

# YAZ16103

# Yazılım Mühendisliğine Giriş

Dr. Öğr. Üyesi Bora ASLAN

## İÇİNDEKİLER

### BÖLÜM 4: GEREKSİNİM MÜHENDİSLİĞİ

#### FONKSİYONEL VE FONKSİYONEL OLMAYAN GEREKSİNİMLER

#### GEREKSİNİM MÜHENDİSLİĞİ SÜRECİ

#### GEREKSİNİMLERİN AÇIĞA ÇIKARILMASI

#### GEREKSİNİMLERİN SPESİFİKASYONU

#### GEREKSİNİMLERİN DOĞRULANMASI

#### GEREKSİNİMLERİN DEĞİŞİMİ

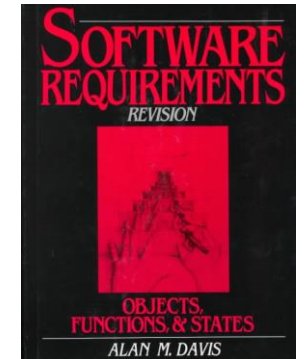
# 4.BÖLÜM GEREKSİNİM MÜHENDİSLİĞİ

Bir sistemin gereksinimleri sistemin sunacağı servislerin ve o sistemin işleyişi üstündeki kısıtların tanımıdır.

Sistem servislerinin ve kısıtlamalarının bulunması, analiz edilmesi, kaydedilmesi kontrol edilmesi süreci gereksinim mühendisliği (GM) olarak adlandırılır.

*«Eğer bir şirket büyük bir yazılım geliştirme projesi için bir sözleşme hazırlamak isterse, gereksinimlerini yeterince soyut, belli bir çözüme işaret etmeksizin tanımlamalıdır. Gereksinimler birkaç yüklenicinin sözleşme için teklif vereceği ve belki de şirketin gereksinimlerini farklı biçimlerde karşılayacağı şekilde yazılmalıdır. Proje için seçilen yüklenici, müşterinin geliştirilecek yazılımın ne yapacağını anlayabildiği ve doğrulayabildiği, çok daha ayrıntılı bir sistem tanımı yazmalıdır. Bu dokümanların her ikisi sistemin gereksinim dokümanları olarak adlandırılabilir.»*

Alan Davis



Kullanıcı gereksinimleri ve sistem gereksinimleri aşağıdaki gibi tanımlanabilir:

1. Kullanıcı gereksinimleri sistemin kullanıcıya ne tür servisler sunacağı ve hangi kısıtlamalar altında çalışmak zorunda olduğunun doğal dilde ve çizimler ile ifadesidir. Kullanıcı gereksinimleri gerekli sistem özelliklerinin genel olarak anlatıldığı ifadelerden sistem işlevselliğinin ayrıntılı ve kesin olarak tariflendiği ifadelere kadar değişebilir.
2. Sistem gereksinimleri yazılım sisteminin fonksiyonlarının, servislerinin, işletme kısıtlarının çok daha ayrıntılı açıklamasıdır. Sistem gereksinimleri dokümanı (bazen fonksiyonel spesifikasyon olarak adlandırılır) tam olarak ne gerçekleştirileceğini tanımlamalıdır. Sistemi satın alan ve yazılım geliştiriciler arasındaki kontratın bir parçası olabilir.

Bir sistem hakkındaki bilginin farklı okuyuculara aktarılabilmesi için farklı türlerde gereksinimlere ihtiyaç duyulur.

#### 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

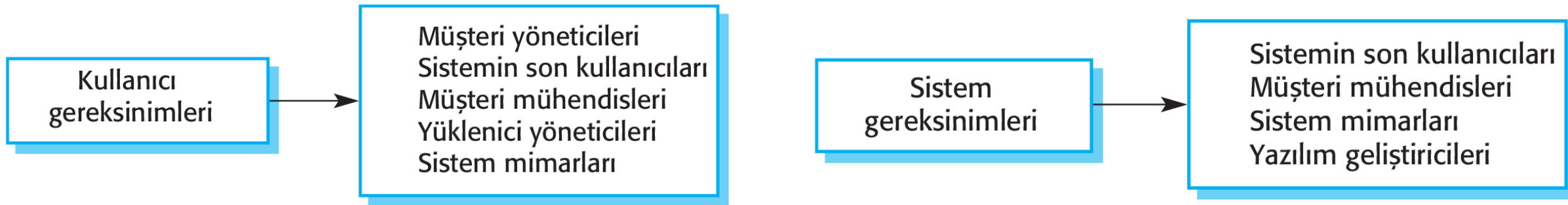
- 1.1 Her ayın son çalışma gününde, reçetelenen ilaçların, maliyetlerinin, hangi klinik tarafından reçetelendiklerinin gösterildiği özet rapor hazırlanmalıdır.
- 1.2 Sistem raporu basılması için ayın son çalışma gününde saat 17.30'dan sonra hazırlanmalıdır.
- 1.3 Rapor her klinik için hazırlanmalıdır 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.

Sistem gereksinimleri;  
gerçekleştirilecek sistemin  
servisleri ve fonksiyonları için  
çok daha özgün bilgi sunar.

Farklı tipteki okuyucular gereksinimleri değişik şekillerde kullandıkları için, gereksinimleri farklı ayrıntı düzeylerinde yazmalısınız.

Kullanıcı gereksinimlerinin okuyucuları, sistemin ayrıntılı olanaklarını bilmek zorunda olmayan yöneticiler olabilir ve genellikle sistemin nasıl gerçekleştirileceği ile ilgilenmez.

Sistem gereksinimlerinin okuyucuları sistemin iş süreçlerini nasıl destekleyeceği ile ilgilendikleri ya da sistemin gerçekleştirilmesinde yer aldıkları için, sistemin ne yapacağını kesin olarak bilmek durumundadır.





Paydaşlar son kullanıcılardan yöneticilere, sistemin kabulünü onaylayacak denetleyiciler gibi dış paydaşlara kadar değişir.

ZihinSaS sisteminin paydaşları aşağıdakilerdir:

1. Bilgileri sisteme kaydedilmiş hastalar ve yakınları
2. Hastaları değerlendirmekten ve tedavi etmekten sorumlu olan doktorlar.
3. Doktorlar ile görüş alışverişlerini düzenleyen ve bazı tedavileri uygulayan hemşireler.
4. Hastaların randevularını yöneten tıbbi danışma görevlileri.
5. Sistemi kurmak ve bakmakla yükümlü BT çalışanları.
6. Sistemin geçerli hasta sağlığı etik kurallarını karşıladığını garanti eden tıp etiği yöneticisi.
7. Sistemden yönetim bilgisi edinen sağlık servisi yöneticileri.
8. Sistem bilgisinin bakım yapılabilir ve korunabilir olduğunu ve kayıt tutma yöntemlerinin uygun işletildiğini garanti etmekle yükümlü tıbbi kayıt elemanları.



Gereksinim mühendisliği genellikle yazılım mühendisliğinin ilk fazı olarak sunulur. Ancak, sistemin tedarik edilmesine ya da geliştirilmesine karar vermeden öncel sistem gereksinimlerinin belli bir düzeyde anlaşılması gereklidir.

Bu erken-faz (*early-stage*) gereksinim mühendisliği sistemin neler yapabileceğinin ve ne tür yararlar sağlayabileceğinin üst-düzey (*high-level*) görüntüsünü belirler.

Bunlar daha sonra sistemin teknik ve parasal olabilirliğinin değerlendirileceği bir olabilirlik çalışmasında (*feasibility study*) dikkate alınabilir.

Bu çalışmanın sonuçları yönetime, sistemin tedarik edilmesi ya da geliştirilmesine devam edilip edilmeyeceği konusunda yardımcı olur.



## Olabilirlik çalışmaları

Olabilirlik çalışması GM sürecinin erken safhalarında yapılması gereken kısa, odaklanmış bir çalışmadır. Bu çalışma üç anahtar soruya cevap vermelidir. (1) Sistem organizasyonun büyüyen hedeflerine katkıda bulunuyor mu? (2) Sistem mevcut teknoloji kullanılarak zamanında ve bütçesi içinde gerçekleştirilebilir mi? (3) Sistem kullanılan diğer sistemler ile bütünleştirilebilir mi?

Eğer bu soruların herhangi birinin cevabı hayır ise, proje ile devam etmemelisiniz.

Büyük sistemlerin çoğu için gerçekleştirim (*implementation*) başlamadan önce açıklıkla belirlenebilen bir gereksinim mühendisliği fazı vardır.

Bu fazın çıktısı sistem geliştirim sözleşmesinin bir parçası olabilecek gereksinimler dokümanıdır.

Doğal olarak gereksinimler daha sonra değişikliğe uğrayabilir ve kullanıcı gereksinimleri daha ayrıntılı sistem gereksinimlerine genişletilebilir.

Bazen gereksinimlerin, sistem geliştirimi ile eş zamanlı açığa çıkarıldığı çevik (*agile*) yaklaşımlar, gereksinimleri ayrıntılandırmak ve saflaştırmak için kullanılabilir.

