



**a n a d o l u m**  
e K a m p ü s  
ve  
**a n a d o l u m o b i l**  
dilediğin yerden,  
dilediğin zaman,  
öğrenme fırsatı!



(ekampus.anadolu.edu.tr)



(mobil.anadolu.edu.tr)

**ekampus.anadolu.edu.tr**



Açıköğretim Destek Sistemi  
Açıköğretim Sistemi ile ilgili

merak ettiğiniz her şey AOS Destek Sisteminde...

- ✉ Kolay Soru Sorma ve Soru-Yanıt Takibi
- 🏠 Sıkça Sorulan Sorular ve Yanıtları
- 📞 Canlı Destek (Hafta İçi Her Gün)
- ☎ Telefonla Destek

**aosdestek.anadolu.edu.tr**

AOS DESTEK Sistemi İletişim ve Çözüm Masası

**0850 200 46 10**

[www.anadolu.edu.tr](http://www.anadolu.edu.tr)



T.C. ANADOLU ÜNİVERSİTESİ YAYINI NO: 3247  
AÇIKÖĞRETİM FAKÜLTESİ YAYINI NO: 2111

# SİSTEM ANALİZİ VE TASARIMI

## *Yazarlar*

*Prof.Dr. Osman PARLAKTUNA (Ünite 1, 4)*

*Dr.Öğr.Üyesi Nihat ADAR (Ünite 2, 3)*

*Dr.Öğr.Üyesi Celal Murat KANDEMİR (Ünite 5, 6)*

*Doç.Dr. Metin ÖZKAN (Ünite 7, 8)*

## *Editörler*

*Prof.Dr. Hasan DURUCASU*

*Dr.Öğr.Üyesi Şenay LEZKİ*

Bu kitabın basım, yayım ve satış hakları Anadolu Üniversitesi'ne aittir.  
“Uzaktan Öğretim” teknüğine uygun olarak hazırlanan bu kitabı bütün hakları saklıdır.  
İlgili kuruluştan izin alınmadan kitabı tümü ya da bölmeleri mekanik, elektronik, fotokopi, manyetik kayıt  
veya başka sekillerde çoğaltılamaz, basılamaz ve dağıtılamaz.

Copyright © 2016 by Anadolu University  
All rights reserved

No part of this book may be reproduced or stored in a retrieval system, or transmitted  
in any form or by any means mechanical, electronic, photocopy, magnetic tape or otherwise, without  
permission in writing from the University.

**Öğretim Tasarımcısı**  
*Öğr.Gör. Orkun Şen*

**Grafik Tasarım Yönetmenleri**

*Prof. Tevfik Fikret Uçar  
Doç.Dr. Nilgün Salur  
Öğr.Gör. Cemalettin Yıldız*

**Dil ve Yazım Danışmanı**  
*Prof.Dr. Emine Kolaç*

**Ölçme Değerlendirme Sorumlusu**  
*Ahmet Umay*

**Kapak Düzeni**  
*Prof.Dr. Halit Turgay Ünalan*

**Grafikerler**  
*Özlem Çayırlı  
Burcu Üçok*

**Dizgi ve Yayıma Hazırlama**  
*Kitap Hazırlama Grubu*

Sistem Analizi ve Tasarımı

E-ISBN

978-975-06-2624-1

Bu kitabın tüm hakları Anadolu Üniversitesi'ne aittir.

ESKİŞEHİR, Ağustos 2018

2941-0-0-0-1609-V01

# İçindekiler

Önsöz ..... ix

<b>Sistem Analizi ve Tasarımına Giriş.....</b>	<b>2</b>	<b>1. ÜNİTE</b>
GİRİŞ .....	3	
SİSTEM VE BİLGİ SİSTEMİ .....	3	
BİLGİ SİSTEMLERİNİN SINIFLANDIRILMASI .....	4	
Kayıt İşleme Sistemleri .....	4	
Ofis Otomasyon Sistemleri .....	5	
Yönetim Bilgi Sistemleri .....	5	
Karar Destek Sistemleri .....	5	
Uzman Sistemler .....	5	
Grup Karar Destek Sistemleri .....	5	
BİLGİ SİSTEMİNİN PAYDAŞLARI .....	5	
Sistem Sahipleri .....	6	
Sistem Kullanıcıları .....	6	
Dâhilî Sistem Kullanıcıları .....	6	
Harici Sistem Kullanıcıları .....	6	
Sistem Tasarımcıları .....	7	
Sistem Kurucuları .....	7	
Sistem Analistleri .....	8	
Sistem Analistinin Rolü .....	8	
Sistem Analistinin Sahip Olması Gereken Yetenekler .....	9	
Hariçî Servis Sağlayıcıları .....	10	
Proje Yöneticisi .....	10	
BİLGİ SİSTEMLERİNİ ETKİLEYEN FAKTÖRLER .....	10	
Ticari Eğilimler .....	10	
Küresel Ekonomi .....	10	
Elektronik Ticaret ve İş .....	11	
Güvenlik ve Gizlilik .....	11	
İş Birliği ve Ortaklık .....	11	
Bilgi Varlığı Yönetimi .....	12	
Sürekli İyileştirme ve Toplam Kalite Yönetimi .....	12	
İş Süreçlerinin Yeniden Tasarımı .....	12	
Teknolojik Faktörler .....	13	
Ağlar ve İnternet .....	13	
Mobil ve Kablosuz Teknolojiler .....	14	
Nesne Teknolojileri .....	14	
İş Birlikçi Teknolojiler .....	14	
Kurumsal Uygulamalar .....	14	
SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ .....	15	
Kapsam ve Planlama .....	16	
Gereksinim Analizi .....	16	
Tasarım .....	16	
Gerçekleştirme ve Test .....	16	
Kurulum ve Dağıtım .....	17	

Operasyon ve Bakım .....	17
Özet .....	18
Kendimizi Sınayalım .....	20
Kendimizi Sınayalım Yanıt Anahtarları .....	21
Sıra Sizde Yanıt Anahtarları .....	21
Yararlanılan ve Başvurulabilecek Kaynaklar .....	22

**2. ÜNİTE****Bilgi Sistemlerinin Geliştirilmesi ..... 24**

GİRİŞ .....	25
BİLGİ SİSTEMLERİ GELİŞTİRME .....	25
Kapsam ve Planlama .....	26
Gereksinim Analizi .....	26
Tasarım .....	27
Gerçekleştirme ve Test .....	29
Kurulum ve Dağıtım .....	29
Operasyon ve Bakım .....	30
BİLGİ SİSTEMİ GELİŞTİRME MODELLERİ .....	31
Doğrusal Ardışık Model .....	31
Prototip Model .....	32
Artımlı Model .....	33
Spiral Model .....	35
Bileşen Tabanlı Geliştirme Modeli .....	36
Çevik Model .....	37
Scrum .....	37
Sistem Geliştirme Modellerinin Seçimi .....	38
BİLGİSAYAR DESTEKLİ SİSTEM GELİŞTİRME ARAÇLARI .....	40
Bilgisayar Destekli Yazılım Mühendisliği Araçları .....	40
Süreç Yönetim Araçları .....	41
Proje Yönetim Araçları .....	41
Uygulama Geliştirme Ortamları .....	41
Özet .....	42
Kendimizi Sınayalım .....	44
Kendimizi Sınayalım Yanıt Anahtarları .....	45
Sıra Sizde Yanıt Anahtarları .....	45
Yararlanılan ve Başvurulabilecek Kaynaklar .....	45

**3. ÜNİTE****Proje Yönetimi..... 46**

GİRİŞ .....	47
Proje Yöneticisinin Rolü .....	47
PROJE YÖNETİM ARAÇLARI .....	48
PERT Şeması .....	48
Gantt Şeması .....	50
Proje Yönetim Yazılımları .....	50
PROJE YAŞAM ÇEVİRİMİ .....	52
Kapsam Tanımlama .....	53
Teknik Fizibilite .....	53
Ekonomik Fizibilite .....	54
Operasyonel Fizibilite .....	54

Planlama .....	54
Tahminleme .....	55
Çizelgeleme .....	56
Organize Etme .....	58
Yönetme .....	60
İzleme-Kontrol Etme .....	60
İlerleme Raporları .....	60
Kapatma .....	62
Özet .....	63
Kendimizi Sınayalım .....	64
Kendimizi Sınayalım Yanıt Anahtarları .....	65
Sıra Sizde Yanıt Anahtarları .....	65
Yararlanılan ve Başvurulabilecek Kaynaklar .....	65

## Sistem Gereksinimlerinin Belirlenmesi .....

## 4. ÜNİTE

GİRİŞ .....	67
SİSTEM GEREKSİNİMLERİ .....	67
GEREKSİNİMLERİ BELİRLEME SÜRECİ .....	69
Problemi Belirleme ve Analiz .....	70
Gereksinimlerin Tanımlanması .....	71
Gereksinimlerin Analizi ve Belgelendirilmesi .....	71
Taslak Gereksinimlerin Belgelendirilmesi .....	71
Gereksinimlerin Analizi .....	71
Gereksinimlerin Resmîleştirilmesi .....	71
Gereksinimlerin Yönetimi .....	72
GERÇEKLERİ BELİRLEME TEKNİKLERİ .....	72
Mevcut Belge, Form ve Dosyaların Örneklenmesi .....	72
Araştırma ve Yerinde Ziyaret .....	73
İş Ortamının Gözlemlenmesi .....	73
Anketler .....	74
Avantajları .....	74
Dezavantajları .....	74
Anket Tipleri .....	74
Anketin Hazırlanması .....	75
Mülakatlar .....	76
Avantajlar .....	76
Dezavantajlar .....	76
Prototipleme .....	77
Avantajlar .....	77
Dezavantajlar .....	77
Ortak Gereksinim Planlaması .....	77
Sponsor .....	78
Kolaylaştırıcı .....	78
Kullanıcı ve Yöneticiler .....	78
Yazıcılar .....	78
Bilgi Teknolojileri Çalışanları .....	78
Toplantı Hazırlıkları ve Yürütlmesi .....	79
Gerçekleri Bulma Etiği .....	80

Özet .....	81
Kendimizi Sınayalım .....	82
Kendimizi Sınayalım Yanıt Anahtarları .....	83
Sıra Sizde Yanıt Anahtarları .....	83
Yararlanılan ve Başvurulabilecek Kaynaklar .....	84

**5. ÜNİTE**

<b>Kullanım Durum ve Veri Modelleme .....</b>	<b>86</b>
GİRİŞ .....	87
KULLANIM DURUM MODELLEME .....	87
KULLANIM DURUM MODELLEME TEMEL KAVRAMLARI .....	88
Kullanım Durum Diyagramı .....	89
Kullanım Durumlar .....	89
Aktörler .....	89
İlişkiler .....	90
Kullanım Durum Hikâyeleri .....	92
VERİ MODELLEME VE VERİ YAŞAM DÖNGÜSÜ .....	93
Veri Yaşam Döngüsü .....	94
Veri İhtiyacı .....	95
Gerekli Verilerin Belirlenmesi .....	95
Gerekli Verilerin Toplanması .....	95
Verilerin Saklanması .....	95
Verilerin Kullanılması .....	95
Eski Verilerin Silinmesi .....	95
Kullanılmayan Verilerin Arşivlenmesi .....	95
FARKLI BİLGİ SEVİYELERİNDE VERİ MODELLEME .....	96
Haricî Veri Modeli .....	96
Kavramsal Veri Modeli .....	97
Mantıksal Veri Modeli .....	97
Fiziksel Veri Modeli .....	98
VARLIK İLİŞKİ DİYAGRAMLARI .....	98
Özet .....	101
Kendimizi Sınayalım .....	103
Kendimizi Sınayalım Yanıt Anahtarları .....	104
Sıra Sizde Yanıt Anahtarları .....	104
Yararlanılan ve Başvurulabilecek Kaynaklar .....	105

**6. ÜNİTE**

<b>Süreç Modelleme ve Analizi.....</b>	<b>106</b>
GİRİŞ .....	107
VERİ AKIŞ DİYAGRAMLARI VE BİLEŞENLERİ .....	107
VERİ AKIŞ DİYAGRAMLARININ OLUŞTURULMASI .....	112
Bağlam Diyagram Çizimi .....	113
Düzey 0 VAD Çizimi .....	113
Alt Düzey VAD'ların Çizilmesi .....	114
MANTIKSAL VE FİZİKSEL VERİ AKIŞ DİYAGRAMLARI .....	117
SÜREÇLERİN MANTIKSAL MODELLENMESİ .....	119
Yapısal Dil .....	120
Karar Tabloları .....	121
Karar Ağaçları .....	122

Özet .....	124
Kendimizi Sınayalım .....	126
Kendimizi Sınayalım Yanıt Anahtarı .....	127
Sıra Sizde Yanıt Anahtarı .....	127
Yararlanılan ve Başvurulabilecek Kaynaklar .....	129
<b>UML ile Nesne Yönelimli Analiz ve Modelleme ..... 130</b>	<b>7. ÜNİTE</b>
GİRİŞ .....	131
NESNE MODELLEMEYE İLİŞKİN TEMEL KAVRAMLAR .....	131
Nesneler, Öznitelikler, Davranışlar ve Kapsülleme .....	131
Sınıflar, Genelleşme ve Özelleşme .....	132
Sınıf İlişkileri .....	134
Mesajlar ve Mesaj Gönderme .....	135
Çok Biçimlilik .....	135
UML DİYAGRAMLARI .....	136
Kullanım Durum Diyagramları .....	137
Sınıf Diyagramları .....	138
Dizge Diyagramları .....	139
Durum Geçiş Diyagramları .....	139
Etkinlik Diyagramları .....	140
NESNE MODELLEME SÜRECİ .....	141
Sistem Fonksiyonlarının Modellenmesi .....	141
Kullanım Durum Modeli Oluşturma .....	141
Kullanım Durum Etkinliklerini Modelleme .....	142
Sistem Dizge Diyagramlarının Çizilmesi .....	142
Nesnelerin Saptanması ve Tanımlanması .....	143
Nesnelerin Düzenlenmesi ve İlişkilerinin Tanımlanması .....	143
Özet .....	144
Kendimizi Sınayalım .....	145
Sıra Sizde Yanıt Anahtarı .....	146
Kendimizi Sınayalım Yanıt Anahtarı .....	146
Yararlanılan ve Başvurulabilecek Kaynaklar .....	147
<b>Sistem Tasarımı..... 148</b>	<b>8. ÜNİTE</b>
GİRİŞ .....	149
SİSTEM TASARIM YAKLAŞIMLARI .....	150
Model Güdümlü Yaklaşımlar .....	150
Hızlı Uygulama Geliştirme .....	151
Ortak Uygulama Geliştirme .....	152
SİSTEM TEMİN ALTERNATİFLERİ .....	152
Kurum İçi Geliştirme .....	152
Ticari Paket Yazılım Alımı .....	153
Hizmet Alımı .....	153
SİSTEM MİMARİSİ VE MODELLEME .....	154
Mimari Tasarım Bileşenleri .....	154
Uygulama Mimarisi .....	155
KULLANICI ARAYÜZ TASARIMI .....	155
Gezinim Tasarımı .....	156

Gezinim Kontrol Türleri .....	157
Girdi Tasarımı .....	158
Girdi Türleri .....	159
Girdi Doğrulama .....	160
Çıktı Tasarımı .....	160
VERİTABANI TASARIMI .....	162
Veritabanı Kavramları .....	162
Özet .....	164
Kendimizi Sinayalım .....	165
Kendimizi Sinayalım Yanıt Anahtarı .....	166
Sıra Sizde Yanıt Anahtarı .....	167
Yararlanılan ve Başvurulabilecek Kaynaklar .....	167

## Önsöz

Sevgili öğrenciler,

Anadolu Üniversitesi Açıköğretim Fakültesi'nin Yönetim Bilişim Sistemleri lisans programında yer alan Sistem Analizi ve Tasarımı dersi için hazırlanmış bu kitap 8 üniteinden oluşmaktadır.

1. Ünite, sistem ve bilgi sistemlerinin temellerinden hareketle sistem analizi ve tasarımıyla ilgili konulara genel bir bakış içermektedir.

2. Ünite bilgi sistemlerinin geliştirilmesi sırasında yararlanılan yaklaşımın yanı sıra sistem geliştirme yaşam döngüsünü ele alıp incelemektedir. Sistem geliştirme yaşam dönüğü, kitabın izleyen bölümlerinde ele alınacak sistem analizi ve tasarımları yaklaşımını kullanmayı ve uygulamayı öğreten bir çerçeve oluşturmaktadır.

Günümüzde proje yönetiminin başarıyla uygulandığı alanların biri de bilişimdir. Bu nedenle kitabın 3. Ünitesi proje yönetimine ayrılmıştır. Bu ünitede proje, proje yönetimi ve proje yönetim yazılımları konularına yer verilmiştir.

Bilgi sistemi gereksinimleri doğru olarak tanımlanmadığında, sistem geliştirme süreci ilerledikçe bunların düzeltmesinin maliyeti daha yüksek olabilmektedir. Bu duruma yol açılmaması, daha başlangıçta gerçekleri belirleme tekniklerinin iyi bilinmesi ve uygulanmasına bağlıdır. Bu nedenle sistem gereksinimlerinin belirlenmesi adlı 4. Ünite, gerçekleri belirleme tekniklerine ayrılmıştır.

Kullanım durum ve veri modelleme adlı 5. Ünitede, sistem gereksinimleri doğrultusunda geliştirilen sistem içinde oluşan ve kullanılan verileri düzenleyen ve belgeleyen veri modelleme tekniği ele alınmıştır. Veritabanı olarak tasarlanan ve işletme verisini yapılandırma aracı olarak kullanılan kurumsal ilişki diyagramlarının geliştirilmesi de bu ünitenin kapsamındadır.

6. Ünitede veri akış diyagramları yardımıyla sistem içinde verinin hareketi, geçtiği süreç ya da değişimler ve bunların çıktıları incelenmiştir. Bu yapılrken bilgi sistemi içinde veri akışında ve süreçlerde uygulanan ilke ve yaklaşımın veri akış diyagramlarında gösterilmesi konusuna da değinilmiştir.

7. Ünite, nesne modellemeye ilişkin temel kavramların aktarılması sonrasında hedeflenen bilgi sisteminin UML diyagram yapıları ile modellenmesi konusuna ayrılmıştır. Ayrıca nesne modellemenin faydalari da bu ünite kapsamında ele alınmıştır.

Kitabın son ünitesinde ise, analizi tamamlanan sistemin tasarım yaklaşımı, sistem temin alternatifleri, sistem mimarileri gibi konular üzerinde durulmuştur. Ayrıca kullanıcı arayüz tasarımını ile veritabanı tasarımını kavramları ve faaliyetleri de incelenmiştir.

Yukarıda tanıtılan 8 ünite yardımıyla, sistem analizi ve tasarım konusundaki araç, teknik ve yöntemlerin yanı sıra son kullanıcı, yönetici ve bilişim sistemi uzmanları arasındaki ilişkileri öğrenme fırsatı da sunulmaktadır. Böylelikle, yeni durumlar ve yöntemlere adapte olabilecek bir birimin okuyucuya kazandırılması amaçlanmaktadır.

Bu kitabın programa alınmasından elinize geçmesine kadar olan süreçte emeği geçen, başta yazarlar olmak üzere herkese isim belirtmeksiz teşekkür ederiz.

Kitabın, dersten sorumlu olanlar kadar sistem analizi ve tasarımına ilgi duyan diğer öğrencilere ve öğrenci olmayan okurlarına da yararlı olmasını dileriz.

Editörler

Prof.Dr. Hasan DURUCASU

Dr.Öğr.Üyesi Şenay LEZKİ

1

## Amaçlarımız

**Bu üniteyi tamamladıktan sonra;**

- Sistem ve bilgi sistemini tanımlayabilecek,
  - Farklı bilgi sistemlerini ayırt edebilecek,
  - Bilgi sistemi paydaşlarının işlevlerini betimleyebilecek,
  - Bilgi sistemlerini etkileyen faktörleri açıklayabilecek,
  - Sistem geliştirme yaşam döngüsünün fazlarını sıralayabilecek bilgi ve becerilere sahip olabileceksiniz.

## Anahtar Kavramlar

- Sistem
  - Bilgi Sistemi
  - Sistem Analisti
  - Sistem Geliştirme Yaşam Döngüsü

## **İçindekiler**



# Sistem Analizi ve Tasarımına Giriş

## GİRİŞ

İşletmeler uzun süredir insan ve hammadde gibi kaynakların yönetiminin önemini bilmektedir. Günümüzde bilgi de önemli bir kaynak olarak karşımıza çıkmaktadır. Bilgi, iş yapmanın bir yan ürünü olmak yerine işletmenin başarılı olup olmayacağı konusunda kritik öneme sahiptir.

Bir işletme, diğer kaynakları yönettiği gibi faydasını artırmak için bilgiyi de doğru biçimde yönetmelidir. Bilginin doğru kullanımı, işletmeyi rekabetçi bir pozisyon'a getirmesi için önemlidir. Ama bunun da bir bedeli vardır. Bu nedenle yöneticiler; bilginin üretimi, dağıtım, güvenliği ve depolanması için gerekli kaynakları ayırmalıdır.

Ağ bağlantısı olan bilgisayarların varlığı ve İnternet erişiminin yaygınlaşması ile genel olarak toplumda ve özellikle işletmelerde bilgi patlamasına yol açmıştır. Elle üretilen verilere kıyasla bilgisayarlar tarafından büyük miktarlarda veri üretilebilmekte ve daha güvenli olduğu kabul görmektedir. Ancak bu verilerin organizasyonu ve sürdürülebilir olarak kullanımının maliyeti çok yüksek olabilir. Bu nedenle günümüzde bilgi sistemlerinin önemi artmıştır.

Bu bölümde; bilgi sistemlerinin temel kavramları ve farklı bilgi sistemleri ele alınacak, bilgi sisteminin paydaşları ve görevleri tanımlanacak, sistem analistinin görevleri ayrıntılı biçimde aktarılacak ve bilgi sistemlerini etkileyen ticari ve teknolojik faktörler incelenecaktır. Ayrıca bilgi sistemi geliştirirken uygulanan fazların kavramsal bir modeli olan Sistem Geliştirme Yaşam Döngüsü de ele alınacaktır.

## SİSTEM VE BİLGİ SİSTEMİ

Sistem, istenilen bir sonuca ulaşmak için birbiriley ilişkili ve birlikte çalışan bir grup bileşendir. DVD oynatıcı, alıcı, hoparlör ve monitörden oluşan bir ev sinema sistemi örnek olarak verilebilir.

Bilgi sistemi, işletmenin ihtiyaçlarını desteklemek için veri toplayan, işleyen, depolayan; insan, veri, süreçler ve bilgi teknolojilerinin etkileşimde bulunduğu bir yapıdır.

Tipik bir bilgi sistemi aşağıdaki bileşenlerden oluşur:

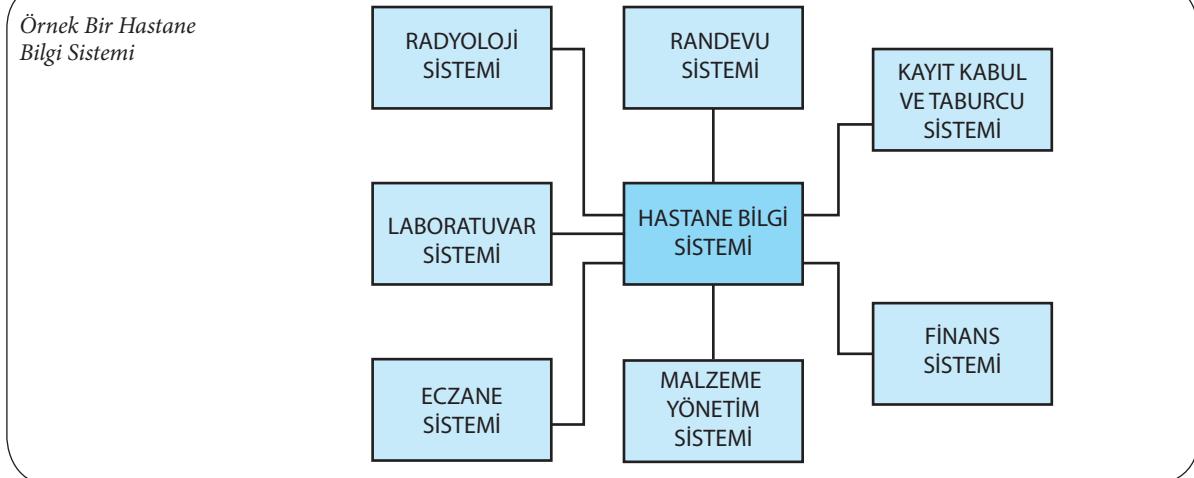
- *Donanım kaynakları*: Sunucu, bilgisayar, monitör, klavye, yazıcı vb. sayısal ürünler.
- *Yazılım kaynakları*: Verileri düzenleyen, işleyen ve analiz eden programlar. Verilerin nasıl işleneceği ve analiz edileceğini belirten süreçler ve yordamlar.
- *İnsan kaynakları*: Bilgi sistemi sahibi, bilgi sistemini tasarlayan, kuran uzmanlar ve bilgi sistemini kullanan çalışanlar.

- Veri kaynakları:* Bilgi sistemlerinin kullandığı ve ürettiği verilerin tutulduğu veritabanı (database) ve bilgi tabanı (knowledge base).
- Ağ kaynakları:* İşletme içinde ve işletme dışındaki farklı birimlerin aynı bilgi sistemi kullanması durumunda birbirile bağlanması sağlayen yapılar.

Bilgi sistemlerine örnek olarak işletme bilgi sistemleri, öğrenci bilgi sistemleri, coğrafi bilgi sistemleri ve hastane bilgi sistemleri verilebilir. Bu kitapta verilecek bilgiler bir hastane bilgi sistemi örneğinden hareketle aktarılacaktır.

Hastane bilgi sistemi, hastanenin idari ve tıbbi bilgilerinin yönetimini kolaylaştırmak ve sağlık hizmetlerinin kalitesini yükseltmek için düzenlenmiş bilgi sistemi olarak tanımlanmaktadır. Hastane bilgi sistemi, randevu sistemi, hasta kayıt kabul ve taburcu sistemi, malzeme yönetim sistemi, finans sistemi, laboratuvar sistemi, radyoloji sistemi, eczane sistemi gibi birbirile ilişkili alt sistemlerden oluşabilir. Şekil 1.1'de örnek bir hastane bilgi sisteminin blok şeması verilmiştir.

**Şekil 1.1**



SIRA SİZDE

1

Sizce Şekil 1.1'de örnek olarak verilen hastane bilgi sisteminin bileşenleri nelerdir?

Bilgi sistemleri, işletmeyi ve çalışanlarını, müşterilerini, tedarikçilerini ve birlikte çalıştığı işletmeleri destekleyen yararlı bilgiyi üretir ve yönetir. Bir işletmenin rekabet edebilmesi veya rekabetçi avantaj kazanabilmesi için bilgi sistemlerinin geliştirilmesi ve kullanımında tüm çalışanların katılımlarının sağlanması kritik önemdedir.

## BİLGİ SİSTEMLERİNİN SINIFLANDIRILMASI

Bilgi sistemleri, kullanıcıların ve işletmelerin ihtiyaçları doğrultusunda farklı biçimlerde ve boyutlarda olabilir. Bu sistemler yerine getirdiği işlevlere göre aşağıdaki şekilde sınıflandırılabilir.

### Kayıt İşleme Sistemleri

Günümüzde Kayıt İşleme Sistemleri (KİS), işletmenin muhasebe ve envanter gibi operasyonel düzeydeki işlemlerini kaydeden bilgisayar tabanlı sistemlerdir. Bu sistemler işletmenin dış dünya ile etkileşimde olduğu bir arayözdür. Bilgisayar tabanlı KİS, eskiden elle yapılan sipariş, ödeme ve rezervasyon kayıtları gibi işlemler için gereken süreyi kısaltır. Yöneticiler işletmenin güncel durumunu bu sistemin ürettiği verileri kullanarak belirlediği için sistemin düzgün ve kesintisiz çalışması önemlidir.

## Ofis Otomasyon Sistemleri

Ofis Otomasyon Sistemleri (OOS) yeni bilgi üretmezler. Bu sistemler, bilgiyi analiz eden, gerekliyorsa bilgi üzerinde değişiklik yapan ve sonrasında gerekli birimlerle paylaşan çalışanları desteklemek üzere geliştirilirler. Bu sistemlere örnek olarak verilebilecek kelime işlemciler, tablolama ve masaüstü yayıcılık programları, e-posta ve sesli haberleşme sistemleri çalışanların günlük faaliyetlerini desteklemek için gereken belgeleri oluşturur ve paylaşmalarını sağlar.

## Yönetim Bilgi Sistemleri

Yönetim Bilgi Sistemleri (YBS), işlem verilerini kullanarak yöneticilerin ihtiyaç duydukları bilgileri üretir ve rapor hâlinde sunar. Bu sistemler, veriye erişmek için ortak bir veritabanı kullanır ve kullanıcıların veriye erişimine, yorumlamasına, uygulamasına yardımcı olmak için hem veriyi hem de modelleri veritabanına depolar.

## Karar Destek Sistemleri

Karar Destek Sistemleri (KDS), karar vericilerin alternatifleri belirlemesine ve bu alternatifler arasından birini tercih etmesine yardımcı olur. KDS ile YBS ortak veritabanını kaynak olarak kullanmaları nedeniyle birbirlerine benzemektedir. Ancak KDS, karar verme sürecinin her aşamasında kullanıcıya yardımcı olmasıyla YBS'den farklılaşmaktadır.

## Uzman Sistemler

Uzman sistemler (US), uzman bir problem çözümünün veya karar vericinin bilgisine sahip olacak şekilde geliştirilir. Uzmanın düşüncelerini taklit ederek işletmede ortaya çıkan bir probleme çözüm üretebilir. US, KDS'den farklı olarak son kararı karar vericiye bırakmaz. Bir probleme veya problem sınıfına en iyi çözümü seçer. US'nin temel bileşenleri bilgi tabanı, **Sonuç Çıkarma Motoru** ve kullanıcı arayüzüdür.

**Sonuç Çıkarma Motoru**, yapısal sorgulama dillerinden gelen sorgulamaları işleyerek kullanıcı ve sistemi birleştirir.

## Grup Karar Destek Sistemleri

Grup karar destek sistemleri (GKDS), bir grubun verilen problemi çözmeye yardımcı olmak üzere geliştirilir. Problem çözümünde oylama, anket, beyin fırtınası ve senaryo geliştirme gibi yöntemler kullanılır. Grup karar uygulaması gerekli donanımın, yazılımın ve grup üyelerinin birlikte olduğu bir ortamda kolaylaştırıcı görevi gören bir moderator tarafından yönetilir. GKDS yazılımları, tüm grup üyelerinin aykırı da olsa fikirlerini belirtmekten kaçınmayacakları ve atılgan grup üyelerinin diğer grup üyeleri üzerindeki basklarını en aza indirecek biçimde tasarlanmalıdır.

**Sizce Şekil 1.1'deki örnek hastane bilgi sistemi için tasarlanacak KDS ve US'nın işlevleri ve farkları ne olmalıdır?**



SIRA SİZDE

## BİLGİ SİSTEMİNİN PAYDAŞLARI

Bilgi sisteminin kurulması konusunda çalışacak **sistem analisti** kuracağı sistemin paydaşlarını bilmek zorundadır. Bilgi sistem paydaşları beş grupta sınıflandırılabilir. Bunlar; Sistem Sahipleri, Sistem Kullanıcıları, Sistem Tasarımcıları, Sistem Kurucuları ve Sistem Analystleridir. Sistem paydaşlarının görevlerini anlatmadan önce bir kişinin birden fazla paydaş kimliğine sahip olabileceğini belirtmeliyiz. Örneğin bir sistem analisti aynı zamanda sistem tasarımcısı olabilir.

**Sistem analisti** bir işletmenin problemlerini ve ihtiyaçlarını belirler. Bu konularda iyileştirmeler yapmak için çalışır. Veri, süreçler ve bilgi teknolojisinde yapılacak değişiklikleri planlar ve gerçekleştirilmesini sağlar.

## Sistem Sahipleri

Küçük veya büyük her bilgi sisteminin bir veya daha fazla sahibi olur. Sistem sahipleri genelde yönetim kademesinden birisidir. Sistem sahipleri teknik ayrıntılar ile ilgilenmeksızın, sistemin maliyeti ve kurulacak sistemin işletmeye kazandıracakları ile ilgilenir.

## Sistem Kullanıcıları

Sistem kullanıcıları bir bilgi sistemindeki çalışanların çoğunluğunu oluşturur. Sistem kullanıcıları sistemin maliyetleri ve faydaları ile daha az ilgilenirler. Genelde sistemin gerçekleştirdiği fonksiyonlar veya sistemin öğrenilmesinin ve kullanımının kolaylığı ile ilgilenirler. Bu nedenle kullanıcılarla yapılacak konuşmalar teknik düzeyden çok işlerin gerektirdiği özellikler üzerine olmalıdır. Sistem kullanıcıları dâhilî ve haricî kullanıcılar olarak iki gruba ayrılabilir.

### Dâhilî Sistem Kullanıcıları

Dâhilî kullanıcılar bilgi sistemlerinin kurulduğu işletmelerin çalışanlarıdır ve genelde bilgi sistemi kullanıcılarının büyük kısmını oluştururlar.

*Büro ve servis çalışanları:* İşletmelerdeki günlük birçok işlemi yapan çalışanlardır. Büro çalışanları sipariş, fatura, ödemeler ve benzeri işlemleri gerçekleştirir, yazışmaları yapar ve dosyalar. Servis çalışanları ise mağazalarda siparişleri veya hizmet bilgilerini bilgi sisteme girer. Bir işletmedeki temel verilerin çoğunluğu bu çalışanlar tarafından üretilir. Bu kişilere yönelik bilgi sistemlerinde işlemlerin gerçekleştirilme hızı ve doğruluğu ile ilgilenilmelidir.

*Teknik ve profesyonel personel:* Yüksek yetenek gerektiren ve özelliği olan işleri yerine getiren çalışanlar bu grupta yer alır. Bu çalışanlara hukukçular, muhasebeciler, mühendisler, bilim insanları, pazar analistleri, reklamcılar ve istatistikçiler örnek verilebilir. Teknik ve profesyonel personeli hedefleyen bilgi sistemleri, veri analizi ve problem çözümü için zamanında bilgi üretimi üzerine odaklanmalıdır.

*Yöneticiler:* Üst düzey yöneticiler uzun dönemli stratejik planlama ve karar alma, orta düzey yöneticiler kısa dönemli taktiksel planlama ve karar alma, alt düzey yöneticiler ise günlük planlama ve karar alma ile ilgilenir. Yöneticilerin problemleri belirlemek, çözmek ve doğru kararlar vermek için doğru zamanda doğru bilgiye ihtiyaçları vardır. Bu nedenle yöneticilerle ilgili bilgi sistemleri bilgiye ulaşım üzerine odaklanmalıdır.

### Haricî Sistem Kullanıcıları

İnternet, geleneksel bilgi sistemlerinin sınırlarını başka işletmeleri ve müşterileri de içine alacak şekilde genişletmiştir. Haricî kullanıcılara ait bazı örnekler aşağıda verilmiştir.

*Müşteriler:* İşletmenin ürünlerini ve hizmetlerini satın alan diğer tüm işletmeler ve kişiler müşteri grubuna girmektedir. Günümüzde müşteriler, eskiden dâhilî bir kullanıcının müdahalesini gerektiren bir alışveriş işlemini işletmenin bilgi sistemini kullanarak gerçekleştirebilmekte ve böylece bilgi sisteminin doğrudan kullanıcısı olmaktadır. Örneğin, bir işletmenin ürününü İnternet üzerinden satın alan kişi o işletmenin satış bilgi sisteminin kullanıcısı olmaktadır.

*Tedarikçiler:* İşletmenin bir ürünü veya hamaddeyi satın aldığı bir diğer işletme, bilgi sistemiyle doğrudan etkileşime geçerek işletmenin ihtiyaçlarını belirleyebilir ve bu ihtiyaçları karşılayacak siparişleri hazırlayabilir. Böylece, tedarikçilerin siparişleri başlatmaları için her zaman dâhilî bir kullanıcıya ihtiyacı kalmamaktadır.

*Taşeronlar:* İşletmenin ürün ya da hizmet satın aldığı veya ortak çalıştığı diğer işletmelerdir. Birçok işletme aslı görevlerinin dışındaki temizlik vb. işlerini bir başka işletmeye yaptırmaktadır.

**Çalışanlar:** Ev, ofis veya işletmenin bulunduğu şehirden başka bir şehirde çalışanlar, harici sistem kullanıcısı olarak kabul edilir. Örneğin satış temsilcileri genelde şehir dışında çalışmak zorundadır. Bu çalışanların da dahilî kullanıcılarla aynı bilgi sistemine ulaşma ihtiyacı vardır.

Harici kullanıcılar *uzak kullanıcı* veya *mobil kullanıcı* olarak da adlandırılmaktadır. Bu kullanıcılar bilgi sistemlerine dizüstü bilgisayar, el bilgisayarları veya akıllı telefonlarla bağlanırlar. Bu cihazlar için bilgi sistemlerinin tasarımları en güncel problemlerden biridir.

## Sistem Tasarımcıları

Sistem tasarımcıları, bilgi sistemlerine teknolojik katkı sağlayan kişilerdir. Sistemin farklı işlevlerini tasarlayan bu uzman kişiler ve görevleri aşağıda sıralanmıştır.

**Veritabanı yöneticileri:** İşletmenin veritabanını tasarlayan ve değişikliklerini düzenleyen veritabanı uzmanlarıdır.

**Ağ mimarları:** Yerel ağ, İnternet ve diğer ağlara bağlantıyı tasarlayan, kurulan, yapılandırılan ve destekleyen ağ ve haberleşme uzmanlarıdır.

**Web mimarları:** İşletmeler için web sayfalarını düzenleyen uzmanlardır.

**Grafik sanatçıları:** Grafik teknolojileri ve metodlarında uzmanlaşmış kişilerdir. Bilgisayarlar, WEB ve akıllı telefonlar için kolay kullanılabilen arayüzleri tasarlarlar.

**Güvenlik uzmanları:** Veri ve ağ güvenliğini sağlamakta kullanılan teknoloji ve metodlarda uzmanlaşmış kişilerdir. Bilgi sisteminin güvenliği üzerine çalışmalar yaparlar.

**Teknolojik uzmanlar:** Bilgi sisteminde yer alacak özel teknolojiler konusunda uzmanlıkla sahip kişilerdir (örneğin veritabanı yazılımında Oracle uzmanı veya ağ donanımında Cisco ağ uzmanı gibi). Bilgi sisteminde kullanılacak teknoloji konusunda bu kişilerden danışmanlık veya kurulum hizmeti alınır.

## Sistem Kurucuları

Sistem tasarımcılarının belirttikleri özellikler doğrultusunda sistemi kurulan uzman grubudur. Küçük işletmelerde veya küçük bilgi sistemlerinde sistem tasarımcıları ve sistem kurucuları aynı kişilerdir. Ancak büyük bilgi sistemlerinde aşağıda belirtilen farklı uzmanlar bu görevleri yerine getirir.

**Uygulama programcıları:** İşletmenin iş süreçlerinin ve problemlerinin gerekliliklerini bilgisayar diline çeviren uzman kişilerdir. Uygulama programcıları veriyi elde edip depolamak ve depolanmış veriyi okumak için bilgisayar programlarını geliştirir ve test ederler.

**Sistem programcıları:** İşletim sistemi düzeyinde yazılımlar ve hizmetler geliştiren, test eden ve uygulayan uzman kişilerdir. Ayrıca yeniden kullanılabilir yazılım bileşenleri geliştirerek uygulama programlarının kullanımına sunarlar.

**Veritabanı programcıları:** Veritabanı programları ve teknolojileri üzerine uzmanlaşmış kişilerdir. Veritabanı yapılarını ve bu yapıları kullanan programları kurmak, uyarlamak ve test etmek bu uzmanların görevidir.

**Ağ yöneticileri:** Bilgisayar ağlarını tasarlayan, kurulan, sorunlarını gideren ve optimize eden uzman kişilerdir.

**Güvenlik yöneticileri:** Bilgisayar ağlarındaki güvenlik ve kişisel gizlilik kontrollerini tasarlayan, kurulan ve sorunlarını gideren uzman kişilerdir.

**Webmaster:** Web hizmetlerini kodlayan ve yürütlenen kişilerdir.

**Yazılım bireştiricileri:** Yazılım paketlerini donanım, ağ ve diğer yazılım paketleriyle birleştiren kişilerdir.

## Sistem Analistleri

Sistem sahipleri, kullanıcıları, tasarımcıları ve kurucularının gerçekleştirilecek ve kullanılacak bilgi sistemlerine bakış açıları farklıdır. Sistem sahipleri ve kullanıcıları genel özelliklerle, tasarımcılar ve kurucular ise ayrıntılarla ilgilendir. Başka bir bakış açısıyla da sistem tasarımcıları ve kurucuları teknik, sistem sahipleri ve kullanıcıları ise genelde teknik olmayan kişilerden oluşur. Bu durum, bilgisayar tabanlı işletme çözümlerine ihtiyacı olanlarla bilgi teknolojilerini bilen kişiler arasında bir iletişim boşluğu oluşturur. Sistem analisti bu boşluğu doldurmalıdır. Sistem analisti, sistem sahipleri ve sistem kullanıcıları için işletmenin problemlerini ve ihtiyaçlarını belirler ve onaylar. Sistem tasarımcıları ve sistem kurucuları için ise önerilen teknik çözümün ihtiyaçları karşıladığı ve işletmeye uyumunu denetler. Böylece, sistem analisti diğer paydaşlarla etkileşim içinde bulunarak, bilgi sisteminin gelişimini kolaylaştırıcı görevini yerine getirir.

## Sistem Analistinin Rolü

Sistem analisti, işletmenin örgütel süreçlerinin iyileştirilmesi amacıyla girilen verileri inceler. İncelediği verilerden kullanıcıların teknolojiyle nasıl etkileştiğini ve işletmenin nasıl çalıştığını, problemlerini ve fırsatlarını belirler. İşletme ve bilgi gereksinimlerini, farklı teknik uzmanlar tarafından gerçekleştirilecek bilgi sistemleri özelliklerine dönüştürür.

Sistem analistinin en önemli görevi işletmede **değişim** başlatmaktadır. İyi bir analist, işletmeye rekabetçi avantaj getirecek, yeni pazarlar ve hizmetler bulacak şekilde değişim sürecini başlatabilmelidir. Sistem analistinin üç temel rolü; danışman, destekleyici uzman ve değişim ajansı olmaktadır.

**Danışman Olarak Sistem Analisti:** Sistem analistleri sıklıkla kişilere veya işletmelere sistem danışmanlığı yaparlar. İşletmeler tarafından özellikle bilgi sistemleri sorunlarını belirlemek için işe alınırlar. Böylece harici bir danışman olarak, işletmenin ve çalışanlarının sahip olmadığı yeni ve farklı bir bakış açısını işletmeye kazandırabilirler. Ancak işletmenin yönetimsel kültürünü bilmemiş olduğu için harici analistin çalıştırılması dezavantaj olarak da kabul edilebilir.

**Destekleyici Uzman Olarak Sistem Analisti:** Sistem analisti işletmedeki başka bir sistem görevi için destekleyici uzman olarak görevlendirilebilir. Örneğin sistem analisti bilgisayar donanımı, yazılımı ve bunların işletmede kullanımı üzerine çalışabilir. Burada yapılacak iş genellikle asıl bilgi sistem projesinde çalışmak yerine işletmedeki küçük bir bölümü etkileyen ufak bir uyarlanmanın yapılmasıdır. Destekleyici uzman olarak bilgi sistemi projesini yönetmek yerine, projeyi yöneten kişilere kaynak olarak hizmet vermektedir. Bir üretim veya hizmet işletmesinde görevli sistem analistinin günlük işlevlerinin çoğu bu rol üzerine olur.

**Değişim Ajansı Olarak Sistem Analisti:** Sistem analistinin bir işletmede dâhilî veya harici olarak alacağı en kapsamlı ve sorumluluk isteyen rolü değişim ajansı görevidir. Değişim ajansı, değişim için katalizör olan, değişim planını geliştiren ve gerçeklenmesinde başka görevlilerle birlikte çalışan kişi olarak tanımlanabilir. Değişim ajansının işletmedeki varlığı işletmenin değişimine sebep olur. Sistem analisti bu gerçeğin farkına varmalı ve analizlerinin başlangıç noktası olarak kullanmalıdır. Değişimin gerekli olup olmadığıni belirlemek üzere kullanıcılar ve yöneticilerle iletişime geçmelidir. Analizlerden sonra değişimin gerekli olduğu ortaya çıkarsa bu değişimini gerçekleştirecek kişilerle değişim planını geliştirmelidir. Planı oluşturulan değişimin yapılması kararı verildikten sonra da değişimini gerçekleştirecek kişilerle sürekli temas hâlinde olmalıdır.

Bilgi sistemindeki değişim işletmenin diğer birimlerinde de değişikliklere sebep olur. Bu nedenle değişim ajansı rolündeki sistem analisti, bilgi sistemlerinin kullanımındaki değişimde izlenecek yollardan bir tanesinin savunucusu olmalı ve kullanıcılarla değişim sürecini öğretmelidir.

**Değişim**, bilgi sistemlerinin kullanımı sonucu işletmede gerçekleştirilen iyileştirmeler.

## Sistem Analistinin Sahip Olması Gereken Yetenekler

Yukarıda açıklanan rollerden anlaşılacığı gibi, sistem analistinin birçok yeteneğe sahip olması gereklidir. Her şeyden önce sistem analisti bir problem çözücüdür. Ancak iyi bir iletişim ve yönlendirme yeteneğine de sahip olmalıdır. İzleyen kesimde sistem analistinin sahip olması gereken yetenekler anlatılmaktadır.

*Bilgi teknolojileri bilgisi:* Sistem analisti mevcut ve gelişen bilgi teknolojileri konusunda bilgi sahibi olmalıdır. Bu bilgi, üniversiteden alınan dersler, profesyonel gelişim ders ve seminerleri veya işletme içi eğitim programları ile elde edilebilir. Analist aynı zamanda konu hakkında yayını okuyarak ve profesyonel topluluklara üye olarak bilgilerini güncelleyebilir.

*Programlama deneyimi ve uzmanlığı:* Programlama deneyimi olmayan bir sistem analistinin, programcılar için uygun bilgi sisteminin işletme ve teknik özelliklerini hazırlaması zor olacaktır. Bu nedenle sistem analistinin bir veya daha çok programlama dili konusunda uzman olması gereklidir.

*İş süreçleri ve terminolojisi konusunda genel bilgi:* Sistem analisti, iş dünyasının problemlerini ve ihtiyaçlarını anlayabilmek için iş uzmanlarıyla iletişim kurmak zorundadır. Bazı bilgilerse sadece deneyimlerle kazanılabilir. Ancak iddialı sistem analistleri temel işletme derslerini üniversitelerden alarak bu konuda bilgi sahibi olurlar. Uygun dersler, finansal muhasebe, işletme ve maliyet muhasebesi, finans, pazarlama, üretim veya süreç yönetimi, kalite yönetimi, ekonomi ve işletme hukuku olabilir.

*Genel problem çözme yeteneği:* Sistem analisti büyük bir işletme problemini ele alır, problemi parçalara ayırır, problemin sebeplerini ve etkilerini belirler ve bir çözüm önerir. Analist, problemi analiz etmeden önce bir çözüm sunma eğiliminden kaçınmalıdır. Sistem analisti, problem çözme yeteneğini kazanmak için, problem çözme tekniklerinin, kritik düşünme ve muhakeme konularının felsefesini anlatan dersler alabilir.

*Kişiler arası iyi iletişim yeteneği:* Bir analist hem sözlü hem de yazılı olarak etkin iletişim kurabilmelidir. Bir analistin kariyerinde başarılı olmasının en önemli faktörü teknik yetenekler yerine iletişim yetenekleridir. Üniversiteler iş ve teknik konularda yazma ve konuşma, mülakat ve dinleme konularında dersler sunmaktadır. Bu derslerin hepsi sistem analisti için yararlıdır.

*Kişiler arası ilişki kurma yeteneği:* Sistem analisti bilgi sistemi projesi geliştirme sürecinde tüm paydaşlar ile etkileşim içindedir. Bu etkileşimler, analistin grup dinamигine, işletme politikasına, çalışma ve değişimle başa çıkabilecek etkin kişiler arası ilişki yeteneğine sahip olmasını gerektirir. Üniversiteler kişiler arası yetenekleri geliştiren; takım çalışması, ikna etme prensipleri, değişim ve çatışmayı yönetme ve liderlik konularında dersler açmaktadır.

*Esneklik ve uyumluluk:* Hiç bir proje bir diğeri ile aynı değildir. Bu nedenle tüm projelere uygulanabilecek tek ve sihirli bir yaklaşım veya standart çözüm bulunmamaktadır. Başarılı olmak için sistem analisti esnek olmayı, özel zorluklara ve durumlara nasıl uyum sağlayacağını öğrenmelidir.

*Karakter ve etik:* Sistem analistinin yaptığı işin doğası kuvvetli bir karakter ve doğru ile yanlışı ayırmayı yeteneği gerektirir. Analistler genellikle kamuya açıklanması sakıncalı olan hassas ve gizli gerçeklere ve bilgilere tanık olurlar. Sistem analizi ve tasarıımı sonucu ortaya çıkan ürün, işverenin fikri mülkiyeti kabul edilir. Analist tanık olduğu ve tasarlanan ürün hakkındaki bilgileri üçüncü kişi ve kuruluşlarla paylaşmamalı, bu konuda etik kurallarına uymalıdır. Analistin uyacağı bilgisayar etiği konusunda birçok standart bulunmaktadır. Örnek olarak Bilgisayar Etik Enstitüsünün hazırladığı "Bilgisayar Etiğinin On Kanunu" Tablo 1.1'de verilmiştir.

**Tablo 1.1**  
Bilgisayar Etiğinin On Kanunu

**Kaynak:** WHITTEN, J. L. ve BENTLEY L.D. (2007), s. 16.

1	Bilgisayar bir başka insana zarar vermek için kullanılmamalıdır.
2	Bir başka kişinin bilgisayarla yaptığı işe müdahale edilmemelidir.
3	Bir başka kişinin bilgisayar dosyalarına gizlice erişilmemelidir.
4	Bilgisayar hırsızlık için kullanılmamalıdır.
5	Bilgisayar yalancı şahitlik için kullanılmamalıdır.
6	Ücreti ödenmeyen tescilli yazılım kopyalanmamalı ve kullanılmamalıdır.
7	Başkasının bilgisayar kaynakları izinsiz veya uygun ödemeyi yapmadan kullanılmamalıdır.
8	Bir başkasının fikri ürünü kendine mal edilmemelidir.
9	Yazılacak veya tasarlanacak programların sosyal sonuçları düşünülmelidir.
10	Bilgisayar her zaman insanlara saygılı olmayı garanti edecek bir şekilde kullanılmalıdır.

## Harici Servis Sağlayıcıları

Herhangi bir paydaş pozisyonu, dâhilî olduğu kadar harici bir çalışan ile de doldurulabilir. Danışmanlar, harici servis sağlayıcılarına bir örnektir. Harici servis sağlayıcı belli bir projeye özel uzmanlık ve deneyim kazandıran sistem analisti, tasarımcısı veya kurucusu olabilir. Satış mühendisleri, sistem danışmanları ve sözleşmeli programcılar harici servis sağlayıcılara örnek olarak verilebilir.

## Proje Yöneticisi

Yukarıda özellikleri açıklanan sistem paydaşlarının, işletmenin faydalananacağı bilgi sistemi ve uygulamalarının geliştirilmesi sürecinde bir takım olarak birlikte çalışması gereklidir. Her takımın bir lideri ihtiyacı vardır. Bu nedenle, bilgi sistemlerinin zamanında, belirlenen bütçe sınırlarında ve kabul edilebilir bir kalitede geliştirilmesi için bir veya birkaç paydaşın proje yöneticisi görevini üstlenmesi gerekmektedir. İşletmelerde genelde tecrübeli bir sistem analisti proje yöneticisi olmaktadır.

SIRA SİZDE

3

Sizce Şekil 1.1'de verilen hastane bilgi sisteminin paydaşları kimlerdir?

## BİLGİ SİSTEMLERİNİ ETKİLEYEN FAKTÖRLER

Bilgi sistemlerinin gelişimini etkileyen faktörler güncel ticari eğilimler ve teknolojik gelişmeler olarak iki grupta sınıflandırılabilir. Takip eden bölümde bu faktörler anlatılacaktır.

## Ticari Eğilimler

Günümüzde iş ve ticaret yapma eğilimleri değişmektedir. Ortaya çıkan bu yeni eğilimler bilgi sistemlerini de etkilemektedir. Birçok yeni eğitim kısa sürede yok olurken, bazı ticari eğilimler sistemlerin gelişmesinde etkili olmaktadır. Bu eğilimlerin bir kısmı birbirile iliskili ve bütünlüğiktir. Aşağıdaki bölümlerde bu eğilimlerden bir kısmına yer verilmiştir.

## Küresel Ekonomi

1990'lı yillardan itibaren ekonominin küreselleşme eğilimi giderek artmaktadır. Gelişmekte olan ülkelerin sunduğu düşük maliyetli, yüksek kaliteli alternatif ürünler sonucu rekabet de küreselleşmektedir. Bu nedenle bilgi sistemleri ve bilgisayar uygulamaları uluslararası olmak zorundadır. Uluslararası kullanılan bir bilgi sistemi, farklı yabancı dilleri, döviz kuru değişimlerini, uluslararası ticaret kurallarını ve değişik iş kültürü ve pratiklerini desteklemelidir. Ayrıca birçok bilgi sistemi performans analizi yapmak ve karar vermek için bilginin sağlam ve güvenilir olmasını gerektirmektedir. Ancak yukarıda bahsedilen dil, döviz kuru, sınır ötesi bilgi düzenlemeleri bu tür bir güvenilirliğin elde edilmesini

zorlaştırmaktadır. İlave olarak, yönetici ve kullanıcılarla değişik dillerde, sözlü ve yazılı olarak iletişim kurabilen sistem paydaşlarına ihtiyaç duyulmaktadır. Böylece sistem analistlerinin uluslararası işletmelerde çalışma imkânı genişlemektedir.

### **Elektronik Ticaret ve İş**

Ekonominin küreselleşmesi ve İnternet'in yaygınlaşması sonucu, işletmeler işlerini elektronik ticaret (e-ticaret) ve elektronik iş (e-iş) şekline dönüştürmekte veya genişletmekte dir. İnternet işlerin yapılmış kurallarını değiştirmektedir. E-ticaret ve e-iş kavramlarının kullanıldığı üç tip bilgi sistemi uygulaması bulunmaktadır;

- Görüntü, ürün ve hizmetlerin pazarlanması en basit e-ticaret uygulamasıdır. Müşterileri işletmenin ürünleri, hizmetleri ve politikaları hakkında bilgilendirmek için web ortamı kullanılır. İşetmelerin çoğunluğu bu tür e-ticaret uygulamasını gerçekleştirmektedir.
- İşletme-müşteri e-ticareti: Günümüzde müşteriler İnternet üzerinden bir ürünü araştırmakta, sipariş verebilmekte ve ücretini ödeyebilmektedir. İnternet üzerinden ürünler, yemekler, konser biletleri benzeri ürünler satın alıbmaktadır.
- İşletme-şirket e-ticareti bu tür ticaretin en karmaşık olanıdır ve kaçınılmaz olarak e-iş şekline dönüşecektir. Bu tür e-ticaretin bir örneği elektronik ihaledir. Tüm işletmeler hamaddeler, donanım ve gereçler satın almaktadır. İşetmeler arası e-ticaretle çalışanlar, elektronik katalogları inceleyebilmekte, ürün ve servisleri ismarlayabilmekte ve ücretleri ödeyebilmektedir. Böylece, bu işlemler geleneksel, zaman kaybedici ve pahalı kâğıt akışı veya bürokrasi olmadan yapılabilmektedir.

Sunucu ve ağ mimarilerindeki gelişmeler ve İnternet kullanımının yaygınlaşması nedeniyle yeni bilgi sistemlerinin çoğunluğu e-ticaret ve e-iş uygulamalarını destekleyecek biçimde tasarılanmakta ve geliştirilmektedir.

### **Güvenlik ve Gizlilik**

Sayısal ekonomi gelişmişçe insanlar ve işletmeler güvenlik ve gizlilik konularını öne çıkarmaktadır. "Güvenlikle ilgili konularda çalışanlar, ticari aktiviteleri kesintiye uğratabilir bir felaket veya ihlalde ticaret nasıl devam edecektir?", "İşletme sayısal varlıklarını dış tehditlerden nasıl koruyacaktır?" gibi sorulara cevap aramaktadır. Bu sorular temelde teknolojik olarak çözümlenebilir ancak her güvenlik tedbirine karşı zaman içinde bir saldırı yöntemi geliştiriliyor olması işletmelerin temel kaygısı olmaktadır.

Güvenlikle ilgili bir diğer konu ise gizliliktir. Müşteriler sayısal ekonomide, artan oran da gizlilik talep etmektedir. Hükümetler de gizlilik konularında düzenlemeler yapmaktadır. Bu nedenle, hemen her işletmenin bir gizlilik politikası vardır. Günümüzde müşteriler grupları gizlilik politikalarını izlemekte, işletmeleri sorumlu tutmakta ve hükümetlerden daha sık düzenlemeler ve uygulamalar talep etmektedir.

Bilgi sistemleri gelişmeye ve değişimeye devam ederken sistem tasarımcıları, sistem kurucuları ve sistem analistlerinden daha sıkı güvenlik ve gizlilik kontrolü yapmaları, küresel ekonomide bir ülkeden diğerine büyük oranda değişen düzenlemeleri bilmeleri ve uygulamaları beklenmektedir.

### **İş Birliği ve Ortaklık**

İş birliği ve ortaklık, bilgi sistemi uygulamalarını etkileyen önemli bir iş eğilimidir. Günümüzde yeni bir ürünün tasarımı, işletmedeki farklı birimlerden çapraz fonksiyonlu takımların oluşturulmasını gerektirmektedir. Yönetimler, işletmenin farklı bölümleri ve işlevlerini ayıran duvarların yıkılmasını ve çapraz fonksiyonlu takımların iş birliği yaparak işletmenin ortak hedeflerine ulaşılmasını istemektedir. Bu takımlarda mühendislik, pazarlama, satış, üretim, envanter kontrolü, dağıtım ve bilgi sistemleri temsilcileri bulunmalıdır.

Benzer şekilde, iş birliği eğilimi işletme dışına taşarak diğer işletmelerle hatta rakiplerle birlikte çalışacak şekilde genişlemektedir. İşletmeler ortak girişimle doğrudan iş birliği yaparak daha etkin olabilmektedir. Örneğin Microsoft ve Oracle her biri kendi veritabanı yönetim sistemini satmakta ve rekabet etmektedir. Ancak Microsoft ve Oracle, Oracle uygulamalarının Microsoft işletim sistemleri üzerinde çalışabilmesi için iş birliği yapmaktadır. Sonuç olarak, her iki işletme bu iş birliğinden finansal olarak yararlanmaktadır.

### **Bilgi Varlığı Yönetimi**

Bilgi, ham verilerin işlenerek faydalı enformasyona dönüştürüldüğü bir sürecin sonucudur. Bilgi sistemleri, iş gerçeklerini (ürünler, çalışanlar, müşteriler vb.) ve ticari işlemleri işleyerek ham verileri toplar. Veriler yöneticilere işleri planlayacak ve yönetecek enformasyon üretmek için birleştirilir, ayıklanır, düzenlenir ve analiz edilir. Sonuçta enformasyonun kişiler tarafından özümsemenesile bilgi ve uzmanlık ortaya çıkar. İşletmeler kendilerine "Rekabetçi avantaj için bilgi nasıl yönetilmeli ve paylaşılmalı?" sorusunu sormaktadır. Ayrıca bazı çalışanlar işten ayrılp yerine yeni çalışanlar geldiğinde işletmedeki bilgi ve uzmanlığın nasıl korunabileceği önemlidir. Bu konular bilgi varlığı yönetimi ihtiyacını ortaya çıkarmaktadır. Bu ihtiyaç bilgi sistemlerini çeşitli şekilde etkilemektedir. Bilgi sistemlerinde büyük miktarda veri ve bilgi toplanmaktadır. Ancak birçok işletmede tekrarlayan ve çelişen veri ve bilgi bulunmaktadır. Yeni bilgi sistemi geliştirilirken tekrarlayan ve çelişen veri ve bilgiler gözden geçirilerek düzeltilmeli, diğer mevcut veri ve bilgilerin korunması sağlanmalıdır.

### **Sürekli İyileştirme ve Toplam Kalite Yönetimi**

Bilgi sistemleri iş süreçlerini otomatikleştirir ve destekler. Sürekli iyileştirme, bir seri ufak değişikliklerle iyileştirme yapmak için iş süreçlerini denetler. Süreçlerdeki değişiklikler maliyetlerde azalmaya, iyileştirilmiş etkinliğe veya artan değer ve kazanca sebep olabilir. Sistem analistinin bilgi sistemini tasarlama ve uygulama aşamalarında bu türden iyileştirme faaliyetlerini önermesi veya başlatması beklenmektedir.

Bir diğer önemli faktör de toplam kalite yönetimidir. Kalite, rekabette önemli bir başarı faktörüdür ve kalite yönetimi işletmenin sattığı ürünlerle başlayıp bitmez. Kalite yönetimi, işletmedeki herkesin sorumlu olduğunu kabul eden bir kültürle başlar. Toplam kalite yönetimi, bilgi servisleri de dahil olmak üzere tüm işletme fonksiyonlarının kalite göstergelerinin tanımlamasını, kalitenin ölçülmesi ve iyileştirilmesi için uygun değişiklikler yapılmasını gerektirir.

### **İş Süreçlerinin Yeniden Tasarımı**

İşletmelerin iş süreçlerinde yıllarca kayda değer bir değişiklik yapılmayabilir veya bu süreçler etkisiz ve pahalı olabilir. Bazı işletmelerin iş süreçleri de aşırı bürokratiktir ve atılan adımlar işletmeye bir değer katmayabilir. Bu nedenle iş süreçlerinin yeniden tasarımı, geniş bir zaman aralığında iş süreçlerinde önemli değişiklikler yapılması ile ilgilendir. Bu kapsamda, iş süreçleri zamanlama, darboğazlar, maliyetler ve her adımın ya da görevin işletmeye bir değer katmadığı açısından dikkatlice analiz edilir ve sonuçlar kayıt altına alınır. Daha sonra iş süreçleri maksimum etkinlik ve minimum maliyet elde etmek için yeniden tasarlanır.

İş süreçlerinin yeniden tasarlanması bilgi sistemlerini etkilemektedir. Bu nedenle bilgi sistemi yeni iş süreçlerine uyumlu hâle getirilmelidir. Herhangi bir bilgi sistemini iş süreçlerine uyarlamanın temelde iki yolu bulunmaktadır. Bunlar bilgi sisteminin işletmede geliştirilmesi veya satın alınmasıdır. Bilgi sisteminin işletmede geliştirilmesi durumunda öncelikle iş süreçleri yeniden tasarlamlı daha sonra bu süreçleri otomatik yapacak yazılım geliştirilmelidir. Böylece mevcut sistemdeki yetersizliklerin otomatikleşmesi engel-

lenir. Benzer işletmeler için geliştirilmiş olan ve standartlaşmış bilgi sistemi yazılıminın satın alınması durumunda ise alınan yazılımin mevcut iş süreçlerine uyumlu çalışması zorlanmamalıdır. Aksine iş süreçleri alınan yazılımla uygun çalışacak şekilde yeniden düzenlenmelidir.

## Teknolojik Faktörler

Bilgi teknolojileri de bilgi sistemleri için itici bir faktör olabilir. Eskimiş teknolojiler önemli problemlere ve bu problemlerin çözümü için bilgi sistemi projelerinin geliştirilmesine sebep olabilir. Yeni teknolojiler ise iş fırsatları ortaya çıkarabilir. Aşağıda günümüz bilgi sistemlerini etkileyen teknolojiler sunulmaktadır.

### Ağlar ve Internet

Günümüz bilgi teknolojileri yerel ve geniş alan ağlarından oluşan bir ağ mimarisi üzerine kurulmaktadır. Bu ağlar ana bilgisayar, ağ sunucuları ve çeşitli masaüstü, dizüstü, tablet bilgisayarlar tarafından kullanılmaktadır. Günümüzdeki en yaygın ağ teknolojileri İnternet tabanlıdır. Aşağıda bazı önemli İnternet teknolojileri anlatılmaktadır.

- HTML 5, xHTML ve XML, Web sayfası hazırlama ve İnternet uygulaması geliştirmekte kullanılan temel dillerdir. HTML 5, HTML işaretleme standardının beşinci sürümüdür. xHTML Web sayfası oluşturmakta kullanılan HTML dilinin ikinci neslidir. XML ise İnternet'i kullanarak veri alışverişi yapan sistemler ve platformlar arasındaki veri iletişimini standart hâle getirmek için tasarlanan bir işaretleme dilidir. Giriş düzeyi HTML 5 ve XML dersleri birçok üniversitede bilgi sistemleri programlarının temel dersi olarak sunulmaktadır.
- Script dilleri, özellikle İnternet uygulamaları için geliştirilmiş programlama dilleridir. Örnek olarak Perl, VBScript ve JavaScript verilebilir. Bu diller Web geliştirme ve programlama derslerinde öğretilmektedir.
- Java ve Cold Fusion gibi programlama dilleri, birçok sunucu ve web tarayıcı içeren web tabanlı karmaşık uygulamalarda kullanılmak üzere geliştirilmiştir. Bu dersler de üniversitelerde öğretilmektedir.
- Intranet, bir işletmedeki çalışanların kullanması için tasarlanan özel ağ mimarisi dir. İnternet altyapı cihazlarını kullanır ancak güvenlik ve güvenlik duvarları, kullanıcıların sadece işletme çalışanları olması şeklinde düzenlenir.
- Extranet, işletmeler arası kullanım için geliştirilen özel ağ mimarisidir. İnternet alt yapısı kullanılmakla beraber, güvenlik ve erişim tanımlamaları ile sadece belirtilen işletmelerin çalışanları extranet'e erişim sağlayabilir. Örneğin bir otomobil üreticisi sadece bayilerinin kullanacağı bir extranet kurabilir. Bu extranet aracılığıyla üretici parçalar, problemler, satış teşvikleri vb. bilgiyi bayileri ile paylaşabilir.
- Portal, farklı kullanıcıların ihtiyaçları doğrultusunda yapılandırılmış ana sayfadır. Örneğin portal teknolojisi kullanılarak aynı şirkette farklı rolleri olan çalışanlara uygun bilgiler sağlayan web sayfaları tasarlabilir. Her kişinin rolü web sayfasında hangi bilgiyi ve uygulamayı kullanabileceğini belirler. Rollere örnek olarak "müşteri", "tedarikçi" ve farklı tipte "çalışan" verilebilir. Portal aynı zamanda kullanıcıların kişisel web sayfalarını İnternet, intranet ve extranet içeriklerine entegre edebilir.
- Web servisleri, yeniden kullanılabilen ve web tabanlı programlar veya başka İnternet programları tarafından çağrılabilen program parçalarıdır. Örneğin kredi kartı ödemelerini İnternet üzerinden kabul eden bir programa ihtiyaç duyulur. Çözüm olarak kredi kartı onaylamasını yapacak programı kendimiz geliştirebiliriz. Buna alternatif bir diğer yaklaşım ise web üzerinden kredi kartı onayı yapabilen bir web servis programının kullanım hakkının satın alınması olabilir. Bu durumda kredi kartı onaylama kodunun kontrol sorumluluğu alınmamış olur.

### **Mobil ve Kablosuz Teknolojiler**

Mobil ve kablosuz teknolojiler bilgi sistemlerini büyük oranda değiştirmektedir. Günümüzde akıllı telefonlar ve tabletler mobil uygulamalarda, web göz atmasında ve e-posta iletiminde yaygın olarak kullanılmaktadır.

### **Nesne Teknolojileri**

Günümüzde birçok bilgi sistemi nesne teknolojileri kullanarak geliştirilmektedir. Bu konuda yaygın olarak kullanılan C++, Java, Smalltalk, Visual Basic.NET ve C# gibi programlama dilleri nesne tabanlıdır. Nesne teknolojileri programcılar yazılımı nesne olarak adlandırılan program parçalarından oluşturmaya izin verir. Nesne tabanlı yazılım iki önemli avantaj sunmaktadır. Birincisi, nesneler bir defa tasarılanıp geliştirildikten sonra birçok bilgi sistemi ve uygulamada yeniden kullanılabilir. Bu ilerde geliştirilecek yazılım uygulamaları için gerekli zaman ve maliyeti azaltacaktır. İkinci avantaj ise nesnelerin genişleyebilir olmasıdır. Nesneler, tasarlandıkları uygulamaları olumsuz etkilemeksiz kolaylıkla değiştirilebilir ve genişletilebilir. Bu durum yazılımın sürdürülmeye ve iyileştirme maliyetlerini azaltır.

### **İş Birlikçi Teknolojiler**

İletişim ve İş birliği sistemi (groupware) aynı bilgiler üzerinde, farklı ortamlarda (bina, kat, oda) ve ortak bir ağ üzerinde çalışan bir grup insanın kullanımını için tasarlanan yazılımlara denir.

### **Kurumsal Uygulamalar**

Küçük veya büyük tüm işletmelerin işlerini yürütmek için kurumsal uygulamaların çekirdek bir sete ihtiyacı vardır. Birçok işletme için çekirdek uygulamalar finansal yönetim, insan kaynakları yönetimi, pazarlama ve satış, envanter ve üretim kontrolüdür. Bir zamanlar işletmelerin kendilerinin geliştirdiği bu uygulamalar günümüzde satın alınıp ve kullanılmaktadır. Yukarıdaki çekirdek uygulamalara ilave olarak günümüzde, işletmeyi tedarikçileri ve müşterileri ile bütünleştirecek uygulamalar da kullanılmaktadır. Bu uygulamalar müşteri ilişkileri yönetimi ve tedarik zinciri yönetimi olarak adlandırılmaktadır.

Satın alınan kurumsal uygulamaların kullanım eğilimi sistem analizi ve tasarımını önemli oranda etkilemektedir. Satın alınan ve kurulan uygulamalar işletmedeki bilgi sistemlerinin tüm gereksinimlerini tam olarak karşılamayabilir. Bu durumda sistem analisti ve diğer geliştiriciler, işletmenin ilave gereksinimlerini yerine getirmek üzere katma değeri yüksek uygulamalar geliştirmelidir. Bazen satın alınan uygulama bir teknoloji kısıtına sahip olabilir. Bu durumda, geliştirilecek herhangi bir uygulama satın alınan uygulama ile entegre olabilmelidir. Bu işlem sistem entegrasyonu olarak adlandırılır. İzleyen kesimde yaygın olarak kullanılan bazı kurumsal uygulamalar ve sistem analizi ve tasarımını üzerindeki etkileri ele alınmıştır.

