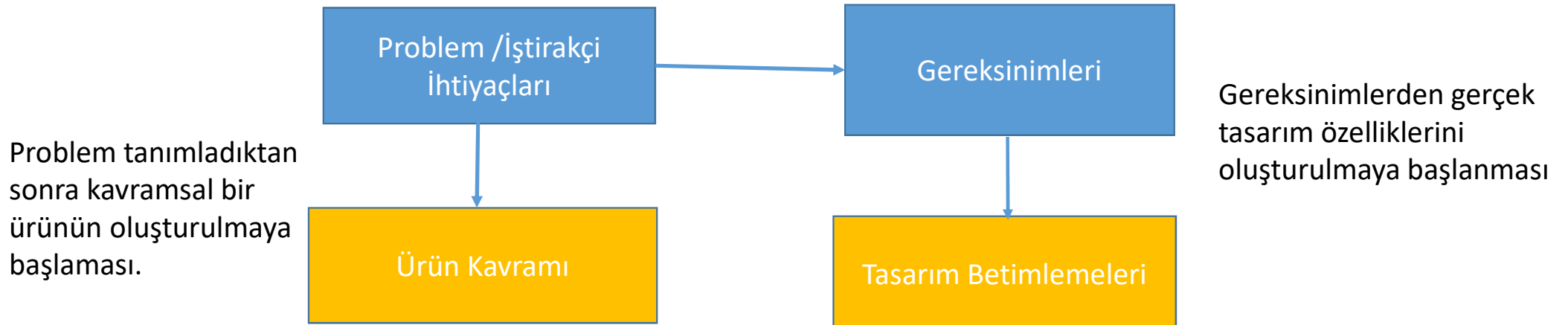
❑ **Gereksinim** ürünün, sistemin veya yazılımın ne yapması gerektiğinin bildirilmesi

❑ **Gereksinimler** geliştirilecek doğru ürününün (right product) belirlenmesi

&

ürünün doğru şekilde (right way) geliştirilmesi için kullanılan araç (tool)

## İzlenebilirliğin ve Hiyerarşinin Sınıflandırılmasının Önemi



# Tasarım (Design)

- ❑ Tasarım spesifikasyonu gereksinimlere yanıt verilmesi &
- ❑ Tasarım spesifikasyonu ürünün, sistemin veya yazılımın ne yaptığının belirtmesi

## Spesifikasyonların Etkileri

- ❑ Doğru olarak üretilecek ürünün belirlenmesinde etkili değil
- ❑ Geliştirilecek ürünün ne olduğunun ve nasıl çalıştığının anlaşılmasında etkili

## Tasarım Spesifikasyonları ile Gereksinimlerin Karıştırılması

- ❑ Geleneksel tasarım spesifikasyonlarının gereksinim olarak kullanılmasıyla problemlerle karşılaşılması
  - ❖ Örneğin aşırı kısıtlanmış tasarımcıların ortaya çıkışı

## Gereksinimlerde Amaç

Geliştirilecek ürünün tasarım spesifikasyonu sağlanmadan ya da herhangi bir kısıtlama olmadan ürün, sistem veya yazılım kullanıcılarının ve müşterilerin ihtiyaçlarının çıkartılması