# FONKSİYONEL VE FONKSİYONEL OLMAYAN GEREKSİNİMLER

Yazılım sistemi gereksinimleri genellikle fonksiyonel ve fonksiyonel olmayan gereksinimler olarak sınıflandırılır:

1. Fonksiyonel gereksinimler (*Functional requirements*) Bunlar sistemin sunması gerekli servislerin, sistemin belli girdilere nasıl tepki vermesi gerektiğinin, belli durumlarda sistemin nasıl davranması gerektiğinin ifadesidir. Fonksiyonel gereksinimler bazı durumlarda sistemin ne yapmaması gerektiğini de açıkça belirtebilir.
2. Fonksiyonel olmayan gereksinimler (*Non-functional requirements*) Bunlar sistemin sunduğu servisler ve fonksiyonlar üstündeki kısıtlamalardır. Zaman kısıtlamalarını, geliştirme süreci üstündeki kısıtlamaları ve standartlarla dayatılan kısıtlamaları içerirler. Fonksiyonel olmayan gereksinimler genellikle tek tek sistem olanakları ve servislerinden çok, sistemin bütününe uygulanır.

# FONKSİYONEL VE FONKSİYONEL OLMAYAN GEREKSİNİMLER

Gerçekte çeşitli gereksinimler arasındaki farklılık bu basit tanımların önerdiği kadar açık değildir.

Erişimi yetkili kullanıcılar ile sınırlandıran bir spesifikasyonda olduğu gibi güvenliği ilgilendiren bir kullanıcı gereksinimi fonksiyonel olmayan bir gereksinim gibi gözükebilir. Ancak, daha ayrıntılı geliştirildiğinde bu gereksinim, kimlik denetleme olanaklarının sistemde yer alması gerektiği gibi, açıkça fonksiyonel olan başka gereksinimler oluşturulabilir.

Bu durum gereksinimlerin birbirinden bağımsız olamadığını ve bir gereksinimin genellikle başka bir gereksinimi yarattığını ya da kısıtladığını gösterir.

Bu nedenle sistem gereksinimleri sadece sistemin gerekli servislerini ve olanaklarını belirtmez; aynı zamanda bu servis/olanakların etkin sunumunu garanti altına alacak gerekli fonksiyonları da belirtir.

# FONKSİYONEL GEREKSİNİMLER

- Bir sistemin fonksiyonel gereksinimleri o sistemin ne yapması gerektiğini tanımlar.
- Bu gereksinimler geliştirilecek yazılım tipine, yazılımı kullanması beklenen kullanıcılara ve organizasyonun gereksinimleri yazarken izlediği genel yaklaşıma bağlıdır.
- Fonksiyonel gereksinimler kullanıcı gereksinimleri olarak ifade edildiklerinde, sistem kullanıcılarının ve yöneticilerinin anlayabileceği şekilde doğal dilde yazılmadır.
- Fonksiyonel sistem gereksinimleri kullanıcı gereksinimlerini genişletir ve sistem geliştiricileri için yazılır.
- Sistemin fonksiyonlarını, onların girdi ve çıktılarını ve istisnai durumları ayrıntılı olarak tanımlamalıdır.



# FONKSİYONEL GEREKSİNİMLER

- Fonksiyonel sistem gereksinimleri sistemin ne yapması gerektiğini içeren genel gereksinimlerden, yerel çalışma biçimlerine ya da organizasyonun mevcut sistemlerine kadar çeşitlilik gösterir.
- Örneğin, ZihinSaS sisteminin akıl ve ruh sağlığı tedavisi gören hastalar ile ilgili bilgi saklamak için belirlenmiş fonksiyonel gereksinimleri aşağıdadır:
  1. Kullanıcı bütün klinikler için randevu listelerini sorgulayabilmelidir.
  2. Sistem her gün, her klinik için, o gün randevuya gelecek hastalar listesi üretmelidir.
  3. Sistemi kullanan her personelin kimliği, kendisinin sekiz haneli personel numarası ile özgün olarak tanımlanmalıdır.



# FONKSİYONEL GEREKSİNİMLER

- Fonksiyonel gereksinimler, adlarından da anlaşılacağı gibi, geleneksel olarak sistemin ne yapması gerektiğine yoğunlaşır.
- Organizasyon mevcut bir hazır sistem yazılımı ürününün ihtiyaçlarını karşılayacağına karar verirse, ayrıntılı fonksiyonel tanımlamalar yapmanın fazla anlamı yoktur.
- Bu durumda, insanların işlerini yapmaları için gerek duyacakları bilgiyi tanımlayan bilgi gereksinimlerinin geliştirilmesine odaklanılmalıdır.
- Bilgi gereksinimleri gerek duyulan bilgiyi ve bu bilginin organizasyona nasıl sunulacağını belirler.

ZihinSaS sistemi için bir bilgi gereksinimi, bir randevu gününde beklenen hastalar listesinde ne tür bilginin yer alması gerektiğini belirtebilir.

# FONKSİYONEL GEREKSİNİMLER

- Gereksinimlerin tanımlanmasında belirsizlikler müşteriler ve yazılım geliştiricileri arasında tartışmalara neden olabilir.
- Belirsiz bir gereksinimin yazılım geliştiricileri tarafından geliştirme işlemini kolaylaştıracak şekilde yorumlanması doğaldır. Ama sıklıkla bu, müşterinin istediği bir durum değildir.
- Yeni gereksinimler belirlenmeli ve sistem değiştirilmelidir. Bu da sistemin teslimini geciktirir ve maliyetini arttırır.
- Örneğin, ZihinSaS sistem gereksinimi kullanıcının bütün klinikler için randevu listelerini sorgulayabilmesini belirtir. Bu gereksinimin altında yatan mantık ruh ve akıl sağlığı hastalarının bazen şaşırdıklarıdır. Bir klinikte randevuları olabilir ama başka bir kliniğe gidebilirler. Eğer bir randevuları varsa, gittikleri kliniğe bakmaksızın randevularına geldikleri kaydedilecektir.

# FONKSİYONEL GEREKSİNİMLER

- Örneğin, ZihinSaS sistem gereksinimi kullanıcının bütün klinikler için randevu listelerini sorgulayabilmesini belirtir. Bu gereksinimin altında yatan mantık ruh ve akıl sağlığı hastalarının bazen şaşırdıklarıdır. Bir klinikte randevuları olabilir ama başka bir kliniğe gidebilirler. Eğer bir randevuları varsa, gittikleri kliniğe bakmaksızın randevularına geldikleri kaydedilecektir.
- Sorgulama gereksinimini belirten bir sağlık personeli "sorgulamanın" sistemin ismi girilen bir hastanın tüm kliniklerdeki tüm randevularını göstermesi anlamına geldiğini düşünebilir. Halbuki bu durum gereksinimde açık değildir.
- Sistem geliştiricisi, gereksinimi, geliştirimi kolaylaştıracak biçimde yorumlayabilir. Sistemin sorgulama fonksiyonu kullanıcının bir klinik seçmesini ve ardından bildirilen hastanın bu kliniğe gidip gitmediğinin sorgulanması biçiminde geliştirilebilir.
- Bu durumda kullanıcı daha fazla girdi yapacak ve dolayısı ile sorgulama daha uzun sürecektir.

# FONKSİYONEL GEREKSİNİMLER

- İdeal olarak sistemin fonksiyonel gereksinimlerinin tanımı tam ve tutarlı olmalıdır.
  - Tam, kullanıcı tarafından istenen bütün servislerin ve bilginin eksiksiz tanımlanması demektir.
  - Tutarlılık gereksinimlerin birbirleri ile çelişmiyor olması anlamındadır.
- Pratikte gereksinimlerin tamlık ve tutarlılıkları çok küçük yazılım sistemlerinde sağlanabilir.
- Büyük ve karmaşık sistemler için gereksinimler yazılırken hata yapmanın ve atlamaların kolaylığı, nedenlerden biridir. Bir diğer neden, büyük sistemlerin çeşitli geçmişe sahip, farklı beklentilerde pek çok paydaşlarının olmasıdır.
- Paydaşlar genellikle farklı ve tutarsız gereksinimlere sahiptir.
- Bu tutarsızlıklar gereksinimler ilk kez belirlenirken bariz olmayabilir ve ancak sistem geliştirimi sırasında daha ayrıntılı analizler sonunda ortaya çıkabilir.

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

- Fonksiyonel olmayan gereksinimler sistemin kullanıcılarına sunduğu servisler ile direkt ilgili olmayan gereksinimlerdir.
- Fonksiyonel olmayan gereksinimler genellikle sistemin bir bütün olarak özelliklerini belirler ya da kısıtlar.
  - Bunlar güvenilirlik, cevap verme zamanı, bellek kullanımı gibi yeni belirmekte olan sistem nitelikleri ile ilgili olabilir.
  - Girdi/çıkış cihazlarının yetenekleri, diğer sistemler ile ara yüzlerde kullanılan veri gösterimleri gibi sistem gerçekleştirimi üstündeki kısıtlamalar olabilir.

# FONKSİYONEL OLMAYAN GEREKSİNİMLER

- Fonksiyonel olmayan gereksinimler genellikle tek tek fonksiyonel gereksinimlerden daha kritiktir.
- Sistem kullanıcıları genellikle kendi ihtiyaçlarını karşılamayan sistem fonksiyonlarının arkasından dolanacak yollar bulabilirler.
- Ancak, bir fonksiyonel olmayan gereksinimi karşılamamak bütün sistemi kullanışsız kılmak demek olabilir.

Örneğin, bir uçak sistemi güvenilirlik gereksinimini karşılayamazsa, çalışması güvenli olarak sertifikalanamaz; eğer bir gömülü kontrol sistemi performans gereksinimlerini karşılayamazsa, kontrol fonksiyonunu doğru yerine getiremez.



