

Gereksinim Mühendisliği

Süreçleri

Gereksinimlerin Tanımlanması

- Sistem servislerinin ve kısıtlarının bulunması, analiz edilmesi, kaydedilmesi ve kontrol edilmesi süreci Gereksinim Mühendisliği olarak tanımlanır.
- Sistem gereksinimleri müşterilerin gereksinimlerini yansıtır.
- Kullanıcı ve sistem gereksinimleri vardır.
- Sistem hakkında bilginin farklı kullanıcılara aktarılabilmesi için farklı türlerde gereksinimlere ihtiyaç duyulabilir.

İhtiyaç Türleri

- Kullanıcı Gereksinimleri; Doğal dildeki ifadeler ve sistemin sağladığı hizmet şemaları ve operasyonel kısıtları. Müşteri işbirliği ile yazılır.
- Sistem Gereksinimleri; Sistemin işlevlerinin, hizmetlerinin ve operasyon kısıtlarının ayrıntılı ortaya konularak oluşturulan yapılandırılmış bir belgedir. Neyin uygulanacağını tanımlar ve müşteri ile firma arasında sözleşme parçası olabilir.
- Kullanıcı gereksinimleri daha genel olurken sistem gereksinimleri sistem servis ve fonksiyonları için daha özgün bilgi sunar.

Kullanıcı Gereksinimleri

Sistem her ay klinik için reçetelenen ilaçların maliyetini gösteren aylık yönetim raporları üretilmelidir.

Sistem Gereksinimleri

1. Her ayın son çalışma gününde, reçetelenen ilaçların, maliyetlerinin, hangi klinik tarafından reçetelendirildiğinin gösterildiği özet rapor hazırlanmalıdır.
2. Sistem raporu basılabilmesi için ayın son çalışma günü saat 17.30'dan sonra hazırlanmalıdır.
3. Rapor her klinik için hazırlanmalıdır ve tek tek ilaç isimlerini, toplam reçete sayısını, reçetelenen doz sayısını ve reçetelenen ilaçların toplam maliyetini listelemelidir.
4. Eğer ilaç farklı dozlarda bulunabiliyorsa her doz birimi için ayrı rapor hazırlanmalıdır.

Gereksinimleri Okuyucuları

- Kullanıcı gereksinimleri;
 - Müşteri yöneticileri
 - Son kullanıcılar
 - Müşteri mühendisleri
 - Yüklenici yöneticileri
 - Sistem mimarları
- Sistem gereksinimleri;
 - Son kullanıcılar
 - Müşteri mühendisleri
 - Sistem mimarları
 - Yazılım geliştiriciler