*Kurumsal Kaynak Planlaması:* 1990 larda birçok işletmede geliştirilen bilgi sistemi uygulamalarının kullanışsız olarak birleştirilmiş dosyaları ve veri tabanları bulunmaktadır. Pek çok işletme bu sistemleri düzeltmek için çalışmış ancak genelde başarısız olmuştur. Bu çalışmalar sonucunda, birçok işletmenin ihtiyacı olan temel uygulamaların birbirine benzer olduğunu fark eden yazılım endüstrisi, çözüm olarak kurumsal kaynak planlamasını (ERP-Enterprise Resource Planning) geliştirdi. Bir ERP çözümü, işletme fonksiyonları tarafından ortak kullanılan veritabanı etrafında kurgulanır. ERP çözümü tüm işletme için çekirdek bilgi sistemi fonksiyonlarını sağlar. Ancak işletmeler ERP çözümünü tam

olarak kullanabilmek için iş süreçlerini yeniden tasarlamalıdır. Birçok işletme kendi işlerine özgü gereklilikleri yerine getirmek için ERP çözümünü, geliştirilen yazılım uygulamaları ile desteklemektedir. ERP uyarlaması ve entegrasyonu çoğu işletme için en büyük bilgi sistemi projesi olabilir. Bunun için yüksek miktarlarda para harcanması ve yönetici, kullanıcı, analist, teknik uzman, programcı ve danışmanlardan oluşan geniş bir çalışma ekibi gerekecektir.

ERP uygulamaları sistem analisti için birçok nedenden dolayı önemlidir. Sistem analisti öncelikle ERP çözümünü seçme ve satın alma süreciyle ilgilenecektir. İkinci olarak, ERP çözümünün işletmeye uyarlanması ve kullanılabilmesi için işletmenin iş süreçlerini yeniden tasarlayacaktır. Günümüzde ERP, tedarik zinciri yönetimi ve müşteri ilişkileri yönetimi olarak adlandırılan yeni iş sistemleriyle desteklenmektedir.

*Tedarik Zinciri Yönetimi:* Günümüzde işletmeler, tedarikçileri ve bayileriyle ham madde ve ürünlerin daha etkin akışı için çekirdek iş uygulamalarını genişletmektedir. Tedarik zinciri yönetimi (SCM-Supply Chain Management) uygulamaları, entegrasyon ve haberleşme için İnternet'i kullanmaktadır. Örneğin bir restoran zincirindeki tedarik zinciri birçok işletmeyi ve nakliyeciyi kapsamaktadır. Zincirdeki herhangi bir halkada olacak gecikme veya problem zincirdeki her işletmeyi olumsuz etkileyecektir. Bu nedenle birçok işletme, zincirdeki rollerini planlama, uygulama ve yönetmek için SCM yazılım tekniklerini kullanarak tedarik zinciri yönetimi kurmaktadır. Sistem analisti uygun SCM yazılım paketinin seçimi, uygulaması ve gerekirse işletmenin ihtiyaçlarını karşılamak için uyarlanması ile ilgilenir. Bazı durumlarda seçilen SCM çözümüne uyum sağlamaası için işletmenin iş süreçlerinin yeniden tasarlanması da sistem analistinin görevi olabilir.

*Müşteri İlişkileri Yönetimi:* Birçok işletme, yüksek derecede odaklanmış müşteri ilişkileri yönetiminin müşterilerde işletme bağılılığı kazandırdığı ve satışları artttığını keşfetmiştir. Bu nedenle işletmeler, müşterilerin İnternet üzerinden işlemlerini yapabilmesi için müşteri ilişkileri yönetimi (CRM-Customer Relationship Management) çözümlerini kullanmaktadır. CRM çözümlerinin ana teması müsteridir. CRM, müşterilerin ihtiyaçlarını karşılamadan yanı sıra müşteri ilişkilerini ve pazarlamayı iyileştirmek için müşteri profili hakkında bilgi sağlamaktadır. CRM teknolojisi, ERP ve SCM teknolojileriyle benzer bir biçimde sistem analistini etkilemektedir.

Birçok işletme mevcut sistemleri ile ERP, SCM ve CRM çözümlerini birleştirme konusunda zorlanmaktadır. İnternet üzerinde iş yapmak isteyen bir işletme de kendi sistemlerini diğer işletmelerin farklı sistem ve teknolojileri ile birlikte kullanmak istediği içinde uyumlandırma problemiyle karşılaşmaktadır. Bu zorlukları yenebilmek için işletmeler kurumsal uygulamaların entegrasyonu (EAI-Enterprise Application Integration) yazılımına ihtiyaç duymaktadır. EAI, satın alınan veya işletmede geliştirilen uygulamaları birbirleri ile şeffaf olarak çalışabilecek şekilde ilişkilendirmektedir. Bu amaçla çeşitli uygulamalar ve teknolojiler arasındaki iletişim yollarını tanımlamak ve oluşturmak için farklı araçlar kullanılmaktadır. Günümüzde geliştirilen yeni bir bilgi sistemi, kendisinden önce geliştirilen tüm bilgi sistemleriyle entegre edilmelidir. Her durumda sistem analistleri geliştirilen yeni bilgi sistemi için kurumsal uygulama entegrasyonunu gerçekleştirmelidir.

## SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

Sistem geliştirme yaşam döngüsü (SGYD), sistem analizi ve tasarımının, analist ve kullanıcı faaliyetlerinin özel yöntemler aracılığıyla geliştirildiği, fazlara bölünmüş bir yaklaşımındır. SGYD içinde kaç faz olacağı konusunda genel bir kabul bulunmamaktadır. Bu kitapta anlatılacak SGYD altı fazlı olacaktır. Bu fazlar, Kapsam ve Planlama, Gereksinim Analizi, Tasarım, Gerçekleştirme ve Test, Kurulum ve Dağıtım, Operasyon ve Bakımıdır.

## Kapsam ve Planlama

Sistem geliştirmenin bu fazında sistem analisti, problemlerin, fırsatların ve amaçların belirlenmesi ile ilgilendir. Bu faz projenin geri kalanı için kritik önemdedir çünkü yanlış bir problem üzerinde çalışıp zaman kaybetmeyi hiç kimse istemez.

Bu fazda analist iş dünyasında ne olduğunu inceler ve işletmedeki diğer üyelerle problemleri tespit eder. Fırsatlar, analistin bilgisayar tabanlı bilgi sistemlerini kullanarak iyileştirileceğine inandığı durumlardır. Fırsatları yakalamak, işletmeye rekabetçi kimlik veya endüstriye yeni bir standart kazandırabilir. Amaçları belirlemek de bu fazdaki önemli bir adımdır. Analist öncelikle işletmenin yapmak istediği keşfetmeli, daha sonra bilgi sistemleri uygulamalarının bu amaçlara ulaşmakta nasıl kullanılabileceğini planlamalıdır. Bu fazla ilgili olan paydaşlar, sistem sahipleri, proje koordinasyonunu sağlayan yöneticiler ve sistem analistleridir.

## Gereksinim Analizi

Analisten ilgileneceği bir sonraki görev sistem gereksinimlerinin analizidir. Analist, veri akış diyagramları, faaliyet diyagramları gibi araçları kullanarak geliştirilen sistemdeki tüm verilerin ve özelliklerinin yer aldığı bir sözlük oluşturur.

Bu fazda analist, şartlar, şartların alternatifleri, aksiyonlar ve aksiyon kurallarını içeren yapısal kararları da analiz eder. Ayrıca analist, kullanıcılar hakkındaki bulguları, mevcut sistemlerin kullanılabilirliklerini, alternatiflerin maliyet-kâr analizlerini içeren bir rapor da hazırlar. Bu rapor doğrultusunda ne yapılması gerektiği konusunda düşüncelerini içeren bir sistem önerisi hazırlar. Öneri yönetim tarafından kabul edilirse bu yönde çalışma devam eder.

## Tasarım

SGYD'nin bu fazında, sistem analisti bilgi sisteminin mantıksal tasarımını için önceden toplanan bilgileri kullanır. Kullanıcıların bilgi sistemine doğru veri girişi yapabilmeleri için yöntemleri tasarlar. Analist, bilgi girişlerinin etkin olarak yapılabilmesi için uygun web sayfası veya ekran pencerelerini tasarlar. Mantıksal tasarımın önemli bir kısmı insan-bilgisayar arayüzüne tasarlamaaktır. Kullanıcı arayüzü duyulabilir, okunabilir, güvenli, çekici ve eğlenceli olmalıdır.

Tasarım fazı aynı zamanda, işletmedeki karar vericilerin ihtiyacı olan verilerin depolacağı veri tabanlarının tasarımını da içerir. Mantıksal olarak iyi tasarlanmış ve kullanıcının yaptığı işlere uygun bir veritabanı kullanıcılar açısından çok faydalı olacaktır.

Analist, sistemi ve verileri korumak amacıyla kontrol ve yedekleme yöntemlerini de tasarlar. Programcılar için program özellik paketlerini ortaya koyar. Paket, giriş-çıkış yapılarını, dosya özelliklerini, işleme detaylarını, karar ağaçlarını veya tablolarını, UML veya veri akış diyagramlarını, önceden yazılmış programların adlarını ve fonksiyonlarını içermelidir.

## Gerçekleştirme ve Test

Tasarım fazında özellikleri belirlenen sistem bu fazda gerçekleştirilecek ve test edilir. Programcılar sistemi oluşturan yazılımı ve eski sistem ile yeni sistem arasındaki arayüzleri geliştirir. Programcılar ve analistler program parçalarında ve tüm sistemde oluşabilecek hataları bulmak ve düzeltmek üzere testler uygular. Testler başlangıçta örnek veriler ile yapılrken teslimden önce gerçek veriler ile tamamlanır. Bu fazda analist, kullanıcılarla birlikte çalışarak yazılım, kullanım prosedürleri, çevrimiçi yardım, sıkça sorulan sorular kısımlarını içeren belgeyi hazırlar. Bu belge, yazılımın nasıl kullanılacağını ve problem çıkışına nasıl çözümleneceğini anlatmalıdır.

## Kurulum ve Dağıtım

Bu fazda analist bilgi sisteminin kurulumuna yardım eder. Kullanıcılar yazılım sağlayıcıları tarafından eğitim verilir. Genelde eğitimleri düzenleyen ve yöneten analisttir. Analist aynı zamanda eski sistemden yenisine düzgün ve sorunsuz geçiş planları. Bu süreç eski formatta yazılan programların yeni formata dönüştürülmesi, bir veritabanının kurulması, cihazların kurulumu ve yeni sistemin çalışır hâle getirilmesini içerir.

## Operasyon ve Bakım

Sistem kurulduktan sonra sürdürülmelidir. Diğer bir ifade ile bilgisayar programları uyarlanmalı ve güncel tutulmalıdır. Sistemi sürdürmek için harcanan zamanın, sistem geliştirme için harcanan zamanın %60'ına kadar ulaşabileceği tahmin edilmektedir. Yazılan programların sayısı arttıkça sistemin sürdürülebilme yükü de artmaktadır.

Bakım iki nedenle yapılır. Birincisi yazılım hatalarının düzeltilmesidir. Sistem ne kadar test edilirse edilsin böcekler (bugs) ve hatalar oluşabilir. Ticari programlarda böcekler yeni sürümde veya bir ara sürümde düzeltilebilir. Ancak özel olarak geliştirilen programlarda böcekler algılandığı anda düzeltilmelidir.

Sistem bakımının ikinci sebebi de değişen işletme ihtiyaçlarını karşılamak için yazılım kapasitesinin iyileştirilmesidir. İşletme ihtiyaçları aşağıdaki üç durumdan biri nedeniyle değişimdir:

- i. Kullanıcılar bilgisayar sistemi ve kabiliyetlerine aşina olduktan sonra ilave özellikler isteyebilir.
- ii. İşin doğası zamanla değişimdir.
- iii. Yazılım ve donanım teknolojileri hızlı bir şekilde değişmektedir.

Özetle bakım, bilgi sisteminin hayat döngüsü içinde sürekli var olan bir süreçtir. Bilgi sistemi kurulduktan sonra bakım, önceden belirlenmemiş hataların düzeltilmesi şeklini alır. Bu hatalar düzeldikçe sistem olgun bir duruma gelir ve kullanıcılar güvenilir hizmet vermeye başlar. Bu dönemde önceden bulunamamış birkaç böcek temizlenir veya ufak iyileştirmeler yapılır. Ancak zaman geçtikçe iş ve teknoloji değişimlerinden dolayı bakıma harcanan zaman ve para önemli ölçüde artmaya başlar. Bakım maliyetlerinin çok fazla artmasıyla birlikte mevcut sistemin yaşam döngüsünün sonuna gelinmiş olur. Artık yeni bir sistem geliştirmenin zamanı gelmiştir.

## Özet



### Sistem ve bilgi sisteminin tanımlamak

Sistem, istenilen bir sonucu oluşturmak için birbiriyle ilişkili ve birlikte çalışan bir grup bileşendir. Bilgi sistemi, veri toplamak, depolamak ve işlemek, bu verilerden sayısal bilgi üretmek amacıyla biraraya getirilmiş, birbiriyle uyumlu bileşenler bütündür. Bilgi sisteminin bileşenleri; donanım kaynakları, yazılım kaynakları, veri kaynakları, ağ kaynakları ve insan kaynaklarıdır.



### Farklı bilgi sistemlerini ayırdetmek

Sistemler yerine getirdikleri işlevlere göre sınıflandırılabilir. *Kayıt İşleme Sistemleri*, işletmenin operasyonel düzeydeki işlemlerini kaydeden bilgisayar tabanlı sistemlerdir. *Ofis Otomasyon Sistemleri*, yeni bilgi üretmeyen, ancak gerektiğinde bilgi üzerinde değişiklik yapan ve sonuçları ilgili birimlerle paylaşarak çalışanları destekleyen sistemlerdir. *Yönetim Bilgi Sistemleri*, yöneticilerin ihtiyaç duyduğu bilgileri üretmek için işlem verilerini kullanan ve sonuçlarını rapor halinde yöneticiye sunan sistemlerdir. *Karar Destek Sistemleri*, karar vericilerin alternatifler arasından seçim yapabilmesine yardımcı olacak şekilde bilgi üreten sistemlerdir. *Uzman Sistemler*, belli bir konudaki uzmanın düşüncesini taklit ederek o konudaki bir probleme çözüm üretebilen sistemdir. *Grup Karar Destek Sistemleri*, bir grubun belli bir problemi çözmesine yardımcı olmak için geliştirilen sistemlerdir.



### Bilgi sistemi paydaşlarının işlevlerini betimlemek

*Sistem Sahibi*, sistemin maliyeti ve kurulacak sistemin işletmeye kazandıracakları ile ilgilenen, yönetim kademesinden bir kişidir. *Sistem Kullanıcıları*, geliştirilen sistemi kullanan kişilerdir. Dâhilî ve harici kullanıcılar olarak ikiye ayrılırlar. *Sistem Tasarımcıları*, bilgi sistemine ihtiyaç duyduğu teknolojik katkıyı sağlayan, bilgi sisteminin farklı alt birimlerini tasarlayan kişilerdir. *Sistem Kurucuları*, sistem tasarımcılarının belirttikleri özellikler doğrultusunda sistemi kuran uzman grubudur. *Sistem Analisti*, işletmedeki insanların, süreçlerin ve bilgisayar teknolojisinin işlerin en iyi biçimde yürütülmesi amacıyla analiz edilmesiyle ilgilenen kişidir. İşletmenin örgütsel süreçlerinin iyileştirilmesi amacıyla girilen verileri inceler. İncelediği verilerden kullanıcıların teknolojiyle nasıl etkileştiğini ve işletmenin nasıl çalıştığını, problemlerini ve fırsatlarını belirler. İşletme ve bilgi gereksinimlerini,

farklı teknik uzmanlar tarafından gerçekleştirilecek bilgi sistemi özelliklerine dönüştürür. *Harici Servis Sağlayıcıları*, bir projeye işletme dışından katkı sağlayan ve belli konularda uzman olan kişilerdir. *Proje Yöneticisi*, bilgi sisteminin zamanında, belirlenen bütçe sınırlarında ve kalite düzeyinde geliştirilmesini sağlayan kişidir.



### Bilgi sistemlerini etkileyen faktörleri açıklamak

Bilgi sistemlerini etkileyen faktörler ticari eğilimler ve teknolojik gelişmeler olarak iki grupta toplanabilir. Ticari eğilimler: küresel ekonomi, elektronik ticaret ve iş, güvenlik ve gizlilik, işbirliği ve ortaklık, bilgi varlığı yönetimi, sürekli iyileştirme ve toplam kalite yönetimi, iş süreçlerinin yeniden tasarımları başlıklarında toplanabilir.

*Ekonominin küreselleşmesi*; farklı yabancı dilleri, döviz kuru değişimlerini, uluslararası ticaret kuralları ile farklı ülkelerin iş kültürü ve pratiklerini içeren bilgi sisteminin geliştirilmesini gerektirmektedir. İnternetin yaygınlaşması ve küresel ekonomi *elektronik ticaret ve iş uygulamalarının hızla gelişmesine* sebep olmuştur. Bu uygulamaları destekleyecek bilgi sistemlerinin geliştirilmesi önem kazanmıştır. Elektronik ticarette *güvenlik ve gizlilik* son derece önemlidir. Bu nedenle işletmeler güvenli bilgi sistemleri geliştirilmesini talep etmektedir. İşletmeler, işletme içindeki birimlerin ortak çalışması ve benzer şekilde diğer işletmelerle *ortak çalışma ve işbirliği* yapabilmesi için uygun bilgi sistemlerinin geliştirilmesine ihtiyaç duymaktadır. İşletmede mevcut bilgi ve verilerin yeni kurulan bilgi sistemine doğru aktarılması, bulunan çelişkili veya tekrarlayan veri ve bilgilerin ayıklanması, personel değişimlerinden işletmedeki bilgi birikiminin en az etkilenmesi için *bilgi varlığı yönetimi* uygulamaları önem kazanmaktadır. Ortaya çıkan yeni gereksinimler bilgi sisteminde *sürekli iyileştirmeler* yapılmasını, *toplum kalite yönetimi* için bilgi sisteminin sahip olacağı kalite göstergelerinin tanımlanmasını gerektirir. Güncellliğini kaybeden *iş süreçlerinin yeniden tasarılanması*, yeni iş süreçleri ile uyumlu bilgi sistemlerinin geliştirilmesine yol açar. Teknolojik faktörler başlığı altında; ağlar ve Internet, mobil ve kablosuz teknolojiler, nesne teknolojileri, iş birlikçi teknolojiler, kurumsal uygulamalar sayılabilir. Günümüz bilgi teknolojileri, ana bilgisayar, ağ sunucuları ve çeşitli masaüstü, dizüstü, tablet bilgisayarlar

tarafından kullanılan *ağlar ve Internet* üzerinde kurulmaktadır. Bu ağların kurulumu ve gerekli programların yazılması bilgi sistemleri geliştirilmesinde önemli bir yer tutar. Akıllı telefonlar ve tabletler gibi *mobil ve kablosuz teknolojilerin* uygulamalarda, web göz atmasında ve e-posta iletiminde yaygın olarak kullanılması, bu araçların bilgi sistemlerine entegrasyonu için uygulamaların geliştirilmesini gerektirmektedir. Bir bilgi sistemi için geliştirilen uygulamanın bir başka bilgi sisteminde kullanılabılmasına imkân sağlayan *nesne tabanlı teknolojiler*, bilgi sistemlerinin geliştirilme sürecini kısaltmakta ve maliyetleri düşürmektedir. Kişiler arası iletişimi ve takım çalışmasını iyileştiren önemli *iş birlikçi teknolojiler*; e-posta, anlık mesajlaşma, iletişim ve işbirliği sistemi ve iş akışıdır. İşletmelerin işlerini yürütürken bazı *kurumsal uygulamaları* kullanmaları gerekmektedir. Bu kurumsal uygulamaların geliştirilmesi, işletmeye uyarlanması ve sürdürülmesi bilgi sistemi geliştiricileri için önemli bir çalışma alanıdır. Kurumsal kaynak planlaması (ERP), tedarik zinciri yönetimi (SCM), müşteri ilişkileri yönetimi (CRM) önemli kurumsal uygulama yazılımlarıdır. İşletmeler mevcut sistemleri ile ERP, SCM ve CRM çözümlerini birleştirme konusundaki zorlukları yenebilmek için kurumsal uygulamaların entegrasyonunu (EAI) yaparlar.



Sistem geliştirme yaşam döngüsünün fazlarını sıralamak

Sistem geliştirme yaşam döngüsü (SGYD) sistem analiz ve tasarımının, analist ve kullanıcı faaliyetlerinin özel yöntemler aracılığıyla geliştirildiği, fazlara bölünmüş bir yaklaşımındır. SGYD'nin başlıca fazları: Kapsam ve Planlama, Gereksinim Analizi, Tasarım, Gerçekleştirme ve Test, Kurulum ve Dağıtım, Operasyon ve Bakımdır.

## Kendimizi Sınayalım

- 1.** Bir işletmenin ihtiyaçlarını destekleyecek şekilde bilgi üretmek üzere verileri toplayan, işleyen, depolayan; insan, veri, süreçler ve bilgi teknolojilerinin etkileşimde bulunduğu yapıya ne ad verilir?

  - a. Yazılım sistemi
  - b. Öğrenci bilgi sistemi
  - c. Bilgi sistemi
  - d. Donanım sistemi
  - e. Hastane bilgi sistemi
- 2.** Karar vericilerin alternatifleri belirlemesine ve bu alternatifler arasından birini tercih etmesine yardımcı olan bilgi sistemi aşağıdakilerden hangisidir?

  - a. Yazılım sistemi
  - b. Karar destek sistemi
  - c. Yönetim sistemi
  - d. Coğrafi bilgi sistemi
  - e. Donanım
- 3.** Aşağıdakilerden hangisi bilgi sisteminde donanım elemanı **değildir**?

  - a. Bilgisayar
  - b. Klavye
  - c. Sunucu
  - d. Monitör
  - e. Program
- 4.** Aşağıdakilerden hangisi bilgi sistemi kurucularından biri **değildir**?

  - a. Veritabanı programcılar
  - b. Sistem sahibi
  - c. Webmaster
  - d. Ağ yöneticileri
  - e. Yazılım bireştiricileri
- 5.** Aşağıdakilerden hangisi sistem analistinin sahip olması gereken yeteneklerden **değildir**?

  - a. Kişiler arası iyi iletişim
  - b. Problem çözebilme
  - c. Esneklik ve uyumluluk
  - d. Programlama deneyimi
  - e. Bilgisayar tamir becerisi
- 6.** Sistem analistinin **en önemli** görevi aşağıdakilerden hangisidir?

  - a. İşletmede değişim başlatmak
  - b. Program yazmak
  - c. Sistemdeki hataları bulmak
  - d. Bilgi sisteminde kullanılacak donanımı belirlemek
  - e. Bilgi sistemi analizini yapmak
- 7.** Sistem analiz ve tasarımının, analist ve kullanıcı faaliyetlerinin özel yöntemler aracılığıyla geliştirildiği, fazlara bölünmüş süreçler toplamına ne ad verilir?

  - a. İnternet döngüsü
  - b. Yazılım döngüsü
  - c. Sistem geliştirme yaşam döngüsü
  - d. İşletme yaşam döngüsü
  - e. Web döngüsü
- 8.** Aşağıdakilerden hangisi Sistem Geliştirme Yaşam Dönüşü aşamalarından biri **değildir**?

  - a. Kapsam ve Planlama
  - b. Gereksinim Analizi
  - c. Gerçekleştirme ve Test
  - d. Operasyon ve Bakım
  - e. Satınalma ve Geliştirme
- 9.** Aşağıdakilerden hangisi bir kurumsal uygulama yaklaşımı **değildir**?

  - a. Kurumsal kaynak planlaması
  - b. Kurumsal uygulamalar entegrasyonu
  - c. Müşteri ilişkileri yönetimi
  - d. Tedarik zinciri yönetimi
  - e. Karar destek sistemi
- 10.** Masaüstü yayincılık, kelime işlemci, belge görüntüleme sistemleri gibi ofis süreçlerinde kullanılan bilgi sistemi aşağıdakilerden hangisidir?

  - a. Ofis otomasyon sistemi
  - b. Karar destek sistemi
  - c. Uzman sistem
  - d. Grup karar destek sistemi
  - e. Kayıt işleme sistemi

## Kendimizi Sınavalım Yanıt Anahtarı

- |       |   |
|-------|---|
| 1. c  | Yanıtınız yanlış ise "Sistem ve Bilgi Sistemi" konusunu yeniden gözden geçiriniz.               |
| 2. b  | Yanıtınız yanlış ise "Bilgi Sistemlerinin Sınıflandırılması" konusunu yeniden gözden geçiriniz. |
| 3. e  | Yanıtınız yanlış ise "Sistem ve Bilgi Sistemi" konusunu yeniden gözden geçiriniz.               |
| 4. b  | Yanıtınız yanlış ise "Sistem Kurucuları" konusunu yeniden gözden geçiriniz.                     |
| 5. e  | Yanıtınız yanlış ise "Sistem Analistleri" konusunu yeniden gözden geçiriniz.                    |
| 6. a  | Yanıtınız yanlış ise "Sistem Analistleri" konusunu yeniden gözden geçiriniz.                    |
| 7. c  | Yanıtınız yanlış ise "Sistem Geliştirme Yaşam Dönüşü" konusunu yeniden gözden geçiriniz.        |
| 8. e  | Yanıtınız yanlış ise "Sistem Geliştirme Yaşam Dönüşü (SGYD)" konusunu yeniden gözden geçiriniz  |
| 9. e  | Yanıtınız yanlış ise "Kurumsal Uygulamalar" konusunu yeniden gözden geçiriniz.                  |
| 10. a | Yanıtınız yanlış ise "Ofis Otomasyon Sistemi" konusunu yeniden gözden geçiriniz.                |

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Şekil 1.1'de verilen hastane bilgi sisteminin bileşenleri aşağıdaki biçimde sıralanabilir.

- Donanım kaynakları:** Sunucu ve istemci bilgisayarlar ve monitör, klavye, modem. Tahlil, EKG, röntgen gibi bilgilerin depolandığı depolama birimleri, randevu sırasını gösteren göstergeler.
- Yazılım kaynakları:** Hasta randevularının, yapılan kan, idrar vb. tahlillerinin, çekilen EKG, röntgen vb. bilgilerin bilgi sisteminin sunucu bilgisayarına depolanması, doktorun istemci bilgisayarında gösterilmesi, yazılan reçete ve raporların eczane bilgisayarında görünebilmesi vb. uygulamalar için geliştirilen yazılımlar.
- Veri kaynakları:** Hastaların kimlik bilgilerinin, tahlil sonuçlarının, çekilen EKG ve röntgenlerin vb. verilerin depolandığı veritabanları.
- Ağ kaynakları:** Hastane içindeki birimler (poliklinik, hasta kabul, kan tahlili, biyokimya, röntgen vb.) ile hastane dışındaki birimler (SGK, eczane vb.) arasındaki bilgi paylaşımı için geliştirilmiş bilgisayar ağları.
- İnsan kaynakları:** Hastane bilgi sisteminin sahipleri (başhekim, yönetici, vb.) ile hastane bilgi sistemini tasarlayan, kur'an ve kullanan kişiler. Bilgi sistemi kullanıcıları hastane personeli (doktor, hemşire, muhasebeci, eczacı vb.) gibi dâhilî kullanıcılar ile hasta ve hasta yakınları, eczacılar, SGK uzmanları ve çalışanları gibi haricî kullanıcılar.

### Sıra Sizde 2

İki sistem arasındaki farkı anlatabilmek için doktorların bir hasta hakkında karar vermesini sağlayan KDS ve US yapılarını inceleyelim. Hastanın sağlık geçmişini, ailedeki sağlık problemlerini, yapılan tahlilleri, farklı görüntüleme metotları ile elde edilen görüntüleri ve hastada mevcut belirtileri kullanarak hastanın geçirmekte olduğu hastalık için:

- KDS:** Muhtemel hastalık seçeneklerini doktora sunar. Doktor bu seçenekleri, kendi tecrübeleri ve izlenimleri doğrultusunda değerlendирerek hastalığı teşhis eder. Reçete yazar ve tedavi süreçlerini belirler. Doktorun, KDS'nin önerdiği hastalıkların dışında bir teşhis koyması da mümkündür.
- US:** Bu sistemde doktorun bilgi ve tecrübe programlanır ve US, KDS ile benzer girdileri kullanarak muhtemel hastalıkları belirler ancak bu seçenekleri doktora sunmak yerine kendisi değerlendirerek teşhis koyar, reçete yazar ve tedavi süreçlerini önerir.

**Sıra Sizde 3**

Şekil 1.1'de verilen hastane bilgi sisteminin paydaşları aşağıda belirtilmiştir.

- i. Sistem sahibi: Hastanenin başhekimi, yönetim kurulu başkanı vb.
- ii. Sistem kullanıcıları: Dâhilî ve haricî kullanıcılar olarak incelenir.  
*Dâhilî kullanıcılar:* Sekreterler, hemşireler, doktorlar, muhasebe birimi elemanları, başhekim vb. hastane çalışanları.  
*Haricî kullanıcılar:* Randevu alan veya tahlil sonuçlarını Internet üzerinden öğrenen hastalar, reçete ve raporları kullanan eczaneler, hasta katılım payını hesaplayan SGK elemanları vb. kullanıcılar.
- iii. Sistem tasarımcıları: Hastane bilgi sistemi konusunda deneyimli bilgisayar mühendisleri ve beraber çalışıkları doktorlar.
- iv. Sistem kurucuları: Sistem tasarımcıları tarafından belirlenen yapıyı kuran (bilgisayarlar, ağlar ile tahlil, görüntüleme vb. cihazları konusunda uzman) kişiler.
- v. Sistem analisti: Hastane bilgi sistemi konusunda (görevi, hastane yönetiminin belirlediği fonksiyonları sağlayacak bilgi sisteminin nasıl gerçekleştirileceğini analiz etmek, önerilerde bulunmak ve önerilerin gerçekleştirilmesi için girişimlerde bulunmak olan) uzman kişi.
- vi. Haricî servis sağlayıcıları: Hastane bilgi sisteminde kullanılan bilgisayar sistemlerini kuran ve bakımını yapanlar, yazılımları güncelleyen ve arızalarını gideren yazılımcılar, ağ sisteminde oluşacak arızaları gideren veya gerekli genişlemeleri yapan uzmanlar vb.
- vii. Proje yöneticisi: Yönetim kurulu başkanı, hastane başhekimi ya da bunların görevlendireceği herhangi bir yönetici.

**Yararlanılan ve Başvurulabilecek Kaynaklar**

- Aydın, S., (2012). **İşletme Bilgi Sistemleri**, (Editör: H. Durucasu), T.C. Anadolu Üniversitesi Yayıncılık No: 2690, Eskişehir.
- Kendall K.E., Kendall J. E., (2006). **Systems Analysis and Design**, 8th Edition, Boston, Pearson Education Inc.
- Sommerville I., (2011). **Software Engineering**, 9th Edition. Boston, Addison-Wesley.
- Valavich J. S., George J. F., Hoffer J. A., (2012). **Essentials of Systems Analysis and Design**, 5th Edition, Boston, Pearson Education Inc.
- Whitten J. L., Bentley L. D., (2007). **Systems and Design Methods**, 7th Edition. New York, Mc Graw-Hill Inc.



# 2

### Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Sistem geliştirme yaşam döngüsünün fazlarını tanımlayabilecek,
  - 🕒 Bilgi sistemlerini geliştirmede kullanılan modelleri sıralayabilecek,
  - 🕒 Alternatif sistem geliştirme modellerinin farklı tip projelerde kullanımını değerlendirebilecek,
  - 🕒 Sistem geliştirmede kullanılan çeşitli otomasyon araçlarını tanımlayabilecek bilgi ve becerilere sahip olabileceksiniz.

### Anahtar Kavramlar

- Sistem Geliştirme Yaşam Döngüsü
- Doğrusal Ardışık Model
- Artımlı Model
- Spiral Model
- Bileşen Tabanlı Geliştirme Modeli
- Çevik Model
- CASE

### İçindekiler

Sistem Analizi ve Tasarımı

Bilgi Sistemlerinin Geliştirilmesi

- GİRİŞ
- BİLGİ SİSTEMLERİ GELİŞTİRME
- BİLGİ SİSTEMİ GELİŞTİRME MODELLERİ
- BİLGİSAYAR DESTEKLİ SİSTEM GELİŞTİRME ARAÇLARI

# Bilgi Sistemlerinin Geliştirilmesi

## GİRİŞ

Önceki üitede açıklandığı gibi bilgi sistemleri yazılım, donanım, veri ve birden fazla kişinin taraf olduğu iletişim ve ağ teknolojileri içeren karmaşık yapılardır. Bu nedenle bilgi sistemlerinin geliştirilmesi için standart hâle getirilmiş yöntemleri içeren modeller (sistem geliştirme modelleri olarak adlandırılır) kullanılmaktadır. **Sistem geliştirme modelleri** çerçevesinde; bilgi sistemi projelerinin yönetiminin kontrolü sağlanmakta ve uygulanan yöntemler tanımlanmaktadır. Modeldeki veri ve süreç tanımları ile iş ürünleri (diyagramlar, belgeler, formlar, veri ve raporlar, vb.) üretilmekte, kilometre taşları belirlenmekte ve kalite yönetimi sağlanmaktadır.

Günümüzde işletmeler standartlaşmış sistem geliştirme modellerini, artan biçimde projelerine uyumlandıracak kullanmaktadır. Sistem geliştirme modellerinin kullanımı ile farklı projeler arasında aynı kaynakların paylaşılması sağlanmıştır. Ayrıca tutarlı belgeleme ile de, sistemlerin yaşam boyu bakım masrafları azaltılmaktadır. İlave olarak, kamu ya da büyük işletmeler tarafından talep edilen sistemlerin geliştirilmesinde kalite yönetim gereksinimleri de zorunluluk olarak istenmektedir. Sistem geliştirme modelinin tutarlı olarak kullanımını kalite yönetimini de destekleyecektir.

Bu üitede bilgi sistemlerinin analiz ve tasarımında kullanılan sistematik bir geliştirme süreç modeli inceleneciktir. Bu süreç modeli Sistem Geliştirme Yaşam Döngüsü (SGYD) olarak adlandırılmaktadır.

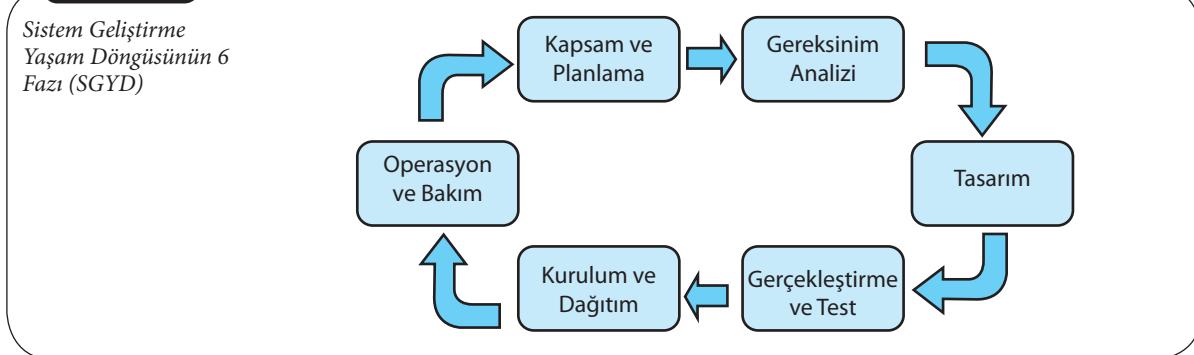
**Sistem geliştirme modeli;**  
faaliyetler, teknikler, uygulamalar,  
çıktılar ve sistem paydaşlarının  
kullandığı otomatik araçlar ile  
bilgi sistemlerinin geliştirilmesini  
ve sürekli iyileştirilmesini içerir.

## BİLGİ SİSTEMLERİ GELİŞTİRME

Sistem Geliştirme Yaşam Döngüsü, bilgi sistemi geliştirilirken uygulanan fazların kavramsal bir modelidir. Bu kitapta sistemlerin geliştirilmesinde kullanılan SGYD modeli 6 temel fazdan oluşmaktadır (Şekil 2.1). Bu fazlar Kapsam ve Planlama, Gereksinim Analizi, Tasarım, Gerçekleştirme ve Test, Kurulum ve Dağıtım, Operasyon ve Bakım biçiminde sıralanır. Farklı sistem geliştirme modellerinde farklı sayıda faz olabilir. Faz sayısı konusunda analistler farklı görüşte olsalar da (tanımlar 3 fazdan 20 faza kadar değişebilmektedir), bu fazlarda ifade edilen genel yaklaşımın kullanılmasının gerekliliği konusunda herkes ortak anlayıştadır. SGYD fazları sıralı olarak verilmiş olmakla beraber sıralı uygulanması gereklidir. Bu fazlar projeye bağlı olarak gerektiğiinde kullanılabilceğini göstermektedir. Örneğin SGYD' nin herhangi bir fazında projedeki gerekliliklere göre daha önceki fazlara geri dönülebilir. Benzer biçimde iyi çalışmayan bir ürün kullanıldan geri çekilerek tekrar geliştirilmek üzere başlangıç fazına geri döndürülebilir. SGYD' de bir fazdaki bazı faaliyetleri diğer fazdaki faaliyetlerle birlikte tamamlamak ta mümkündür. Büyük iş-

letmeler genel olarak kendileri için özel olarak geliştirdikleri SGYD modellerini kullanırlar. Ancak bu özelleştirilmiş modeller de Şekil 2.1'de verilmiş olan genel sistem geliştirme yaşam döngüsü fazlarını, tekniklerini ve araçlarını kullanır. Burada verilmiş olan SGYD fazları, bilgi sistemlerinin neredeyse tamamının yaşam döngüsünde kullanılabilir.

**Şekil 2.1**



### Kapsam ve Planlama

Sistem Geliştirme Yaşam Döngüsünün ilk fazında problemin kapsamı, fırsatları ve hedefleri tanımlanır. Projenin geri kalanının başarısı problemin doğru tanımlanmasına bağlıdır. Problemin kapsamında; projenin büyüklüğü, sınırları, projenin vizyonu, kısıt ve limitleri, projenin paydaşları, bütçesi ve zamanlaması oluşturulur. Bu aşamada ortaya konulan problemler bilgisayar destekli araçların kullanımı ile çözümlerek işletmenin rekabet etme kabiliyeti artırılır ve endüstri standartlarının yakalanması sağlanır.

Bu aşamada gerçekleştirilen önemli faaliyetlerden biri geliştirilecek bilgi sisteminin uygulanması ile işletmelerin kazanımlarının belirlenerek, hedeflerin tanımlanmasıdır. İlk fazın paydaşları sistem sahipleri, dâhilî sistem yöneticileri, proje yöneticileri ve sistem analistleridir. Genelde sistem kullanıcıları bu fazda yer almaz. Sistem sahipleri ve dâhilî sistem yöneticileri ile görüşmeler yapılır, bulgular özeti alınır, projenin kapsamı tahmin edilir ve sonuçlar raporlanır.

SGYD'nin ikinci fazından önce sistemin işletmeye olan ekonomik ve örgütsel etkisini saptamak üzere fizibilite çalışması yapılır ve rapor oluşturulur. Fizibilite raporu problem tanımını ve hedefleri içerir. Bu rapor sonucuna göre yönetim, projenin devamı için bir karar vermek zorundadır. Eğer bütçe yeterli değilse veya problemlerin çözümünde bilgi sisteminin kullanımına gerek olmayan farklı çözümler önerilirse proje bu aşamada sonlandırılabilir.

Sistem kapsam ve planlanmanın diğer bir çıktısı da proje takımı tarafından belirli bir plandır. Bu proje planı standart SGYD'ni özelleştirir ve gerçekleştirilmesi için gereken zaman, para ve kaynakları belirler. İlk fazda oluşturulan en önemli çıktılardan biri iş beyanıdır. İş beyanı, bilgi sisteminin geliştirilmesini ifade eden bir sözleşme ya da anlaşmadır. Bu beyanda problemin tanımı, kapsamı ve projedeki tüm kaynakların genel zamanlama ve bütçesi yer alır.

### Gereksinim Analizi

İkinci fazda sistem analisti sistem ihtiyaçlarını analiz eder. Analist, sosyal araç ve tekniklerin yardımıyla gereksinimleri tanımlar. Bu gereksinimler işlevsel, işlevsel olmayan ve ortam gereksinimleri olarak üçe ayrılabilir. İşlevsel gereksinimleri belirlemek için;

- Yeni sistem, kullanıcılarına hangi kabiliyetleri sunar?
- Hangi veri toplanmalı ve depolanmalıdır?
- Hangi performans düzeyi beklenmektedir?

gibi sorulardan yola çıkarak, gereksinimler tanımlanır ve iş gereksinimleri önceliklendirilir. Ayrıca analist bu aşamada insan-bilsayar etkileşimi ve sistemin çalışacağı ortamı ilgilendiren sorular ve cevapları da kullanır.

- Sistemi kanunlara uygun ve güvenilir yapmak için ne gerekir?
- Yeni sistem, kullanıcı dostu olması için nasıl tasarlanmalıdır?
- Sistem, kullanıcıların iş görevlerini daha verimli yapacak özellikleri destekliyor mu?
- Sistemin çalışma ortamına göre desteklemesi gereken özellikler var mıdır?

Bu soruların cevabı ise işlevsel olmayan ve ortam gereksinimlerini oluşturur.

Sistem analisti, bu fazda bilgi kullanıcılarının işini yapması için nelerin gerekli olduğunu anlamaya çalışmaktadır. Mevcut sistem fonksiyonlarının ayrıntılarını öğrenmek için kim (sistemi kimler kullanıyor), ne (işletmenin faaliyetleri), nerede (ürünün çalışacağı yer), ne zaman (zamanlama) ve nasıl (mevcut iş süreçleri) sorularına cevap arar.

Bu aşamanın paydaşları proje yöneticileri, sistem analistleri ve sistem sahipleridir. Paydaşlara ayrıca sistem kullanıcıları da dahil edilebilir. Sistem tasarımcıları ve geliştirme mecliler bu aşamada yer almaz.

Bu fazda analist, işletmenin mevcut iş süreçlerini ve sipariş alma, ödeme sistemleri vb. bilgi sistemlerini ayrıntılı olarak inceler. Bu analiz birçok alt adımdan oluşur. Birinci adımda analist, kullanıcıların önerilen sistemden isteklerini belirler. Bu adım elle veya bilsayar desteğiyle çalışan, değiştirilen veya iyileştirilen mevcut sistemin dikkatlice incelenmesini içerir. Sistem geliştirme hedefleri ile iş gereksinimleri kontrol edilir. Her bir gereksinim, bir veya daha fazla sistem geliştirme hedefine katkıda bulunmalıdır. İzleyen adımda, gereksinimler üzerinde çalışılır ve aralarındaki ilişkilere göre yapılandırılarak tekrarlar kaldırılır. Yapılandırmanın bir parçası olarak, gereksinimleri karşılayan alternatif tasarımlar üretilir. Daha sonra bu alternatifler maliyet, işçilik ve geliştirme sürecinin gerektireceği teknik düzeyler açısından kıyaslanarak, hangisinin gereksinimleri en iyi karşıladığı belirlenir. Gereksinim Analizi fazının çıktısı alternatif bir çözüm önerisidir. Önerinin yönetim tarafından kabul edilmesi hâlinde önerilen sistemin çalıştırılması veya donanım ve sistem yazılımının oluşturulması için planlar oluşturulur. Ünite 4'te kullanıcı gereksinimlerini belirleme ve belgeleme için kullanılan sistem analizi araç ve teknikleri tanıtılacaktır.

Bu fazda belirlenen gereksinimler önceliklendirilmelidir. Böylece proje zamanlamasında sıkışıklık oluşursa bu önceliklendirme kullanılabilir. Ayrıca ürünün **iteratif geliştirilmesi** hâlinde her bir iterasyonda tasarım ve geliştirmede kullanılacak gereksinimler bu önceliklere göre belirlenebilecektir.

Bu faz hiçbir şekilde atlanılmamalı veya kısaltılmamalıdır. Kullanıcı gereksinimleri ni tam olarak anlamadan teknik çözümle uğraşan sistem tasarımcısı ve geliştiricilerinin oluşturduğu yeni sistem, kullanıcıların isteklerini karşılamayacak ve memnuniyetsizlik yaratacaktır.

**Iteratif geliştirmede**, bilgi sisteminin birden fazla tekrar (iterasyon) geliştirme ile daha iyi hâle getirilmesi hedeflenir.

## Tasarım

SGYD 'nin üçüncü fazı sistem tasarımıdır. Sistem tasarımında analist, alternatif çözüm önerilerini mantıksal ve fizikal sistem özelliklerine dönüştürür. Sistemin girdileri, çıktıları, arayüzleri, raporları, veritabanları ve bilsayar süreçlerinin tamamı bu fazda tasarlanır. Bir önceki fazda belirlenen iş gereksinimleri genellikle sözel ifadelerden oluşur. Sistem analistleri, kelimeleri **sistem modelleri** olarak adlandırılan diyagramlara dönüştürür ve gereksinimlerin tutarlığını ve bütünlüğünü test eder. Tasarım fazı Mantıksal tasarım ve Fiziksel tasarım olmak üzere iki aşamadan oluşur.

**Sistem Modeli** sistemin soyut modelinin grafikler ile gösterilmesidir. Gereksinim Analizi fazında bilgi sisteminin gereksinimlerini belirlemek, tasarım fazında sistem geliştiricilerine sistemi tanımlamak, bilgi sisteminin kurulum ve uygulamaya alma fazında da belgeleri oluşturmak için sistemi ifade eden modeller kullanılır.

Mantıksal tasarım aşaması iş gereksinimlerini sistem modellerine dönüştürür. Daha önce belirlenen veri ve bilgi gereksinimlerinden mantıksal veri modelleri oluşturulur. İş süreç gereksinimlerinden mantıksal süreç modelleri, iş ve sistem arayüz gereksinimlerinden ise mantıksal arayüz modelleri oluşturulur. Bu modellerde genel sistemin kavramsal, etkinlik ve kullanım durum diyagramları kullanılır. Mantıksal tasarımda çözüm önerisi kullanılacak teknolojilerden bağımsız oluşturulur. Bu aşamada çalışan paydaşlar ve görevleri şu şekildedir: Proje yöneticileri modellerin standartları karşıladığı ve proje sürecini kötü yönde etkilemediğini kontrol eder. Sistem analistleri modelleri oluşturur. Sistem kullanıcıları ise modelleri onaylar. Farklı yöntemler kullanılarak, farklı miktarda ve derecede sistem modeli ve mantıksal tasarım oluşturabilir. Mantıksal tasarım belirli bir donanım ve yazılım platformuna bağlı değildir. Teorik olarak tasarlanan sistem herhangi bir donanım ve yazılım platformunda gerçekleşebilir.

Fiziksel tasarıma geçmeden önce, verilen iş gereksinimleri ve mantıksal tasarım modelleri aşağıda anlatılan fizibilite sonuçlarına göre değerlendirilir ve bir mantıksal tasarıma karar verilir. İzleyen kesimde, çözüm önerilerinin hangi açılardan değerlendirildiğinin açıklanacağı fizibilite türleri yer almaktadır.

**Teknik fizibilite:** Önerilen çözüm, teknik olarak uygulanabilir olmalıdır. Tasarım ve geliştirme ekipleri önerilen çözüm için teknik yeterlige sahip olmalıdır.

**Operasyonel fizibilite:** Çözümün kullanıcı gereksinimlerini karşılaması, kullanıcıların çalışma ortamını nasıl değiştirdiği, kullanıcıları memnun edip etmeyeceği incelenmelidir.

**Ekonominik fizibilite:** Çözümün maliyeti proje bütçe hedefleri ile uyumlu olmalıdır.

**Zamanlama fizibilitesi:** Çözümün tasarım ve geliştirilmesi, projenin genel zaman planıyla uyumlu bir zaman aralığında yapılmalıdır.

**Risk fizibilitesi:** Önerilen teknik ve süreçler kullanıldığında çözümün başarıyla geliştirilme ihtimali yüksek olmalıdır.

Mantıksal tasarıma karar verildikten sonra yazılı ve/veya sözlü sistem önerisi oluşturulur. Bu öneri gereklirse genel sistem mimarisini de içerebilir. Sistemin mantıksal tasarım önerisinin onayından sonra **Fiziksel tasarıma** geçilir. Fiziksel tasarımda, Mantıksal tasarımlar fiziksel veya teknik özelliklere dönüştürülür. Fiziksel tasarımın amacı iş gereksinimlerinin sistemin oluşturulmasına yön verecek şekilde fiziksel tasarım özelliklerine dönüştürülmesidir. Fiziksel tasarım, Mantıksal tasarımda önerilmiş olan mimari kısıtlara uygun yapılır. Sistemin geliştirileceği ve çalıştırılacağı; fiziksel iş süreçleri, yazılım tasarım özellikleri, fiziksel kullanıcı ve sistem arayüz özellikleri bu aşamada oluşturulur. Ayrıca, varsa fiziksel veritabanı tasarım özellikleri, dosya yapıları, programlama dilleri, donanım platformu, işletim sistemi ve ağ ortamı mimarisi belirlenir. Tasarım aşamasından sonra bir proje çok önemli bütçe sorunları ya da zamanlama gecikmesi olmadığı sürece nadiren iptal edilir. Modern sistem geliştirme modellerinde (Çevik model, Scrum vb.) tasarım, geliştirme ve test fazlarının birleştirilmesi yoluna gidilmektedir.

Tasarım fazındaki faaliyetler, farklı uygulamalar için geliştirilen bilgi sistemlerine göre değişiklik gösterir. Bilgi sistemlerinin tasarıımı aşamasında aşağıda verilen dört temel faaliyet gerçekleştirilir:

**Yapisal tasarım:** Sistem, sistemin temel (alt parçalar ya da modüller) parçaları, parçalar arası ilişkiler ve dağılımını ifade eden bütün yapısı tanımlanır.

*Arayüz tasarımı:* Sistemin temel parçaları arasındaki arayüzler tanımlanır. Arayüz özellikleri kesin ve net olarak tanımlanmalıdır. Bu sayede bir alt sistem, nasıl geliştirildiklerini bilmeden diğer alt sistemlerle birlikte kullanılabilir. Arayüz özelliklerine karar verildikten sonra tüm bileşenler paralel olarak tasarılanabilir ve geliştirilebilir.

*Bileşen tasarımları:* Her bir sistem bileşeninin tanımlanmış arayüz özelliklerine göre tasarlanması kapsar. Kullanılan geliştirme sürecine göre bu aşamada uygulamadan beklenilen fonksiyonları ifade etmek yeterli olabilir. Uygulama özelinde tasarım, sistem ge-

**Fiziksel Tasarım**, kullanıcı gereksinimlerinin teknik olarak gerçekleştirilemesini gösteren sistem modelidir. Teknik tasarım veya uygulama modeli olarak da adlandırılır.

lijitiricilerine bırakılır. Bileşen tabanlı sistem geliştirme modeli yaklaşımında ise tekrar kullanılabilir bileşenlerde yapılacak değişiklikler ve yeni tasarlanacak bileşenler ayrıntılı olarak tanımlanır.

*Veritabanı tasarımları:* Sistem veri yapısı tasarılanır ve bunların veritabanında karşılık gelen yapıları gösterilir.

Bu fazın paydaşları sistem tasarımcıları, sistem analistleri ve iş akışını paylaşmak üzere sistem kullanıcılarıdır. Tasarım sürecinin çıktıları, sistem geliştiricilere ve programcılara verilebilecek, sistem özelliklerini anlatan diyagram ve/veya yazılı raporlardan oluşur.

## **Gerçekleştirme ve Test**

Sistem gerçekleştirme ve test fazında, özellikleri ortaya konulmuş olan sistem geliştirilir ve testleri tamamlanarak çalışır hale getirilir. Programlama sırasında, programcılar sistemi oluşturan yazılımı ve eski sistem ile yeni sistem arasındaki arayüzleri geliştirir. Gerçekleştirmenin en önemli yönlerinden biri olan testler sırasında da programcılar ve analistler program parçalarında ve tüm sistemde oluşabilecek hataları bulmak ve düzeltmek üzere testler uygular. Sistem, kullanıcılarla teslim edilmeden önce hataların yakalanması halinde, hataların çözümü çok daha az maliyetli olacaktır. Testler başlangıçta örnek veriler ile yapılrken teslimden önce testler gerçek veriler ile tamamlanır.

İlave olarak, bu fazda analistler kullanıcılar ile çalışarak sistemin kullanımını ve hata olması durumunda yapılacakları açıklayan belgeler oluşturulur. Bu belgeler kullanım kitapçıkları, çevrim içi yardım, sıkça sorulan soruları içeren Internet sayfaları, yeni sistem ile gönderilen beni oku ve kurulum açıklamalarını içerir. Belge oluşturma, sistemin tüm yaşam döngüsü boyunca devam eder ve eğitimler (ve öğrenme) proje ilk oluşturulduğunda başlar. Sistem analistleri, sistem kullanıcıları, proje yöneticileri ile birlikte programcılar ve test yapanlar tasarım, kodlama ve hataları giderme üzerinde çalışıklarından bu fazın anahtar paydaşlarıdır.

Bu fazın sonunda proje ekibi, kurulum ve dağıtım için çalışan fonksiyonel bir sistem ortaya koyar. Veri tabanları, ticari yazılım paketleri ve/veya özel geliştirilmiş yazılımlar ve kullanıcı-sistem arayüzleri bu sistemin parçalarını oluşturur. Testler tamamlandıktan sonra sistem kurulum ve dağıtıma hazırır.

## **Kurulum ve Dağıtım**

Gerçekleştirme ve test fazında oluşturulan fonksiyonel sistem, kurulum ve dağıtım fazının girdisini oluşturur. Sistem geliştiricileri, uygulama yazılımını, geliştirme ortamından üretim ortamına geçirmek üzere mevcut ya da yeni donanım üzerine kurarlar. Sistem analistleri sistem kullanıcılarını eğitir, birçok kullanıcı el kitaplarını ve üretim kontrol el kitaplarını oluşturur, eski veritabanının yeni veri tabanına aktarımını ve en son sistem testlerini yapar. Herhangi bir problem olması hâlinde hatayı düzeltmek üzere önceki fazlardan tekrar çalışma başlatılması gerekebilir. İşveren yeni sistemler için bazı eğitimler düzenleyebilir, ancak eğitimin gözetimi sistem analistinin sorumluluğundadır.

Kurulum ve uygulamaya alma işlemleri çok fazla analiz ve çalışma gerektirdiğinden bu faaliyetlerin planlamasına proje kapsam ve plan aşamasından başlanmalıdır. İlave olarak, analist eski sistemden yeni sisteme sorunsuz nasıl geçilebileceğini de planlamalıdır. Bu plan, belirlenen bir tarihte eski sistemin tamamen kapatılarak yeni sisteme geçilmesi veya eski ve yeni sistemin paralel olarak (yeni sistemin kesin onayına kadar) beraber çalıştırılması biçiminde olabilir. Analistlerin, yöneticilerin ve programciların tüm gayretlerine rağmen sistemin devreye alınması ve kurulumu kolay bir işlem olmaz. Pek çok iyi tasarlanmış sistem, uygulamaya alma işlemlerinin iyi yönetilememesinden dolayı başarısız olabilir. Bu fazın çıktısı işverenin ve kullanıcıların günlük faaliyetlerinin bir parçası olan çalışan sistemdir.

## Operasyon ve Bakım

**Sistem desteği**, sistem kullanıcılarına sistem çalıştığı sürece verilen teknik desteği ve herhangi bir hata durumunda veya yeni gereksinimler ortaya çıktığında verilen bakım hizmetlerini kapsar.

**Veritabanının son kurtarma noktası**, herhangi bir sebepten dolayı veritabanında bir bozukluk veya tutarsızlık meydana geldiğinde oluşan hatalardan kurtarmak için veritabanının yedeklerden geri yüklenmesini tanımlar.

SGYD'nin son fazı Operasyon ve Bakımdır. Sistemin kurulup çalışmaya başladığı bu evre en uzun yaşam döngüsü fazıdır. Sistem işletmelerde çalışmaya başladıkten sonra kullanıcılar bazen işleyiş ile ilgili problemler olduğunu fark eder, yaşam döngüsünün erken fazlarında görülmeyen problemler ortaya çıkar ya da iyileştirme yapmayı talep eder. Bu nedenle sistemin kullanıldığı süre içerisinde, programcılar sistem desteği sağlamak üzere kullanıcıların talep ettiği, iş süreçlerindeki değişimler doğrultusunda sistemi günceller.

**Sistem desteği** aşağıdaki faaliyetlerden oluşur:

**Kullanıcıya yardımcı olma**: Yeni kullanıcı eklenmesi vb. durumlarda son kullanıcıya ilave destek sağlamak gerekebilir.

**Yazılım hatalarını Onarma**: Sistem geliştirilmesi ve testleri yapılmış olsa da bazı hatalar kalmış olabilir. Bu hatalar sistem desteği ile giderilmelidir.

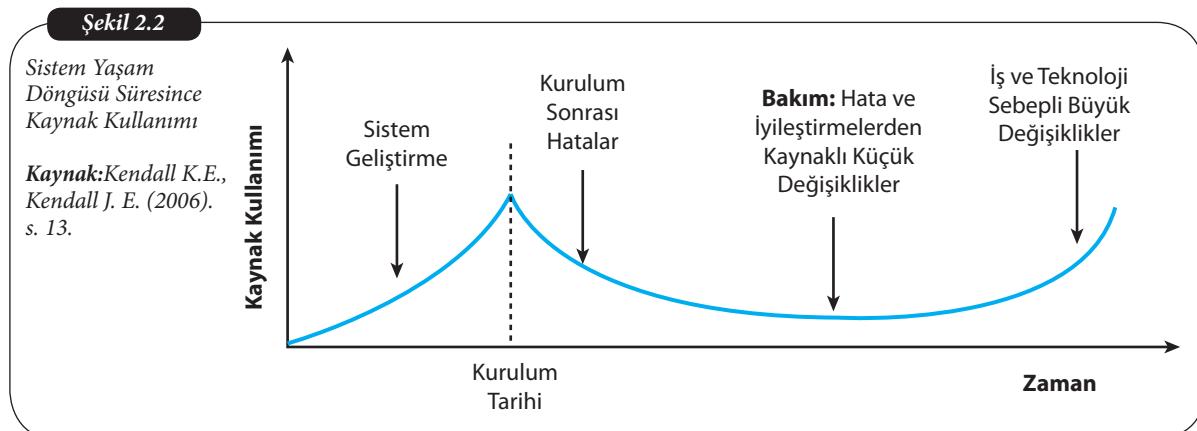
**Sistemi kurtarma**: Zaman zaman sistem hatası programın olmadık şekilde kapanması na ve hatta veri kaybına sebep olabilir. Bu durumda sistemin yeniden başlatılması, sistem dosya veya **veritabanının son kurtarma noktasına** geri getirilmesi gerekebilir.

**Sistemi yeni gereksinimlere göre güncelleme**: Değişiklik istekleri; yeni ortaya çıkan iş problemlerini, teknik problemleri ve teknoloji gereksinimlerini içerebilir. Yeni değişiklik istekleri için tasarım, geliştirme ve testler gerçekleştirilerek çalışan bilgi sisteminde güncelleme yapılır.

Bu tip değişiklikler sistemi kullanaklı tutmak için gereklidir. Sistem üzerinde bu şekilde değişiklikler yapmak için gereken zaman ve çaba büyük ölçüde yaşam döngüsünün önceki fazlarında uygulanan yöntemlere bağlıdır.

Şekil 2.2'de bilgi sistemi geliştirme ve bakımında kullanılan kaynakların (zaman ve para) sistem yaşam döngüsü süresince değişimi verilmektedir. Eğrinin altındaki alan toplam maliyeti gösterir. Zaman içindeki değişiklik gereksinimleri doğrultusunda, güncellemeler yapılarak sistemin operasyonda kalması sağlanacaktır (Bakım). Ancak, sistem üzerinde yapılan güncellemeler arttıkça bilgi sisteminin istediği ölçüde performans sergileyemediği, sistemi çalışır durumda tutmak için gereken maliyetlerin çok arttığı, işletmenin ihtiyaç duyduğu bilgi sistemi ihtiyaçlarının büyük ölçüde farklılaşlığı zamanlar gelecektir (Şekil 2.2'de İş ve Teknoloji Sebepli Büyük Değişiklikler Bölgesi). Bu tip problemler artık bilgi sisteminin değiştirilme zamanının geldiğini gösterir. Operasyondaki bilgi sisteminin yaşam döngüsü sonlanmış ve yeni bir sistem için yeni bir yaşam döngüsüne başlamadan zamanı gelmiştir.

**Şekil 2.2**



Sistem bakımı ve bakım belgelemesi bu fazda başlar ve bilgi sisteminin yaşam döngüsü boyunca devam eder. Bakım süresince, programcı güncellemeler üzerinde rutin olarak çalışmaya devam ederken, işletmeler bu aşamada bakım ücretini öder. Bazı program güncelleme ve bakımlar bilgi sistemi üreticisinin internet sitesinden otomatik olarak yapılabilir.

## BİLGİ SİSTEMİ GELİŞTİRME MODELLERİ

Sistem analizi ve tasarım süreçlerini farklı yaklaşımlar ile gerçekleştiren birçok model vardır. Bu modeller ile farklı uygulama alanlarındaki bilgi sistemlerinin uygun sistem analizi ve gerçekleştirme araçları ile geliştirilmesi desteklenmiş olur.

Sistem geliştirme yazılım modellerinden Doğrusal Ardisık Model, Prototip Model, Artımlı Model, Spiral Model, Bileşen Tabanlı Geliştirme Modeli ve Çevik Model izleyen kesimde açıklanacaktır.

### Doğrusal Ardisık Model

Sistem analizi ve tasarımında kullanılan Doğrusal ardisık model yaklaşımında fazlar sadece bir defa olmak üzere sıra ile işletilir. Doğrusal ardisık model aynı zamanda Şelale Modeli (WaterFall Model) olarak da isimlendirilir. Şekil 2.3'te gösterildiği gibi Doğrusal ardisık model bu ünitelerin ilk kesiminde açıklanmış olan genel SGYD sürecinden oluşmuştur. Modelin ana aşamaları bilgi sisteminin doğrudan temel geliştirme faaliyetlerini göstermektedir.

- i. *Kapsam ve Planlama:* Sistemin sağlayacağı hizmetler, kısıtları ve amaçları sistem kullanıcıları ile belirlenir.
- ii. *Gereksinim Analizi:* Sistem geliştirme hedefleri ile iş gereksinimleri kontrol edilir. Her bir gereksinim, bir veya daha fazla sistem geliştirme hedefine katkıda bulunan şekilde ayrıntılı olarak tanımlanır ve tüm sistemin özellikleri sistem kullanıcıları ile gözden geçirilir.
- iii. *Tasarım:* Sistem tasarımı, gereksinimlerin tamamını karşılayacak donanım ve yazılım sistemlerinden oluşan sistem mimarisini oluşturur. Tasarım, geliştirilen bilgi sistemi yazılımının dört ayrı özelliğine odaklanır: veri yapısı, yazılım mimarisi, arayüz göstergeleri ve algoritma yapılandırması. Gereksinimlerde olduğu gibi tasarım da belgelenir ve sistem yapılandırmasının bir parçası haline gelir.
- iv. *Geliştirme ve Test:* Bu aşamada yazılım tasarımı, programlar veya program parçaları olarak gerçekleştirilmektedir. Program testleri yazılım geliştirme ile başlar. Testler, yazılımın tüm gereksinimleri karşıladığı göstermek ve olası hataları bulmak için uygulanır.
- v. *Kurulum ve Dağıtım:* Sistem mimarisi çerçevesinde oluşturulmuş olan program parçaları bir araya getirilerek tüm sistem gereksinimlerinin karşılandığı test edilir. Testler sonrasında geliştirilen sistem, müşterinin kullanacağı şekilde kurulur ve uygulanmaya alınır.
- vi. *Operasyon ve Bakım:* Normalde bu SGYD içinde en uzun yaşam döngüsüne sahip fazdır. Sistem kurularak kullanımına başlanır. Bakım aşaması daha önceki keşfedilmemiş olan hataların düzeltilmesi, sistemin iyileştirilmesi ve sistemde oluşturulmak istenen yeni özelliklerin geliştirilmesini içerir. Yazılım bakım aşamasındaki değişiklik gereksinimleri için mevcut programda uygulanmış olan önceki fazlar tekrar uygulanmaktadır.

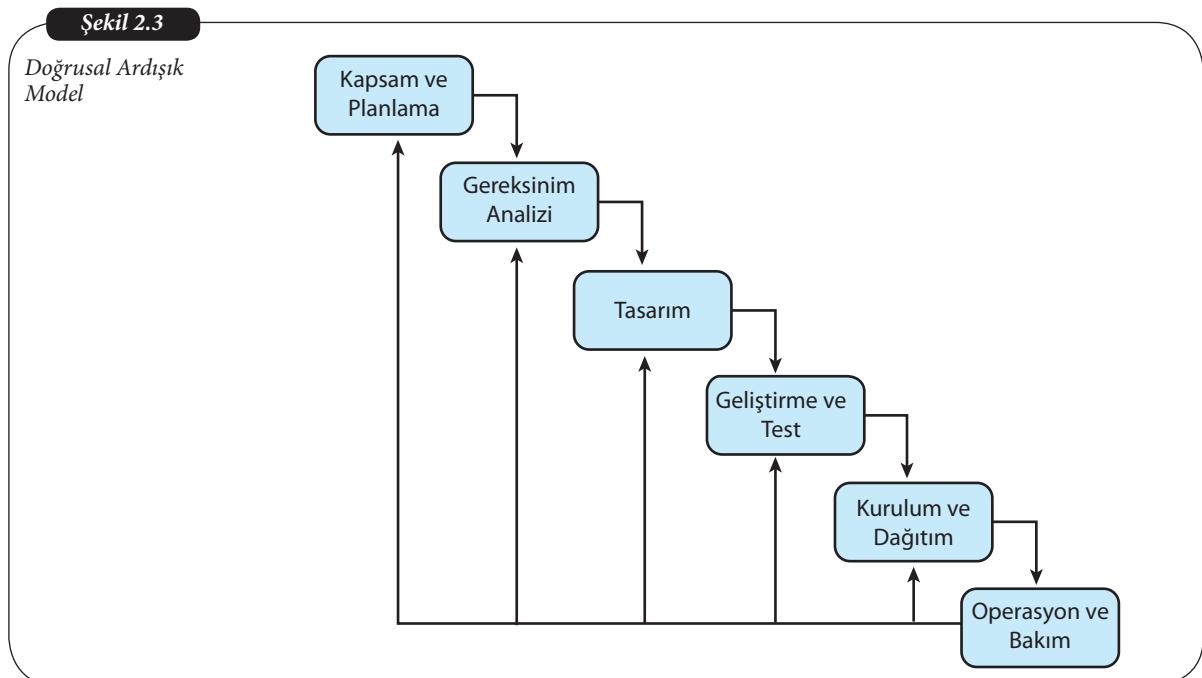
Doğrusal ardisık model, her bir fazında gerçekleştirilen faaliyetler ve belgeleme ile yöneticilerin proje planlarına göre ilerlemeyi izlemelerine imkân sağlar. Bununla birlikte projenin esnek olmayan şekilde alt fazlara bölmüş olması en önemli problemini oluşturur. Bu nedenle Doğrusal ardisık model en eski ve en yaygın kullanılan model olmakla

birlikte, verimliliği en aktif destekleyenler tarafından bile sorgulanmaktadır. Doğrusal ardışık modelin uygulanmasında ortaya çıkan bazı problemler aşağıda sıralanmıştır:

- i. Gerçek projelerde modelde kullanıldığı gibi ardışık bir sıralama nadiren uygulanır. Değişiklikler her aşamada ortaya çıkabilir. Proje takımının değişiklikleri yönetmesi zordur ve yüksek oranda kaynak (zaman, maliyet) tüketimine sebep olur.
- ii. Bazı sistemlerde müşteri tüm gereksinimleri en baştan ayrıntılı olarak belirtemez. Oysa Doğrusal ardışık model tüm gereksinimlerin en başta tüm ayrıntılıyla tanımlanmasını gerektirir ve başlangıçta olabilecek belirsizliklerin yönetimi zordur.
- iii. Müşteri sistemin gerçekleştirilerek uygulamaya alınmasında sabırı olmak zorundadır. Sistemin çalışan sürümü projenin gerçekleştirme sürecinin sonlarına doğru hazır olur. Çalışan programın gözden geçirilmesinde ortaya çıkabilecek ana sorun projenin başarısızlığına dahil sebep olabilir.
- iv. Yaşam döngüsünde kullanılan ardışık yaklaşım, bazı proje elemanlarının diğer fazlardan gelecek olan çıktıları beklemesinden dolayı bloklanması ve üretim için harcayacağı zamanları israf etmesine sebep olabilmektedir.

Doğrusal ardışık model gereksinimlerin tam olarak bilindiği/anlaşıldığı ve sistem geliştirme sırasında radikal değişikliklerin gerekmeyeceği durumlarda kullanılmalıdır.

**Şekil 2.3**



SIRA SİZDE



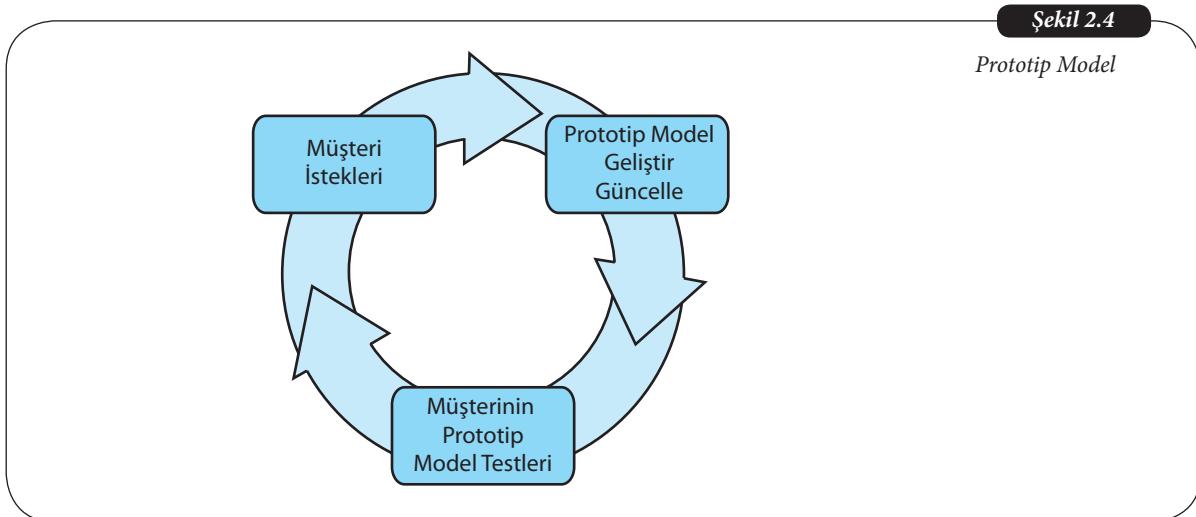
1

Şelale modelinde iterasyona niçin izin verilmez?

