

Doç. Dr. Sedat Akleylek

([sedat.akleylek@omu.edu.tr](mailto:sedat.akleylek@omu.edu.tr)) tarafından

hazırlanan bu videonun 5846 sayılı Fikir ve Sanat Eserleri Kanunu kapsamında başka bir yolla yayımlanmasına izin verilmemektedir.

Başka bir ifadeyle, **bu videonun izin alınmaksızın BİL 407 Yazılım Mühendisliği dersine kayıtlı olmayan kişiler ile paylaşılması, OMÜ ortamı dışına taşınması, paylaşılması veya saklanması yasaktır.**

# BİL 407 Yazılım Mühendisliği

## Hafta 6

Doç. Dr. Sedat Akleylek

[sedat.akleylek@bil.omu.edu.tr](mailto:sedat.akleylek@bil.omu.edu.tr)

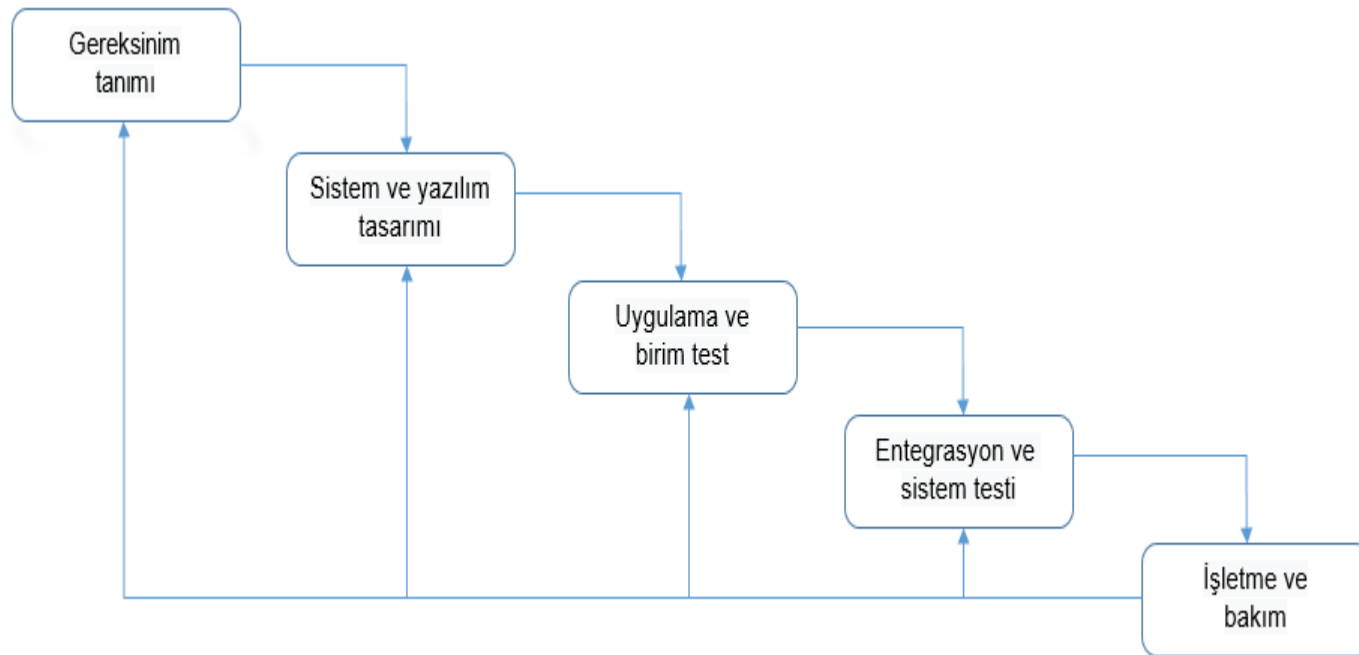
# İçerik

- Büyük Proje Geliştirme
- Gereksinim Mühendisliği
- İşlevsel ve İşlevsel Olmayan Gereksinimler