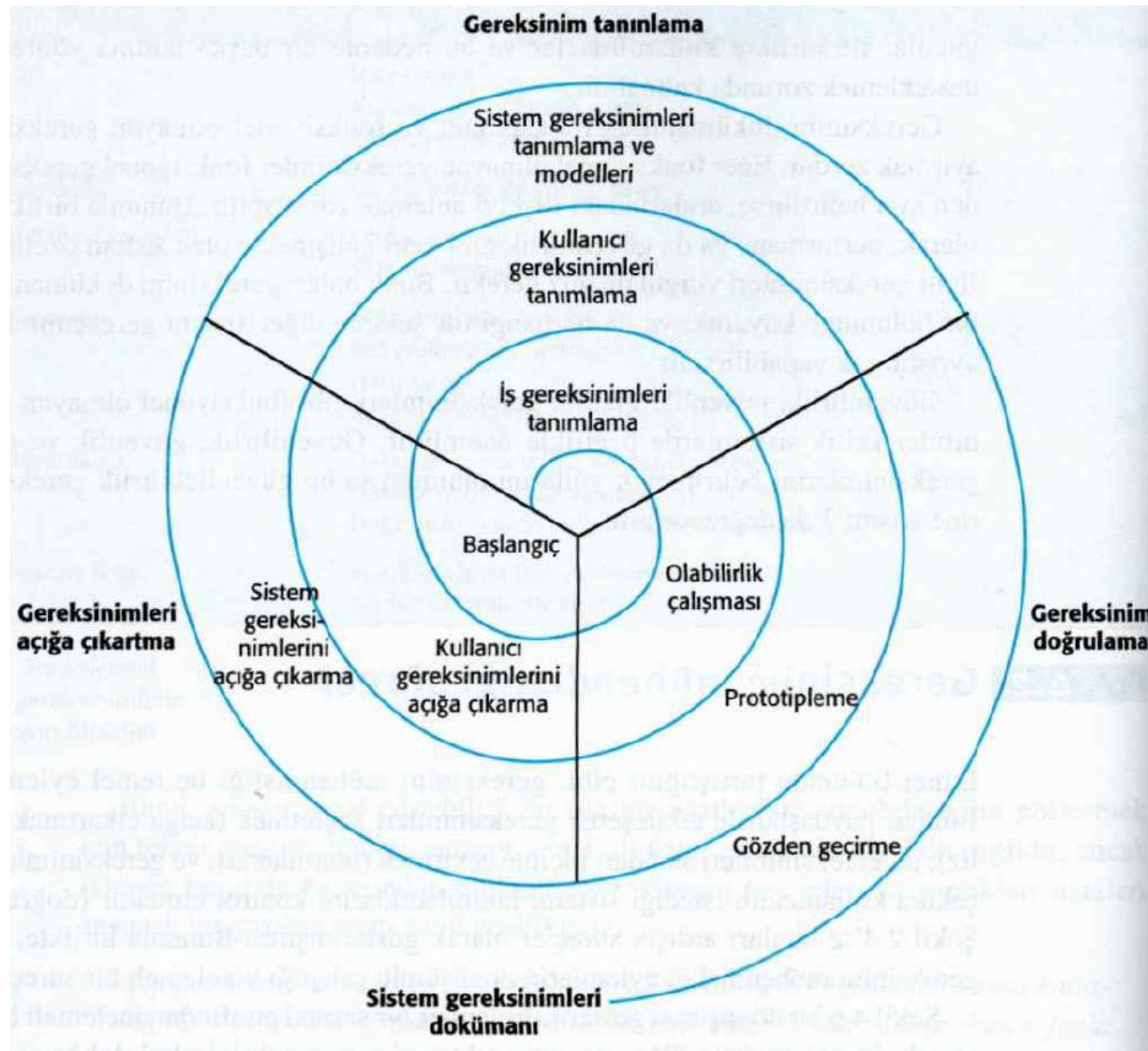
- Erken-faz gereksinim mühendisliği sayesinde sistemin teknik ve parasal olabilirliği değerlendirilebilir.
- Çevik yaklaşımlarda gereksinimler sistem geliştirimi ile eş zamanlı yapılır.

Gereksinim Mühendisliği

Süreçleri

- Gereksinim mühendisliği için kullanılan süreçler, uygulama alanına, ilgili kişilere ve gereksinimleri geliştiren kuruluşa bağlı olarak büyük ölçüde değişir.
- Bununla birlikte, tüm süreçler için ortak olan birkaç genel faaliyet vardır.
 - Gereksinimlerin açığa çıkarılması
 - Gereksinimlerin analizi
 - Gereksinimlerin onaylanması (doğrulama)
 - Gereksinimlerin yönetilmesi
- Pratikte, gereksinim mühendisliği, bu süreçlerin araya eklendiği yinelemeli bir faaliyettir.