### Prototip Model

Bazı projelerde müşteri, sistemin genel hedeflerini tanımlar fakat ayrıntılı girdi, iş süreçleri veya çıktı gereksinimlerini belirtmez. Bazı durumlarda da geliştirme ekibi kullanılacak işletim sisteminden, yazılım verimliliğinden ve insan makine arayüzlerinin nasıl olması gerekiğinden emin olmayıabilen. Bu ve benzeri durumlarda gereksinim oluşturmada prototip yaklaşımı en uygun yöntem olacaktır. Prototip modelde ilk adım müşteri istekleri doğrultusunda gereksinimlerin belirlenmesidir. Geliştirme ekibi ve müşteri birlikte yazılı-

nın tüm hedeflerini tanımlar, mümkün olabildiği kadarıyla gereksinimleri belirler ve açık kalan alanları kayıt altına alır. Devamında sistemin kullanıcılar tarafından görülür kısımları için hızlı bir tasarım yapılır. Hızlı tasarımın devamında prototip geliştirilir. Prototip, kullanıcılar tarafından değerlendirildikten sonra gereksinimler tekrar gözden geçirilir ve süreç tekrar başlar. Prototip modelde (Şekil 2.4) iterasyon (döngü), ürünün gelişimi müşterinin gereksinimlerini oluşturuncaya ve geliştircilerin ne yapılması gerektiğini anlamasına kadar devam eder.



Idealde prototip, sistemin tüm gereksinimlerinin belirlenmesine yardımcı olur. Prototip modelde, başlangıçta müşteri ile geliştirilen prototipin gereksinimleri tanımlamak için bir mekanizma olduğu konusunda anlaşılmalıdır. Gereksinimler açık olarak tanımlandıktan sonra prototip atılmalı ve gerçek yazılım, kalite, bakım ve yapılabılırlik özellikleri ile yeni baştan ele alınmalıdır.

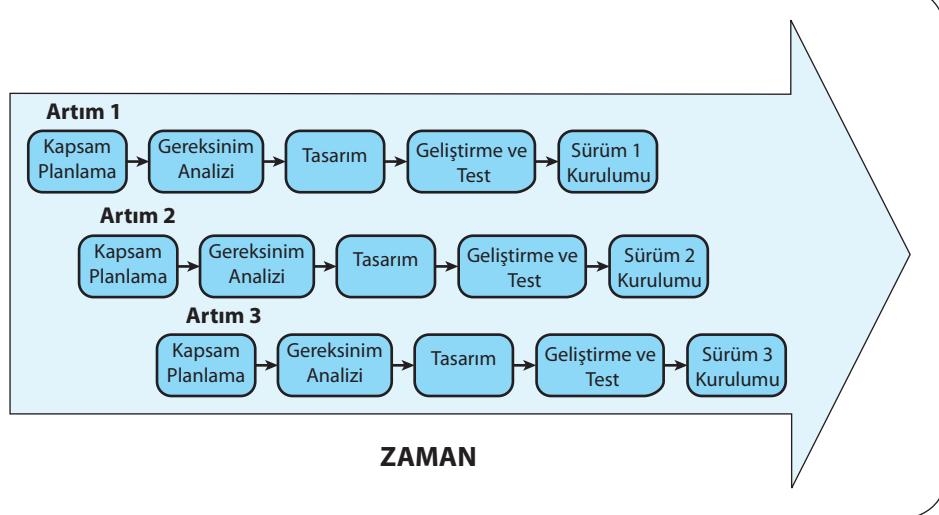
### Artımlı Model

Artımlı model (Incremental model), ilk uygulamanın oluşturularak kullanıcı görüşüne başvurulması ve istenilen son sistem geliştirilinceye kadar birçok sürüm oluşturulması fikrine dayanır. Tasarım ve Özellikleri oluşturma, Geliştirme ve Testler ayrı ayrı yapılmak yerine her bir ara sürüm için ayrı ayrı ve paralel yapılır.

Artımlı model, Doğrusal ardışık model ile Prototip modeldeki iteratif yaklaşımın birleşimidir. Iteratif yapısıyla e-ticaret ve masaüstü sistemlerin geliştirilmesinde Doğrusal ardışık modelden daha iyi bir yaklaşımdır. Artımlı model geliştirilen sistemi birçok alt sürümden oluşan şekilde Doğrusal ardışık modeldeki yaklaşımı kullanarak zaman içinde aşamalı olarak gerçekleştirir. Şekil 2.5'te gösterildiği gibi her bir alt sürüm bir artımı oluşturur. Her bir yeni sürüm müşterinin ihtiyacı olan bir özelliği sistem içine ekler. Genelde sistemin ilk iterasyonları daha önemli ve acil özellikleri içerir. Yazılımın artan şekilde geliştirilmesi tasarım ve geliştirme aşamalarında değişikliklerin daha kolay ve az maliyetli yapılmasını sağlar. Bu iteratif yaklaşım, bir sürümde sisteme eklenilen özellik istenileni sağlanamırsa izleyen sürümde değişiklik yapma ve ihtiyaçları karşılayan bir sürüm oluşturulmasına imkân verir. Örneğin kelime işlemci yazılımı geliştirilirken ilk sürüm; dosya yönetimi, yazı düzenleme ve belge üretme fonksyonlarını içerirken, ikinci sürümde; hece sayımı, gramer kontrolü, üçüncü sürümde; gelişmiş sayfa yapısı fonksyonları vb. gerçekleştirilmiş olur. Artımlı modelde üretilen ilk sürüm temel özellikleri taşıyan ana üründür.

**Sekil 2.5**

Artımlı Model



Doğrusal ardışık model ile kıyaslandığında Artımlı model bazı önemli kazanımlar sağlar. Bunlar;

- Müşteri gereksinimlerinde değişikliğin maliyeti (analiz, tasarım, testler ve belgeleme) daha düşük olur.
- Geliştirme aşamasında müşteri geri dönüşü daha kolay alınmış olur. Müşteri, tasarım belgeleri yerine yazılımın çalışan bir sürümü üzerinden daha kolay yorum yapar.
- Yazılımın tamamını olmasa da önemli özelliklerini taşıyan ilk sürümünü müşterinin kullanımına sunmak daha hızlı olacaktır. Böylece müşterinin yazılımı önceden kullanmaya başlaması iş süreçlerinde önemli kazanımlar elde etmesini sağlayabilir.
- Sistemin son tarihinde teslim için yeterli sayıda geliştirme personeli yok ise, ilk sürümler (temel ürün) daha küçük bir ekip ile istenilen zamanda geliştirilir. Bir kez temel ürün geliştirildikten sonra gerekirse ekip büyütülerek artım ile izleyen sürümler gerçekleştirilir.

Artımlı model ile geliştirme, Doğrusal ardışık model yaklaşımıyla, Çevik model yaklaşımı (izleyen kesimde anlatılacak) ya da her ikisinin karışımı ile gerçekleştirilebilir. Doğrusal ardışık model yaklaşımıyla sistemin her bir artımı önceden belirlenirken, Çevik model yaklaşımında ilerleyen sürümlerin geliştirilmesi süreç ve müşteri önceliklerine göre gerçekleştirilir. Bununla birlikte Artımlı modelin kullanımı da bazı problemler oluşturabilir:

- Süreç izlenemez. Yöneticiler ilerlemeyi izlemek için ara çıktılar ihtiyaç duyar. Sistemler çok hızlı geliştirildiğinden, erken ve ara sistem sürümlerini gösteren belgeleri uygun maliyetlerle üretmek mümkün değildir.
- Sistem, sürümler ilerledikçe yapısal olarak geriler. Sistemde sürekli değişikliklerin uygulanması yapısal bozulmaya yol açar. Bu nedenle sistemin yapısal iyileştirilmesi için zaman ve para harcanması gereklidir. Aksi hâlde değişiklik uygulamasının sebep olduğu yapısal bozukluklar, yeni değişikliklerin uygulanmasının zorluğunu ve maliyetini artırır.
- Sistemlerin çoğu, alt parçalarında bazı ortak imkân ve kaynakları kullanır. Büttün gereksinimlerin başlangıçta açık olarak bilinmediği durumlarda, bu ortak imkânların öngörülmesi zor olacaktır.

- iv. Mevcut sistemin yerine yapılan sistemlerin geliştirilmesinde iteratif yaklaşımı kullanmak zor olabilir. Sistem kullanıcıları, mevcut/eski sistemin tüm özelliklerini olmadan, kısıtlanmış özellikleriyle yeni sistemde çalışmayı istemez. Bu yüzden de müşteri geri beslemesini almak zorlaşır.
- v. Artımlı geliştirme sürecinde gereksinimler geliştirilen sistem ile birlikte oluşturulur. Bu ise, sistemin tüm özelliklerini tanımlayan bir şartnameyle yapılan satın alma süreçlerinde kullanılamaz. Çünkü Artımlı modelde son sürümé kadar tüm gereksinimler bilinmemektedir. Bu nedenle büyük devlet ve kurum ihalelerinde bu yöntemi uygulamak zor olacaktır.

**Artımlı modeli kullanmanın avantajları nelerdir?**



SIRA SİZDE

## Spiral Model

Spiral model, Prototip modelin iteratif yapısının, Doğrusal ardışık modelin sistematik yaklaşımıyla kontrol edilmesini sağlayan evrimsel bir modeldir. Sistem yazılımının artan sürümlerinin hızlı geliştirilmesine olanak tanır. Bu modelde de yazılım artan sürümler hâlinde geliştirilir. İlk iterasyonlardaki artan sürümler kâğıt üzerindeki gereksinimleri veya prototipi oluştururken ilerleyen iterasyonlarda sistemi tam olarak gerçekleyen artan sürümler geliştirilmiş olur. Spiral modelde sistem geliştirme tipik olarak 3 ile 6 arasında değişen sayıda bölümden oluşan çerçeve faaliyetlerine bölünmüştür. Şekil 2.6'da 6 bölümden oluşan Spiral model görülmektedir. Bu bölümler;

**Müşteri iletişim:** Geliştirici ve müşteri arasında etkin bir iletişim kurulması görevlerini,

**Planlama:** Kaynakların, zaman çizelgelerinin ve proje ile ilgili diğer bilgilerin tanımlanması görevlerini,

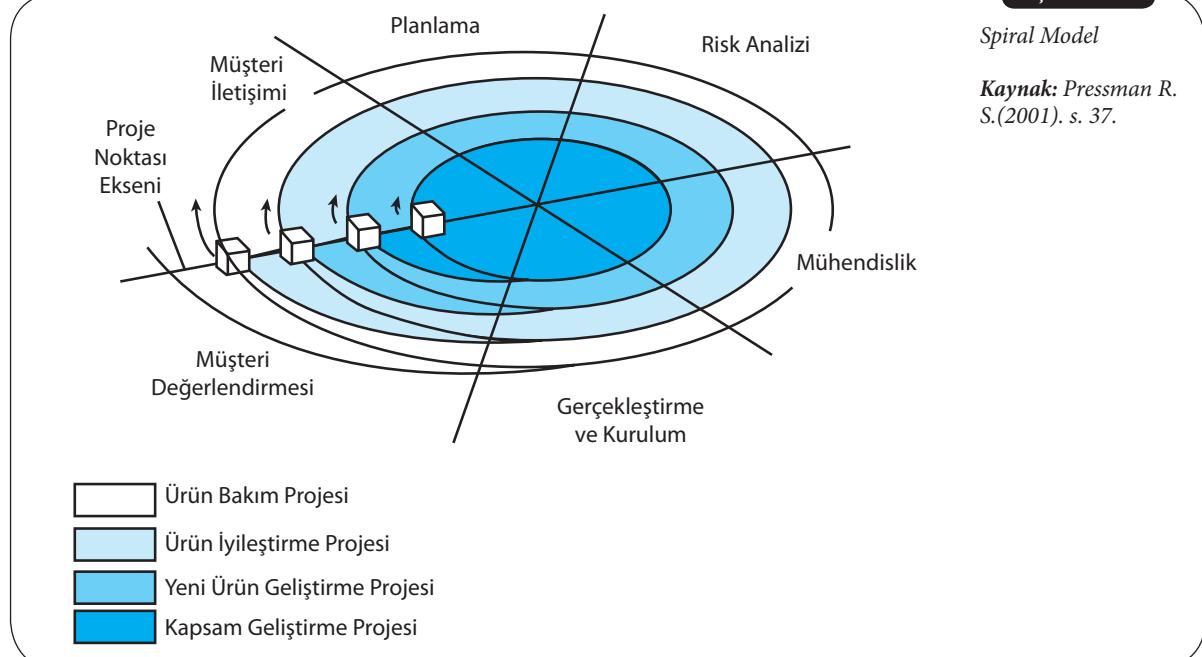
**Risk Analizi:** Teknik ve yönetim risklerinin belirlenmesi görevlerini,

**Mühendislik:** Uygulamanın bir veya daha fazla çözümünün oluşturulması görevlerini,

**Gerçekleştirme ve Kurulum:** Sistemin inşa edilmesi, test, kurulum ve kullanıcı desteği (belgeleme, kullanıcı eğitimleri) oluşturma görevlerini,

**Müşteri Değerlendirmesi:** Mühendislik çözümleri, gerçekleştirme ve kurulum sonrasında sistem hakkında müşteri değerlendirmesinin alınması görevlerini içerir.

**Şekil 2.6**



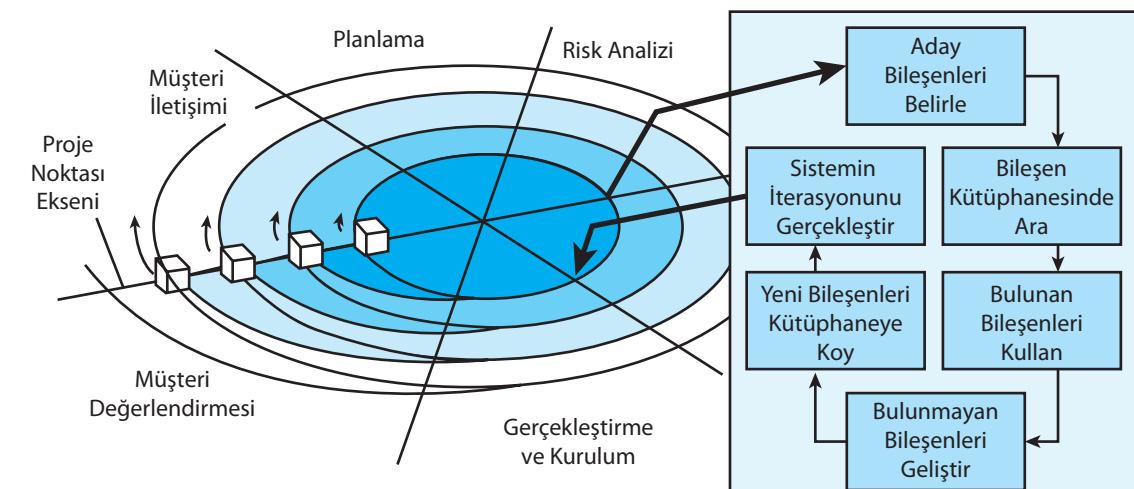
Bu evrimsel süreçte, sistem geliştirme takımı merkezden başlayarak saat yönünde bölgüler ile ilgili görevleri gerçekleştirir. Spiralde bir tur tamamlandığında ürün özellikleri tamamlanabilir; ikinci turda prototip geliştirilebilir; sonraki turlarda da adım adım sistemin daha gelişmiş sürümleri oluşturulabilir. Planlama bölmesinden her bir geçişte proje planında daha hassas düzenlemeler yapılır. Maliyet ve zamanlama müşterinin değerlendirmesine göre tekrar gözden geçirilerek ayarlanır. İlave olarak, proje yöneticisi sistemi tamamlamak için gereken planlanmış iterasyon sayısını ayarlar. Spiral model aynı zamanda sistemin yaşam döngüsü boyunca da uygulanabilir. Spiral model projenin her adımda teknik ve yönetim risklerini dikkate alarak risklerin büyük problemlere sebep olmadan giderilmesini sağlar.

### Bileşen Tabanlı Geliştirme Modeli

Nesne tabanlı teknolojiler Bileşen tabanlı geliştirme modelinin (Component-based development model) teknik çerçevesini oluşturur. Nesne, içinde veri ve bu veriler üzerinde işlem yapacak olan metotları bulunduran yazılım bileşenidir. Nesneler her uygulamada tekrar kullanılabilir. Bir nesne yapısı, bir sınıf (class) içinde tanımlanır. Sınıf içinde nesneyi oluşturan üye değişkenler ve metotlar açıkça tanımlanır. Eğer uygun bir şekilde tasarılanın ve geliştirilirse nesne tabanlı sınıflar farklı uygulamalar arasında ve bileşen tabanlı sistem mimarilerinde kullanılabilir. Bileşen tabanlı geliştirme modeli (Şekil 2.7) Spiral modelin bölgelerini uygular. Bununla birlikte, bileşen tabanlı modelde mühendislik, geliştirme ve kurulum bölgelerinde hazır paketlenmiş yazılım bileşenleri (sınıflar) kullanılır. Mühendislik bölümünü çalışmaları aday bileşenlerin tanımlamalarıyla başlar. Çözümde kullanılacak veri ve algoritmaların bileşenler oluşturulur.

**Şekil 2.7**

#### Bileşen Tabanlı Geliştirme Modeli



**Kaynak:** Pressman R. S.(2001). s.42.

Aday bileşenler tanımlandıktan sonra, geçmiş bilgi sistemi geliştirme projelerindeki bileşenlerden oluşturulmuş olan bileşen kütüphanesinden halen mevcut ya da yakın bileşenler bulunur ve yeniden kullanılır. Eğer aday bileşen kütüphanede yoksa nesne tabanlı metotlar ile geliştirilir. Son aşamada da kütüphaneden çıkarılan bileşenler ve uygulama ihtiyaçları doğrultusunda yeni geliştirilmiş bileşenler ile sistem geliştirilerek spiralin ilk iterasyonu tamamlanmış olur. Bileşen tabanlı model tekrar kullanılabilirliği destekleyerek kaynak (maliyet, zamanlama) kullanımında önemli kazanımlar sağlar.

## Çevik Model

Çevik model (Agile Model) sistemin gereksinimlerini belirleme, tasarım, geliştirme ve kurulumunda artan yaklaşımı kullanır. Geliştirme sürecinde sistem gereksinimleri genelde değiştiği durumlarda Çevik model en uygun yaklaşılardan biridir. Çevik modelde çalışan yazılım müşteriye hızlıca verilerek, yeni ve değişen gereksinimler ileri iterasyonlarda dâhil edilir. Burada amaçlanan, uzun zaman alan belgelemeyi azaltarak, tüm çabayı sistem geliştirmeye vermektedir. Çevik modelin felsefesi aşağıdaki Çevik Yazılım Geliştirme Manifestosunda verilmiştir.

*Bizler daha iyi yazılım geliştirme yollarını uygulayarak ve başkalarının da uygulamasına yardım ederek ortaya çıkartıyoruz.*

*Bu çalışmalarım sonucunda:*

*Süreçler ve araçlardan ziyade bireyler ve etkileşimlere,  
Kapsamlı dokümantasyondan ziyade çalışan yazılıma,  
Sözleşme pazarlıklarından ziyade müşteri ile işbirliğine,  
Bir plana bağlı kalmaktan ziyade değişime karşılık vermeye  
değer vermeye kanaat getirdik.*

*Özetle sol taraftaki maddelerin değerini kabul etmekle birlikte, sağ taraftaki maddeleri daha değerli bulmaktayız.*

**Kaynak:** <http://agilemanifesto.org/iso/tr/> adresinden alınmıştır.

Çevik modelin uygulanmasında dört teknik yaklaşım ön plandadır. Bunlar: kısa sürede sürüm teslimi (2 haftada bir), haftada 40 saat uygulama geliştirme (belgeleme yok), müşteri ile içi içe geliştirme ve çift kişi programlamadır. Çevik model kullanan yaklaşımalar, Uç Programlama (eXtreme Programming), Adaptif Yazılım Geliştirme (Adaptive Software Development), Scrum ve Özellik Güdümlü Programlamadır (Feature Driven Development).

Çevik yaklaşımın aşağıdaki tip sistemlerin geliştirilmesinde daha başarılı olduğu görülmüştür.

- i. Küçük ve orta büyüklükte sistemleri geliştiren işletmelerin ürün geliştirmesinde
- ii. Müşterinin geliştirme süreci içinde olduğu ve sistemin gereksinimlerinde belirli standartların (kalite, güvenlik vb.) aranmadığı durumlarda, işletme içinde özelleştirilmiş sistemlerin geliştirilmesinde

Özelliklerde ortaya çıkabilecek hataların önemli sorunlara yol açabileceği kritik sistemlerde çevik yaklaşım kullanılmamalıdır. Çünkü bu durumlarda geliştirme başlamadan önceki fazlarda tüm gereksinimler tam ve ayrıntılı tanımlanmış olmalıdır. Günümüzde Çevik geliştirme yaklaşımından en yaygın kullanılan Scrum aşağıdaki kesimde anlatılacaktır.

## Scrum

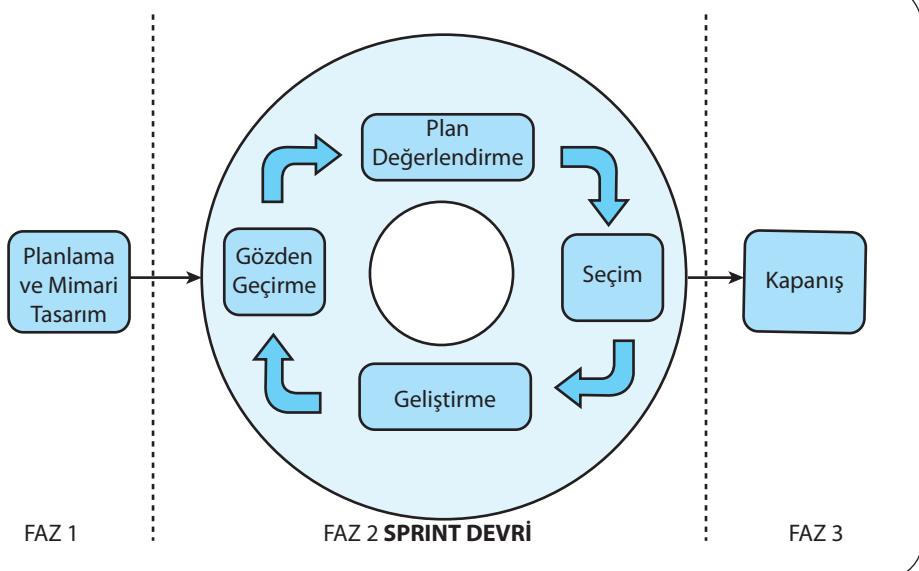
Scrum yaklaşımı çevik yöntemlerin teknik yaklaşımı yerine iteratif geliştirme yöntemi-ne odaklanmış bir çevik metottur.

Şekil 2.8'de Scrum yönetim süreci gösterilmektedir. Scrum' da 3 faz vardır. Birinci fazda proje genel hedefleri ve yazılım mimarisinin tasarımını oluşturmak üzere ana hat planlama yapılır. Bu aşamada oluşturulan projenin ana gereksinimleri önem sırasına göre bir liste (Product Backlog) olarak toplanır (Faz 1).

*Sekil 2.8*

*Scrum Yaklaşımı*

**Kaynak:** Sommerville I. (2015). s. 73.



Bu aşamayı her turda sistemde bir artım sağlayan *sprint devirleri* (sprint cycle) takip eder (Faz 2). Son aşamada da ürün belgeleri (yardım belgeleri, kullanıcı el kitapları) ve projeden elde edilen deneyimlerin değerlendirildiği proje kapanış fazı vardır (Faz 3). Scrum yaklaşımının iteratif yapısını 2'nci fazdaki Scrum *sprint devri* oluşturur. *Sprint devirleri* sabit zaman aralıklarında yapılır (tipik olarak 2-4 hafta). *Sprint devri* plan değerlendirme ünitesiyle başlar. Ardından geliştirilecek özellikler seçilir ve yazılım geliştirilir. *Sprint devrinde* geliştirilecek özelliklerin seçimi tüm proje ekibi katılır. Yazılım geliştirme sırasında ise geliştirme ekibi müşteriden izole edilerek sadece yazılım ile ilgilenir. *Sprint devrinin* sonunda, tamamlanmış sürüm sistem paydaşlarına dağıtılarak gözden geçirilmesi sağlanır.

SIRA SİZDE

3

**Hangi tip bilgi sistemlerinin geliştirilmesinde Cevik modelin kullanılması önerilmez?**

Sistem Geliştirme Modelerinin Seçimi

Anlatılan tüm geliştirme modeli yaklaşımlarında sistem analisti öncelikle organizasyonun yapısını, işleyiş biçimini, süreçleri vb. anlamak zorundadır. Sonra ayrıntılı sorular ile veri toplanır ve işletmedeki iş süreçleri belirlenir. Böylece gerçekleştirilecek uygulamanın tipi ortaya konulur. Farklı uygulama tiplerinde farklı sistem geliştirme teknikleri kullanılmalıdır. Bunun sebeplerinden bazıları aşağıda verilmiştir:

*Maliyet ve değişiklik isteklerinin sikliği:* Bazı sistemlerde değişiklik gereksinimlerini uygunluk çok aşırı maliyetli olabilirken, diğer sistemlerde müşteri gereksinimleri doğrultusunda sık sık değişiklik yapmak gerekebilir. Pahalı ön analiz gerektiren ve gereksinimlerin tutarlı ve doğru olarak tüm müşteri isteklerini kapsadığının gösterildiği sistemlerde değişikliğin maliyetleri yüksek olacaktır.

*İşlevsel olmayan gereksinimlerin önemi:* Farklı sistemlerde işlevsel olmayan gereksinimlerin öncelikleri farklı olur. Örneğin bankacılık bilgi sisteminde güvenlik gereksinimi birinci öncelikli olacaktır. Diğer yandan interaktif işlemlerin yoğun olduğu sanal gerçeklik sistemlerinde ise hızlı cevap verme ve kullanışlılık öncelikli gereksinimler olacaktır. Bu nedenle bankacılık sistemlerinde interaktif sanal gerçeklik tarzı arayüzler için çok fazla arayüz tasarımları gerekmeyecek, sanal gerçeklik kullanılan oyun tarzı sistemlerde ise yüksek güvenliği sağlamak üzere teknikler kullanılacaklardır.

*Bilgi sistemi ömür ve teslim zamanlaması:* Bazı sistemlerin ömrü nispeten kısa olurken (mobil, web tabanlı sistemler) diğerlerinin onlarca yıllık ömrü olur (büyük kumanda ve kontrol sistemleri). Bazı sistemler, yararlı olma ömrü kısa olduğundan hızlı olarak teslim edilmelidir. Bu tip sistemlerde kısa ömürlü geliştirme ve hızlı teslim etme imkânı olan teknikler tercih edilir. Uzun yaşam ömrü olan sistemlerde ise uzun dönem bakım ve destek hizmetlerini sağlayabilecek tasarım modelleri ve sistem süreçleri kullanılır.

Sistem geliştirme yaklaşımlarından, Doğrusal ardışık modelde kapsamlı planlama ve belgeleme vardır. Tüm gereksinimler ve geliştirme özellikleri ilgili fazlarda kayıt altına alınır. Artımlı model ve Çevik model gibi iteratif sistem geliştirme yaklaşımlarında ise gereksinim belirleme aşamasında ana hat planlama ve temel yazılım mimarisi ortaya konulur. SGYD fazlarını ardışık olarak kullanan Doğrusal ardışık modelde, sistem geliştirmesi başlandıktan sonra son sistem ortaya konuluncaya kadar uzun bir süre müşteri ürün hakkında bilgi almadır ve geliştirme sırasında değişikliğe açık değildir. Artımlı model, Bileşen tabanlı geliştirme modeli ve Çevik modelde ise sistem küçük artımlar (ara sürümler) ile gerçekleştirilmektedir. Böylece alt sürümler küçük ekipler ile kısa zamanda geliştirilebilir ve yeni sürüme geçmeden müşteri görüşü alınarak olası yanlış anlamaların ve uzun termin sürelerinin önüne geçirilmiş olur. Ancak bu yaklaşılarda belgeleme minimum tutulduğu için sistemin bakımı aşamasında zorluklar yaşanır. Bu nedenle küçük ve orta büyülükteki yazılım geliştirme şirketlerinde, kullanılan teknolojilerin hızlı değiştiği sistemlerde, çevik ve diğer iteratif yaklaşımlar kullanılır. Yaşam döngüsü uzun, büyük sistemlerin geliştirilmesinde ve/veya gereksinimlerinde belirli yönetmeliklere uyalma şartı olan sistemlerde Doğrusal ardışık model kullanılır. Uzun yaşam döngüsü olan sistemler, çok uzun süre kullanımda olacağının gereksinim, tasarım, geliştirme, değişiklik vb. belgeler olmadan bakım sürecini yürütmek çok fazla kaynak (maliyet, zaman) tüketimine neden olacaktır. Bu nedenle büyük ölçekli, uzun süre kullanımda olacak, belirli yönetmeliklere uygun sistemlerin geliştirilmesinde Doğrusal ardışık model kullanılır.

Önceki kesimde anlatılmış olan modellerin fazları karşılaştırmalı olarak Tablo 2.1'de verilmiştir. Tablo 2.1'de görüleceği gibi Prototip model sadece gereksinim oluşturma aşamasında kullanıldığından diğer sistem geliştirme modellerine alternatif değildir. Diğer modellerde ise süreç içindeki alt fazlar arasında çok fazla bir fark yoktur. Modellerdeki süreçler arasındaki farklar: her bir fazı kaç kez uyguladığı (plan güdümlü veya iteratif yaklaşım), her bir fazda harcanılan kaynak miktarı (zaman, maliyet), fazlarda üretilen belgelerin miktarı ve geliştirme aşamasında değişiklik isteklerinin nasıl yönetildiğine göre oluşmaktadır.

**Çevresel faktörlerin etkisini en aza indirecek interaktif seyahat planlama sistemi geliştirmek için hangi sistem geliştirme modeli kullanılmalıdır? Sebeplerini yazınız.**



SIRA SİZDE

4

**Tablo 2.1**  
Bilgi Sistemi Geliştirme  
Modellerinin Fazlarının  
Karşılaştırılması

Sistem Geliştirme Yaşam Döngüsü (SGYD)	Doğrusal Ardişik Model	Prototip Model	Artımlı Model	Spiral Model	Bileşen Tabanlı Geliştirme Modeli	Çevik Model (Scrum)
Kapsam ve Planlama	Kapsam ve Planlama	Tur 1, 2, ... Müşteri İstekleri	Artım 1,2,... Kapsam Planlama	Tur 1, 2, ... Müşteri iletişim	Tur 1, 2, ... Müşteri iletişim	Planlama ve Mimari Tasarım
Gereksinim Analizi	Gereksinim Analizi	Tur 1, 2, ... Prototip Model Geliştirme Güncelle Müşterinin Prototip Model Testleri	Artım 1,2,... Gereksinim Analizi	Tur 1, 2, ... Planlama Risk Analizi	Tur 1, 2, ... Planlama Risk Analizi	Devir 1,2,... Plan Değerlendirme Devir 1,2,... Seçim
Tasarım	Tasarım		Artım 1,2,... Tasarım	Tur 1, 2, ... Mühendislik	Tur 1, 2, ... Mühendislik Bileşen Tabanlı	Devir 1,2,... Geliştirme
Gerçekleştirme ve Test	Geliştirme ve Test		Artım 1,2,... Geliştirme ve Test	Tur 1, 2, ... Gerçekleştirme ve Kurulum Müşteri Değerlendirmesi	Tur 1, 2, ... Gerçekleştirme ve Kurulum Müşteri Değerlendirmesi	Devir 1,2,... Geliştirme Gözden Geçirme
Kurulum ve Dağıtım	Kurulum ve Dağıtım		Artım 1,2,... Sürüm 1,2,... Kurulumu	Tur 1, 2, ... Müşteri Değerlendirmesi	Tur 1, 2, ... Müşteri Değerlendirmesi	Kapanış
Operasyon ve Bakım	Operasyon ve Bakım					

## BİLGİSAYAR DESTEKLİ SİSTEM GELİŞTİRME ARAÇLARI

Bilgisayar destekli yazılım mühendisliği araçları CASE araçları olarak adlandırılır.

Bilgisayar destekli sistem geliştirme araçları izleyen kesimde 4 kategoride açıklanacaktır.

### Bilgisayar Destekli Yazılım Mühendisliği Araçları

CASE (Computer-Aided Software Engineering tools) sistem geliştiricilerinin karmaşık bilgi sistemlerini yüksek kalitede, planlanan zaman ve bütçede tamamlamasını destekleyen araçlardır. CASE araçlarının amacı yazılım geliştirme modellerindeki süreçlerin kontrol edilebilmesi, ölçeklenebilmesi ve kolay yönetilebilmesidir. CASE araçları yazılım geliştirmek için kullanılan yazılımlar olarak özetlenebilir.

CASE araçlarının mimarisinin merkezinde geliştiricilerin veritabanı vardır. Bu veri tabanı CASE deposu (CASE Repository) olarak adlandırılır. Bu deposun etrafında sistem modeli ve belgeleri oluşturmak için kullanılan araçlar, destekleyici kaynaklar bulunur.

CASE kaynakları:

- *Diyagram araçları*; sistem süreç, veri ve kontrol yapılarının grafiksel gösteriminde kullanılır. UML (Unified Modelling Language) tasarım araçları, Veri Modelleme araçları (Data Modelling tools) örnek verilebilir.
- *Bilgisayar gösterim ve raporlama üreticileri*; kullanıcılar sistemin nasıl ve neye benzediğini göstermeye yardımcı olur. Bu araçlar sayesinde sistem analisti veri gereksinimlerini ve aralarındaki ilişkileri daha kolay oluşturur.

- *Analist araçları*; diyagramlardaki, gösterim ve raporlamadaki eksik veya yanlış özelliklerin otomatik bulunmasına yardımcı olur.
- *Kalite yönetim araçları*; sistem modellerinin, tanımlamalarının, özelliklerinin, tasarımın bütünlüğünün, tutarlığının uygulanılan geliştirme süreci kural ve tanımlamalarına uygunluğunu analiz ve kontrol eden araçlardır.
- *Proje Yönetim araçları*; proje yöneticisinin proje planlarının, kaynak atamalarının, zaman çizelgesinin oluşturulması, projenin tanımlanmış olan zaman ve bütçe içinde yürütülmesini destekleyen araçlardır.
- *Tasarım ve program üretici araçları*; veritabanlarının ve program parçalarının otomatik olarak üretilmesini sağlar. Kod üreten yazılımlar (Source code generation tools), Versiyon kontrol ve Yazılım ayarlama araçları (Configuration tools), Model Dönüşüm yazılımları (Model Transformation), Kod Yeniden üretim yazılımları (Code refactoring) örnek uygulamalarıdır.
- *Test araçları*; işlemlerin ve veri trafiğinin benzetimini destekler, performansı ölçer ve test planlarının hazırlanarak yönetilmesini sağlar.
- *Belgeleme araçları*; sistem modellerinin, tanım ve gereksinim belgelerinin oluşturulması, organize edilmesi ve raporlanması sırasında kullanılır.

## Süreç Yönetim Araçları

Sistem geliştirme modelindeki fazların yönetilmesi, belgelenmesi, çıktıların yönetilmesi için süreç yönetim araçları kullanılır. Ayrıca geliştirilen sistemler kalite standartlarına uyumak zorunda ise kalite yönetim standartlarını yönetmek için yardımcı olur.

## Proje Yönetim Araçları

Proje yönetim faaliyetleri; planlama, tahmin ve kaynak atama, zaman çizelgesi oluşturma, proje zaman ve bütçesine göre ilerlemeyi izleme ve kontrolünden oluşur. Bu faaliyetler sistem geliştirme modeli fazları ile örtüşür. Proje yönetim araçları, proje yöneticisinin bilgi sistemini verilen gereksinimler, zamanlama ve bütçe ile geliştirmesi için yapılan faaliyetleri destekler, raporlamasında yardımcı olur.

## Uygulama Geliştirme Ortamları

Hızlı ve kaliteli yazılım geliştirme talebi nedeniyle, program derleyicilerinin kullanımı yerine uygulama geliştirme ortamlarına (ADE - Application Development Environment) geçilmiştir. Bu uygulama geliştirme ortamları sayesinde programlar daha basit ve verimli geliştirilebilir. Bunun için standart derleyicilerin yanı sıra uygulama ortamlarına aşağıdaki (farklı uygulama geliştirme ortamlarında farklı ilave özellikler olabilir) bazı ek özellikler eklenmiştir.

*Programlama dilleri ve derleyiciler*: Önemli hata ayıklama ve program yazma desteği ile programların problemleri kolayca bulmasını ve çözmесini sağlar.

*Arayüz oluşturma araçları*: Programcılar bileşen kütüphaneleri aracılığı ile hızlı arayüzler oluşturmasını sağlar.

*Ara katman yazılımları*: Programcılar geliştirdikleri yazılımı değişik veritabanlarına, bilgisayar ağlarına hızlı bir şekilde entegre etmesini destekler.

*Test araçları*: Geliştirilen yazılımın test senaryoları ile test edilmesini sağlar.

*Sürüm kontrol*: Ekip olarak çalışan programciların geliştirilen ve geliştirme sonrası kurulmuş ve çalışmakta olan yazılımların (kaynak ve çalışan yazılım) farklı sürümlerinin yönetimini, saklanması ve erişmesini sağlar.

*Yardım yazarlığı araçları*: Çevrimiçi yardım, çevrimiçi eğitim ve kullanıcı el kitaplarının yazılmamasında kullanılır.

## Özet



### Sistem geliştirme yaşam döngüsünün fazlarını tanımlamak

Sistem geliştirme yaşam döngüsü bilgi sistemlerinin planlanması, geliştirilmesi, kullanım ve bakım faaliyetlerini içeren fazlardan oluşur. Bu fazlar; Kapsam ve Planlama, Gereksinim Analizi, Tasarım, Gerçekleştirme ve Test, Kurulum ve Dağıtım, Operasyon ve Bakım biçiminde sıralanır.

**Kapsam ve Planlama:** Problemin kapsamı, fırsatları ve hedefleri tanımlanır.

**Gereksinim Analizi:** Sosyal araç ve tekniklerin yardımıyla gereksinimleri tanımlanır. Bu gereksinimler analiz edilerek işlevsel, işlevsel olmayan ve ortam gereksinimleri olarak sınıflanır. Bu fazın çıktısı alternatif çözüm önerisidir.

**Tasarım:** Alternatif çözüm önerileri mantıksal ve fiziksel sistem özelliklerine dönüştürülür.

**Gerçekleştirme ve Test:** Özellikleri ortaya konulmuş olan sistem geliştirilir ve testleri tamamlanarak çalışır hâle getirilir.

**Kurulum ve Dağıtım:** Uygulama yazılımı, geliştirme ortamından üretim ortamına geçirilmek üzere mevcut ya da yeni donanım üzerine kurulur. Sistem kullanıcıları eğitilir, eski verilerin yeni sisteme aktarımı sağlanır ve en son sistem testleri yapılır.

**Operasyon ve Bakım:** Kullanıcı talepleri ya da iş süreçlerindeki değişimler doğrultusunda sistem desteği sağlanır.



### Bilgi sistemlerini geliştirmede kullanılan modelleri sıralamak

Sistem geliştirme modelleri, bilgi sisteminin geliştirilmesi ve bakımındaki faaliyetleri, metotları, uygulamaları, çıktıları, sistem geliştirici ve yöneticilerinin kullandığı otomatik araçları tanımlar. Farklı sistem mimarileri, geliştirme ve bakım süreleri, alternatif sistem geliştirme yaklaşımının kullanılmasını destekler. Sistem geliştirmede kullanılan farklı modeller; Doğrusal ardışık model, Artımlı model, Spiral model, Bileşen tabanlı geliştirme modeli ve Çevik modeldir.



### Alternatif sistem geliştirme modellerinin farklı tip projelerde kullanımını değerlendirmek

Farklı uygulama tiplerinde farklı geliştirme teknikleri kullanılmalıdır. Pahali ön analiz gerektiren ve tüm gereksinimlerin tutarlı ve doğru olarak gösterildiği sistemlerde değişiklik maliyetleri yüksek olacaktır. Bu tip sistemlerde Doğrusal ardışık model kullanılır. Müşteri gereksinimleri doğrultusunda sık sık değişiklik yapmak gerektiğinde, bilgi sisteminin geliştirilmesinde iteratif modeller (Artımlı model, Spiral model veya Çevik model) tercih edilmelidir. Farklı sistemlerde işlevsel olmayan gereksinimlerin öncelikleri farklı olur. Örneğin bankacılık bilgi sisteminde güvenlik gereksinimi birinci öncelikli olacaktır. Diğer yandan interaktif işlemlerin yoğun olduğu sanal gerçeklik sistemlerinde ise hızlı cevap verme ve kullanışlılık öncelikli gereksinimler olarak karşımıza çıkacaktır. Bu nedenle bankacılık sistemlerinde güvenlik için önce ayrıntılı gereksinim ve tasarım çalışması tamamlanmalı daha sonra geliştirmeye başlanmalıdır. Bu amaçla Doğrusal ardışık model kullanılır. Öte yandan interaktif sanal gerçeklik sistemleri gibi müşteri gereksinimlerinin sıklıkla değişiklik gösterdiği sistemlerde ise iteratif modeller kullanılır. Bazı sistemlerin (mobil, web tabanlı sistemler), kullanım ömrü göreceli olarak kısa (2-3 yıl ya da daha az) olurken diğerlerinin (büyük sistemler; bankacılık sistemi, nükleer santral kontrol kumanda sistemi vb.) onlarca yıllık ömrü olur. Bu tip büyük sistemlerin geliştirilmesinde olduğu kadar gereksinimlerinde belirli yönetmeliklere uyulma şartı olan sistemlerde de Doğrusal ardışık model kullanılır. Kimi sistemler ise yararlı olma ömrüleri kısa olduğundan hızlı olarak teslim edilmelidir. Bu tip sistemlerde kısa ömürlü geliştirme ve hızlı teslim etme imkânı sunan teknikler tercih edilir. Bu nedenle, kullanılan teknolojilerin hızlı değiştiği sistemlerde, çevik ve iteratif modeller kullanılır.



Sistem geliştirmede kullanılan çeşitli otomasyon araçlarını tanımlamak

Otomasyon araçları bütün sistem fazlarını destekler;

- *Bilgisayar destekli yazılım mühendisliği* (CASE) sistem modellerinin çizimine, analizine destek olan ve sistem modellerini uygulama programlarına dönüştüren, destekleyen yazılım araçlarıdır.
- *Uygulama geliştirme ortamları* (ADE) bütünlüşmiş yazılım geliştirme araçlarıdır. Yeni uygulama yazılmalarının maksimum hız ve kalitede geliştirilmesine yardımcı olacak imkânları sunarlar.
- *Süreç yönetim araçları* sistem süreçlerini yönetmek, belgelemek, çıktıları yönlendirmek ve kalite yönetim standartlarını yönetmekte yardımcı olur.
- *Proje yönetim araçları* sistem geliştirme faaliyetlerine; planlama, tahmin ve kaynak atama, zaman çizelgesi oluşturma, proje zaman ve bütçesine göre ilerlemeyi izleme, kontrol ve proje ilerlemesini raporlamada yardımcı olur.

## Kendimizi Sınayalım

- 1.** Aşağıdakilerden hangi SGYD fazlarından biri **değildir?**
  - a. Kapsam ve Planlama
  - b. Geliştirme ve Test
  - c. Tasarım
  - d. Ekip oluşturma
  - e. Operasyon ve Bakım
- 2.** Aşağıdakilerden hangisi kapsam belirlemenin katılımcılarından biri **değildir?**
  - a. Sistem sahipleri
  - b. Dâhili sistem yöneticileri
  - c. Sistem geliştiricileri
  - d. Proje yöneticileri
  - e. Sistem analistleri
- 3.** Aşağıdakilerden hangisi iteratif sistem geliştirme modellerinden biri **değildir?**
  - a. Şelale modeli
  - b. Artımlı model
  - c. Spiral model
  - d. Scrum
  - e. Çevik model
- 4.** Yaşam döngüsü uzun olan bilgi sistemlerinin geliştirilmesinde hangi yöntem tercih edilmelidir?
  - a. Doğrusal ardışık model
  - b. Artımlı model
  - c. Spiral model
  - d. Prototip
  - e. Çevik model
- 5.** Aşağıdakilerden hangisi CASE (Bilgisayar Destekli Yazılım Mühendisliği) kaynaklarından biri **değildir?**
  - a. Diyagram araçları
  - b. Veritabanı araçları
  - c. Analist araçları
  - d. Kalite yönetim araçları
  - e. Proje yönetim araçları
- 6.** Aşağıdakilerden hangisi bilgi sistemleri tasarım aşamasında gerçekleştirilen dört temel faaliyetten biri **değildir?**
  - a. Veritabanı tasarımı
  - b. Bileşen tasarımı
  - c. Yapısal tasarımcı
  - d. Gereksinim tasarımcı
  - e. Arayüz tasarımcı
- 7.** Aşağıdakilerden hangisi Spiral Modelin bölümlerinden biri **değildir?**
  - a. Müşteri iletişim
  - b. Planlama
  - c. Risk Analizi
  - d. Gerçekleştirme ve Kurulum
  - e. Operasyon ve Bakım
- 8.** Fiziksel tasarıma geçmeden önce, verilen iş gereksinimleri ve mantıksal tasarım modellerinin değerlendirilmesinde hangi fizibilite yönetimi **kullanılmaz?**
  - a. Teknik fizibilite
  - b. Operasyonel fizibilite
  - c. Ekonomik fizibilite
  - d. Risk fizibilitesi
  - e. Ekip fizibilitesi
- 9.** Aşağıdakilerden hangisi sistem desteği faaliyetlerinden biri **değildir?**
  - a. Kullanıcıya yardımcı olma
  - b. Sistemi yeni gereksinimlere göre güncelleme
  - c. Bileşen tabanlı geliştirme kütüphanesini oluşturma
  - d. Sistemi kurtarma
  - e. Yazılım hatalarını onarma
- 10.** Sistem Yaşam döngüsü fazlarından birincisi aşağıdakilerden hangisidir?
  - a. Gereksinim Analizi
  - b. Mantıksal Tasarım
  - c. Kapsam ve Planlama
  - d. Kurulum ve Dağıtım
  - e. Gerçekleştirme ve Test

## Kendimizi Sınavalım Yanıt Anahtarı

- |       |  |
|-------|--|
| 1. d  | Yanıtınız yanlış ise “Bilgi Sistemleri Geliştirme” konusunu yeniden gözden geçiriniz.                    |
| 2. c  | Yanıtınız yanlış ise “Kapsam ve Planlama” konusunu yeniden gözden geçiriniz.                             |
| 3. a  | Yanıtınız yanlış ise “Bilgi Sistemi Geliştirme Modeleri” konusunu yeniden gözden geçiriniz.              |
| 4. a  | Yanıtınız yanlış ise “Sistem Geliştirme Modellerinin seçimi” konusunu yeniden gözden geçiriniz.          |
| 5. b  | Yanıtınız yanlış ise “Bilgisayar Destekli Sistem Geliştirme Araçları” konusunu yeniden gözden geçiriniz. |
| 6. d  | Yanıtınız yanlış ise “Tasarım” konusunu yeniden gözden geçiriniz.  |
| 7. e  | Yanıtınız yanlış ise “Spiral Model” konusunu yeniden gözden geçiriniz.                                   |
| 8. e  | Yanıtınız yanlış ise “Tasarım” konusunu yeniden gözden geçiriniz.  |
| 9. c  | Yanıtınız yanlış ise “Operasyon ve Bakım” konusunu yeniden gözden geçiriniz.                             |
| 10. c | Yanıtınız yanlış ise “Kapsam ve Planlama” konusunu yeniden gözden geçiriniz.                             |

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Şelale modeli belge güdümlü modeldir. Her fazın sonunda belgeler üretilir. Bu yöntemde ilgili fazın faaliyetlerini tekrar yapmak, belgeleri tekrar üretmek ve onaylamak çok maliyetli olacağından iterasyonlara izin verilmez.

### Sıra Sizde 2

Artımlı model kullanmanın avantajları aşağıdaki gibi sıralanır;

- Kritik fonksiyonlar müşteriye daha erken teslim edilir.
- Erken sürümler gereksinimleri incelemek için prototip gibi çalışır.
- Bütün projenin başarısızlık ihtimali düşüktür.
- Kritik müşteri gereksinimleri çok daha yaygın ve uzun sürede test edilir.

### Sıra Sizde 3

Aynı mekânda olmayan ekiperce geliştirilen büyük sistemlerde çevik metodlar kullanılmamalıdır. Bir ekibin çalışmasını diğer ekiperle koordine etmek çok zor olacaktır.

Sistem gereksinimlerinde kalite, güvenlik vb. standartların arandığı bilgi sistemlerinin geliştirilmesinde çevik yaklaşım lar kullanılmaz. Çünkü bu tip sistemlerin gerektirdiği belgeleme, işlemleri kayıt altına alma gereksinimleri çevik sistemlerce desteklenmez.

Özelliklerde ortaya çıkabilecek hataların önemli sorunlara yol açabileceği kritik sistemlerde çevik yaklaşım lar kullanılmamalıdır. Çünkü bu durumlarda geliştirme başlamadan önceki fazlarda tüm gereksinimler tam ve ayrıntılı tanımlanmış olmalıdır.

### Sıra Sizde 4

İnteraktif seyahat planlama sistemi karmaşık bir kullanıcı arayüzüne sahip ancak güvenilir ve kararlı olmalıdır. Bu nedenle gerçek kullanıcıların gereksinimlerine göre oluşturulan sistemlerin geliştirilmesinde en uygun yöntem olan Artımlı modelin kullanımı önerilmelidir.

## Yararlanılan ve Başvurulabilecek Kaynaklar

- Kendall K.E., Kendall J. E. (2006). **Systems Analysis and Design**, 8th Edition, Boston, Pearson Education Inc.  
 Pressman Roger S. (2001). **Software engineering: a practitioner's approach**, 5th Edition, New York, Mc Graw-Hill Inc.  
 Schwaber K., Beedle M. (2001). **Agile Software Development with Scrum**. 1st Edition, Englewood Cliffs, NJ: Prentice Hall.  
 Sommerville I. (2015). **Software Engineering**, 10th Edition, Boston, Pearson Education Inc.  
 Valavich J. S., George J. F., Hoffer J. A. (2012). **Essentials of Systems Analysis and Design**, 5th Edition, Boston, Pearson Education Inc.  
 Whitten J. L., Bentley L. D. (2007). **Systems and Design Methods**, 7th Edition, New York, Mc Graw-Hill Inc.

# 3

### Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Proje ve proje yönetimi kavramlarını tanımlayabilecek,
  - 🕒 Proje yöneticilerinin temel işlevlerini tanımlayabilecek,
  - 🕒 Proje yaşam çevrimi faaliyetlerini açıklayabilecek,
  - 🕒 Proje yönetim yazılımlarını tanıyararak kullanabilecek bilgi ve becerilere sahip olabileceksiniz.

### Anahtar Kavramlar

- Proje
- Proje Yönetimi
- PERT Şeması
- Gantt Şeması
- Proje Kapsam Belgesi
- İş Kırılım Yapısı
- Kritik Yol
- Zaman Çizelgesi
- Proje Yaşam Çevrimi

### İçindekiler

Sistem Analizi ve Tasarımı

Proje Yönetimi

- 
- GİRİŞ
  - PROJE YÖNETİM ARAÇLARI
  - PROJE YAŞAM ÇEVRİMİ

# Proje Yönetimi

## GİRİŞ

Sistem analistleri bilgi sistemi geliştirme konusunda proje yönetimi yaklaşımından yararlanırlar. Proje yönetimi; sistem geliştirme projelerinin müşteri bekłentilerini karşılayacak şekilde, belirlenen bütçe ve sürede tamamlanmasını sağlamayı amaçlar. **Proje**, zaman ek-seninde başı ve sonu olan, bir hedefe ulaşmak için planlı bir dizi faaliyetin verilen zaman ve bütçede uygulanması sonucunda çeşitli ürünlerin elde edildiği bir çalışmadır.

Etkin bir proje yönetimi insan, ürün, süreç ve proje olmak üzere dört ana unsuru odaklanır. **Proje yöneticisinin** insan unsurunu dikkate almadan proje yönetiminde başarılı olması mümkün olmayacağıdır. Projenin başlangıcında müşteriyle ortaklaşa yapılan yoğun çalışmalar sonucunda, projenin kapsamının doğru oluşturulmaması durumunda da ortaya konulan ürün yetersiz olacaktır. Proje yöneticisinin süreç yönetimine dikkat etmemesi (doğru araç, teknik ve yöntemleri kullanmaması) projeyi başarısız kılabilecektir. Başarılı bir proje yönetimi eksikliği de projenin başarısını tehlkeye atmış olur.

**Proje**, başlangıcı ve bitisi belli olan planlı faaliyetler dizisidir.

**Proje yöneticisi**; yönetim, liderlik, teknik ve müşteri ilişkilerini iyi yönetme vasıflarına sahip tecrübeli sistem analistidir.

## Proje Yöneticisinin Rolü

Proje yöneticisi; yönetim, liderlik, teknik ve müşteri ilişkileri konusunda tecrübeli bir sistem analistidir. Bu nedenle bu ünitede proje yöneticisinin sistem analisti olduğu varsayılarak her iki ifade birbirinin yerine kullanılacaktır. Proje yöneticisi projenin başlatılması, planlanması, yürütülmesi ve kapatılmasından sorumludur. Bazı organizasyonlarda proje yönetiminden tecrübeli bir sistem analisti sorumlu olurken bazılarında bu rol birden fazla analistin desteklediği tecrübeli bir analist tarafından yürütülebilir. Bu nedenle proje yönetim sürecini anlamak, sistem analistlerinin gelecekteki başarıları için önemlidir. Proje yöneticisi, alan ve genel proje yönetimi ile ilgili yeterliklerinin yanı sıra etkin proje yönetimi için aşağıdaki özellikleri de bünyesinde barındırmalıdır.

*Performans*: Proje yöneticisi sahip olduğu proje yönetim yaklaşımını uygularken neyi yapabildiğini ve neyi getirebileceğini bilmelidir.

*Kişilik*: Proje veya proje ile ilgili faaliyetleri yerine getirirken proje yöneticisi ekipine nasıl davranışması gerektiğini bilmelidir. Proje yöneticisi, proje hedeflerine proje kısıtlarını dengeliyerek yürüterken proje ekipine yön gösterme yeteneği olan liderlik, bireysel tutumlar ve kişiye özel karakteristiklerinden yararlanır.

Proje yöneticisi projenin tüm yönlerinden sorumludur. Proje yöneticisinin projede yerine getireceği başlıca görevler şunlardır:

- Proje yönetim planını ve tüm ilgili bileşenlerin planlarını hazırlamak,
- Projenin oluşturulan zaman çizelgesi ve bütçeye göre ilerlemesini sağlamak,
- Riskleri tanımlamak, izlemek ve oluşması hâlinde müdahale etmek,
- Proje ölçümlerini doğru ve zamanlı olarak oluşturmaktır.

Bilgi sistemleri projelerini başarıyla oluşturmak ve uygulamak için kaynakların, faaliyetlerin ve görevlerin yönetimi gereklidir. Bu yönetim faaliyetleri aşağıdaki başlıklar altında toplanabilir.

*Kapsam tanımlama:* Projenin sınırlarını tanımlar. Proje yönetici, faaliyetleri planlamak, beklenenleri yönetmek ve maliyetleri tahmin etmek için proje beklenen ve kısıtlarının kapsamını belirlemelidir.

*Planlama:* Projenin tamamlanması için yerine getirilmesi gereken görevlerin planlanması gereklidir. Planlama, yöneticilerin proje kapsam ve metodolojisinden anladıklarını temel alır.

*Tahminleme:* Projenin tamamlanması için gereken tüm görevler tahmin edilmek zorundadır. "Ne kadar zaman gereklidir?", "Kaç kişi gereklidir?", "Kişilerin yetenekleri ne olmalıdır?", "Bir görev başlamadan önce hangi görevler bitirmeli?", "Proje maliyeti nedir?" gibi soruların cevapları tahmin edilmeli ve proje modelleme araçları ile çözümlenmelidir.

*Çizelgeleme:* Proje yönetici, proje planı doğrultusunda bütün proje faaliyetlerinin zaman çizelgesini oluşturmak zorundadır. Hangi faaliyetin ne zaman gerçekleştirileceği belirlenmelidir. Proje zaman çizelgesi, tanımlanmış olan görevlerin süreleri ve öncelikleri dikkate alınarak oluşturulur.

*Organize etme:* Proje yönetici, proje ekibinin görevlerini belirler ve ekibin görevlerini anladığından emin olur.

*Yönetme:* Proje yönetici proje başladıkta sonra da ekibin faaliyetlerini yönetir. Bu anlamda proje yönetici ekip üyelerine karşı koordine etme, delege etme, motive etme, tavsiye etme, takdir etme ve ödüllendirme örneklerini sergilemelidir.

*İzleme - Kontrol etme:* Proje yöneticilerinin en zor ve önemli görevlerinden biri de projenin kontrolüdür. Proje yönetici hedefler, zaman çizelgesi ve maliyetler açısından projeyi izlemeli ve gerektiğinde uygun müdahalelerde bulunarak ilerleme durumunu raporlamalıdır.

*Kapatma:* Proje yönetici, proje sonunda başarı ve hataları değerlendirmelidir. Bu sayede deneyim kazanılmış ve sistem geliştirme sürecinin sürekli iyileştirilmesi için planlama yapılmış olur. İyileştirilmiş geliştirme süreci ise ileride yapılacak olan yeni projelerin başarısının teminatıdır.

## PROJE YÖNETİM ARAÇLARI

Proje yönetimini üstlenmiş olan sistem analisti, projenin başarıyla sonuçlanması için yönetim faaliyetlerini dikkatle yürütmem zorundadır. Bu amaçla proje yöneticilerini destekleyen araç ve teknikler geliştirilmiştir. Bunlardan en yaygın olarak kullanılanlar grafik değerlendirme ve gözden geçirme teknigi (GERT: Graphical Evaluation and Review Technique), proje değerlendirme ve gözden geçirme teknigi (PERT: Project Evaluation and Review Technique), Kritik yol metodu (CPM: Critical Path Method) ve Gantt şemalarıdır. İzleyen kesimde PERT ve Gantt Şemalarının kullanımı ele alınmıştır.

### PERT Şeması

PERT şemaları ilk olarak 1950'lerin son dönemlerinde Amerikan Deniz Kuvvetlerinin Polaris nükleer denizaltı projesinde, planlama ve kontrol amacıyla geliştirilmiştir. Bu projede PERT şeması kullanımının proje geliştirme süresinden iki yıl tasarruf ettirdiği raporlanmıştır. PERT, özellikle faaliyetlerin ardışık yerine paralel yapılması durumunda kullanılmıştır. PERT şeması bir projenin görevlerini ve bu görevler arasındaki ilişkileri gösteren grafik ağ modelidir. Şekil 3.1 (a)'da önceki üitede verilmiş olan SGYD fazları örnek bir PERT şemasında gösterilmektedir. Bu PERT şemasına göre *Mantıksal tasa-*

**PERT şeması;** proje görevleri arasındaki bağımlılıkları göstermek için kullanılan grafiksel ağ modelidir.

rum sonrasında *Tasarım Analizi* yapılması durumunda proje 405 günde, *Tasarım Analizi* yapılmadığı durumda ise 390 günde tamamlanacaktır. Bu şekilde projenin PERT şemasında başlangıçtan bitişe olan tüm yolların süresi hesaplanır ve bu alternatif yollardan en uzununa **kritik yol** denir. Kritik yol üzerindeki herhangi bir görevin teslimindeki gecikme, projenin tesliminde gecikmeye sebep olacaktır. Bu nedenle kritik yol üzerindeki görevler (kritik görevler olarak da adlandırılır) proje yöneticisi tarafından dikkatle izlenmelidir ve olası sorunlar üzerinde önemle durulmalıdır. PERT, projede görevlerin zaman çizelgelemesine başlamadan önce görevler arası ilişkileri ortaya çıkarmak için geliştirilmiştir. Kutular projeye görevlerini temsil etmektedir. Bu kutular projedeki görevlerin zamanlamalarını, tamamlanma oranlarını, gerçek başlama ve bitiş zamanlarını göstermek için kullanılır. Şekil 3.1 (b, c, d, e)'de sırasıyla %51 tamamlanmış kritik görev, başlanmış görev, bitmiş görev, %67 tamamlanmış görevin PERT şemasındaki kutu gösterimleri verilmiştir. Kutuların aralarındaki oklar ise bir görevin, diğer bir görevin başlangıç ya da bitişine bağlı olduğunu gösterir.

**Kritik yol;** kritik faaliyetlerden oluşur.

### Kritik yol niçin önemlidir?

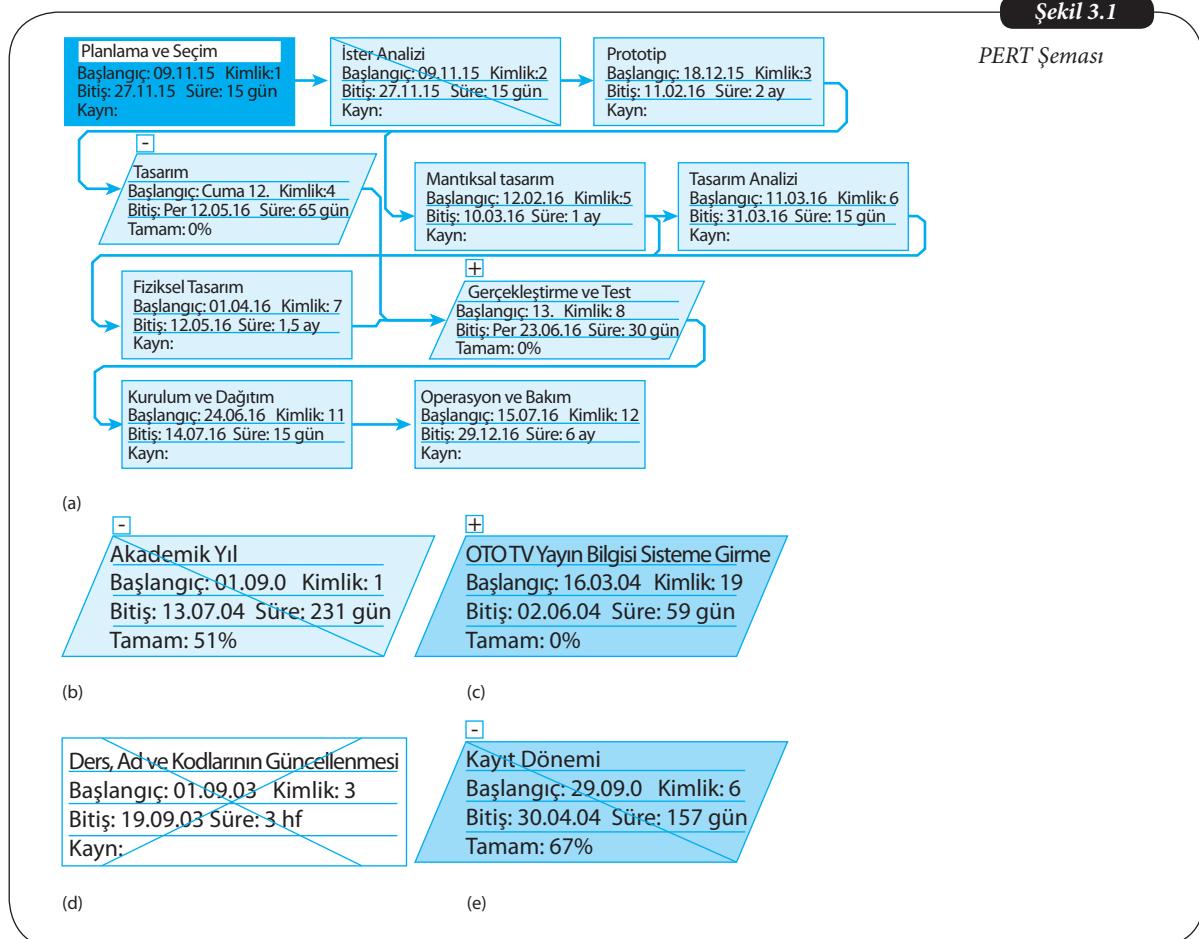


SIRA SİZDE

1

Şekil 3.1

PERT Şeması

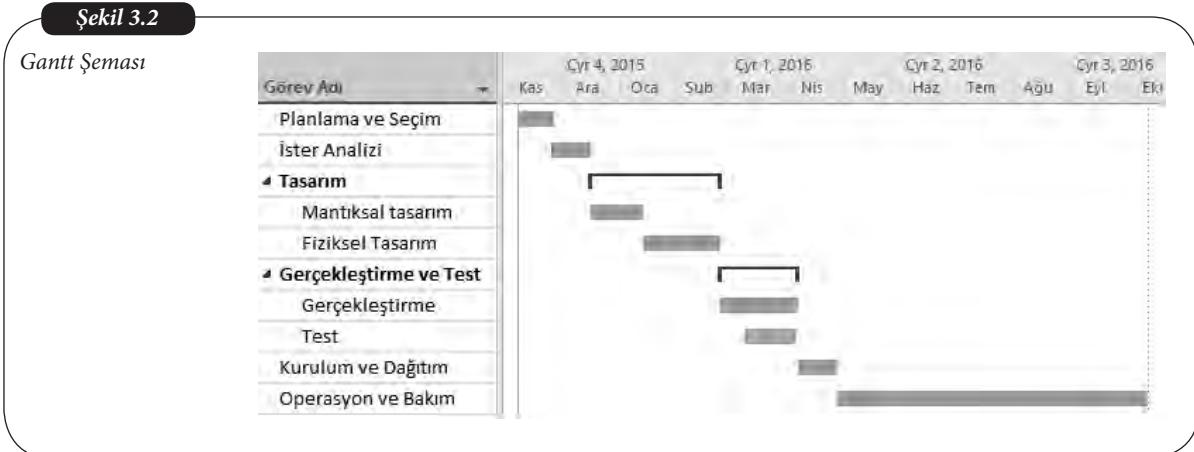


## Gantt Şeması

**Gantt şeması:** Proje görevlerini grafik çubukları ile bir takvim üzerinde göstermek için kullanılır.

Gantt şeması 1917 yılında Henry L. Gantt tarafından tasarlanmıştır. Gantt şeması proje zaman çizelgesinin oluşturulması ve değerlendirilmesi için kullanılan bir araçtır. Gantt şemasında, projede yürütülecek görevler bir takvim üzerinde yatay çubuklar ile gösterilir. Her bir yatay çubuk projedeki bir görevle karşılık gelir. Görevler en solda dikey olarak listelenir. Yatay eksen takvim zaman çizgisini oluşturur. Şekil 3.2'de 2. Ünitede'ki SGYD fazları Gantt şeması olarak gösterilmiştir.

**Şekil 3.2**



Gantt şeması, aynı zaman aralığında gerçekleşen diğer bir ifade ile örtüsen görevleri açıkça gösterir. Çubuklar tamamlanma yüzdelerini ve proje ilerlemesini göstermek üzere renklendirilebilir. Bu şemada bir bakısta hangi görevlerin zaman çizelgesinde ileride ya da geride olduğu görülebilir. Ancak görevlerin birbirleri ile olan bağlantıları ve öncelikleri şema üzerinde belirtilemez. Bu nedenle Şekil 3.2'deki Gantt şemasında, Gerçekleştirme görevinin takvimdeki başlangıcının Fiziksel Tasarım görevinden sonra geldiği görülmektedir. Ancak Fiziksel Tasarım görevindeki bir gecikmenin, Gerçekleştirme görevinde de bir gecikmeye sebep olup olmayacağı diğer bir ifadeyle ardışıklığın tesadüfen olup olmadığı belirlenemez. Gantt şeması, hazırlanması, öğrenmesi, okunması ve kullanımı kolay olduğu için çok kullanılır.

Gantt ve PERT şemaları proje görevlerinin grafiksel gösterimleridir. Bununla birlikte Gantt şeması, proje zaman çizelgesinin takibi açısından daha kullanışlıken PERT şeması görevler arası ilişkilerin izlenmesinde daha kullanışlıdır.

## Proje Yönetim Yazılımları

Proje yönetim yazılımları, proje yöneticilerine proje süresince yardımcı olmak için geliştirilmiştir. Proje yöneticisi proje planlama ve zaman çizelgesi oluşturmak, bütçe oluşturmak, ilerleme ve maliyetleri izlemek, raporlama ve değişikliklerin etkisini görmek için bu yazılımları kullanır. Bu amaçla geliştirilmiş bazı proje yönetim yazılımları şunlardır: Microsoft Project, Primavera P6, Jira, Daptiv PPM, Innotas PPM, Instantis EnterpriseT-track ve PowerSteering PPM.

İlerleyen kesimlerde proje yönetim yazılımları ile proje yönetim ve modelleme yaklaşımları açıklanacaktır. Bu açıklamalar yapılrken Microsoft firmasının üniversite akademik paketi, öğrencilerin ücretsiz kullanımına açık olduğu için Microsoft proje yazılımı olan MS Project (MS Proje olarak adlandırılır) kullanılacaktır. Diğer proje yönetim yazılımları gibi MS Proje yazılımı da PERT ve Gantt şemalarının oluşturulmasını ve iki şemanın ortak kullanımını destekler. Şekil 3.3 (a)'da örnek bir MS Proje Gantt şeması verilmiştir. Şekil 3.3 (b)'de ise Gantt şemasında örneği verilen projenin MS Proje PERT şeması gösterilmektedir. PERT şeması gösteriminde kullanılan görev kutularının içerikleri özelleştirilebilir.

Şekil 3.3 (a) üzerindeki baloncuqların gösterdiği öğelerin anlamları aşağıda verilmektedir.

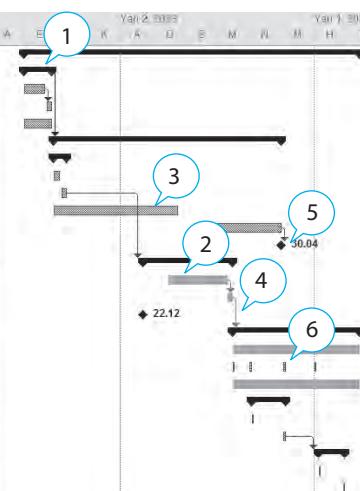
1. Siyah çubuklar *özet görevlerdir*. Bunlar projenin fazlarını göstermektedir ve özet görevlerin altında alt görevler vardır.
2. Kritik zamanlamayı oluşturan görevler kırmızı çubuklar ile gösterilir. Bu görevlerdeki herhangi bir gecikme diğer görevlerin ve dolayısı ile projenin (zaman planının) gecikmesi anlamına gelir.
3. Mavi çubuklar, projenin zaman çizelgesi için kritik olmayan görevlerin gösteriminde kullanılır. Bu görevlerdeki (proje zaman planındaki son tarihe kadar olan) gecikme diğer görevlerin ve projenin gecikmesine sebep olmaz.
4. Kırmızı oklu çizgi, iki kritik görev arasındaki bağımlılığı gösterir. Kritik olmayan görevler arasındaki bağımlılık ise mavi oklu çizgi ile gösterilir.
5. Projedeği kilometre taşları, karo şekli ile gösterilir. Kilometre taşlarının süresi yoktur. Bunlar önemli bir görevin bitimini ya da çıktıyı gösterir.
6. Düzenli bir zamanlama izlemek kaydıyla birkaç günde bir ya da her ay yinelenecek bir görev varsa MS Proje bunun yinelenen görev olarak oluşturulmasına izin verir. Böylece, görevi (örn: şekil 3.3 a'da görev 19) bir kez ayarlamak çoklu gösterim için yeterli olur.
7. MS Proje Gantt şemasında görev adı (Task Name) sütunu proje çalışmalarını ve görevlerini adlandırmak için kullanılır.