# FONKSİYONEL OLMAYAN GEREKSİNİMLER

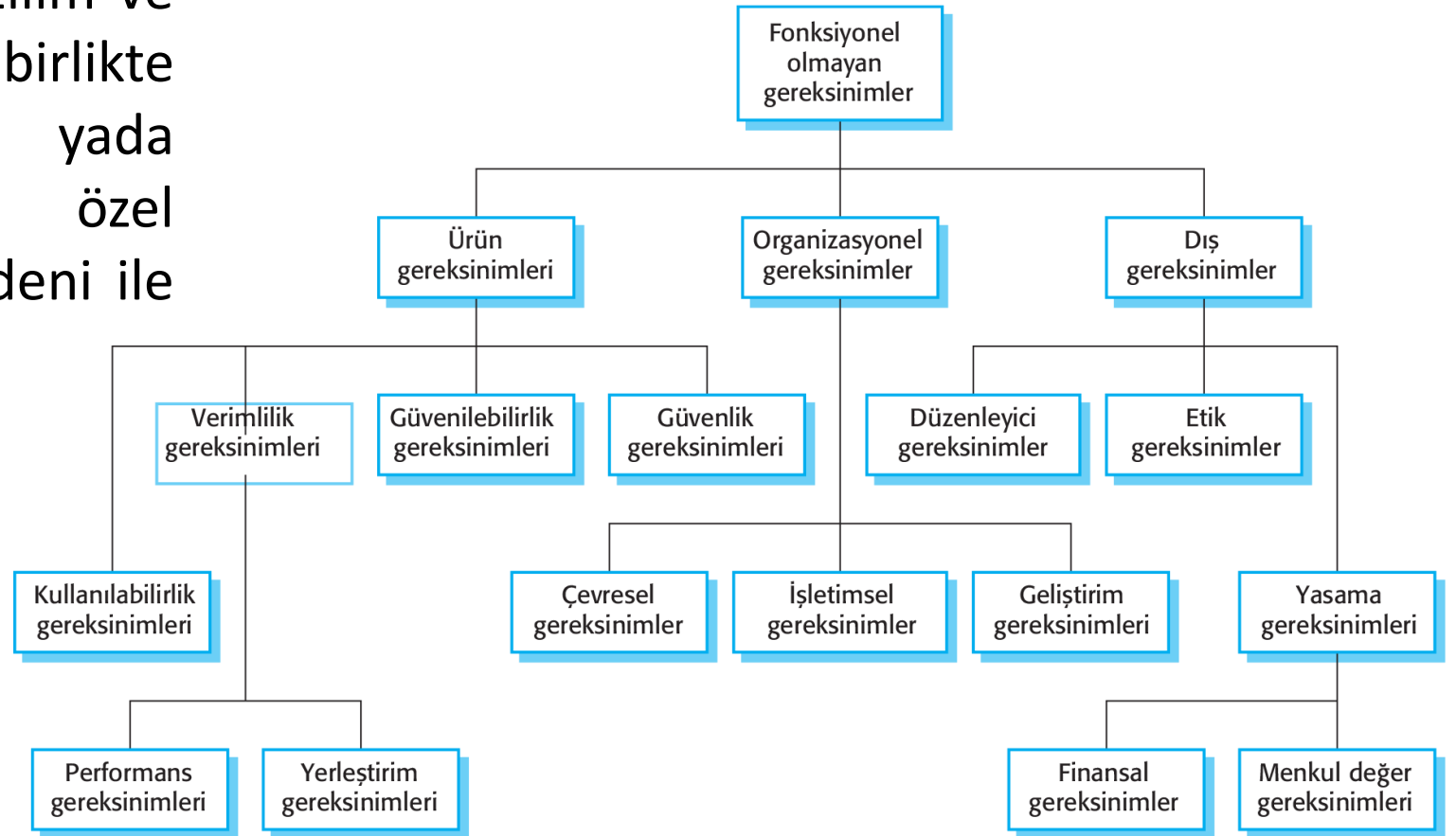
- Hangi sistem bileşeninin hangi fonksiyonel gereksinimi karşıladığını (örneğin, raporlama gereksinimlerini karşılayan bir biçimleme bileşeni olabilir) belirlemek mümkün olmakla birlikte, bu durum fonksiyonel olmayan gereksinimler için çok daha güçtür.
- Bu gereksinimlerin gerçekleştirimi iki nedenden dolayı tüm sisteme yayılmış olabilir:
  1. Fonksiyonel olmayan gereksinimler tekil bileşenleri değil tüm sistem mimarisini etkiliyor olabilir.
    - Örneğin, gömülü bir sistemde performans gereksinimlerini karşılayabilmek için, sistemi, bileşenleri arasındaki iletişimi en aza indirecek şekilde düzenlemeniz gerekebilir.
  2. Tek bir fonksiyonel olmayan gereksinim, örneğin güvenlik gereksinimi, bu gereksinimi gerçekleştirebilmek için gerekecek yeni sistem servislerini tanımlayan ilgili birkaç fonksiyonel gereksinim yaratabilir. Ek olarak, mevcut gereksinimleri kısıtlayan gereksinimler de yaratabilir; örneğin, bilgi sistemine erişimi kısıtlayabilir.

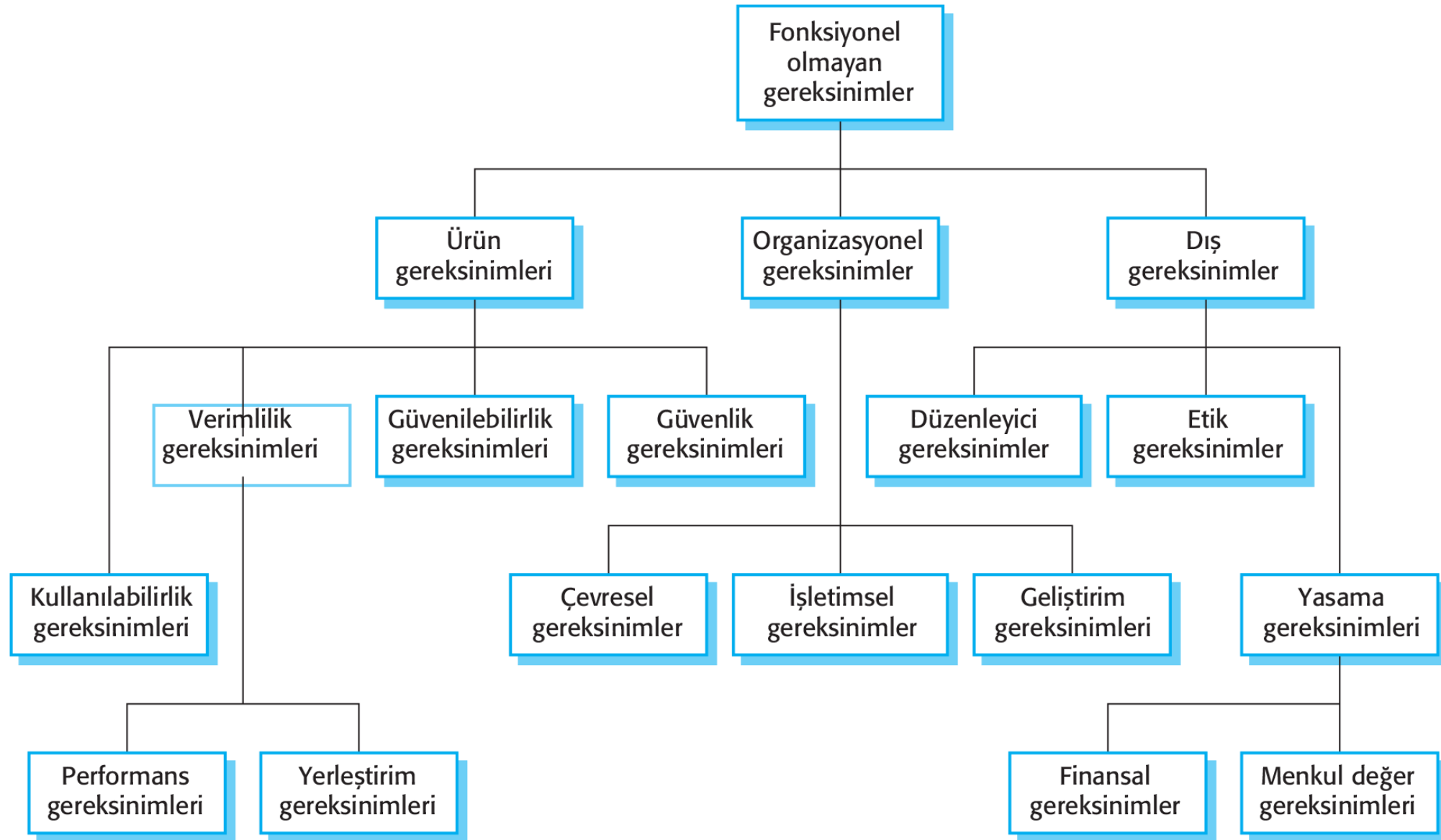


# FONKSİYONEL OLMAYAN GEREKSİNİMLER

- Fonksiyonel olmayan gereksinimler kullanıcının bütçe kısıtlamaları, işletme politikaları, diğer yazılım ve donanım sistemleri ile birlikte çalışabilme gereksinimi, yada güvenlik düzenlemeleri, özel yasama gibi dış etkenler nedeni ile ortaya çıkar.