Ekstrüzyon Mühendisliği Sürecinin Sarmal (Spiral) Görünümü



Gereksinimleri Açığa Çıkarma

- Gereksinim keşfi olarak da adlandırılır. İlk aşama olması sebebiyle sıkıntıları vardır.
- Yazılım mühendisleri, uygulama alanı, sistemin sağlaması gereken hizmetler, gerekli sistem performansı, donanım kısıtlamaları, diğer sistemler vb. hakkında bilgi edinmek için bir dizi sistem paydaşıyla birlikte çalışır.
- Uygulama alanı, sistemin sunması gereken hizmetler ve sistemin operasyonel kısıtlamaları hakkında bilgi edinmek için müşterilerle birlikte çalışan teknik personeli içerir.
- Son kullanıcıları, yöneticileri, bakımla ilgilenen mühendisleri, alan uzmanlarını, sendikaları vb. içerebilir. Bunlara paydaşlar denir.
- Gereksinimleri basit bir dil veya grafikler kullanılmalıdır.

- Gereksinimlerin açığa çıkarılması ve analiz süreci;
 - Gereksinimlerin bulunması ve anlaşılması
 - Gereksinimlerin sınıflandırılması ve düzenlenmesi
 - Gereksinimleri önceliklendirme ve uzlaşma
 - Gereksinimlerin dökümantasyonu

Gereksinimlerin Açığa Çıkarılması Teknikleri

- Önerilen sistem ile ilgili bilginin bulunması için farklı paydaşlar ile toplantılar yapılmasını içerir. Bu bilgi;
 - Mevcut sistem bilgisi
 - Paydaşların o bilgiyi kullanımı
 - Dökümanlardan gelen bilgi olabilir

- Gereksinimleri açığa çıkarmanın iki temel yaklaşımı vardır;
- Karşılıklı görüşme: İnsanlarla ne yaptıklarını konuşursunuz.
- Gözlem ya da etnografya: İnsanları kullandıkları araçları ve onları nasıl kullandıklarını görmek için izlersiniz.

Karşılıklı Görüşme

- Görüşmenin türleri;
 - Kapalı görüşmeler; Önceden belirlenmiş soru listesine dayalı görüşmeler.
 - Açık görüşmeler: Paydaşlarla çeşitli konuların araştırıldığı görüşmeler.

- Karşılıklı görüşmenin zorlukları;
- Uygulama uzmanları, kendi çalışma alanlarına özgü dil kullanır.
- Paydaşlar bazı alan gereksinimlerine çok aşina olduklarından ya açıklamakta zorlanırlar yada bahsetmeye değmeyecek kadar temel olduğunu düşünürler.
- Kurulustaki bazı insanlar arasında var olan güç ilişkisi sebebiyle sıkıntılıdır.

- İyi bir karşılıklı görüşmeci;
 - Açık fikirli olmalıdır.
 - Paydaşları dinlemeye istekli olmalıdır.
 - Paydaşlar sürpriz gereksinimlerle gelirse değişime açık olunmalıdır.
 - Paydaşları çeşitli sorular, prototipler üzerinde çalışarak harekete geçirilmelidir.
 - Paydaşlarla istediğini anlat demek yerine belirli bir içerik üzerinde konuşmalıdır.

Gözlem ya da Etnografya

- Bir çok yazılım sisteminin hiç kullanılmama sebebi, gereksinimlerinin sosyal ve organizasyonel etkenlerin sistemin pratik işleyişini nasıl etkilediğinin dikkate alınmamasındandır.
- Bunu önlemek için sosyal ve örgütsel önemli faktörler gözlemlenebilir.
- Kişiler ne kadar kendi işlerini anlasa da kuruluştaki diğer işlerle ilişkisini anlayamayabilir.
- Analist, insanların gerçekte nasıl çalıştığını gözlemek ve analiz etmek için kendisini sistemin kullanılacağı iş ortamına yerleştirir.
- İnsanların işlerini açıklamaları veya ifade etmeleri gerekmez. İnsanların farklı çalışma biçimlerini yansıtan üstü kapalı sistem gereksinimleri belirlenir.
- Etnografik araştırmalar, çalışmanın basit sistem modellerinin önerdiğinden genellikle daha zengin ve daha karmaşık olduğunu göstermiştir.

