

## **Bölüm 4**

# **Gereksinim Mühendisliği (Requirements Engineering)**

# Hedefler

- Kullanıcı ve sistem gereksinimleri kavramlarını ve bu gereksinimlerin neden farklı yollarla yazıldığını anlamak ;
- Fonksiyonel ve fonksiyonel olmayan yazılım gereksinimleri arasındaki farkları anlamak;
- Bir yazılım gereksinim dokümanında gereksinimlerin nasıl düzenleneceğini anlamak;
- Gereksinimleri ortaya çıkarma, analiz ve doğrulama gibi temel gereksinim mühendisliği aktivitelerini ve bu aktiviteler arasındaki ilişkileri anlamak;
- Gereksinim yönetimin neden gerekli olduğunu ve diğer gereksinim mühendisliği aktivitelerini nasıl etkilediğini anlamak.

# İçerik

- Fonksiyonel ve fonksiyonel olmayan gereksinimler
- Yazılım gereksinim dokümanı
- Gereksinim tanımı
- Gereksinim mühendisliği süreçleri
- Gereksinim çıkarma ve analiz
- Gereksinim doğrulama
- Gereksinim yönetimi

# Gereksinim Mühendisliği

- Gereksinim mühendisliği, müşterinin sistemden beklediği hizmetlerin tespit edildiği ve geliştirme ile çalışma üzerindeki kısıtların belirlendiği süreçtir
- Gereksinimler, gereksinim mühendisliği sürecinde oluşturulan sistem hizmetleri ve kısıtlarıdır

# Gereksinim Nedir?

- Yüksek seviyeli soyut bir ifade de olabilir, bir matematik fonksiyonunun tanımlaması da olabilir
- Yoruma açık olabilir
- Detaylı tanımlanmalıdır

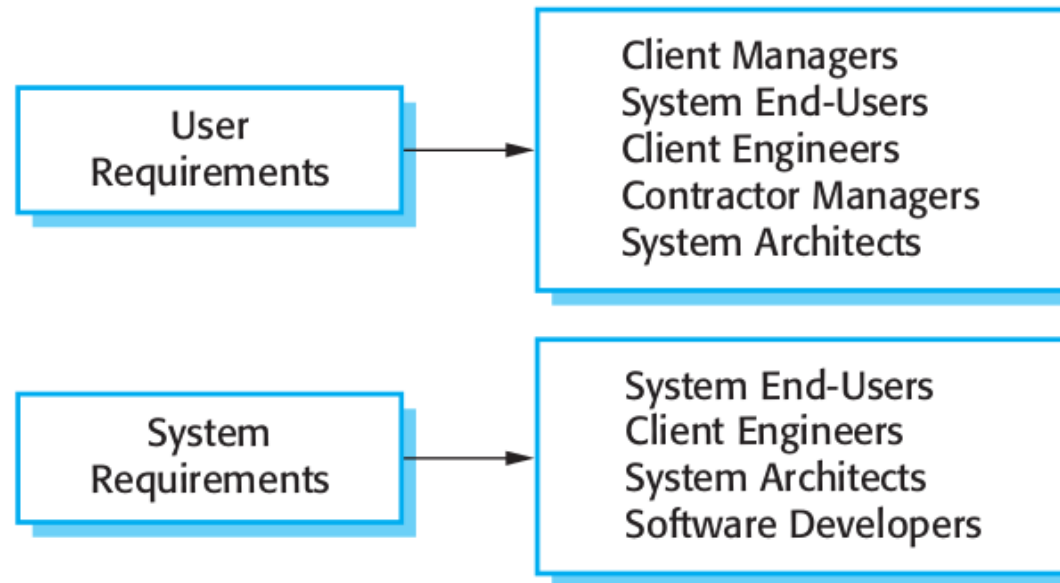
# Gereksinim Türleri

- ┌ Kullanıcı gereksinimleri
  - ┌ Doğal dil ve diyagramlarla anlatılmış, sistemin ne yapacağını ve kısıtlarını, kullanıcılara neler sağlayacağını anlatan ifadeler
- ▢ Sistem gereksinimleri
  - ▢ Sistemin fonksiyonel detaylarını tanımlayan yapısal bir doküman; müşteri ile yazılım firması arasındaki sözleşmeye konu olabilir

# Kullanıcı ve Sistem Gereksinimleri

- Psikiyatrik Hasta Yönetim Sistemi için
  - Kullanıcı gereksinimleri
    - Sistem aylık olarak, herbir klinik tarafından o ay içinde reçetelenmiş ilaçların maliyetini gösteren bir rapor üretmelidir.
  - Sistem Gereksinimleri
    - Her ayın son iş gününde reçetelenen ilaçlar, ücretleri, hangi klinik tarafından reçete edildiğini özetleyen bir rapor üretilecektir.
    - Sistem ayın son iş gününde saat 17.30'dan sonra görüntülenmek üzere otomatik olarak raporu üretir
    - Rapor her bir klinik için üretilecek olup, reçetelenen ilaç isimlerini, kaç defa reçete edildiklerini, kaç doz bilgisini, ve toplam tutarı içerecektir
    - Eğer ilaçların farklı dozlarda türleri varsa (örn. 10 mg, 20 mg), her doz birimi için ayrı rapor üretilecektir.
    - Ücret raporlarına erişim yetkilendirilmiş kullanıcılarla (yönetim erişimi olan) sınırlandırılacaktır.

# Kullanıcı ve Sistem Gereksinimleri Kimleri İlgilendirir?





# Fonksiyonel ve Fonksiyonel Olmayan Gereksinimler

## ▯ Fonksiyonel Gereksinimler

- ▯ Sistemin vermesi gereken hizmetler. Ne yapmaması gerektiği de bulunabilir.