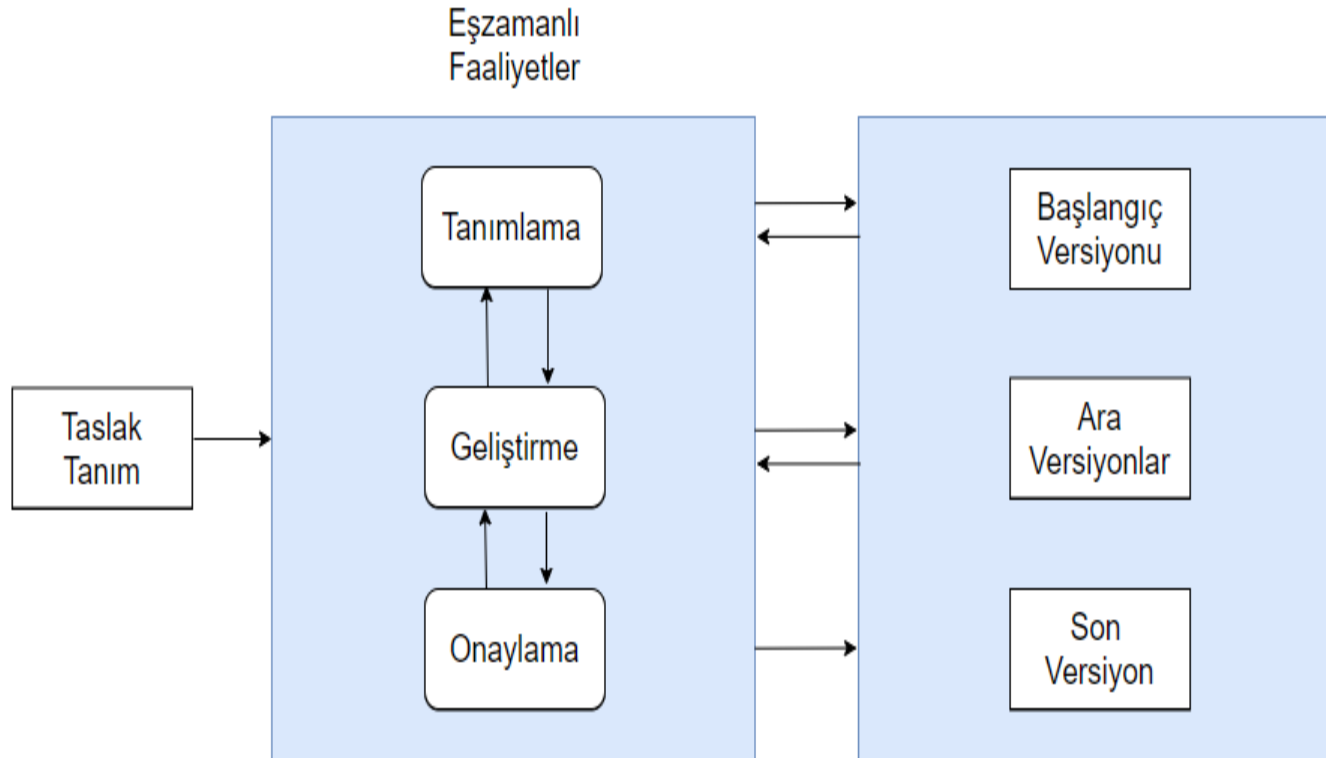
# Kaynaklar

- *Ian Sommerville, Software Engineering, 10th Edition, Addison-Wesley, 2015.*
- *SWEBOK, Guide to the Software Engineering Body of Knowledge: 2004 Version, IEEE.*
- <https://www.youtube.com/user/SoftwareEngBook>
- IEEE-Software Requirements Specification Template