- İki tip gereksinimin bulunmasında etkilidir;
- İnsanların nasıl çalışmaları gerektiğini söyleyen iş süreç tanımları yerine gerçekte çalışma biçiminden türetilen gereksinimler.
- İş birliği ve diğer insanların eylemlerinin farkındalığından çıkarılan gereksinimler.
- Etnografya, mevcut süreçleri anlamak için etkilidir ancak bir sisteme eklenmesi gereken yeni özellikleri belirleyemez.

Hikayeler ve Senaryolar

- Senaryolar ve kullanıcı hikayeleri, bir sistemin nasıl kullanılabileceğinin gerçek hayattan örnekleridir.
- İnsanların ne yaptıklarını, ne tür bilgi kullandıklarını ve ürettiklerini ve bu süreç sırasında hangi sistemleri kullanabileceklerini tanımlar.
- Hikayeler öyküsel metinler olarak yazılır ve sistem kullanımının üst düzey tanımını sunar. Senaryolar genellikle girdiler, çıktılar gibi toplanmış özgün bilgi ile yapılandırılır.
- Pratik bir duruma dayandıkları için, paydaşlar onlarla ilişki kurabilir ve hikayeye ilgili durumları hakkında yorum yapabilirler.

- Senaryolar planlanmış bir kullanıcı hikayesidir.
- Senaryolar şunları içermelidir;
 - Senaryo başladığında sistem ve kullanıcıların neyi beklediklerinin tanımı.
 - Normal olay akışının tanımı.
 - Neyin yanlış gidebileceğinin ve oluşacak problemlerin nasıl ele alınacağının belirtilmesi.
 - Diğer eşzamanlı faaliyetler hakkında bilgi.
 - Senaryo bittiğinde sistemin durumun belirtilmesi.

Gereksinimlerin Tutarsızlığı

- Gereksinimler tam olarak belirlenmediğinde sorunlar ortaya çıkar.
- Belirsiz gereksinimler kullanıcı ve geliştiriciler tarafından farklı yorumlanabilir.

Gereksinimlerin Spesifikasyonu

- Kullanıcı ve sistem gereksinimlerinin bir gereksinim dökümanına yazılması sürecidir.
- Kullanıcı gereksinimleri, teknik bir geçmişi olmayan son kullanıcılar ve müşteriler tarafından anlaşılabilir olmalıdır.
- Sistem gereksinimleri daha ayrıntılı gereksinimlerdir ve daha fazla teknik bilgi içerebilir. Formlar, grafiksel ve matematiksel modellerle gösterime ihtiyaç duyabilir.
- İşlevsel ve işlevsel olmayan gereksinimleri içermelidir.
- Gereksinimler, sistem geliştirme sözleşmesinin parçası olabilir. Bu nedenle, bunların olabildiğince eksiksiz olması önemlidir.

- Gereksinimler sistemin ne yapması gerektiğini belirtmeli ve tasarım bunu nasıl yaptığını açıklamalıdır.
- Pratikte, gereksinimler ve tasarım birbirinden ayrılamaz. Nedenleri;
 - Gereksinimleri yapılandırmak için sistemin bir ilk tasarımı gerçekleştirilebilir.
 - Sistem, tasarım gereksinimleri oluşturan alt sistemlerle göre düzenlenebilir.
 - Mevcut sistem ile çalışması gerekliliği tasarımı kısıtlar ve yeni sistem gereksinimlerini zorlar.
 - Fonksiyonel olmayan gereksinimleri karşılamak için belli bir mimari kullanılması gerekebilir.

Sistem Gereksinimlerini Yazmak İçin;

Doğal Dil Cümleleri

Gereksinimler, doğal dilde numaralandırılmış cümleler kullanılarak yazılmıştır. Her cümle bir şartı ifade etmelidir.