## ▯ Fonksiyonel Olmayan Gereksinimler

- ▯ Geliştirme süresi, standartlar, zamanlama gibi gereksinimler.

# Fonksiyonel Gereksinimler

- ▯ Sistemin ne yapması gerektiğinin tanımlanması
- ▯ Geliştirilen yazılımın türüne, sistemi kaç kişinin kullanacağına, kuruluşun gereksinim hazırlama kültürüne bağlıdır
- ▯ Fonksiyonel kullanıcı gereksinimleri sistem kullanıcıları tarafından anlaşılabilir şekilde daha soyut olarak yalın bir dille ifade edilebilir
- ▯ Fonksiyonel sistem gereksinimleri daha detaylı ve teknik olarak ifade edilebilir

# Fonksiyonel Gereksinimler (Psikiyatri)

- ▯ Bir kullanıcı randevu aramasını tüm klinikler için yapabilmelidir
- ▯ Sistem herbir gün ve klinik için o gün randevusu olan hastaları listeleyebilmelidir
- ▯ Sistemi kullanan her çalışan sekiz haneli sicil numaraları ile tanımlanmalıdır

# Gereksinimlerdeki Problemler

- ▯ Gereksinimlerdeki belirsiz noktalar yazılım mühendisliği sürecinde birçok probleme neden olabilir
- ▯ Sistem geliştiriciler işleri basitleştirmek adına belirsiz gereksinimleri yanlış yorumlayabilir
- ▯ Psikiyatri sisteminde “arama” hem hasta hem de sağlıkçılar açısından farklı anlamlar taşıyabilir

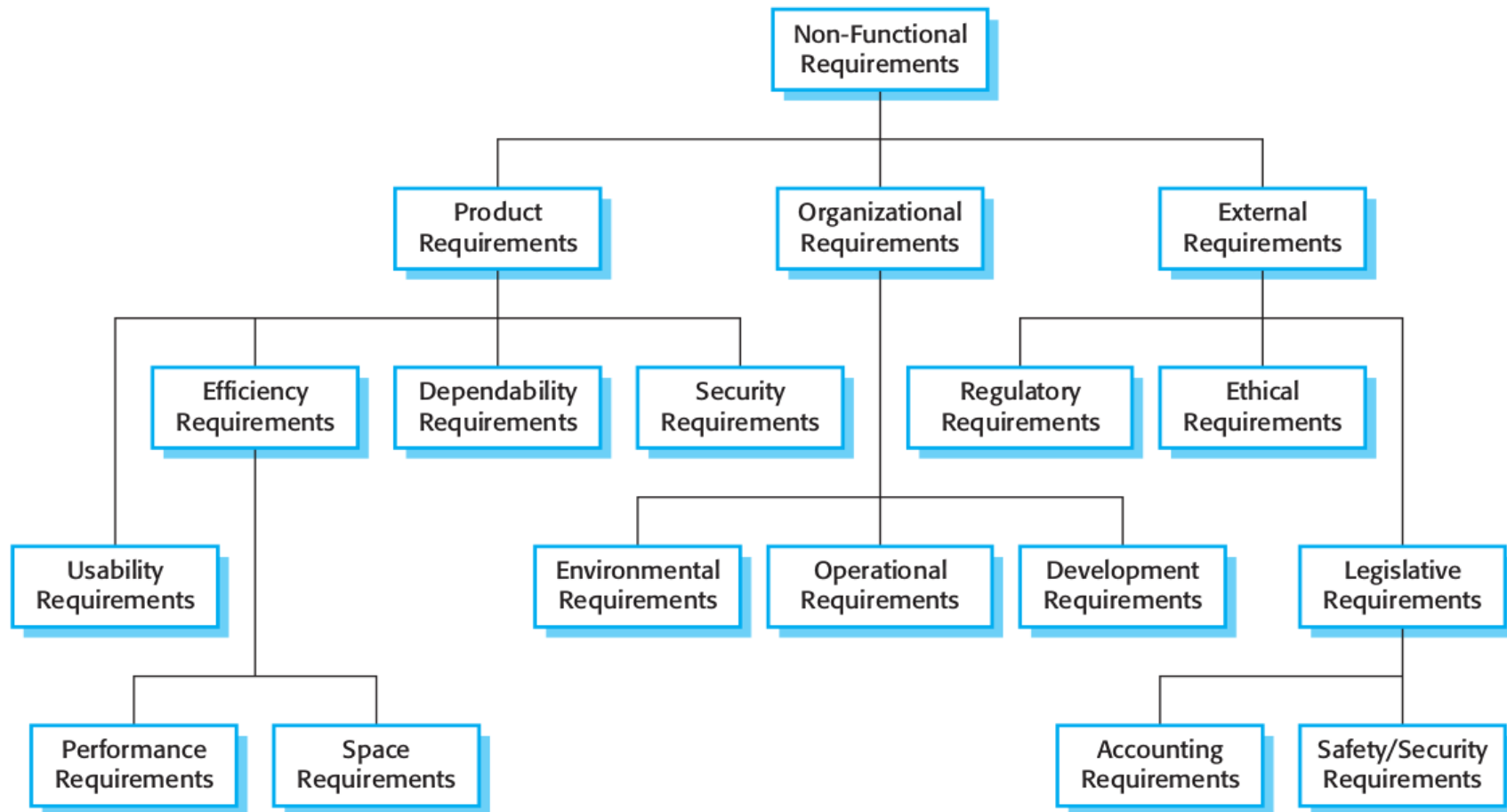
# Gereksinimlerin Eksiksiz ve Tutarlı Oluşu