## Gereksinim Örnekleri

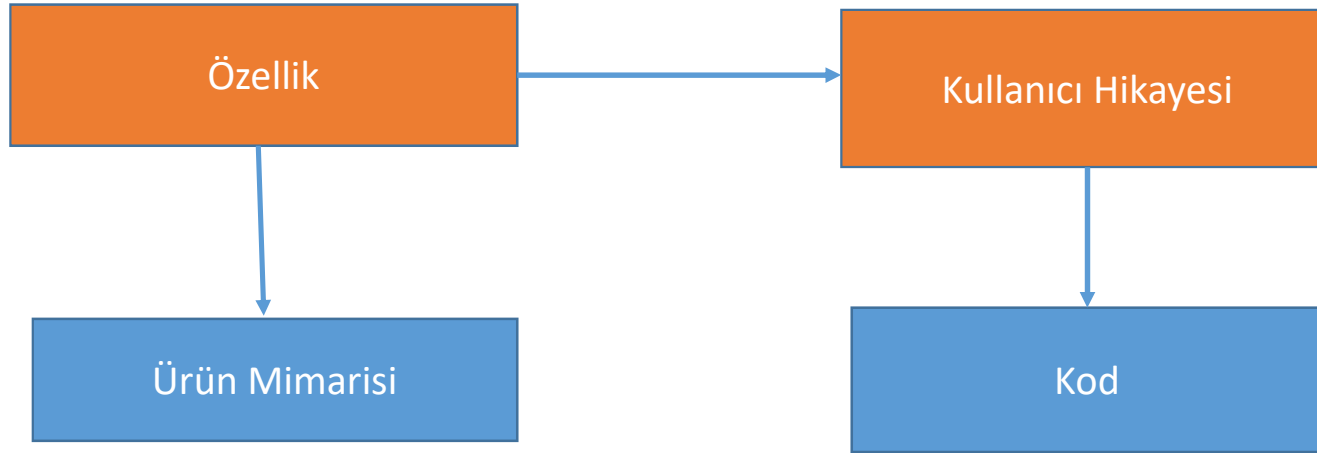
*Kötü ifade edilmiş gereksinim (açıklamada tasarım da yer almaktadır)*

Yazılım algoritması, her biri 100 desibel dinamik aralığa sahip iki ses sinyalini alacak ve bunları en az 140 desibelBdinamik aralığa sahip tek bir çıkış sinyalinde birleştirecektir; böylece güç tasarrufu sağlanabilecektir.

*İhtiyaçları gösteren iyi ifade edilmiş gereksinim (tasarım kısıtlaması olmadan)*