Yapısal Doğal Dil

Gereksinimler, standart bir form veya şablon üzerinde doğal dilde yazılmıştır. Her alan, gereksinimin bir yönü hakkında bilgi sağlar.

Grafik Gösterim

Metin açıklamalarıyla desteklenen grafik modeller, sistemin fonksiyonel gereksinimlerini tanımlamak için kullanılır; UML kullanım durumu ve sıra diyagramları yaygın olarak kullanılır.

Matematiksel
Spesifikasyon

Bu gösterimler, sonlu durum makineleri veya kümeleri gibi matematiksel kavramlara dayanır. Bu kesin tanımlar gereksinimler belgesindeki belirsizliği azaltabilse de, çoğu müşteri resmi bir tanımlı anlamamaktadır. İstediklerini ifade edip etmediğini kontrol edemezler ve bunu bir sistem sözleşmesi olarak kabul etme konusunda isteksizdirler.

Doğal Dil ile Gereksinimleri Yazmak İçin;

- Standart bir format icat edin ve tüm gereksinimler için kullanın.
- Dili tutarlı bir şekilde kullanın. Zorunlu gereksinimler için ve istenen gereksinimler için kullanılmalıdır.
- Gereksinimin önemli kısımlarını belirlemek için metin vurgulamayı kullanın.
- Bilgisayar terimleri kullanmaktan kaçının.
- Bir gereksinimin neden gerekli olduğuna dair bir açıklama (mantık) ekleyin.

Yapısal Doğal Dil

- Gereksinimlerin standart bir şekilde yazıldığı gereksinim yazma şeklidir. Sistem gereksinimlerini belirtmek için şablonlar kullanır.
- Bu, bazı gereksinim türleri için uygundur (örneğin gömülü kontrol sistemi gereksinimleri), ancak bazen iş sistemi gereksinimlerini yazmak için çok katıdır.

•

Fonksiyonel gereksinimleri belirlemek için standart bir form kullanıldığında şu bilgileri içermelidir;

- Fonksiyonun veya nesnenin tanımı.
- Girdilerin ve girdi kaynaklarının tanımı.
- Çıktıların ve çıktıların nereye gittiğinin tanımı.
- Hesaplama için gerekli bilgiler ve kullanılan diğer varlıklar hakkında bilgiler.
- Yapılacak eylemin tanımı.
- Fonksiyon çağrılmadan önce ve çağrıldıktan sonra ön ve son koşullar (uygunsa).
- Operasyonun yan etkileri (varsa).

İnsülin Pompası/Denetleme Yazılımı/SRS/3.3.2

Fonksiyon	İnsülin dozunu hesapla: Güvenli şeker düzeyi.
Tanım	O an ölçülen şeker düzeyi 3 ile 7 birim arasındaki güvenli dilimdeyken verilmesi gereken insülin dozunu hesaplar.
Girdiler	O anki şeker okuması (r_2), önceki iki okuma (r_0 ve r_1).
Kaynak	Sensörden gelen o anki şeker okuması. Bellekten gelen diğer okumalar.
Çıktılar	CompDose-Verilmesi gereken insülin dozu.
Hedef	Ana denetleme döngüsü.
Eylem	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).
Gerektirir	Şeker düzeyi değişim hızının hesaplanabilmesi için önceki iki okuma.
Ön koşul	İnsülin deposu en az tek dozda en çok verilebilecek doz kadar insülin içerir.
Son koşul	r_0 r_1 ile değiştirilir sonra r_1 r_2 ile değiştirilir.
Yan etkileri	Yok