- Teoride gereksinimler eksiksiz ve tutarlı olmalı
- Eksiksizlik
  - Gerekli olan herşeyi içermeli
- Tutarlılık
  - Gereksinimler arasında çelişki veya çatışma olmamalı
- Pratikte, büyük ve karmaşık sistemler için eksiksiz ve tutarlı bir gereksinim dokümanı hazırlamak imkansız
  - Büyük sistemlerde çok paydaş var
  - Karmaşık sistemlerde gereksinimler hatalı yazılabilir veya iptal edilebilir

# Fonksiyonel Olmayan Gereksinimler

- ▯ Güvenilirlik, yanıt süresi, depolama alanı ihtiyacı gibi sistem özelliklerini tanımlar
- ▯ Süreç gereksinimleri belirli bir IDE, programlama dili veya geliştirme yöntemini zorlayabilir
- ▯ Fonksiyonel olmayan gereksinimler, fonksiyonel gereksinimlerden daha kritik olabilir
- ▯ Karşılanmadıkları takdirde sistem kullanışsız hale gelebilir

# Fonksiyonel Olmayan Gereksinimlerin Sınıflandırılması



# Fonksiyonel Olmayan Gereksinimlerin Sınıflandırılması

- Ürünün gereksinimleri
  - Hız, güvenilirlik vb.
- Organizasyonel gereksinimler
  - Standartlar, geliştirme detayları vb.
- Dış kaynaklı gereksinimler
  - Beraber çalışabilirlik, yasal düzenlemeler vb.



# Psikiyatri Sistemi İçin Fonksiyonel Olmayan Gereksinimler

- Ürün Gereksinimi
  - Psikiyatri Sistemi tüm kliniklerde normal çalışma saatlerinde kullanılabilecektir (Pzt–Cum, 08.30–17.30). Normal çalışma saatleri içindeki aksaklıklar gün başına 5 saniyeyi geçmeyecektir.
- Organizasyonel Gereksinim
  - Psikiyatri Sisteminin kullanıcıları sağlık yetkilisi olduklarına dair bir kimlik kartı ile sisteme kendilerini tanıtır.
- Dış Kaynaklı Gereksinim
  - Sistem, 10.06.2006/135 nolu yönetmeliğin 2. maddesinde belirtilen hasta mahremiyeti koruma hükümlerini uygulamalıdır.

# Fonksiyonel Olmayan Gereksinimler ve Sistem Hedefleri

- Fonksiyonel olmayan gereksinimlerle ilgili genel sorun, müşterilerin veya kullanıcıların gereksinimleri çok genel hedefler olarak tanımlamasıdır. Örneğin, kolay kullanım, hata düzeltme yeteneği veya hızlı çalışma.
- Bir yöneticinin kullanışlılık gereksinimi olarak ortaya koyması gereken örnek sistem hedefi şu şekilde olmalıdır:
  - Sistem sağlık personeli tarafından kolayca kullanılabilmeli ve kullanıcı hatalarını en aza indirgeyen bir yöntemle düzenlenmelidir.

# Test Edilebilir Gereksinim Tanımı

- Gereksinim tanımı test edilebilir şekilde şöyle değiştirilebilir:
  - Sağlık personeli 4 saatlik eğitimden sonra sistemin bütün fonksiyonelliklerini kullanabilecektir. Bu eğitimden sonra kullanıcıların hata oranı 2 hata/saat'i geçmeyecektir.

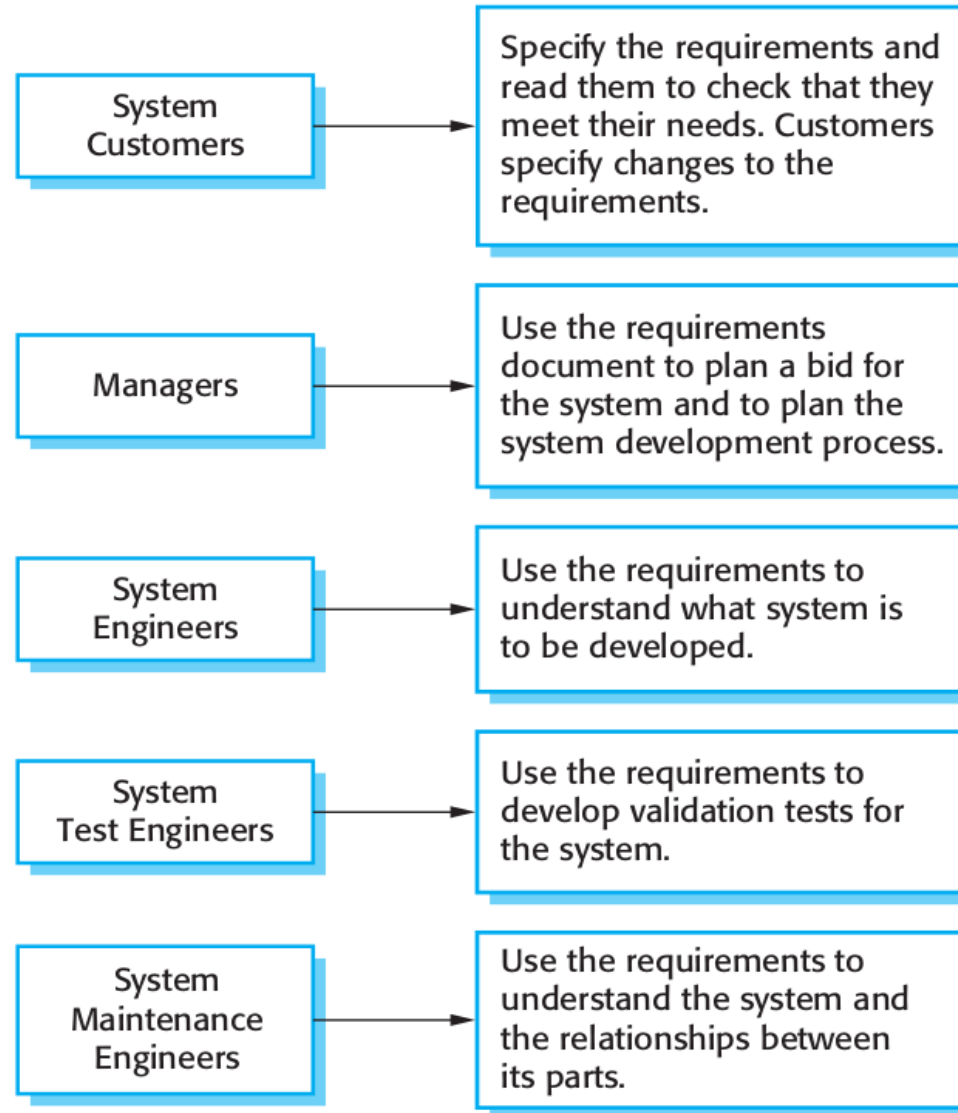
# Fonksiyonel Olmayan Gereksinimleri Tanımlama Ölçütleri