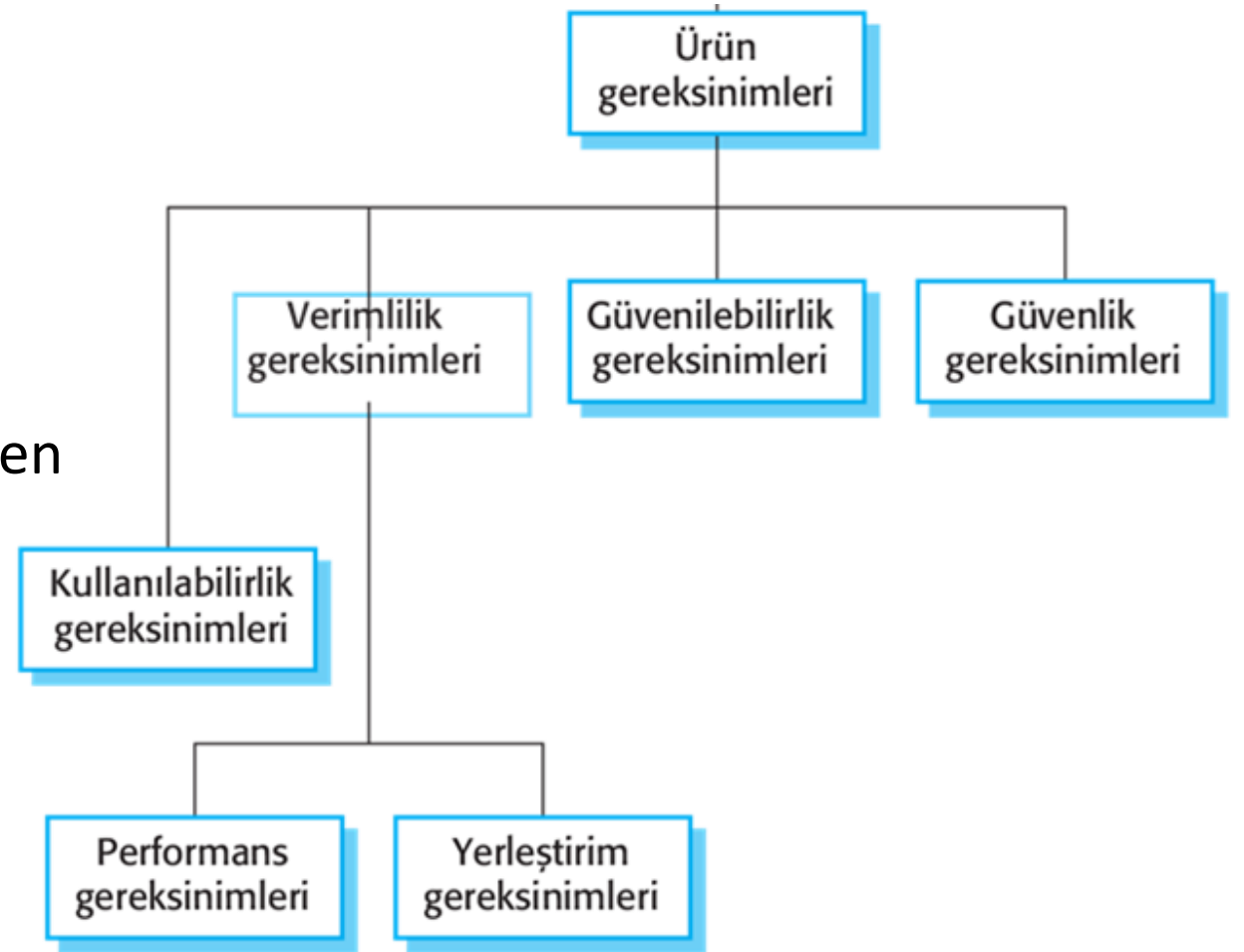
Fonksiyonel olmayan gereksinimlerin bir sınıflandırılması





## Ürün gereksinimleri

- Bu gereksinimler yazılımın çalışma zamanındaki davranışlarını belirler.
  - Sistemin ne kadar hızlı çalışması gerektiğini ve ne kadar belleğe gereksinim duyulduğunu belirleyen performans gereksinimleri
  - Kabul edilebilir hata oranını belirleyen güvenilirlik gereksinimleri;
  - Güvenlik gereksinimleri ve kullanılabilirlik gereksinimleri bu gereksinimlere örnektir.



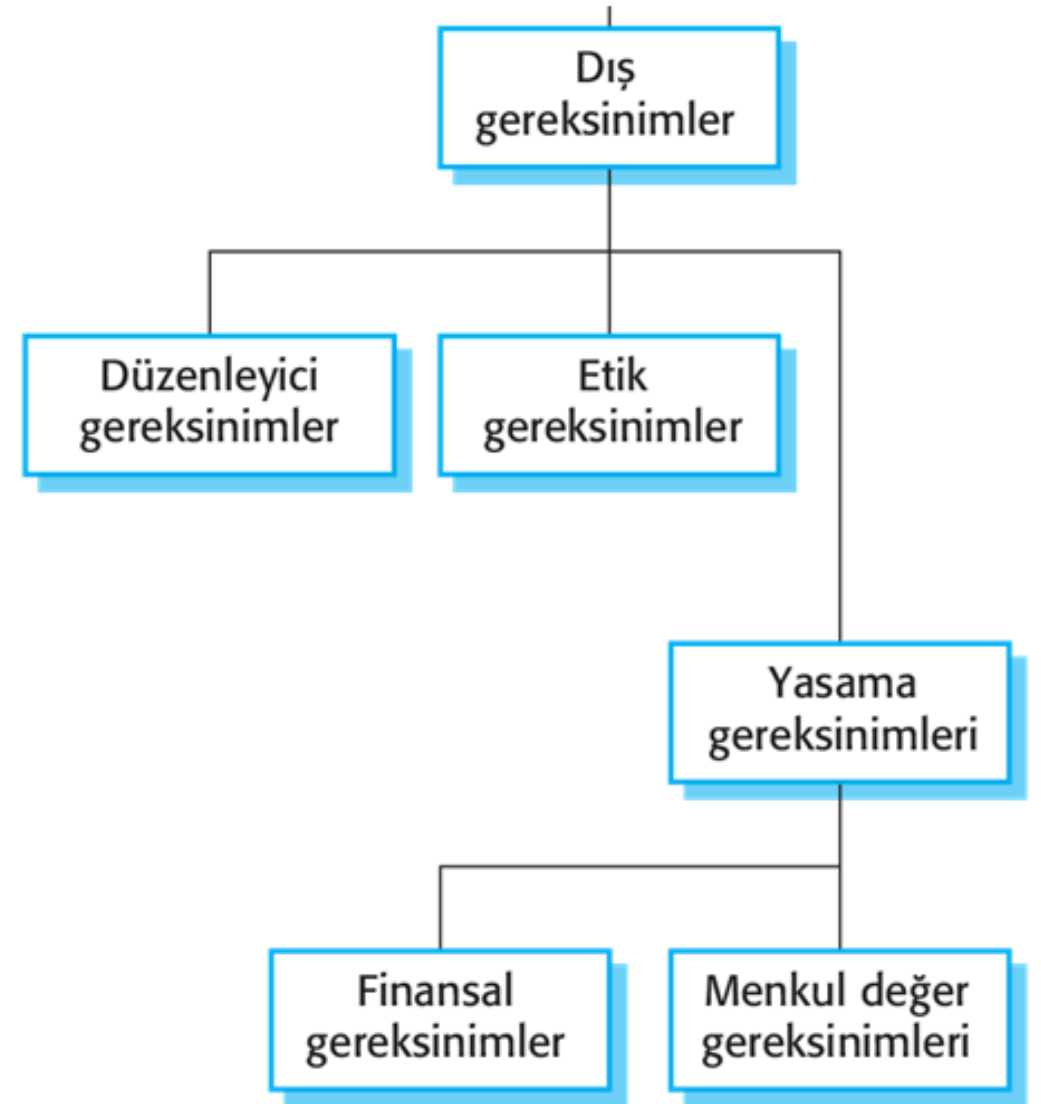
## Organizasyonel gereksinimler

- Bu gereksinimler müşterinin ve geliştiricinin organizasyonunda yer alan politikalar ve prosedürlerden türeyen geniş sistem gereksinimleridir.
  - Sistemin nasıl kullanılacağını belirleyen organizasyonel süreç gereksinimleri;
  - Programlama dilini belirleyen geliştirim süreci gereksinimleri;
  - Kullanılması gerekli geliştirim ortamı ve süreç standartları;
  - Sistemin işletim ortamını belirleyen çevresel gereksinimler bu gereksinimlere örnektir.



## Dış gereksinimler

- Bu geniş başlık sistemin ve geliştirim sürecinin dışındaki etkenlerden kaynaklanan tüm gereksinimleri kapsar.
  - Sistemin bir denetleyici tarafından onaylanması için yapılması gerekenlerin belirtildiği denetleyici gereksinimleri (nükleer güvenlik yetkilisi);
  - Sistemin yasalara uygun işletildiğini garanti etmek için izlenmesi gerekli yasama gereksinimleri;
  - Sistemin kullanıcıları ve/veya toplum tarafından kabul edilebileceğini garanti eden etik gereksinimleri bu gereksinimlere örnektir.