Tablo Tanımı

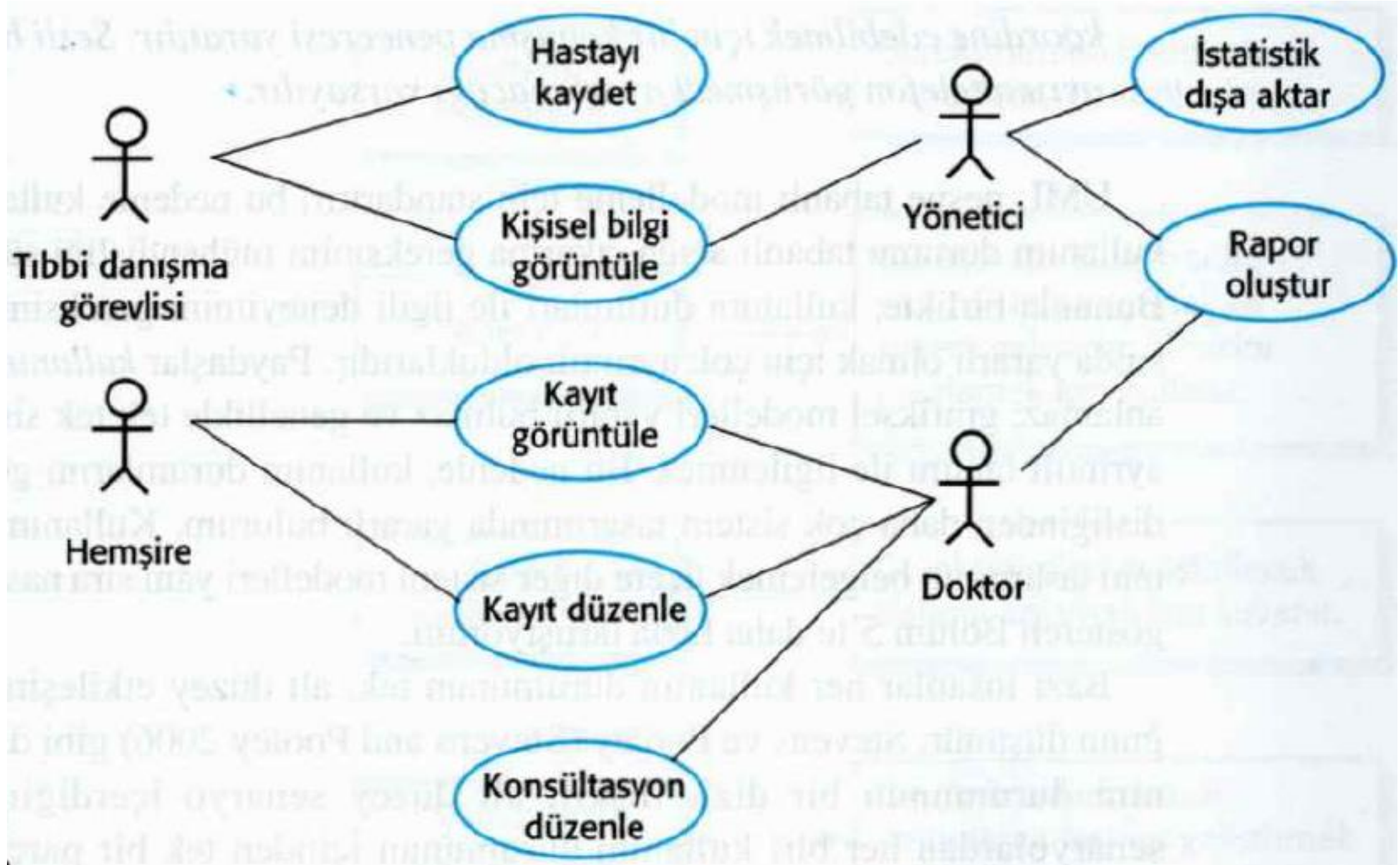
- Doğal dili desteklemek için kullanılır.
- Bir dizi olası alternatif eylem planı tanımlamanız gerektiğinde yararlıdır.

.