Şekil 3.3

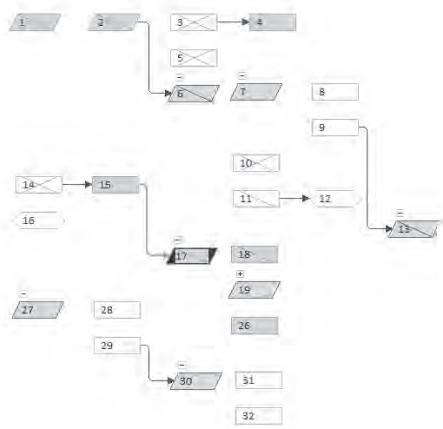
MS Proje (a) Gantt Şeması (b) PERT Şeması

Task Name	Süre	Başlangıç	Bitiş
1 Akademik Yıl	7	231 gün	Pzt 01.09.03 Sal 13.07.04
2 Dönen Öncesi Hazırlık	20 gün	Pzt 01.09.03	Cum 26.09.03
3 Ders, Ad ve Kodamını Güncellemesi	3 tf	Pzt 01.09.03	Cum 19.09.03
4 Akademik Takvimin Girişimi	5 gün	Pzt 22.09.03	Cum 26.09.03 3
5 Yeni Kayıt Öğrencileri İçin Çalışma	1 ay	Pzt 01.09.03	Cum 26.09.03
6 Kayıt Dönemi	157 gün	Pzt 29.09.03	Cum 30.04.04 2
7 Ders Kayıtları	10 gün	Pzt 29.09.03	Cum 10.10.03
8 Yeni Kayıtlar(1.Sınıf)	1 tf	Pzt 29.09.03	Cum 03.10.03
9 Diğer Kayıtlar	1 tf	Pzt 06.10.03	Cum 10.10.03
10 Harç Ödemesi	4,25 ay	Pzt 29.09.03	Cum 23.01.04
11 Cezalı Harç Ödemesi	2,85 ay	Pzt 16.02.04	Cum 30.04.04
12 Harç Ödemesi Raporu	8 gün	Cum 30.04.04	Cum 30.04.04 11
13 Öğrenci Bilgileri	61 gün	Pzt 22.12.03	Pzt 15.03.04 9
14 Kayıt Sonrası Öğrenci Bilgileri Alınması	2 ay	Per 15.01.04	Çar 10.03.04
15 Öğrenci Bilgilerinin Sisteme Girişmesi	3 gün	Per 11.03.04	Pzt 15.03.04 14
16 Ders İşlemleri kontrol raporu	0 gün	Pzt 22.12.03	Pzt 22.12.03
17 Dersler	90 gün	Sal 16.03.04	Sal 13.07.04 15
18 TV-Radyo Yayını	90 gün	Sal 16.03.04	Sal 13.07.04
19 OTO TV Yayın Bilgisi Sisteme Girmesi	59 gün	Sal 16.03.04	Çar 02.06.04
25 Danışmanlık Dersleri	90 gün	Sal 16.03.04	Sal 13.07.04
27 Ara Sınav	24 gün	Cmt 03.04.04	Sal 04.05.04
28 Ara Sınav Yapılması	2 gün	Cmt 03.04.04	Paz 04.04.04
29 Ara Sınav Bilgisi Sisteme Girmesi	2 gün	Pzt 03.05.04	Sal 04.05.04
30 Final Sınavı	19 gün	Cmt 05.06.04	Sal 29.06.04 29
31 Final Sınav Yapılması	2 gün	Cmt 05.06.04	Paz 06.06.04
32 Final Sınav Bilgisi Sisteme Girmesi	2 gün	Pzt 28.06.04	Sal 29.06.04

(a)



(b)



SIRA SİZDE



**PERT ve Gantt şemaları nedir? Hangisinin kullanılacağına nasıl karar veririz?**

## PROJE YAŞAM ÇEVİRİMİ

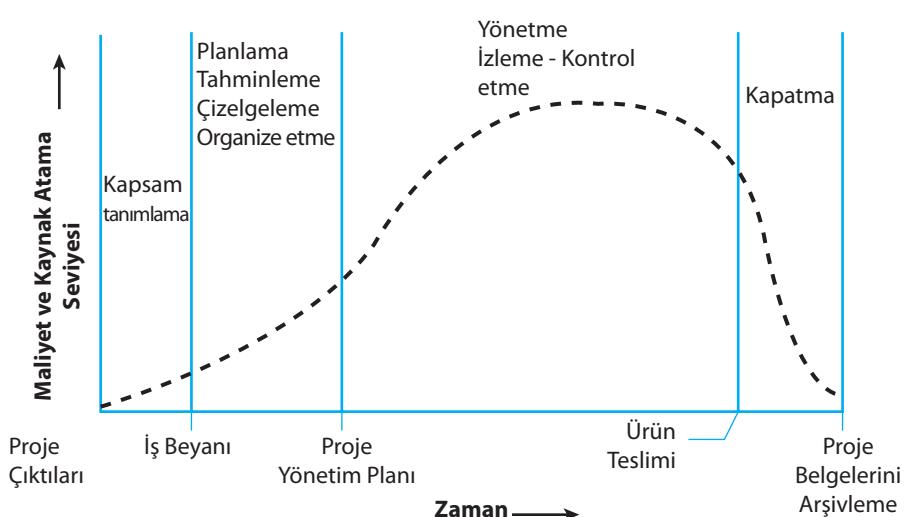
Proje yaşam çevrimi faaliyetleri, aslında ünitenin başlangıcında incelemiş olan proje yönetim faaliyetleridir. Proje yönetim sürecinde proje ekibi, projenin kapsamını, kaynaklarını, zaman çizelgesini ve bütçesini oluşturur, projenin izleme – kontrol etme ve kapanış faaliyetlerini gerçekleştirir.

Projeler boyut ve karmaşıklık açısından farklılık gösterir. Bu farklılıklara rağmen, bütün projelerde yaşam çevrimi boyunca, Şekil 3.4'teki eğriye benzer *maliyet* ve *kaynak atama* seviyeleri gözlenir. Grafikte görüldüğü gibi en fazla maliyet ve kaynak kullanımı *yönetme – izleme – kontrol etme* faaliyetlerinden oluşmaktadır.

**Şekil 3.4**

Proje Yaşam Çevrimi  
Boyuncu Tipik Maliyet  
ve Kaynak Atama  
Seviyesi Değişimi

**Kaynak:** Kendall K.E.,  
Kendall J. E. (2006),  
s.13'ten uyarlanmıştır.



Projeye ilişkin yaşam çevrimi faaliyetleri Tablo 3.1'de verilmiştir. Bu faaliyetlerin 2.Ünite'deki sistem geliştirme yaşam döngüsü fazlarının tamamıyla örtüşmekte olduğu kolayca izlenebilmektedir. SGYD'deki Kapsam ve Planlama, Gereksinim Analizi fazlarında proje yaşam çevrimi faaliyetlerinden *kapsam tanımlama*, *planlama*, *tahminleme*, *çizelgeleme* ve *organize etme* gerçekleştirilir. SGYD'deki Tasarım, Gerçekleştirme ve Test fazlarında proje yaşam çevrimi faaliyetlerinden *yönetme ve izleme – kontrol etme* yapılır. SGYD Kurulum ve Dağıtım fazında da proje yaşam çevrimi faaliyetlerinden *kapatma* gerçekleştirilir.

Proje Yaşam Çevrimi Faaliyetleri	Faaliyet Çıktıları	Ekip/Proje faaliyetleri
Kapsam tanımlama	İş beyanı	Proje tüzüğü ve ekip oluşturma
Planlama	İş kırılım yapısı	Deneyim ve yöntemler
Tahminleme	Görev süreleri	Tahminler ve öngörüler (tamamlanmış proje deneyimleri kullanılır)
Çizelgeleme	PERT ve Gantt şemaları	Zaman kısıtları
Organize etme	Kaynak atama ve bütçeleme	Kaynak taahhüdü (proje bütçe kısıtları içinde)
Yönetme	Ekip çalışmasını yönetme, Proje ara teslimleri ve kilometre taşı tamamlama	Delegasyon ve yönetme
İzleme – Kontrol etme	Proje tamamlama raporu, harici proje ilerleme raporları oluşturma	Görev ilerleme izleme ve kontroller
Kapatma	Proje deneyimlerinin toplanması Sistem geliştirme süreci ve proje yönetiminde iyileştirmeler	Deneyimler, gözlemler ve öneriler

**Tablo 3.1**  
Proje Yaşam Çevrimi  
Faaliyetleri - Sistem  
Geliştirme Süreci

İzleyen kesimde, proje yaşam çevrimi faaliyetleri incelenecuk ve her biri için hangi proje yönetim araçlarının ve tekniklerinin nasıl kullanıldığı açıklanacaktır.

## Kapsam Tanımlama

Kapsam, projeden beklentileri tanımlar. Proje kapatıldığında ya da sonlandırıldığında bu beklentilerin karşılanma oranı projenin başarı derecesini belirler. Kapsam; üzerinde çalışılan, analiz edilen, tasarlanan, inşa edilen ve uygulanılan iş parçalarının ve projenin sınırlarını tanımlar. Kapsam aynı zamanda sistemin proje dışında kalan yönlerini de tanımlar. Aşağıda verilmiş olan beş temel faktör hakkındaki soruların cevapları, proje kapsamının belirlenmesine büyük ölçüde yardımcı olur.

- Ürün: Ne isteniyor?
- Kalite: Ne kadar iyi olmalı?
- Zaman: Ne zaman isteniyor?
- Maliyet: Proje bütçesi ne kadar?
- Kaynaklar: Hangi kaynaklar kullanılacaktır?

Yukarıdaki faktörler üzerine yapılacak tartışmalar sonucunda, gerçekleştirilecek projenin görevlerini tanımlayan hedefler oluşturulur. Bununla birlikte, sistem analisti projeyi başlatmadan önce, işletme ve proje kullanıcıları için projenin gerçekleştirilmesinin mümkün olup olmadığını saptamalıdır. Bu işlem fizibilite çalışması olarak adlandırılır. Fizibilite çalışmasının doğru yapılması işletme ve analistlere zaman ve para kazandıracaktır. Analist bir projeye başlamadan önce onun teknik, ekonomik ve operasyonel açılarından uygun olduğunu göstermelidir.

## Teknik Fizibilite

Analist, yeni sistemi mevcut teknik kaynaklar ile geliştirmenin mümkün olup olmadığını araştırmalıdır. Aynı zamanda analist, proje hedeflerini geliştirecek ekibin teknik yeterliğinin olup olmadığını da incelemelidir. Eğer ekip yeterli değilse ilave programcılar, test mühendisleri veya farklı programlama yetenekleri olan kişiler proje için kiralananabilir. Ba-

zen de bilgi sisteminin geliştirilecek bir bileşeninde teknik yetersizlik varsa doğrudan dış kaynaktan temin edilebilir. Diğer bir konu da yazılım paketlerinin mevcut olup olmadığı veya gereksinimleri karşılayıp karşılayamadığının araştırılmasıdır. Projenin başarısı için başlamadan önce teknik yetersizlikler tamamlanmalıdır.

### **Ekonominik Fizibilite**

Ekonominik fizibilite ile analiz edilecek temel kaynakların maliyetleri; sistemin çalışması için müşterilerin ve sistem analiz ekibinin maliyeti, işletmedeki çalışanların zaman maliyeti, öngörülen donanımın ve geliştirilecek yazılımın maliyetidir. Eğer sistem ekonomik olarak yapılabılır değilse sistem geliştirme projesi sonlandırılmalıdır.

### **Operasyonel Fizibilite**

Sistemin teknik ve ekonomik fizibilitesinin yeterli olduğu varsayılsa bile sistem analisti operasyonel fizibilitenin de uygunluğunu kontrol etmelidir. Operasyonel fizibilitede; yeni sistemin kurulum ve tesliminden sonra, son kullanıcıların isteklerini karşılayıp karşılaşmadığına bakılır. Eğer kullanıcılar mevcut sistemlerinden memnunsa, herhangi bir problem görmüyorlarsa ve yeni sistemin gereksinimlerinin oluşturulması sürecine dâhil edilmederse, yeni sistemin uygulamaya alınmasına güçlü bir direnç gösterirler. Kullanıcıların işlemlerini daha kolay ve verimli bir şekilde yapabilmesi hâlinde yeni sistemin kabulü daha kolay olacaktır. Bu nedenle operasyonel fizibilitede daha çok kullanıcı arayüzleri ve operasyonel kullanımındaki veriler incelenir.

**Proje kapsam belgesi:** Projede yapılacak olan işlerin özet bir tanımıdır. Genel olarak kapsamı, proje tanımını, projeye genel bakış ve bilgi sistemi paydaşları arasında uzlaşmayı içerir.

Projenin fizibilite çalışması sonucuna göre sistemi geliştirme kararı verildikten sonra, kapsamı belirlemek üzere, projede gerçekleştirilecek işleri tanımlayan bir **proje kapsam belgesi** (iş beyanı) oluşturulur.

### **Planlama**

PMBOK proje planına göre oluşturulacak proje ana planı içinde önemle ele alınması gereken dokuz temel konu bulunur. Bunlar; Genel bakış, Amaçlar, YönetSEL yaklaşım, Sözleşme konuları, Takvim, Kaynak ihtiyaçları, İnsan kaynağı, Değerlendirme yöntemleri, Potansiyel problemlerdir. Bu konular proje tiplerine göre dikkate alınmalı ve uygulanmalıdır.

Proje kapsamı oluşturulduktan sonra proje ana planındaki konular dikkate alınarak işler belirlenir. İşler projedeki yapılacak olan çalışmalar ve görevlerdir. Proje yaşam çevriminin planlama faaliyeti içinde ayrıntılı görev belirlemesi yukarıdan aşağıya iş ayrıştırma yaklaşımıyla yapılır. Böylece proje içindeki çalışmalar ve proje çıktılarının teslimatları yönetilebilir alt gruplara bölünmüş olur. Planlanan çalışmaların bölündüğü bu alt gruplar genellikle çalışma paketi adını alır. Önceki ünitede bilgi sistemi geliştirme fazları verilmiştir. Bu fazların proje planlama ve zaman çizelgesinde çalışma paketi olarak kullanılması hâlinde çok fazla kaynak ve zaman alabilecektir. Bu nedenle, sistem geliştirme fazlarının planlanabilir, zamanlanabilir ve atanabilir şekilde daha küçük çalışma paketi ve görevlere ayrıstırılması gereklidir. Bazı uzmanlar çalışma paketlerinin iki hafta veya daha az sürede gerçekleştirilebilecek şekilde alt görevlere bölünmesini önermektedir.

**İş kırılm yapısı:** Projenin fazlara, faaliyetlere ve görevlere hiyerarşik olarak ayırtılmasını ifade etmek için kullanılan grafiksel gösterimdir.

Sonuç olarak, proje yöneticisi proje çalışmalarına ve çıktılarına göre yapılacak işleri detaylandıracaktır. Genelde sistem geliştirme süreci fazları, her bir projenin doğasına göre alt parçalara (çalışma paketleri ve görevlere) ayrılır. Bu şekilde proje fazlarını, çalışmalarını, görevlerini belirleme ve kayıt altına almakta kullanılan araç iş kırılm yapısıdır. **İş kırılm yapısı (İKY)** bir projeyi fazlara, çalışma paketlerine ve görevlere hiyerarşik olarak ayırtırmayı tanımlar. Aşağıdaki örnekte de görüldüğü gibi bir projede fazlar, çalışma paketleri ve görevler içe kaydırma ve numaralandırma ile ayırtılmaktadır. Bu örnekte Faz 1 iki çalışma paketinden oluşurken Faz 2 bir çalışma paketinden oluşmaktadır. Faz 1 altındaki çalışma paketi 1 ise iki alt görevden oluşmaktadır.

1. Proje Faz 1
  1. 1 Faz 1 Çalışma paketi 1
    1. 1. 1 Faz 1 Çalışma paketi 1 Görev 1
    1. 1. 2 Faz 1 Çalışma paketi 1 Görev 2
  1. 2 Faz 1 Çalışma paketi 2
2. Proje Faz 2 Çalışma paketi 3
3. ...

MS Proje'de İKY, Gantt şemasında ana hat stilinde (çalışma paketlerinin ve görevlerin içe kaydırılarak yazılması ile) gösterilir. Şekil 3.3 (a)'da görüldüğü gibi MS Proje Gantt şemasında bir sütun (ışaret no 7) İKY için kullanılır. İKY'de özel görevler kullanılabilir.

**Kilometre taşları** projenin önemli aşamalarında, çıktılarında kontrol amaçlı kullanılır. Örneğin bilgi sistemi projelerinde, gereksinim fazı sonucunda gereksinim belgesi tanımlama, bir kilometre taşı olabilir. Bunun gibi proje yaşam çevrimi süresince önemli alt faaliyet ve alt ürün çıktılarını göstermek ve kontrol noktası oluşturmak için kilometre taşı kullanılabilir.

**Kilometre taşı:** Projenin bazı faz, faaliyet ya da görevleri tamamlandığında varılan önemli teslim zamanlarını, çıktıları gösteren olaydır.

#### İş kırılımı nedir? Ne amaçla kullanılır?



SIRA SİZDE

### Tahminleme

İş kırılım yapısı uygun bir seviyede oluşturulduktan sonra, proje yöneticisi her bir görevin süresini tahmin etmek zorundadır. Herhangi bir görevin süresi ise geliştirme ekibinin büyüğüğe, kullanıcı sayısına, kullanıcıların yeteneğine, iş sisteminin karmaşıklığına, bilgi teknolojisinin yapısına, ekip çalışanlarının deneyimine, diğer projeler için ayrılmış olan zamana ve diğer projelerdeki deneyimlere bağlı olarak değişir.

MS Proje'de tüm fazlar, faaliyetler ve görevlerin tamamı görevler (task) olarak adlandırılır. İş kırılım yapısı özet görevleri ve basit görevleri içerir. Özet görev, alt görevleri ve alt özet görevleri içermekte olup fazları veya faaliyetleri tanımlar. Basit görevler tek olup alt görevleri içermez. Proje yöneticisinin görevi, basit görevlerin zamanını tahmin ederek tanımlamaktır. Kilometre taşlarını gösteren basit görevlerde ise süre yoktur. MS Proje'de kilometre taşları, süresi sıfır olan basit görevler olarak girmekte ve grafikte karo şeklinde gösterilmektedir. Özet görevlerin alt parçalarını oluşturan basit görevlerin süre ve başlangıç tarihleri girildiğinde, özet görevlerin süre ve başlangıç-bitiş tarihleri MS Proje tarafından otomatik olarak oluşturulur. Basit görevlerin süreleri tahmin edilerek girilirken *geçen süre* kavramına göre düzenlenmelidir. Geçen süre kişilere göre iki önemli faktörü dikkate alır. Bunlar;

**Verimlilik:** Hiç bir çalışan %100 verimlilikle çalışmaz. Her çalışan kahve arası, öğlen yemeği, lavabo ihtiyacı, haberleşme, yazışma, proje ile ilgili olmayan görevlerde yer alma için ara verir. Bir çalışanın verimliliğinin oranı için farklı görüşler bulunsa da genelde %75 varsayılar. Tamamlanmış projelerden toplanmış olan veriler varsa daha doğru verimlilik tahminleri yapılabilir.

**Kesme:** Çalışanlar telefon görüşmeleri ve diğer plansız kesmeler (interrupts) nedeniyle işine ara vermek zorunda kalır. Kesmelerin oranı, farklı çalışanlar için değişmekle birlikte bir çalışan için günde %10'dan %50'ye kadar değişebilir.

Böylece bir görev %100 verim ve hiç kesme olmadan 10 saatte tamamlanırsa %75 verimle ve günde %15 kesme ile çalışan bir işçi için doğru geçen süre tahmini 15.6 saat olacaktır. Verimlilik ve kesme ile yapılan toplam saat hesabı aşağıdadır:

- 10 saatlik çalışma için %75 verimlilikle 13.3 saat çalışmak gereklidir ( $10/0.75=13.3$ ).
- 13.3 saatlik çalışma %15 kesme ile yapıldığında toplamda **15,6** saat geçen süre gereklidir ( $13.3/[1-0.15]=15.6$ ).

Görev sürelerinin tahmini için birçok teknik vardır. Bu klasik tekniklerden bir tanesi de *Büyük ihtimal ile gerçekleşme süresinin* hesaplanmasıdır. *Büyük ihtimal ile gerçekleşme süresi; iyimser, kötümser ve beklenen* sürelerin doğrusal oranda birleşimiyle oluşturulur. *Büyük ihtimal ile gerçekleşme süre tahmini* aşağıdaki gibi hesaplanır:

*İyimser Süre (IS) tahmini:* Hastalık için bile olsa kesme ve gecikmelerin olmadığını varsayar.

*Kötümser (KS) Süre tahmini:* Grevler, hastalık, gereksinimlerin yanlış belirlenmesi, gerekli teçhizatın zamanında gelmemesi, sistem karmaşıklığının yanlış tahmini gibi olabilecek tüm kesme ve gecikmelerin olacağını varsayar.

*Beklenen Süre (BS) tahmini:* Çalışanların nadir olarak hastalanacaklarını varsayar ve deneyimsiz personelin ve bunların eğitimlerinin sebep olacağı kesme ve gecikmeleri belirleyerek varsayımda bulunur.

*İyimser, kötümser ve beklenen süre tahminlerine* dayalı olarak *Büyük ihtimal ile gerçekleşme süre tahmini (GS)* aşağıdaki gibi bulunur.

$$GS = ((1 \times IS) + (4 \times BS) + (1 \times KS)) / 6$$

Burada verilen 1, 4, 1 katsayıları varsayılan olarak kullanılmış ve üç süre tahmin varsayımlının ağırlıklarılmış ortalaması bulunmaktadır. IS, KS ve BS değerlerini tahmin etmek zordur ve deneyim gerektirir. Deneyimli bir proje yöneticişi önceden tamamlamış olduğu projelerden topladığı verileri kullanarak daha doğru tahminler yapacaktır. Buna benzer diğer yaygın görev süresi tahmini teknikleri aşağıda verilmiştir.

*Ayrıştırma:* Proje küçük ve yönetilebilir parçalara ayrıstırıldıktan sonra, daha önceden bitirilmiş olan benzer karmaşıklıkta görevlerin sürelerinden yararlanılarak, görevlerin süreleri tahmin edilir.

*COCOMO (COmputational COst Model-Yapısal Maliyet Modeli):* adından da anlaşılacağı gibi model tabanlı bir tekniktir. Bu tahmin tekniğinde ilk önce, sistemde gerçekleştirilecek olan ekran, raporlama ve bileşen sayıları kullanılarak projenin karmaşıklığını ifade eden nesne puanları tahmin edilir. Daha sonra, önceki projelerde kullanılan nesne puanlarına, üretkenlik oranına ve yeni sistem için hesaplanan nesne puanına dayalı olarak yeni proje ve görevlerin tahmini süreleri oluşturulur. Yaygın kullanımından dolayı daha kapsamlı olan COCOMO II modeli de kullanılmıştır.

*Fonksiyon Puanlama:* Model tabanlı olan bu teknikte geliştirilen projeler sonuç ürününün girdi, çıktı, dosya ve sorgularına bağlı olarak ölçümlenmekte ve fonksiyon puanları bulunmaktadır. Daha sonra benzer projelerde ölçümlenmiş olan fonksiyon puanları kullanılarak yeni projenin süreleri tahmin edilmektedir.

## Çizelgeleme

Bütün görevlerin tahmini süreleri ortaya konulduktan sonra proje zaman çizelgesi oluşturulabilir. Proje zaman çizelgesi sadece görev sürelerine değil aynı zamanda görevler arası bağımlılığa göre oluşturulur. Başka bir ifadeyle bir görevin başlangıç veya bitişti başka görev(ler)in başlangıç ve bitişine bağlı olabilir. Görevler arası bağımlılığın dört türü vardır:

*Bitiş-Başlangıç* (FS-Finish to Start); bir görevin bitisi diğer bir görevin başlangıcını tetikler.

*Başlangıç-Başlangıç* (SS-Start to Start); bir görevin başlangıcı diğer bir görevin başlangıcını tetikler.

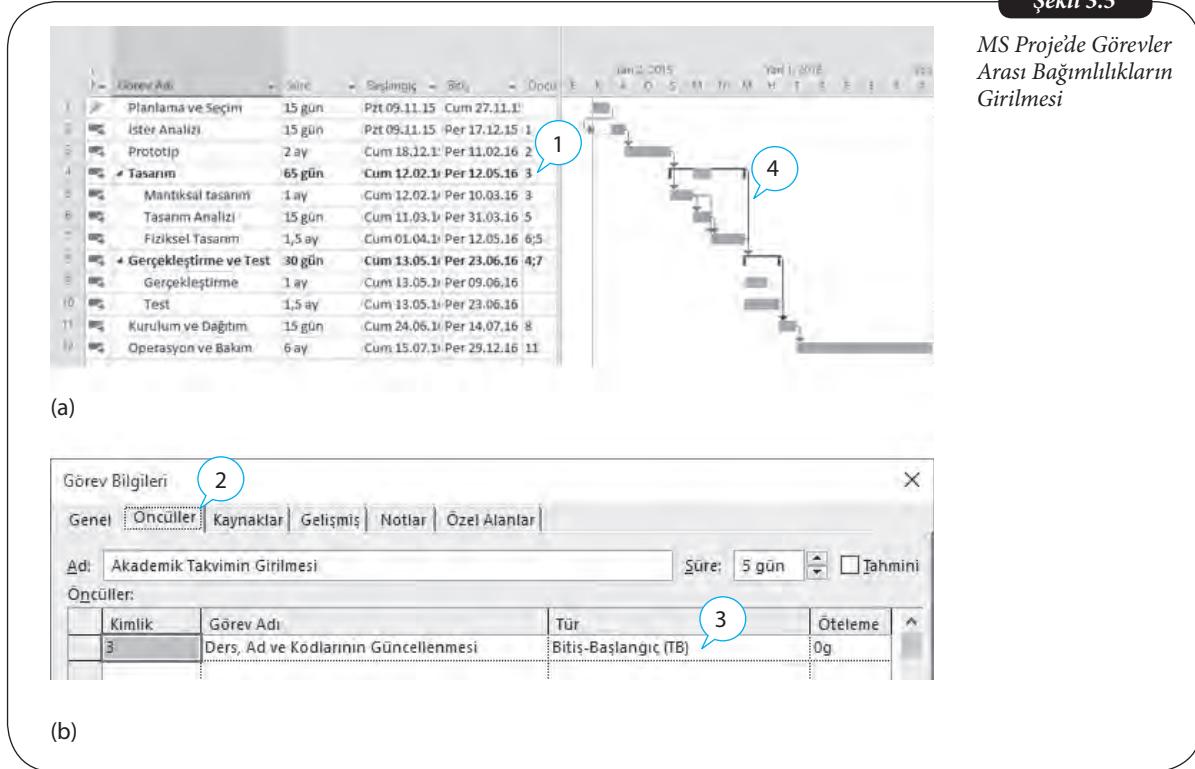
*Bitiş-Bitiş* (FF-Finish to Finish); bir görevin bitisi diğer bir görevin bitisini tetikler.

*Başlangıç-Bitiş* (SF-Start to Finish); bir görevin başlangıcı diğer bir görevin bitisini tetikler.

MS Proje'de görevler arası bağımlılıklar hem Gantt hem de PERT şemalarında gösterilebilir. Şekil 3.5'te MS Proje'de Gantt şemasında görevler arası bağımlılıkların (öncüller) nasıl girildiği gösterilmektedir. Şekil üzerindeki baloncukların gösterdiği öğelerin anımları:

1. Görevler arası bağımlılıklar, Gantt şemasında *Öncüller* sütununa, bağlı olduğu görevin sıra numarası ile girilebilir. Bir görev, aynı zamanda birden fazla görevle bağlı olabilir.
2. Ayrıca görevler arası bağımlılıklar, verilen görev için açılabilen *Görev Bilgileri* penceresinde *Öncüller* sekmesinden girilebilir.
3. Her bir bağımlı görev için bağımlılık türü, *Görev Bilgileri* penceresi *Öncüller* sekmesinden girilebilir.
4. Görevler arası bağımlılıklar, bağımlı görevleri temsil eden çubuklar arasında okla sonlandırılmış çizgiler ile gösterilir. Okların yönü başlangıç ve bitiş görevlerine göre belirlenir.

Şekil 3.5



MS Proje'deki Gantt şemasında tamamlanacak görevler, görevlerin süreleri ve bağımlılıkları tanımlanarak zaman çizelgesi oluşturulur. Projenin başlangıç tarihi verildiğinde de oluşturulan zaman çizelgesi yardımıyla, proje tamamlama tarihi belirlenebilir. Proje zaman çizelgesini kullanarak proje tamamlama tarihini hesaplamak için iki yaklaşım bulunur:

**İleri yönlü zamanlama:** Projenin başlangıç tarihi verildiğinde bu tarihten başlayarak zaman çizelgesi oluşturulur. Gerekli görevlerin planlanan süreleri, aralarındaki bağımlılıklar ve kaynak atamalarına bağlı olarak proje tamamlama tarihi hesaplanabilir.

**Geri yönlü zamanlama:** Projenin tamamlama tarihi verilerek, bu tarihten geriye doğru zaman çizelgesi oluşturulur. Yine görevlerin süreleri, bağımlılıkları ve atanmış kaynakları kullanılarak proje başlangıç tarihi hesaplanır.

MS Proje'de bir görevin başlangıç tarihi *başlangıç* sütununa, bitiş tarihi *bitiş* sütununa ve görevler arası bağımlılıklar *öncüler* sütununa girilir. Gantt şemasında görev çubukları süreleri göstermek üzere uzatılırken başlangıç ve bitişe göre sola ve sağa kaydırılır.

## Organize Etme

Önceki kesimde proje zaman çizelgesinde kullanılacak kaynaklar taslak olarak belirlenmişti. Organize etme faaliyetinde ise projede kullanılacak gerçek kaynakların ataması açıklanacaktır. Kaynaklar aşağıdaki biçimde sınıflandırılır:

*İnsan:* Sistem sahiplerini, kullanıcıları, analistleri, tasarımcıları, geliştiricileri ve proje-de yardım olacak tüm kişileri kapsar.

*Hizmetler:* Dışarıdan alınan kalite kontrol hizmetleri buna örnek olarak verilebilir.

*Tesisler ve cihazlar:* Projeyi tamamlamak için gerekli tüm mekân, teknoloji ve donanı-mi içerir.

*Malzemeler ve araç gereçler:* Kalem, kâğıt vb. kırtasiye ve sarf ürünlerinin tamamını kapsar.

*Para:* Yukarıda verilmiş olan kaynakların tümünün proje bütçesindeki karşılığıdır.

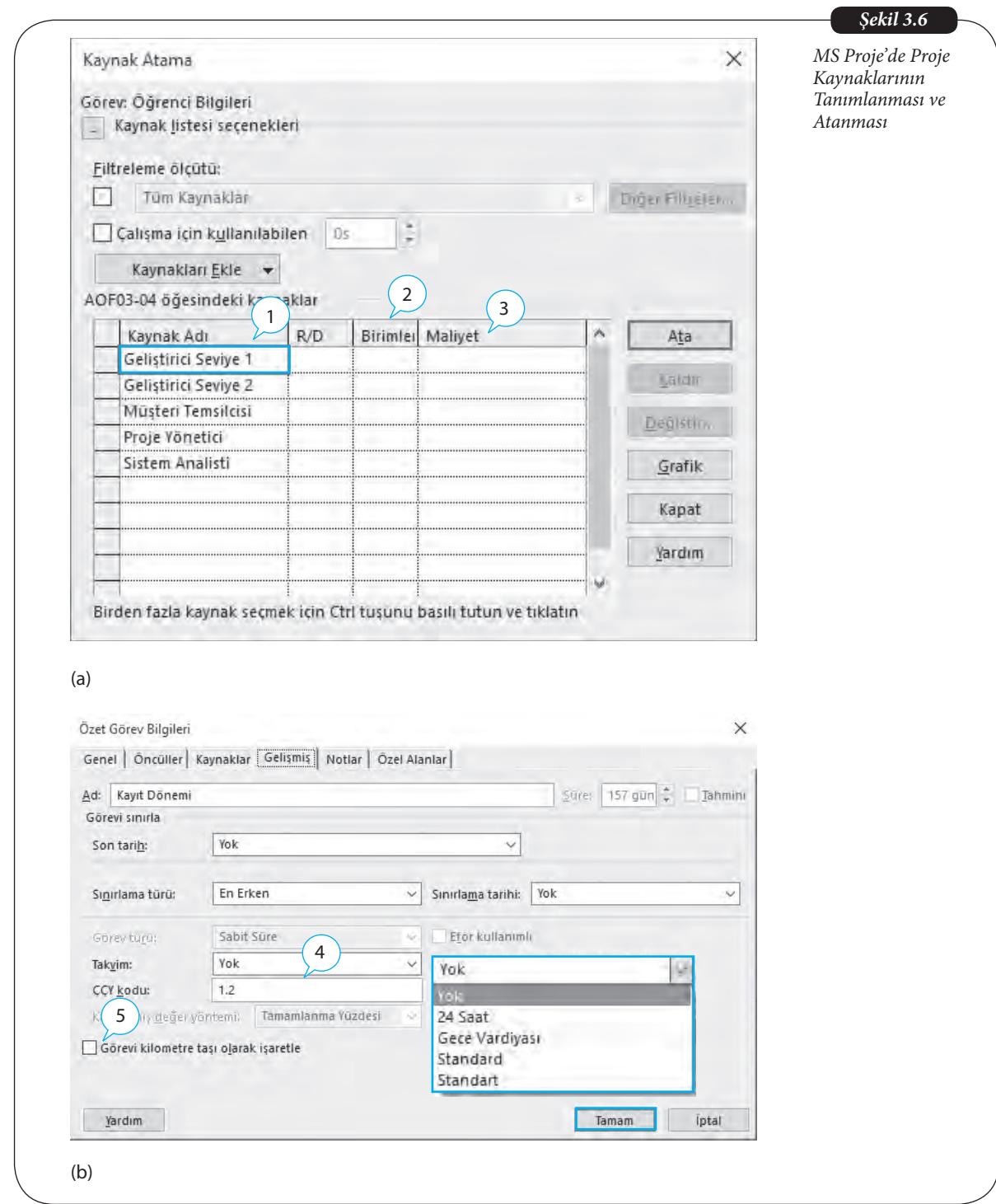
Kaynakların uygunluğu projenin zaman çizelgesini önemli ölçüde değiştirebilecektir.

MS Proje her bir görev için gereken insan kaynağını, roller olarak tanımlar ve kullanır. Proje yönetici ya tanımlı rollere uygun yetenekteki insanları atar ya da uygun yetenekte insan yok ise yönetimden tanımlanmış roller için insan bulmasını/ise almasını talep eder. Yeni elemanların alınması onaylandığında, uygun rollerde görevlendirilmek üzere kaynak ataması tamamlanır.

Şekil 3.6 (a)'dan izlenebileceği gibi, MS Proje *Kaynak Atama* penceresinde roller ve rollere yapılan atamalar belirtilir.

1. Proje yönetici *Kaynak Adı* sütununa kişilerin adlarını ya da unvanlarını girer. Aynı şekilde kaynaklar olarak özel hizmetler, tesisler, cihazlar, malzeme ve araç gereçler tanımlanabilir.
2. *Birimler* sütununa, kaynağı atanmış olduğu projedeki kullanımının yüzdesi girilir. Örneğin uygulama arayüz grafik tasarımcısı projeye yarı zamana (%50 ora-nında) tassis edilebilir. Bir role %100 üzerinde atama yapılrsa, rol için birden fazla kaynak/eleman gereklidir.
3. Uygulama, proje yöneticisinin her bir kaynak için tahmini maliyet atamasına izin verir. Maliyet tahmini firmanın geçmişte yapmış olduğu projelerden, dışarıdan hizmet alımı maliyetlerinden veya işletmenin muhasebe bölümünden alınan bilgi-ler ile yapılır.
4. *Özet Görev Bilgileri* penceresinde her kaynak için kullanım takvimi belirlenebilir (Şekil 3.6 (b)). Bunlar standart, 24 saat, gece vardiyası, hafta içi, hafta sonu biçi-minde olabileceği gibi özel izinler de tanımlanabilir.
5. Görev, kilometre taşı olarak tanımlanacak ise şekil 3.6 (b)'deki *Özet Görev Bilgileri* penceresinde ilgili seçenek işaretlenir.

Şekil 3.6



Kaynaklar tanımlandıktan sonra Şekil 3.6 (a)'da gösterildiği biçimde görevlere atama-  
lar yapılır. Kaynaklar görevlere atanırken proje yöneticisi her bir görev için ilgili kaynağın  
kullanım miktarını belirler. Örneğin bu miktar, verilen görev için bir bireyin harcayacağı  
zamanın yüzdesi olabilir.

Gerçek kaynak atamaları tamamlandıında proje zaman çizelgesi, MS Proje tarafından otomatik olarak ayarlanmış olacaktır. Eğer kaynakların maliyetleri girilmiş ise MS Proje kaynaklara ve zaman çizelgesine göre bütçeyi oluşturacaktır.

## **Yönetme**

Projenin önceki aşamalarında proje için bir ana plan oluşturulmuştur. Bu bölümde ise ana plan çerçevesinde projenin yürütülmlesi ele alınacaktır. Proje yönetimi ağırlıklı olarak insan yönetim faaliyetlerinden oluşur. Bu nedenle yetkin uygulama geliştiricileri proje yönetimine geçtiğinde sıkılıkla başarısız olur. Jerry Weinberg teknik liderlik kitabında liderlik için Motivasyon, Organizasyon, Fikirler ve Yaratıcılıktan oluşan MOI (Motivation, Organization, Ideas and Innovation) modelini önerir:

*Motivasyon:* Proje çalışanlarının tüm yeteneklerini kullanmaları konusunda cesaretlendirilmesidir.

*Organizasyon:* İşletmede kullanılan mevcut süreçleri (ya da yeni süreçleri) başlangıçtaki kapsamdan projenin kapatılmasına kadar uygulama yeteneğidir.

*Fikirler ve Yaratıcılık:* Sistemin geliştirilmesinde çalışanların proje kapsamı içinde kalmak şartıyla yeni fikir ve yaratıcılıklarını kullanımalarının cesaretlendirilmesidir.

McLeod ve Smith, proje ekiplerinin Tablo 3.2'de verildiği gibi olgunlaşma aşamalarından geçtiğini açıklamaktadır.

**Tablo 3.2**  
*Ekip Olgunlaşmasının Aşamaları*

**Kaynak:** McLeod G., Smith D. (1996), s.131.

<b>Biçimlendirme</b>	<b>Oryantasyon Aşaması</b> Yapı ve kuralları oluşturma Ekip üyeleri arasındaki ilişkileri tanımlama Sorumlulukları tanımlama Hedefleri başarmak için plan yapma
<b>Etkileşim</b>	<b>Dâhilî Problem Çözüm Aşaması</b> Kişiler arası çatışmaları çözümleme Kuralları ve hedefleri ayrıntılı açıklama Paylaşım yapılan bir ortam oluşturma
<b>Norm Oluşturma</b>	<b>Büyüme ve Verimlilik Aşaması</b> Hedefler doğrultusunda ekip çalışmasını yönlendirme Ekip üyelerine bilgi verme ve geri bildirim alma Fikirleri paylaşma – gelişen ekip bağlılığı Bireylerin birbirleri ile iyi anlaşması
<b>Uygulama</b>	<b>Değerlendirme ve Kontrol Aşaması</b> Daha fazla geri bildirim ve değerlendirme Ekip normlarına bağlılık Ekibin rollerinin güçlendirilmesi Hedeflerin paylaşımı için güçlü ekip motivasyonu

## **İzleme-Kontrol Etme**

Proje uygulanırken proje yöneticisi, projenin kapsamını, zaman çizelgesi ve verilen bütçe içerisinde yürütüldüğünü izler ve kontrol eder. Proje yöneticisi ilerlemeyi raporlamak ve gerekirse kapsamda, zaman çizelgesinde ve kaynaklarda düzenleme yapmak zorundadır.

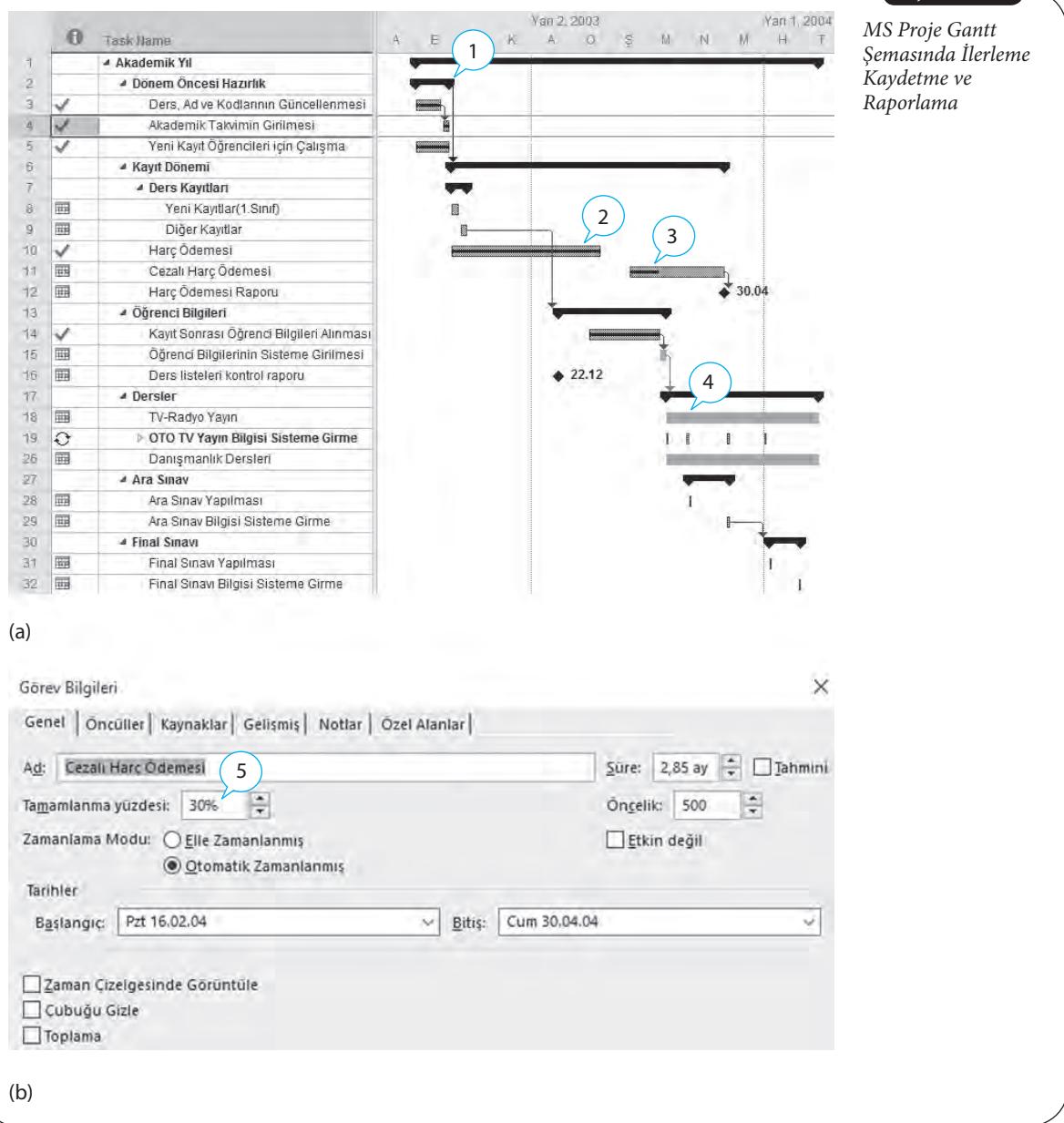
## **İlerleme Raporları**

İlerleme raporları, yeterli sıklıkta oluşturulmalı ancak bu raporlama sıklığı projenin ilerlemesini aksatmayacak biçimde ayarlanmalıdır. Proje ilerleme raporları sözlü veya yazılı olabilir. Bu raporlar doğru ve dürüst biçimde hazırlanmalıdır. Bu sayede problemler büyük sorumlara ve hatta proje sonlandırmamasına sebep olmadan önce belirlenebilir ve çözümleri için çalışma başlatılabilir.

MS Proje'de görevler tamamlandıktan sonra ilerleme durumu kaydedilebilir (Şekil 3.7). Şekil 3.7'deki balonculuklar görevin anlamları aşağıda verilmiştir:

1. Örnek projede ilk fazdaki bütün görevler tamamlandığı için her bir tamamlanan görev çubuğu farklı renkte boyanmıştır.
  2. İkinci fazda ise sadece ilk görev (10 nolu görev) %100 tamamlanmıştır.
  3. İkinci fazdaki 11 no.lu görev %30 oranında tamamlanmış olduğundan sadece görev çubuğu %30 kadarı farklı renkle boyalıdır.
  4. Diğer görevler ise daha başlamadığından planlamada tanımlanlığı gibi durmaktadır.
  5. Herhangi bir görevin ilerlemesi *Görev Bilgileri* penceresinden girilir (Şekil 3.7 b).
- Verilen örnekte proje yöneticisi seçili görevin *Tamamlanma yüzdesini* %30 olarak girmiştir.

Şekil 3.7



MS Proje uygulaması aynı zamanda proje durum bilgisini gösteren birçok hazır ve kişiselleştirilebilen raporlama seçeneği sunar.

### Kapatma

Proje yöneticileri hatalarından ders çıkarmalıdır. Sürekli süreç iyileştirme için hatalar ve başarılar değerlendirilmelidir. Bu faaliyyette, proje ekibinden ve müşterilerden projeyi ilgilendiren konularda deneyimleri ve önerileri alınarak işletmenin proje geliştirme sürecinin iyileştirilmesi amaçlanır. Proje gözden geçirmelerinde bazı temel soruları cevaplamak gereklidir. Bu sorular; “Sonuç (ürün), müşterinin bekleyenlerini karşılıyor mu?”, “Proje zaman çizelgesine uyuldu mu?”, “Proje öngörülen bütçe içinde tamamlandı mı?” biçimindedir. Bu soruların cevaplarına göre, gelecek projelerde kullanılmak üzere sistem geliştirme ve proje yönetim süreçlerinde iyileştirme yapılmalıdır.

SIRA SİZDE



4

Proje yaşam çevrimindeki sekiz önemli faaliyet nedir? Sıralayınız.

## Özet



### *Proje ve proje yönetimi kavramlarını tanımlamak.*

Proje; belirlenmiş zaman, verilen bütçe içinde ve gereksinimlere göre bir amaç/hedef için tamamlanması gereken benzersiz, karmaşık ve bağlı faaliyetler dizisidir. Proje yönetimi; planlama, personel atama, organize etme, yönetme, izleme ve kontrol etme ile verilen zaman aralığında minimum maliyetle kabul edilebilir bir sistemin geliştirilmesi sürecidir. Proje yönetim görevleri ile sistem geliştirme fazları birbirleriyle örtüşür.



### *Proje yöneticilerinin temel işlevlerini tanımlamak.*

Proje yöneticisi, proje hedeflerini sağlamak için gereken organizasyonu yapmak üzere atanır. Bu, önemli yükümlülükler getiren, değişen öncelikler ile çalışan, zorlu, yüksek profilli bir roldür. Bu rol esneklik, iyi karar verme, kuvvetli liderlik, müzakere becerisi ve proje yönetim uygulamaları hakkında önemli bilgiye sahip olmayı gerektirir. Proje yöneticisi projenin ayrıntılarını anlamalı ancak proje yönetimini bütün proje perspektifi üzerinden yapmalıdır.



### *Proje yaşam çevrimi faaliyetlerini açıklamak.*

Proje yaşam çevrimi faaliyetleri:

Kapsam tanımlama: Kapsam projenin sınırlarını belirler ve projede yapılacak işlerin özet olarak verildiği *proje kapsam belgesi* ile sonlanır.

Planlama: Projenin görev ve alt görevlere hiyerarşik olarak ayırtıldığı *iş kırılım yapısı* oluşturulur. Proje de önemli alt kontrol veya ara ürün çıktılarının tarihlerini belirtmek için *kilometre taşı* olarak adlandırılan görevler kullanılır.

Tahminleme: Görev sürelerinin tahmini için birçok yöntem mevcuttur. Deneyimli proje yöneticisi, önceden tamamlanmış olan projelerden elde ettiği veriler ile kaynak atama ve zamanlamada daha doğru tahminler oluşturur.

**Çizelgeleme:** Projenin tanımlanan görevleri arasındaki bağımlılıklar ve tahmin edilen süreleri kullanılarak zaman çizelgesi oluşturulur. Bir görevin başlangıç veya bitişi başka görevlerin başlangıç veya bitişlerine bağlı olabilir. Bu bağımlılıklar projenin tamamlanmasını büyük ölçüde etkiler. *Kritik yol*; projedeki bağımlı görevler dizilerinden birisidir. Projede başlangıçtan bitişe kadar olan tüm yolların süreleri hesaplanır ve bu alternatif yollardan en uzun olanına kritik yol denir. Kritik yol projenin muhtemel en erken bitiş tarihini verir. Kritik yol üzerindeki herhangi bir görevin tamamlanmasındaki gecikme projenin tesliminde gecikmeye sebep olur.

**Organize etme:** Projenin zaman çizelgesinde belirtilen görevleri etkileyen kaynaklar; insan, hizmetler, tesisler, cihazlar, malzemeler, araç gereçler ve paranın uygun biçimde tahsis edilmesini içerir.

**Yönetme:** Ekip çalışmasını yönetmenin en önemli boyutu proje çalışanlarının kontrol ve denetimidir. McLeod ve Smith proje ekiplerinin, *Birimlendirme, Etkileşim, Norm Oluşturma ve Uygulama* - aşamalarından geçtiğini açıklamaktadır.

**İzleme – Kontrol etme:** Proje ilerlerken, proje yöneticisi projeyi kapsam, zaman çizelgesi ve bütçe yönlerinden izlemek zorundadır. Gerektiği durumlarda da kapsamda, zaman çizelgesinde ve kaynaklarda düzenlemeler yapabilir.

**Kapatma:** Proje çalışanlarından ve müşterilerden proje deneyimlerinin ve önerilerinin geri bildirimini içeriir. Bu geri bildirimler ile işletmenin sistem geliştirme sürecinin ve proje yönetim süreçlerinin iyileştirmesi hedeflenir.



### *Proje yönetim yazılımlarını tanıyararak kullanmak.*

PERT şeması, proje görevlerinin ve görevler arası ilişkilerin gösterildiği grafik ağ modelidir.

Gantt şeması, proje görevlerinin yatay çubuklar ile bir takvim üzerinde gösterildiği grafik modelidir.

MS Proje proje yönetim yazılımıdır. Bu uygulama ile projenin Gantt ve PERT şemaları oluşturularak proje üzerindeki ilerleme (zaman çizelgesi, bütçe) izlenebilir.

## Kendimizi Sınavalım

- 1.** Başlangıcı ve bitisi belli olan, verilen bütçe içinde ve istenilen özelliklere göre tamamlanması gereken, bir hedefi veya amacı olan özgün, karmaşık ve birbiriyile bağlantılı faaliyetler bütününe ne ad verilir?
  - a. Sürekli faaliyet
  - b. Üretim
  - c. İş dökümü
  - d. Proje
  - e. Hedef
  
- 2.** Aşağıdakilerden hangisi MS Proje'de Görevler arası bağımlılık türlerinden biri **değildir**?
  - a. Başlangıç-Başlangıç
  - b. Başlangıç-Tamamlanma
  - c. Başlangıç-Bitiş
  - d. Bitiş-Başlangıç
  - e. Bitiş-Bitiş
  
- 3.** İş kırılım yapısı, proje yaşam çevrimi faaliyetlerinden hangisinde oluşturulur?
  - a. Kapsam tanımlama
  - b. Planlama
  - c. Çizelgeleme
  - d. İzleme – Kontrol etme
  - e. Organize etme
  
- 4.** İş kırılım yapısı aşağıdakilerden hangisinde doğru olarak ifade edilmiştir?
  - a. Yapılacak işlerin sıralanmasıdır.
  - b. İşleri kişiler arasında paylaşmaktadır.
  - c. Projeyi yapılabılır küçük işlere ayırtmaktadır.
  - d. Proje içindeki ana faaliyetlerdir.
  - e. Proje yönetim faaliyetleridir.
  
- 5.** Proje ana planı içinde aşağıdaki konulardan hangisi **bulunmaz**?
  - a. Kaynak ihtiyaçları
  - b. İnsan kaynağı
  - c. Amaçlar
  - d. Ödüller
  - e. Potansiyel problemler
  
- 6.** Proje yaşam çevriminde **en fazla** maliyet ve kaynak kullanımı hangi faaliyetlerde yapılır?
  - a. Kapsam tanımlama ve Planlama
  - b. Planlama ve Çizelgeleme
  - c. Çizelgeleme ve Tahminleme
  - d. Yönetme ve İzleme – Kontrol etme
  - e. Organize etme ve Kapatma
  
- 7.** Aşağıdaki ifadelerden hangisi proje zamanlama teknik, yöntem ve yaklaşımlarından biri **değildir**?
  - a. Gantt şeması (Gantt)
  - b. İş kırılım yapısı (İKY)
  - c. Program Değerlendirme ve Gözden Geçirme Tekniği (PERT)
  - d. Kritik Yol Metodu (CPM)
  - e. Grafik Değerlendirme ve Gözden Geçirme Tekniği (GERT)
  
- 8.** Aşağıdakilerden hangisi kritik yol tanımıdır?
  - a. Projenin başlangıcından sonuna kadar giden ve herhangi bir gecikme olduğunda tüm projenin gecikmesine yol açacak olan faaliyetler serisidir
  - b. Projede görevler arasındaki bağlantılardan oluşan en kısa yoldur
  - c. Projenin gerektirdiği görev veya bir grup görevdir
  - d. Projede dikkate değer gelişmeyi gösteren tanımlanabilir ve önemli olaylardır
  - e. Projenin başlangıcından ilk kilometre taşına kadar olan yoldur.
  
- 9.** Aşağıdakilerden hangisi proje yaşam çevriminin son aşamasıdır?
  - a. Çizelgeleme
  - b. İzleme – Kontrol etme
  - c. Kapatma
  - d. Yönetme
  - e. Kapsam tanımlama
  
- 10.** Bir görev %100 verim ve hiç kesme olmadan 100 saatte tamamlanıyorrsa, %50 verimle ve %20 kesme ile çalışan bir işçi için doğru geçen süre tahmini kaç saatdir?
  - a. 40
  - b. 142,9
  - c. 170
  - d. 240
  - e. 250

## Kendimizi Sınavalım Yanıt Anahtarı

- |       |  |
|-------|--|
| 1. d  | Yanıtınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.                  |
| 2. b  | Yanıtınız yanlış ise “Çizelgeleme” konusunu yeniden gözden geçiriniz.            |
| 3. b  | Yanıtınız yanlış ise “Planlama” konusunu yeniden gözden geçiriniz.               |
| 4. c  | Yanıtınız yanlış ise “Planlama” konusunu yeniden gözden geçiriniz.               |
| 5. d  | Yanıtınız yanlış ise “Planlama” konusunu yeniden gözden geçiriniz.               |
| 6. d  | Yanıtınız yanlış ise “Proje Yaşam Çevrimi” konusunu yeniden gözden geçiriniz.    |
| 7. b  | Yanıtınız yanlış ise “Proje Yönetim Araçları” konusunu yeniden gözden geçiriniz. |
| 8. a  | Yanıtınız yanlış ise “Proje Yönetim Araçları” konusunu yeniden gözden geçiriniz. |
| 9. c  | Yanıtınız yanlış ise “Proje Yaşam Çevrimi” konusunu yeniden gözden geçiriniz.    |
| 10. e | Yanıtınız yanlış ise “Tahminleme” konusunu yeniden gözden geçiriniz.             |

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Kritik yol projenin başlangıcından sonuna kadar giden ve herhangi bir gecikme olduğunda tüm projenin gecikmesine yol açacak olan faaliyetler serisidir. Bu kritik faaliyetlerdeki bir gecikme tüm projeyi geciktireceğinden, faaliyetlerin yakından izlenmesi proje için kritik öneme sahiptir.

### Sıra Sizde 2

Gantt şeması, proje yöneticisine, projenin hangi aşamasında hangi faaliyetin yürütülüyor olması gerektiğini gösteren çubuk grafiktir. PERT şeması, planlama aşamasında faaliyetler arasındaki bağlantıları gösteren grafiktir. Gantt şeması gerçek zamanlı faaliyetler arasında karşılaştırma olanağı verir ve eş zamanlı faaliyetleri bulmak için kullanılır. PERT şemasında ise görevler arasındaki bağlantıların bilinmesi ile hangi faaliyetin projeyi daha çok etkileyeceği (kritik yol) ve hangi faaliyetlerde daha esnek olunabileceği görülebilir. PERT daha büyük ve karmaşık projelerde kullanılan bir tekniktir.

### Sıra Sizde 3

Proje çalışma ve çıktı teslimatlarının yönetilebilir alt gruplara ayrıştırılmasına iş kırılımı denir. Planlanan çalışmaların bölündüğü bu alt grplara genellikle İKY'nın çalışma paketi (MS Proj'e özet görev) adı verilir. Her çalışma paketine ilişkin zaman ve kaynaklar ve çalışma paketinin uygulanması sonunda elde edilecek çıktılar belirlenir. Böylelikle projenin diğer faaliyetleri olan projenin yürütülmesi, izlenmesi ve kontrolü süreçlerinde kullanılabilen ayrıntılı bir yol haritası oluşturulmuş olur.

### Sıra Sizde 4

Proje yaşam çevriminin sekiz faaliyeti; kapsam tanımlama, planlama, tahminleme, çizelgeleme, organize etme, yönetme, izleme - kontrol etme ve kapatma biçiminde sıralanır.

## Yararlanılan ve Başvurulabilecek Kaynaklar

Kendall K.E., Kendall J. E., (2006). **Systems Analysis and Design**, 8th Edition, Boston, Pearson Education Inc.

Maylor H., (2003). **Project Management**, 3rd Edition, Pearson Education Ltd. England.

McLeod G., Smith D., (1996). **Managing Information Technology Projects**, Cambridge, MA, Course Technology.

**Proje Yönetimi Kılavuzu (A Guide to Project Management Body of Knowledge)**, (2009). 4. baskı (PMBOK Kılavuzu), Project Management Institute, Newton Square, PA 19073-299 USA.

Uysal O., (2013). **Proje Yönetimi**, (Editör: H. Durucasu), T.C. Anadolu Üniversitesi yayını no: 2980, Eskişehir.

Whitten J. L., Bentley L. D., (2007). **Systems and Design Methods**, 7th Edition, New York, Mc Graw-Hill Inc.

Weinberg G.M., (2014). **Becoming a Technical Leader**, eBook Edition.

# 4

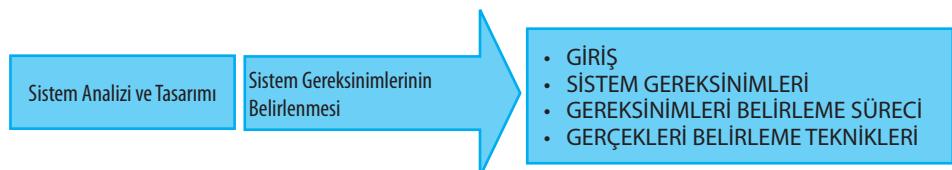
### Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Sistem gereksinimlerine ilişkin temel kavramları açıklayabilecek,
  - 🕒 Gereksinimleri belirleme sürecinin adımlarını açıklayabilecek,
  - 🕒 Gerçekleri tespit etme tekniklerini sıralayabilecek bilgi ve becerilere sahip olabileceksiniz.

### Anahtar Kavramlar

- Gereksinim
- İşlevsel Gereksinim
- İşlevsel Olmayan Gereksinim
- Anket
- Mülakat
- Beyin Fırtınası
- Prototipleme
- Grup Gereksinim Planlaması

### İçindekiler



# Sistem Gereksinimlerinin Belirlenmesi

## GİRİŞ

Tecrübesiz birçok analist problem analizi fazını bitirdikten sonra önemli bir hata yapar ve hemen alternatif çözümlerin ne olacağı konusunu araştırmaya başlar. Yeni geliştirilmiş bilgi sistemlerindeki hatalardan bahseden cümleler genellikle “sistem teknik olarak çok iyi çalışıyor ancak bizim istediklerimizi yapmıyor” biçimindedir. Buradaki problem, sistemin ne yapması gerektiğini açıklayan gereksinimleri belirleme fazının atlanmış olmasıdır. Analist sistemin nasıl geliştirileceği konusundan önce hangi işlevlerin gerçekleştirilmesi gereği ile ilgilenmelidir. Ancak, analistler sıklıkla teknik çözümlerle meşgul olur ve çözüm için gerekli sistem gereksinimlerini eksik veya yanlış tanımlayabilir.

Herhangi bir sistemi geliştirmek için öncelikle kullanıcı gereksinimlerinin ve sistemin sahip olacağı yeteneklerin belirlenmesi, analiz edilmesi ve anlaşılması gerekmektedir.

## SİSTEM GEREKSİNİMLERİ

Sistem gereksinimleri ile bilgi sisteminden beklenen işler veya bilgi sisteminin sahip olması gereken özellik ve nitelikleri belirtilir. Gereksinimler, **İşlevsel gereksinimler** ve **İşlevsel olmayan gereksinimler** olarak iki grupta toplanabilir.

Ünite 1'de örnek olarak verilen hastane bilgi sisteminin işlevsel ve işlevsel olmayan gereksinimlerinden bir kısmı aşağıda verilen biçimde ifade edilebilir:

İşlevsel gereksinimler:

- Randevu almak isteyen hastalar kliniklerdeki boş randevu saatlerini görebilmeli ve randevusunu alabilmeli.
- Klinik sekreterleri her gün randevusu olan hastaları ve hasta bilgilerini görebilmeli.
- Hemşireler tahlil veya röntgen için gelen hastaların hangi tahlilleri yaptıracağını veya röntgeni çektireceğini izleyebilmeli.
- Doktorlar çıkan tahlil sonuçlarını ve röntgen filmlerini bilgisayar ekranında görebilmeli.
- Yazılan reçeteler ve raporlar eczanelerdeki bilgisayarlarda görülebilmeli.

İşlevsel olmayan gereksinimler:

- Hastane bilgi sistemi hafta içi tüm kliniklerde 8:00-17:00 arası çalışıyor olmalıdır.
- Dâhilî kullanıcılar verilen kartlar aracılığıyla kendilerini sisteme tanıtarak sistemi kullanabilmelidir.
- Gün içinde sistemin arızalı olma süresi 5 dakikadan uzun olmamalıdır.

**İşlevsel gereksinimler** bilgi sisteminin yapması gereken işleri tanımlayan gereksinimlerdir.

**İşlevsel olmayan gereksinimler** güvenlik, kolay kullanım veya performans gibi sistemin sahip olacağı özellik ve kaliteyi belirten gereksinimlerdir.

Sistem gereksinimleri, gereksinim çeşidine göre sınıflandırılabilir. Buna göre;

- Performans gereksinimleri
- Bilgi gereksinimleri
- Ekonomi gereksinimleri
- Kontrol ve güvenlik gereksinimleri
- Etkinlik gereksinimleri
- Servis gereksinimleri

birimde bir sınıflama söz konusudur. Her gereksinim sınıfına ilişkin açıklamalar Tablo 4.1'de verilmiştir. Farklı gereksinim çeşitlerinin sınıflandırılmasının faydası, raporlama, izleme ve onaylama için kolaylık sağlamaasıdır. Ayrıca sınıflandırma, gözden kaçan gereksinimlerin belirlenmesine de yardımcı olur.

**Tablo 4.1**  
Sistem Gereksinimlerinin Sınıflandırılması

**Kaynak:** WHITTEN J.L., BENTLEY L.D. (2007), s. 209.

Gereksinim	Açıklama ve Gereksinimleri Belirlemek İçin Sorulacak Sorular
Performans	Performans gereksinimleri kullanıcının ihtiyaçlarını karşılamak için gerekli sistem performansını temsil eder. <ul style="list-style-type: none"> <li>• Kabul edilebilir çıktı hızı nedir?</li> <li>• Kabul edilebilir tepki zamanı nedir?</li> </ul>
Bilgi	Bilgi gereksinimleri kullanıcının içerik, zamanlama, doğruluk ve biçimle ilgili isteklerini temsil eder. <ul style="list-style-type: none"> <li>• Gerekli girdi ve çıktılar nelerdir? Ne zaman oluşturulurlar?</li> <li>• Depolanacak gerekli veri nedir?</li> <li>• Bilgi ne kadar güncel olmalıdır?</li> <li>• Harici sistemlere arayızlar nedir?</li> </ul>
Ekonomi	Ekonomi gereksinimleri, maliyetleri azaltmak ve kârları artırmak gayretlerini temsil eder. <ul style="list-style-type: none"> <li>• Maliyetlerin azaltılacağı sistem alanları nelerdir?</li> <li>• Maliyetler ne kadar azaltılmalı ve kârlar ne kadar arttırılmalıdır?</li> <li>• Bütçe kısıtları nelerdir?</li> <li>• Sistem geliştirme için zaman kısıtları nelerdir?</li> </ul>
Kontrol ve güvenlik	Kontrol ve güvenlik gereksinimleri sistemin çalışacağı ortamı ve sağlanacak güvenlik düzeyini belirler. <ul style="list-style-type: none"> <li>• Sistem ve bilgiye erişim kontrollü olmalı mıdır?</li> <li>• Gizlilik gereksinimleri nedir?</li> <li>• Kritik verilerin yönetimi özel tedbirleri (yedekleme, uzakta depolama vb.) gerektirir mi?</li> </ul>
Etkinlik	Etkinlik gereksinimleri sistemin en az kayıp vererek çıktı üretebilme kabiliyetini gösterir. <ul style="list-style-type: none"> <li>• Elenecek süreçlerde tekrarlanan adımlar var mı?</li> <li>• Sistem kaynaklarının kullanımındaki kayipları azaltacak yollar var mıdır?</li> </ul>
Servis	Servis gereksinimleri sistemin güvenilir, esnek ve genişletilebilir olması için gereken ihtiyaçları temsil eder. <ul style="list-style-type: none"> <li>• Sistemi kimler kullanacak ve bu kişiler nerede bulunuyorlar?</li> <li>• Farklı tipte kullanıcılar olacak mı?</li> <li>• Sisteme eklenecek eğitim cihazları ve materyali nedir?</li> <li>• Sistemin ayrı olarak hangi eğitim cihazları ve materyali (program, veritabanı vb.) geliştirilmelidir?</li> <li>• Güvenilirlik/geçerlik gereksinimleri nedir?</li> <li>• Sistem nasıl paketlenecek ve dağıtılacek?</li> <li>• Nasıl bir belgelendirme gerekiyor?</li> </ul>

Gereksinimlerin belirlenmesinin temel amacı; bilgi sisteminin kullanıcıları için bilgi, süreç ve haberleşme ihtiyaçlarının doğru tanımlanmasıdır. Sistem gereksinimlerinin doğru tespit edilememesi aşağıdakilerden bir veya birkaçıının oluşmasına sebep olabilir:

- Sistem öngörülen maliyetten pahalı olabilir.
- Sistem öngörülen tarihten daha sonra teslim edilebilir.

- Sistem, kullanıcıların bekленmelerini karşılamayabilir ve bunun sonucu sistem satın alınmamayabilir.
- Üretim aşamasında sistemin sürdürülmesi ve iyileştirilmesi çok pahalı olabilir.
- Sistem güvenilmez, hatalara ve bozulmalara açık olabilir.
- Herhangi bir başarısızlıkta bilgi teknolojileri (IT- Information Technologies) çalışanlarının itibarları hatayı kimin yaptığına bakılmaksızın lekelenir.

Tablo 4.2'de bilgi teknolojileri ekonomisi uzmanı Barry W. Boehm'in yazılım geliştirme projelerinde, sistem geliştirme fazında gereksinimlerdeki hataların geç tespit edilmenin maliyetleri üzerine araştırma sonuçları verilmiştir.

Hatanın Tespit Edildiği Faz	Maliyet Oranı
Gereksinimler	1
Tasarım	3-6
Kodlama	10
Geliştirme ve testler	15-40
Kabul testleri	30-70
Operasyon ve bakım	40-1000

**Tablo 4.2**  
Bir Hatayı Düzeltme  
Fazının Göreceli  
Maliyeti

**Kaynak:** WHITTEN  
J.L., BENTLEY L.D.  
(2007), s. 210.

Tablo 4.2'den izlenebileceği gibi tasarım fazında tespit edilen hatalı gereksinimin düzeltme maliyeti, gereksinim fazında tespit edilene göre 3 ila 6 kat daha fazladır. Operasyon ve bakım fazında tespit edilen hatalı gereksinimin düzeltme maliyeti yine gereksinim fazında tespit edilene göre 40 ila 1000 kat daha fazla olabilecektir. Yüksek maliyetlere katlanmamak için sistem gereksinimleri aşağıdaki ölçütlerde uygun olarak tanımlanmalıdır.

**Tutarlılık:** Gereksinimler çelişkili veya belirsiz olmamalıdır.

**Tamlik:** Gereksinimler tüm girdileri ve tepkileri tanımlamalıdır.

**Gerçeklenebilirlik:** Gereksinimler, mevcut kaynaklar ve kısıtlar çerçevesinde gerçekleştirilebilir olmalıdır.

**Gereklilik:** Gereksinimler gerçek ihtiyacı tanımlamalı ve sistemin hedeflerini karşılamalıdır.

**Doğruluk:** Gereksinimler doğru ifade edilmelidir.

**Takip edilebilirlik:** Gereksinimler doğrudan sistemin fonksiyonlarına ve özelliklerine karşılık gelmelidir.

**Onaylanabilirlik:** Gereksinimler testler sırasında gözlemlenebilir olmalıdır.

Gereksinimleri belirleme süreci zaman alıcı, zor ve sinir bozucu olabilir. Bu durum işletmelerin ve kişilerin zamandan ve paradan tasarruf amacıyla genelde kısa yolu tercih etmelerine sebep olur. Ancak bu dar görüşlüük genelde yukarıda bahsedilen problemlere ve hatta projenin başarısızlığına neden olur.