### **ÜRÜN GEREKSİNİMİ**

ZihinSaS sistemi bütün klinikler için normal çalışma saatlerinde (Pazartesi-Cuma, 08.30-17.30) erişilebilir olmalıdır.

### **ORGANİZASYONEL GEREKSİNİM**

ZihinSaS sisteminin kullanıcıları kendilerini sağlık yetkilisi kimlik kartlarını kullanarak tanıtmak zorundadır.

### **DIŞ GEREKSİNİM**

Sistem Hstan-03-2006-priv de belirtilen gizliliği hükümlerini yerine getirmelidir.

- ZihinSaS sistem spesifikasyonunda yer alabilecek ürün, organizasyonel ve dış gereksinim örneklerini yukarıdadır.
- Ürün gereksinimi sistemin her gün ne zaman kullanılması gerektiğini ve ne zaman çalışmayabileceğini belirten kullanılabilirlik gereksinimidir. Bu gereksinim ZihinSaS sisteminin fonksiyonelliği ile ilgili hiçbir şey söylemez ve sistem tasarımcısı tarafından dikkate alınması gerekli bir kısıtlamayı açıkça belirtir.



**ÜRÜN GEREKSİNİMİ**

ZihinSaS sistemi bütün klinikler için normal çalışma saatlerinde (Pazartesi-Cuma, 08.30-17.30) erişilebilir olmalıdır.

**ORGANİZASYONEL GEREKSİNİM**

ZihinSaS sisteminin kullanıcıları kendilerini sağlık yetkilisi kimlik kartlarını kullanarak tanıtmak zorundadır.

**DIŞ GEREKSİNİM**

Sistem Hstan-03-2006-priv de belirtilen gizliliği hükümlerini yerine getirmelidir.

- Organizasyonel gereksinim kullanıcıların kendilerini sisteme nasıl tanıtacaklarını belirtir.
- Sistemi işleten sağlık otoritesi tüm yazılımlar için kullanıcıların bir giriş ismine sahip olmaları yerine, kimlik kartlarını okutarak kendilerini tanıtacakları yeni bir standart kimlik doğrulama sürecine geçmektedir.



**ÜRÜN GEREKSİNİMİ**

ZihinSaS sistemi bütün klinikler için normal çalışma saatlerinde (Pazartesi-Cuma, 08.30-17.30) erişilebilir olmalıdır.

**ORGANİZASYONEL GEREKSİNİM**

ZihinSaS sisteminin kullanıcıları kendilerini sağlık yetkilisi kimlik kartlarını kullanarak tanıtmak zorundadır.

**DIŞ GEREKSİNİM**

Sistem Hstan-03-2006-priv de belirtilen gizliliği hükümlerini yerine getirmelidir.

- Dış gereksinim sistemin gizlilik hükümlerine uyması gerekliliğinden kaynaklanmaktadır.
- Gizlilik sağlık servisleri sektöründe çok önemli bir konudur ve gereksinim sistemin ulusal gizlilik standartlarına uyumlu geliştirilmesi gerektiğini belirtir.

- Paydaşların gereksinimleri kullanım kolaylığı, sistemin hatadan kaçınma becerisi ya da kullanıcıya hızlı cevap vermesi gibi genel amaçlar olarak önermeleri fonksiyonel olmayan gereksinimlerde yaygın bir problemdir.
- Amaçlar iyi niyeti gösterir ama sistem geliştiricileri bunları yorumlamak için konudan uzaklaştıkça problemlere ve ardından sistem teslim edildiğinde tartışmalara neden olur.

*«Sistem tıp personeli tarafından kolay kullanılabilmeli ve kullanıcı hatalarını en aza indirecek şekilde organize edilmelidir.»*

bir yöneticinin kullanılabilirlik gereksinimi

- Sistem amacını objektif olarak sınamak mümkün değildir, ancak en azından kullanıcıların sistemi test ederken yaptıkları hataları sayacak bir yazılım aracı dahil edilebilir.

*«Tıp personeli tüm sistem fonksiyonlarını iki saatlik bir eğitimden sonra kullanabilmelidir. Bu eğitimden sonra deneyimli kullanıcılar saatte ikiden fazla hata yapmamalıdır.»*

*«Sistem tıp personeli tarafından kolay kullanılabilmeli ve kullanıcı hatalarını en aza indirecek şekilde organize edilmelidir.»*

- zAmaçlar iyi niyeti gösterir ama sistem geliştiricileri bunları yorumlamak için konudan uzaklaştıkça problemlere ve ardından sistem teslim edildiğinde tartışmalara neden olur.

*«Tıp personeli tüm sistem fonksiyonlarını iki saatlik bir eğitimden sonra kullanabilmelidir. Bu eğitimden sonra deneyimli kullanıcılar saatte ikiden fazla hata yapmamalıdır.»*

bir yöneticinin kullanılabilirlik gereksinimi

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

- Mümkün olan her zaman fonksiyonel olmayan gereksinimleri objektif test edilebilmeleri için nicel olarak yazılmalıdır.
- Fonksiyonel olmayan sistem özelliklerini belirlerken kullanabileceğiniz ölçütler ekrandaki gibi olabilir.
- Bu nitelikleri sistemi, fonksiyonel olmayan gereksinimleri karşılayıp karşılamadığını test ederken ölçebilirsiniz.

- Fonksiyonel olmayan gereksinimler genellikle diğer fonksiyonel ve fonksiyonel olmayan gereksinimler ile çelişir ve etkileşir.

### **ORGANİZASYONEL GEREKSİNİM**

ZihinSaS sisteminin kullanıcıları kendilerini sağlık yetkilisi kimlik kartlarını kullanarak tanıtmak zorundadır.

- Kimlik kartı tanıma gereksinimi sisteme bağlı her bilgisayara bir okuyucu bağlanmasını gerektirir.
- Öte yandan doktorların ve hemşirelerin tabletleri ya da akıllı telefonları ile sisteme bağlanabilmeleri gerektiğini belirten bir başka gereksinim olabilir.
- Bu cihazlar genellikle kart okuyucular ile birlikte kullanılmazlar ve bu nedenle bir başka tanıma yöntemini de desteklemek zorunda kalınabilir.

- Gereksinim dokümanında fonksiyonel ve fonksiyonel olmayan gereksinimleri ayırmak zordur.
- Eğer fonksiyonel olmayan gereksinimler fonksiyonel gereksinimlerden ayrı belirtilirse, aralarındaki ilişkiyi anlamak zor olabilir.
- Bununla birlikte, ideal olarak, performans ya da güvenilirlik gibi yeni gelişmekte olan sistem özellikleri ile ilgili gereksinimleri vurgulamanız gerekir.
- Bunu onları gereksinim dokümanının ayrı bir bölümüne koyarak, ya da herhangi bir şekilde diğer sistem gereksinimlerinden ayırıştırarak yapabilirsiniz.
- Güvenilirlik (reliability), güvenlik (safety), gizlilik (confidentiality) gereksinimleri gibi fonksiyonel olmayan gereksinimler kritik sistemlerde özellikle önemlidir.