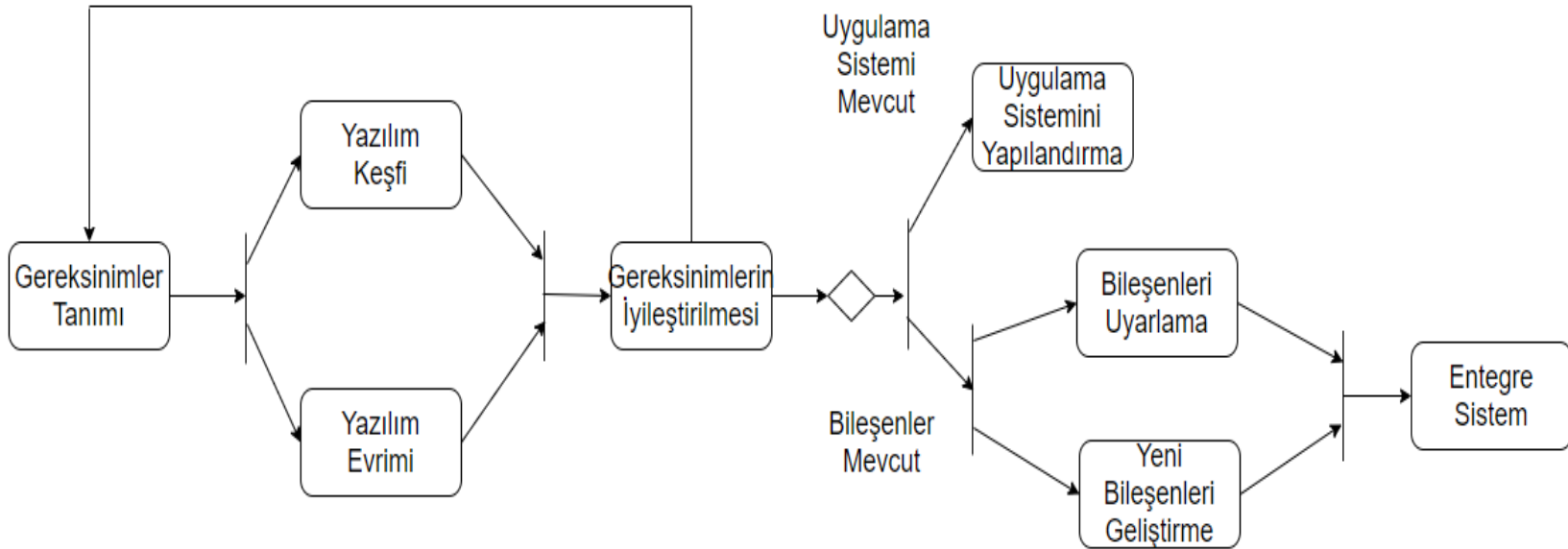
# Şelale (Waterfall) Modeli



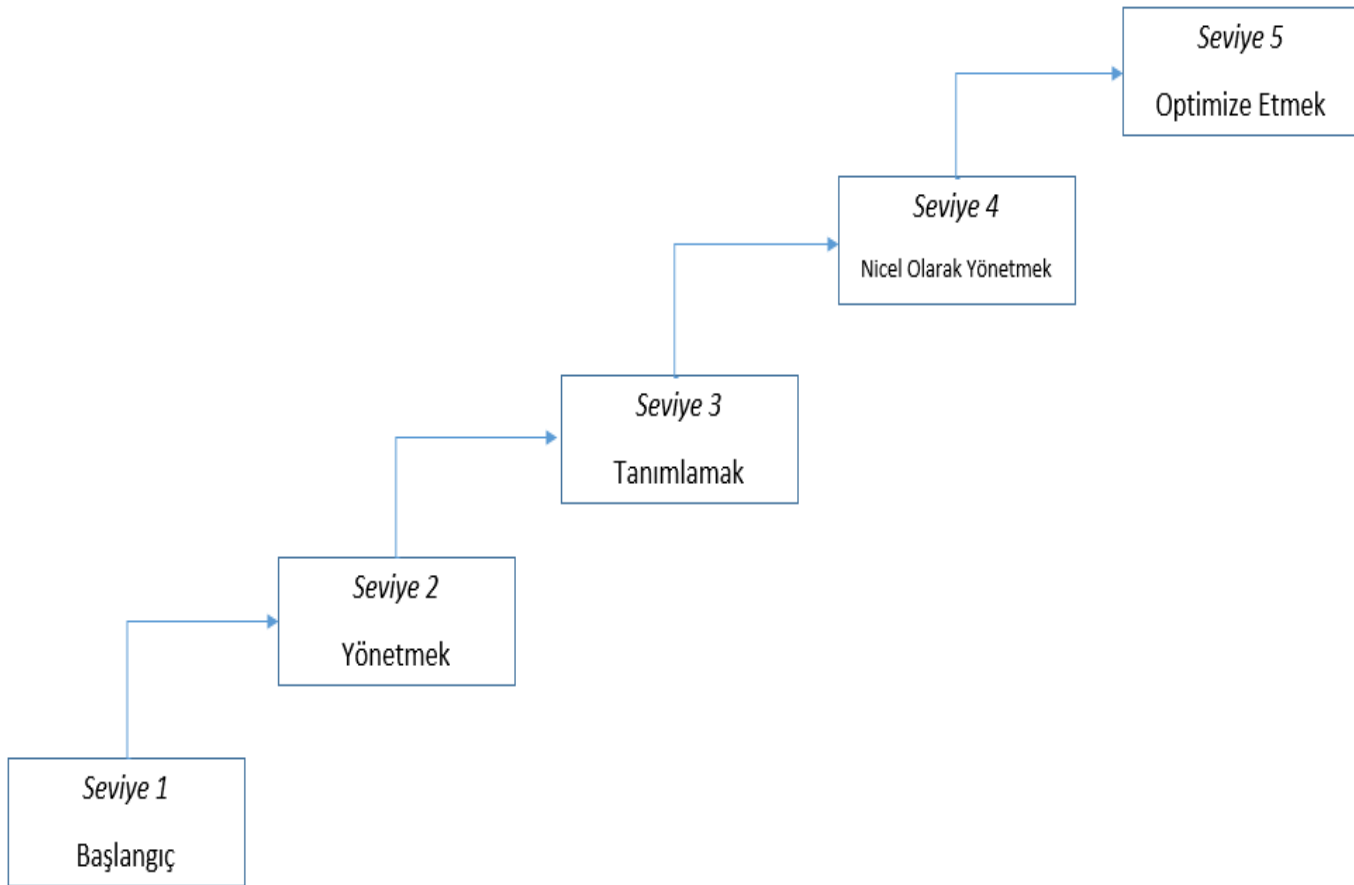
# Arttırımlı (Incremental) Geliştirme



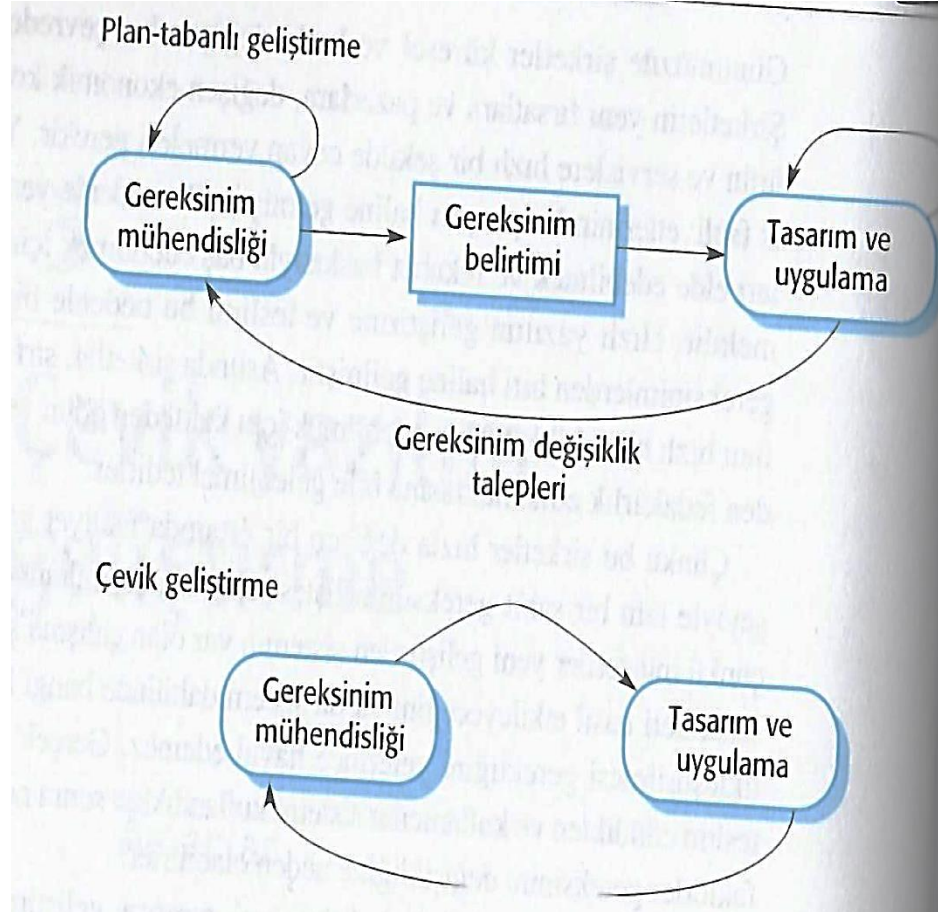
# Yeniden Kullanım Odaklı Yazılım Mühendisliği