Property	Measure
Speed	Processed transactions/second User/event response time Screen refresh time
Size	Mbytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

# Yazılım Gereksinim Dokümanı

- Sistem geliştiricilerinin ihtiyaç duyduğu resmi doküman
- Kullanıcı gereksinimlerini ve sistem gereksinimlerinin detaylarını barındırmalı
- Tasarım dokümanı DEĞİLDİR. Sistemin NE yapması gerektiğini anlatır, NASIL yapması gerektiğini DEĞİL

# Gereksinim Dokümanını Kimler Kullanır?



# Gereksinim Dokümanını Kimler Kullanır?

- ▯ Müşteriler – Gereksinimleri belirler ve ihtiyaçlarını karşılayıp karşılamadığını kontrol etmek için dokümanı okurlar
- ▯ Yöneticiler – Bütçeyi ve geliştirim sürecini planlayabilmek adına kullanırlar
- ▯ Sistem Mühendisleri – Nasıl bir sistem geliştirilmesi gerektiğini anlamak için kullanırlar
- ▯ Sistem Test Mühendisleri – Doğrulama testlerini geliştirebilmek için kullanırlar
- ▯ Sistem Bakım Mühendisleri – Sistemi ve parçaları arasındaki ilişkileri anlamak için kullanırlar

# Gereksinim Dokümanının Yapısı

- **Önsöz** - Dokümanı kim okuyacak, sürüm tarihçesi, bu sürümde neler değişti
- **Giriş** – Sistem ihtiyaçları, basitçe sistemin fonksiyonlarını, sistemin diğer sistemlerle nasıl birlikte çalıştığı
- **Sözlük** – Teknik terimler
- **Kullanıcı gereksinim tanımlamaları** – Kullanıcı için sağlanan servisler, fonksiyonel olmayan sistem gereksinimleri, doğal dil veya müşterinin anlayabileceği notasyonlar kullanılır, ürün ve süreç standartlarından bahsedilir
- **Sistem mimarisi** – fonksiyonların sistem modüllerine dağılışı, tekrar kullanılabilir mimari bileşenleri



# Gereksinim Dokümanının Yapısı

- **Önsöz** - Dokümanı kim okuyacak, sürüm tarihçesi, bu sürümde neler değişti
- **Giriş** – Sistem ihtiyaçları, basitçe sistemin fonksiyonlarını, sistemin diğer sistemlerle nasıl birlikte çalıştığı
- **Sözlük** – Teknik terimler
- **Kullanıcı gereksinim tanımlamaları** – Kullanıcı için sağlanan servisler, fonksiyonel olmayan sistem gereksinimleri, doğal dil veya müşterinin anlayabileceği notasyonlar kullanılır, ürün ve süreç standartlarından bahsedilir
- **Sistem mimarisi** – fonksiyonların sistem modüllerine dağılışı, tekrar kullanılabilir mimari bileşenleri

# Gereksinim Dokümanının Yapısı

**Sistem gereksinimlerinin detayları** – Fonksiyonel ve fonksiyonel olmayan gereksinimler daha detaylı tanımlanır

**Sistem modelleri** – Sistem bileşenleri ile sistem ve çevresi arasındaki ilişkinin grafiksel modeli

**Sistemin Evrimi** - Değişen kullanıcı ihtiyaçları, donanımdaki gelişmeler ile ilgili öngörüler. Sistem tasarımcıları için kullanışlı bir bölüm

**Ekler** – Geliştirilen uygulama ile ilgili detaylı bilgiler, donanım ve veritabanı tanımları

**Index** – Alfabetik olabilir

# Gereksinim Tanımlama

- Kullanıcı gereksinimleri teknik olmayan insanların anlayabileceği şekilde ve dilde yazılmalı
- Sistem gereksinimleri daha detaylı olabilir ve teknik bilgiler içerebilir
- Gereksinimler sözleşmenin bir parçası olabileceği için mümkün olduğu kadar eksiksiz olmalı

# Gereksinim Tanımlama Yöntemleri

- ▯ Doğal dil
- ▯ Yapısal Doğal Dil
- ▯ Tasarım Tanımlama Dili
- ▯ Grafiksel Gösterimler
- ▯ Matematiksel Tanımlamalar

# Doğal Dil Kullanımı

- Standart bir format belirlemeli ve bütün gereksinimler için bu kullanılmalı
- Dil doğru şekilde kullanılmalı
  - Zorunluluklar için “yapmalı”
  - olsa iyi olur gereksinimleri için “yapılabilir” ifadeleri kullanılmalı
- Bilgisayar jargonu kullanmaktan kaçınılmalı
- Gereksinime neden ihtiyaç duyulduğuna dair mantıklı bir açıklama eklenmeli