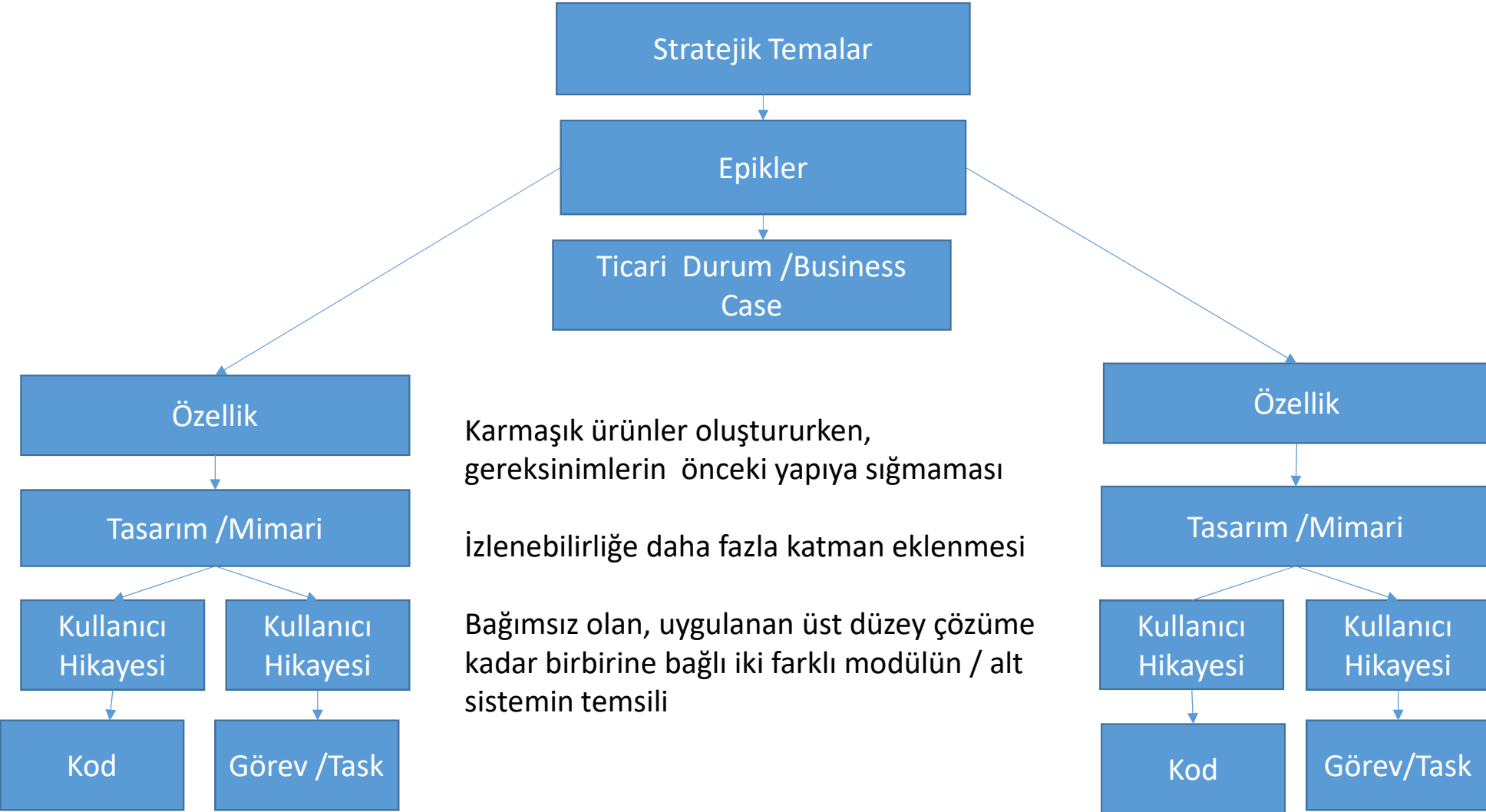
Geliştirilen çözüm, çalıştırma sırasında rakip cihazların endüstri standardının altında kalarak 20miliamperden az tüketecektir.