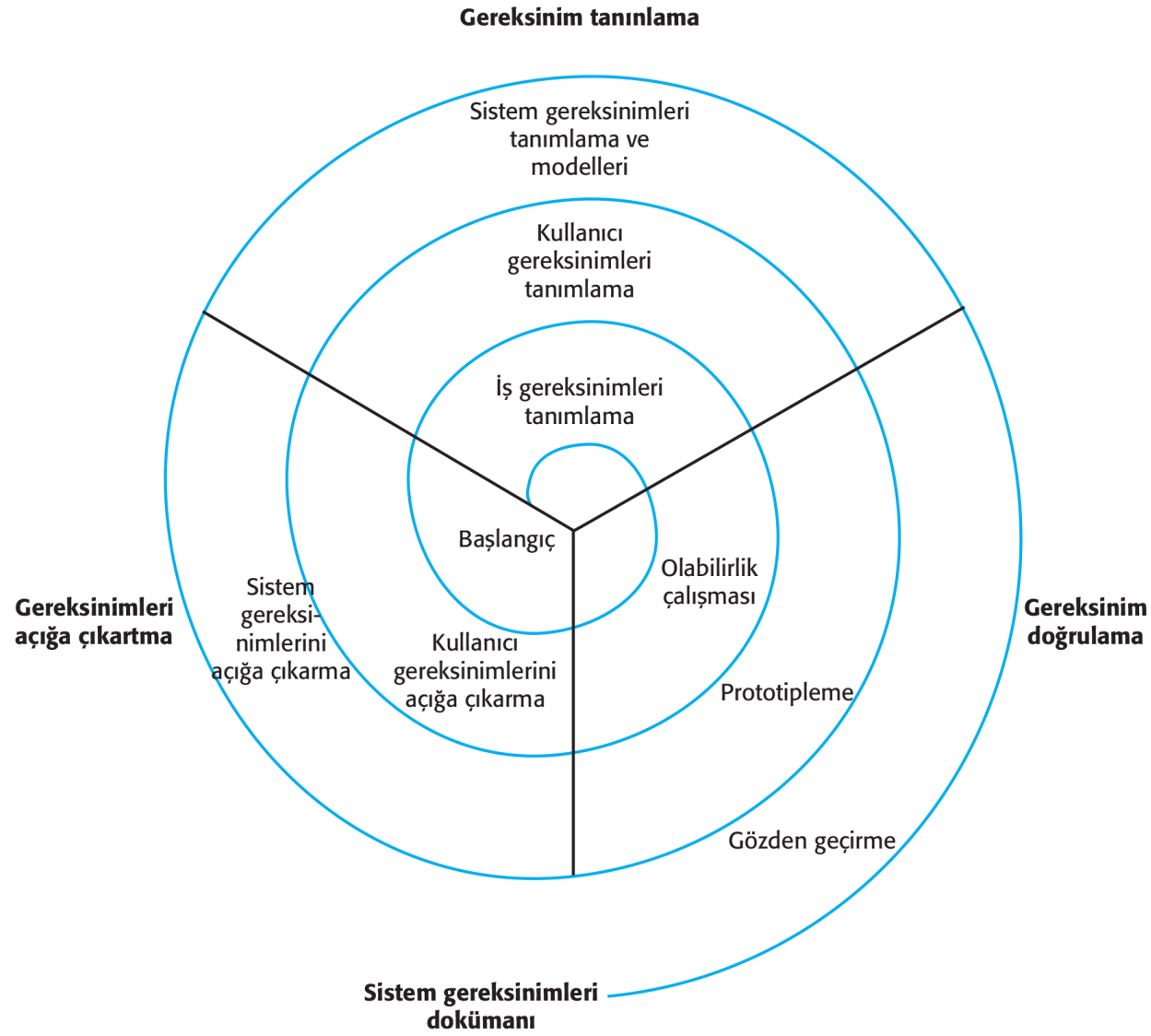
# GEREKSİNİM MÜHENDİSLİĞİ SÜRECİ

Gereksinim mühendisliği üç temel eylem içerir. Bunlar, paydaşlar ile etkileşerek gereksinimleri keşfetmek (açığa çıkartmak ve analiz); bu gereksinimleri standart biçime çevirmek (tanımlama); ve gereksinimlerin gerçekten kullanıcının istediği sistemi tanımladıklarını kontrol etmektir (doğrulamak).

Pratikte gereksinim mühendisliği eylemlerin dönüşümlü çalıştığı yinelemeli bir süreçtir.

Eylemler bir sarmal etrafında yinelemeli bir süreç olarak düzenlenmiştir. GM sürecinin çıktısı sistem gereksinimleri dokümanıdır. Bir döngüde her eylem için harcanan zaman ve emek tüm sürecin evresine, geliştirilen sistemin tipine ve ayrılmış bütçeye bağlıdır.



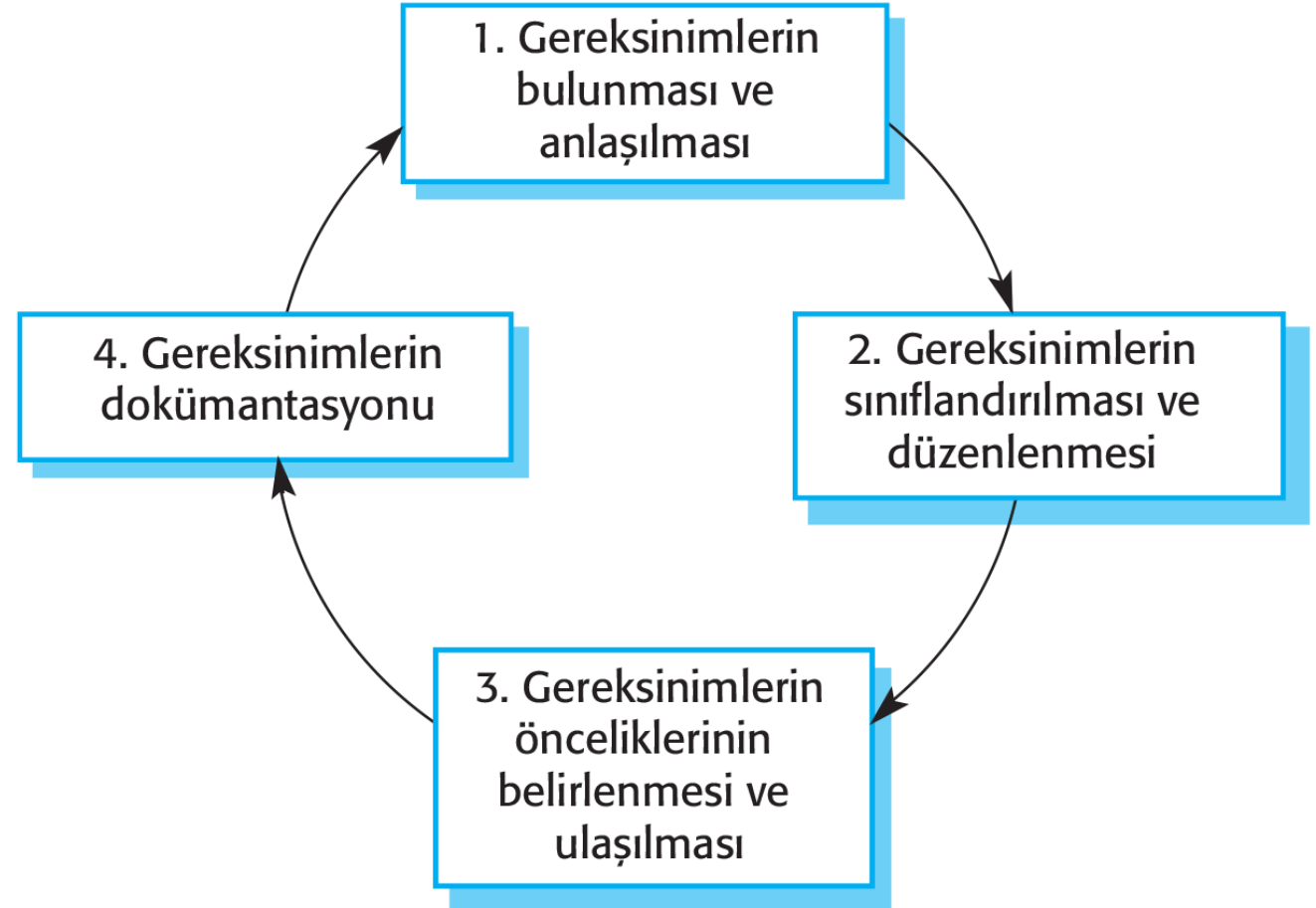


**Şekil 4.6** Gereksinim mühendisliği sürecinin sarmal görüntüsü

# GEREKSİNİMLERİN AÇIĞA ÇIKARILMASI

Gereksinimlerin açığa çıkarılması sürecinin amaçları paydaşların yaptıkları işin ve yeni sistemi bu işi desteklemek için nasıl kullanabileceklerinin anlaşılmasıdır. Paydaşlardan gereksinimleri açığa çıkarmak ve anlamak birkaç nedenden ötürü zor bir süreçtir:

1. Paydaşlar genellikle çok genel gereksinimler dışında bir bilgisayar sisteminden ne istediklerini bilmezler; sistemin ne yapmasını istediklerini dile getirmekte zorlanabilirler; yapılabilir ve yapılamaz şeyleri bilmedikleri için gerçekçi olmayan isteklerde bulunabilirler.
2. Bir sistemdeki paydaşlar doğal olarak gereksinimleri kendi terimleri ve işlerinin üstü kapalı bilgisi ile ifade eder.
3. Değişik gereksinimlere sahip farklı paydaşlar gereksinimlerini farklı yollardan ifade edebilir.
4. Politik etkenler bir sistemin gereksinimlerini etkileyebilir. Yöneticiler kuruluştaki etkilerini arttıracak sistem gereksinimleri talep edebilir.

**Şekil 4.7**

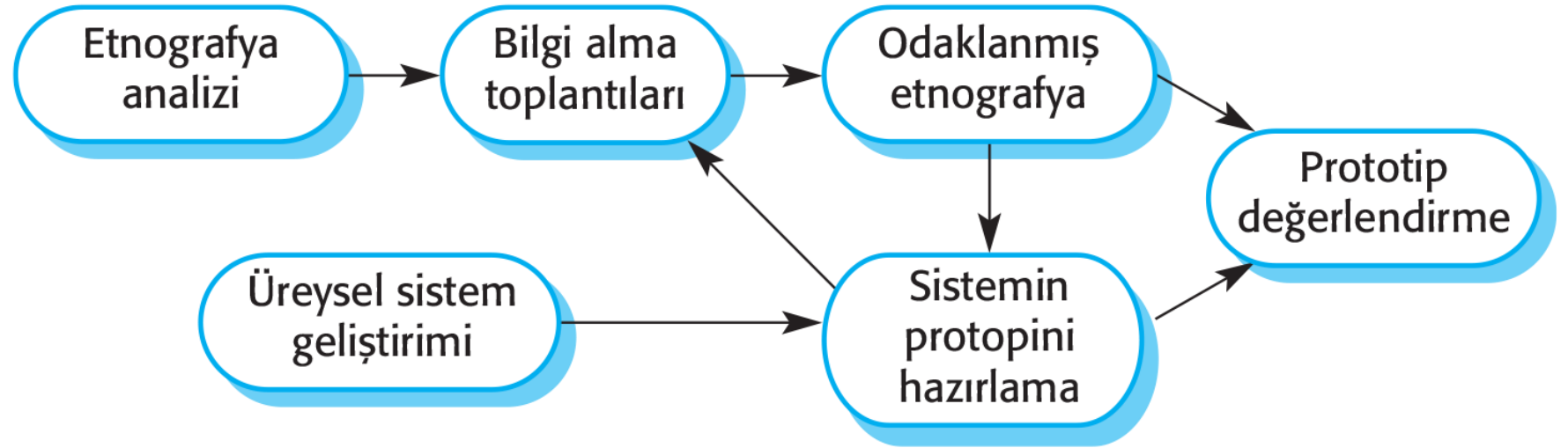
Gereksinimlerin açığa çıkarılması ve analizi süreci



## Bakış açıları

Bir bakış açısı ortak bir şeyleri olan bir grup paydaştan bir dizi gereksinimi toplama ve düzenlemenin bir yoludur. Bu nedenle, her bakış açısı bir dizi sistem gereksinimi içerir. Bakış açıları son kullanıcılardan, yöneticilerden ya da diğerlerinden gelebilir. Gereksinimleri konusunda bilgi sağlayabilecek kişilerin belirlenmesinde ve gereksinimlerin analiz için yapılandırılmasında yararlı olurlar.