# Doğal Dil Kullanımının Problemleri

- Yeteri düzeyde anlaşılır olmama
- Gereksinimlerin karışması
  - Fonksiyonel olanlar ve olmayanlar gibi
- Gereksinimleri birbirine girmesi
  - Birden fazla gereksinimin tek gereksinim gibi anlatılması

# Doğal Dil İle İnsülin Pompası İçin Gereksinim Hazırlama

- Sistem kan şekerini ölçmeli ve gerekliyse her 10 dakikada bir insülin vermelidir
- Sistem her dakika Tablo1'de tanımlanmış olan olaylara ilişkin kendini test etme rutinini çalıştırmalıdır (böylece donanım ve yazılım hataları farkedilebilir)

# Yapısal Doğal Dil Kullanımı

- Gereksinim yazmadaki özgürlüğü kısıtlandığı bir yöntemdir
- Gömülü sistemler gibi sistemler için gereksinim oluştururken faydalı olabilir ancak iş uygulamaları için fazla katı olabilir



# Form Tabanlı Uygulama

- Fonksiyonun/Varlığın tanımı
- Girişlerin tanımı
- Çıkışların tanımı
- Hesaplama için gerekli olan bilgi
- Sonuçta gerçekleştirilecek olan aksiyon
- Ön şartlar ve sonrasındaki şartlar

# Yapısal Doğal Dil İle İnsülin Pompası İçin Gereksinim Hazırlama

Insulin Pompası/Kontrol Yazılımı/SRS/3.3.2

**Fonksiyon** İnsülin dozunu hesaplama: güvenli şeker seviyesi

**Tanım** Ölçülen şeker düzeyi 3 ila 7 ünite arasındaki güvenli alanda iken gönderilecek insülin dozu

**Girdiler** O anki şeker düzeyi ( $r_2$ ); en son okunan iki şeker düzeyi ( $r_0$  ve  $r_1$ ).

**Kaynak** O anki şeker düzeyi sensörden diğer okumalar bellekten

**Çıktılar** CompDose— gönderilecek insülin dozu

**Hedef** Ana kontrol döngüsü.

# Sekmeli Tanımlama

- Doğal dili desteklemek için kullanılır
- Birkaç alternatif aksiyonu tanımlarken faydalı olabilir

# Sekmeli Tanımlama İle İnsülin Pompası İçin Gereksinim Hazırlama

Condition	Action
Sugar level falling ( $r_2 < r_1$ )	CompDose = 0
Sugar level stable ( $r_2 = r_1$ )	CompDose = 0
Sugar level increasing and rate of increase decreasing ( $(r_2 - r_1) < (r_1 - r_0)$ )	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ( $(r_2 - r_1) \geq (r_1 - r_0)$ )	CompDose = round $((r_2 - r_1)/4)$ If rounded result = 0 then CompDose = MinimumDose

# Gereksinim Mühendisliği Süreçleri

- ▯ Süreçler projeye göre, sahaya göre, müşteriye göre değişse de genel aktivitelerden söz edilebilir:
- ▯ Gereksinimlerin keşfi
- ▯ Gereksinimlerin analizi
- ▯ Gereksinimlerin doğrulanması
- ▯ Gereksinimlerin yönetimi
- ▯ Pratikte Gereksinim Mühendisliği, bu süreçlerinin iç içe geçtiği bir aktivitedir

# Gereksinimlerin Keşfi ve Analiz

- ▯ Teknik kadronun müşteri ile beraber çalışmasını gerektirir ve uygulama sahasının sistemin sağlayacağı servislerin ve sistemin operasyonel kısıtlarının belirlenmesini içerir
- ▯ Son kullanıcılar, yöneticiler, saha uzmanları, sendikalar vb. unsurlar süreçte bulunabilir. Bunlara genel olarak paydaş denir

# Gereksinim Analizindeki Problemler

- ▯ Paydaşlar gerçekten ne istediklerini bilmiyor olabilirler
- ▯ Paydaşlar ihtiyaçlarını kendi terimleri ile anlatırlar
- ▯ Farklı paydaşların çatışan ihtiyaçları olabilir
- ▯ Analiz sürecinde gereksinimler değişebilir. Yeni paydaşlar katılabilir

# Gereksinimlerin Keşfi ve Analiz

## ▯ Aşamaları

- ▯ Gereksinimleri ortaya çıkarma
  - ▯ Paydaşlarla görüşme
- ▯ Gereksinimleri sınıflandırma ve organize etme
  - ▯ Amaçlarına göre sınıflandırma
- ▯ Gereksinimleri önceliklendirme
  - ▯ Önceliklendirme ve çatışmaları çözme
- ▯ Gereksinimleri detaylandırma
  - ▯ Belgeleme



# Gereksinimlerin Ortaya Çıkarılması

- ▯ İstenen ve var olan sistemler hakkında bilgileri toplayıp bu bilgilerden sistem ve kullanıcı gereksinimleri filtrelenebilir
- ▯ Paydaşlarla irtibat kurulur ve bu paydaşlar birden çok olabilir

# Psikiyatri Sisteminin Paydaşları

- ▯ Hastalar- Sistemde bilgileri kayıtlı olan hastalar
- ▯ Doktorlar – Hastaları muayene ve tedavi etmekten sorumlu kişiler
- ▯ Hemşireler – Doktorların muayenelerini koordine ederler ve bazı tedavileri gerçekleştirirler
- ▯ Tıbbi resepsiyonistler – Hasta randevularını yönetirler
- ▯ IT çalışanları – sistemin yüklenmesi ve bakımından sorumlu kişiler