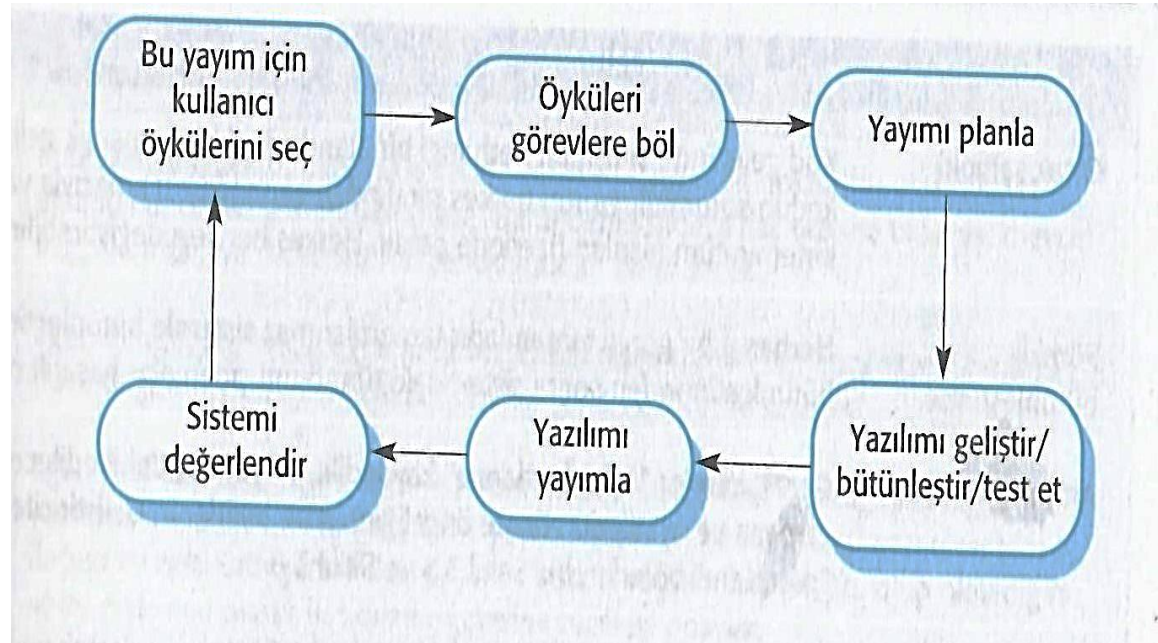
# Yetenek Olgunluk Seviyeleri



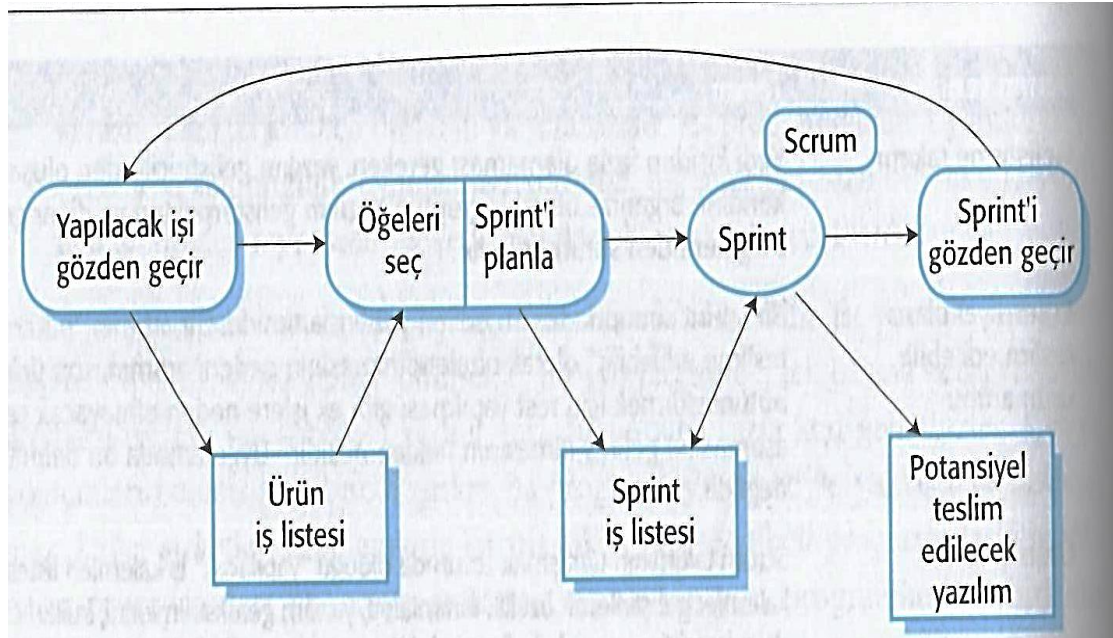
# Plan-Güdümlü ve Çevik Geliştirme



# Uç Yayım Döngüsü



# Scrum Sprint Döngüsü



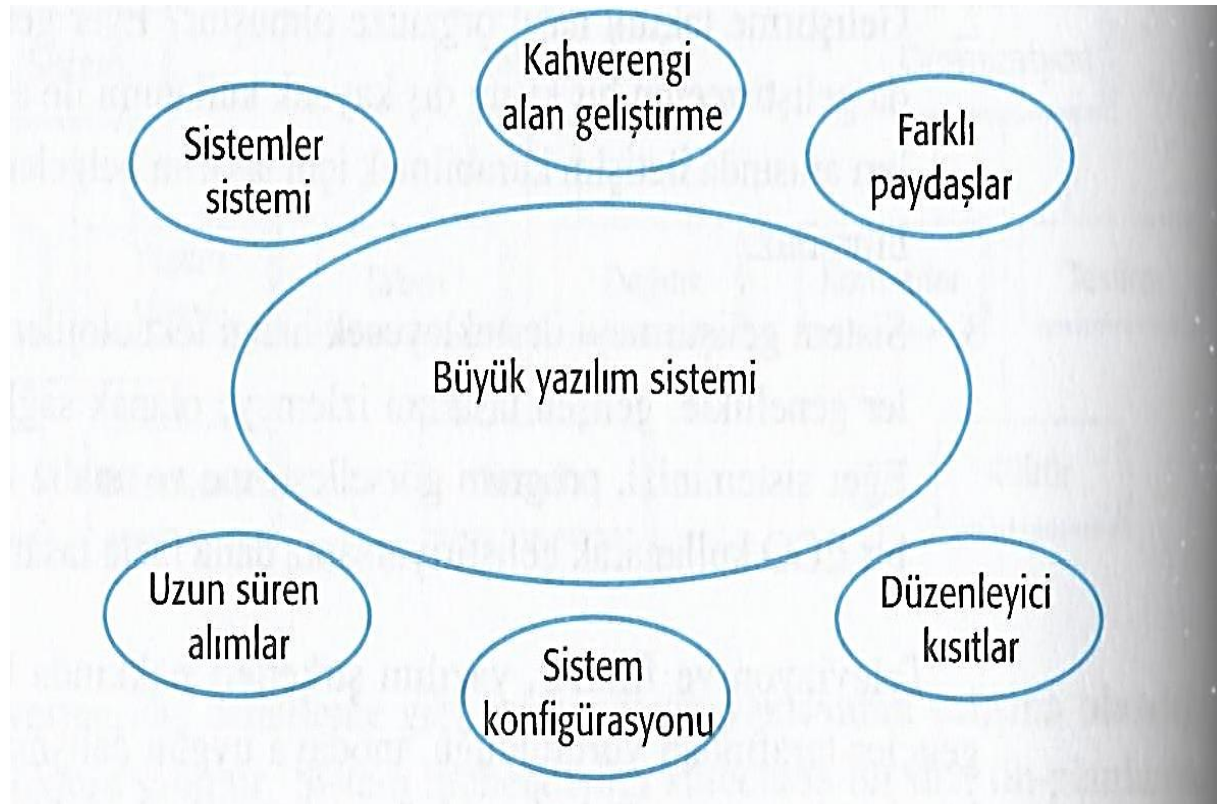
# Büyük Sistemler İçin Çevik Yöntemler

- Büyük sistemler genellikle, her sistemi ayrı ekiplerin geliştirdiği ayrı, iletişim halindeki sistemlerin birleşimleridir. Sıklıkla bu ekipler farklı yerlerde, bazen farklı saat dilimlerinde çalışırlar.
- Büyük sistemler, "**terkedilmiş sistemleridir**", yani bir dizi mevcut sistemi içerir ve bunlarla etkileşime girer. Sistem gereksinimlerinin çoğu bu etkileşimle ilgilidir ve bu nedenle esneklik ve aşamalı gelişime gerçekten katkıda bulunmazlar.
- Bir sistem oluşturmak için birkaç sistemin entegre edildiği yerlerde, geliştirmenin önemli bir kısmı, orijinal kod geliştirmeden ziyade sistem yapılandırmasıyla ilgilidir.