# Çevik Terimlerle Gereksinimlere Bakış

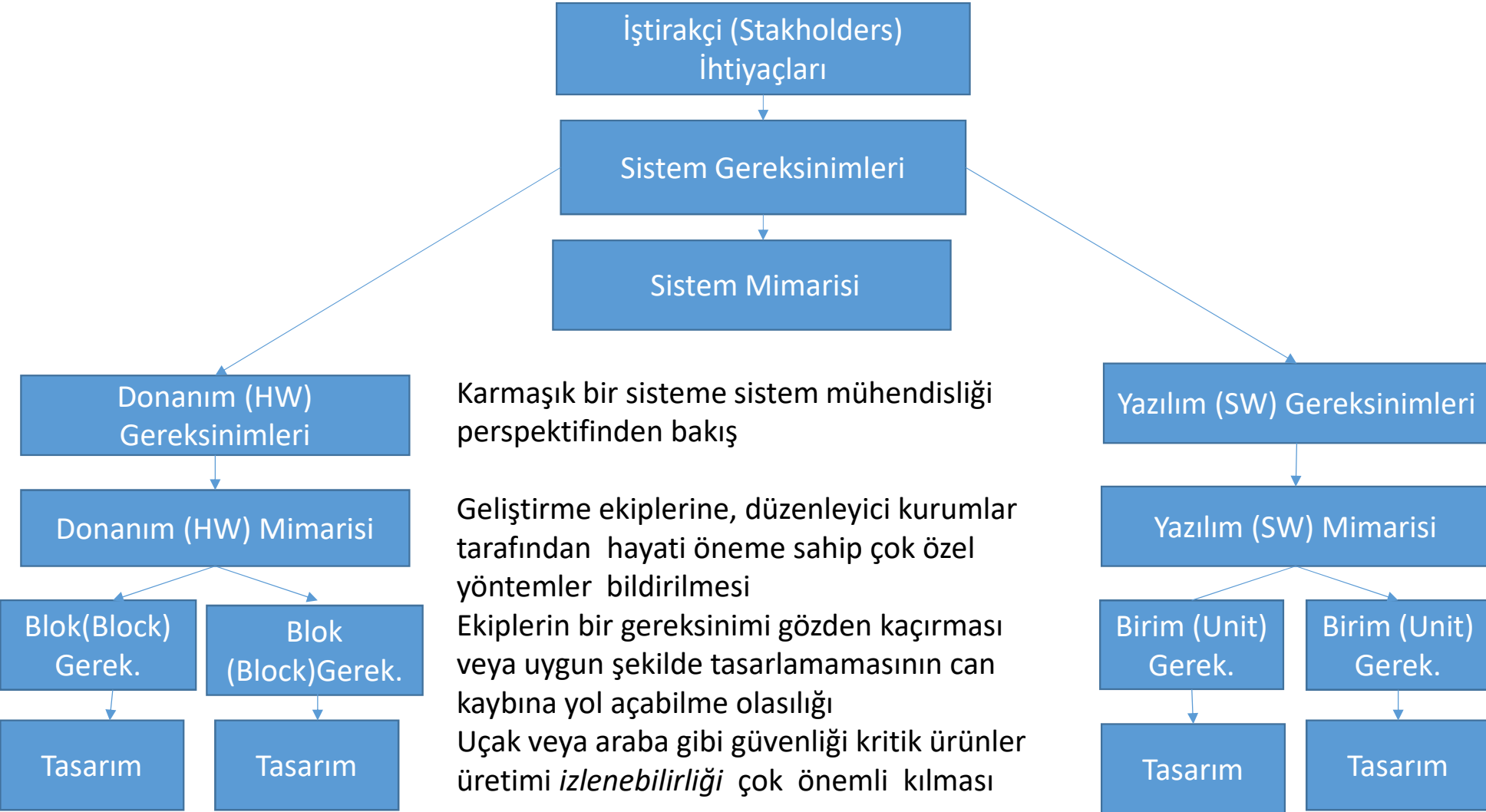


- ❑ Kullanıcı hikayelerine ayrıştırmak istenilen özelliklerin veya epiklerin bir listesinin yazılması
- ❑ İlkelerin değişmemesine rağmen anlambilim değişimi
- ❑ Dokümantasyonun nasıl ve ne zaman yapılacağına ilişkin süreçlerdeki farklılıklar

# Ölçeklenmiş Çevik (Scaled Agile)



# Karmaşık Yapıda Düzenlenmiş Sistemler



# İzlenebilirlik ile İletişim Boşluğunun Kapatılması

- ❑ Sektör veya metodoloji ne olursa olsun, gereksinimler izlenebilirliğini açıkça tanımlamalı ve gereksinimler ile tasarım arasındaki ayırım kavranmalı
- ❑ Çoğu zaman farklı ekiplerdekilerin aynı şeyi ifade eden farklı terimler kullanmaları
- ❑ Anlambilim konusunda net olunması, düzeylerin her birinin ne anlama geldiğini tanımlayacak bir sözlük hazırlanması
  - ❖ Sözlük, seviyelerin her birinin ne anlama geldiğini, süreçten ve sonuçlardan kimin sorumlu olduğunu tanımlamalı
  - ❖ Anlambilim, hem gereksinim yönetimi planında hem de gereksinim yönetimi aracında olmalı.
- ❑ Gereksinim seviyelerinde *izlenebilirlik* oluşturulduğunda, ihtiyaçların daha ayrıntılı doğrulama (*verification*) ve onaylama (*validation*) kapsamına sahip olarak daha fazla ayrıntılanabilmesi
- ❑ *İzlenebilirliğin* sağlanmasının, ihtiyacın belgelenmesini çözümün belgelenmesinden ayırabilmesi