# Psikiyatri Sisteminin Paydaşları

- ▯ Tıbbi etik yöneticisi – sistemin hasta bakımı ile ilgili etik kurallarını uyguladığını garanti eden kişi
- ▯ Sağlık yöneticileri – Bilgi sisteminin yönetim kısmından yararlanan kişiler
- ▯ Tıbbi kayıt çalışanları – Sistemin kayıt prosedürlerini doğru olarak yerine getirip getirmediğini kontrol eden kişiler

# Görüşme/Mülakat

- Paydaşlarla yapılan resmi veya resmi olmayan görüşmeler
- Görüşme tipleri
  - Önceden belirlenmiş sorulardan oluşan kapalı görüşme
  - Paydaşlarla çeşitli konuların keşfedildiği açık görüşme
- Etkili Görüşme
  - Yeni görüşlere açık olun. Gereksinim toplamayla ilgili önyargılı olmayın ve paydaşları dinlemeye istekli olun
  - Görüştüğünüz kişinin, sorular üzerinden, bir gereksinim dokümanı taslağından veya bir prototipten yola çıkarak açıklayıcı olmasını sağlayın

# Pratikte Görüşmeler

- Açık ve Kapalı görüşmelerin karışımı şeklinde olur.
- Görüşmeler, sistemin geneli ile ilgili bilgileri almak için ve kullanıcıların sistemden ne beklediklerini anlamak için faydalıdır.
- Alan gereksinimlerini anlamak için faydalı değildir
  - Gereksinim mühendisleri alan terminolojisini anlamazlar
  - Bazı alan bilgilerini anlatmak, o işin uygulayıcıları için çok zor olabilir ya da kolayca atlanabilir

# Senaryolar

- ▯ Senaryolar, bir sistemin nasıl kullanılacağını gerçek örnekleridir
- ▯ Bir senaryo şunları içermelidir
  - ▯ Başlangıç durumunun bir tanımı
  - ▯ Olayların normal akışları
  - ▯ Nelerin yanlış gidebileceği
  - ▯ Eş zamanlı diğer aktiviteler ile ilgili bilgiler
  - ▯ Senaryo bittiğindeki durumun tarifi

# Psikiyatri Sistemi İçin Hasta Öyküsü Toplama Senaryosu

**İlk Varsayımlar:** Tıbbi kayıt personeli hastanın bilgilerini sisteme girer. Bir hemşire sisteme giriş yapar ve tıbbi öykü toplar.

**Normal:** Hemşire soyadına göre hasta arar. Aynı soyadıyla birden fazla hasta varsa hastanın adı ve doğum tarihi ile hastayı bulur. Hemşire tıbbi öykü eklemek için menüden seçim yapar. Daha sonra sistem adımlarını takip ederek hastanın mevcut hastalıklarını (menüden seçerek), kullandığı ilaçları (menüden seçerek), alerjilerini (metin) ve aile hayatını (form) sisteme girer.

**Neler Yanlış Gidebilir:** Hasta kaydı bulunamayabilir. Hemşire yeni kayıt oluşturmalıdır. Menüdekiler dışında hastalıklar veya kullanılan ilaçlar olabilir. Hemşire bu durumlarda “diğer” seçeneğini seçip metin kutusuna bilgileri girmelidir. Hasta tıbbi geçmişi ile ilgili bilgi ver(e)meyebilir. Hemşire bu durumu bir metin kutusuna yazmalıdır.

# Psikiyatri Sistemi İçin Hasta Öyküsü Toplama Senaryosu

**Diğer Aktiviteler:** Bir kayıt girilmekteyken diğer sağlık personelleri kaydı görebilir ancak güncelleyemez.

**İşlem Bitince Sistem Durumu:** Kullanıcı sisteme girmiş olur. Hasta kaydı veritabanına yazılır, sistem loguna bir kayıt eklenir (o oturumun başlangıç ve bitiş zamanı ve hemşire)