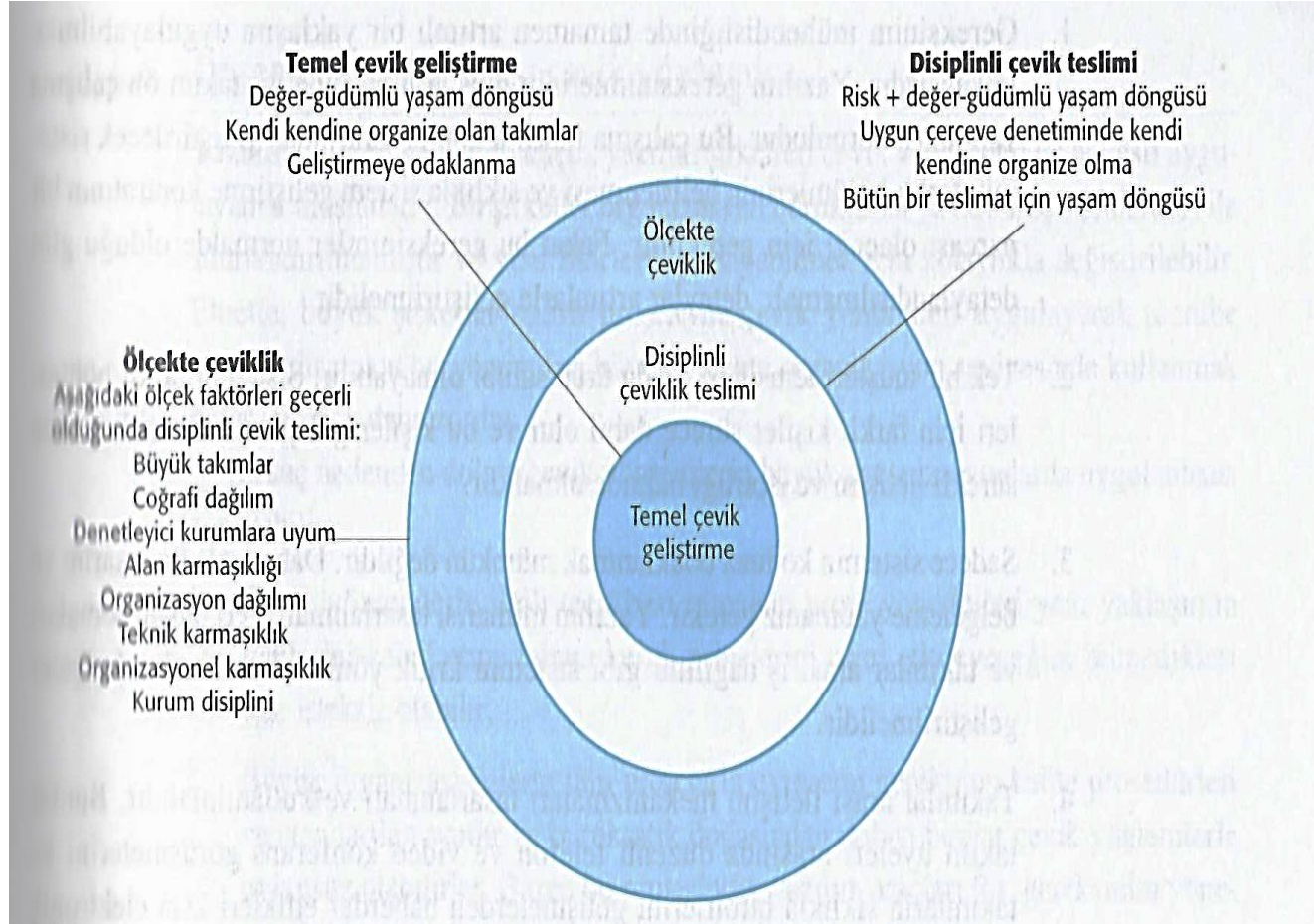
## Büyük Proje Geliştirme

- Büyük sistemler ve geliştirme süreçleri, genellikle geliştirilme şekillerini sınırlayan dış kurallar ve düzenlemelerle sınırlandırılır.
- Büyük sistemler uzun bir tedarik ve geliştirme süresine sahiptir. İnsanların kaçınılmaz olarak başka işlere ve projelere geçmesi nedeniyle, o dönem boyunca sistemi bilen uyumlu ekipler oluşturmak zordur.
- Büyük sistemler genellikle çeşitli paydaşlara sahiptir. Tüm bu farklı paydaşları geliştirme sürecine dahil etmek neredeyse imkansızdır.

# Büyük Proje Özellikleri



# IBM'in Çeviklik Ölçekleme Modeli



# Büyük Sistemlere Ölçekleme

- Gereksinim mühendisliğine tamamen artımlı bir yaklaşım imkansızdır.
- Tek bir ürün sahibi veya müşteri temsilcisi olamaz.
- Büyük sistem geliştirme için sadece sistemin koduna odaklanmak mümkün değildir.
- Ekipler arası iletişim mekanizmaları tasarlanmalı ve kullanılmalıdır
- **Sürekli entegrasyon** pratik olarak imkansızdır. Ancak, sık sistem kurulumlarının ve sistemin düzenli sürümlerinin sürdürülmesi çok önemlidir.

# Çok-Takımlı Scrum

- **Rol çoğaltma:**
  - Her takımın kendi çalışma bileşeni ve Scrum uzmanı için bir Ürün Sahibi vardır.
- **Ürün mimarları:**
  - Her ekip bir ürün mimarı seçer ve bu mimarlar genel sistem mimarisini tasarlamak ve geliştirmek için işbirliği yapar.
- **Hizalamayı serbest bırak:**
  - Her takımın ürün çıkış tarihleri, kanıtlanabilir ve eksiksiz bir sistem üretilecek şekilde hizalanır.
- **Scrum'ların Scrum'ı:**
  - Her takımdan temsilcilerin ilerlemeyi tartışmak ve yapılacak işi planlamak için bir araya geldiği günlük bir Scrum'ların Scrum'ı vardır.

# Kuruluşlar Arasında Çevik Yöntemler

- Çevik yöntemlerle ilgili deneyimi olmayan proje yöneticileri, yeni bir yaklaşımın riskini kabul etme konusunda isteksiz olabilir.
- Büyük kuruluşlar, genellikle tüm projelerin uyması beklenen kalite prosedürlerine ve standartlarına sahiptir ve bürokratik yapıları nedeniyle, bunların çevik yöntemlerle **uyumsuz** olması muhtemeldir.
- Çevik yöntemler, ekip üyeleri nispeten yüksek bir beceri düzeyine sahip olduğunda en iyi şekilde çalışır. Bununla birlikte, büyük kuruluşlarda, çok çeşitli beceriler ve yetenekler olması muhtemeldir.
- Özellikle geleneksel sistem mühendisliği süreçlerini kullanma konusunda uzun bir geçmişe sahip olan kuruluşlarda, çevik yöntemlere kültürel direnç olabilir.