**Sistem gereksinimlerini doğru tanımlamak neden önemlidir? Sistem gereksinimlerini tanımlarken sağlanması gereken ölçütler nelerdir?**



SIRA SİZDE

## GEREKSİNİMLERİ BELİRLEME SÜRECI

Gereksinimleri belirleme süreci aşağıdaki adımlardan oluşur:

- Problemi belirleme ve analiz
- Gereksinimlerin tanımlanması
- Gereksinimlerin analizi ve belgelendirilmesi
- Gereksinimlerin yönetimi

## Problemi Belirleme ve Analiz

Sistem analistinin başarılı olması problem analizi konusunda yetenekli olmasına bağlıdır. Problem analizi konusunu tam olarak anlayabilmek için bir örnek verelim. Bir anne kulak ağrısı, ateş ve burun akıntısı olan kızını doktora götürmüştür. Doktor gerçek problemin ne olduğunu anlamaya çalışmaktadır. Anne ağrıyi azaltmak için kızına ağrı kesici vermiş ancak kızının durumu daha iyi olmamıştır. Görünen, annenin gerçek problem yerine belirtileri tedavi ettiğidir. Doktor durumu daha iyi analiz etmek için eğitimlidir ve çocuğu muayene ettikten sonra kulak iltihabı olduğunu tespit etmiştir. Problem belirlendikten sonra sıra bir tedavi (çözüm) önermeye gelir. Normalde kulak iltihabının tedavisinde bir antibiyotik önerilir. Ancak bu öneriden önce herhangi bir kısıt olup olmadığı belirlenmelidir. Çocuk kaç yaşında ve kaç kilodur? Herhangi bir ilaca karşı alerjisi var mıdır? Hap yutabilir mi? Bu kısıtlar öğrenildikten sonra bir reçete yazılabilir. Sistem analisti problem çözerken doktorun kullandığı sürece benzer bir süreç kullanır, doktordan farklı olarak tıbbi teşhisler yerine sistem problemlerini teşhis etmektedir.

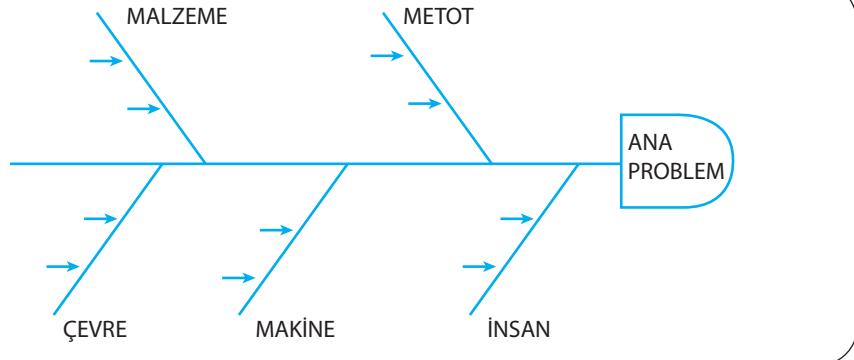
Problemi analiz ederken tecrübesiz sistem analistlerinin yaptığı ortak hatalardan en önemlisi bir belirtiyi problem olarak belirlemeleridir. Bunun sonucu olarak tasarlayıp gerçekleştirecekleri çözüm, gerçek problemi çözemeyecek ve belki de yeni problemlere yol açacaktır. Problemleri belirleme, analiz ve çözümünde yaygın olarak kullanılan bir araç Ishikawa diyagramıdır. Balık kılçığı şeklindeki diyagramın fikirbabası Japonya'da Kawasaki tersanelerinde kalite yönetimi süreçlerini başlatan Kaoru Ishikawa'dır.

Balık kılçığı diyagramı, diyagramın sağına (balık kafası) problemin adını yazarak başlar. Problemin muhtemel sebepleri ana omurgaya bağlı kılçıkların etrafına yazılır ve oklar ana omurgaya doğru çizilir. Kılçıklar genelde malzeme, makine, insan gücü, çevre ve metot olarak adlandırılır. Bu adlandırma, problemin ana nedenlerinin kaynaklarını sınıflandırma amacıyla yapılır. Problemin ana nedeni olarak farklı kategoriler de (yer, süreç, politika vb.) birer ana kılçık olarak diyagrama eklenebilir.

Önemli olan problemin nedenlerini içeren üç ila altı ana kategorinin belirlenmesidir. Ana kılçıklara ilave edilecek alt sebepleri bulabilmek için **beyin fırtınası** teknigi kullanılabılır. Beyin fırtınası tamamlandığında balık kılçığı, ilgili problemin temel ve alt sebeplerini gösteren bir diyagrama dönüştürülür. Daha sonra sistem geliştirme ekibi problemin ana sebeplerini ve buna bağlı alt sebepleri ortadan kaldırmak için nasıl bir yol izleneceğini belirlemek için diyagramı kullanır. Şekil 4.1'de balık kılçığı diyagramının genel gösterimi verilmektedir.

**Şekil 4.1**

Balık Kılçığı  
Diyagramı



**Beyin fırtınası**, katılımcıların kısa bir süre içinde, belirli bir konuda, herhangi bir analiz yapmadan mümkün olduğunda çok fikir üretmeleri için uygulanan ve tüm fikirler söylemeyeceye kadar devam eden bir fikir üretme tekniğidir.

## Gereksinimlerin Tanımlanması

Problemlerin anlaşılmasıından sonra sistem analisti gereksinimleri tanımlamaya başlayabilir. Gereksinimleri tanımlayabilmesi için, sistem analistinin etkili bilgi toplama (gerçeklerin bulunması) teknikleri konusunda bilgili olması gereklidir. **Gerçeklerin bulunması**, sistem geliştirmenin tüm fazlarında kullanılır. Ancak özellikle gereksinimlerin analizi fazında kullanımı çok önemlidir. Gerçeklerin bulunması tamamlandıktan sonra, kullanım durum diyagramları, veri modelleri, süreç modelleri veya nesne modelleri kullanılarak gerçekler belgelendirilir ve sonuçlar çıkarılır.

**Gerçeklerin bulunması;**  
arastırma, toplantılar, mülakatlar, anketler, örnekleme ve diğer teknikleri kullanarak sistem problemleri, gereksinimleri ve seçenekleri konusunda bilgi toplamak için kullanılan bir süreçtir.

## Gereksinimlerin Analizi ve Belgelendirilmesi

Sistem analisti, gerçeklerin bulunması sürecinde toplanan bilgileri düzenli bir biçimde, anlaşılır ve anlamlı olarak bir araya getirmeli ve belgelendirmelidir. Sistem analisti bu belgeleri, gereksinimlerin analizinde ve proje için doğru gereksinimleri tanımlama tekniklerinde girdi olarak kullanır. İzleyen kesimde gereksinimlerin analizi ve belgelendirmesinin alt aşamaları ele alınmıştır.

### Taslak Gereksinimlerin Belgelendirilmesi

Sistem analisti ilk bulgularını belgelendirmek için çeşitli araçlar kullanır. Haricî kullanıcıların bakış açısından ve onların anlayacağı terminolojiyi kullanarak sistem fonksiyonlarını tanımlayan *kullanım durumlarını* yazar. *Kullanım durumlarının* bulunması konusu 5. ünitede ele alınmıştır. İşletmenin karmaşık iş politikalarını ve karar verme kurallarını tanımlamak için ise *Yapısal dil, Karar tabloları* veya *Karar ağaçları* kullanılır. Veri akışlarını belirlemekte kullanılan bu yöntemler 6. Ünite'de ele alınmıştır. Bu aşama sonucunda oluşturulan *Gereksinim tablosu* ise tüm gereksinimleri tanımlayan belgedir.

### Gereksinimlerin Analizi

Birçok durumda, gerçeklerin bulunması faaliyetleri sonucunda birbiriyle çelişen gereksinimler belirlenir. Örneğin, gereksinimlerin farklı kullanıcılarından toplanması durumunda, her kullanıcının yeni sistemin işlevleri ve özellikleri konusunda farklı düşünceleri bulunabilir. Böylece toplanmış olan gereksinimler çelişen istekleri içerebilir. Gereksinim analizinin amacı; gereksinimler arasındaki çelişkileri bulmak, çözümlemek ve paydaşları tatmin edecek şekilde uyarlayarak bir uzlaşmaya varmaktr. Bu aşamada “proje için doğru sistem gereksinimlerine ulaştık mı?” sorusu sorulmalıdır. Taslak belge eksik, çelişkili, geçersiz, çakışan ve belirsiz gereksinimler içerebilir. Gereksinimler arasındaki bu çelişkilerin çözülmemesi ileride maliyeti yüksek çözümlere yol açabilir. Paydaşların sistem gereksinimleri konusunda anlaşmaları için bir müzakere süreci kaçınılmazdır. Bu süreç sistem analistinin yönlendirmesi altında gerçekleştirilek sistem gereksinimleri düzenlenir ve basitleştirilir.

### Gereksinimlerin Resmîleştirilmesi

Sistem gereksinimleri, temel paydaşlarla paylaşmak amacıyla resmî olarak belgelendirilir. Bu belge (**gereksinim tanımları belgesi**), sistem sahipleri ile sistem geliştirme ekibi arasında, yeni sistemde nelerin olacağı konusunu içeren bir kontrat olarak kullanılır. Bu nedenle üzerinde anlaşmaya varılmadan önce birçok gözden geçirme ve düzeltmeden geçer. Gereksinim tanımları belgesi aşağıdaki özelliklerini içermelidir:

**Gereksinim tanımları belgesi,**  
sistem projesi için bir kontrat niteliğinde olan ve önerilen projenin gereksinimlerini temel paydaşlara bildiren resmî belgedir. Bilgi sistemi şartnamesi olarak da adlandırılır.

- Sistemin sağlayacağı fonksiyonlar ve hizmetleri içeren işlevsel gereksinimler.
- Sistem özelliklerini, karakteristiklerini ve niteliklerini içeren işlevsel olmayan gereksinimler.

- Sistemin geliştirilmesini sınırlayan ve sistemin çalışacağı ortamın kısıtlarını içeren gereksinimler.
- Sistemin bir arayüze bağlanacağı diğer sistemler hakkında bilgileri tanımlayan gereksinimler.

İNTERNET



<http://www.saglik.gov.tr/TR/dosya/1-53183/h/web-sayfasindan-ornek-bir-Hastane-Bilgi-Sistemleri-Alimi-Cerceve-Ilkelerini-iceren-gereksinim-tanimlari-belgesine-erisilebilir>.

Sistem modelindeki hataları, yazım veya dil bilgisi hatalarını ve çelişen gereksinimleri düzeltmek için son taslak üzerinde gereksinim onaylama işlemi yapılmalıdır.

### Gereksinimlerin Yönetimi

Projenin yaşam çevrimi içinde yeni gereksinimler ortaya çıkabilir veya gereksinimlerde değişiklik yapılması gerekebilir. Bazı çalışmalar, sistem üretime başlamadan önce belirlenen gereksinimlerin %50 veya daha fazlasının değiştigini ortaya koymuştur. Değişimin getireceği problemleri hafifletmek için gereksinimlerin yönetimi süreci kullanılmalıdır. Bu süreç bir gereksinimde yapılacak değişiklikte izlenecek politikaları ve yöntemleri içerir.

### GERÇEKLERİ BELİRLEME TEKNİKLERİ

Bu bölümde yaygın olarak kullanılan *gerçekleri belirleme teknikleri* ele alınacaktır. Söz konusu bu teknikler:

- Mevcut belge, form ve dosyaların örnekleşmesi
- Araştırma ve yerinde ziyaret
- İş ortamının gözlemlenmesi
- Anketler
- Mülakatlar
- Prototipleme
- Ortak gereksinim planlaşmasıdır.

Analist, bilgi sistemi projesi süresince bu tekniklerden birkaç tanesini birlikte kullanabilir. İlgili projede kullanılacak en uygun tekniği seçebilmesi için sistem analistinin her tekniğin avantaj ve dezavantajları hakkında bilgi sahibi olması gereklidir.

### Mevcut Belge, Form ve Dosyaların Örnekleşmesi

Bilgi sisteminin kullanılacağı işletme hakkında bilgi edinmek için bakılabilecek ilk belge organizasyon şemasıdır. Organizasyon şeması, projenin sahipleri ve kullanıcıları arasındaki ilişkileri tanımlamakta faydalı olur. Analist aynı zamanda projenin geçmişini de öğrenmek isteyecektir. Bu konuda incelenecek belgeler:

- Ofis içi yazışmalar, öneri kutusu notları, problem konusunda müşteri şikayetleri ve raporları
- Muhasebe kayıtları, performans ve çalışma özetleri, bilgi sistemleri proje istekleri
- Problemi tanımlayan belgelere ilave olarak işletmeyle ilgili aşağıdaki belgeler de incelenebilir.
  - İşletmenin misyonu ve stratejik planı
  - Projede üzerinde çalışılan altbirimlerin resmî hedefleri
  - Önerilen herhangi bir bilgi sistemi üzerine yazılan politika el kitapları
  - Günlük işler için standart çalışma prosedürleri, iş taslakları ve talimatları
  - İşlem döngüsünün çeşitli noktalarında yapılan işlemleri gösteren formlar
  - Bilgisayar veritabanlarının, ekran ve raporlarının örnekleri

Ayrıca, önceki sistem analistleri ve danışmanları tarafından yapılan sistem çalışmaları ve tasarımları hakkındaki belgeler de kontrol edilmelidir. Bu belgeler aşağıda sıralanmıştır.

- Farklı tipte akış diyagramlarını gösteren belgeler
- Proje sözlüklerini, giriş, çıkış ve veritabanlarını gösteren tasarım belgeleri
- Program belgesi, bilgisayar işlemlerinin ve eğitimlerinin el kitapları

Toplanan tüm belgeler güncel olup olmadığı konusunda analiz edilmelidir. Toplanan bilgileri onaylamak veya güncellemek için gerekli olabileceklerinden eski tarihli belgeler de gözardı edilmemelidir. Analist söz konusu bu belgelerin içinden aşağıdaki bilgileri toplayabilir:

- Problemin belirtileri ve sebepleri
- İşletmede problemi bilen kişiler
- Mevcut sistemi destekleyen iş fonksiyonları
- Sistem tarafından toplanması ve bildirilmesi gereken veri tipleri
- Analistin belgelerden anlayamadığı ve mülakatlarda belirlenebilecek noktalar

Her formu, bir dosyadaki veya veritabanındaki her kaydı incelemek pratik olmadığı için, analist örneklemeye tekniklerini kullanarak sistemde olan bitenleri anlamak üzere yeterli sayıda veriyi inceler. Örnek sayısının tüm kayıt ya da belgeleri temsil edip etmediği, istatistiksel örneklemeye tekniklerini kullanarak belirlenebilir. Örneklemeye konusunda daha bilgili olabilmek için istatistik dersinin alınması önemlidir.

**Bilgi sisteminin durumunu belirlemek için analist ne tür belgeleri incelemelidir?**



SIRA SİZDE  
2

### Araştırma ve Yerinde Ziyaret

Problemlerin birçoğu sadece ilgili işletmeye özgü değildir ve benzer diğer işletmeler ilgilenen problemi daha önceden çözmüş olabilir. Bu nedenle kabul edildiği takdirde, benzer problemi çözen işletmeyi yerinde ziyaret ederek çözümleri konusunda bilgi alınabilir. Bu şekilde kazanılan bilgi, bilgi sisteminin geliştirme süresi ve maliyetinde büyük oranda düşüş sağlayacaktır. İnternet üzerinde yapılacak araştırmada benzer problemlerin başkaları tarafından nasıl çözüldüğü konusunda önemli bilgiler verebilir.

### İş Ortamının Gözlemlenmesi

Gözlem bir sistem hakkında veri toplama için en etkili tekniklerden biridir. Analist sistem hakkında bilgi edinmek için insanları ve faaliyetlerini gözlemler. Bu teknik genelde diğer yöntemlerle toplanmış ancak soru işaretleri olan verilerin onaylanması veya kullanıcıların sistem hakkında kesin bir açıklama yapmadıkları durumlarda kullanılmaktadır.

Sistem analisti gözlem yaparak gerçekleri nasıl tespit edebilir? Doğrudan gözlem bölgесine gidip gördüklerini kayıt altına almalı mıdır? Bu sorunun cevabı “hayır” olacaktır. İş ortamının gözlemlenmesinden önce birçok hazırlık yapılmalıdır. Bu amaçla;

- Verileri hızlıca kaydetmek için özel bir form gereklidir?
- Kişiler, yaptıklarının gözlemlenmesi ve kayıt altına alınmasından rahatsız olur mu?
- Gözlemlenecek işin yoğunluğu hangi zaman aralıklarında az, normal veya fazladır?

sorularına cevap arayan sistem analisti gözlem yapacağı ideal zamanı belirlemelidir. Analist gözlemlerini normal iş yükü koşullarında yapmalı, daha sonra yoğun zamanda da gözlem yaparak iş hacmindeki artışın etkilerini ölçmelidir. İyi bir gözlemin nasıl yapılması gereği aşağıda maddeler hâlinde verilmiştir.

- Kimin, neyin, nerede, ne zaman, neden ve nasıl gözlemleneceği belirlenmelidir.
- Gözlem için uygun şef ve yöneticilerden izin alınmalıdır.
- Kimlerin hangi amaçla gözlemleneceği bildirilmelidir.
- Gözlem sırasında sessiz kalınmalıdır.

- Gözlem sırasında veya hemen sonrasında notlar alınmalıdır.
- Gözlem notları uygun kişilerle gözden geçirilmelidir.
- Kişiler iş başında rahatsız edilmemelidir.
- Basit işlemler üzerine odaklanılmamalıdır.
- Ön kabullen yapılmamalıdır.

## Anketler

Diğer bir “gerçekleri belirleme” tekniği de anketler aracılığıyla araştırma yapmaktadır. Anketler aracılığıyla çok sayıda kişiden veri toplanabilir. Anketlerin avantaj ve dezavantajları aşağıda belirtilmiştir:

### Avantajları

- Anketler hızlı bir şekilde cevaplanabilir.
- Göreceli ucuz olarak çok sayıda kişiden veri toplama tekniğidir.
- Anketler cevaplayan kişilerin kimliklerinin gizli kalmasına izin verir. Bu nedenle kişiler, patronların verilmesini istediği cevaplar yerine kendi düşüncelerini anlatan cevaplar verebilir.
- Cevaplar hızlı bir şekilde tablo hâline getirilip analiz edilebilir.

### Dezavantajları

- Cevap veren kişi sayısı genelde düşüktür.
- Bir kişinin tüm sorulara cevap vermesi garanti edilemez.
- Anketler genelde esnek değildir. Analistin kişilerden gönüllü bilgi edinmesi veya yanlış yorumlanan soruları yeniden yazması mümkün değildir.
- Sistem analistinin anketi cevaplayan kişinin vücut dilini görmesi ve analiz etmesi mümkün değildir.
- Bulanık veya tam olmayan soruları hemen açıklığa kavuşturma imkânı yoktur.
- İyi anket hazırlamak zordur.

## Anket Tipleri

Serbest format anketi ve sabit format anketi olmak üzere iki tip anket bulunmaktadır. Serbest format anketleri kullanıcıya soruları cevaplamakta daha fazla serbestlik verir. Aşağıda iki serbest format sorusu verilmiştir.

- Bu günlerde ne tür raporlar alıyorsun ve bu raporlar nasıl kullanılıyor?
- Bu raporlarla ilgili bir problem var mı (yanlış mı, yetersiz bilgi var mı, okuması ve kullanılması zor mu)? Varsa lütfen açıklayın.

Bu tür soruların cevaplarını tablo hâline getirmek zordur. Cevaplar sorulan sorularla uyuşmayabilir. Serbest format anketlerinde faydalı cevap almayı garantilemek için, analist soruları basit cümlelerle kurmalı ve kişiler tarafından farklı değerlendirilecek kelimeleri (örneğin iyi, güzel vb.) kullanmaktan kaçınmalıdır. Analist soruları üç veya daha az cümleyle cevaplanacak şekilde hazırlamalıdır. Aksi hâlde anket, cevaplananın ayırmak istediği süreden daha uzun zaman alabilir.

Sabit formatlı anketler kullanıcının önceden hazırlanmış cevap şıkları arasından seçim yapmasına izin verir. Bu, sonuçların daha kolay tablo hâline getirilmesini sağlar. Ancak anketi cevaplayandan faydalı olabilecek ilave bilgilerin elde edilmesini sağlayamaz.

Sabit formatlı soru biçimlerinden en yaygın kullanılan üç tanesi aşağıda açıklanmıştır.



*Çoktan seçmeli sorular:* Anketi cevaplayana seçme yapabileceği birçok seçenek verilir. Cevaplayana birden fazla cevap işaretleyip işaretleyemeyeceği söylenmelidir. Bazı çoktan seçmeli sorular çok kısa serbest format soruları da içerebilir. Aşağıda çoktan seçmeli sabit formatlı sorulara örnekler verilmiştir:

Geri dönen siparişler çok sık mı oluyor?

Evet  Hayır

Güncel alacak hesapları raporu faydalı mı?

Evet  Hayır

Cevabınız hayırsa lütfen açıklayınız.

.....

*Derecelendirme soruları:* Cevaplayana, verilen bir cümle hakkındaki görüşünü belirteceği seçenekler verilir. Yanlış cevap verilmesini engellemek için olumlu ve olumsuz cevaplar eşit sayıda olmalıdır. Aşağıda bir örnek verilmiştir:

Miktar indirimi uygulaması müşteri siparişlerinde artışa sebep olabilir

- Kuvvetle hemfikirim  
 Hemfikirim  
 Fikrim yok  
 Hemfikir değilim  
 Kuvvetle hemfikir değilim

*Sıralama soruları:* Cevaplayıcıya tercih veya tecrübeye bağlı olarak sıralayacağı birkaç muhtemel cevap verilir. Sıralama sorusu içeren sabit formatlı bir örnek aşağıdadır:

Aşağıdaki işlemleri harcadığınız zamana göre sıralayınız

- \_\_\_\_\_ Yeni müşteri siparişleri  
\_\_\_\_\_ Sipariş iptalleri  
\_\_\_\_\_ Sipariş değişiklikleri  
\_\_\_\_\_ Ödemeler

### Anketin Hazırlanması

İyi bir anket hazırlamak zordur. Aşağıdaki prosedür etkin bir anket hazırlamak için faydalı olabilir:

- i. Hangi gerçeklerin ve fikirlerin toplanacağına ve bunların kimlerden toplanacağına karar verilmelidir. Kişi sayısı çoksa, rastgele seçilmiş az sayıda kişiden oluşan bir cevaplayan grubu belirlenmelidir.
- ii. Toplanmak istenen gerçekler ve fikirlere bağlı olarak en iyi cevapları üretecek serbest veya sabit formatlı anket kararı verilir. Bazı durumlarda sabit formatlı cevaplarda isteğe bağlı serbest açıklama soruları eklenebilir.
- iii. Sorular yazılır. Sorular anlam hataları veya yanlış yorumlamalar açısından gözden geçirilir. Soruların, anket hazırlayıcının kişisel görüşlerini yansitmamasına dikkat edilmelidir.
- iv. Sorular, az sayıda cevaplayana uygulanır. Bu cevaplayanların sorularla ilgili problemi olursa sorular düzelttilir.
- v. Anket çoğaltılar ve dağıtılar.

## Mülakatlar

Kişisel mülakatlar en önemli ve en çok başvurulan gerçekleri tespit etme tekniğidir. Mülakatlar, doğrudan ve yüz yüze temasla gerçeklerin bulunması, onaylanması, anlaşılması, gereksinimlerin belirlenmesi, fikir ve önerilerin sorulması amacıyla gerçekleştirilir. Bir mülakatta iki rol bulunmaktadır. Sistem analisti, mülakatı düzenlemekten ve gerçekleştirmekten sorumlu olan *mülakat yapan* rolündedir. Sistem kullanıcısı veya sahibi ise sorulan bir dizi soruya cevap veren *mülakat veren* kişidir. Mülakatta birden fazla mülakat yapan ve/veya mülakat veren kişi olabilir. Yani mülakatlar bire bir veya çoklu olarak gerçekleşebilir.

Herkesin değerleri, öncelikleri, fikirleri, motivasyonu ve kişiliği farklı olduğu için mülakatı yapan sistem analistinin iyi insan ilişkilerine sahip olması gereklidir. Diğer tekniklerde olduğu gibi mülakatta her durumda kullanılacak en iyi teknik değildir. Mülakatların da avantaj ve dezavantajları vardır.

## Avantajlar

- Mülakat, mülakat verenin soruları açık ve serbest şekilde cevaplamasına imkân verir.
- Analistin, mülakat verenden daha fazla geri besleme olmasını sağlar.
- Mülakatlar, sistem analistinin soruları kişiye göre değiştirmesine izin verir.
- Mülakatlar analiste, mülakat verenin sözsüz iletişimini gözlemesine imkân oluşturur. Başarılı bir sistem analisti, mülakat verenin sorularına verdiği cevapları dinlerken vücut hareketleri ve yüz ifadelerinden bilgi edinebilir.

## Dezavantajlar

- Mülakatlar çok zaman alıcıdır. Bu nedenle pahalı bir gerçekleri bulma tekniğidir.
- Mülakatin başarısı, mülakat yapanın insan ilişkileri yeteneğine bağlıdır.
- Mülakatlar, mülakat verenlerin bulunduğu yer nedeniyle pratik olmayabilir.

Yapışal ve yapışal olmayan olmak üzere iki çeşit mülakat vardır. Yapışal olmayan mülakatta konuşmayı mülakat verenin yönlendirmesine izin verilir. Bu tür mülakatta “Toplanamayan alacaklar raporunu neden beğenmiyorsun?” gibi **çıkıcı sorular** sorulur. Bazen konuşmalar yörüngeinden çıkabilir, bu durumda sistem analistinin müdahale ederek mülakatı ana hedefe ve konuya yönlendirmesi gereklidir. Yapışal mülakatlarda kısa ve doğrudan cevap alınması beklenen **kapalı çıkışlı sorular** sorulur. Bu tür sorulara örnek olarak “Toplanamayan alacaklar raporunu zamanında alabiliyor musun?” ve “Toplanamayan alacaklar raporu doğru bilgiler içeriyor mu?” verilebilir. Sorular dikkatle seçilmeli ve ifade edilmelidir. Soruların çoğunluğu kim, ne, ne zaman, nerede, niçin ve ne kadar gibi standart kelimelerle başlar. Aşağıdaki tip sorulardan kaçınılmalıdır:

- “Raporda sütunların her ikisini de bulundurmak zorunda mıyız?” gibi (anlam, duyu, vurgu) **yüklü sorular**. Soruda mülakat yapan kişinin belli seçeneklere kendiliğinden çekici ya da itici özellik kazandırma eğilimi vardır.
- “Bu operatör kodunu kullanmayacaksın, değil mi?” gibi **yönlendirici sorular**. Soru mülakat vereni gerçek fikri ne olursa olsun “Elbette hayır” cevabını vermeye yönlendirmektedir.
- “Envanter dosyasında gıda sınıflandırması için ne kadar kod ihtiyacımız var? Ben 20 yeterlidir diye düşünüyorum” gibi **yanlış sorular**. Yanlış sorular mülakat verenin cevabını etkileyecektir.

Üçüncü tip soru şekli **soruşturma** sorusudur. En kuvvetli ve basit soruşturma sorusu “Neden?” dir. Bir başka soruşturma sorusu örneği “Çevrimiçi fatura ödeme sisteminizde karşılaşığınız güvenlik problemlerini örnekler vererek açıklar misiniz?” olabilir. Soruşturma sorusunun amacı daha fazla bilgi edinmek için başlangıç cevabının ötesine geçmektir. Bu sorular açık çıkışlı veya kapalı çıkışlı olabilir.

**Açık çıkışlı soru**, mülakat verenin uygun gördüğü şekilde soruları cevaplamasına izin veren sorudur.

**Kapalı çıkışlı soru**, cevapları belli seçeneklerle veya kısa ve net sorularla sınırlı sorudur.

Soruların sıralanmasında tümevarım ve tümdengelim biçiminde iki yol izlenebilir. Tümevarım sıralama, piramit yapıda düşünülebilir. Buna göre önce ayrıntılı ve kapsalı sorularla başlanır, sonra daha genel cevapların alındığı açık uçlu sorulara yer verilir. Piramit yapısı mülakat vereni konuya ısnadırmak amacıyla kullanılabilir. Tümdengelim yaklaşımında ise açık uçlu, genel cevaplı sorularla başlayıp kapalı sorularla devam edilir. Bu yaklaşım mülakata kolay şekilde başlanması sağlar. Tümdengelim yaklaşımı mülakat verenin konu hakkında istekli olması ve fikirlerini serbestçe ifade etmek istemesi durumunda da kullanılmalıdır.

## Prototipleme

Bir başka gerçekleri bulma tekniği prototiplemedir. Prototipleme genelde bir tasarım teknigidir, ancak bu yaklaşım sistem geliştirme yaşam döngüsünün ilk dönemlerinde gerçeklerin bulunması ve gereksinimlerin analizi amacıyla da kullanılabilir. Bu amaçla bir prototip oluşturulması işlemi **keşif prototiplemesi** olarak adlandırılır.

Keşif prototipleme, sistem geliştirme ekibinin sistem gereksinimlerini belirlemeye zorlandığı durumlarda uygulanır. Bu yaklaşımın felsefesi, kullanıcıların prototipi gördüklerinde gereksinimleri keşfedeceğidir. Prototipin hızla geliştirilmesi, daha sonra geliştirme fazında kullanılabilmesi açısından önemlidir. Genelde sadece gereksinimlerin belirlenemediği alanlarda prototipleme yapılır, bu nedenle birçok istenen işlev ve kalite garantisini ihmali edilebilir. Prototiplemede performans ve güvenilirlik gibi işlevsel olmayan gereksinimlere son ürüne göre daha az önem verilir. Prototip üretimi, sistem geliştiricilerinin ve kullanıcılarının sistemle ilgili konuları daha iyi anlamalarına yardımcı olur. Bu teknik, kullanıcı ihtiyaçlarını veya teknik gereksinimleri karşılamayan bir sistemin piyasaya sürülmESİ riskini azaltır. Prototipleme metodunun avantaj ve dezavantajları aşağıda verilmiştir.

**Keşif prototiplemesi**  
kullanıcının gereksinimlerini  
belirlemek ve doğrulamak için bu  
gereksinimlerin ufak ölçekli bir  
örneğini veya çalışan modelini  
kurma eylemidir.

### Avantajlar

- Sistem kullanıcılarının ve sistem geliştiricilerinin yazılımı denemesine ve sistemin nasıl çalıştığı hakkında bilgi edinmesine izin verir.
- Yüksek geliştirme masrafları olmadan sistemin yapılabırlığının ve faydasının anlaşılmasına yardımcı olur.
- Kullanıcılar için eğitim aracı olarak kullanılabilir.
- Sistemin test sürecinde kullanılacak test planları ve senaryolarının geliştirilmesine yardımcı olur.
- Gerçekleri bulma sürecini kısaltabilir, daha kararlı ve güvenilir gereksinimlerin tanımlamasına yardımcı olur.

### Dezavantajlar

- Geliştiriciler, prototipleme konusunda eğitilmelidir.
- Kullanıcılar prototipin performansı, güvenilirliği ve özelliklerine bakarak gerçekçi olmayan gereksinimler geliştirebilir. Prototipler sadece sistemin işlevsellliğini taklit eder ve doğal olarak tam değildir. Kullanıcılar bu konuda eğitilmeli ve yanlış yönelmeleri önlenebilir.
- Prototip yapımı sistem geliştirme sürecini uzatabilir ve geliştirme maliyetlerini artırabilir.

## Ortak Gereksinim Planlaması

Birçok işletme ayrı ve sayılı mülakatlar yerine grup çalışma oturumlarını kullanmaktadır. Grup çalışma oturumu uygulamalarının bir örneği de ortak gereksinim planlamasıdır. Problemlerin tanımlanması ve analizi ile gereksinimlerin belirlenmesi amacıyla yapılan-

dırılmış grup toplantıları düzenlenir. Bu ve benzeri teknikler ayrıntılı eğitim süreci gerektirmekle birlikte gerçekleri bulma sürecinin süresini büyük oranda azaltır.

Bu teknik geniş bir aralıktaki katılımcı ve rol gerektirir. Her katılımcının tüm oturumlarla katılımı ve aktif olması beklenir. Aşağıda ortak gereksinim planlaması tekniğinin katılımcıları incelemiştir.

### **Sponsor**

Genelde işletmenin üst yönetimindeki bir kişidir. Sistem projelerindeki tüm bölümleri ve kullanıcıları bilen birisidir. Sponsor, muhtemel kullanıcıların oturumlara isteyerek ve aktif katılımlarını teşvik ederek projeye destek verir. Sponsor, projenin devam edip etmeyeccione karar veren kişidir. Katılımcıları tanıtarak toplantıının başlangıcını yapar, ayrıca toplantıının kapanış konuşmasını da gerçekleştirir. Aynı zamanda oturumlara kullanıcılarından kimlerin katılacağını, toplantı zamanını ve yerini belirler.

### **Kolaylaştırıcı**

Toplantılarda lider veya kolaylaştırıcı rolünü üstlenen bir kişi olmalıdır. Kolaylaştırıcı, bir sistem projesi için yapılan tüm toplantıları yönlendirir. Kolaylaştırıcı, çok iyi iletişim kurabilen, pazarlık yapabilen ve grup tartışmalarını çözebilen biri olmalıdır. İş konusunda bilgisi yeterli, organizasyon yetisi yüksek ve verilecek kararlara karşı tarafsız olmalıdır.

Bazen bu özelliklere sahip bir kişiyi işletme içinden bulmak zor olabilir. Bu durumda işletme, geniş bir ortak gereksinim planlaması eğitimi düzenlemeli veya işletme dışından bu rolü gerçekleştirecek bir kişiyi bulmalıdır.

Kolaylaştırıcının görevi, toplantıyı planlamak, toplantıyı yönetmek ve sonuçlarını takip etmektir. Toplantı sırasında tartışmayı yönetir ve katılımcıların aktif olmasını teşvik eder. Çıkan tartışmaları çözümlemekten ve toplantıının amaç ve hedeflerinin gerçekleşmesini garanti etmekten sorumludur. Toplantıda uyulacak kuralları koymak ve katılımcıların bu kurallara uymalarını sağlamak kolaylaştırıcının sorumluluğudur.

### **Kullanıcı ve Yöneticiler**

İşletmenin farklı birimlerindeki kullanıcı ve yöneticiler günlük işlerinden izinli olarak ortak gereksinim planlama toplantılarına katılır. Bu katılımcıları genellikle sponsor belirler ve her katılımcının toplantılarında katkıda bulunacak yeterli iş bilgisine sahip olmasına dikkat edilir.

Kullanıcıların toplantıda rolü, iş kuralları ve gereksinimlerini etkili olarak anlatmak, prototip tasarımları gözden geçirmek ve kabul kararı vermektedir. Yöneticilerin görevi ise proje hedeflerini onaylamak, proje önceliklerini belirlemek, maliyet ve süreleri onaylamak ve belirlenen eğitim ihtiyaçları ve uygulama planlarını onaylamaktır.

### **Yazıcılar**

Toplantıda tartışılan her konuya ilgili kayıtları tutan kişilerdir. Tutulan kayıtlar, toplantıdan hemen sonra basılmalı ve katılımcılara dağıtılmalıdır. Yazıcılar, sistem analizi ve tasarımında CASE araçlarını kullanma kabiliyetine sahip olmalıdır. Genellikle sistem analistleri bu görevi üstlenir.

### **Bilgi Teknolojileri Çalışanları**

Toplantılara, kullanıcılar ve yöneticiler tarafından dile getirilen konu ve gereksinimleri dinleyip not alan bilgi teknolojileri personeli de katılabilir. Genelde bilgi teknolojileri personeli soru sorulmadıkça konuşmaz, modellerin ve gerçeklerle ilgili diğer belgelerin geliştirilmesi için yazıcılarla çalışırlar. Gerekirse bazı teknik konularda kolaylaştırıcı bir bilgi teknolojileri uzmanına söz verilebilir.

## Toplantı Hazırlıkları ve Yürütülmesi

Çoğu ortak gereksinim planlama toplantısı üç ila beş gün, bazı durumlarda iki hafta kadar sürebilir. Toplantı tarihinden çok önce toplantı planı yapılmalıdır. Sistem analisti, sponsor ile çalışarak toplantılarının çerçevesini belirler. Aynı zamanda toplantıların yüksek düzeyli gereksinimleri ve beklenmeleri belirlenmelidir. Bu amaçla, sistem projesinde çalışacak bölümlerden veya projenin gerçekleştireceği fonksiyonlardan sorumlu seçilmiş kişilerle mülakatlar yapılır. Planlama sırasında yer, katılımcılar ve toplantılarının gündemi belirlenir. Toplantılar için farklı büyülükte odalar gereklidir. Tüm katılımcıların katılacağı toplantılar bir konferans salonunda yapılırken, katılımcıların küçük gruplara bölündüğü ve özel konuların tartışıldığı toplantılar daha küçük salonlarda yapılabilir. Odalarda yazıcıların çalışmaları kaydedebileceğinin yazılımları da içeren bilgisayarlar bulunmalıdır. Uzaktan katılımcıların tartışmalara dâhil olabilmesi için telekonferans imkânlarının bulunması gereklidir.

Kolaylaştırıcı, toplantıların hedefleri ve içeriği konusunda katılımcıları bilgilendirmek üzere belgeler hazırlamalıdır. Gündem üç parçadan oluşturulur: giriş, gövde ve sonuç. Girişte toplantılardan beklenmeler, uygulanacak kurallar ve katılımcıları motive edecek cümleler olmalıdır. Gövde, toplantılarda konuşulacak konuları detaylandırmak amacıyla yazılmıştır. Sonuçta ise günlük toplantıının özeti ve çözülemeyen konuların katılımcılara hatırlatılması yapılır.

Bir ortak gereksinim planlama toplantısı açılışı, giriş ve toplantıının hedef ve gündemi içeren bir konuşma ile başlar. Kolaylaştırıcı toplantıyı hazırlanmış gündem çerçevesinde yönetir. Toplantının başarılı şekilde yönetilmesi için kolaylaştırıcı aşağıdaki maddeleri uygulamalıdır:

- Gerekli olmadıkça gündemden sapılmamalıdır.
- Zaman çizelgesine uyulmalıdır (gündem maddelerine belirli zamanlar atanır).
- Yazıcıların not alabildiklerinden emin olunmalıdır (kullanıcı ve yöneticiler fikirlerini daha yavaş ve açık şekilde ifade etmelidir).
- Teknik dil kullanmaktan kaçınılmalıdır.
- Çalışmaları giderici yöntemler kullanılmalıdır.
- Serbest aralar verilmelidir.
- Grup anlaşması teşvik edilmelidir.
- Bazı kişilerin toplantıda baskın olmasına izin vermeden kullanıcı ve yöneticiler katılma teşvik edilmelidir.
- Kullanıcıların belirlenmiş toplantı kurallarına uymaları garanti edilmelidir.

Ortak gereksinim planlama toplantısının amacı bir problemi çözmek için muhtemel fikirlerin ortaya çıkarılmasıdır. Bunun için izlenecek yollardan biri beyin firtinasıdır. Beyin firtinasında, söylenenlerin geçerliliğini onaylamak için, analiz etmeden katılımcılar tarafından mümkün olduğu kadar çok fikir ortaya atılması sağlanır. Etkin bir beyin firtinası toplantı için aşağıdaki kurallar uygulanmalıdır.

- Katılımcıların rahatsız edilmeyeceği ve kesintisiz çalışabileceği bir ortam hazırlanmalıdır.
- Katılımcıların hepsi toplantıının amacı ve problemlerde odaklanılacak konular hakkında bilgilendirilmelidir.
- Katılımcılarından biri fikirleri kayıt altına almak üzere görevlendirilmelidir. Kayıt altına alınan fikirlerin herkes tarafından görülebilmesi için büyük yazı kâğıtları, tahta veya projeksiyon cihazı kullanılmalıdır.

Herkes aşağıdaki beyin firtinası kuralları için bilgilendirilmelidir.

- Doğal olunmalı, fikirler akla geldiği anda söyleyenmelidir.
- Fikirler üretilirken hiçbir şekilde eleştirilmemeli, analiz edilmemeli ve değerlendirilmemelidir. Bir fikir ancak bir başka fikri tetiklerse faydalıdır.

- Hedef, kaliteli fikirler yerine çok fazla fikir üretilmesidir.
- Verilen bir süre içinde takım elemanları düşünebildikleri fikirleri söylemelidir.
- Grubun yeni fikir üretemediği durumda beyin fırtınası durdurulur ve tüm fikirler kayıt altına alınır. Daha sonra fikirler analiz edilmeli ve değerlendirilmelidir.
- Son olarak üretilen fikirler gözden geçirilmeli, birleştirilmeli ve iyileştirilmelidir.

SIRA SİZDE



3

**Gerçekleri belirleme teknikleri nelerdir? Her bir tekniğin kullanılma nedenleri nedir?**

### **Gerçekleri Bulma Etiği**

Gerçekleri bulma sürecinde sistem analisti çok hassas bir bilgiyle karşılaşabilir veya bilgiyi analiz etmesi gerekebilir. Örneğin işletmenin bir ihalede vereceği teklifin yapısını, çalışanların ücretini, performans ve medikal geçmişini ve kariyer planlarını içeren bilgileri görebilir. Analist bu tür bilgilerin güvenlik ve gizliliğini korumak konusunda çok dikkatli davranışmalıdır. Hassas bir bilginin yanlış ellere geçmesi hâlinde sistem analisti, kullanıcılar ve yönetimin gözünde saygı ve güvenilirlik kaybına uğrar.

## Özet



**Sistem gereksinimlerine ilişkin temel kavramları açıklamak**

Sistem gereksinimleri, işletmenin bilgi sisteminin gerçekleştirmesini istediği işlevleri ve sistemin sahibi olabileceği özellikleri ve nitelikleri belirtir. İşlevsel gereksinimler bilgi sisteminin yapması gereken işleri tanımlar. İşlevsel olmayan gereksinimler ise güvenlik, kolay kullanım veya performans gibi sistemin sahibi olduğu özellik ve kaliteyi belirtirler.



**Gereksinimleri belirleme sürecinin adımlarını açıklamak**

Gereksinimleri belirleme süreci aşağıdaki adımlardan oluşur.

- Problemi belirleme ve analiz  
Gereksinimlerin doğru belirlenebilmesi için analist öncelikle problemi belirlemelidir. Problemi belirleme, analiz ve çözümünde yaygın olarak kullanılan bir araç Balık Kılçığı diyagramıdır. Bu diyagramda problemin adı balık kafasına karşı gelen diyagramın sağına yazılır. Problemin muhtemel sebepleri ise ana omurgaya bağlı kılıçıkların etrafına yazılır. Ana sebepler, malzeme, makine, insan gücü, çevre, yöntem, yer, süreç olarak adlandırılabilir. Beyin firtınası tekniği kullanılarak problemin ana sebeplerinin alt sebepleri belirlenebilir. Beyin firtinası tamamlandığında balık kılçığı, problemin temel ve alt sebeplerini gösteren bir diyagram olur.
- Gereksinimlerin tanımlanması  
Problemin belirlenmesinden sonra sistem gereksinimlerinin tanımlanması aşamasına geçilir. Gereksinimlerin tanımlanması için araştırma, toplantılar, mülakatlar, anketler ve örnekleme gibi gerçeklerin bulunması teknikleri kullanılarak sistem problemleri, gereksinimleri ve seçenekleri konusunda bilgi toplanır. Gerçeklerin bulunması tamamlandıktan sonra, kullanım durum, veri modelleri, süreç modelleri veya nesne modelleri kullanılarak gerçekler belgelendirilir ve sonuçlar çıkarılır.
- Gereksinimlerin analizi ve belgelendirilmesi  
Sistem analisti, gerçeklerin bulunması sürecinde toplanan bilgileri düzenli bir biçimde, anlaşılır ve anlamlı olarak toparlmalıdır ve belgelendirmelidir. Gereksinimlerin analizi ve belgelendirmesinin alt aşamaları; *taslak gereksinimlerin belgelendirilmesi, gereksinimlerin analizi ve gereksinimlerin resmileştirilmesi* olarak adlandırılır. Alt aşamaların sonunda sistem sahipleri ve sistem geliştirici ekibi arasında, yeni sistemde nelerin olacağını belirten bir kontrat olan *gereksinim tanımları belgesi* üretilir.

- Gereksinimlerin yönetimi

Projenin yaşam çevrimi içinde yeni gereksinimler ortaya çıkabilir veya gereksinimlerde değişiklik yapılması gerekebilir. Değişimin getireceği problemleri hafifletmek için, bir gereksinimde yapılacak değişiklikte izlenecek politikaları ve yöntemleri içeren gereksinimlerin yönetimi süreci kullanılmalıdır.



**Gerçekleri tespit etme tekniklerini sıralamak**

Gereksinimlerin belirlenmesi için analist, bilgi sisteminin gerçeklerini belli teknikleri kullanarak tespit etmelidir. Bu teknikler aşağıda verilmiştir:

- Mevcut belge, form ve dosyaların örneklenmesi
- Araştırma ve yerinde ziyaret
- İş ortamının gözlemlennmesi
- Anketler
- Mülakatlar
- Prototipleme
- Ortak gereksinim planlaması

## Kendimizi Sınayalım

- 1.** Bir üniversite öğrenci bilgi sistemi ile ilgili aşağıda verilenlerden hangisi işlevsel **olmayan** gereksinimdir?
  - a. Öğrenciler kayıt olacağı dersleri internet üzerinden seçebilmelidir.
  - b. Her öğrenci notlarını sistem üzerinden sorgulayabilecektir.
  - c. Aynı anda 250 öğrenci kayıt yapabilmelidir.
  - d. Danışman öğretim üyesi danışmanı olduğu öğrencilere durumlarını sistem üzerinden sorgulayabilmelidir.
  - e. Öğretim üyeleri sınav sonuçlarını sisteme girip ilan edebilmelidir.
  
- 2.** Aşağıdakilerden hangisi sistem gereksinimlerinin sağlanması gereken ölçütler arasında **yer almaz**?
  - a. Tutarlılık
  - b. Gerçeklenebilirlik
  - c. Doğruluk
  - d. Onaylanabilirlik
  - e. Kolayca yapılabılırlik
  
- 3.** Aşağıdakilerden hangisi gereksinim belirleme adımlarından biri **değildir**?
  - a. Problemi belirleme
  - b. Gereksinimleri gerçekleştirmeye
  - c. Belgelendirme
  - d. Gereksinimlerin yönetimi
  - e. Problemin analizi
  
- 4.** Katılımcıların kısa bir süre içinde, belirli bir konuda, herhangi bir analiz yapmadan mümkün olduğunda çok fikir üretmeleri için uygulanan ve tüm fikirler söylenilinceye kadar devam eden fikir üretme tekniği aşağıdakilerden hangisidir?
  - a. Anket
  - b. Mülakat
  - c. Balık kılıçlığı
  - d. Beyin fırtınası
  - e. Ortak gereksinim planlaması
  
- 5.**
  - I. Taslak gereksinimlerin belgelenmesi
  - II. Gereksinimlerin yönetimi
  - III. Gereksinimlerin resmileştirilmesi

Yukarıdakilerden hangisi(leri) gereksinim analizi ve belgeleştirilmesi adımdında yapılır?

  - a. Yalnız I
  - b. Yalnız III
  - c. I ve II
  - d. I ve III
  - e. II ve III
  
- 6.** Aşağıdakilerden hangisi gereksinim tanımaları belgesinin içermesi gereken özelliklerden biri **değildir**?
  - a. Sistemin maliyeti
  - b. Sistemin sağlayacağı hizmetler
  - c. Sistemin geliştirilmesini sınırlayan gereksinimler
  - d. Sistemin bağlanacağı diğer sistemler hakkında bilgi
  - e. İşlevsel olmayan gereksinimler
  
- 7.** Aşağıdakilerden hangisi mevcut belge, form ve dosyaların örneklenmesi aşamasında izlenecek belgelerden biri **değildir**?
  - a. Müşteri şikayetleri
  - b. Veritabanlarını gösteren tasarım belgeleri
  - c. Muhasebe kayıtları
  - d. Çalışanların devam çizelgesi
  - e. Bilgi sistemleri proje istekleri
  
- 8.** Analist iş ortamını gözlemlerken hangi davranıştan kaçınmalıdır?
  - a. Çalışanları rahatsız etmemek
  - b. Not almak
  - c. Şeften izin almak
  - d. Sessiz olmak
  - e. İstediği her an gözlem yapmak
  
- 9.**
  - I. "Günde kaç saat TV izliyorsunuz?"
  - II. "Televizyon izleme günlük yaşamınızı etkiliyor mu?"
  - III. "Size göre iletişim nedir?"

Yukarıda verilen mülakat soru tipleri hangi seçenekte doğru sıralanmıştır?

  - a. Açık uçlu, açık uçlu, kapalı uçlu
  - b. Kapalı uçlu, açık uçlu, kapalı uçlu
  - c. Açık uçlu, kapalı uçlu, açık uçlu
  - d. Her üç soru açık uçludur
  - e. Her üç soru kapalı uçludur
  
- 10.** Aşağıdaki ifadelerden hangisi prototiplemenin dezavantajlarından biridir?
  - a. Sistem geliştirme süreci uzayabilir.
  - b. Sistemin çalışması konusunda bilgi elde edilebilir.
  - c. Prototip kullanıcılarla eğitim aracı olarak kullanılabilir.
  - d. Gerçekleri bulma süresi kısalabilir.
  - e. Geliştirme masrafi azalabilir.

## Kendimizi Sınavalım Yanıt Anahtarı

1. c Yanınız yanlış ise “Sistem Gereksinimleri” konusunu yeniden gözden geçiriniz.
2. e Yanınız yanlış ise “Sistem Gereksinimleri” konusunu yeniden gözden geçiriniz.
3. b Yanınız yanlış ise “Gereksinimleri Belirleme Süreçi” konusunu yeniden gözden geçiriniz.
4. d Yanınız yanlış ise “Problemi Belirleme ve Analiz” konusunu yeniden gözden geçiriniz.
5. c Yanınız yanlış ise “Gereksinimlerin Analizi ve Belgelendirme” konusunu yeniden gözden geçiriniz.
6. a Yanınız yanlış ise “Gereksinimlerin Resmiletirilmesi” konusunu yeniden gözden geçiriniz.
7. d Yanınız yanlış ise “Mevcut Form, Belge ve Dosyaların Örneklenmesi” konusunu yeniden gözden geçiriniz.
8. e Yanınız yanlış ise “İş ortamının Gözlemlenmesi” konusunu yeniden gözden geçiriniz.
9. c Yanınız yanlış ise “Mülakatlar” konusunu yeniden gözden geçiriniz.
10. a Yanınız yanlış ise “Prototipleme” konusunu yeniden gözden geçiriniz.

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Sistem gereksinimleri belirlenme aşamasında doğru tanımlanmazsa, bilgi sistemi geliştirme süreci ilerledikçe yanlış tanımlanan gereksinimin düzeltilmesi için yapılacak değişikliğin maliyeti çok yüksek olabilir. Bu nedenle gereksinim tanımlanırken belli ölçütleri sağlamalıdır. Bu ölçütler: *tutarlılık, tamlık, gerçeklenebilirlik, gereklik, doğruluk, takip edilebilirlik, onaylanabilirlik* olarak sayılabilir.

### Sıra Sizde 2

Bir sistem hakkında bilgi edinmek için incelenebilecek belgeler aşağıdaki biçimde sıralanabilir:

- Organizasyon şeması
- Ofis içi yazışmalar
- Öneri kutusu notları
- Problem konusunda müşteri şikayetleri ve raporlar
- Muhasebe kayıtları
- Performans ve çalışma özetleri
- Bilgi sistemleri proje istekleri
- İşletmenin misyonu ve stratejik planı
- Projede üzerinde çalışılan işletmenin alt birimlerinin resmi hedefleri
- Önerilen herhangi bir sistem üzerine yazılan politika el kitapları
- Günlük işler için standart çalışma prosedürleri
- İş taslakları ve talimatları
- İşlem döngüsünün çeşitli noktalarında yapılan işlemleri gösteren formlar
- Bilgisayar veritabanlarının ekran ve raporlarının el kitabı örnekleri,
- Akış diyagramları
- Proje sözlükleri
- Girdi, çıktı ve veri tabanlarını gösteren tasarım belgeleri
- Program belgesi
- Bilgisayar işlemlerinin ve eğitimlerinin el kitapları

### Sıra Sizde 3

Gerçekleri belirleme teknikleri ve kullanılma nedenleri aşağıda verilmiştir:

- *Mevcut belge, form ve dosyaların örneklenmesi:* Analist, geliştirilecek bilgi sisteminin kullanılacağı işletme hakkında bilgi edinmek, projenin sahipleri ve kullanıcıları arasındaki ilişkileri tanımlamak ve projenin geçmişini öğrenmek için mevcut belgeleri incelemelidir.
- *Araştırma ve yerinde ziyaret:* İşletmenin geliştireceği bilgi sisteminde karşılaşacağı problemler benzer diğer işletmeler tarafından önceden çözülmüş olabilir. Bu durumda,

## Yararlanılan ve Başvurulabilecek Kaynaklar

kabul edildiği takdirde, benzer problemi çözen işletmeyi yerinde ziyaret ederek çözümleri konusunda bilgi alınamasıl. Bu şekilde kazanılan bilgi, bilgi sisteminin geliştirme süresi ve maliyetinde büyük oranda düşüş sağlayacaktır. İnternet üzerinde yapılacak araştırma da benzer problemlerin başkaları tarafından nasıl çözüldüğü konusunda önemli bilgiler verebilir.

- *İş ortamının gözlemlenmesi:* Gözlem bir sistem hakkında veri toplama için en etkili tekniklerden biridir. Analist sistem hakkında bilgi edinmek için insanları ve faaliyetlerini gözlemler. Bu teknik genelde diğer yöntemlerle top- lanmış ancak soru işaretleri olan verilerin onaylanmasında veya kullanıcıların sistem hakkında kesin bir açıklama yapamadıkları durumlarda kullanılmaktadır.
- *Anketler:* Çok sayıda kişiden veri toplayarak bilgi sistemi- nin sağlanması istenilen özelliklerin belirlenebilmesi için anket çalışması yapılabilir.
- *Mülakatlar:* Kişisel mülakatlar en önemli ve en çok baş- vurulan gerçekleri tespit etme tekniğidir. Mülakatlar, doğrudan ve yüz yüze temasla gerçeklerin bulunması, onaylanması, anlaşılması, gereksinimlerin belirlenmesi, fikir ve önerilerin sorulması amacıyla gerçekleştirilir.
- *Prototipleme:* Prototipleme, sistem geliştirme ekibinin sis- tem gereksinimlerini belirlemekte zorlandığı durumlarda uygulanır. Bu yaklaşımın felsefesi, kullanıcıların prototipi gördüklerinde gereksinimleri keşfedeceğidir.
- *Ortak gereksinim planlaması:* Birçok işletme ayrı ve sayı- siz mülakatlar yerine grup çalışma oturumlarını kullan-maktadır. Grup çalışma oturumu uygulamalarının bir örneği de ortak gereksinim planlamasıdır. Problemlerin tanımlanması ve analizi ile gereksinimlerin belirlenmesi amacıyla yapılandırılmış grup toplantıları düzenlenir.

Kendall K.E., Kendall J. E., (2006). **Systems Analysis and De- sign**, 8th Edition, Boston, Pearson Education Inc.

Sommerville I., (2011). **Software Engineering**, 9th Edition, Boston, Addison-Wesley.

Valavich J. S., George J. F., Hoffer J. A., (2012). **Essentials of Systems Analysis and Design**, 5th Edition. Boston, Pe- arson Education Inc.

Whitten J. L., Bentley L. D., (2007). **Systems and Design Methods**, 7th Edition, New York, Mc Graw-Hill Inc.



# 5

## Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Sistem geliştirme yaşam döngüsünün analiz aşamasında, kullanım durumlarının bulunması gerekliliğini açıklayabilecek,
  - 🕒 Kullanım durum modelleme temel kavramlarını tanımlayabilecek,
  - 🕒 Veri yaşam döngüsünü açıklayabilecek,
  - 🕒 Farklı bilgi seviyelerinde kullanılan veri modellerini açıklayabilecek,
  - 🕒 Varlık ilişki diyagramlarının oluşturulmasında kullanılan temel öğeleri açıklayabilecek bilgi ve becerilere sahip olabileceksiniz.

## Anahtar Kavramlar

- Kullanım Durum Diyagramları
- Veri Modelleme
- Veri Yaşam Döngüsü
- Harici Veri Modeli
- Kavramsal Veri Modeli
- Mantıksal Veri Modeli
- Fiziksel Veri Modeli
- Varlık İlişki Diyagramları

## İçindekiler

Sistem Analizi ve Tasarımı

Kullanım Durum ve Veri Modelleme

- GİRİŞ
- KULLANIM DURUM MODELLEME
- KULLANIM DURUM MODELLEME TEMEL KAVRAMLARI
- VERİ MODELLEME VE VERİ YAŞAM DÖNGÜSÜ
- FARKLı BİLGİ SEVİYELERİNDE VERİ MODELLEME
- VARLIK İLİŞKİ DİYAGRAMLARI

# Kullanım Durum ve Veri Modelleme

## GİRİŞ

Sistem geliştirme yaşam döngüsünün (SGYD) analiz aşamasında, sistem gereksinimleri üzerinden sistemin gerçeklestireceği işe odaklanılmaktadır. Bu aşamada, verilerin nasıl oluşturulacağı, depolanacağı ya da veriler üzerinde gerçekleştirilecek işlemlerin nasıl yapılacağı gibi teknik ayrıntılarla uğraşılmak istenilmez. Bu amaçla analistler, sistemi ifade etmek amacıyla bir **model** oluştururlar. Veri modeli, teknik ayrıntılara girilmeden verilerin mantıksal olarak düzenlenmesi ve sunulmasıdır. Benzer amaçla, geliştirilecek sistem içindeki kullanıcıların işlemleri tamamlayabilmek için gerekli olan etkileşimin açıklanması ve belgelenmesi amacıyla **kullanım durum** kullanılmaktadır. Kullanım durumlar, nesne tabanlı geliştirmenin bir parçasıdır. Bununla beraber yeni bir sistem için gereksinimlerin belirlenmesi sürecinin önemli bir bileşeni olarak da karşımıza çıkmaktadır. Kullanım durum, kullanıcıların sistem ile etkileşimi ve sistemin bu kullanıcılarla verdiği yanıtı görsel olarak anlatmaktadır. Kısacası sistemin nasıl davranışını ve çalıştığını göstermektedir. Veri modelleme ise bu sistem içerisinde oluşturulan ve kullanılan verilerin gösterimi için kullanılan biçimsel bir yöntemdir. Sistem analisti, veri modelleri yardımıyla yer veya nesnelerden hangileri ile ilgili bilgilerin gerekli olduğunu ve bu bilgilerin birbirleriyle ilişkilerini göstermektedir. Bu üitede, kullanım durum bileşenlerinin neler olduğu, sistem kullanıcılarının sistem ile gerçekleştirdiği işlemleri belirten kullanım durumlarının nasıl oluşturulacağı ele alınmıştır. Ayrıca en yaygın kullanılan kavramsal veri modelleme tekniklerinden varlık ilişki diyagramlarının bileşenleri ve nasıl oluşturulacağı açıklanmıştır.

## KULLANIM DURUM MODELLEME

Bilgi sistemleri başarılı bir planlama, analiz, üretim ve dağıtım sürecinden geçmelidir. Bu amaçla analistlerin, öncelikle paydaşların gereksinimlerini ve bilgi sisteminin neden geliştirilmek istendiğini anlamaları gerekmektedir. Bu bağlamda **kullanıcı merkezli sistem geliştirme**, analistlerin sistem kullanıcıları üzerine odaklanarak sistemin nasıl kullanılacağına ve nasıl oluşturulacağına yoğunlaşmalarını sağlar. Kullanım durum modellemesi kullanıcı ve kullanım merkezli geliştirmeyi kolaylaştıran bir yaklaşımındır.

Kullanım durum, kısaca bir kullanıcı ve bir sistem arasındaki etkileşimi anlatan senaryo topluluğu olarak da tanımlanabilir. Dr. Ivar Jacobson senaryoyu, aktör ile sistem arasında gerçekleştirilen, sonucunda aktöre fark edilir faydası oluşan etkileşimli diyalog olarak tanımlamaktadır.

**Model:** Gerçek sistemin grafiksel gösterimidir.

**Kullanım Durum (Use Case):** İşletme faaliyetleri, bu faaliyetlerin kim tarafından başlatıldığı ve bu faaliyetlere bilgi sisteminin nasıl cevap vereceği açısından sistem fonksiyonlarının modelidir.

**Kullanıcı merkezli sistem geliştirme**, paydaşların ihtiyaçlarının ve sistemin geliştirilme nedenlerinin anlaşılması temelinde dayanan sistem geliştirme süreci.

**Ivar Jacobson:** 1986 yılında kullanım durumu belirtmede kullanılan görsel modelleme tekniğini ilk kez kodlamıştır.

## İNTERNET



Ivar Jacobson'ın kurucusu ve yönetim kurulu başkanı olduğu, endüstriyel yazılım projelerinin yönetim ve teknik liderliği konularında hizmet veren Ivar Jacobson International web sitesine [https://www.ivarjacobson.com/ adresinden erişebilirsiniz.](https://www.ivarjacobson.com/)

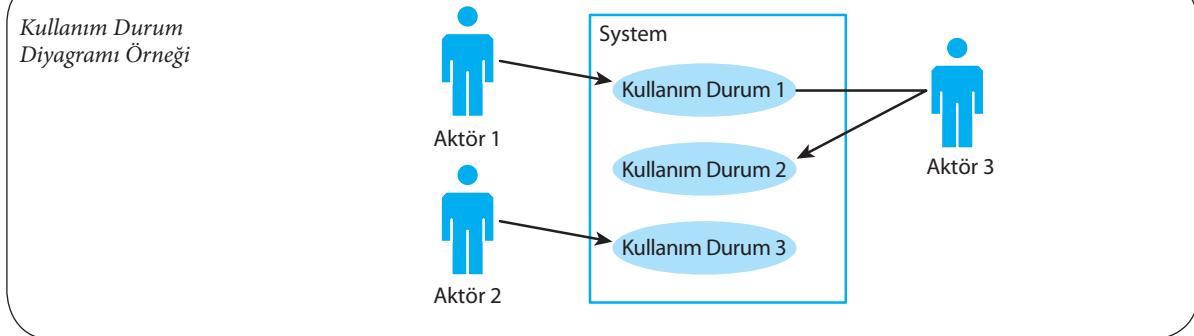
Kullanıcı katılımı projenin başarıya ulaşmasındaki kritik faktörlerden biridir. Kullanım durum modellemesi kullanıcı katılımını kolaylaştırır ve teşvik eder. Kullanım durum modellemenin sağladığı faydalardan bazıları aşağıda verilmiştir:

- İşlevsel gereksinimleri belirleyen bir araç sağlar.
- Geliştirilecek sistemin kapsamının, yönetilebilir alt parçalara ayrıştırılmasına yardımcı olur.
- Sistem içerisindeki farklı paydaşların kolaylıkla anlayabileceği ortak bir dil sağlar.
- Sistem geliştirme etkinliklerinin tanımlanmasına, atanmasına, izlenmesine, kontrol edilmesine ve yönetilmesine olanak sağlar.
- Proje kapsamının, harcanacak çabanın ve zamanlamasının tahmin edilmesine yardımcı olur.
- Hem kullanıcı yardım sistemi ve kılavuzlarının hem de sistem geliştirme raporlarının belgelenmesine yardımcı olur.
- Test planlarının yapılması ve test durumlarının tanımlanması için uygun bir araçtır.
- Gereksinimlerin izlenebilirliğini sağlar.
- Veri nesneleri veya varlıkların tanımlanması için bir başlangıç noktasıdır.
- Kullanıcı ve sistem arayüzlerinin tasarımları için işlevsel özelliklerin belirlenmesine yardımcı olur.
- Ekleme, değiştirme, silme ve okuma gibi veritabanı erişim gereksinimlerinin tanımlanmasına olanak sağlar.
- Sistem geliştirme projesinin idaresi için bir çerçeve sunar.

## KULLANIM DURUM MODELLEME TEMEL KAVRAMLARI

Kullanım durum modellemesi iki temel çıktı sağlar. İlkisi, geliştirilen sistemde bulunan kullanım durumları, kullanıcıları ve ikisi arasındaki ilişkileri gösteren kullanım durum diyagramıdır. Örnek bir kullanım durum diyagramı Şekil 5.1'de verilmiştir.

**Şekil 5.1**



**Use Case Narrative (Kullanım Durum Hikâyesi):** Kullanım durumun yazılı olarak ifade edilmesidir.

Diğer temel çıktı, her bir işletme faaliyetinin ayrıntılarının verildiği kullanım durum hikâyesidir. İş etkinliği süresince kullanıcı ile sistem arasında gerçekleşecek etkileşimin, nasıl gerçekleşeceği kullanım durum hikâyesinde belirtilir. Kullanım durum hikâyesi, kullanım durum formu olarak da adlandırılmaktadır ve bu belgelerin kesin bir biçimde bulunmamaktadır. Kullanım durum formları içinde bulunması gereken bilgiler tamamen sistemin büyüklüğüne ve sistemde bulunacak içeriğe bağlıdır.

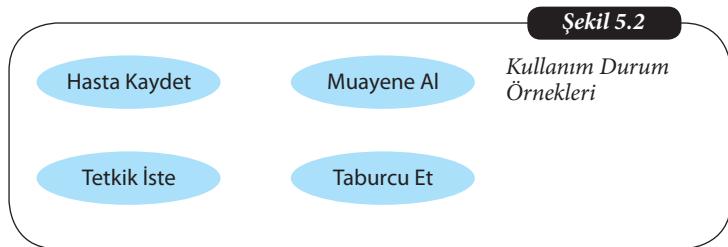
## Kullanım Durum Diyagramı

Bir kullanım durum diyagramı, kapsamı içerisindeki kullanım durumlarıyla ilgili olarak ayrıntılı bilgi bulundurmaz. Kullanım durumlar, aktörler ve sistemler arasındaki ilişkiler hakkında bilgi verir. Kullanıcıların, kullanım durumlar tarafından temsil edilen faaliyetleri nasıl gerçekleştirecekleri kullanım durum diyagramlarında verilmez. Kullanım durum diyagramlarının dört temel bileşeni bulunmaktadır:

- i. Kullanım Durumlar
- ii. Aktörler
- iii. İlişkiler
- iv. Sistem Sınırı

### Kullanım Durumları

*Kullanım durumları*, harici kullanıcıların bakış açısından sistem işlevlerini tanımlarlar. Grafiksel olarak, Şekil 5.2'de görüldüğü gibi, üzerinde veya içinde kullanım durumun adının yazıldığı bir elips şekli ile gösterilir. Bir kullanım durum sistemi içinde gerçekleştirilecek tek bir işlevi gösterir. Gösterilen bu işlev, bir dizi etkinlik



ve kullanıcı etkileşimi yardımıyla gerçekleştirilir. Kullanım durumlar, SGYD içerisinde başlangıç gereksinimlerin belirlenmesi aşamasında tanımlanır ve yaşam döngüsü içinde iyileştirilir. Şekil 5.2'de örnek olarak bir hastane bilgi sisteminde yer alabilecek bazı kullanım durumları görülmektedir.

### Aktörler

Kullanım durum modellemeye, belli bir sistem işlevini gerçekleştirmeye yönelik olarak sistemle etkileşime giren kullanıcılar **aktör** olarak tanımlanmaktadır. Aktörler; kullanıcılar, kurumlar, bilgisayar programları, harici donanımlar kısacası sistemde gerçekleşen olayları başlatan veya tetikleyen sistem bileşenleridir. Sistemde bulunan **zamansal olaylarda** ise aktör zamandır. Birçok bilgi sisteminde belirli takvim veya saatlerde tetiklenen iş olayları bulunmaktadır.

**Zamansal Olay (Temporal Event):** Zaman tarafından tetiklenen sistem olaylarını ifade eder.

**Tüm aktörler aynı zamanda birer paydaş olabilir fakat tüm paydaşlar aktör olmayıpabilirler. Paydaşların birçoğu sistemle doğrudan etkileşimde bulunmayıpabilir. Örneğin, bir hastane bilgi sisteminde sosyal güvenlik kurumu yönetim kurulu üyeleri paydaş olarak kabul edilebilirler ama sistemle doğrudan etkileşimde bulunmadıkları için aktör olarak tanımlanamazlar.**



DİKKAT

Aktörler,

- Birincil iş aktörleri,
- Birincil sistem aktörleri,
- Harici sunucu aktörleri,
- Harici alıcı aktörler olmak üzere dört grup altında toplanabilir.

**Birincil iş aktörleri**, kullanım durumun gerçekleştirilmesi sonucunda doğrudan ölçülebilir ve gözlemlenebilir bir fayda sağlayan paydaştır. Birincil iş aktörü sistem işlevlerini başlatabileceği gibi başlatmasının gerekmeliği durumlar da olabilir. Banka ödeme sistemi tarafından çalışanın hesabına her ay belirli bir tarihte maaşının yatırılması bu duruma örnek verilebilir. Maaş ödenmesi faaliyetinden doğrudan fayda sağlayan bir paydaş olan çalışanın, bu faaliyetin başlatılması için herhangi bir işlem gerçekleştirmesine gerek yoktur.

*Birincil sistem aktörü*, bir iş veya sistem olayını başlatmak ya da tetiklemek için sistem ile doğrudan etkileşimde bulunan paydaşlardır. Bazı durumlarda birincil iş aktörü ve birincil sistem aktörü aynı kişi olabilir. Bu duruma hastanede muayene olmak amacıyla telefon ya da İnternet yardımıyla randevu alan hastayı örnek verebiliriz.

*Harici sunucu aktörü*, kullanım durum isteklerine cevap veren paydaş olarak tanımlanır.

*Harici alıcı aktör*, birincil iş aktörü olmadığı hâlde kullanım durumdan ölçülebilir ve gözlemlenebilir bir çıktı elde eden paydaştır.

SIRA SİZDE

1

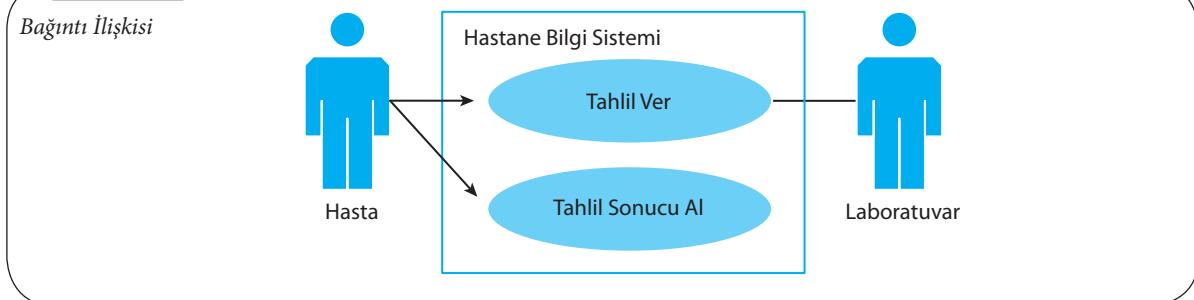
**Bir hastane yönetim bilgi sisteminde birincil iş aktörü, birincil sistem aktörü, harici sunucu aktörü ve harici alıcı aktör neler ya da kimler olabilir örnek veriniz.**

### İlişkiler

Kullanım durumları ve aktörler belirlendikten sonra kullanım durum diyagramlarını oluşturmak için bu iki bileşeni ilişkilendirmek gerekmektedir. Bir kullanım durumla aktör arasındaki ilişki, söz konusu aktörle kullanım durumu birbirine bağlayan bir çizgiyle ifade edilir. Böylece aktörün sistem üzerinde hangi işlevi gerçekleştirebileceği tanımlanmış olur.