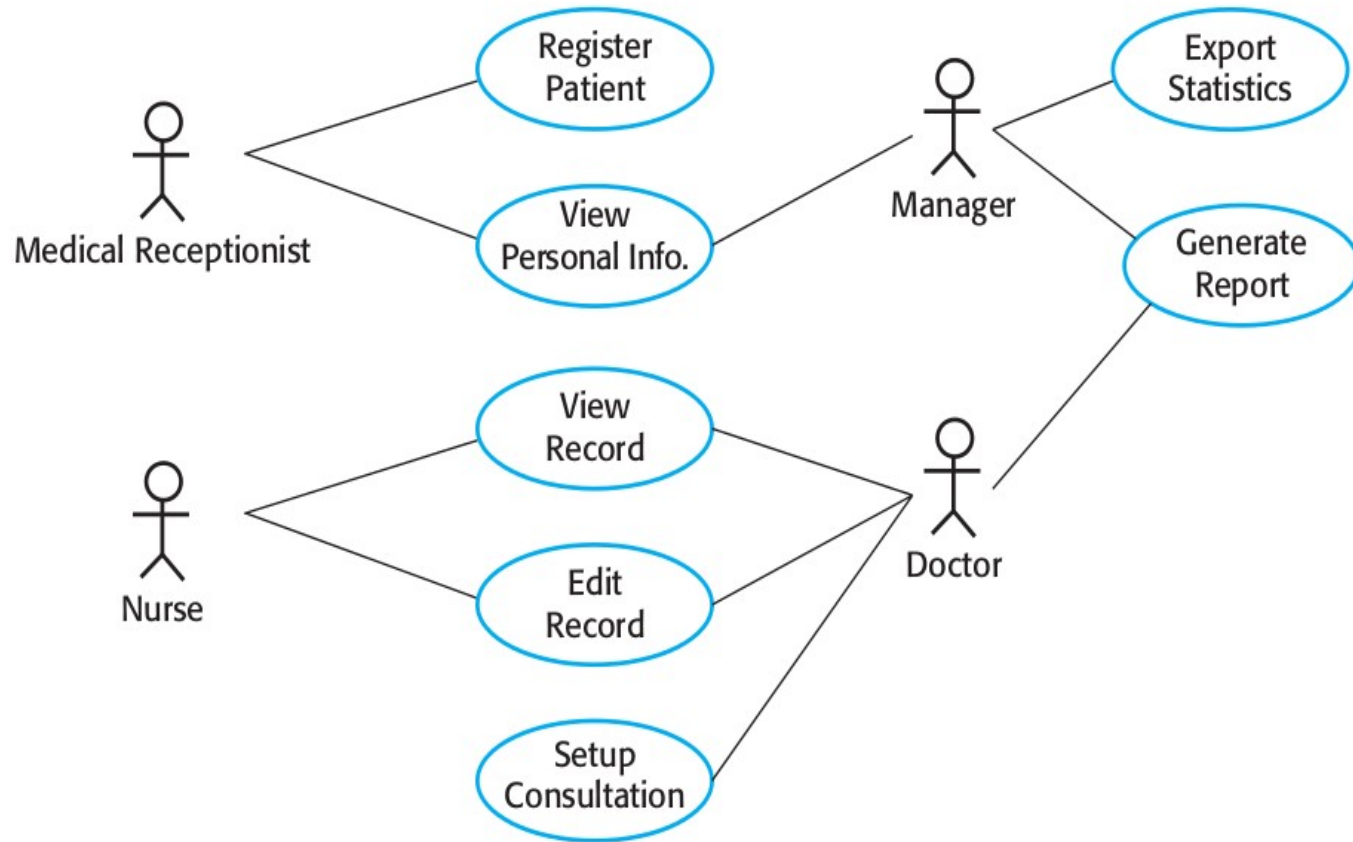


# Kullanım Durumları (Use Cases)

- ▯ Kullanım durumları, UML içerisinde senaryo tabanlı olarak kullanılan ve aktörler ile sistem arasındaki etkileşimlerin tanımlandığı varlıklardır
- ▯ Kullanım durumlarının bir kümesi ile sistemin bütün olası kullanım senaryoları tanımlanmalıdır
- ▯ Yüksek seviyeli grafiksel modeller, sekmeli tanımlama ile desteklenmelidir
- ▯ Daha detaylı bilgi eklemek için Sequence diyagramları kullanılabilir

# Psikiyatri Sistemi İçin Kullanım Durumları

□



# Ethnography (iş yapma alışkanlıklarını anlama)

- Bir sosyal bilimci, insanların gerçekten nasıl iş yaptıklarını anlamak için uzun zamanlar harcar
- İnsanlar ne yaptıklarını anlatmak zorunda değildirler
- Sosyal ve organizasyonel faktörlerin önemi daha iyi anlaşılır
- Etnografik çalışmalar göstermiştir ki; çalışma biçimleri, önerilen belirli modellerden daha karmaşık ve zengindir

# Etnografinin Kapsamı

- Gereksinimler, insanların anlattıkları haliyle değil yaptıkları şekliyle toplanır.
- Gereksinimler, işbirliği ve diğer insanların aktivitelerinin farkında olunarak toplanır
  - Başka insanların yaptıkları, benim yapacaklarımda neleri değiştirir?
- Etnografi, var olan işleyişi anlamak için oldukça faydalıdır ancak sisteme yeni işlevsellikler eklemek için yetersiz kalır

# Etnografinin Kapsamı

- Gereksinimler, insanların anlattıkları haliyle değil yaptıkları şekliyle toplanır.
- Gereksinimler, işbirliği ve diğer insanların aktivitelerinin farkında olunarak toplanır
  - Başka insanların yaptıkları, benim yapacaklarımda neleri değiştirir?
- Etnografi, var olan işleyişi anlamak için oldukça faydalıdır ancak sisteme yeni işlevsellikler eklemek için yetersiz kalır

# Gereksinimlerin Doğrulanması

- ▯ Gereksinimler gerçekten müşterinin istediği sistemi tarif ediyor mu?
- ▯ Gereksinimlerdeki hatalar pahalıya malolabilir.
  - ▯ Teslimden sonraki bir gereksinim hatasını düzeltmek, geliştirme aşamasında düzeltmekten 100 kat daha pahalı

# Gereksinim Kontrolleri

- ▯ Geçerlilik: Sistem, müşterinin ihtiyaç duyduğu fonksiyonellikleri en iyi biçimde sağlıyor mu?
- ▯ Bütünlük: Gereksinimler tutarlı mı?
- ▯ Tamlık: Müşterinin istediği bütün fonksiyonellikler var mı?
- ▯ Gerçekçilik: Gereksinimler, belirlenen bütçe ve teknoloji ile yapılabilir mi?
- ▯ Doğrulanabilirlik: Gereksinimler kontrol edilebilir mi?