# Önemli Noktalar

- Çevik yöntemler, hızlı yazılım geliştirmeye, yazılımın sık sürümlerine, dokümantasyonu en aza indirerek ve yüksek kaliteli kod üreterek işlem giderlerini azaltmaya odaklanan artımlı geliştirme yöntemleridir.
- Çevik geliştirme uygulamaları şunları içerir:
  - Sistem özellikleri için kullanıcı hikayeleri
  - Yazılımın sık yayınlanması,
  - Sürekli yazılım geliştirme
  - Test-Önce geliştirme
  - Geliştirme ekibine müşteri katılımı.

# Önemli Noktalar

- Scrum, bir proje yönetimi çerçevesi sağlayan çevik bir yöntemdir.
  - Bir sistem artışı geliştirildiğinde sabit zaman periyotları olan bir dizi sprint etrafında ortalınır.
- Birçok pratik geliştirme yöntemi, plan-güdümlü ve çevik geliştirmenin bir karışımıdır.
- Büyük sistemler için çevik yöntemleri ölçeklendirmek zordur.
  - Büyük sistemler, ön tasarıma ihtiyaç duyar ve bazı dokümantasyon ve organizasyonel uygulamalar, çevik yaklaşımlarının gayri resmi olmasıyla çelişebilir.

# Gereksinim Mühendisliği

- Müşterinin bir sistemden ihtiyaç duyduğu hizmetleri ve onun altında çalıştığı ve geliştirildiği kısıtlamaları kurma süreci.
- Sistem gereksinimleri, gereksinim mühendisliği sürecinde ortaya çıkan sistem hizmetleri ve kısıtlamaların açıklamalarıdır.

# Gereksinim Nedir ?

- Bir hizmetin veya bir sistem kısıtlamasının yüksek seviyeli soyut bir ifadesinden ayrıntılı bir matematiksel fonksiyonel tanımına kadar değişebilir.
- Gereksinimler ikili bir fonksiyona hizmet edebileceğinden bu kaçınılmazdır
  - Bir sözleşme teklifinin temeli olabilir - bu nedenle yoruma açık olmalıdır;
  - Sözleşmenin kendisinin temeli olabilir - bu nedenle ayrıntılı olarak tanımlanmalıdır;
  - Her iki ifade de gereksinimler olarak adlandırılabilir.

# Gereksinim Türleri

- Kullanıcı gereksinimleri
  - Doğal dildeki ifadeler ve sistemin sağladığı hizmetlerin şemaları ve operasyonel kısıtlamaları. Müşteriler için yazılmıştır.
- Sistem gereksinimleri
  - Sistemin fonksiyonlarının, hizmetlerinin ve operasyonel kısıtlamaların ayrıntılı tanımlarını gösteren yapılandırılmış bir belge. Neyin uygulanması gerektiğini tanımlar, bu nedenle müşteri ve yüklenici arasındaki bir sözleşmenin parçası olabilir.

# Kullanıcı ve Sistem Gereksinimleri

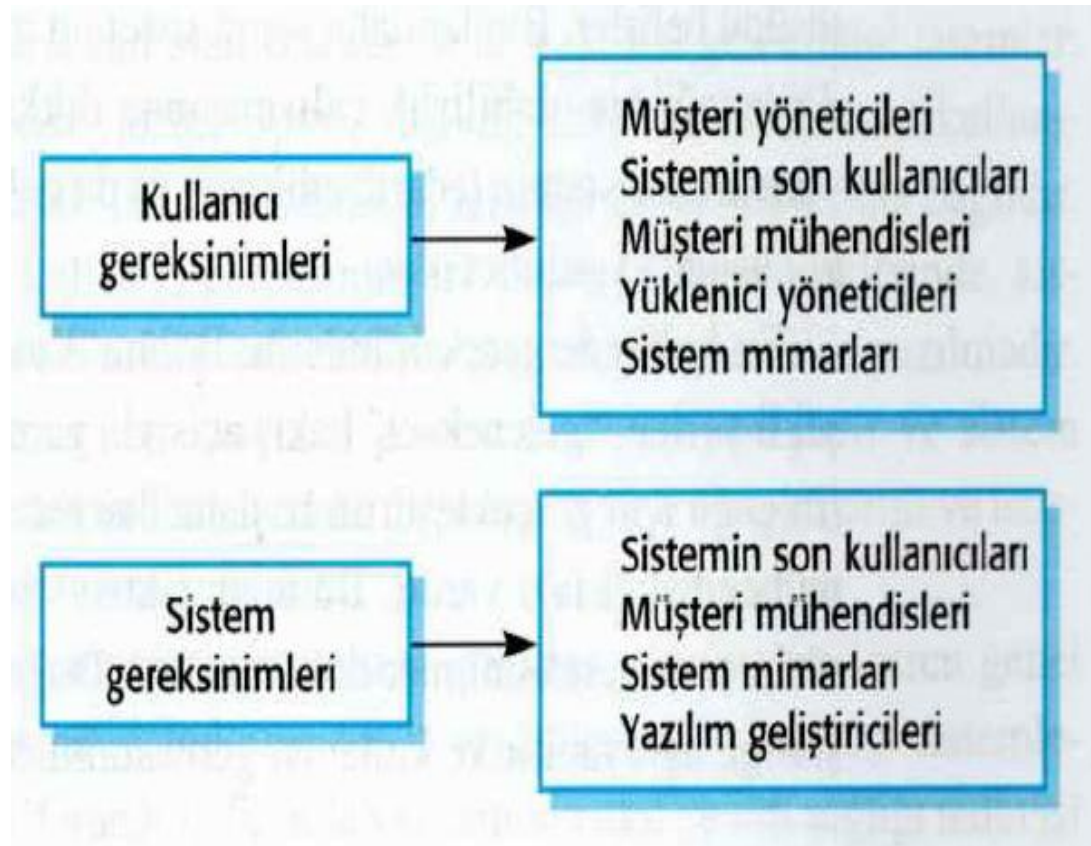
## Kullanıcı gereksinimleri tanımı

- 1.** ZihinSaS sistemi her ay her klinik için, o ay, o klinikte reçetelenen ilaçların maliyetini gösteren aylık yönetim raporları üretmelidir.