**Bağıntı ilişkisi**, bir aktör ve kullanım durum arasında gerçekleşen ilişkidir. Bir aktör en az bir kullanım durum ile bağıntılı olmalıdır. Bir aktör birden fazla kullanım durumla bağıntılı olabildiği gibi, bir kullanım durum birden fazla aktörle bağıntılı olabilir. Bağıntı, aktör ile kullanım durumu birbirine bağlayan tek bir çizgi ile temsil edilir. Kullanım durum ve aktör (birincil iş aktörü veya birincil sistem aktörü) arasındaki bağıntıyı gösteren çizginin kullanım duruma bağlılığı noktadaki ok; kullanım durumun, çizginin diğer ucuna bağlı olan aktör tarafından gerçekleştirildiğini göstermektedir. Şekil 5.3'te görüldüğü gibi çizgide ok bulunmaması, kullanım durum ile harici sunucu veya alıcı aktör arasında bir bağıntı olduğunu göstermektedir.

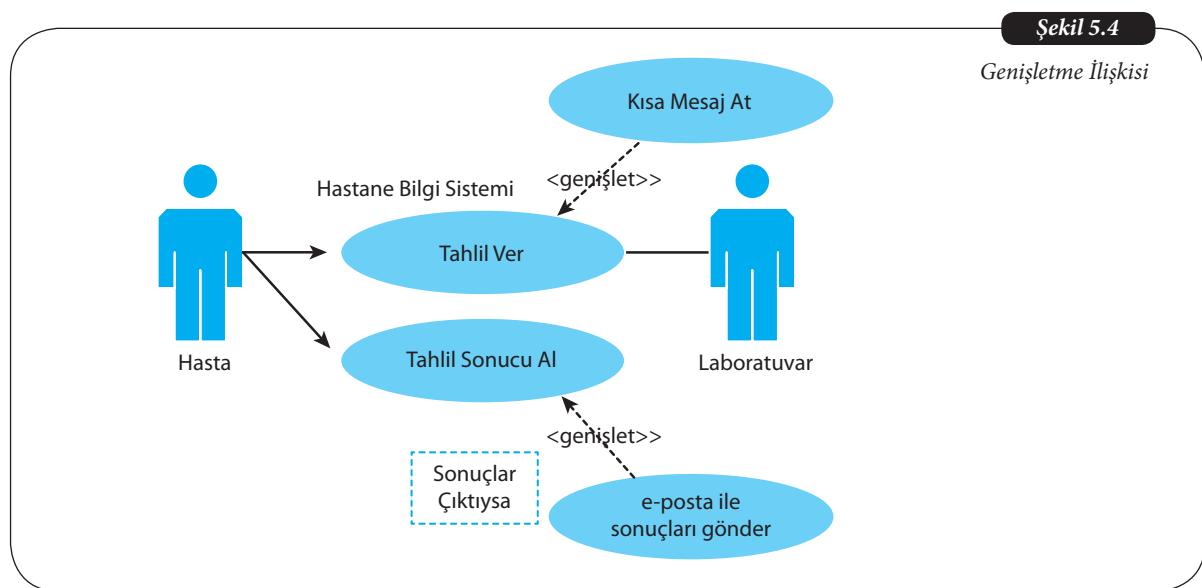
**Şekil 5.3**



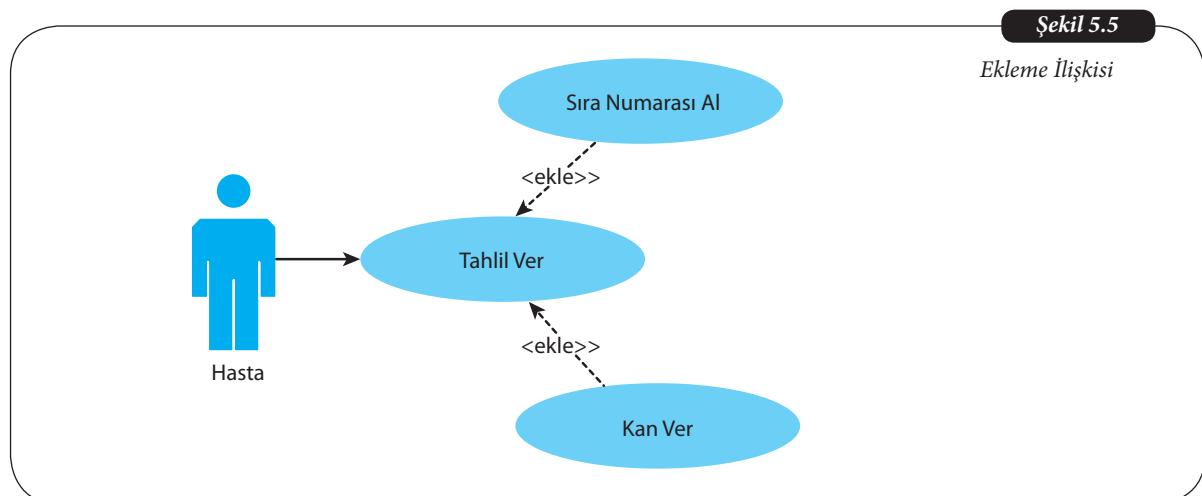
**Genişletme ilişkisi (Extends Relationship)**: Bir kullanım durumun yeni işlevler kazandırılması amacıyla başka bir kullanım durum ile ilişkilendirilmesidir.

**Eklenti Kullanım Durum (Extension Use Case)**: Birden fazla kullanım durumda yer alabilecek ortak işlevleri içeren kullanım durumdur.

**Genişletme ilişkisi**, bir kullanım durumun başka kullanım durumları ile ilişkilendirilerek yeni işlevler kazandırılması amacıyla gerçekleştirilir. Bir kullanım durum çoğu zaman birkaç adımdan oluşan karmaşık işlevler içerebilir. Bu karmaşık işlevler, kendi alt kullanım durumlarına ayrıtırılarak **eklenti kullanım durumlar** elde edilir. Böylelikle bir kullanım durum genişletilmiş olur. Bir kullanım durum birden fazla eklenti kullanım durum ile ilişkili olabilmektedir. Ayrıca genişletilmiş kullanım durum tek başına anlamlı olmalıdır. Eklenti kullanım durum olmadan da davranış sergileyebilmelidir. Örneğin Şekil 5.4'te görüldüğü gibi tahlil örneklerinin verilmesinin ardından laboratuvara sonuçlar elde edilmez hastanın e-postasına gönderilmesi sağlanabilir. Ayrıca, başka bir kullanım durum ile de tahlil örneği verildikten sonra, tahlil sonuçlarının ne zaman alınacağına ilişkin bilgi kısa mesaj olarak iletilenbilir. Eklenti kullanım durum, genişletilmiş kullanım duruma bağımlıdır. Örneğin, "kısa mesaj at" kullanım durum, "tahlil ver" kullanım durum olmadan anlamsızdır. Benzer şekilde "tahlil sonucu al" kullanım durumu "e-posta ile sonuçları gönder" kullanım durumu anlamlı kılmaktadır.

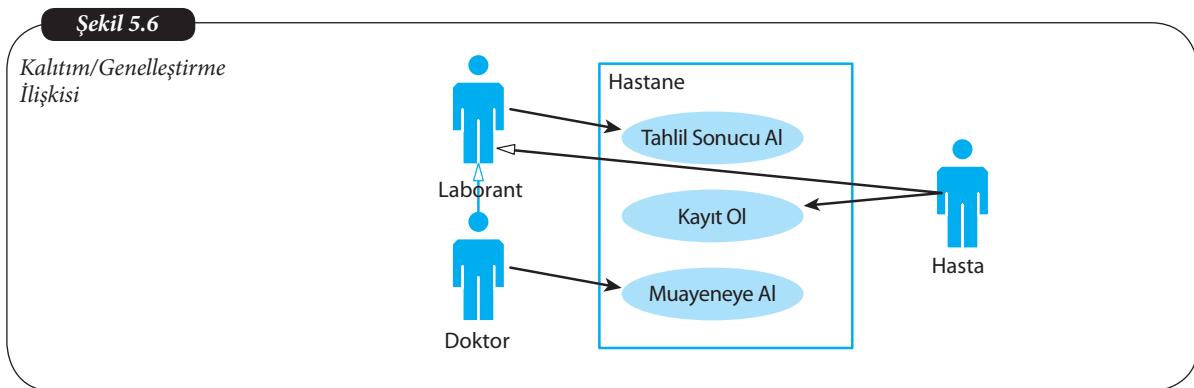


**Ekleme ilişkisi**, bir kullanım durum içindeki adımların diğer kullanım durumlarda da yer almasıyla meydana gelen ilişkidir. Ana amaç, ortak işlevlerin sistem içindeki tüm bileşenlerde aynı şekilde tanımlanmasıdır. Böylelikle bu işlevlerin tekrar kullanılabilirliği sağlanmaktadır. Şekil 5.5'te hastane bilgi sisteminde kayıtlı hasta tahlil verirken önce sıra numarası almalı sonrasında da kan vererek tahlil verme durumunu tamamlamalıdır.



**Kalıtım ilişkisi**, bir aktörün kendisiyle aynı işlevleri meydana getiren aktörler ile olan ilişkisi olarak tanımlanmaktadır. Aynı işlevleri kullanan birden fazla aktör olduğunda ortak işlevler ortaya çıkarılarak, bu işlevleri kullanan bir **soyut aktör** oluşturulur. Örneğin, Şekil 5.6'da hasta ve doktor aktörleri ile "tahlil sonucu al" kullanım durumunu başlatabilen laborant soyut aktörü arasında kalıtım ilişkisi bulunmaktadır. Bu ilişki aynı zamanda genelleştirme ilişkisi olarak da tanımlanmaktadır.

**Soyut Aktör (Abstract Actor):**  
Kullanım durum diyagramlarında, birden fazla aktöre ait olabilecek ortak özellikleri taşıyan en temel aktördür.

**Şekil 5.6**

**Sistem sınırı**, kullanım durum diyagramlarının dördüncü ve son bileşenidir. Kullanım durumları, aktörleri ve bunlar arasındaki ilişkilerin tamamını kapsamaktadır. Sistemin adının yazıldığı bir dörtgen ile temsil edilmektedir.

### Kullanım Durum Hikâyeleri

Kullanım durum modellemesi gerçekleştirilirken oluşturulan kullanım durum diyagramları kadar önemli ikinci temel çıktı kullanım durum hikâyeleridir. Kullanım durum formu olarak da isimlendirilen bu belgenin belli bir standart biçimde bulunmamaktadır. Sistemin büyüklüğüne ve sistemde bulunacak içeriğe bağlı olarak değişmektedir. Kullanım durum hikâyesi içinde bulunması gereken bilgilerden bazıları aşağıda açıklanmıştır:

- **İsim:** Her kullanım durum, sistem içerisinde diğer kullanım durumlardan ayrılmalarını sağlayan bir isme sahip olmalıdır. İsimler kullanım durumun amacını belirtmelidir.
- **Amaç/Açıklama:** Kullanım durumun ne amaçladığı yazılır.
- **Aktör/Aktörler:** Kullanım durumun aktör ya da aktörleri belirtilir. Aktörler belirlenirken aşağıdaki sorular sorulabilir:
  - Sisteme girdileri kim ya da ne sağlıyor?
  - Sistemden çıktıları kim ya da ne alıyor?
  - Diğer sistemler için bir arayüz ihtiyacı var mı?
  - Önceden hesaplanabilen bir zamanda otomatik olarak tetiklenen olaylar var mı?
  - Sistemdeki bilgilerin bakım ve güvenliğini kim sağlayacak?
- **Ön koşullar:** Kullanım durum başlatılmadan ya da tetiklenmeden önce sağlanması gereken koşulları gösterir.
- **Son koşullar:** Kullanım durum tamamlandıktan sonra sağlanması gereken koşulları gösterir.
- **Tetikleyiciler:** Kullanım durumun hangi aktör tarafından tetikleneceğini gösterir.
- **Ana başarı senaryosu:** Kullanım durum içinde gerçekleştirilecek işlevleri ve adımları içerir. Kullanım durum başlatıldığından ya da tetiklendiğinde normal koşullarda gerçekleşmesi beklenenleri açıklar.
- **Hata senaryoları:** Kullanım durum içinde karşılaşılabilecek olası hatalar listelenir.
- **Alternatif senaryolar:** Aktörler ya da sistem tarafından kaynaklanan ve ana başarı senaryosunun normal işleyişinden sapmalar meydana geldiğinde nasıl davranışlı sorusuna cevap verilir. Kisacası yukarıda bahsedilen hata senaryoları meydana geldiğinde, kullanım durumun nasıl davranışacağını anlatıldığı bölümdür.

Kullanım durum hikayesinin hazırlanması için gerekli adımlar ve açıklamaları Tablo 5.1'de listelenmiştir.

[http://creately.com/diagram-community/all?term=“use case”](http://creately.com/diagram-community/all?term=“use%20case”) bağlantısından çevrimiçi uygulama yardımıyla hazırlanmış olan kullanım durum diyagramı şablonları üzerinde çalışabilirsiniz.



INTERNET

Adım	Etkinlikler	Sorulacak Sorular
1. Kullanım durumları belirleyin.	Her bir kullanım durum için isim, amaç ve tetikleyici bilgilerinin bulunduğu formların doldurulmasıyla başlayın.	Her bir kullanım durum ile ilgili kim, ne, ne zaman ve nerede sorularını sorun. Gerçekleştirilecek büyük adımlar neler? Bu kullanım durumu ne tetikler?
2. Her bir kullanım durum için büyük öneme sahip adımları tanımlayın.	Her bir kullanım durum için, işlevin tamamlanması için gerekli büyük adımları yazın.	Her bir kullanım durum ile ilgili nasıl sorusunu sorun. Hangi bilgi/form/rapor gerekiyor? Bu bilgi/form/rapor kim tarafından sağlanacak? Bu rapor nasıl üretilicek? Rapor üzerindeki bilgi nasıl değiştirilecek? Formlar nasıl işlenecek? Hangi araçlar kullanılacak? (örneğin, kâğıt, e-posta, telefon)
3. Adımlar içindeki bileşenleri tanımlayın.	Her bir adımın tetikleyicilerini, girdilerini ve çıktılarını tanımlayın.	Her bir adım ile ilgili nasıl sorusunu sorun. Kullanıcılar bu adımı ne zaman gerçekleştireceğini nasıl bileyec? Bu adımda üretilen form/rapor/veriler neler? Bu adımda ihtiyaç duyulan form/rapor/veriler neler? Bu form/rapor/veri mevcut olmadığından ne olacak?
4. Kullanım durumu onaylayın.	Her bir kullanım durum için, doğru ve eksiksiz olduğunu doğrulayın.	Kullanıcıdan kullanım durumda yazılı her bir adımı yürütmesini isteyin.

Tablo 5.1

Kullanım Durum Hikâyesinin Yazılması İçin Gerekli Adımlar

**Kaynak:** Roth,  
R.M., Dennis,A.,  
Wixom,B.H., (2012),  
s. 157.

Bir hastanın hastanede muayene olmak için gerekli kullanım durum hikâyesini hazırlayınız.



SIRA SİZDE

## VERİ MODELLEME VE VERİ YAŞAM DÖNGÜSÜ

Veri modelleme, sistemdeki paydaşlar için gerçek dünya ihtiyaçlarının daha kolay anlaşılması sağlayan bir yöntemdir. Kullanım durum bölümünde, bir bilgi sistemi tasarımları için işlevsel gereksinimlerin nasıl tanımlanacağı anlatılmıştı. Bu kısımda ise bir başka önemli kavram olan veri modelleme üzerinde durulacaktır. Veri modelleme ne tipte veriye gereklilik duyuoduğu ve verilerin ne şekilde düzenleneceği üzerine odaklanmaktadır. Veri modeli, bir sistemin gereksinimlerine bağlı olarak gerçekleştirilecek veritabanı sistemini, kullanıcıların anlamasına yardımcı olmaktadır. Ayrıca, veritabanı uygulayıcılarının bilgi ihtiyaçlarına uygun veritabanı sistemi geliştirmelerine olanak sağlamaktadır. Veri modeli, Resim 5.1'de görüldüğü üzere kullanıcılar ile iletişimde kritik bir araç olarak hizmet vermektedir. Bununla beraber sistem geliştiricilere veritabanı sistemi için bir **kavramsal taslak** olarak hizmet vermektedir. Aynı zamanda, bir sistem içinde yer alan **veri yaşam döngüsünün** farklı fazlarında izleyen kesimde ele alınacağı gibi faydalı amaçlara hizmet etmektedir.

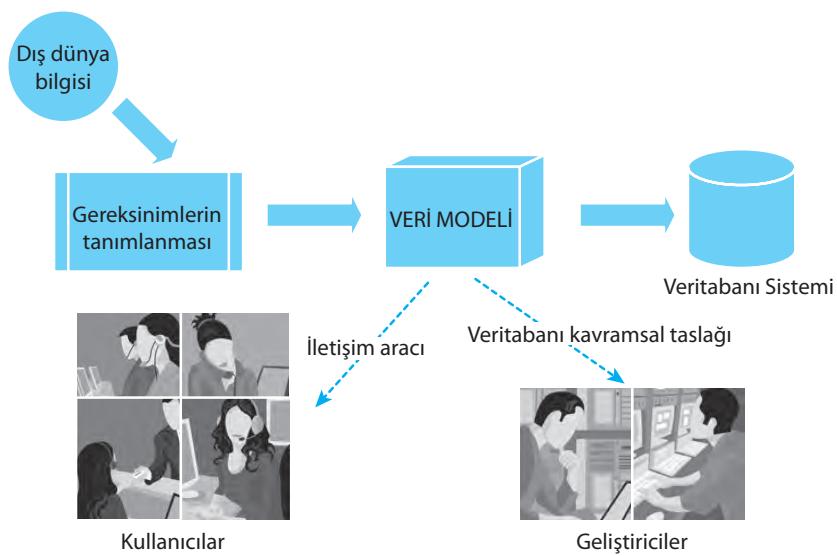
**Kavramsal Taslak (Blueprint):** Fizikal bir nesnenin nasıl yapılacağını ya da oluşturulacağını gösteren ayrıntılı planlardır.

**Veri Yaşam Döngüsü (Data Life Cycle):** Sistem içinde gerekli olduğuna karar verilen verilerin dış dünyadan toplanmasından, eskimesine veya veriye ihtiyacın ortadan kalkmasına kadar geçen süreçtir.

**Resim 5.1**

*İletişim Aracı ve Veritabanı Kavramsal Taslağı Olarak Veri Modeli*

*Kaynak: Ponniah, P.,(2007), s. 5.*



### Veri Yaşam Döngüsü

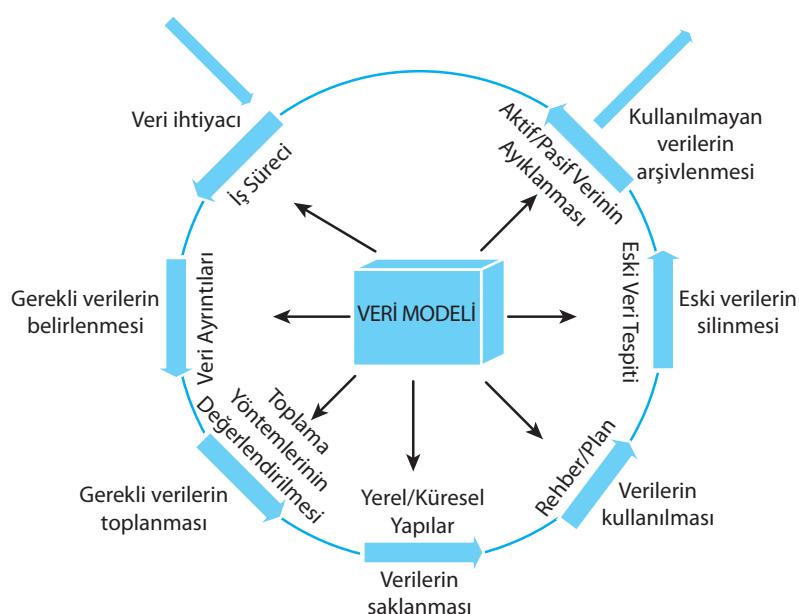
Bir sistem içinde verinin bulunduğu fazları inceleyelim. İlk olarak veri ihtiyacı sistem içerisinde farklı iş süreçlerinin gerçekleştirilmesi amacıyla ortaya çıkar. Daha sonra tam olarak hangi verilerin gerekli olduğu belirlenir. Belirlenen veriler toplanır ve veritabanı sistemlerinde depolanır. Sonraki aşamalarda veri, depolama ortamlarından okunur, istenilen farklı şekillerde birleştirilir ve değiştirilir. Bir süre sonra bazı veriler arşivlenir ve başka yerlerde saklanırlar. Bazı verilerin kullanışlığını kaybetmesinin ardından ilgili veri bileşenleri veritabanı sistemlerinden silinirler. Şekil 5.7, bir sistem içerisindeki veri yaşam döngüsünün fazlarını ve farklı fazlardaki veri modeli etkileşimlerini göstermektedir.

Şimdi veri yaşam döngüsünün her bir fazını ve her fazda veri modelinin sisteme ne şekilde katkı verdiği ve yardımcı olduğunu inceleyelim.

**Şekil 5.7**

*Sistem İçerisinde Veri Yaşam Döngüsü Fazları ve Veri Modeli Etkileşimi*

*Kaynak: Ponniah, P., (2007), s. 8.*



## **Veri İhtiyacı**

Sistem içindeki farklı iş süreçlerinin gerçekleştirilmesi için veri ihtiyacı bulunur. Örneğin bir hastane bilgi sisteminde hasta muayene işlemi için hasta özlük bilgilerine, hasta muayene ve tetkik bilgilerine ihtiyaç duyulur. Fatura üretmek için muayenede yapılan işlemler ve hasta sosyal güvenlik bilgileri ile ilgili veriler gereklidir. Böylece, bu fazda sistem için veri ihtiyacı olduğu anlaşılır. Üst seviye *kavramsal veri modeli* farklı iş süreçleri ve bu süreçlerde yaratılmış kullanılabilecek veriler için kullanışlıdır.

## **Gerekli Verilerin Belirlenmesi**

Veri ihtiyacı olduğu anlaşıldıktan sonra iş süreçlerinin gerçekleştirilmesi için hangi veri bileşenlerinin gerekli olduğu belirlenmek zorundadır. Bu fazda, çok farklı türdeki verilerden hangilerinin gerçekten gerekli hangilerinin gereksiz ya da fazla olduğu belirlenir. İhtiyaç duyulan verilere ait tüm gerekli ayrıntılar, veri modeli içinde keşfedilir ve belgelendirilir.

## **Gerekli Verilerin Toplanması**

Gerekli olan verilerin belirlenmesi fazından sonra iş süreçleri için ihtiyaç duyulan verilen toplanması ve önemsiz verilerin ayıklanması bu fazda gerçekleşmektedir. Veri toplama denemeleri ve toplama yöntemleri, veri modeli yardımıyla araştırılır ve değerlendirilir.

## **Verilerin Saklanması**

Toplanan veriler uygun depolama yöntemleri kullanılarak veritabanında saklanmalıdır. Veriye erişim ve kullanım için kullanıcı ihtiyaçlarına en uygun depolama yöntemi ve ortamı dikkate alınmalıdır. Veri modeli bu fazda veri deposu bileşenlerinin bir araya getirilmesine yardımcı olmaktadır. Veri modelinin her parçası, yerel veri yapısını belirler ve bu yapıların tamamının bir araya gelmesi sonucunda genel veri depolama yapısı ortaya çıkar.

## **Verilerin Kullanılması**

Verilerin toplanmış ve saklanmış olması kullanılabileceği anlamını taşımaktadır. Bu aynı zamanda veri yaşam döngüsünün esas amacıdır. Bu fazda, iş süreci boyunca farklı veri bileşenleri bir araya getirilerek kullanım için alınmakta, değiştirilmekte ve yeni veriler derlenip eklenmektedir. Veri modeli bu fazda, verilerin birleştirilmesi ve kullanılması açısından bir rehber ve plan olarak hareket eder.

## **Eski Verilerin Silinmesi**

Bir süre sonra, saklanan verilerin belirli bir miktarı eskiyebilir ve geçerliliğini yitirebilir. Veri bu nedenle bir daha hiç faydalı olmaz ve işlemler tarafından erişilmez. Örneğin, hasta kullanılan tetkik istekleri, tahlil malzemeleri vb. ve faturaların yasal zorunluluk sürelerinden sonra saklanmalarına gerek yoktur. Sistem belli bir süre sonunda bu tür verileri silmeye karar verebilir. Eski verilerin silinmesi süreci ayrı bir işlem olarak devam eder ve belirli veriler silinmek üzere bu işlem sürecine girer. Bu fazda veri modeli, çeşitli verileri belirtlen süreler sonunda tespit edip güvenli bir şekilde silmek amacıyla kullanılmaktadır.

## **Kullanılmayan Verilerin Arşivlenmesi**

Bazı veriler kullanıcıları üzerinden uzun zaman geçmesine rağmen yararlı bilgi taşıyabilirler. Örneğin hasta muayene, tanı ve tedavi bilgileri hastalığın gelecekteki seyrini belirlemek için kullanılabilir. Geçmiş tarihli bu tür veriler sistem veri ambarı için faydalıdır. Bu tür faydalı veriler mevcut veritabanı sisteminden kaldırılarak ayrı bir veri saklama birimi içinde depolanır. Bu fazda veri modeli, veri depolama, verinin orijinal ve son yerini göstermek ve aktif bölümde arşivlenmiş depoya hareketin izlenebilmesi amacıyla kullanılır.

Veri yaşam döngüsü fazlarında, veri modellemenin sistem içerisinde her iş sürecinde neden gerekli olduğu açık bir şekilde gösterilmektedir. Peki, her fazda ihtiyaç duyulan ve yardımcı olan veri modeli aynı mıdır? Bir sonraki kısımda sizlere bu konu hakkında bilgi verilmektedir.

## FARKLI BİLGİ SEVİYELERİNDE VERİ MODELLEME

Veri modelinin birinci amacı, kullanıcılarla iletişim kurmaktır. Bu durumda, kullanıcılar için geliştirilecek olan sistem içinde yer alan bileşenlerin tarif edilmesi gerekmektedir. Ayrıca, farklı bileşenler arasındaki bağlantıların açıklanması ve geliştirilen sistemdeki asıl veri sisteminin kullanıcılar tarafından kolay anlaşılması sağlanmalıdır. Bunun için veri modelinin karmaşıklıktan uzak olması gerekmektedir. Veri yapıları teknik anlamda kullanıcılarından gizli olmalıdır. Veri modeli içinde fiziksel depolama ile ilgili olarak herhangi bir bilgi bulunmamalıdır.

Veri modelinin ikinci önemli amacı veritabanı sistemleri için kavramsal taslak olarak kullanılmıdır. Veritabanı uygulayıcı ve geliştiricileri her bir veri modelini tek tek ele alarak veritabanı sistemini tasarlamak ve yaratmak için kullanmaktadır. Veri modelinin kavramsal taslak olarak kullanılabilmesi için modelin veri yapısı hakkında da ayrıntılı bilgi içermesi, veriler arasındaki ilişkileri göstermesi gerekmektedir. Veritabanı sistemini kurmak için verinin fiziksel olarak nasıl depolanacağını belirlemek gerekmektedir. Ayrıca veri erişim ve kullanımının ne şekilde gerçekleşeceği de tanımlanmalıdır. Bu işlemlerin gerçekleştirilebilmesi için, veri modeli içinde daha karmaşık ayrıntıların mevcut olması gerekmektedir.

Veri modelinin bir düzeyde genel olması gerekenken başka bir düzeyde ayrıntılı olarak verilmesi gerekmektedir. Bunun anlamı bilgi gösterimlerinin farklı paydaşlar için farklı düzeylerde yapılmasının zorunlu olduğunu düşündür. Bu amaçla; *harici, kavramsal, mantıksal ve fiziksel* olmak üzere dört temel bilgi düzeyi tanımlanmaktadır. Buna göre her bilgi düzeyinde farklı veri modeli kullanılmaktadır.

### Harici Veri Modeli

Harici veri modeli, geliştirilen sistemin kullanıcılarının bakış açısından veritabanı sisteminin tasvir edilmesidir. Bu model kullanıcı toplulukları ile iletişimde bulunmak amacıyla kullanılmaktadır. Birbirinden ayrı her bir kullanıcı grubu kendileriyle ilgili özel işlevleri gerçekleştirmek için kullanacakları veri öğeleri ile ilgilidirler. Bu kullanıcı gruplarının ilgili oldukları veri öğeleri kümesi, sistemdeki belirli bir kullanıcı grubunun harici veri modelinin parçalarını oluşturur.

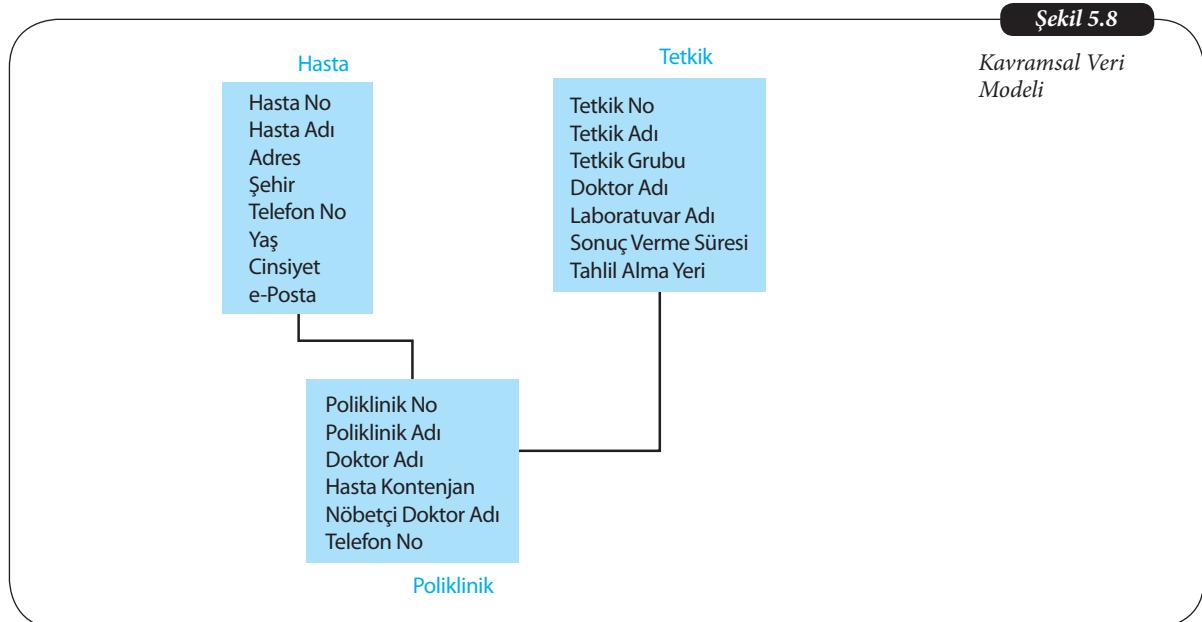
**Tablo 5.2**  
*Harici Veri Modeli*

Kullanıcı Grubu	Harici Veri Modeli
Hasta Kayıt Kabul	Hasta No, Hasta Adı, Adres, Şehir, Telefon No, Yaş, Cinsiyet, e-posta
Laboratuvar İşlemleri	Hasta Adı, Tetkik Adı, Tetkik Grubu, Sonuç Verme Süresi, Doktor Adı, Poliklinik Adı, Laboratuvar Adı
Muayene İşlemleri	Poliklinik Adı, Doktor Adı, Hasta Adı, Tetkik Adı, Tanı, Laboratuvar Adı, Hasta Kontenjanı

Örnek olarak, hastane bilgi sisteminde; hasta kayıt kabul, laboratuvar işlemleri ve poliklinik işlemleri olmak üzere üç kullanıcı grubuna ait minimum veri ögesi kümesi Tablo 5.2'de görülmektedir.

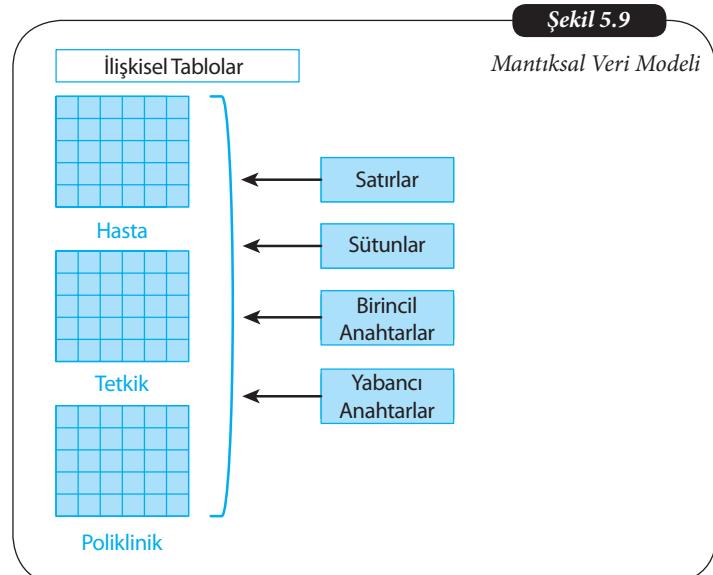
## Kavramsal Veri Modeli

Kavramsal veri modeli yüksek ve genel seviyede bulunur ve esas olarak kullanıcı topluluğu ile iletişim aracı amaçlı kullanılmaktadır. Model içinde, veri yapısı veya donanım ve veritabanı yazılımı için herhangi bir bilgi yer almamaktadır. Asıl veritabanı sisteminin, ilişkisel veritabanı sistemi ya da başka bir türde veritabanı sistemi olarak uygulamaya alınacağı bilgisi de model içinde verilmemektedir. Bununla birlikte, model tam ve yeterli bileşenleri içermek suretiyle geliştirilecek sistemin bilgi gereksinimlerinin gerçek bir gösterim şekli olmaktadır. Şekil 5.8 bir kavramsal veri modeli örneğini göstermektedir. Tablo 5.2'de verilen harici veri modeli kavramsal veri modelinde hasta, tetkik ve poliklinik iş nesneleri olarak gösterilmektedir. Kavramsal veri modelinin oluşturulmasında kullanılan **varlık ilişki diyagramları**, varlık ilişki diyagramları bileşenleri ve bu diyagramların nasıl hazırlanacağı bu ünitenin son kısmında anlatılmaktadır.



## Mantıksal Veri Modeli

Mantıksal veri modeli bir anlamda harici veri modelinin tüm parçalarının bir araya getirilmesidir. Tablo 5.2'de örnek olarak verilen harici veri modelinde üç kullanıcı grubu yer almaktadır. Bu üç kullanıcı grubu için mantıksal veri modeli tüm bilgi gereksinimlerinin birleşimini göstermektedir. İlişkisel veritabanı sistemleri için mantıksal model, bilgi gereksinimlerini Şekil 5.9'da görüldüğü gibi iki boyutlu bir tablo olarak ifade etmektedir.



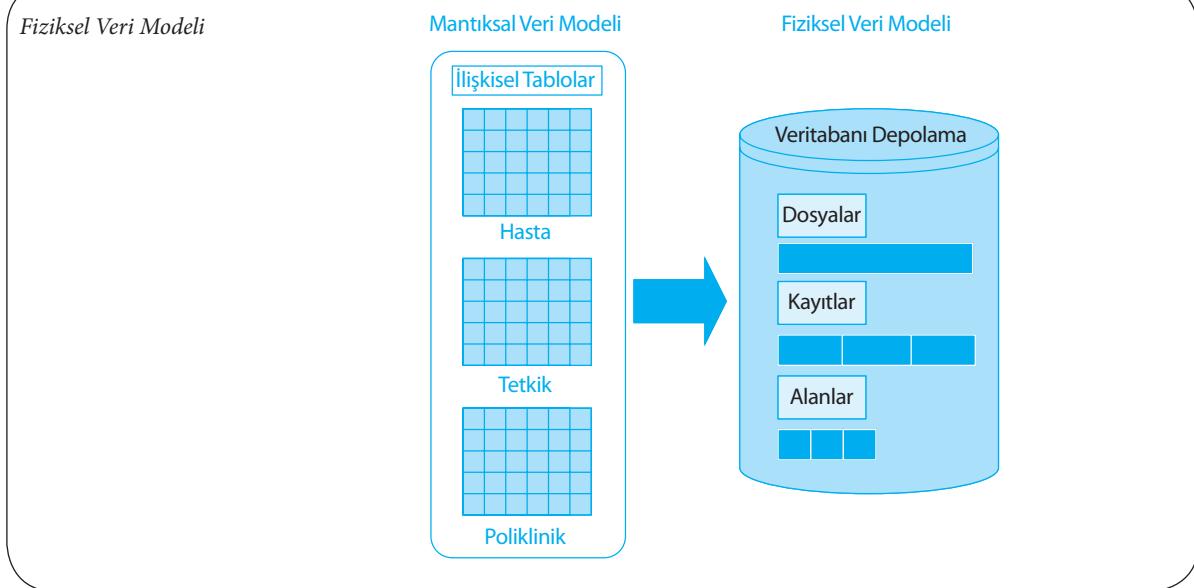
**Varlık İlişki Diyagramları**  
(Entity Relationship Diagram):  
Kısaca ER olarak isimlendirilen diyagramlar, ilişkisel veritabanlarının tasarılarında verinin kavramsal gösterimi amacıyla kullanılır.

## Fiziksel Veri Modeli

Fiziksel veri modeli, kullanıcı grupları ile iletişim aracı olarak kullanımını en düşük veri modelidir. Birinci amaç, veritabanı sistemi geliştirici ve uygulayıcıları için kavramsal taslaç olarak kullanılmıştır. Fiziksel veri modelinin içeriği ayrıntılar kullanıcıların kolaylıkla anlayabileceği ve yorumlayabileceği bilgi olmaktan çok uzaktır. Model, çok fazla karmaşık ayrıntı içermektedir. Geliştirilecek sistemde kullanılacak *Veritabanı Yönetim Sistemi* (VTYS) ve donanım ortamı hakkında belirli bilgiler bulundurmaktadır.

Fiziksel veri modelinin; dosyaları, veri bloklarını, kayıtları, alanları ve benzeri bilgi gerektirdimlerini nasıl temsil ettiği Şekil 5.10'da görülmektedir. Model, çok karmaşık ayrıntıları içeren en düşük seviyede veri gösterim modelidir.

**Şekil 5.10**



## VARLIK İLİŞKİ DİYAGRAMLARI

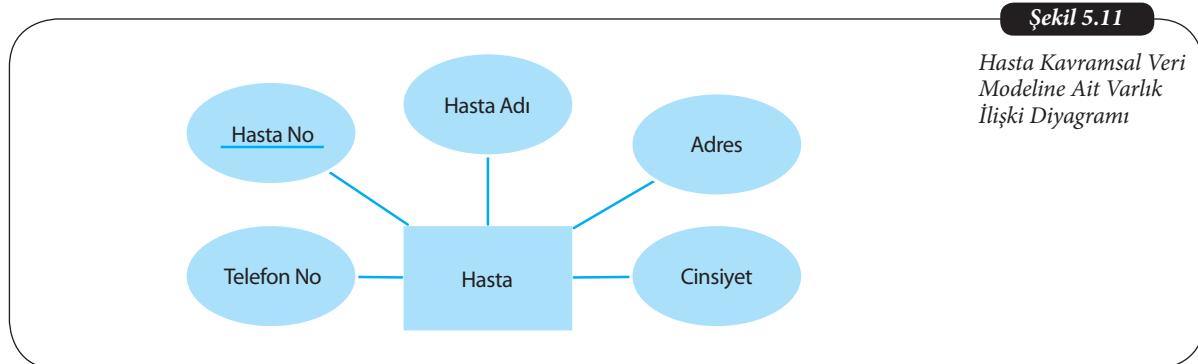
Kavramsal veri modeli gösterimi için kullanılan en yaygın yöntemlerden biri varlık ilişkili diyagramlardır. Varlık ilişkili diyagramları, veritabanı yönetim sistemlerinden ayrı olarak modelleme yapılmasına yardımcı olmaktadır. Ayrıca, iş nesneleri arasındaki ilişkilerin tanımlanmasına da yardımcı olmaktadır.

Varlık ilişkili diyagramlarında; varlık, öznitelik ve ilişki olmak üzere üç temel öğe bulunmaktadır.

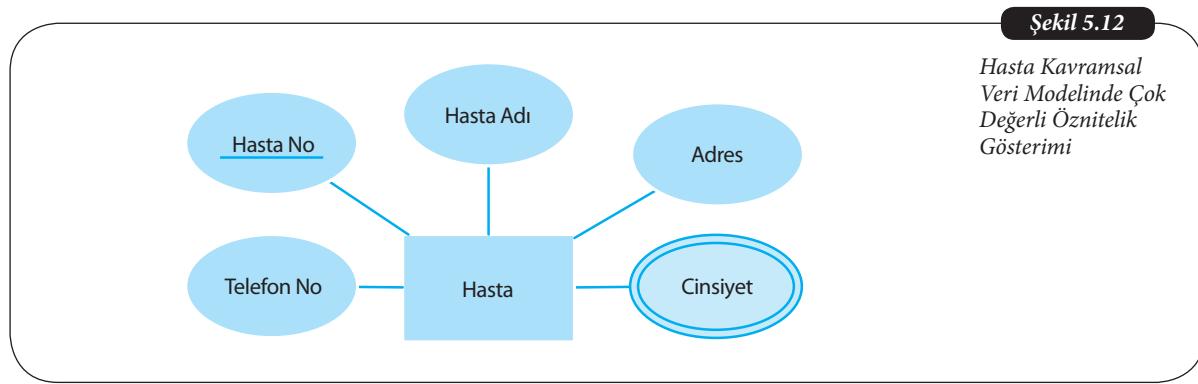
*Varlık*, diyagramın ve dolayısıyla kavramsal veri modelinin en temel öğesidir. Benzerlerinden çeşitli özellikler yardımıyla ayrılan ve var olan tüm nesneler varlıktır. Örneğin, hastane bilgi sisteminde hasta, doktor, poliklinik, laboratuvar, tetkik vb. birer varlık olarak tanımlanmaktadır. Birden fazla varlığın oluşturduğu kümeye varlık kümesi adı verilmektedir. Kavramsal veri modeli içindeki varlık kümeleri diyagramda içinde varlığın adının yazılı olduğu bir dörtgen sembol olarak gösterilmektedir. Veritabanı sistemleri açısından bakıldığında varlık kümesi tabloya karşılık gelirken varlık kümesindeki her bir varlık da kayıtlara karşılık gelmektedir.

*Öznitelik*, varlıkların sahip olduğu her bir özellik öznitelik olarak ifade edilmektedir. Örneğin; hasta no, hasta adı, telefon no, yaş, e-posta ve cinsiyet bilgileri hasta varlığının öznitelikleridir. Diyagramda öznitelikler, içinde öznitelik açıklamasının yazılı olduğu oval semboller ile ifade edilmektedir. Öznitelikler ilgili varlığa bir çizgi ile birleştirilir. Veritabanı

sistemleri açısından bakıldığından öznitelikler tabloların her bir sütununa karşılık gelmektedir. Bir özniteliğin değeri tanımladığı varlıklar içinde benzersiz bir değer taşıyorsa bu öznitelik **anahtar öznitelik** olarak belirlenir. Anahtar öznitelik diyagramda öznitelik adının altı çizilerek gösterilir. Örneğin her bir hastayı birbirinden ayırt etmek için hasta no. (vatan-daşlık no.) özniteliği, anahtar öznitelik olarak belirlenebilir. Veritabanı sistemleri açısından bakıldığından, aynı tablo içinde aynı değerde anahtar özniteliğe sahip iki varlık bulunamaz. Şekil 5.11'de hasta kavramsal veri modeline ait varlık ilişki diyagramı görülmektedir.



Bir öznitelik, Şekil 5.11'deki özniteliklerde görüldüğü gibi tek bir değere sahip olabiliyor-sa tek değerli öznitelik olarak isimlendirilir. Örneğin yukarıda bulunan diyagrama göre tüm öznitelikler tek değer almaktadır. Bazı durumlarda, bir öznitelik birden fazla değer içerebilir. Örneğin yine Şekil 5.11'deki model göz önüne alınacak olursa bir hastanın cinsiyet özniteliği kadın veya erkek olmak üzere iki farklı değerden birini alabilecektir. Bu şekilde bir kategori ayrimına gidilen sistemlerde hasta varlık kümesinin cinsiyet özniteliği **çok değerli bir özniteliktir** ve çift çizgili oval simbol ile ifade edilir. Bu durum Şekil 5.12'de görülmektedir.



**Şekil 5.8'de verilen, hastane bilgi sistemindeki hasta, tetkik ve poliklinik kavramsal veri modellerine ait varlık ilişki diyagramını çiziniz.**

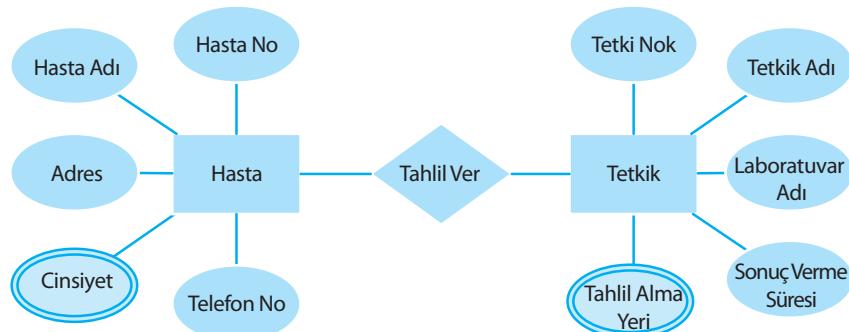


SIRA SİZDE

**İlişki**, varlık ilişki diyagramındaki farklı varlıklar arasında kurulan fizikal ve mantıksal bağlantıları temsil eden yapılardır. Örneğin, hasta ve tetkik ayrı varlık kümeleridir. Fakat hasta istenen bir tetkik için tahlil örneği verebilmektedir. İki varlık arasında "tahlil ver" ilişkisi bulunmaktadır. Bu ilişki varlık ilişki diyagramı içinde baklava dilimi simbolü ile temsil edilir ve içine varlıklar arasında gerçekleştirilen ilişkiye ait eylem yazılır. Baklava dilimi simbolü, ilişkili varlıklara düz çizgi ile bağlanmaktadır. İki varlık kümesi arasında birden fazla ilişki bulunabilir. Şekil 5.13'te, hastane bilgi sistemi kavramsal veri modelindeki hasta ve tetkik varlıkları arasındaki "tahlil ver" ilişkisi varlık ilişki diyagramı ile gösterilmektedir.

**Şekil 5.13**

*Hasta ve Tetkik Varlıklarını Arasındaki "Tahlil Ver" İlişkisinin Gösterimi*



KİTAP



Turgut Özseven'in, *Veritabanı Yönetim Sistemleri* (Bursa: Ekin Yay., 2012) adlı kitabındaki İlişkisel Veritabanının Kavramsal Tasarımı (ss. 46-51) bölümünde varlık ilişki diyagramlarıyla ilgili bilgileri bulabilirsiniz.

**Şekil 5.14**

*Varlık İlişki  
Diyagramı Sembol  
ve Açıklamaları*

Sembol	Açıklama
	Varlık Kümesi
	Öznitelik
	Anahtar Öznitelik
	İlişki
	Çok Değerli Öznitelik

Şekil 5.14'te varlık ilişki diyagramlarında kullanılan semboller ve açıklamaları yer almaktadır.

SIRA SİZDE



4

Bir hastane bilgi sisteminde hasta ile poliklinik arasındaki "muayeneye al" ilişkisini gösteren varlık-ilişki modelini çiziniz.

İNTERNET



<http://creately.com/diagram-community/all?term=erd> bağlantısından çevrimiçi uygulama yardımıyla hazırlanmış olan varlık ilişki diyagramı şablonları üzerinde çalışabilirsiniz.

## Özet



*Sistem geliştirme yaşam döngüsünün analiz aşamasında, kullanım durumların bulunması gerekliliğini açıklamak*

Kullanım durum modellemesi kullanımını, projenin başarıya ulaşmasındaki kritik faktörlerden biri olan kullanıcı katılımını kolaylaştırır ve teşvik eder. İşlevsel gereksinimleri belirleyen ve izlenebilir olmalarını sağlayan bir araç sunan kullanım durum diyagramları, geliştirilecek sistemin kapsamının yönetilebilir alt parçalara ayırtılmasına yardımcı olur. Kullanım durum modellemesi, sistem içerisindeki farklı paydaşların kolaylıkla anlayabileceği ortak bir dil sağlayarak, etkinliklerin tanımlanmasına, atanmasına, izlenmesine, kontrol edilmesine ve yönetilmesine olanak sağlar. Hem kullanıcı yardım sistemi ve kılavuzlarının hem de sistem geliştirme raporlarının belgelenmesine, test planlarının ve test durumlarının tanımlanmasına temel oluşturur. Son olarak kullanım durum modellemesi, veri nesneleri veya varlıkların tanımlanması için bir başlangıç noktası oluştururken, aynı zamanda kullanıcı ve sistem arayüzlerinin tasarımları için işlevsel özelliklerin belirlenmesine yardımcı olur ve sistem geliştirme projesinin idaresi için bir çerçeveye sunar.



*Kullanım durum modelleme temel kavramlarını tanımlamak*

Kullanım durum modellemesi ile iki temel çıktı elde edilir. Bunlardan ilki kullanım durum diyagramı, ikincisi ise kullanım durum hikâyesidir. Kullanım durum diyagramlarının kullanım durumlar, aktörler, ilişkiler ve sistem olmak üzere dört temel bileşeni bulmaktadır. Kullanım durumlar, haricî kullanıcıların bakış açısından sistem işlevlerinin tanımlanmasına yardımcı olurken aktörler, belli bir sistem işlevini gerçekleştirmeye yönelik olarak sistemle etkileşime giren kullanıcılar olarak tanımlanırlar. Bağıntı ilişkisi, bir aktör ve kullanım durum arasında gerçekleşen ilişkidir. Bir aktör, en az bir kullanım durum ile bağıntılı olmalıdır. Bir kullanım durumunun başka kullanım durumları ile ilişkilendirilerek, yeni işlevler kazandırılmasına genişletme ilişkisi adı verilmektedir. Ekleme ilişkisinde ana amaç, ortak işlevlerin sistem içindeki tüm bileşenlerde aynı şekilde tanımlanması ve bu işlevlerin tekrar kullanılabilirliğinin sağlanmasıdır. Bir aktörün kendisiyle aynı işlevleri meydana getiren aktörler ile olan ilişkisi kalitim ilişkisi olarak

tanımlanırken aynı işlevleri kullanan birden fazla aktör olduğunda, ortak işlevler ortaya çıkarılarak bu işlevleri kullanan bir soyut aktör oluşturulur. Kullanım durum hikâyesi; kullanım durum ismini, amacını, aktörlerini, ön koşullarını, son koşullarını, tetikleyicileri, ana başarı senaryosu, hata senaryoları ve alternatif senaryoları içeren formlardır.



*Veri yaşam döngüsünü açıklamak*

Sistem içinde gerekli olduğuna karar verilen verilerin, dış dünyadan toplanmasından, eskimesine veya veriye ihtiyacı olan ortadan kalkmasına kadar geçen süreç, veri yaşam döngüsü olarak tanımlanmaktadır. Veri ihtiyacı sistem içerisinde farklı iş süreçlerinin gerçekleştirilmesi amacıyla ortaya çıkar. Daha sonra tam olarak hangi verilerin gerekli olduğu belirlenir. Belirlenen veriler toplanır ve veritabanı sistemle rinde depolanır. Sonraki aşamalarda veri, depolama ortamlarından okunur, istenen farklı şekillerde birleştirilir ve değiştirilir. Bir süre sonra iş süreçleri tarafından uzun süre kullanılmamış fakat geleceğe dönük tahminlerin yapılmasında faydası olacak düşünülen geçmiş tarihli bazı veriler arşivlenir ve başka yerlerde saklanırlar. Bazı verilerin kullanışlığını kaybetmesinin ardından, ilgili veri bileşenleri veritabanı sistemlerinden silinirler.



*Farklı bilgi seviyelerinde kullanılan veri modellerini açıklamak*

Veri modelinin birinci amacı, kullanıcılarla iletişim kurmaktır. Kullanıcılar için geliştirilecek olan sistem içinde yer alan bileşenlerin tarif edilmesi gerekmektedir. Farklı bileşenler arasındaki bağlantıların açıklanması ve geliştirilen sistemdeki asıl veri sisteminin kullanıcılar tarafından kolay anlaşılmasının sağlanmalıdır. Veri modelinin ikinci önemli amacı ise veritabanı sistemleri için kavramsal taslak olarak kullanılmasıdır. Veritabanı uygulayıcı ve geliştiricileri her bir veri modelini tek tek ele alıp veritabanı sistemini tasarlama ve yaratmak için kullanmaktadır. Veri modelinin kavramsal taslak olarak kullanılabilmesi için modelin veri yapısı hakkında da ayrıntılı bilgi içermesi, veriler arasındaki ilişkileri göstermesi gerekmektedir. Bu iki amacada ulaşılması için veri modelinin bir düzeyde genel olması gerekliden başka bir düzeyde ayrıntılı olarak verilmesi gerekmektedir. Bunun anlamı, bilgi gösterimlerinin farklı paydaşlar için farklı düzeylerde yapılmasının zorunlu olduğunu. Bu amaçla; harici, kavramsal, mantıksal ve fiziksel olmak üzere dört temel bilgi düzeyi tanımlanmaktadır ve her bir düzeyde farklı veri modeli kullanılmaktadır. Harici veri modeli, geliştirilen sistemin kullanıcılarının bakış açısından veritabanı sisteminin tasvir edilmesidir. Bu model kullanıcı toplulukları ile iletişimde bulunmak amacıyla kullanılmaktadır. Kavramsal veri modeli yüksek ve genel seviyede bulunur ve esas olarak kullanıcı topluluğu ile iletişim aracı amaçlı kullanılmaktadır. Model içinde, veri yapısı veya donanım ve veritabanı yazılımı için herhangi bir bilgi yer almaz. Mantıksal veri modeli, harici veri modelinin tüm parçalarının bir araya getirilmesi olarak ifade edilebilmektedir. Fiziksel veri modelinin içeriği ayrıntılar kullanıcıların kolaylıkla anlayabileceğii ve yorumlayabileceğii bilgi olmaktan çok uzaktır. Fiziksel veri modeli, çok fazla karmaşık ayrıntı içermektedir. Geliştirilecek sistemde kullanılacak Veritabanı Yönetim Sistemi (VTYS) ve donanım ortamı hakkında belirli bilgiler fiziksel veri modelinde bulunmaktadır.



*Varlık ilişki diyagramlarının oluşturulmasında kullanılan temel öğeleri açıklamak*

Varlık ilişki diyagramları, veritabanı yönetim sistemlerinden ayrı olarak modelleme yapılmasına ve iş nesneleri arasında ilişkilerin tanımlanmasına yardımcı olmaktadır. Varlık ilişki diyagramlarında, varlık, öznitelik ve ilişki olmak üzere üç temel öğe bulunmaktadır. Varlık, diyagramın en temel öğesidir. Benzerlerinden çeşitli özellikler yardımıyla ayrılan ve var olan tüm nesneler varlık ve birden fazla varlığın oluşturduğu kümeye varlık kümesi adı verilmektedir. Veritabanı sistemleri açısından bakıldığına varlık kümesi tabloya karşılık gelirken varlık kümesindeki her bir varlık da kayıtlara karşılık gelmektedir. Öznitelik, varlıkların sahip olduğu her bir özellik olarak ifade edilmektedir. Veritabanı sistemleri açısından bakıldığına öznitelikler tablolardan her bir sütununa karşılık gelmektedir. Bir öznitelik, varlık kümesi içindeki her varlık için tek bir değer alıysa bu öznitelik anahtar öznitelik olarak belirlenir. Bir öznitelik tek bir değere sahip olabiliyorsa, tek değerli öznitelik, birden fazla değer içinden seçilebiliyorsa, çok değerli öznitelik olarak isimlendirilir. Varlık ilişki diyagramındaki farklı varlıklar arasında kurulan fiziksel ve mantıksal bağlantıları temsil eden yapılar ise ilişkiler olarak tanımlanmaktadır.

## Kendimizi Sınayalım

- 1.** Aşağıdakilerden hangisi kullanım durum modellemenin sağladığı faydalardan biri **değildir**?
  - a. İşlevsel gereksinimleri belirleyen bir araç sağlaması
  - b. Geliştirilecek sistemin kapsamının, yönetilebilir alt parçalara ayrıştırılmasına yardımcı olması
  - c. Farklı paydaşların kolaylıkla anlayabileceği ortak bir dil sağlaması
  - d. Gereksinimlerin izlenebilirliğini sağlaması
  - e. Proje kapsamında ihtiyaç duyulanların tahmin edilmesine yardımcı olması
  
- 2.** Aşağıdakilerden hangisi kullanım durum diyagram bileşeni **değildir**?
  - a. Kullanım durumlar
  - b. Sistem sınırı
  - c. Varlıklar
  - d. Aktörler
  - e. İlişkiler
  
- 3.** “Bir aktörün kendisiyle aynı işlevleri meydana getiren aktörler ile olan ilişkisi” olarak tanımlanan kullanım durum ilişkisi aşağıdakilerden hangisidir?
  - a. Kalıtım ilişkisi
  - b. Ekleme ilişkisi
  - c. Genişletme ilişkisi
  - d. Bağıntı ilişkisi
  - e. Birleşim ilişkisi
  
- 4.** Veri yaşam döngüsü fazlarından hangisinde aktif ve pasif verilerin ayıklanması söz konusudur?
  - a. Veri ihtiyacı fazı
  - b. Gerekli verilerin belirlenmesi fazı
  - c. Gerekli verilerin toplanması fazı
  - d. Kullanılmayan verilerin arşivlenmesi fazı
  - e. Verilerin saklanması fazı
  
- 5.** Veri yaşam döngüsü fazlarından hangisinde oluşturulan veri modeli bir rehber olarak kullanılır?
  - a. Veri ihtiyacı fazı
  - b. Gerekli verilerin belirlenmesi fazı
  - c. Gerekli verilerin toplanması fazı
  - d. Kullanılmayan verilerin arşivlenmesi fazı
  - e. Verilerin kullanılması fazı
  
- 6.** Veri yaşam döngüsü fazlarından hangisinde ihtiyaç duyulan verilere ait tüm gerekli ayrıntılar veri modeli içinde keşfedilir ve belgelendirilir?
  - a. Kullanılmayan verilerin arşivlenmesi
  - b. Tekrarlanamayan okuma
  - c. Gerekli verilerin toplanması
  - d. Gerekli verilerin belirlenmesi
  - e. Verilerin kullanılması
  
- 7.** Geliştirilen sistemin kullanıcılarının bakış açısından, veritabanı sisteminin tasvir edildiği veri modeli aşağıdakilerden hangisidir?
  - a. Dahili veri modeli
  - b. Harici veri modeli
  - c. Kavramsal veri modeli
  - d. Mantıksal veri modeli
  - e. Fiziksel veri modeli
  
- 8.** Kullanıcı grupları ile iletişim aracı olarak kullanımı **en düşük** veri modeli aşağıdakilerden hangisidir?
  - a. Dahili veri modeli
  - b. Harici veri modeli
  - c. Kavramsal veri modeli
  - d. Mantıksal veri modeli
  - e. Fiziksel veri modeli
  
- 9.**  Yanda görülen simbol varlık ilişki diyagramlarında aşağıda verilen hangi bileşeni ifade etmektedir?
  - a. Varlık kümesi
  - b. İlişki
  - c. Öznitelik
  - d. Çok değerli öznitelik
  - e. Anahtar öznitelik
  
- 10.**  Yanda görülen simbol varlık ilişki diyagramlarında aşağıda verilen hangi bileşeni ifade etmektedir?
  - a. Varlık kümesi
  - b. İlişki
  - c. Öznitelik
  - d. Çok değerli öznitelik
  - e. Anahtar öznitelik

## Kendimizi Sınavalım Yanıt Anahtarı

1. e Yanınız yanlış ise “Kullanım Durum Modelleme Temel Kavramları” konusunu yeniden gözden geçiriniz.
2. c Yanınız yanlış ise “Kullanım Durum Modelleme Temel Kavramları” konusunu yeniden gözden geçiriniz.
3. a Yanınız yanlış ise “Kullanım Durum Modelleme Temel Kavramları” konusunu yeniden gözden geçiriniz.
4. d Yanınız yanlış ise “Veri Modelleme ve Veri Yaşam Döngüsü” konusunu yeniden gözden geçiriniz.
5. e Yanınız yanlış ise “Veri Modelleme ve Veri Yaşam Döngüsü” konusunu yeniden gözden geçiriniz.
6. d Yanınız yanlış ise “Veri Modelleme ve Veri Yaşam Döngüsü” konusunu yeniden gözden geçiriniz.
7. b Yanınız yanlış ise “Farklı Bilgi Seviyelerinde Veri Modelleme” konusunu yeniden gözden geçiriniz.
8. e Yanınız yanlış ise “Farklı Bilgi Seviyelerinde Veri Modelleme” konusunu yeniden gözden geçiriniz.
9. e Yanınız yanlış ise “Varlık İlişki Diyagramları” konusunu yeniden gözden geçiriniz.
10. b Yanınız yanlış ise “Varlık İlişki Diyagramları” konusunu yeniden gözden geçiriniz.

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Bir hastane bilgi sisteminde yapılan muayeneden doğrudan fayda sağlayacağından dolayı *Hasta* birincil iş aktördür. *Doktorlar* hastayı mayene ettiğlerinde ve muayene bulgularını ve tanıyi sisteme girdiklerinde birincil sistem aktördürler. İstenen tetkiklerin sonuçlarını sisteme giren *laboratuvar teknisyeni* ya da tetkiki gerçekleştirerek tetkik sonuçlarını otomatik olarak bilgi sistemine yükleyen *donanım* harici sunucu aktör olurken yapılan işlemlerin faturalarını saklayan *muhasebe birimi* harici alıcı aktördür.

### Sıra Sizde 2

**Kullanım Durum Adı:** Hasta Kayıt

**Amaç:** Bu kullanım durum hastaneye gelen hastanın muayene alınması için ilgili polikliniği kayıt eder.

**Aktörler:** Kayıt Kabul Çalışanları, Hasta

#### Ön koşullar:

1. Hasta bilgilerinin hastane bilgi sistemine kayıtlı olması gerekmektedir.
2. İlgili polikliniğin bugün hasta kabul yapabiliyor olması gerekmektedir.

#### Son koşullar:

1. Söz konusu hasta eğer kontenjan izin veriyor ise poliklinik sırasına kayıt edilmiştir.
2. Söz konusu hasta kontenjan izin vermediğinde başka bir güne randevu verilmiş olması gerekmektedir.
3. Hastaya barkod basılmış olmalıdır.
4. Hastaya fatura basılmış olmalıdır.
5. Kayıt kabul ekranı yeni hasta kayıt etme ekranına dönmelidir.

**Tetkileyiciler:** Kayıt Kabul

#### Ana Başarı Senaryosu:

1. Hasta numarasını sisteme gir.
2. Hasta bilgileri sisteme kayıtlı ise bilgileri ekrana getir, değilse hasta bilgilerinin girildiği ekranı göster.
3. Poliklinik kontenjan bilgisini al.
4. Kontenjan açıksa hastayı polikliniği kaydet.
5. Fatura çıktısını al.
6. Hasta için barkod bas.
7. Yeni hasta kaydı için bekle.

#### Hata Senaryosu:

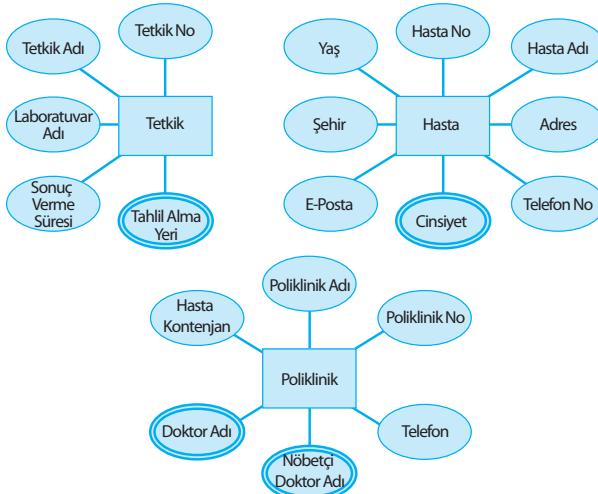
1. Poliklinik açık değil, hasta kabul etmiyor.
2. Poliklinik kontenjanı dolu.
3. Eksik ya da hatalı hasta bilgisi bulunuyor.

## Yararlanılan ve Başvurulabilecek Kaynaklar

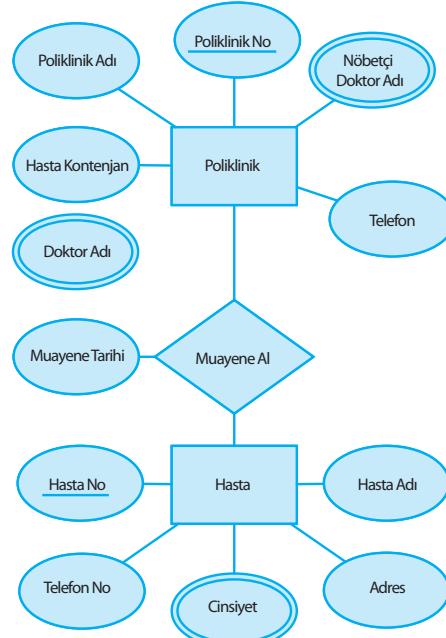
### **Alternatif Senaryolar:**

1. Poliklinik açık değilse kayıt kabule hata bildirimi göster ve hastaya en yakın poliklinik çalışma zamanının bildirileceği ekranı göster.
2. Kontenjan dolu ise hastaya en yakın poliklinik çalışma zamanı için randevu verme ekranını göster.
3. Eksik ya da hatalı bilgileri göster ve kayıt kabul görevlisine bu bilgileri hastadan almasının gerekligini gösteren bir bildirim yap.

### Sıra Sizde 3



### Sıra Sizde 4



Fowler, M., ve Scott, K., (2003). **Rafine UML**, (Çeviren: M. Çömlekçi)1. baskı, Alfa Yayınları, İstanbul.

Özseven T., (2013), **Veritabanı Yönetim Sistemleri 1**, 4. Bası̄, Ekin Basım Yayın, Bursa.

Ponniah, P., (2007), **Data modeling fundamentals: a practical guide for IT professionals**, John Wiley&Sons Inc.

Roth, R.M., Dennis, A., ve Wixom, B.H., (2012). **Systems Analysis and Design, 5th Edition**, John Wiley&Sons Inc.

Satzinger, J.W., Jackson, R.B., ve Burd, S.D., (2012). **Systems Analysis and Design in Changing World, 6th Edition**, Course Technology CENGAGE Learning.

Tegarden, D., Dennis, A., ve Wixom, B.H., (2012). **Systems Analysis and Design with UML, 4th Edition**, John Wiley&Sons Inc.

Whitten, J. ve Bentley, L., (2005). **Systems Analysis and Design Methods, 7th Edition**, McGraw-Hill.

# 6

## Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Süreç modelleme kavramını tanımlayabilecek,
  - 🕒 Veri akış diyagramları oluştururken dikkat edilmesi gereken kuralları açıklayabilecek,
  - 🕒 Mantıksal ve fiziksel veri akış diyagramları arasındaki farkları ayırt edebilecek,
  - 🕒 Süreçlerin mantıksal olarak modellenmesinde kullanılan yöntemleri açıklayabilecek
- bilgi ve becerilere sahip olabileceksiniz.

## Anahtar Kavramlar

- Süreç Modelleme
- Veri Akış Diyagramları (VAD)
- Mantıksal Veri Akış Diyagramı
- Fiziksel Veri Akış Diyagramı
- Yapısal Dil
- Karar Tabloları
- Karar Ağaçları

## İçindekiler

Sistem Analizi ve Tasarımı

Süreç Modelleme ve Analizi

- 
- GİRİŞ
  - VERİ AKIŞ DİYAGRAMLARI VE BİLEŞENLERİ
  - VERİ AKIŞ DİAGRAMLARININ OLUŞTURULMASI
  - MANTİKSAL VE FİZİKSEL VERİ AKIŞ DİYAGRAMLARI
  - SÜREÇLERİN MANTİKSAL MODELLENMESİ

# Süreç Modelleme ve Analizi

## GİRİŞ

Sistem analistleri, geliştirilmesi planlanan sistem içindeki verinin nasıl hareket edeceğini, hangi süreçlerden ya da değişimlerden geçeceğini ve çıktıların neler olacağını kavramsal-laştırmalıdır. Görüşme, anket vb. diğer yöntemler, geliştirilecek olan sistem ile ilgili olarak sözel bir anlatım sağlar. Sistemin görsel olarak kavramsallaştırılması ve betimlen-mesi, gereksinimlerin kullanıcı ve analistler tarafından daha kolay bir şekilde anlaşılma-sına yardımcı olmaktadır.

**Süreç modelleme**, verinin elde edilmesi, işlenmesi, saklanması ve sistem bileşenleri arasında gerçekleşen veri hareketlerinin grafiksel olarak temsil edilmesidir. Günümüzde süreç modelleme amacıyla kullanılan çok farklı modelleme teknikleri bulunmaktadır. Bu üitede, en çok kullanılan süreç modelleme tekniklerinden biri olan **veri akış diyagramları (VAD)** nasıl kullanılacağı ele alınmıştır. VAD'lar sistem içindeki süreçlerin man-tıksal ve fiziksel modellerinin görsel olarak tanımlanmasında kullanılmaktadır. Üitede öncelikle diyagram bileşenleri ve bu bileşenleri temsil etmekte kullanılan sembollerin gös-terimleri üzerinde durulacaktır. VAD'ların oluşturulmasında uyulması gereken kurallar üzerinde durulmuştur. Daha sonra bu semboller kullanılarak, basitten karmaşağa doğru VAD'ların nasıl oluşturulacağı hakkında bilgi verilmiştir. Sistem süreçlerinin VAD yardı-mıyla görsel olarak modellemesinden sonra, süreçlerin iş mantığının modellenmesinde kullanılan yöntemler hakkında bilgi verilmiştir.

**Süreç Modelleme (Process Modeling):** Verinin elde edilmesi, işlenmesi, saklanması süreçlerin ve sistem bileşenleri arasında gerçekleşen veri hareketlerinin grafiksel temsilidir.

**Veri Akış Diyagramı (Data Flow Diagram-DFD):** Sistem içindeki harici varlıklar, süreçler ve veri depoları arasında gerçekleşen veri hareketlerini gösteren grafiksel bir araçtır.