# Gereksinim Doğrulama Teknikleri

- ▯ **Gözden geçirme:** Sistematik olarak gereksinimlerin elle analiz edilmesi
- ▯ **Prototipleme:** Sistemin çalıştırılabilir bir modeli ile gereksinimleri kontrol etme
- ▯ **Test durumu oluşturma:** Gereksinimler için testler oluşturma – gereksinimler test edilebilir olmalıdır



# Gereksinimlerin Gözden Geçirilmesi

- ▯ Gereksinim tanımları formülize edilirken düzenli gözden geçirmeler uygulanmalı
- ▯ Müşteri ve yazılım firmasının yetkilileri katılmalı
- ▯ Gözden geçirmeler biçimsel (tamamlanmış bir doküman üzerinden) olabilir de olmayabilir de. İyi bir iletişim, problemleri erken safhalarda çözebilir

# Gözden Geçirme Kontrolleri

- Doğrulanabilirlik
  - Gereksinim gerçekçi biçimde test edilebilir mi?
- Anlaşılabilirlik
  - Gereksinim düzgün biçimde anlaşılıyor mu?
- İzlenebilirlik
  - Gereksinimin nerden çıktığı açıkça belirtilmiş mi?
- Adaptasyon
  - Gereksinim, diğer gereksinimler üzerinde büyük bir etki yaratmadan değiştirilebilir mi?

# Gereksinimlerin Yönetimi

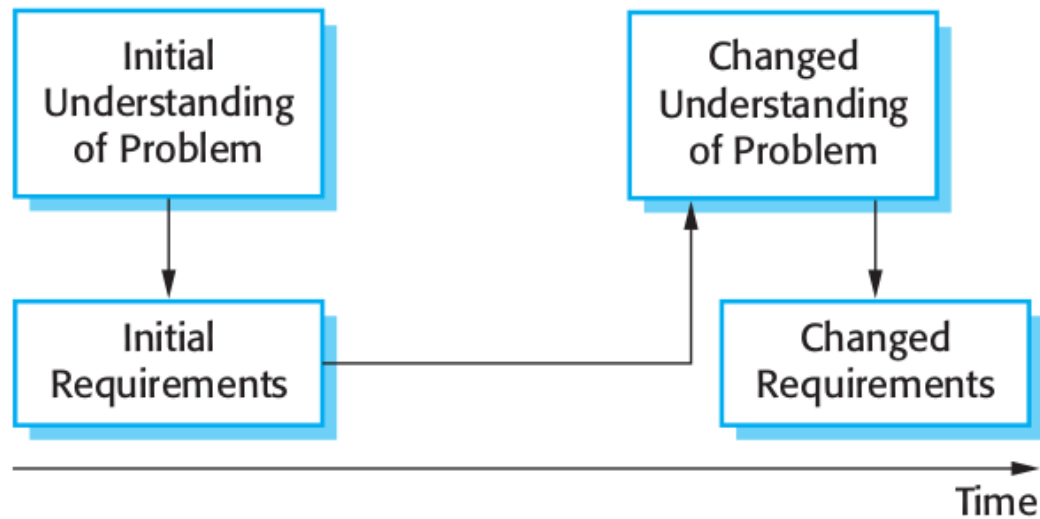
- ▯ Değişen gereksinimlerin, gereksinim mühendisliği ve sistemin geliştirilmesi aşamalarında yönetilmesi
- ▯ Bağımsız gereksinimlerin izleri tutulmalı, bağımlı gereksinimlerin ise bağılılıkları tanımlanmalıdır. Böylece, gereksinimler üzerindeki değişikliklerin etkileri değerlendirilebilir

# Gereksinimlerin Değişimi

- Sistemin teknik ve iş ortamları, yüklemekten sonra mutlaka değişir
  - Yeni donanım eklenebilir. Başka sistemlerle çalışması gerekebilir. İş öncelikleri değişebilir. Yeni mevzuat düzenlemeleri gelebilir
- Bir sistem için parayı ödeyenlerle sistemi kullananlar çoğu zaman farklıdır
  - Öncelikler çatışabilir
- Geniş sistemlerdeki paydaşların gereksinim öncelikleri farklı olabilir ve aralarında çelişki bulunabilir
  - Farklı kullanıcılara verilmesi gereken destek değişecektir

# Gereksinimlerin Evrimi

□



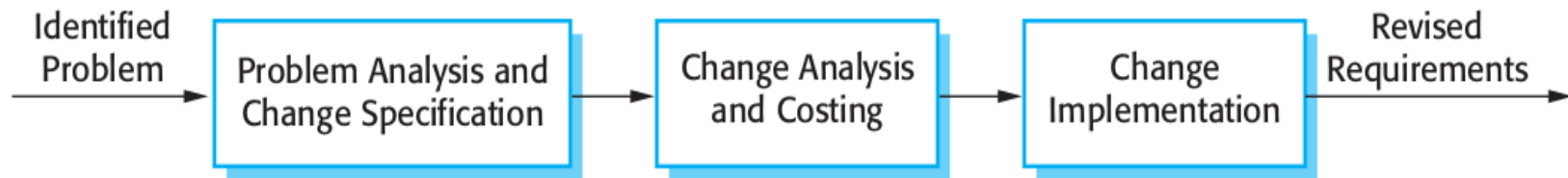
# Gereksinimlerin Yönetim Planı

- Gereksinim Yönetiminde karar verilmesi gereken şeyler
  - Gereksinimlerin kimliklendirilmesi
  - Bir değişim yönetim süreci: değişikliklerin etkilerini ve maliyetini değerlendirecek aktiviteler.
  - İzlenebilirlik ilkeleri: Her bir gereksinimin ilişkisini tanımlayan ilkeler
  - Araç desteği

# Gereksinim Değişim Yönetimi

- Bir gereksinim değişiminin kabulüne karar vermek
  - Problem analizi
- Gereksinim analizi ve maliyet
- İzlenebilirlik ilkeleri ile maliyeti ve uygulanabilirliği değerlendirme
- Gereksinimin hayata geçirilmesi
  - Gereksinimi değiştirme ve ilgili dokümanları güncelleme

# Gereksinim Değişim Yönetimi





Haftaya görüşmek üzere