## Sistem gereksinimleri tanımı

- 1.1** Her ayın son çalışma gününde, reçetelenen ilaçların, maliyetlerinin, hangi klinik tarafından reçetelendiklerinin gösterildiği özet rapor hazırlanmalıdır.
- 1.2** Sistem raporu basılması için ayın son çalışma gününde saat 17.30'dan sonra hazırlanmalıdır.
- 1.3** Rapor her klinik için hazırlanmalıdır ve tek tek ilaç isimlerini, toplam reçete sayısını, reçetelenen doz sayısını ve reçetelenen ilaçların toplam maliyetini listelemelidir.
- 1.4** Eğer ilaçlar farklı dozlarda bulunabiliyorsa (10 mg'lık, 20 mg'lık gibi) her doz birimi için ayrı rapor hazırlanmalıdır.
- 1.5** İlaç maliyet raporlarına erişim yönetim erişim kontrol listesinde gösterilen yetkili kullanıcılar ile sınırlı olmalıdır.

# Farklı Gereksinim Tanımlamalarının Okuyucuları



# Sistem Paydaşları

- Sistemden bir şekilde etkilenen ve dolayısıyla meşru menfaati olan herhangi bir kişi veya kuruluş
- Paydaş türleri
  - Son kullanıcılar
  - Sistem yöneticileri
  - Sistem sahipleri
  - Dış paydaşlar

# Mentcare Sistemindeki Paydaşlar

- Bilgileri sisteme kaydedilen hastalar.
- Hastaların değerlendirilmesi ve tedavisinden sorumlu doktorlar.
- Doktorlarla muayeneleri koordine eden ve bazı tedavileri uygulayan hemşireler.
- Hastaların randevularını yöneten tıbbi resepsiyon görevlileri.
- Sistemin kurulmasından ve bakımından sorumlu IT (bilişim) personeli.

# Mentcare Sistemindeki Paydaşlar

- Sistemin hasta bakımı için mevcut etik kurallara uygun olmasını sağlamalı bir tıbbi etik yöneticisi.
- Sistemden yönetim bilgilerini alan sağlık yöneticileri.
- Sistem bilgilerinin muhafaza edilmesini ve korunmasını sağlamaktan ve kayıt tutma prosedürlerinin uygun şekilde uygulanmasını sağlamaktan sorumlu tıbbi kayıt personeli.

# Fonksiyonel ve Fonksiyonel Olmayan Gereksinimler

- Fonksiyonel gereksinimler
  - Sistemin sunması gereken hizmetlerin beyanları, sistemin belirli girdilere nasıl tepki vermesi gerektiği ve sistemin belirli durumlarda nasıl davranması gerektiği.
  - Sistemin ne yapmaması gerektiğini belirtebilir.
- Fonksiyonel olmayan gereksinimler
  - Zamanlama kısıtlamaları, geliştirme sürecindeki kısıtlamalar, standartlar vb. gibi sistem tarafından sunulan hizmetler veya fonksiyonlar üzerindeki kısıtlamalar.
  - Genellikle, tek tek özellikler veya hizmetler yerine bir bütün olarak sisteme uygulanır.
- Alan gereksinimleri
  - Operasyon alanından sistemdeki kısıtlamalar

# Fonksiyonel (İşlevsel) Gereksinimler

- Fonksiyonelliği veya sistem hizmetlerini açıklayın.
- Yazılımın türüne, beklenen kullanıcılara ve yazılımın kullanıldığı sistem türüne bağlıdır.
- Fonksiyonel kullanıcı gereksinimleri, sistemin ne yapması gerektiğine dair üst düzey ifadeler olabilir.
- Fonksiyonel sistem gereksinimleri, sistem hizmetlerini ayrıntılı olarak açıklamalıdır.

# **Mentcare Sistem: Fonksiyonel Gereksinimler**

- Bir kullanıcı, tüm klinikler için randevu listelerini arayabilecektir.
- Sistem, her klinik için her gün, o gün randevulara gelmesi beklenen hastaların bir listesini oluşturacaktır.
- Sistemi kullanan her personel, 8 basamaklı çalışan numarasıyla benzersiz bir şekilde tanımlanacaktır.

# Gereksinimlerdeki Belirsizlikler

- Fonksiyonel gereksinimler tam olarak belirtilmediğinde sorunlar ortaya çıkar.
- Belirsiz gereksinimler, geliştiriciler ve kullanıcılar tarafından farklı şekillerde yorumlanabilir.
- 1. gereksinimdeki "arama" terimini düşünün
  - Kullanıcının amacı - tüm kliniklerdeki tüm randevularda bir hasta adı aramak;
  - Geliştirici (yazılımcı) yorumu - bireysel bir klinikte bir hasta adı aramak. Kullanıcı kliniği seçer ve ardından arar.

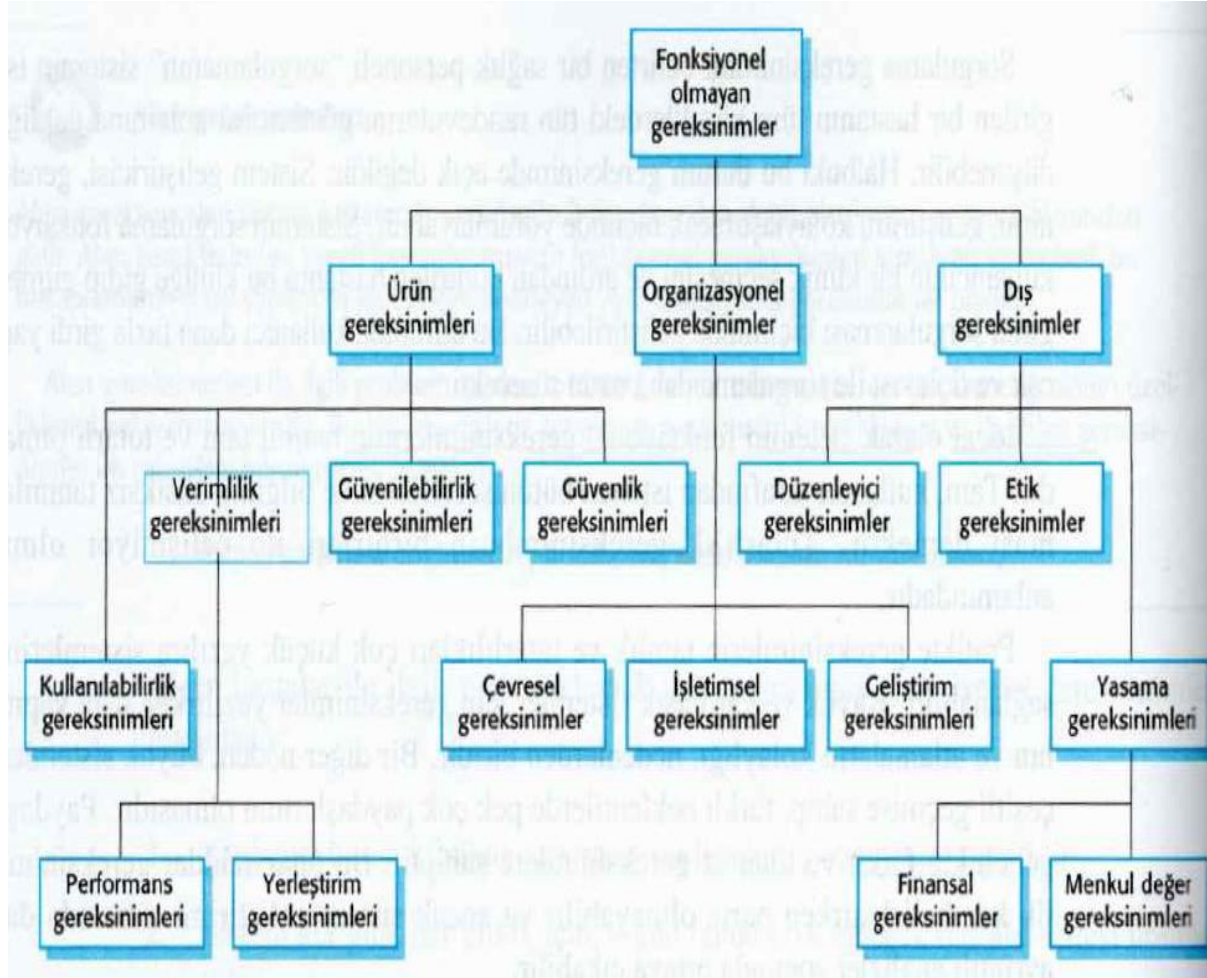
# Gereksinimlerin Eksiksizliği ve Tutarlılığı