**Şekil 4.8**  
Gereksinimlerin analizi  
için etnografya ve  
prototip hazırlama



## Sınıfta resim paylaşmak

Jack Ullapool'da (kuzey Scotland'da bir köy) bir ilkokul öğretmenidir. Bölgedeki balıkçılığın tarihine, gelişimine ve ekonomik etkisine bakarak, bir sınıf projesinin balıkçılık endüstrisi üstüne yoğunlaşmasına karar vermişti. Bu projenin bir parçası olarak çocuklara akrabalarından anılar toplamaları ve paylaşmaları, gazetelerin arşivlerini kullanmaları, balıkçılık ve bölgedeki balıkçı topluluklar ile ilgili eski fotoğraflar toplamaları söylenir. Çocuklar birlikte balıkçılık hikâyelerini toplamak için eOgren, viki ve gazete arşivlerine ve fotoğraflarına ulaşmak için SCRAN (tarih kaynakları sitesi) kullanır. Ancak, Jack çocukların birbirlerinin resimlerini çekmesini ve yorumlamasını ve ailelerinden bulabilecekleri eski fotoğrafları tarayıp yüklemelerini istediği için bir fotoğraf paylaşım sitesine gereksinim duyar.

Jack üyesi olduğu ilkokul öğretmenleri grubuna herhangi biri uygun bir sistem önerebilecek mi diye bir elektronik posta yollar. İki öğretmen cevap verir ve her ikisi de öğretmenlerin içeriğini denetlemesine ve yönetmesine izin veren KidsTakePics fotoğraf paylaşım sitesini kullanmasını önerir. KidsTakePics eOgren kimlik belirleme servisi ile bütünleştirilmediği için Jack bir öğretmen ve sınıf hesabı yaratır. Sınıftaki çocuklar oturumlarını açtıklarında taşınabilir cihazlarından ve sınıf bilgisayarlarından fotoğrafları hemen yükleyebilmeleri için, eOgren kurulum servisini kullanarak, KidsTakePics'i çocuklar tarafından görülen servislere ekler.

### Şekil 4.9

eOgren  
sistemi için  
kullanıcı  
hikayesi



# GEREKSİNİMLERİN SPESİFİKASYONU

Gereksinimlerin spesifikasyonu kullanıcı ve sistem gereksinimlerini bir gereksinim dokümanı içinde yazma sürecidir. İdeal olarak, kullanıcı ve sistem gereksinimleri açık, kesin, kolay anlaşılır, tam ve tutarlı olmalıdır. Pratikte, bunu elde etmek neredeyse imkânsızdır. Paydaşlar gereksinimleri farklı şekillerde yorumlar ve çoğu zaman gereksinimlerde içsel çelişkiler ve tutarsızlıklar vardır.

Kullanıcı gereksinimleri gereksinim dokümanında neredeyse her zaman uygun diyagramlar ve tablolar ile desteklenmiş doğal dilde yazılır. Sistem gereksinimleri de doğal dilde yazılabilir, ancak formlar, grafiksel ya da matematiksel sistem modelleri üstüne kurulu diğer gösterimler de kullanılabilir. Şekil 4.11 sistem gereksinimlerini yazmak için olası gösterimleri özetler.

| Gösterim                      | Tanım   |
|-------------------------------|---|
| Doğal dil cümleleri           | Gereksinimler numaralanmış cümleler kullanılarak doğal dilde yazılır. Her cümle bir gereksinimi açıklamalıdır.  |
| Yapısal doğal dil             | Gereksinimler doğal dilde standart bir form ya da şablon üstüne yazılır. Her alan gereksinimin bir yönü ile ilgili bilgi sağlar.  |
| Grafik gösterim               | Metin açıklamaları ile desteklenmiş grafik modeller sistemin fonksiyonel gereksinimlerini tanımlamak için kullanılır. UML (Unified modeling language) kullanma durumu ve sıra diyagramları sıklıkla kullanılır.   |
| Matematiksel spesifikasyonlar | Bu gösterimler sonlu makineler ya da kümeler gibi matematiksel kavramlara dayanır. Bu kesin spesifikasyonlar bir gereksinim dokümanında belirsizliği azaltsa da, pek çok müşteri resmi bir spesifikasyonu anlamaz. Bunun istediklerini ifade ettiğini kontrol edemezler ve onu bir sistem sözleşmesi olarak kabul etmeye isteksizdirler. (Bu yaklaşımı sistem güvenilebilirliğini içeren Bölüm 10'da tartışıyorum.) |

**Şekil 4.11** Sistem gereksinimlerini yazmak için gösterim

#### Şekil 4.12

İnsülin  
pompa  
yazılım  
sistemi için  
örnek  
gereksinimler

3.2 Sistem kan şekerini ölçmeli ve eğer gerekliyse her 10 dakikada insülin sağlamalıdır. *(Kan şekerinde değişimler oldukça yavaştır, bu nedenle daha sık ölçüm gereksizdir; daha seyrek ölçüm gereksiz yüksek şeker düzeylerine neden olabilir.)*

3.6 Sistem her dakika denenecek koşullar ve Tablo 1’de tanımlanan karşılık gelen eylemler ile bir kendine deneme yordamı çalıştırmalıdır. *(Kendini deneme yordamı donanım ve yazılım problemlerini bulabilir ve kullanıcıyı normal işlevin olanaksızlığı ile ilgili uyarabilir.)*



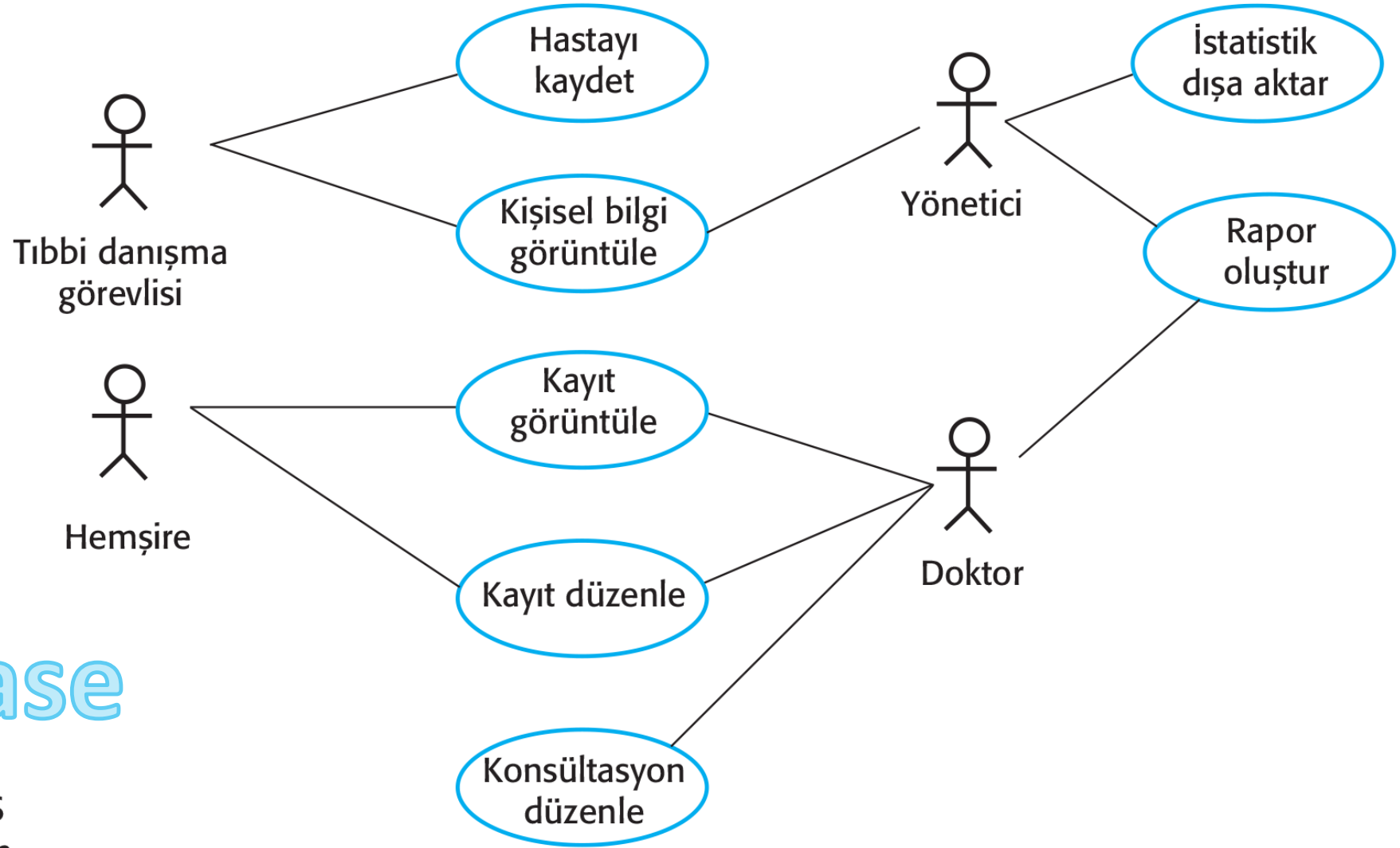
### Gereksinim spesifikasyonu için doğal dil kullanmanın problemleri

Spesifikasyon için yararlı olan doğal dil esnekliği sıkça problemlere neden olur. Açık olmayan gereksinimler yazmak olanaklıdır ve okuyucular (tasarımcılar) kullanıcıdan farklı bir geçmişe sahip oldukları için gereksinimleri yanlış yorumlayabilir. Birkaç gereksinimi tek cümlede birleştirmek kolaydır ve doğal dil gereksinimleri yapılandırmak zor olabilir.