## VERİ AKIŞ DİYAGRAMLARI VE BİLEŞENLERİ

Veri akış diyagramları, SGYD içinde analiz fazında bilginin görsel olarak yapılandırılması amacıyla kullanılan bir araçtır. VAD'lar geliştirilecek olan bilgi sisteminin modellenme-sinde aşağıda belirtilen üç konuda yardımcı olmaktadır:

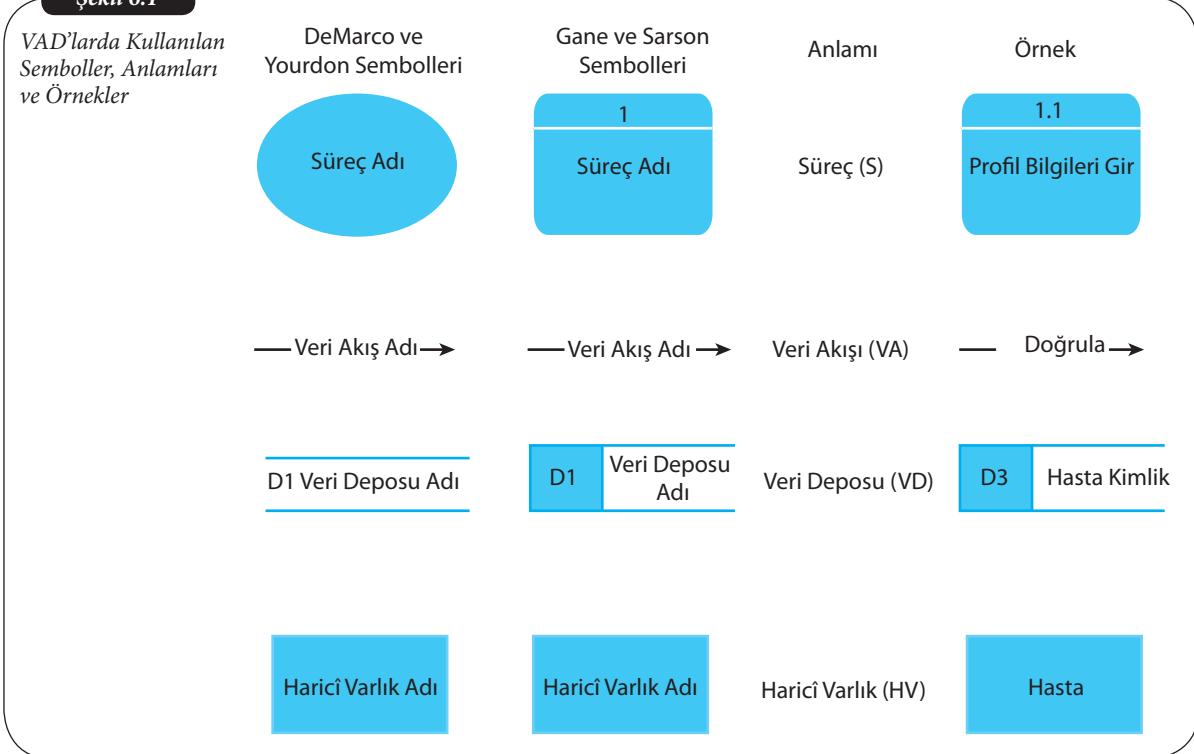
- Verinin nasıl hareket edeceğini gösterir,
- Veri hareketleri arasındaki ilişkilerin neler olduğunu açıklar,
- Verinin belirli bölgelerde nasıl saklanacağını gösterir.

VAD'lar aynı zamanda veriyi değiştiren ve dönüştüren süreçleri de gösterirler. VAD'lar süreçler arasındaki veri hareketlerine yoğunlaştığından dolayı bu diyagramlar aynı za-manda süreç modeli olarak da isimlendirilmektedir.

Sistemin kendisi başlı başına bir iş sürecidir. Bu yüzden VAD kullanımının diğer bir faydası da söz konusu bu iş sürecinin alt süreçlere kolaylıkla ayırtılmasına olanak sağla-masıdır. Bu nedenle VAD, geliştirilmek istenen karmaşık sistemlerin anlaşılması ve işletil-mesi amacıyla mantıksal ve fiziksel süreçlerin modellenmesinde kullanılır.

VAD *haricî varlıklar, veri akışı, süreç ve veri deposu* olmak üzere dört bileşenden oluşur. VAD çiziminde Gane ve Sarson (1977) ile DeMarco ve Yourdon (1977) tarafından geliştirilmiş iki farklı sembol kümesi kullanılmaktadır. Her iki sembol kümesine ait semboller ve anlamları Şekil 6.1'de görülmektedir. Bu ünitede Gane ve Sarson tarafından geliştirilmiş olan sembol kümesi kullanılacaktır.

**Şekil 6.1**



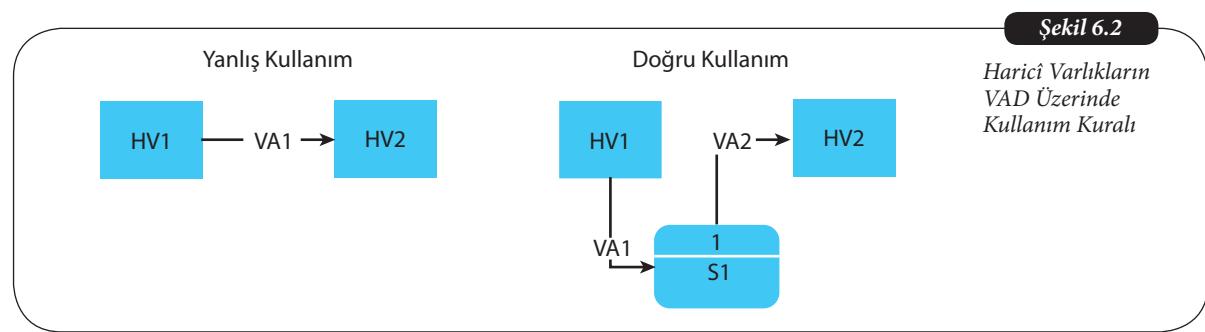
**Haricî Varlıklar (External Entity):** Sistemde bulunan verinin kaynağını ya da verinin gideceği hedefi gösterir.

**Süreç (Process):** Verinin elde edilmesi, dönüştürülmesi, saklanması veya dağıtılması amacıyla gerçekleştirilen işletme ya da bilgi sistemi faaliyetler bütünüdür.

**Haricî varlıklar (HV),** bir kişi, kurum, kurum birimi veya bir sistem olabilir. Haricî varlıklar kullanım durumlarda tanımlanan birincil aktörlere karşılık gelmektedir. Bu varlıklar bazı kaynaklarda arayüz olarak da isimlendirilir. Haricî varlıklar sisteme veri sağlarlar ya da sistemden veri alırlar. Her haricî varlığın bir adı ve açıklaması vardır. Haricî varlıklar, sistem dışında bulunurlar. Bununla birlikte sistemin bulunduğu işletmenin bir parçası olabilirler ya da olmayabilirler. Örneğin, personel veya yönetici gibi sistemden bilgiyi alıp kullanan veya sisteme hangi bilginin gönderileceği kararını veren kişiler haricî varlıklar olarak tanımlanabilir. Diyagram üzerinde veri akışını gösteren bağlantı çizgilerinin birbirini kesmemesi amacıyla aynı haricî varlık birden fazla kullanılabilir. Veri akışını gösteren bağlantı çizgisinin başladığı (bağlantı çizgisinin ok bulunmayan bölümü) haricî varlığa *girdi* ya da *kaynak varlık* adı verilir. Bağlantının bittiği (bağlantı çizgisinin ok bulunan bölümü) haricî varlığa ise *çıktı* ya da *hedef varlık* adı verilir. Haricî varlıkların diyagramda kullanılmasında aşağıdaki kurallara uyulması gerekmektedir.

- Veri bir haricî varlıktan diğerine doğrudan hareket edemez. Eğer bu veri sistem için gerekli ise mutlaka bir süreç üzerinden taşınmalıdır. Aksi durumda VAD üzerinde veri akışı gösterilmmez.
- Haricî varlıklar bir isim veya isim tamlaması ile ifade edilirler.

Haricî varlıkların VAD üzerinde doğru ve yanlış kullanımı Şekil 6.2'de görülmektedir.



**Veri akışı (VA)**, sistem içinde bir yerden diğer bir yere doğru hareket eden bir veriyi ya da mantıksal veri koleksiyonunu (örneğin, “yeni tetkik isteği”, “yeni randevu isteği” vb.) ifade eder. Veri akışı, başlangıç veya bitişinde veri akış yönünü gösteren okların bulunduğu bağlantı çizgileri ile gösterilir. Aynı anda gerçekleşen veri akışları birbirine平行 bağlantı çizgileri ile gösterilir. Bir veri akışı, hasta kayıt barkodu üzerindeki bir veriyi ifade edebileceğ gibi, veritabanı sorgusu sonucu oluşan kayıt kümесini, rapor çıktısının içeriğini veya veri giriş ekranı üzerindeki bir veriyi de ifade edebilir. Veri akışının diyagramda kullanılmasında aşağıdaki kurallara uyulması gerekmektedir.

- i. Veri akışı, semboller arasında sadece tek yönde gerçekleşir. Süreç ve veri deposu semboller arasında güncelleme öncesi yapılan bir okumayı göstermek amacıyla her iki yönde de akış olabilir. Ancak, okuma ve güncelleme genellikle farklı zamanlarda gerçekleştiğinden iki ayrı ok ile ifade edilir.
- ii. Veri akışındaki ayrılma (çatallanma) aynı yerden ayrılan ortak verinin iki veya daha fazla süreç bileşenine, veri deposuna ya da harici varlığa hareket ettiğini gösterir. Genellikle aynı verinin farklı kopyalarının farklı yerlere gittiğini ifade etmek için kullanılır.
- iii. Veri akışındaki birleşme, herhangi iki veya daha fazla farklı süreç, veri deposu veya harici varlıktan tam olarak aynı verinin geldiği anlamına gelir.
- iv. Veri akışı çıkış yaptığı süreç bileşenine doğrudan geri döner. En azından bir sürecin veri akışını işlemesi, başka bir veri akışı üretmesi ve orijinal veri akışının başlangıç sürecine bundan sonra dönmesi gereklidir.
- v. Veri deposuna giden veri akışı, veriyi güncelleme (silme veya değiştirme) anlamına gelir.
- vi. Bir veri deposundan gelen veri akışı, veriyi almak veya kullanmak anlamına gelir.
- vii. Veri akışı isim ya da isim tamlaması olarak tanımlanır. Veri akışları birlikte tek bir paket olarak hareket edebilir. Bu durumda tek bir ok üzerinde birden fazla veri akış ismi bulunabilir.

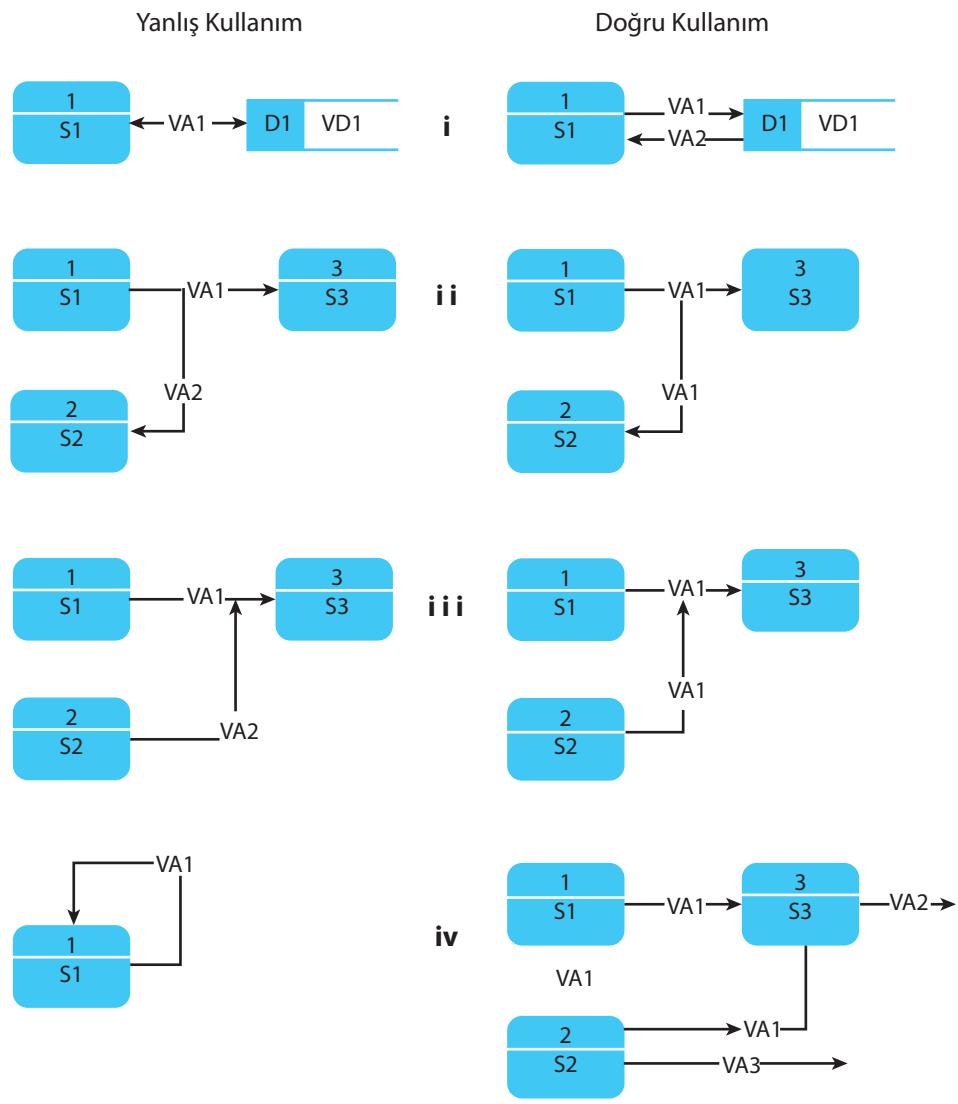
Veri akışının VAD üzerinde doğru ve yanlış kullanımı Şekil 6.3'te görülmektedir.

**Veri Akışı (Data Flow):** Verinin sistem içinde gerçekleştiği ve yönü belli olan hareketidir.

**Mantıksal Veri Koleksiyonu:** Belirli bir mantık altında bir araya getirilmiş birden fazla veriden oluşan yapıdır.

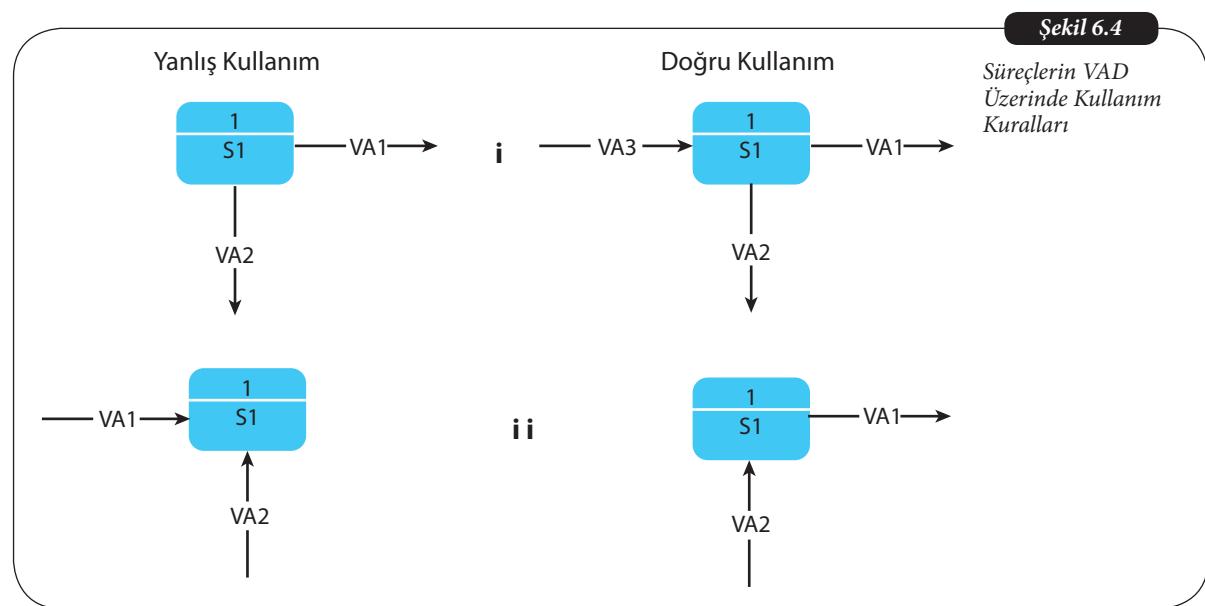
Şekil 6.3

Veri Akışlarının  
VAD Üzerinde  
Kullanım Kuralları



**Süreç (S)**, belirli bir işi gerçekleştirmek amacıyla elle veya bilgisayar ile yürütülen bir etkinlik ya da fonksiyondur. Her süreç, emir kipinde bir yüklem ile bitecek şekilde isimlendirilir (örneğin, "tatkik istek yap", "randevu al" vb.). İsimler kısa olmalıdır ve sürecin ne yaptığına açıklayacak kadar bilgi içermelidir. Genellikle her süreç sadece tek etkinlik gerçekleştirir. Her sürecin mutlaka bir girdi ve bir de çıktı veri akışı bulunur. Süreçler, veri üzerinde değişiklik ya da dönüşüm gerçekleştirmektedir. Bu yüzden süreçte giren veri akışı ve çıkan veri akışı isimleri birbirinden farklı olmak zorundadır. Süreç bileşeninin diyalogda kullanılmasında aşağıdaki kurallara uyulması gerekmektedir.

- Sadece çıktıları olan süreç olamaz. Bir nesne sadece çıktılarına sahip ise bu nesnenin harici girdi varlık olması gereklidir.
  - Sadece girdisi olan süreç olamaz. Bir nesne sadece girdilere sahip ise bu nesnenin harici çıktı varlık olması gereklidir.
  - Süreç isimleri emir kipinde yüklemeler ile sonlanmalıdır.
- Süreç bileşeninin VAD üzerinde doğru ve yanlış kullanımları Şekil 6.4'te görülmektedir.



**Veri deposu (VD)**, sistemdeki verilerin kalıcı olarak bulunduğu yerlerdir. Veri deposu, dosya klasörü, dosyalar ve veritabanı gibi bilgisayar tabanlı olabilir. Ayrıca, dosya dolapları, kayıt defterleri, formlar, bilgisayar çıktısı raporlar, barkodlar vb. fiziksel nesneler de veri depolarına karşılık gelebilmektedir. Veri deposu, müşteriler, hastalar, yönetici, sekreterler ya da tedarikçiler vb. hakkında veri içerebilmektedir. Tüm veri depolarının, en az bir adet giriş veri akış bağlantısı ve benzer şekilde de en az bir adet çıkış veri akış bağlantısına sahip olması gerekmektedir.

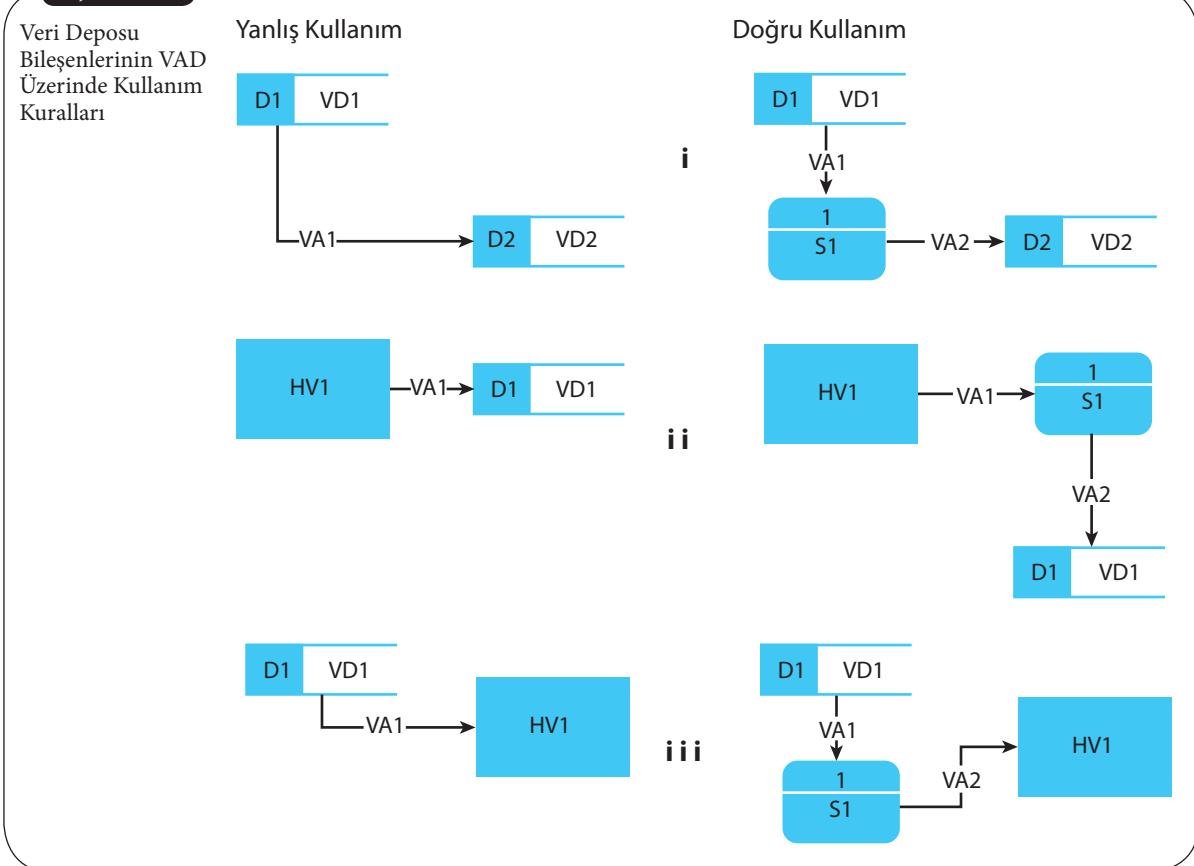
Veri deposu bileşeninin diyagramda kullanılmasında aşağıdaki kurallara uyulması gerekmektedir.

- Veri bir veri deposundan diğer veri deposuna doğrudan hareket edemez. Veri mutlaka bir süreç yardımıyla hareket etmelidir.
- Veri, bir haricî girdi varlığından doğrudan veri deposuna hareket edemez. Veri, haricî girdi varlığından veriyi alan ve aldığı veriyi veri deposuna saklayan bir süreç üzerinden hareket etmek zorundadır.
- Veri, veri deposundan haricî çıktı bir varlığı doğrudan hareket edemez. Veri, veri deposundan veriyi alan ve aldığı veriyi haricî çıktı varlığına gönderen bir süreç üzerinden hareket etmek zorundadır.

Veri deposu bileşeninin VAD üzerinde doğru ve yanlış kullanımları Şekil 6.5'te görülmektedir.

**Veri Deposu (Data Store):** Farklı fiziksel biçimlerde bulunabilen ve sistemdeki verileri saklayan ortamlardır.

Şekil 6.5



### VERİ AKIŞ DİYAGRAMLARININ OLUŞTURULMASI

VAD oluşturmak için birbirini takip eden belirli adımların izlenmesi gerekmektedir. VAD'ların oluşturulmasında **yukarıdan-aşağıya yaklaşımı** kullanılmaktadır. Çoğu iş süreci tek bir VAD ile açıklanamayacak kadar karmaşıktır. Bu amaçla birçok süreç modeli birden fazla VAD içeren VAD kümelerinden oluşmaktadır. İlk VAD, sistem genelinde özet bir bilgi sağlarken, ek VAD'lar kullanılarak tüm iş süreçlerinin her biri için daha fazla ayrıntı sunulmaktadır. Bu yüzden VAD kullanılarak süreç modellemede en önemli özellik; farklı VAD düzeyleri oluşturmaktır. Her bir alt düzey VAD'da, sürecin kapsamı alanı daraltılırken süreçte ait daha fazla ayrıntıya yer verilmektedir.

Sistem analistleri, iş süreçleri ile ilgili olarak VAD'larını oluştururken yukarıdan-aşağıya bakış açısıyla, veri akışlarını aşağıda listelenen yedi adet adımı izleyerek kavramsallaştırır.

- i. İş süreçlerinin bir listesi oluşturulur ve farklı harici varlıklar, veri akışları, süreçler ve veri depoları belirlenir.
- ii. Harici varlıklar ve veri akış yönlerini gösteren **bağlam diyagramı** oluşturulur. Bu diyagram, sisteme ait çok genel bir diyagramdır ve diyagram üzerinde ayrıntılı bir süreç ya da veri deposu bulunmaz.
- iii. İkinci aşamada “düzey 0 VAD” çizilir. Bağlam diyagramda belirtilmiş olan giriş ve çıkışlar sabit kalırken, bağlam diyagramda yer alan süreç üç ile dokuz süreç arasında yeni alt süreçlere ayrıştırılır. Veri depoları da kullanılarak, daha düşük düzeyde yeni veri akışları gösterilir.

**Yukarıdan-Aşağıya Yaklaşım (Top-Down Approach):**  
Veri işleme ve bilgi sıralama stratejisidir. Ayrıştırma (decomposition) olarak da isimlendirilen bu yaklaşımında genel sistem her adında kendini meydana getireceğini alt sistemlere bölünür ve ayrıntılandırılır.

- iv. Düzey 0 VAD içinde yer alan her bir süreç için alt düzey VAD'lar çizilir. Düzey 0 VAD'da yer alan süreçlerin parçalanmasıyla elde edilen diyagamlara *çocuk VAD* ya da *alt VAD* adı da verilmektedir.
- v. Hatalar kontrol edilir ve her bir süreç ile veri akışına verilen isimlerin anlamlı olması sağlanır.
- vi. Elde edilen mantıksal VAD kullanılarak fiziksel VAD elde edilir. El ile gerçekleşen ve bilgisayar ile gerçekleşen süreçler arasında ayrımlar yapılır. Gerçek dosyalar ve raporlamalar, isimleriyle tanımlanır. Sürecin başarı ile tamamlandığını ya da hata meydana geldiğini göstermek amacıyla kontroller eklenir.
- vii. Programlama veya uygulamayı kolaylaştırmak amacıyla, fiziksel VAD üzerindeki diyagram parçaları gruplanabilir.

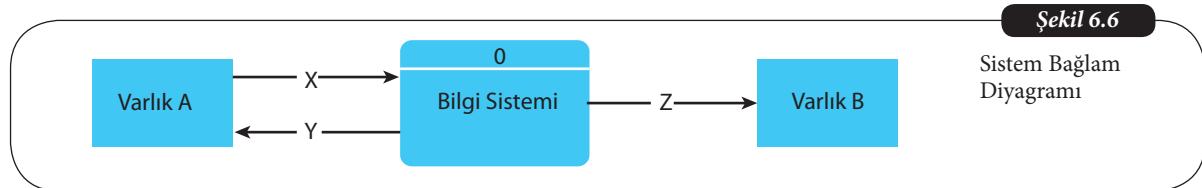
VAD oluşturulurken işletme sistem gereksinimleri harici varlık, veri akış, süreç ve veri depolama olarak dört kategori altında gruplanırlar. Böylelikle işletme gereksinimleri bir liste hâlinde alt gruplara parçalanmış olur. Bu liste, tanımlanan sistemin sınırlarının belirlenmesine yardımcı olacaktır. Temel veri öğeleri listesi elde edildikten sonra **bağlam diyagram** çizimine başlanır.

### Bağlam Diyagram Çizimi

VAD oluşturulmasında yukarıdan-aşağıya yaklaşımın kullanılmasından dolayı diyagramlar genelenle doğru sıralanmaktadır. Bağlam diyagramı adı verilen ilk diyagram, sistem analistlerinin ve kullanıcıların sistem içinde gerçekleşen temel veri hareketlerini anlamalarına yardımcı olur. Fakat genel özellikleri içeriği için sınırlı bilgi sunar. Başlangıç bağlam diyagramı, temel girişleri, genel sistemi ve çıkışları içeren genel bir görünüm sağlar. Tüm harici varlıklar ve bu varlıklara giden veya bu varlıklardan gelen önemli veri akışları bağlam diyagramda gösterilir.

Şekil 6.6'da görüldüğü gibi, bağlam diyagram VAD içinde en üst düzey diyagramdır ve VAD üzerinde 0 olarak etiketlenmiş olan tek süreç tüm sistemi ifade etmektedir. Harici varlıklar sistemin çevresel sınırlarını göstermektedir. Kavramsal olarak sisteme gerçekleşen veri saklama işlemleri tek bir süreç içinde olduğundan bağlam diyagramında veri deposu yer almaz.

**Bağlam Diyagram (Context Diagram):** Sistemin sınırlarını, sisteme etkileşimde bulunan harici varlıklar ve sistem ile harici varlıklar arasındaki temel önemde sahip veri akışlarını gösteren VAD'dır.



### Düzey 0 VAD Çizimi

Bir üst düzey VAD'ların alt düzey VAD'lara ayrıştırılması sonucunda bağlam diyagramının sağladığından daha fazla ayrıntı elde edilebilmektedir. İlk diyagramda belirtilmiş olan giriş ve çıkışlar daha sonraki tüm alt düzey VAD'larda sabit kalmaktadır. Bununla birlikte bir üst düzeyde yer alan diyagramdaki süreçler üç ile dokuz süreç arasında yeni alt süreçlere ayrıstırılır. Alt süreçler veri depolarıyla birlikte daha düşük düzey veri akış diyagamlarını gösterir. VAD'ları alt süreçlere ayırtıran sistem analisti, veri hareketi ile ilgili olarak ayrıntıları belirtmeye başlar. İlk iki veya üç düzey için istisnai durumlar göz ardı edilmektedir.

Şekil 6.7'de görülen düzey 0 VAD, bağlam diyagramının ayrıştırılması sonucu elde edilen diyagramdır ve dokuz adede kadar süreç bulunabilmektedir. Bu düzeyde daha fazla sürecin dâhil edilmesi anlaşılması zor ve dağınık bir diyagram oluşmasına sebep olacak-

tir. Her bir süreç sol üst köşeden başlayarak tam sayı ile numaralandırılır. Tam sayılarla noktadan (.) sonra sıfır (0) rakamının kullanılması düzey 0 VAD'da bulunan bir süreç olduğunu gösterir.

DİKKAT



**Düzey 0 VAD'a özel olarak, Diyagram 0 denmesinin sebebi diyagram üzerindeki tüm süreç numaralarının yanında nokta (.) işaretinden sonra 0 bulunmasıdır. Bir sürecin hangi düzey VAD'a ait olduğu noktalar ile ayrılmış basamak sayısından 1 çıkarılarak tespit edilir.**

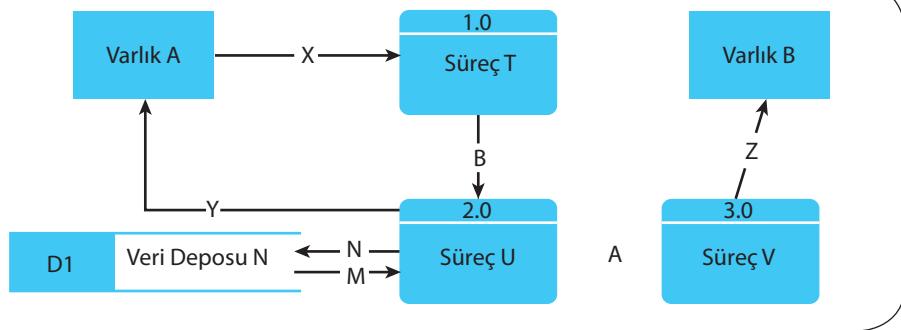
### Alt Düzey VAD'ların Çizilmesi

**Üst Süreç (Parent Process):** Bir sonraki alt düzey VAD içinde alt süreçlere ayrırtılabilen süreçtir.

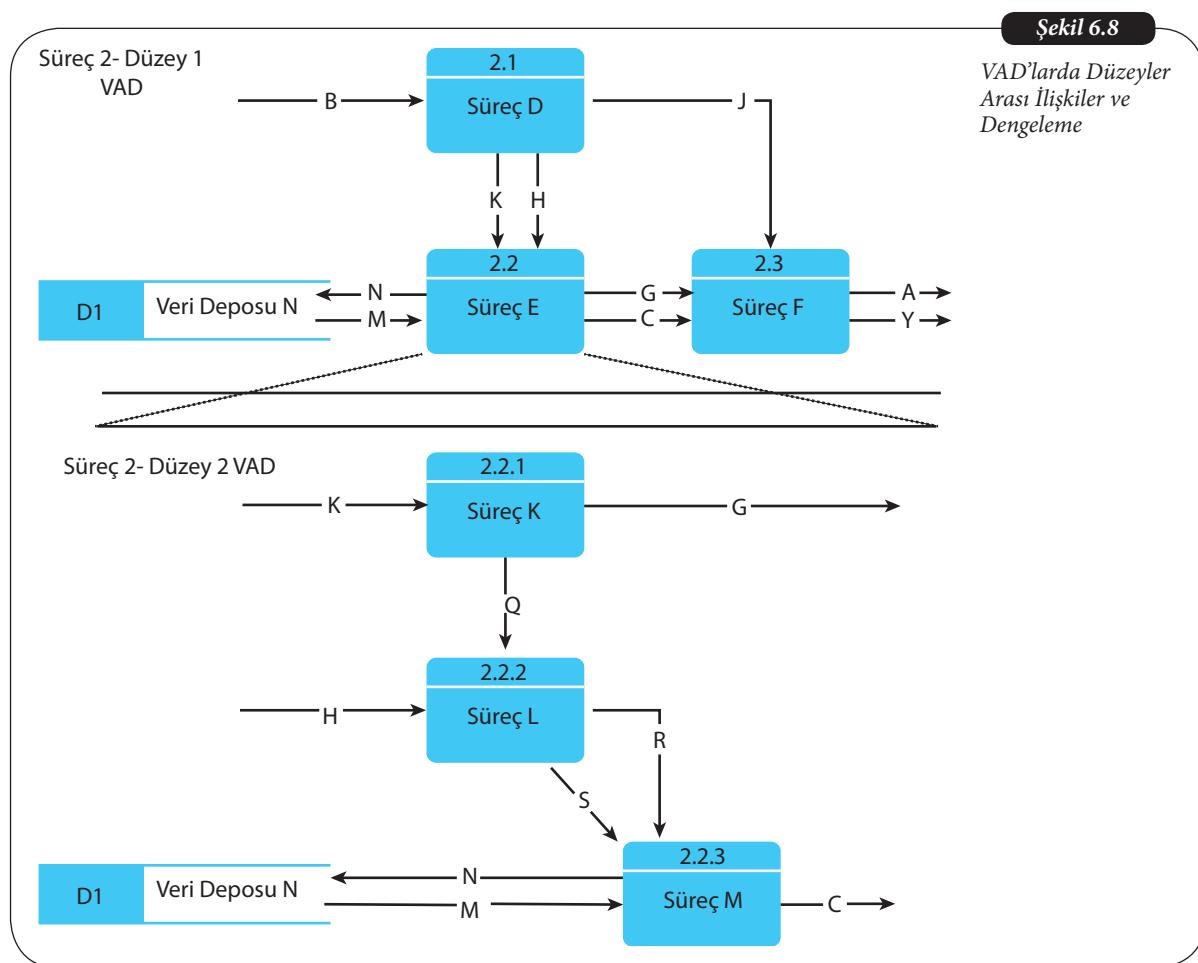
**Alt Düzey VAD (Child DFD):** Bir VAD içinde yer alan sürecin alt süreçlere ayrırtılmasıyla elde edilen VAD'dır.

**Şekil 6.7**

Bağlam Diyagramının Ayrırtılmasıyla Elde Edilen Düzey 0 VAD



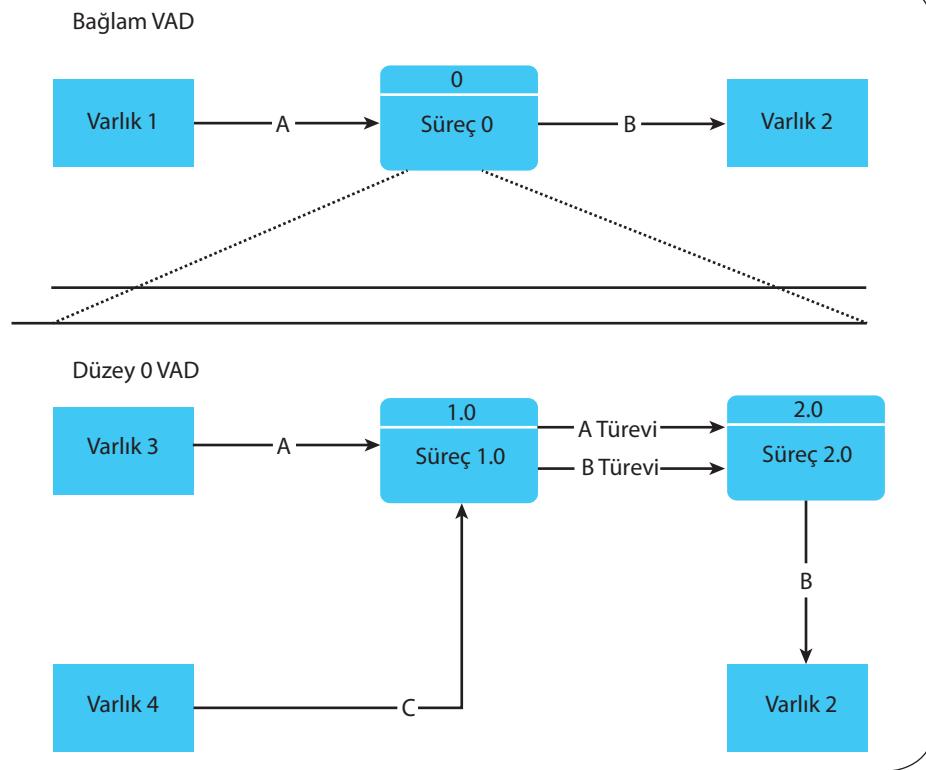
Üst süreçlerin alt düzey VAD'lara ayrıştırılması işlemi Şekil 6.8'de görülmektedir. Burada da 2.2 numaralı E süreci, düzey 2 VAD'a ayrıştırılırken 3 yeni süreç daha ortaya çıkmaktadır. Düzey 1 VAD'da E sürecine K, H ve M veri akışları girdi olurken; G, C ve N veri akışları çıktı olmaktadır. Düzey 1 ve düzey 2 VAD'lar dengeli VAD kümesi olarak tanımlanabilir. Çünkü ayrıştırma sonrasında düzey 2 VAD'da girdi ve çıktılar korunmaktadır. K, H ve M girdi veri akışları sırasıyla K, L ve M süreçlerine girdi veri akışı olarak bağlanırlar. G, C ve N çıktı veri akışları sırasıyla K ve M süreçlerine çıktı veri akışı olarak bağlanmaktadır.



Şekil 6.9'da dengelenmemiş bir VAD düzeyi görülmektedir. Sisteme ait bağlam diyagramında bir adet girdi haricî varlık ve bir adet çıktı haricî varlık bulunmaktadır. Düzey 0 VAD'da ise ek olarak 3 haricî varlık girdisi ve farklı haricî varlıklardan gelen A ve C veri akışları bulunmaktadır. Şekil 6.9'da görülen iki VAD dengeli değildir. VAD'ların tüm düzeyleri boyunca dengeli tutulması çok önemlidir.

**Şekil 6.9**

Bağlam ve Düzey 0 VAD Arasında  
Veri Akışlarının  
Korunmaması  
Sebebiyle Oluşan  
Dengesizlik



SIRA SİZDE



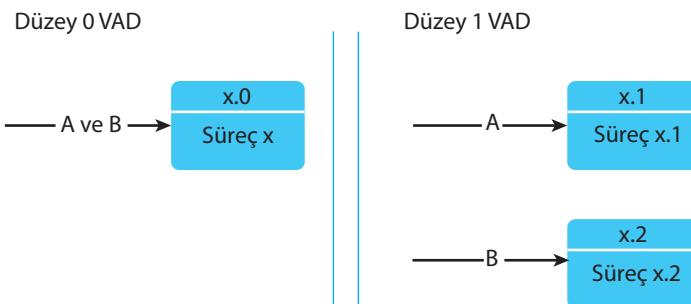
**Şekil 6.9'da görülen dengesiz VAD kümelerini dengeli hale getirmek için ne yapılmalıdır? Dengeli yeni VAD kümelerini çiziniz.**

**Dengeleme (Balancing):** VAD üzerindeki bir sürecin daha alt düzey VAD'a ayrıştırılmasında girdi ve çıktıların korunmasıdır.

Bir veri akışı, herhangi bir düzeye birden fazla veri akışının bileşimi şeklinde süreç tarafından girdi olarak kabul edilebilir. Bu durumda bir sonraki düzeye, ilgili süreç farklı alt süreçlere ayrıstırılır. Bileşik veri akışı bileşenlerinin bu alt süreçler tarafından işlenerek bileşik veri akışının bölünmesi sağlanmış olur. Bu durumda herhangi bir VAD dengesizliği söz konusu değildir. Çünkü her diyagramda hala aynı miktarda girdi bulunmaktadır. Şekil 6.10'da görülen düzey 0 VAD üzerinde A ve B veri akışı A ve B olmak üzere iki farklı veri akışının birleşimidir. Bu yüzden bir sonraki alt düzey olan düzey 1 VAD üzerinde X üst süreci X.1 ve X.2 olmak üzere iki ayrı alt süreç olarak ifade edilmektedir.

**Şekil 6.10**

VAD'larda Bileşik  
Veri Akışları ve  
Alt Düzey VAD'da  
Ayrıştırılmaları



[http://creately.com/diagram-community/all?term="Data Flow Diagram Template"](http://creately.com/diagram-community/all?term='Data Flow Diagram Template') bağlanışından çevrimiçi uygulama yardımıyla hazırlanmış olan VAD şablonları üzerinde çalışabilirsiniz.



INTERNET

## MANTIKSAL VE FİZİKSEL VERİ AKIŞ DİYAGRAMLARI

VAD'lar mantıksal ve fiziksel olmak üzere iki sınıfta incelenirler. *Mantıksal VAD*, yapılacak olan işe ve bu işin nasıl yapılacağına odaklanır. Mantıksal VAD'da sistemde yer alacak işletme faaliyetleri, her bir faaliyet tarafından üretilen ve kullanılacak veriler tanımlanır. *Fiziksel VAD* ise sistemin nasıl uygulanacağını, hangi donanım, yazılım, dosya ve insan kaynağının kullanılacağını gösterir. Tablo 6.1'de mantıksal ve fiziksel VAD özellikleri görülmektedir.

Tasarım Özelliği	Mantıksal VAD	Fiziksel VAD
Modelin gösterdiği nedir?	İşletme nasıl çalışır?	Sistem nasıl hayatı geçirilecek? (veya mevcut sistem nasıl çalışıyor?)
Süreçler neyi gösterir?	İşletme Faaliyetleri	Programlar, program modülleri ve elle yürütülen işlemler
Veri depoları nedir?	Veri koleksiyonu (Verilerin nasıl saklanacağı dikkate alınmaz)	Fiziksel dosyalar ve veritabanları
Veri deposu türleri	Kalıcı veri koleksiyonlarını temsil eden veri depoları	Veri tanımlarını içeren ana dosyalar ve geçici dosyalar
Sistem kontrolleri	İş kontrollerini gösterir	Girdileri doğrulamak, kayıt elde etmek, sürecin başarı ile tamamlandığından emin olmak ve sistem güvenliği için kontroller yapmak

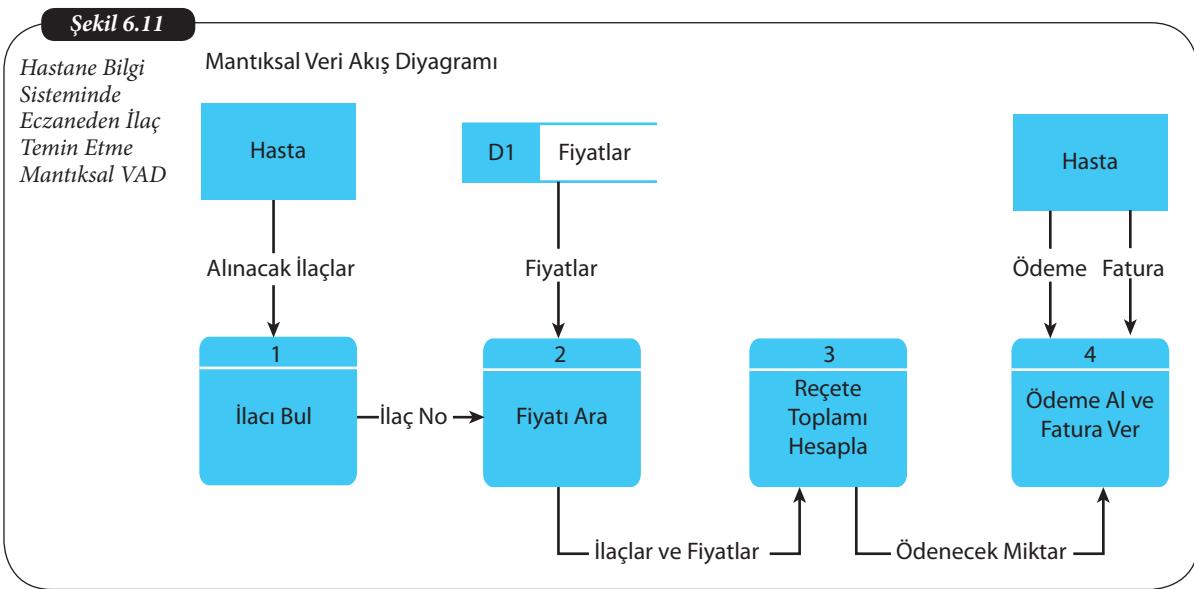
**Tablo 6.1**  
*Mantıksal ve Fiziksel VAD Özellikleri*

**Kaynak:** Kendall, K.E. ve Kendall, J.E. (2011), s. 202.

Sistem geliştirmede genel olarak dört farklı tipte VAD kullanılır. Mevcut sistem analiz edilip, *mevcut fiziksel VAD* elde edilir. Fiziksel VAD'a bağlı olarak *mevcut mantıksal VAD* oluşturulmaya başlanır. Daha sonra mevcut mantıksal VAD'a yeni özellikler eklenecek *önerilen mantıksal VAD* elde edilir. Son olarak yeni geliştirilen (ya da iyileştirilen) sistem için *önerilen fiziksel VAD* oluşturulur.

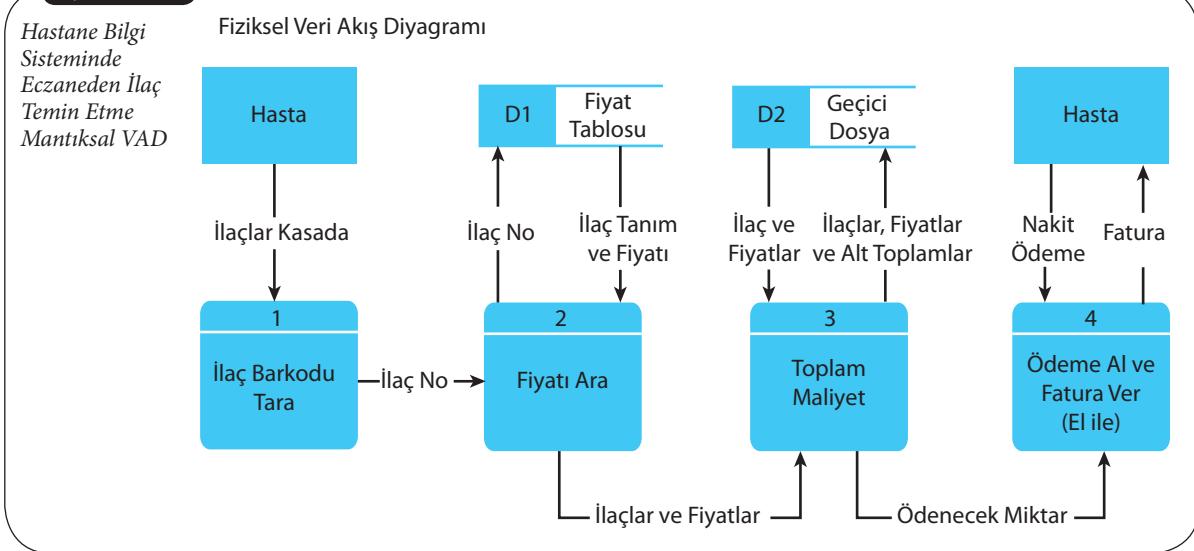
Mevcut fiziksel ve mantıksal VAD'ların oluşturulması, mevcut sistemin gerçekleştirdiği işleri anlamak için gereklidir. Fakat genellikle zaman aldığı için ihmäl edilerek doğrudan önerilen mantıksal VAD oluşturma adıma geçilmektedir. Bu düşünce yanlışdır. Çünkü mevcut mantıksal VAD oluşturmak için zaman ayırmadan önemli bir avantajı, mevcut mantıksal VAD'in yeni geliştirilmesi planlanan sistemde de kullanılabilmesidir. Oluşturulan mevcut mantıksal VAD üzerindeki gereksiz süreçler ayıklanıp çıkarılabilir. Yeni özellikler, etkinlikler, girdiler, çıktılar ve saklanması gerekliliği yeni veriler eklenebilir. Bu şekilde eski sistem üzerindeki olmazsa olmaz temel özelliklerin yeni sistemde de bulunması garanti altına alınmış olur. Önerilecek sistem için taban oluşturan mevcut sistemin mantıksal modelinin kullanılması yeni sistem tasarımasına kademeli bir geçiş yapılmasını sağlamaktadır. Yeni sistemin önerilen mantıksal VAD geliştirmesi tamamlandıktan sonra yeni sistem için kullanılacak olan fiziksel VAD oluşturma aşamasına geçilmektedir.

Şekil 6.11



Şekil 6.11'de bir hastane bilgi sistemindeki, alt eczane sistemine ait mantıksal süreç VAD'ı görülmektedir. Reçetedeki ilaçlar temin edildikten sonra kasada bulunan eczane görevlisi ilaçları tek tek barkod okuyucudan okutmaktadır. Veritabanında ilaç fiyatlarının bulunduğu tablodan ilgili ilaçların fiyatları ilaç numarası ile aranmaktadır. Sistemde ilaç bilgileri ve fiyatları bulunduktan sonra ara toplamlar geçici olarak hareket (transaction) dosyasında tutulmaktadır. Daha sonra reçete toplam maliyeti hastaya bildirilip ödemesi alındıktan sonra ilaçlar ve fatura hastaya teslim edilmektedir. Eczaneden ilaç temin etme sistemine ait fiziksnel VAD ise şekil 6.12'de görülmektedir.

Şekil 6.12



SIRA SİZDE



Geliştirilmek istenen “Randevu Sistemi” için bağlam diyagramını çiziniz. Randevu sistemi açıklaması: Hastane bilgi sistemi içinde bulunan randevu sisteminin kullanıcıları Hasta, Doktor ve İdari Memurdan oluşmaktadır. Hasta sistemden randevu talep eder ve gelen randevu bilgilerinden seçim yapar. İdari memur sisteminde doktorların listesini aldıktan sonra randevu çizelgesi oluşturur. Doktorlar da idari memur tarafından oluşturulmuş olan randevu çizelgesinin kendileri için uygun olup olmadığını onaylamaktadırlar.

## SÜREÇLERİN MANTIKSAL MODELLENMESİ

VAD'lar süreçlerin tanımlanması için kullanışlı araçlar olmalarına rağmen, süreç içinde gerçekleştirilen işlemin mantığını göstermezler. Süreçlerin içinde ne tür işlemlerin gerçekleştiği, veri girdilerinin çıkışta bir bilgiye nasıl dönüşeceği konularında VAD çizimlerinde herhangi bir açıklama bulunmaz. VAD'lar süreçlerin ayrıntılı işlem mantığını göstermek amacıyla tasarlannmadığı için süreç mantığının farklı başka teknikler ile modellenmesi gerekmektedir.

Mantıksal modelleme, VAD üzerinde görülen süreçlerin iç yapılarının ve işlevlerinin ifade edilmesidir. VAD sayesinde süreçlerin isimleri ve hangi işi gerçekleştirdikleri konusunda bilgi sahibi olunabilmektedir. Fakat işi nasıl yaptıkları konusunda tam bir bilgi sahibi olmak mümkün değildir. Herhangi bir bilgi sisteminde sistem süreçlerinin yapı ve işlevselliği önemli bir unsurdur. Bu yüzden de bilgi sisteminin programlama ve uygulama geliştirme aşamasına geçilmeden önce süreçlerin açık ve net bir biçimde tanımlanmaları gerekmektedir.

Süreç tanımlamaları VAD üzerinde bulunan en alt düzey VAD'daki süreçler üzerinden yapılır. Her bir süreç tanımlaması ayrı bir form üzerinde gerçekleştirilir. **Süreç tanımlama formlarına** aşağıdaki bilgiler girilir:

- i. Süreç Numarası: VAD üzerindeki süreç numarası ile aynı olmalıdır. Bu özellik, sistem analistinin herhangi bir süreç üzerinde çalışmasına, süreci gözden geçirmesine olanak sağlar.
- ii. Süreç adı: VAD üzerindeki süreç adı ile aynı olmalıdır.
- iii. Süreç tanımı: Sürecin hangi işlemi gerçekleştirildiği ile ilgili kısa bir açıklama yazılımalıdır.
- iv. Girdi veri akışları: VAD üzerinde süreç içinde işlenmek üzere gelen veri akış listesi belirtilir.
- v. Çıktı veri akışları: VAD üzerinde süreç içinde işlendikten sonra süreç dışına gönderilen veri akış listesi belirtilir.
- vi. Süreç tipi: Sürecin nasıl gerçekleştirileceği açıklanır. Süreç; elle, otomatik ya da çevrimiçi olabilir. Çevrimiçi ve bilgisayar destekli işleyecek süreçler için ekran tasarımı yapılır. Elle işleyecek süreçler için ise ayrıntılı tanımlanmış yönergelerin oluşturulması gerekmektedir.
- vii. Mevcut kod ve yönergeler: Süreç içerisinde daha önceden yazılmış program kodları ve hazırlanmış yönergeler kullanılabilir. Bu bölümde alt program, yordam, fonksiyon, web servisi, uygulama programlama arayüzü (API) ya da kütüphane isimleri vb. program bileşenleri ile yönerge isimleri yer almıştır.
- viii. Süreç mantıksal modeli: Süreçte gerçekleştirilecek işlev ait kuralların programlama dili yerine günlük konuşma dilinde açıklandığı bölümüdür. Bu kurallar; yordamlar, bazı koşul durumları ve formülleri içerir. Bu bölümde çoğu zaman *yapısal dil, karar tablolari* veya *karar ağaçları* modelleme yöntemleri kullanılabilir.
- ix. Çözülmeyen soru ve sorunlar: Sürecin mantıksal modellemesi sırasında karşılaşılan ve cevaplanmamış soru ve sorunlar buraya yazılır. Bu soru ve sorunlarla ilgili olarak ilgili kullanıcılar ile yapılacak görüşmeler sonucunda sistem tasarıma yeni süreçler eklenebilir.

**Şekil 6.12'de görülen fiziksel VAD üzerindeki 4 numaralı süreç olan "Ödemeyi Al ve Fatura Ver" sürecine ait süreç tanımlama formunu hazırlayınız.**

**Süreç Tanımlama Formu:**  
VAD üzerindeki süreçlerin tanımlanması amacıyla kullanılan ve sistem analistleri tarafından hazırlanan bir kâğıt form ya da bilgisayar destekli sistem geliştirme araçlarındaki bir form ekranıdır.



SIRA SİZDE

3

Süreç tanımlama formu, süreç içindeki işlevler hakkında ayrıntılı bilgi verir ve özel süreç adımları ile iş mantığının sunulmasına yardımcı olur. Süreç mantıksal modelinin açıklanmasında kullanılan yöntemler yardımıyla doğru, eksiksiz ve özlü bir mantıksal model yaratılmasına olanak sağlanmaktadır. Sıklıkla kullanılan süreç mantık modelleme yöntemleri arasında *yapısal dil, karar tabloları ve karar ağaçları* yer almaktadır.

## **Yapısal Dil**

Yapısal dil (structured english), süreçlerin mantıksal olarak ifade edilmesine olanak sağlayan ve standart ingilizce kelimelerden oluşturulmuş bir alt kümedir. Yapılandırılmış dil ve ingilizce “pseudo code” kelimesinin karşılığı olan “sözde kod” tanımları eş anlamlı olarak kullanılmaktadır. Süreç mantığını modelleme yöntemi olarak yapısal dil yöntemi kullanıldığında aşağıdaki kurallara uyulması faydalı olacaktır:

- i. Sıralı (ardışık) işlemler, karar işlemleri, durum kontrolleri ve döngü (tekrarlama) işlemleri için oluşturulmuş dört temel yapı bloğu kullanılmalıdır.
- ii. Kolay okunabilir olması amacıyla girintili yazılmalıdır.
- iii. Süreç kurallarını tanımlayıcı özellikte sınırlı sayıda kelime kullanılmalıdır.

Bazı kaynaklarda karar ve durum kontrol yapıları birleştirilerek koşul ya da şartlı ifade blokları olarak tek bir yapı altında toplanmışlardır. Tablo 6.2'de yapısal dil yönteminde kullanılan dört temel yapı bloğu ve bu yapı bloklarına ilişkin örnekler yer almaktadır.

**Tablo 6.2**  
**Yapısal Dil Blokları**

**Kaynak:** Kendall, K.E. ve Kendall, Y.E. (2011), s. 263.

<b>Yapısal Dil Bloğu</b>	<b>Örnek</b>
Ardışık Blok Yaptısı: Herhangi bir yönlendirmenin bulunmadığı, özel işlemler ya da kontrol gerektirmeyen işlem bloklarıdır.	Eylem #1 Eylem#2 Eylem#3 Eylem#4
Karar İşlem Bloğu: IF ifadesinden sonra verilen koşul doğru ise THEN ifadesinden sonraki eylemler uygulanır; aksi durumda ELSE ifadesinden sonraki eylemler uygulanır.	IF Koşul 1 Doğru THEN Eylem#1 uygula ELSE Eylem#2 uygula END IF
Durum Kontrol Bloğu: Karar işlem bloğunun özel bir tipidir. Bir koşulun izleyebileceği birden fazla durum varsa ve bu durumlardan biri oluştuğunda diğerleri oluşamıyorsa kullanılan blok yapısıdır.	READ Kontrol-edilecek-değer SELECT CASE CASE 1 (Kontrol-edilecek-değer=Koşul1) DO Eylem#1 CASE 2 (Kontrol-edilecek-değer=Koşul2) DO Eylem#2 CASE 3 (Kontrol-edilecek-değer=Koşul3) DO Eylem#3 CASE 4 (Kontrol-edilecek-değer=Koşul4) DO Eylem#4 END CASE
Döngü Blok Yaptısı: Koşul sağlanana kadar tekrarlanması gereken eylemler varsa kullanılır.	DO WHILE Koşul1 doğru oldukça Eylem#1 ENDDO Veya DO Eylem#1 UNTIL Koşul1 doğru olunca

Eczane stoklarındaki ilaç miktarına bağlı olarak yapılması gereken eylemlerin ne olması gerektiğine karar veren sürecin mantıksal modellemesi aşağıda durum kontrol bloğu ile gerçekleştirilmiştir. Bu kontrol bloğunda stoklardaki ilaç miktarı belirlenen miktarın

altına düşerse eczacı uyarılmakta ve sipariş verilmektedir. Stokta aranılan ilaç hiç kalmadı ise muadil ilaç seçimi yapılmakta, ilgili ilacın üretimi durduruldu ise ilacın reçeteye yazılıması engellenmektedir.

```

READ İLAÇ_STOK_MİKTARI
SELECT CASE İLAÇ_STOK_MİKTARI
    CASE 1 (İLAÇ_STOK_MİKTARI büyükse KRİTİK_STOK)
        Herhangi bir eylem uygulama
    CASE 2 (İLAÇ_STOK_MİKTARI eşitse KRİTİK_STOK)
        Eczane çalışanına bildirimde bulun
    CASE 3 (İLAÇ_STOK_MİKTARI küçükse KRİTİK_STOK)
        Otomatik olarak sipariş üret
    CASE 4 (İLAÇ_STOK_MİKTARI eşitse SIFIR)
        Muadil ilaçları göster
IF İLAC_URETIMI_DURDU
    THEN İlacın reçeteye yazılmasını engelle
END IF
END CASE

```

## Karar Tabloları

Karar tablosu, süreç içinde karşılaşılabilecek koşul ve bu koşullara karşılık uygulanması gereken kuralların her bir birleşimini gösteren mantıksal yapıdır. Süreç içindeki iş mantığında birçok şartlı durum ile karşılaşılıyor ve karşılaşılan her bir farklı durum için çeşitli işlevler gerçekleştirmek gerekiyorsa süreç iş mantığının yapısal dil ile ifade edilmesi oldukça zordur. Kullanıcıların, süreç iş mantığının modellenmesinde kullanılan çok sayıdaki iç içe IF yapısal bloğu gözden geçirmesi ve yorumlaması zor olmaktadır. Süreç iş mantığının karmaşık olduğu durumlarda, karar tabloları kullanılması, süreç mantıksal modelinin anlaşılmasını kolaylaştırmaktadır. Karar tablosu, süreç iş mantığını tablo biçiminde gösteren bir araçtır. Tablo 6.3'te görüldüğü gibi dört ana bölümden meydana gelmektedir.

Karar Tablosu Adı	Kurallar
Koşullar	Koşul Alternatif Değerleri
Eylemler	Eylem Kayıtları

**Tablo 6.3**  
Karar Tablosu  
Bölümleri

- Koşullar:** Tablonun sol üst bölümüne süreç içerisinde mümkün olan karşılaşabilecek tüm koşullar yazılır. Koşullar içinde birbirinin tersi durumlar söz konusu olduğunda bu koşullar birleştirilir. Böylelikle koşul sayısı da sadeleştirilmiş olur. Örneğin, bir koşulda kullanıcının erkek olması, bir başka koşulda ise kadın olması bulunuyor ve değerleri "E" ve "H" olarak işaretleniyor olsun. Kullanıcı aynı anda erkek ve kadın olamayacağından dolayı bu iki koşul kullanıcının cinsiyeti koşulu adı altında birleştirilebilir. Alabileceği değerler ise kadın için "K" ve erkek için "E" olarak belirlenebilir.
- Koşul Alternatif Değerleri:** Tablonun sağ üst bölümünde bulunur. Koşul alternatif değerleri koşul sayısına bağlı olarak değişir. Koşulun gerçekleşmesi ya da gerçekleşmemesi durumunda hangi koşul alternatif değerinin uygulanacağını gösterir. Koşul alternatif değerleri Evet (veya 1) ya da Hayır (veya 0) lardan oluşur. Belirli bir koşula ilişkin koşul alternatif değerlerinin sayısı diğer koşulların sayısına bağlıdır ve bu sayı aynı zamanda kural sütun sayısını belirler.

- iii. Eylemler: Karar tablosunun sol alt bölümünde yer alır. Koşullara bağlı olarak gerçekleştirilmesi olası tüm eylemler bu bölümde listelenir.
- iv. Eylem Kayıtları: Karar tablosunun sağ alt bölümünde bulunur. İlgili kural için hangi eylemin uygulanacağını belirtir. Gerçekleştirilecek olan eylem satırı ile kural sütunun kesiştiği hücreye "X" işaretini konur.

Hangi eylemin yapılacağını bulmak amacıyla karar tabloları kullanıldığı zaman süreç iş mantığı saat yönünde hareket eder. Önce karşılaşılan koşul altında hangi kuralın uygulanacağı belirlenir. Sonra da kural veya kurallara karşılık gelen eylemler gerçekleştirilir. İlaç satış ücretinin hasta tipine ve fatura bedeline göre belirlenmesi için oluşturulan örnek bir karar tablosu Tablo 6.4'te görülmektedir. Uygulanabilecek dört kural 1'den 4'e kadar numaralandırılmıştır. Koşullar bölümünde hasta tipine ve fatura bedeline göre indirim seçenekleri, Eylemler bölümünde ise seçilen kurala göre uygulanacak indirim tutarı belirlenmektedir. Tablo 6.4'te görüldüğü gibi ilaç toplam tutarı 50 TL'den küçük ya da eşit olan çalışan bir hasta için "indirim oranı 1" uygulanmaktadır. Benzer biçimde, emekli bir hastanın toplam tutarı 50 TL'den büyükse "indirim oranı 4" yanında "ek %10 indirim" de uygulanmaktadır.

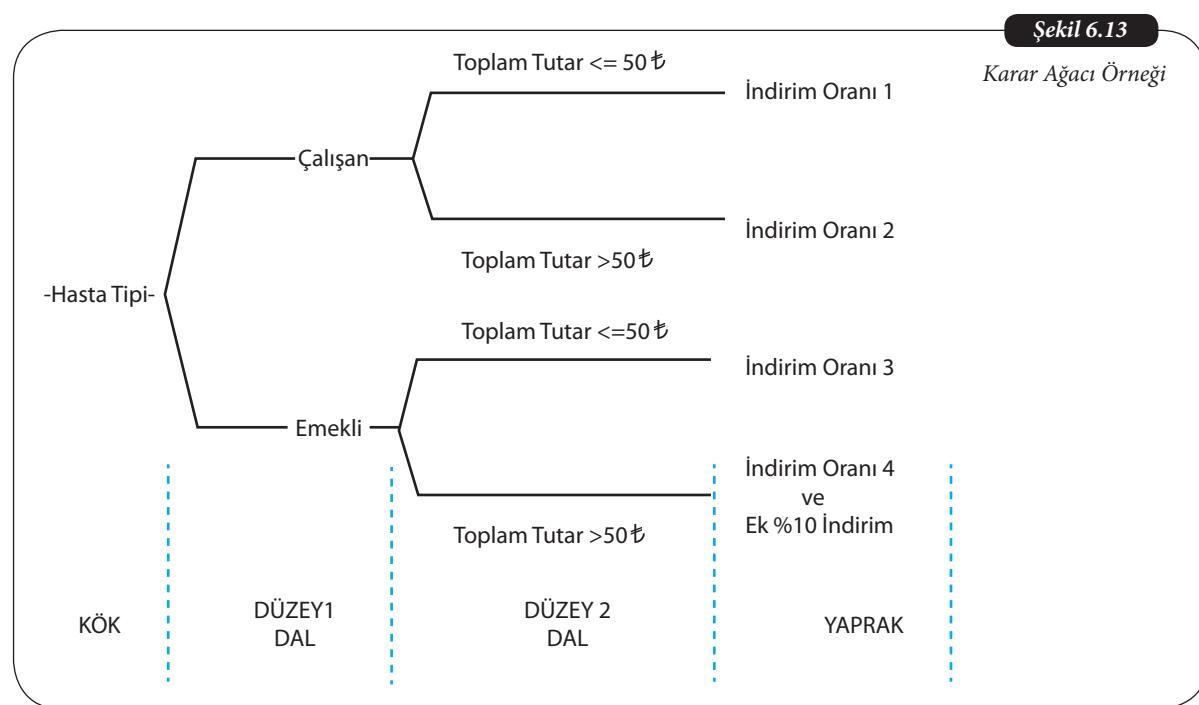
**Tablo 6.4**  
Karar Tablosu Örneği

<b>Ilaç İndirim Hesapla</b>	<b>KURALLAR</b>			
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>KOŞULLAR</b>				
Çalışan Hasta	E	E	H	H
Emekli Hasta	H	H	E	E
Toplam Tutar 50 TL'den küçük ya da eşit	E	H	E	H
Toplam Tutar 50 TL'den büyük	H	E	H	E
<b>EYLEMLER</b>				
İndirim Oranı 1 Uygula	X			
İndirim Oranı 2 Uygula		X		
İndirim Oranı 3 Uygula			X	
İndirim Oranı 4 Uygula				X
Ek %10 İndirim Uygula				X

Bir karar tablosunda kurallara karşılık gelen satırlardaki koşullar iki değerden birini alabilir. Örneğin Evet/Hayır, Var/Yok, Olumlu/Olumsuz vb. Bu durumda koşul sayısına n denilirse, koşul satır sayısına bağlı olarak elde edilebilecek koşul birleşimlerinden oluşmuş kural sayısı en fazla  $2^n$  olacaktır. Tablo 6.4'te verilen karar tablosu için koşul sayısı n=4 olduğundan, Tablo 6.4 için yazılabilen kural sayısı en fazla  $2^4=16$  olacaktır. Bununla birlikte, kural sütun sayısının dört olmasının sebebi bazı koşulların kural içerisinde hibz zaman birlikte bulunamamasından kaynaklanmaktadır. Örneğin, hastanın hem çalışan hem de aynı zamanda emekli olamayacağı; indirim uygulanacak toplam tutarın 50 TL'den küçük ya da eşit olurken aynı zamanda 50 TL'den büyük olamayacağı vb. gibi koşulların birleşimleri ile karşılaşılması mümkün değildir.

## Karar Ağaçları

Yapilandırılmış karar verme süreçlerinde karmaşık dallanmalar oluştukunda sürecin mantıksal modellenmesi için kullanılan yöntemdir. Karar ağacı, karar tablosunda bulunan koşul, eylem ve kuralların grafiksel olarak yatay bir şekilde gösterilmesidir.



Karar ağaçları ve karar tabloları gösterim biçimleri farklı olan fakat aynı sonucu üreten modelleme yöntemleridir. Tablo 6.4'te verilmiş olan karar tablosu ile aynı çıktıyı üreten karar ağacı yapısı Şekil 6.13'te görülmektedir. Bir karar ağacında en soldaki başlangıç noktasına *kök* (root) adı verilmektedir. Koşullara bağlı olarak dallanan karar ağacının en sağında ise uygulanacak eylemin ne olduğunu gösteren *yaprak* (leaf) yapıları bulunmaktadır.

Hangi yöntemin kullanılacağı sistem analistinin tercihine bağlıdır. Karar tablosu kullanılması çok karmaşık koşul birleşimlerinin ele alınmasında etkili bir yöntemdir. Bu nınla beraber görece olarak daha basit koşul birleşimine sahip olan süreçlerin mantıksal modellenmesinde etkili bir yol olarak karar ağaçları kullanılmaktadır.

**Tablo 6.4'te verilmiş olan karar tablosunda en fazla kaç tane kural bulunabilir? Karar tablosunun koşul satırları, verilecek kararları bozmayacak şekilde nasıl sadeleştirilebilir?**



SIRA SİZDE

4

## Özet



### Süreç modelleme kavramını tanımlamak

Süreç modelleme, verinin elde edilmesi, işlenmesi, saklanması ve sistem ile bulunduğu ortam ve sistem bileşenleri arasında hareketlerinin grafiksel olarak temsil edilmesidir. Günümüzde süreç modelleme amacıyla kullanılan çok farklı modelleme teknikleri bulunmaktadır. Mantıksal süreçlerin görsel olarak modellenmesi amacıyla en çok kullanılan süreç modelleme tekniklerinden biri veri akış diyagramlarıdır. Veri Akış Diyagramları, sadece 4 adet simbol kullanarak sisteme veri akışını grafiksel olarak ifade eden çok kullanışlı bir araçtır. Sistem ne kadar karmaşık olursa olsun bu diyagramlar sistemi tarif etmek için gerekli olan bilgileri barındırırlar. Veri akış diyagramı oluşturmak için kullanılan iki standart simbol kümesi bulunmaktadır. Her iki kümede de dört adet simbol bulunmaktadır ve semboller birbirinden çok az farklılık gösteriyor olsa da aynı anlamları ifade etmektedir.