Durum	Aksiyon
Şeker seviyesi düşüyor ($r_2 < r_1$)	CompDose = 0
Şeker seviyesi sabit ($r_2 = r_1$)	CompDose = 0
Şeker seviyesi artıyor ve artış oranı azalıyor ($(r_2 - r_1) < (r_1 - r_0)$)	CompDose = 0
I Şeker seviyesi artıyor ve artış hızı sabit veya artıyor ($(r_2 - r_1) \geq (r_1 - r_0)$)	CompDose = <u>yuvarlak</u> $((r_2 - r_1) / 4)$ <u>Yuvarlanmış sonuç</u> = 0 ise o zaman CompDose = <u>MinimumDose</u>

Grafik Gösterim - Kullanım Durumları (Use Cases)

- Kullanıcı ile sistem arasındaki etkileşimi grafiksel model ve yapısal metin ile tanımlar.
- UML'ye dahil edilen bir tür senaryodur.
- Gereksinimlerin tartışılması için fazla ayrıntılıdır.
- Kullanım senaryoları, bir etkileşimdeki aktörleri tanımlar ve etkileşim tipini belirtir. Sonra etkileşimi tanımlayan bilgi eklenir.
- Bir dizi kullanım durumu, sistemle olası tüm etkileşimleri açıklamalıdır.
- Daha ayrıntılı tablo açıklamaları ile desteklenen üst düzey grafik model.
- UML durum diyagramları, sistemdeki olay işleme sırasını göstererek kullanım durumlarına ayrıntı eklemek için kullanılabilir.



Yazılım Gereksinimleri Dökümanı

- Sistem geliştiricilerinin neyi geliştirmesi gerektiğinin kurumsal ifadesidir.
- Sistemin hem kullanıcı hem de sistem gereksinimlerinin ayrıntılı spesifikasyonudur.
- Bir tasarım dökümanı DEĞİLDİR. Mümkün olduğu kadar, sistemin NASIL yapılması gerektiğinden ziyade sistemin NE yapması gerektiğini belirlemelidir.

Gereksinim Dökümanının Kullanıcıları



- **ISO/IEC/IEEE 29148:2018:** Sistem ve Yazılım Mühendisliği gereksinimleri dokümantasyonu için yaygın olarak kullanılan bir standarttır. Bu standart, yazılım ve sistem geliştirme süreçlerinde gereksinim mühendisliği süreçlerinin yürütülmesini ve gereksinimlerin doğru bir şekilde belgelenmesi için gerekli şablonların içeriğini sağlar. IEEE 830-1998 ilk olarak ISO/IEC/IEEE 29148:2011 sürümü ile yürürlükten kalkmış (superseded/obsolete) ve bu uluslararası standartla birleştirilmiştir. Ardından gelen ISO/IEC/IEEE 29148:2018 ise, bu standardın en güncel sürümüdür

- Standart gereksinim dökümanı örnek

ÖRNEK PROJE

- Kampüs Akıllı Yemekhane Sistemi (KAYS) için gereksinim dökümanını hazırlayın.
- Proje; mobil uygulama, web arayüzü, ödeme sistemleri ve donanım (turnike) etkileşimi içerecektir.
- Bu proje için genel hatlarıyla gereksinim dökümanını oluşturun.

- Proje Adı: Kampüs Akıllı Yemekhane Sistemi (KAYS)
- Amaç: Bu dökümanın amacı, Kampüs Akıllı Yemekhane Sistemi (KAYS) için yazılım gereksinimlerini tanımlamaktır. Bu döküman; yazılım geliştiriciler, proje yöneticileri ve üniversite idaresi (paydaşlar) tarafından kullanılacaktır.

Kaynakça

- Ian Sommerville, Yazılım Mühendisliği
- Doç. Dr. Sedat AKLEYLEK'in ders notları
- <https://yteblog.bilgem.tubitak.gov.tr/dokumantasyon#dok%C3%BCman-haz%C4%B1rlama-standartlar%C4%B1>
- Karadeniz Teknik Üniversitesi Software Requirements Specification Template