- Prensipte olarak, gereksinimler hem eksiksiz hem de tutarlı olmalıdır.
- Eksiksiz (tam)
  - Gerekli tüm tesislerin açıklamalarını içermelidir.
- Tutarlı
  - Sistem tesislerinin açıklamalarında herhangi bir çelişki veya tezat olmamalıdır.
- Uygulamada, sistem ve çevresel karmaşıklık nedeniyle, eksiksiz ve tutarlı bir gereksinimler belgesi oluşturmak imkansızdır.

# Fonksiyonel Olmayan Gereksinimler

- Bunlar, sistem özelliklerini ve kısıtlamaları tanımlar, örn. güvenilirlik, yanıt süresi ve depolama gereksinimleri. Kısıtlamalar, G / Ç cihaz kapasitesi, sistem gösterimleri vb.
- Süreç gereksinimleri, belirli bir IDE, programlama dili veya geliştirme yöntemini zorunlu kılarak da belirtilebilir.
- Fonksiyonel olmayan gereksinimler, fonksiyonel gereksinimlerden daha kritik olabilir. Bunlar yerine getirilmezse, sistem işe yaramayabilir.

# Fonksiyonel Olmayan Gereksinimlerin Türleri



# Mentcare Sistemindeki Fonksiyonel Olmayan Gereksinimlerin Örnekleri

## **Ürün Gereksinimi**

Mentcare sistemi, normal çalışma saatleri boyunca tüm klinikler için mevcut olacaktır (Pzt - Cum, 0830–17.30). Normal çalışma saatleri içindeki aksama süresi, herhangi bir günde beş saniyeyi aşmayacaktır.

## **Örgütsel Gereksinim**

Mentcare sistemi kullanıcıları, sağlık yetkilisi kimlik kartlarını kullanarak kendilerini doğrulayacaklardır.

## **Dış Gereksinim**

Sistem, HStan-03-2006-priv'de belirtilen hasta gizliliği hükümlerini uygulayacaktır.

# Hedefler ve Gereksinimler

- Fonksiyonel olmayan gereksinimlerin tam olarak belirtilmesi çok zor olabilir ve kesin olmayan gereksinimlerin doğrulanması zor olabilir.
- Hedef
  - Kullanım kolaylığı gibi genel bir kullanıcı isteği.
- Doğrulanabilir fonksiyonel olmayan gereksinim
  - Nesnel olarak test edilebilen bazı ölçümleri kullanan bir ifade.
- Amaçlar, sistem kullanıcılarının isteklerini belirttikleri için geliştiricilere yardımcı olur.

# Kullanılabilirlik Gereksinimleri

- Sistem, sağlık personeli tarafından kullanımı kolay olmalı ve kullanıcı hatalarını en aza indirecek şekilde düzenlenmelidir. (Hedef)
- Sağlık personeli dört saatlik eğitimin ardından tüm sistem fonksiyonlarını kullanabilecektir. Bu eğitimden sonra tecrübeli kullanıcılar tarafından yapılan ortalama hata sayısı sistem kullanım saatinde ikiyi geçmeyecektir. (Test edilebilir, fonksiyonel olmayan gereksinim)

# Fonksiyonel Olmayan Gereksinimleri Belirlemek İçin Ölçütler

Nitelik	Ölçüt
Hız	İşlem/saniye Kullanıcı/vaka cevap süresi Ekran yenilenme zamanı
Büyüklik	Megabayt/ROM yonga sayısı
Kullanım kolaylığı	Eğitim zamanı Yardım ekranlarının sayısı
Güvenilirlik	Hata ortalama zamanı Kullanılamama olasılığı Hata sıklığı Kullanılabilirlik
Dayanıklılık	Hatadan sonra tekrar başlama süresi Hataya neden olan vakaların yüzdesi Hata durumunda verinin bozulması olasılığı
Taşınabilirlik	Hedef bağımlı bildirimlerin oranı Hedef sistemlerin sayısı

# Sonraki Ders

- Gereksinim Mühendisliği Süreçleri
- Gereksinimlerin Tanımlanması
- Gereksinimlerin Doğrulanması

# BİL 407 Yazılım Mühendisliği

## Hafta 6

Doç. Dr. Sedat Akleylek

[sedat.akleylek@bil.omu.edu.tr](mailto:sedat.akleylek@bil.omu.edu.tr)

Doç. Dr. Sedat Akleylek

([sedat.akleylek@omu.edu.tr](mailto:sedat.akleylek@omu.edu.tr)) tarafından

hazırlanan bu videonun 5846 sayılı Fikir ve Sanat Eserleri Kanunu kapsamında başka bir yolla yayımlanmasına izin verilmemektedir.

Başka bir ifadeyle, **bu videonun izin alınmaksızın BİL 407 Yazılım Mühendisliği dersine kayıtlı olmayan kişiler ile paylaşılması, OMÜ ortamı dışına taşınması, paylaşılması veya saklanması yasaktır.**