## C- Gereksinimlerin Kalitesini Arttırmak

- Gereksinimleri belirlemede "altın kuralı", paydaşlarla açık ve etkili bir iletişim kurulması
- Mükemmel bir şekilde yazılmış gereksinim, eğer meslektaşlarla paylaşmıyor ve işbirliği yapılmıyorsa sorunludur
- Gereksinimleri alıp tasarlanmış bir çözüme dönüştürecek ekip bu sürecin bir parçası olmalı
- İyi yazılmış (well-written) gereksinimler hem bilimin hem de sanatın bir karışımıdır.



# Etkili Gereksinimlerin Özellikleri #1

- Necessary** Each requirement should be necessary or required
- Her gereksinime, paydaşların gerçekten ihtiyacı olmalı
- Her gereksinim harici (external) bir sistem gereksinimini veya standartını belgelemeli
- Her gereksinim, gereksinimleri belirleme yetkisine sahip bir kaynaktan gelmeli

## Etkili Gereksinimlerin Özellikleri #2

- ❑ **Unambiguous** Each requirement should be simple, concise, and clearly defined.
- ❑ Bir gereksinim ifadesi tüm okuyuculara tek ve tutarlı bir yorum verebilmeli
  - ❖ Doğal dil belirsizliğe çok açık
- ❑ Gereksinimler kullanıcı alanına uygun, basit, kısa ve anlaşılır bir dille yazılmalı
- ❑ Tüm özel terimler bir sözlükte tanımlanmalı

**Örnek:** Ambiguous / Belirsiz : "The car shall be blue"

Unambiguous / Belirli : "The car shall be sky blue (#7BAFD4)"

# Etkili Gereksinimlerin Özellikleri #3

**Feasible** Each requirement must be possible to implement

- ❑ Her gereksinim, sistemin ve işletim ortamının bilinen yetenekleri ve sınırlamalarına göre uygulanmalı
- ❑ Ulaşılabilir gereksinimler kesinlikle belirlenmemeli
  - ❖ Gereksinimlerin çıkarılması sürecince tasarımcının / geliştiricinin iş analistleriyle birlikte çalışması
- ❑ Teknik olarak neyin yapılabileceği / yapılamayacağı ve neyin aşırı maliyetle yapılabileceği gerçeklerinin kontrolü
- ❑ Gereksinimin fizibilitesi değerlendirme yolları: Artımlı geliştirme yaklaşımları ve kavram kanıtlama prototipleri

# Etkili Gereksinimlerin Özellikleri #4

**Verifiable** Each requirement should be measurable or testable

- Bir gereksinim doğrulanabilmeli
  - ❖ Eğer doğrulanamıyorsa, bu gereksinimin doğru bir şekilde uygulanıp uygulanmadığının belirlenmesi objektif olmaz, sadece bir fikir belirtir.
- Eksik (incomplete), tutarsız (inconsistence), uygulanabilir olmayan (infeasible) veya belirsiz (ambiguous) gereksinimler de doğrulanamaz (unverifiable).

**Örnek:** Hatalı: : “The car must run quickly”

Doğru: : “The car must be able to reach 100 mph in less than 10 seconds”

- Zarf yapısındaki sözcüklerden kaçınmak zorunlu
- Zarflar test edilemez.
- Nitelikli bir hedef eklenmeli veya gereksinim bildirimini ölçülebilir veya test edilebilir olmalıdır.

## Etkili Gereksinimlerin Özellikleri #5

**Correct** Each requirement must accurately describe the functionality to be built

Doğruluk (correctness) referansı, gerçek kullanıcı veya üst düzey sistem gereksinimi gibi gereksinimin kaynağıdır.

Ana sistem gereksinimiyle çelişen bir yazılım gereksinimi doğru değildir.

Gereksinimlerin doğruluğunu yalnızca ürün kullanıcıları belirleyebilir.