### İnsülin Pompası/Denetleme Yazılımı/SRS/3.3.2

|                     |  |
|---------------------|--|
| <b>Fonksiyon</b>    | İnsülin dozunu hesapla: Güvenli şeker düzeyi.  |
| <b>Tanım</b>        | O an ölçülen şeker düzeyi 3 ile 7 birim arasındaki güvenli dilimdeyken verilmesi gereken insülin dozunu hesaplar.  |
| <b>Girdiler</b>     | O anki şeker okuması (r2), önceki iki okuma (r0 ve r1).  |
| <b>Kaynak</b>       | Sensörden gelen o anki şeker okuması. Bellekten gelen diğer okumalar.  |
| <b>Çıktılar</b>     | CompDose-Verilmesi gereken insülin dozu.   |
| <b>Hedef</b>        | Ana denetleme döngüsü.   |
| <b>Eylem</b>        | Şeker düzeyi kararlıysa ya da düşüyorsa ya da düzey artıyor ama yükselme hızı düşüyorsa CompDose sıfırdır. Eğer düzey artıyorsa ve artma hızı artıyorsa CompDose o anki ve bir önceki şeker düzeylerinin farkının 4'e bölünmesi ve sonucun yuvarlanması ile hesaplanır. Eğer sonuç sıfıra yuvarlanıyorsa CompDose verilebilecek en az doza ayarlanır (Şekil 4.14'e bakınız). |
| <b>Gerektirir</b>   | Şeker düzeyi değişim hızının hesaplanabilmesi için önceki iki okuma.   |
| <b>Ön koşul</b>     | İnsülin deposu en az tek dozda en çok verilebilecek doz kadar insülin içerir.  |
| <b>Son koşul</b>    | r0 r1 ile değiştirilir sonra r1 r2 ile değiştirilir.   |
| <b>Yan etkileri</b> | Yok  |

**Şekil 4.13** Bir insülin pompası için yapısal spesifikasyon



## Use Case

**Şekil 4.15** ZihinSaS sistemi için kullanım durumları



# GEREKSİNİMLERİN DOĞRULANMASI

Gereksinimlerin doğrulanması müşterinin gerçekten istediği sistemi tanımlayan gereksinimlerin kontrol edilmesi sürecidir. Gereksinim problemlerinin bulunması ile ilgilendiği için açığa çıkarma ve analiz ile örtüşür. Gereksinimlerin doğrulanması süreci sırasında gereksinim dokümanındaki gereksinimler üstünde farklı tip kontroller uygulanmalıdır. Bu kontroller aşağıdakileri içerir:

**1. Geçerlik  
Kontrolleri**

**2. Tutarlılık  
Kontrolleri**

**3. Tamlık  
Kontrolleri**

**4. Gerçekçilik  
Kontrolleri**

**5.  
Doğrulanabilirlik  
Kontrolleri**





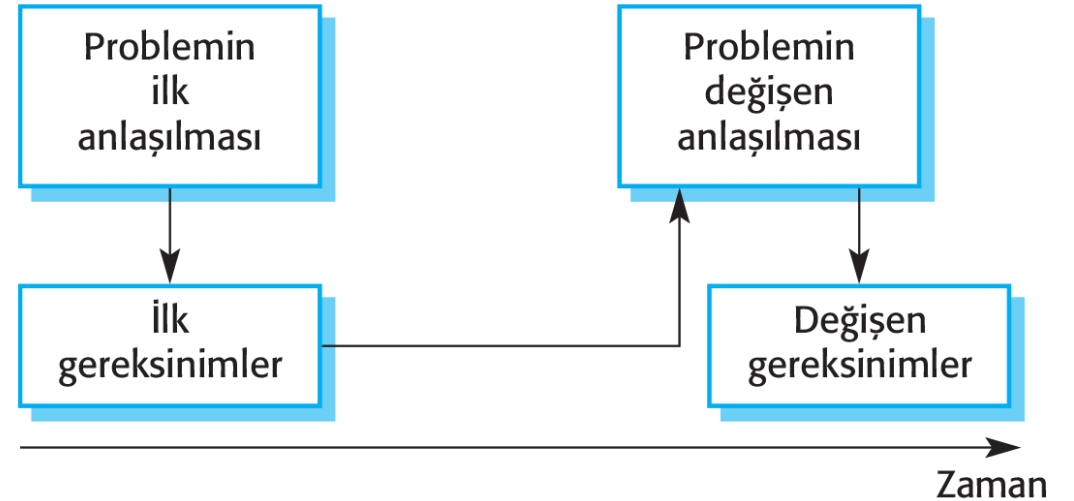
## Gereksinim incelemeleri

Gereksinim incelemesi sistem müşterilerinden ve geliştiricilerinden bir grup insanın gereksinim dokümanını ayrıntılı olarak okuması ve hataları, anormallikleri ve tutarsızlıkları araması sürecidir. Bunlar saptandığında ve kayıt edildiğinde nasıl çözüleceklerini tartışmak müşteri ve geliştiriciye kalır.

# GEREKSİNİMLERİN DEĞİŞİMİ

Büyük yazılım sistemleri için gereksinimler her zaman değişir. Bu sistemlerin genellikle “şeytani” problemleri (bütünüyle tanımlanamayan problemleri) adreslemek üzere geliştiriliyor olması sık değişikliklerin bir nedenidir.

Problemler tam olarak tanımlanamadıkları için, yazılım gereksinimlerinin eksik olması kesin gibidir. Yazılım geliştirimi süreci sırasında paydaşların problemi anlayışları sürekli değişir (Şekil 4.18). Bu durumda sistem gereksinimleri bu problem anlayışındaki değişikliği yansıtmak için değişmek zorundadır.

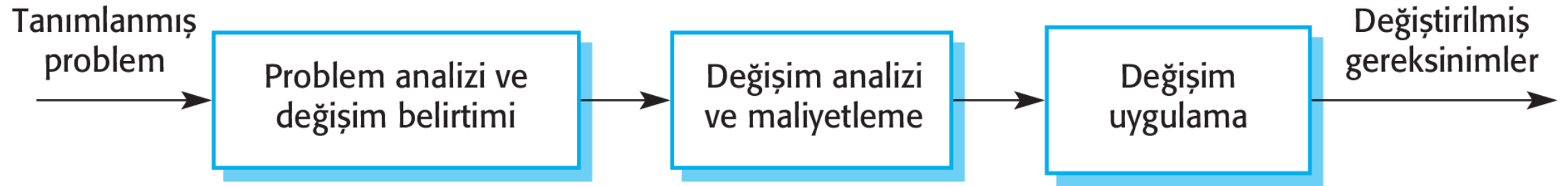


**Şekil 4.18**  
Gereksinimlerin evrimi



## Kalıcı ve uçucu gereksinimler

Bazı gereksinimler değişime diğerlerinden daha duyarlıdır. Kalıcı gereksinimler organizasyonun ana, yavaş değişen eylemleri ile ilişkilidir. Uçucu gereksinimler temel iş eylemleri ile ilişkilidir. Uçucu gereksinimler büyük olasılık ile değişir. Bunlar genellikle işin kendisinden çok organizasyonun o işi nasıl yaptığını yansıtan destekleyici eylemler ile ilişkilidir.



**Şekil 4.19**  
Gereksinim değişimi  
yönetimi

# Okuma ve İzleme Önerisi



İyi Gereksinim Yazma Teknikleri  
Kasım Şen  
Koç Sistem Bilgi ve İletişim Hizmetleri A.Ş.



Software Development Process- Georgia Tech

<https://classroom.udacity.com/courses/ud805>

- Lesson 7: Software Engineering and UML



How to Write A Project Specification, SourceSeek (6 dk)

<https://www.youtube.com/watch?v=MCXi4KtRTG0>

An introduction to Requirements Engineering, Ian Sommerville (10 dk)

<https://www.youtube.com/watch?v=Ec0s0z5uXQ8>

Requirements engineering challenges, Ian Sommerville (12 dk)

<https://www.youtube.com/watch?v=bK-y0CaGkhU>