### Veri akış diyagramları oluştururken dikkat edilmesi gereken kuralları açıklamak

Veri akış diyagramları, harici varlıklar, veri akışı, veri deposu ve süreç olmak üzere dört bileşenden oluşur. Diyagramlar oluşturulurken uyulması gereken bazı kurallar bulunmaktadır. Veri doğrudan bir harici varlıktan diğerine hareket edmez. Eğer sistem için gerekli ise mutlaka bir süreç üzerinden taşınmalıdır. Aksi durumda VAD üzerinde veri akışı gösterilmez. Veri akışı, semboller arasında sadece tek yönde gerçekleşir. Veri akışındaki ayrılma (çatallanma) aynı yerden ayrılan ortak verinin iki veya daha fazla sürece, veri deposuna ya da harici varlığa hareket ettiğini gösterir. Veri akışındaki birleşme, herhangi iki veya daha fazla farklı süreç, veri deposu veya harici varlıktan tam olarak aynı verinin geldiği anlamına gelir. Veri akışı çıkış yaptığı sürece doğrudan geri dönmez. Bir sürecin sadece girdi veya sadece çıktı olamaz. Veri, doğrudan bir veri deposundan diğer veri deposuna hareket edemeyeceği gibi harici varlıklar ile veri deposu arasında da doğrudan veri hareketi gerçekleşmez. Veri akış diyagramı oluşturulmasında yukarıdan aşağıya yaklaşımı kullanılmaktadır. Coğu iş süreci tek bir diyagram ile açıklanamayacak kadar karmaşık olduğundan süreç modelleri diyagram kümelerinden oluşmaktadır. Bağlam diyagramı olarak

da isimlendirilen ilk diyagram sistem genelinde özet bir bilgi sağlarken, ek veri akış diyagramları kullanılarak tüm iş süreçlerinin her biri için daha fazla ayrıntı sunulabilmektedir.

### Mantıksal ve fiziksnel veri akış diyagramları arasındaki farkları ayırt etmek

Mantıksal veri akış diyagramları, yapılacak olan iş ve bu işin nasıl yapılacağına odaklanırlar. Fiziksnel veri akış diyagramı sistemin nasıl uygulanacağını, hangi donanım, yazılım, dosya ve insan kaynağının kullanılacağını gösterir. Mantıksal veri akış diyagramlarında aşağıdaki bilgiler bulunmaktadır;

- i. İşletmenin nasıl çalışacağı,
- ii. İşletme faaliyetleri,
- iii. Veri koleksiyonları,
- iv. Kalıcı veri koleksiyonlarını temsil eden veri depları
- v. İş kontrolleri.

Fiziksnel veri akış diyagramlarında ise aşağıdaki bilgiler bulunmaktadır;

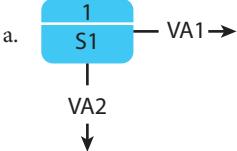
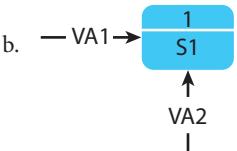
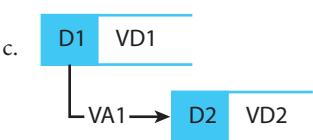
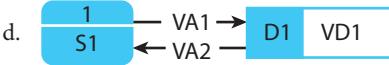
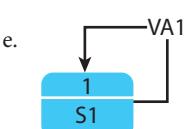
- i. Sistemin nasıl hayatı geçirileceği veya mevcut sistemin nasıl çalıştığı,
- ii. Fiziksnel olarak dosya ve veritabanları, veri tabanlarını içeren ana dosyalar ve geçici dosyaların bilgileri,
- iii. Sürecin başarılı veya hatalı tamamlandığını kontrol eden mekanizmalar.

### Süreçlerin mantıksal olarak modellenmesinde kullanılan yöntemleri açıklamak

Süreçlerin içinde ne tür işlemlerin gerçekleştiği, veri girdilerinin çıkışta bir bilgiye nasıl dönüşeceği konularında veri akış diyagramlarında bir açıklama bulunmaz. Veri akış diyagramları süreçlerin ayrıntılı işlem mantığını göstermek amacıyla tasaranmadığı için süreç mantığının modellenmesi gerekmektedir. Böylelikle, veri akış diyagramları üzerinde görülen süreçlerin iç yapıları ve işlevleri programlama ve uygulama geliştirme aşamasına geçilmeden önce açık ve net bir şekilde tanımlanmış olmaktadır. Yapısal dil, karar tabloları ve karar ağaçları süreçlerin mantıksal modellemesinde en çok başvurulan yöntemlerden üç tanesidir. Yapısal dil, süreçlerin mantıksal olarak ifade edilmesine olanak sağlayan ve standart ingiliz-

ce kelimelerden oluşturulmuş bir alt kümedir. Süreç mantığını modelleme yöntemi olarak yapısal dil yönteminde yararlanıldığında; sıralı (ardışık) işlemler, karar işlemleri, durum kontrolleri ve döngü (tekrarlama) işlemleri için oluşturulmuş dört temel yapı bloğu kullanılmalı, kolay okunabilir olması amacıyla girintili yazılmalıdır. Yapısal dil yönteminde ayrıca süreç kurallarını tanımlayıcı özellikle sınırlı sayıda kelime kullanılmalıdır. Süreç içindeki iş mantığında birçok şartlı durum ile karşılaşılıyor ve karşılaşılan her bir farklı durum için çeşitli işlevler gerçekleştirmek gerekiyorsa, süreç iş mantığının yapısal dil ile ifade edilmesi oldukça zordur. Bu durumda *karar tabloları* kullanılır. *Karar ağacı*, karar tablosunda bulunan koşul, eylem ve kuralların grafiksel olarak yatay bir şekilde gösterilmesidir. Karar ağacı, yapılandırılmış karar verme süreçlerinde karmaşık dallanmalar oluştugunda sürecin mantıksal modellenmesi için kullanılan bir yöntemdir.

## Kendimizi Sınayalım

- 1.** Bir kişi, kurum, kurum birimi veya bir sistemi ifade etmek amacıyla kullanılan veri akış diyagramı bileşeni aşağıdakilerden hangisidir?
- Süreç
  - Hariç Varlıklar
  - Veri Akışı
  - Veri Deposu
  - Veritabanı
- 2.** Veri üzerinde değişiklik ya da dönüşüm gerçekleştirildiğini göstermek amacıyla kullanılan veri akış diyagramı bileşeni aşağıdakilerden hangisidir?
- Süreç
  - Hariç Varlıklar
  - Veri Akışı
  - Veri Deposu
  - Veritabanı
- 3.** Aşağıdaki veri akış diyagramı gösterimlerinden hangisi doğrudur?
- a. 
- b. 
- c. 
- d. 
- e. 
- 4.** Bir bağlam diyagramda aşağıdaki veri akış diyagramı bilenlerinden hangisi **bulunmaz**?
- Süreç
  - Hariç Giriş Varlıklar
  - Hariç Çıkış Varlıklar
  - Veri Deposu
  - Veri akışı
- 5.** Bir sürecin numarası 4.3.2.1.7 olarak belirtilmişse, bu süreç hangi düzey veri akış diyagramı üzerinde yer alır?
- Düzey 1 Veri Akış Diyagramı
  - Düzey 2 Veri Akış Diyagramı
  - Düzey 3 Veri Akış Diyagramı
  - Düzey 4 Veri Akış Diyagramı
  - Düzey 5 Veri Akış Diyagramı
- 6.** “Veri akış diyagramları üzerindeki bir sürecin daha alt düzey VAD'a ayrıtırılmasında girdi ve çıktıların korunması” olarak tanımlanan kavram aşağıdakilerden hangisidir?
- Ayrıştırma
  - Birleştirme
  - Dengeleme
  - Kavramsallaştırma
  - Modelleme
- 7.** Mantıksal veri akış diyagramlarına ilişkin olarak aşağıda verilen ifadelerden hangisi **yanlıştır**?
- İşletmenin nasıl çalıştığını gösterir.
  - İşletme faaliyetlerini gösterir.
  - İş kontrollerini gösterir.
  - Programları ve program modüllerini gösterir.
  - Veri koleksiyonlarını gösterir.
- 8.** Fiziksel veri akış diyagramlarına ilişkin olarak aşağıda verilen ifadelerden hangisi **yanlıştır**?
- İşletme faaliyetlerini gösterir.
  - Mevcut sistemin nasıl çalıştığını gösterir.
  - Geliştirilen sistemin nasıl hayatı geçirileceğini gösterir.
  - Programları ve program modüllerini gösterir.
  - Sistem güvenliği kontrollerini gösterir.

### Kendimizi Sınavalım Yanıt Anahtarı

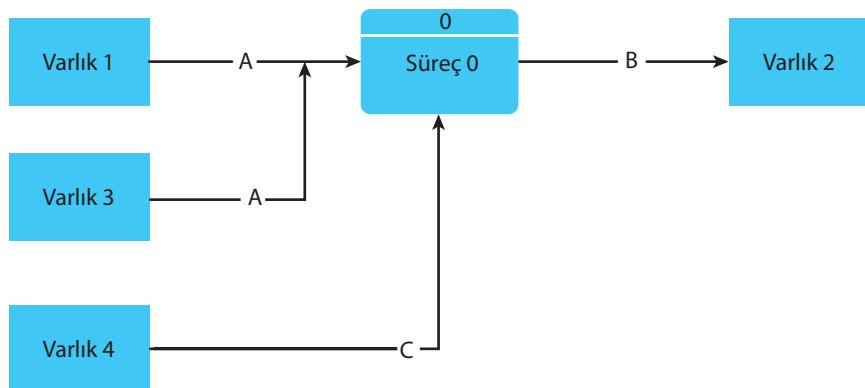
- 9.** Bir karar tablosunda kontrol edilmesi gereken koşul satır sayısı 4 olduğunda bu karar tablosunda koşul bireşimlerinden oluşanak kuralların sayısı **en fazla** kaç olmalıdır?
- 4
  - 8
  - 16
  - 32
  - 64
- 10.** Karar ağaçlarında yaprak düzeyde aşağıdaki karar tablosu bileşenlerinden hangisi bulunur?
- Koşullar
  - Eylemler
  - Kurallar
  - Eylem kayıtları
  - Koşul alternatif değerleri
- |       |  |
|-------|--|
| 1. b  | Yanınız yanlış ise “Veri Akış Diyagramları ve Bileşenleri” konusunu yeniden gözden geçiriniz.        |
| 2. a  | Yanınız yanlış ise “Veri Akış Diyagramları ve Bileşenleri” konusunu yeniden gözden geçiriniz.        |
| 3. d  | Yanınız yanlış ise “Veri Akış Diyagramları ve Bileşenleri” konusunu yeniden gözden geçiriniz.        |
| 4. d  | Yanınız yanlış ise “Veri Akış Diyagramlarının Oluşturulması” konusunu yeniden gözden geçiriniz.      |
| 5. d  | Yanınız yanlış ise “Veri Akış Diyagramlarının Oluşturulması” konusunu yeniden gözden geçiriniz.      |
| 6. c  | Yanınız yanlış ise “Veri Akış Diyagramlarının Oluşturulması” konusunu yeniden gözden geçiriniz.      |
| 7. d  | Yanınız yanlış ise “Mantıksal ve Fiziksel Veri Akış Diyagramları” konusunu yeniden gözden geçiriniz. |
| 8. a  | Yanınız yanlış ise “Mantıksal ve Fiziksel Veri Akış Diyagramları” konusunu yeniden gözden geçiriniz. |
| 9. c  | Yanınız yanlış ise “Süreçlerin Mantıksal Modellemesi” konusunu yeniden gözden geçiriniz.             |
| 10. b | Yanınız yanlış ise “Süreçlerin Mantıksal Modellemesi” konusunu yeniden gözden geçiriniz.             |

### Sıra Sizde Yanıt Anahtarı

#### Sıra Sizde 1

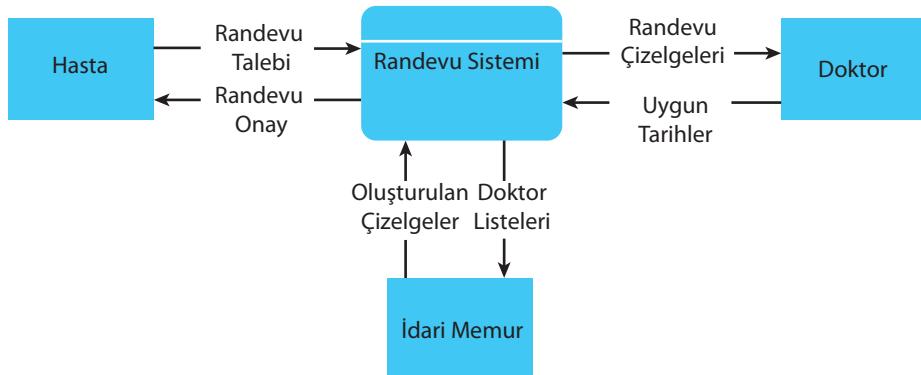
Düzey 0 VAD'da yer alan bir girdinin mutlaka bağlam diyagramında da bulunması gerekmektedir. Bu dengesizliğin düzeltilmesi için Varlık 3 ve Varlık 4 haricî varlıklarının bağlam diyagramına eklenmesi gerekmektedir. Dengeli yeni bağlam diyagramı aşağıda verilen biçimde çizilebilir.

Bağlam VAD



### Sıra Sizde 2

Hastane bilgi sistemi içinde bulunan randevu sisteminin Hasta, Doktor ve İdari Memur (yönetim) olmak üzere 3 haricî varlığı bulunmaktadır. Hastane sisteme randevu talep bilgisi gönderirken, sistem onay için randevu bilgilerini almaktadır. İdari memur sisteminde doktorların listesini aldıktan sonra randevu çizelgesi oluşturmaktır ve doktorlar da bu çizelgelerde kendilerine ait randevu tarihleri ile ilgili uygun olup olmadığını onaylamaktadırlar. Buna göre randevu sistemi için bağlam diyagramı aşağıdaki gibi olacaktır.



### Sıra Sizde 3

Süreç Tanımlama Formu	
No	4
Ad	Ödemeyi Al ve Fatura Ver
Tanım	Reçetedeki ilaçların toplam maliyetine bağlı olarak hasta ödemesi nakit ya da kart ile tahsil edilecek ve fatura verilecek.
Girdi Veri Akışı	<ul style="list-style-type: none"> <li>Süreç 3'ten hesaplanarak geleceklere "ne kadar ödeme yapılacağı" bilgisi,</li> <li>Hastadan yapılacak tahsilat.</li> </ul>
Çıktı Veri Akışı	Fatura
Süreç Tipi	<input type="checkbox"/> Çevrimiçi <input type="checkbox"/> Otomatik <input checked="" type="checkbox"/> El İle
Süreç Mantığı	<p>READ ÖDENECEK_MİKTAR ÖDEME AL SELECT CASE</p> <p>CASE 1 (ÖDEME büyükür ÖDENECEK_MİKTAR) DO Para üstü ver ÖDEME=Tamam</p> <p>CASE 2 (ÖDEME eşittir ÖDENECEK_MİKTAR) DO ÖDEME=Tamam</p> <p>CASE 3 (ÖDEME küçütür ÖDENECEK_MİKTAR) DO ÖDEME=Tamam değil</p> <p>END SELECT</p> <p>IF ÖDEME eşittir Tamam THEN Fatura ver</p> <p>ELSE Hastaya bilgi ver</p> <p>END IF</p>
Çözülmeyen Sorunlar	<ul style="list-style-type: none"> <li>Nakit ödeme dışında bir ödeme biçiminin kabul edilip edilemeyeceği belli değil.</li> <li>Reçete kayıp ya da unutulmuş ise nasıl bir işlem yapılacak?</li> </ul>

### Sıra Sizde 4

Bir karar tablosunda eğer kurallara karşılık gelen satırlardaki koşullar iki değerden birini alabiliyorsa (örneğin Evet/Hayır, Var/Yok, Olumlu/Olumsuz vb.) bu durumda koşul sayısına  $n$  denilirse, koşul satır sayısına bağlı olarak elde edilebilecek koşul bireşimlerinden oluşan kural sayısı en fazla  $2^n$  olacaktır. Tablo 6.4'te dört adet koşul satırı bulunduğundan dolayı bu karar tablosu için elde edilebilecek kural sayısını en fazla  $2^4=16$  olabilir. Bir hastanın hem çalışan hem de emekli olamayacağı için bu iki koşulun Evet ve Hayır şeklinde iki değerle ifade edilmesi yerine bu iki koşul birleştirilerek tek bir koşul olarak tanımlanabilir. Örneğin hasta türü isimli bir koşul oluşturulup çalışan için Ç ve emekli için de E harfi kullanılabilir. Bu durumda koşul satırı sayısı 3'e indirilmiş olur. Koşul satır sayısına bağlı olarak da uygulanabilecek kural sayısı en fazla  $2^3=8$  olur. Tablo 6.4'ü sadeleşmiş haliyle tekrar gösterelim.

KOŞULLAR	KURALLAR			
	1	2	3	4
Hasta Türü	Ç	Ç	E	E
Toplam Tutar 50 TL'den küçük ya da eşit	E	H	E	H
Toplam Tutar 50 TL'den büyük	H	E	H	E
<b>EYLEMLER</b>				
İndirim Oranı 1 Uygula	X			
İndirim Oranı 2 Uygula		X		
İndirim Oranı 3 Uygula			X	
İndirim Oranı 4 Uygula				X
Ek %10 İndirim Uygula				X

## Yararlanılan ve Başvurulabilecek Kaynaklar

- Arifoğlu, A. ve Doğru, A.,(2001). **Yazılım Mühendisliği**, SAS Bilişim Yayıncıları.
- Kendall, K.E. and Kendall, J.E., (2011). **Systems Analysis and Design 8th Edition**, Prentice Hall.
- Öz, E. ve Alp, S., (2010). Sistem Analizi ve Tasarımı – **Sistem Analizinde Kullanılan Sayısal Yöntemler**, Türkmen Yinevi.
- Roth, R.M., Dennis, A., and Wixom, B.H., (2013). **Systems Analysis and Design**, 5th Edition, John Wiley&Sons Inc.
- Shelly, G.B. and Rosenblatt, H.J., (2012). **Systems Analysis and Design, 9th Edition**, Course Technology CENGAGE Learning.
- Silahtaroğlu, G., (2014). **Sistem Analizi ve Tasarımı 2. Basım**, Papatya Yayıncıları.
- Valacich, J.S., George, J.F. and Hoffer, J.A., (2015). **Essentials of Systems Analysis and Design**, Pearson Education Limited.
- Whitten, J. and Bentley, L., (2005). **Systems Analysis and Design Methods, 7th Edition**, McGraw-Hill.

# 7

### Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Nesne modellemeye ilişkin temel kavramları ve nesne modellemenin faydalarını açıklayabilecek,
  - 🕒 UML diyagram türlerini tanımlayabilecek,
  - 🕒 Nesne modelleme sürecini açıklayabilecek bilgi ve becerilere sahip olabileceksiniz.

### Anahtar Kavramlar

- UML
- Nesne Tabanlı Modelleme
- Sınıf
- Miras
- Çok İşlevlilik
- Diyagramlar
- Kapsülleme

### İçindekiler

Sistem Analizi ve Tasarımı

UML ile Nesne Yönelimli  
Analiz ve Modelleme

- **GİRİŞ**
- NESNE MODELLEMELYE İLİŞKİN TEMEL KAVRAMLAR
- UML DİYAGRAMLARI
- NESNE MODELLEME SÜRECİ

# UML ile Nesne Yönelimli Analiz ve Modelleme

## GİRİŞ

Nesne tabanlı programlama dillerinin kullanımını giderek yaygınlaşmaktadır. Bunun nedeni, nesne tabanlı programlama dillerinin, kodların yeniden kullanımına olanak sağlaması ve programlama maliyetlerini azaltmasıdır. Ayrıca, nesne tabanlı programlama yaklaşımı sayesinde, farklı programlama takımlarının bir yazılımın ortaya çıkmasında iş birliği içinde çalışması mümkün olmaktadır. Bu sprogramlama grupları coğrafi olarak farklı konumlarda ikamet edebilmektedir. Her programlama grubu, belirlenmiş bir arayüze bağlı kalarak, bir ya da daha çok nesnenin gerçekleştirilmesi için gereken programlama kod parçalarını bağımsız olarak geliştirmek ile sorumlu olabilmektedir.

Programlamada nesne tabanlı yaklaşım, nesne tabanlı analiz (NTA) ve nesne tabanlı tasarım (NTT) için birtakım tekniklerin kullanılmasını gerektirir. Bu tekniklerde nesne tabanlı diyagramlar kullanılır. Sistem nesne tabanlı ortamda gerçekleşmeyecekse bu diyagramlar içinden sınıf (class) ve dizge (sequence) diyagramlar gibi bazlarının kullanımı, uygun değildir. Nesne tabanlı analiz ve tasarım için geliştirilen diğer diyagramlar, herhangi bir ortam içinde kullanılabilir. Örneğin, **kullanım durum** (use case) diyagramları, nesne tabanlı ya da geleneksel yapısal analizler için kullanılabilir. Benzer şekilde, nesne tabanlı analiz ve tasarım için geliştirilen, **etkinlik** (activity) ve **dağılım** (deployment) diyagramları diğer metodolojiler ile birlikte de kullanılabilir.

## NESNE MODELLEMELYE İLİŞKİN TEMEL KAVRAMLAR

Nesne tabanlı analizi gerçekleştirmek için, bazı kavramları bilmek gerekmektedir. Bu kavramlar, sistemler ve sistem geliştirme süreçleri üzerinde yeni bir düşünme yolu gerektirir. GelenekSEL yöntemlerde yaptığı gibi veri ve veri üzerinde işlem modelleri tanımlamak yerine; nesne tabanlı analiz statik veri yapıları ve dinamik davranış modelleri oluşturmaktadır. Nesne modellemeye ilişkin temel kavramlar izleyen kesimde verilmiştir.

## Nesneler, Özellikler, Davranışlar ve Kapsülleme

Sistem geliştirmede nesne tabanlı yaklaşım, sistem ortamının **nesnelerden** oluşması üzere kuruludur. Yaşadığımız dünya nesnelerden oluşmaktadır. Gerçekten de çevremizde, kapilar, masalar, dolaplar, ağaçlar gibi nesneler ve her nesnenin kendine özgü nitelikleri ve işlevleri bulunmaktadır. Ayrıca, nesneler birbirleri ile etkileşim içindedir.

Nesneler, sadece görülebilen ve dokunulabilen somut şeyler olmayabilir. Örneğin, arkadaşınızdan telefon çağrısı bekliyorsunuz. Gelen çağrı, görülemez ve dokunulamaz. Ancak, algılayabildiğimiz telefon çağrısı soyut bir nesnedir. Buna göre, içinde bulunduğu-

**Nesnenin**, sözlük tanımı  
"Görülebilir, dokunulabilen ya  
da daha farklı algılanabilen şey"  
biçimindedir.

muz çevre somut ya da soyut nesnelerden oluşmaktadır. Sistem geliştirmede nesne tabanlı yaklaşım kullanıldığında, sistemi oluşturacak nesnelerin belirlenmesi önemlidir. Nesneler, bir insan, bir yer, bir olay ya da bilgiyi bünyesinde tutmak istediğimiz bir şey olabilir. Bir çalışan, müşteri, eğitmen ve öğrenci *insan nesneleri*; bir depo, ofis, bina ve oda *yer nesneleri*; bir ürün, otomobil, bilgisayar ve bilgisayar ekranındaki pencere bir *şey nesnesi*; sipariş, ödeme, fatura ve rezervasyon, birer *olay nesnesi* olabilir.

**Öznitelikler (attributes)**, bir nesnenin sahip olduğu özellikleri belirten verilerdir.

**Davranışlar (behaviors)**, bir nesnenin yapabildiği şeylerleri belirtir ve nesnenin verileri üzerinde işlem yürüten fonksiyonlar olarak geçerlidir.

**Fonksiyon**, bilgisayar programındaki belirli bir işlevi yerine getirmek üzere oluşturulmuş alt programdır.

**Kapsülleme (encapsulation)**, verilerin ve bu veriler ile ilişkili işlemlerin bir varlık içerisinde tutulmasıdır.

Her nesne, öznitelik olarak adlandırılan ve sahip olduğu özellikleri belirten verilere sahiptir. Örneğin, “müşteri” olarak tanımlanan bir nesne, müşteri numarası, isim, soyisim, ev adresi, iş adresi, ev telefon numarası, iş telefon numarası gibi **özniteliklere** sahip olabilir. Şekil 7.1'de, benzer öznitelik tiplerine sahip iki nesne gösterilmektedir. Ali ve Mehmet iki hasta nesnesidir. Her iki nesne aynı tipte özniteliklere sahip olmalarına karşın özniteliklerin aldığı değerler farklıdır.

Her nesne, **davranışlara** sahiptir. Bu davranışlar, nesnenin neler yapabileceğini belirtir. Örneğin, bir nesne olarak kapı ile eşleştirilen davranışlar vardır. Kapı açılabilir, kapanabilir, kilitlenebilir ve kilidi açılabilir. Tüm davranışlar, kapı ile ilişkili davranışlardır. Başka bir nesne ile ilgisi yoktur. Nesne tabanlı programlamada, davranışlar **fonksiyonlar** olarak oluşturulur.

**Kapsülleme** kavramı, nesne tabanlı sistemler ile ilişkilidir. Kapsülleme, nesneleri, nesnelerin özniteliklerini belirten verileri ve kendi verileri üzerinde işlem gerçekleştiren fonksiyonları bünyesinde barındırır.

**Şekil 7.1**

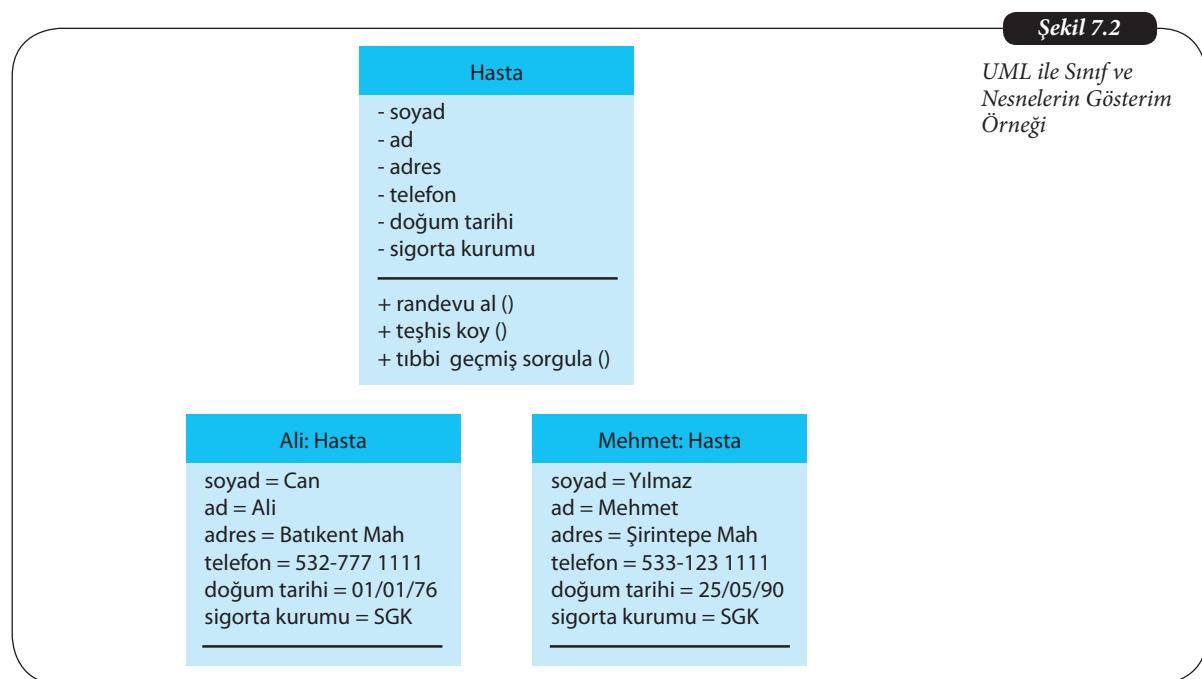
Aynı Öznitelik Türlerine Sahip Nesne Örnekleri

Ali: Hasta	Mehmet: Hasta
soyad = Can	soyad = Yılmaz
ad = Ali	ad = Mehmet
adres = Batıkent Mah	adres = Şirintepe Mah
telefon = 532-777 1111	telefon = 533-123 1111
doğum tarihi = 01/01/76	doğum tarihi = 25/05/90
sigorta kurumu = SG	sigorta kurumu = SGK

## Sınıflar, Genelleşme ve Özelleşme

**Sınıf (class)**, belirli nesneleri tanımlamak ve örneklemek üzere kullanılan genel bir şablondur.

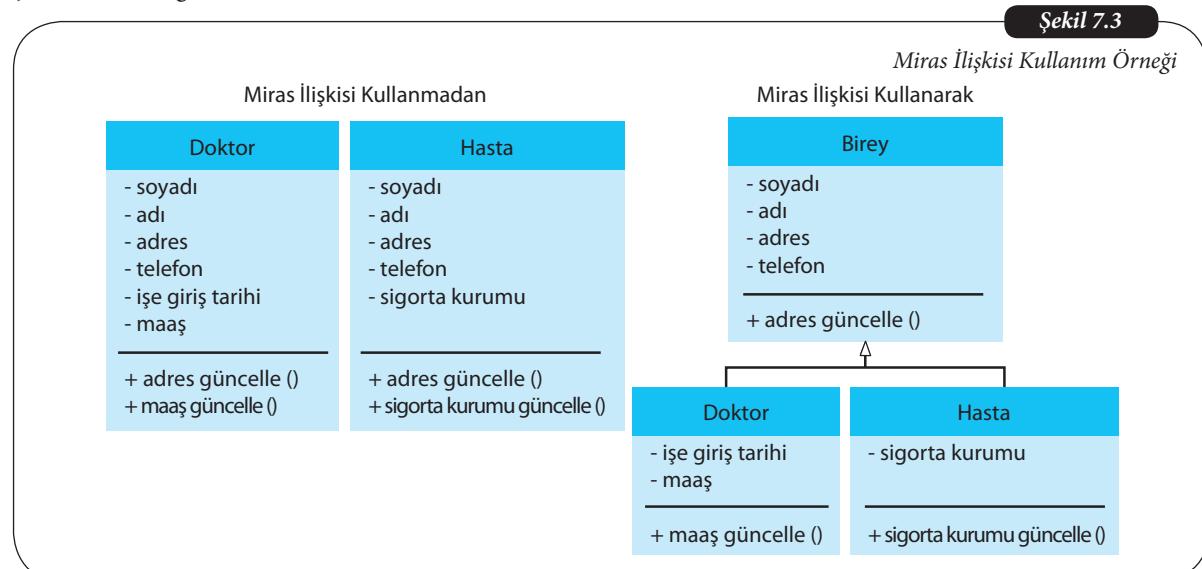
Nesne modellemede bir başka önemli kavram, nesnelerin gruplandırılması için kullanılan sınıflardır. Benzer öznitelik tiplerine ve davranışlara sahip nesneler, **sınıf** olarak tanımlanır. Belirli bir sınıfın ait nesne örneklenliğinde, o sınıf için önceden tanımlanan özniteliklere ve davranışlara sahip bir nesne var olur. Her nesnenin örneklentiği bir sınıf vardır. Her sınıfın, farklı özniteliklere sahip birden çok nesne örneklenebilir. Şekil 7.2'de sınıf ve nesneler gösterilmektedir. “Hasta” sınıfı, bir hasta nesnesinin sahip olması gereken öznitelikleri ve fonksiyonları tanımlamaktadır. “Hasta” sınıfından Ali ve Mehmet olarak örneklenen nesneler, aynı tip özniteliklere ve fonksiyonlara sahiptir. Ancak, her nesnenin öznitelik değerleri kendine özgüdür.



Sınıflar, üst sınıf ve alt sınıf olarak düzenlenebilmektedir. Bu düzenleme içinde, üst ve alt sınıflar arası ilişki **miras (inheritance)** ilişkisi olarak tanımlanmaktadır.

Miras kullanılarak tasarlanan sınıfların ortak veri ve fonksiyonları üst sınıf altında toplanmaktadır. Bu teknik, **genelleşme** olarak tanımlanabilir. Üst sınıftaki veri ve fonksiyonları kullanmak üzere oluşturulan sınıflar alt sınıf olarak adlandırılır. Alt sınıflar, üst sınıftan kullandıkları veri ve fonksiyonlara ilave olarak yeni veri ve fonksiyonlar da içerir. Buna da **özelleştirme** denir. Örneğin, "insan" sınıfı, "öğrenci" ve "öğretmen" sınıfları için bir üst sınıfı oluşturur. "öğrenci" ve "öğretmen" sınıfları, "insan" sınıfının sahip olduğu veri ve fonksiyonları kapsar. Ayrıca, bu sınıflar ilave veriler ve fonksiyonlar içerek özelleşirler. Şekil 7.3'te gösterildiği gibi, miras kullanımı ile ortak özellikler üst sınıfta toplanmaktadır. Böylece, "Doktor" ve "Hasta" sınıflarında ortak öznitelikler ve fonksiyonlar sadece bir kere programlanmaktadır. Böylelikle aynı şeylerin tekrar kodlanması nedeniyle ortaya çıkacak hata engellenmekte zaman ve emekten kazanılmaktadır.

**Miras (inheritance)**, bir nesnenin sahip olduğu veri ve fonksiyonların başka bir nesne tarafından miras yolu ile kullanılmasını belirten kavramıdır.



## Sınıf İlişkileri

Nesneler, diğer nesnelerden izole edilmiş (yalıtılmış) varlıklar değildir. Bir iş ya da işlem yerine getirmek üzere, birbirleri ile etkileşim içindedir. Buna göre bir nesne işin tamamlanması için bir diğer nesneden destek alabilir. Örneğin, bir yere giderken arabayı kullanarak onunla etkileşim içinde bulunulur. Ya da uzaktaki birisi ile konuşmak için telefon kullanılır. Benzer şekilde, nesneler de diğer nesneler ile etkileşim içindedirler. Örneğin, "Hasta" sınıfı ile "Randevu" sınıfı, bir bilgi sistemi ortamında var olabilir ve bir randevu gerçekleştirmeye amacı ile birlikte işlem yürütebilir. Buna göre;

- Bir *hasta*, sıfır ya da daha fazla *randevu* alabilir.
- Bir *randevu*, sadece ve sadece bir *hasta* tarafından alınır.

**Çarpan (multiplicity)**, bir sınıfın ait nesnenin diğer sınıfa ait kaç nesne ile ilişkilendirildiğini belirtir.

Sınıf ilişkilerini grafiksel olarak göstermeye ilişkin örnekler Şekil 7.4'te verilmiştir. Bu gösterimde, sınıflar arasında bulunan çizgi bu iki sınıfın birbirleri ile ilişkili olduğunu belirtmektedir. Çizgi üzerindeki, sınıflara yakın durumda verilen rakamlar **çarpan** olarak adlandırılmaktadır. Çarpan değeri, bir sınıfın diğer sınıfa ait kaç tane nesne ile ilişkilendirileceğini belirtmektedir.

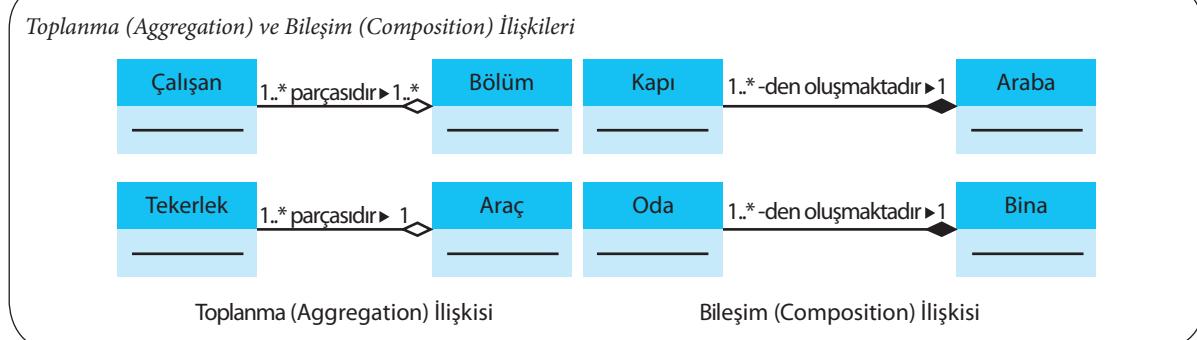
**Şekil 7.4**



Nesneler arasındaki ilişkinin türü farklılık gösterebilir. Söz konusu ilişki türleri, *bir parçasıdır* (*is a part of*) ya da *-dan oluşmaktadır* (*is entirely made of*) cümlecikleri ile ifade edilir.

**Toplanma (aggregation)**, bir nesnenin bir ya da daha çok nesneyi içermesini belirten ilişki türüdür. Örneğin, bilgisayar ve klavye ilişkisi bir örnek olarak verilebilir. Klavye, bilgisayarın *bir parçasıdır*. Bilgisayar, klavyeyi bir araç olarak kullanır. Ancak, bilgisayar kullanılamaz olduğunda, klavye bir başka bilgisayar ile kullanılmaya devam edebilir. Bu ilişkide, bütün içindeki bir parça, bütün olmadan da var olmaya devam eder.

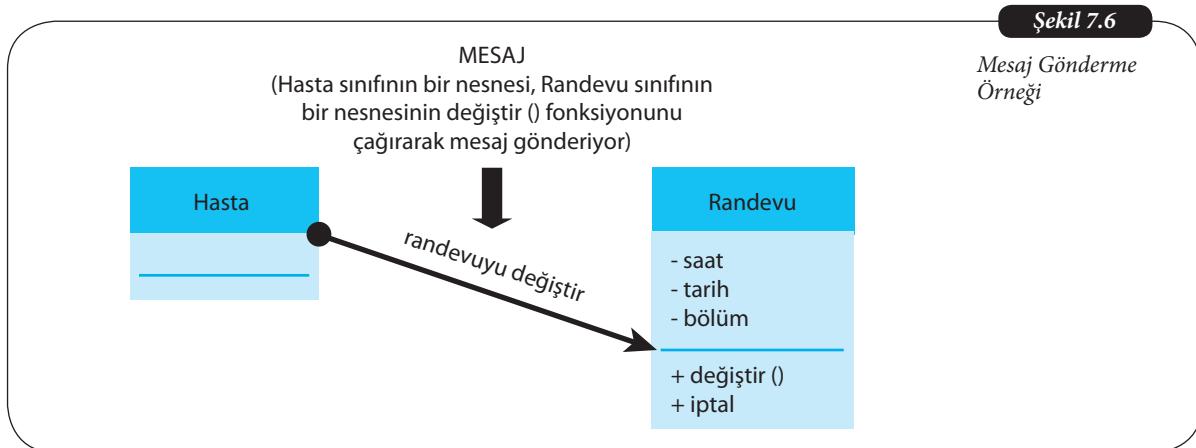
**Şekil 7.5**



*Toplanmaya* göre daha sıkı ilişki, **bileşim (composition)** ile tanımlanmaktadır. Örneğin, masa ve ayak ilişkisi bu kategoride değerlendirilebilir. *Masa, ayaklardan oluşmaktadır.* Ayaklar, masa var olduğu sürece var olacaktır. Bu ilişkide bütün, parçalardan oluşmuştur. Bütün, parçaların örneklenmesinden ve yok edilmesinden tamamen sorumludur. Ayrıca, parça sadece bir bütüne aittir. Şekil 7.5'te toplanma ve bileşim ilişki tipleri örneklenmiştir.

## Mesajlar ve Mesaj Gönderme

Nesnelerin etkileşimi ya da haberleşmesi ifadeleri, bir nesnenin diğer nesneye mesaj göndermesi anlamında kullanılır. Mesaj göndermek ise bir nesnenin sahip olduğu fonksiyona çağrı yapılmasını belirtmektedir. Mesaj gerçekte bir nesneden bir başka nesneye bir fonksiyon çağrısidir. Örneğin, hasta ve randevu birer sınıf olsun. Hasta randevunun mevcut durumunu kontrol etmek üzere, randevu nesnesinin *durumGoster* fonksiyonuna çağrı yaparak bir mesaj gönderir. Şekil 7.6'da "Hasta" sınıfının bir nesnesi, "Randevu" sınıfı nesnesinin *degistir()* fonksiyonuna çağrı yaparak, randevu değişikliği talebini mesaj ile iletmış oluyor.



## Çok Biçimlilik

Nesnelerin mesajlaşması ile ilgili önemli bir kavram, **çok biçimlilik (polymorphism)** olarak adlandırılmaktadır. Bu kavram, farklı nesneler tarafından aynı mesajın farklı yorumlanabilmesi olarak tanımlanabilir. Örneğin, kuş, balık ve kurbağa birer nesne olsun. Bütün bu nesnelerin ortak davranışları vardır: Hepsi hareket etme davranışına sahiptir. Ancak bu davranış her birinde farklı biçimde ortaya çıkmaktadır. Kuş uçarak, balık yüzerek, kurbağa zıplayarak bu davranışları sergiler. Böylece, ortak bir davranış tipi farklı nesneler tarafından farklı biçimlerde ortaya çıkmaktadır. Bu yapının güzel tarafı, bir eylem bir nesne tarafından yürütülürken nasıl yapıldığı ile ilgilenilmemesidir. Basitçe, bir mesaj nesneye iletildiğinde, o nesne mesajı uygun şekilde yorumlamak ve yerine getirmekten sorumlu olmaktadır.

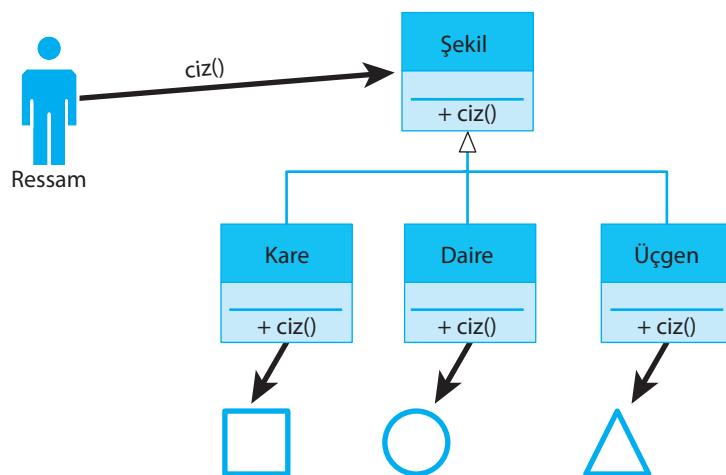
**Alt sınıfın yerine geçme**, üst sınıfının alt sınıf türü gibi davranışabilmesi ve alt sınıf türü gibi kullanılabilmesidir.

Çok biçimlilik, miras sınıf hiyerarşisi ile oluşturulmaktadır. Üst sınıfta tanımlanan davranışların aynıları alt sınıflarda da bulunur. Harekete geçirilmek istenen davranışlar için mesajlar üst sınıfı ilettilir. Üst sınıf, hangi **alt sınıfın yerine geçti** ise, o sınıf'a ait davranış ortaya çıkar. Hiyerarşide bulunan tüm alt sınıflara üst sınıf üzerinde erişilir. Böylece, üst sınıf, alt sınıflar için bir arayüz görevi üstlenmiş olur. Nesne ilişkileri arayüz nesnesi üzerinden oluşturulacağından, arayüz nesnesi altına eklenen yeni alt sınıf, sistemde az bir değişiklik yapılarak ya da hiç değişiklik yapılmadan kullanılabilir. Bu yapı, sistemin

kolay geliştirilebilir olmasını sağlar. Şekil 7.7'de görüldüğü gibi, Ressam, "Şekil" sınıfına *ciz()* mesajı göndermektedir. "Şekil" sınıfı, "Kare" sınıfının bir nesnesinin yerine geçti ise, mesaj bir kare çizimiyle; daire sınıfının bir nesnesinin yerine geçti ise mesaj bir daire çizimiyle gerçekleşmektedir.

**Şekil 7.7**

Cök Biçimlilik  
(Polymorphism)



SIRA SİZDE



Sınıf ilişkileri nelerdir? Tanımlayınız.

## UML DİYAGRAMLARI

Birleşik modelleme dili (UML-Unified Modeling Language)nin amacı, herhangi bir sistem geliştirme projesinin, analizinden tasarıma kadar modellemesi için yeterince zengin ortak bir kelime dağarcığının sağlanmasıdır. UML kapsamında yer alan diyagram yapıları, hedeflenen bilgi sisteminin modellemesinde kullanılır. Modelin, düz yazı ile ifade edilmesi yerine grafiksel olarak temsil edilmesinin önemli bir gereklisi vardır. Grafiksel gösterim, modelin daha kolay anlaşılmasını sağlar. Ayrıca, bilgi teknolojileri uzmanlarının sistem analizi ve tasarımı üzerinden daha hızlı ve doğru bir şekilde haberleşmesi, UML diyagramları kullanımı ile mümkün olabilmektedir. UML 2.5 sürümünde, 15 farklı diyagram teknigi bulunmaktadır. Bu diyagramlar, yapısal diyagramlar ve davranış diyagramları olmak üzere iki grupta toplanabilir. *Yapısal diyagramlar*, bir bilgi sisteminde bulunan veriler ve statik ilişkileri temsil etmek üzere kullanılır. *Davranış diyagramları* ise bilgi sistemini temsil eden nesneler arasındaki dinamik ilişkileri tanımlamak için kullanılır. Ayrıca bu davranış diyagramları, bir nesnenin var olduğu sürece sergilediği dinamik davranışları modellemeye de imkân sağlar. Tablo 7.1'de yapısal diyagramlar ve davranış diyagramlarının kullanılma sebepleri verilmiştir.

Diyagram Adı	Kullanılma Sebebi
<b>Yapısal Diyagramlar</b>	
Sınıf (Class)	Sistemde modellenen sınıflar ve ilişkilerini gösterir.
Nesne (Object)	Sistemde modellenen nesneler ve ilişkilerini gösterir.
Paket (Package)	Sınıflar ve diğer UML yapılarının paket içinde nasıl düzenleneceğini gösterir.
Dağılım (Deployment)	Sistemin fiziksel mimarisini gösterir. Ayrıca, fiziksel mimaride yer alan yazılım bileşenlerini göstermek üzere kullanılabilir.
Bileşen (Component)	Yazılım bileşenlerinin fiziksel ilişkilerini gösterir.
Birleşik Yapı (Composite Structure)	Bir sınıf, bileşen ya da kullanım durum'un iç yapısını gösterir.
Profil (Profile)	UML için uzantılar geliştirmekte kullanılır.
<b>Davranışsal Diyagramlar</b>	
Etkinlik (Activity)	Kullanım durum ya da iş sürecinin, sınıflardan bağımsız olarak kapsadığı etkinliklerin ardışık akışını gösterir.
Dizge (Sequence)	Bir kullanım durum ya da operasyonun işletilme sürecinde nesnelerin davranışlarını modeller. Faaliyetlerin zaman tabanlı sıralanışına odaklanır.
Haberleşme (Communication)	Bir kullanım durum ya da faaliyetlerin işletilme sürecinde nesnelerin davranışlarını modeller. İş birliği yapan nesnelerin haberleşmesine odaklanır.
Etkileşime Bakış (Interaction Overview)	Bir sürecin kontrol akışına bir bakış gösterir.
Zamanlama (Timing)	Mesaj aracılığıyla nesnelerin haberleşmesini gösterir. Ayrıca zaman ekseninde gerçekleşen durum değişimlerini gösterir.
Davranışsal Durum Makinesi (Behavioral State Machine)	Bir nesnenin yaşam döngüsü süresinde durum değişimlerini gösterir.
Protokol Durum Makinesi (Protocol State Machine)	Bir sınıfın farklı arayüzleri arasında bağımlılıkları gösterir.
Kullanım Durum (Use-case)	Sistem için iş gereksinimlerini incelemek ve sistem ile çevresi arasında etkileşimi göstermek için kullanılır.

**Tablo 7.1**  
UML 2.5 Diyagram Türleri

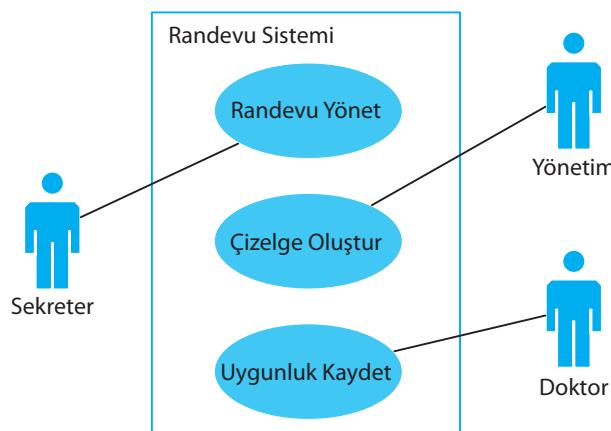
## Kullanım Durum Diyagramları

Kullanım durum'lar, bütün UML diyagram teknikleri arasında en yaygın olarak kullanılır. Kullanım durum, bilgi sistemi ile bu sistemin kullanıcıları arasındaki etkileşimi modellemek için kullanılır.

Kullanım durum diyagramlarında, aktörler, kullanım durumlar, sistem sınırı ve ilişkiler için grafiksel gösterim elemanları bulunmaktadır. Aktörler, adam şeklinde gösterilir. Aktör belirli bir kullanıcı değil, kullanıcının sistem ile etkileşiminde oynadığı roldür. Aktör aynı zamanda, sistem ile etkileşim içinde olan başka bir sistem de olabilir. Bu durumda aktör, içinde <> yazan bir dikdörtgen şeklär ile gösterilir. Kullanım durum'lar ve aktörler arasındaki ilişki, aralarında bulunan bir çizgi ile gösterilir. Kullanım durum'lar, elips şeklär ile gösterilir ve içinde kullanım durum'u belirten bir isim-fiil ifadesi yer alır. Aktörün isteği ile sistem tarafında yapılacak işlemler bu elipsler ile gösterilir. Kullanım durum'lar, bir sistem sınırı içinde toplanır. Sistem sınırı bir kutu şeklinde gösterilir ve sistemin kapsamını belirtir. Şekil 7.8'de, hastane randevu sistemindeki basit bir kullanım durum diyagramı örneği verilmiştir.

**Şekil 7.8**

Kullanım Durum  
Diyagramı Örneği



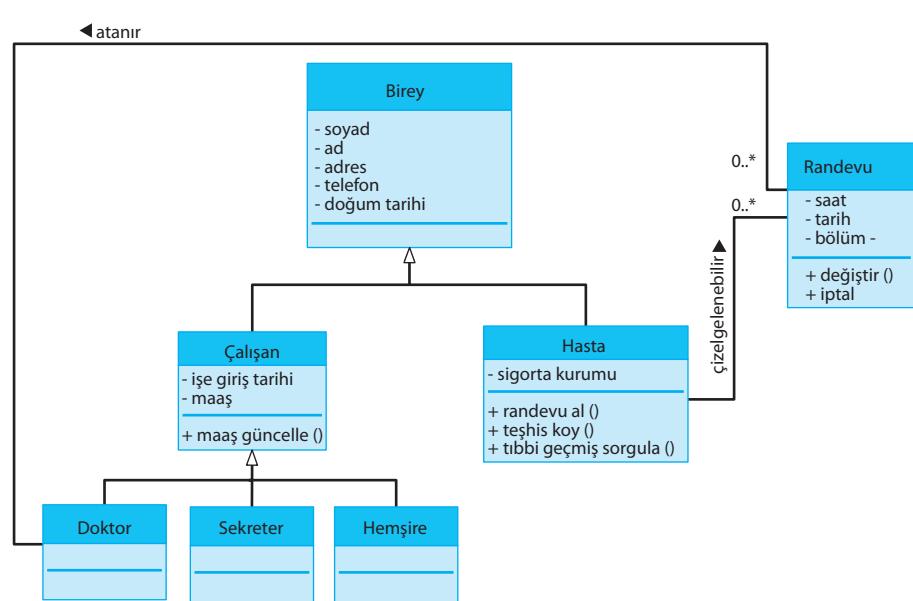
### Sınıf Diyagramları

Başlıca diyagram tekniklerinden bir diğeri, sınıf diyagramlarıdır. Sınıf diyagramları, gelişen sistemin statik görüntüsünü veren statik bir modeldir. Sınıfları ve sınıflar arasındaki ilişkileri gösterir. Bu diyagramda, ana yapısal blok sınıfıtır. Üç parçalı dikdörtgen şekli ile gösterilmektedir. En üst parçada sınıfın adı, orta parçada öznitelikleri ve en alt parçada ise fonksiyonları bulunur.

Sınıf üyelerinin erişilebilir olması, bilginin saklanma düzeyleri ile tayin edilmektedir. Buna göre, *public* tanımlı üyelerle, bütün nesneler tarafından erişilebilir. *protected* tanımlı üyelerle, sınıfın alt sınıfları dışındaki tüm sınıflar ve nesnelerin erişimi engellenir. *private* tanımlı üyelerle, bütün diğer sınıfların ve nesnelerin erişimi engellenir. Sınıfın diyagramda gösteriminde, *public* üyeleri (+), *protected* üyeleri (#) ve *private* üyeleri (-) işaretleri ile gösterilmektedir. Şekil 7.9'da hastane randevu sistemi için örnek bir sınıf diyagramı verilmiştir.

**Şekil 7.9**

Sınıf Diyagramı Örneği



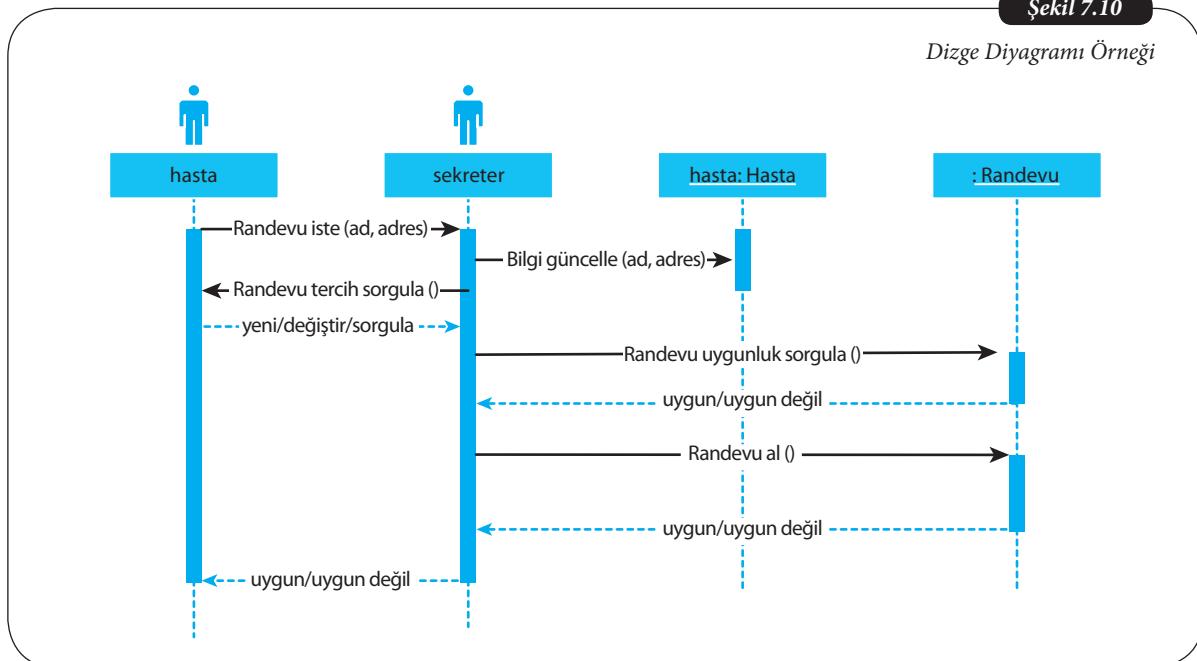
## Dizge Diyagramları

Dizge diyagramları, bir kullanım durum içindeki faaliyetleri gerçekleştiren nesneleri ve bir kullanım durum için aralarında geçen mesajları gösterir. Geliştirilen sisteme dinamik bir bakış sağlayan dinamik bir modeldir. Tanımlı bir etkileşim içinde, nesneler arasında gerçekleşen mesaj dizisi gösterilir.

Dizgenin katılımcıları (aktörler ya da nesneler), diyagramda en tepede yan yana yerleştirilir. Nesneler belirtilirken, üstüste iki noktanın solunda nesne adı, sağında ise sınıf belirtilir. Aynı isimde başka bir nesne ya da aktör yoksa üst üste iki noktanın solunda nesne adı yazılabilir (bu durum Şekil 7.10'da Randevu sınıfı ile örneklenmiştir). Aktörler, kullanım durumda bulunan sembollerle ya da dikdörtgen şekli ile gösterilmektedir. Her bir nesne ve aktörün altında dikey kesikli bir çizgi bulunmakta ve bu çizgi her biri için hayat çizgisini belirtmektedir. Ayrıca, kesikli çizgi üzerinde bulunan ince dikdörtgen, nesnenin mesaj gönderdiği ya da aldığı zamanı belirtmektedir. Nesnelerin hayat çizgileri arasında gösterilen oklar ise bir nesneden diğerine aktarılan mesajı belirtir. Şekil 7.10'da hasta randevu sistemi için bir dizge diyagramı örneği verilmiştir.

Şekil 7.10

Dizge Diyagramı Örneği



## Durum Geçiş Diyagramları

Sınıf diyagramındaki bazı sınıflar, var oldukları sürece bazı durumlar arasında geçiş yaparlar. Durum geçiş diyagramları, tek bir sınıfın olaylara tepki olarak geçiş yaptığı farklı durumları gösteren dinamik bir modeldir. Bütün sınıflar için kullanılmaz ancak, karmaşık sınıfların fonksiyonları için geliştirilen algoritmaların basitleştirilmesine yardım eder. Böylece durum geçiş diyagramları, sınıfların farklı durumlarını ve bir sınıfın bir durumdan diğerine geçişine neden olan olayları gösterir.

Bu diyagramlar oluşturulurken kullanılan elemanlardan biri **durum**dur. Durum, belli bir zamanda belli bir noktada, bir nesneyi tanımlayan değerler kümesidir. Şekil 7.11'den de görülebileceği gibi durumlar, kenarları yuvarlatılmış dikdörtgen ile gösterilir. Başlangıç durumu, küçük ve içi dolu bir daire ile gösterilir. Son durum ise bir daire ve daire içinde küçük ve içi dolu bir diğer daire ile gösterilir.

Diyagramda olaylar, nesnelerin bir durumdan diğerine geçmesinde tetikleyici olarak görev alır. Geçiş, diyagramda ok ile gösterilir ve nesnenin hangi durumdan hangisine geçeceğini gösterir. Geçişin tetikleyen olay, ok çizgisinin üzerinde bulunur. Şekil 7.11'de hasta sınıfı nesnesi için durum değişimleri gösterilmektedir.

**Şekil 7.11**

Durum Geçiş  
Diyagramı Örneği



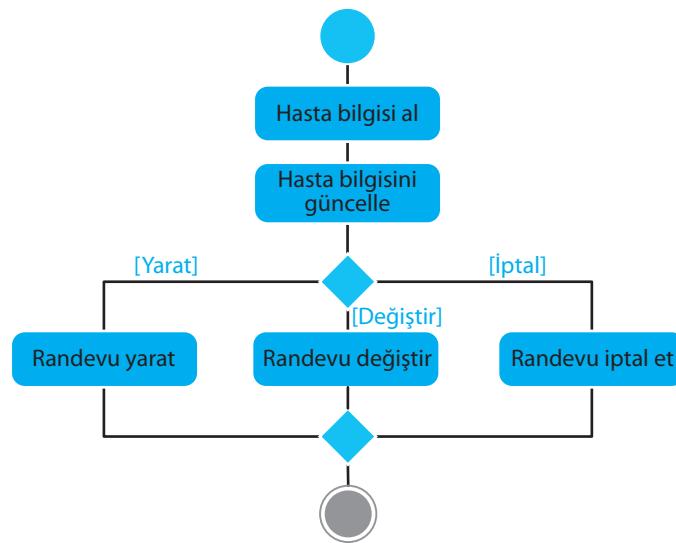
### Etkinlik Diyagramları

Kullanım durum ya da iş süreç faaliyetlerinin ardışık akışını göstermek için etkinlik diyagramları kullanılır. Etkinlik diyagramı oluşturulurken kullanılan başlıca bileşenler, eylemler ve faaliyetlerdir. Bu elemanlar, diyagramda köşeleri yumuşatılmış dikdörtgen şekli ile oluşturulurlar. İçinde, fil ve isim çiftinden oluşan kelimeler vardır. İsimler, kolayca anlaşılması için yeterince bilgi içerecek biçimde mümkün olduğunda kısa olmalıdır. Eylem ile faaliyetler arasındaki fark, faaliyetlerin daha sonra bir dizi eylem ve alt faaliyetlere bölünebilmesidir. Ancak, eylemler bölünemezler.

Ayrıca diyagramda, düğümler bulunmaktadır. Temelde, başlangıç, karar, birleşim ve bitiş düğümü olarak dört tür düğüm bulunmaktadır. *Başlangıç düğümü*, faaliyet ve eylemlerin başlangıcını tanımlar. *Bitiş düğümü*, faaliyetlerin sonlanması gösterir. Bu düğüm ayrıca bir nesne akışı ya da kontrol akışının sonlanmasını da gösterir. *Karar düğümü*, akışın hangi yoldan ilerleyeceği kararının verilmesi gereken noktayı temsil eder. *Birleşim düğümü*, birden çok akışın birleşmesini sağlar. Şekil 7.12'de randevu yönetimi kullanım durum'u için gerçekleştirilen faaliyetler diyagramda gösterilmiştir.

**Şekil 7.12**

Etkinlik Diyagramı  
Örneği



## NESNE MODELLEME SÜRECİ

Nesne tabanlı analiz gerçekleştirmenin amacı diğer sistem analiz metodlarında olduğu gibi, sistemin ve sistemden beklenen işlevlerin daha iyi anlaşılmasınadır. Böylece, nesne tabanlı analiz sürecinde, kullanıcı açısından istenen sistem fonksiyonları belirlenmektedir. Ayrıca, gereksinimlerin yerine getirilmesini sağlayacak olan nesnelerin, sahip oldukları verilerin, davranışlarının ve ilişkilerinin belirlenmesi söz konusudur.

Nesne tabanlı analiz, aşağıda sıralanan üç aşamada gerçekleştirilir:

- Sistem fonksiyonlarının modellenmesi
- Nesnelerin saptanması ve tanımlanması
- Nesnelerin düzenlenmesi ve ilişkilerinin tanımlanması

Aşağıda, bu aşamaların ayrıntılı açıklamaları verilmiştir.

### Sistem Fonksiyonlarının Modellenmesi

Bu aşamada öncelikle kullanım durum modeli oluşturulur. Sonra, kullanım durum faaliyetleri modellenir. Son olarak da sistemin dizge diyagramları çizilir.

#### Kullanım Durum Modeli Oluşturma

Daha önce (Ünite 5'te) sistem gereksinimlerinin modellenmesi sırasında elde edilen kullanım durum modelleri, nesne tabanlı analiz sürecinde, aşağıdaki adımlar izlenilerek oluşturulur.

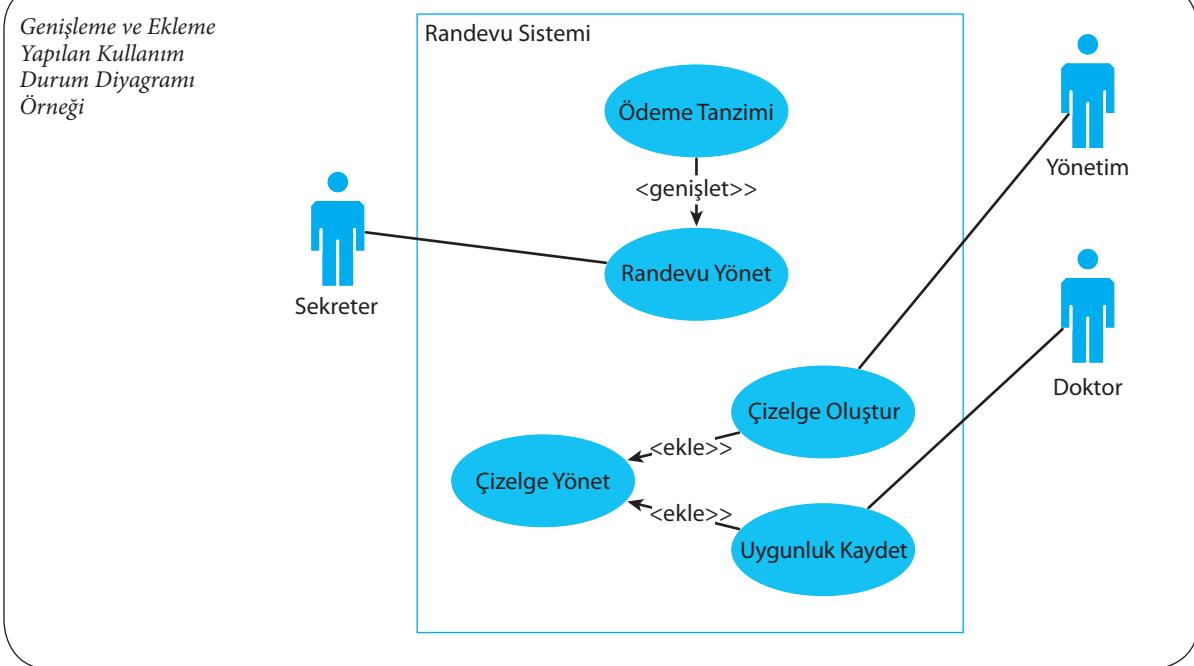
*Adım 1. Yeni aktörlerin saptanması, tanımlanması ve belgelendirilmesi:* İş gereksinimlerinin kullanım durum modellerine, nesne tabanlı analiz sırasında yeni aktörlerin eklenmesi gerekebilir. Bu aşamada, eklenecek aktörler belirlenir ve aktörlerin tanımlaması yapılır.

*Adım 2. Yeni kullanım durumlarının saptanması, tanımlanması ve belgelendirilmesi:* Yeni aktörlerin eklenmesi ile yeni kullanım durum'ların eklenmesi de söz konusu olabilir. Bu adımda, yeni kullanım durum'lar saptanır ve tanımlanır.

*Adım 3. Yeniden kullanım olasılıklarının belirlenmesi:* Aynı iş amacına sahip olan bir den çok kullanım durum'u olabilir. Bu kullanım durumları, farklı kullanıcılar için aynı adımlara sahip olabilir. Fazlalık adımları atmak için bu ortak adımlar çıkarılıp ayrı bir **soyut (abstract)** kullanım durum altında toplanabilir. Diğer taraftan, kullanım durumlar analiz edilirken, anlaşılması güçleştirilen karmaşık işlevlerin bulunduğu adımlar saptanırsa, daha karmaşık adımlar **genişleme (extension)** ve **ekleme (include)** kullanım durumlar olarak adlandırılan ayrı bir kullanım durum içinde toplanabilir. Şekil 7.13'te ekleme ve genişleme kullanılan kullanım durum diyagramı örneği verilmiştir.

*Adım 4. Kullanım durum model diyagramının düzeltmesi:* Yapılan eklemelere uygun olarak kullanım durum diyagramlarını yeniden düzenlemek gerekebilir.

*Adım 5. Sistem analizi kullanım durum hikâyelerinin belgelelenmesi:* Tüm iş gereksinimlerine göre oluşturulan kullanım durum'lar, kullanıcılar tarafından incelenir ve onaylanır. Sonra, her bir kullanım durum, sistem işlevsellliğini ayrıntılı olarak anlatacak ve daha fazla bilgi içerecek biçimde düzenlenir.

**Şekil 7.13**

### Kullanım Durum Etkinliklerini Modelleme

Bir başka diyagram tipi olan etkinlik diyagramı, süreç adımlarını ya da sistem faaliyetlerini modellemek için kullanılır. Bir iş süreci ya da kullanım durum'un faaliyet akışı grafiksel olarak gösterilir. Akış diyagramlarına benzerlik göstermesine rağmen, bu diyagram paralel işleyen faaliyetleri gösterecek mekanizmalara sahiptir. Etkinlik diyagramları oluştururken aşağıdaki süreci izletilebilir.

- Başlangıç noktası olarak, bir başlangıç düğümü ile başlanır.
- Analizle ilgili ise **bölütler** eklenir.
- Kullanım durum'un her bir temel adımı ya da aktörün başlattığı her bir temel adım için bir eylem eklenir.
- Her bir eylemden diğerine, bir akış, bir karar düğümü ya da bir bitiş düğümü eklenir. Yüksek hassasiyet için, her bir eyleme bir gelen bir de giden akış olmalıdır.
- Değişen rotalara bölünen akışlar için karar düğümleri eklenir. Akışların, birleştirici ile bir araya gelmesine de dikkat edilir.
- Etkinlikler, tek bir bitiş düğümde bitirilir.

### Sistem Dizge Diyagramlarının Çizilmesi

Sistem fonksiyonlarının modellenmesinde kullanılan bir başka araç, dizge diyagramlarıdır. Dizge diyagramları, kullanım durum'un işleyişi sırasında, mesajlar aracılığıyla nesnelerin etkileşimini gösterir. Sistem dizge diyagramlarını oluşturmak için aşağıdaki adımlar izlenebilir.

- Gösterilecek kullanım durum'un senaryosu belirlenir.
- Bir bütün olarak sistemi birleştiren bir dikdörtgen çizilir ve altında hayat çizgisi genişletilir.
- Sisteme girdi sağlayan ya da sistemin çıktısını alan aktörler belirlenir ve her birinin hayat çizgisi çizilir.

**Bölüt**, etkinlik diyagramında faaliyetleri gerçekleştirecek aktör ya da nesnelerin satır veya sütun olarak gösterildiği kısımlardır. Faaliyetler ilgili oldukları aktör ya da nesnelerin bölütleri altında gösterilir.

- Sistem girdi ve çıktılarını belirlemek için kullanım durum senaryoları incelenir. Sistem içinde gerçekleşen mesajlar dikkate alınmaz. Aktörlerden sisteme ve sistemden aktörlere giden mesajlar hayat çizgileri arasındaki yatay ok ile gösterilir. Girdiler isimlendirilir. Bunlar daha sonra, nesnelerin verilerini ve davranışlarını belirlemekte yardımcı olacaktır.
- Tercihli mesajlar için koşullarla birlikte çerçeveler eklenir. Çerçeveler ayrıca, döngüler ve alternatifleri de gösterebilir.
- Mesajların yukarıdan aşağıya doğru uygun bir sırada gösterildiği teyit edilir.

### Nesnelerin Saptanması ve Tanımlanması

Bu konuda en çok benimsenen yaklaşım, sistem gereksinim belgeleri ve diğer ilgili belgelerin taraması ve potansiyel nesne olabilecek isimlerin altlarının çizilmesi yoluna gidilir. Bu işlem neticesinde, çok sayıda isim, potansiyel nesne olarak ortaya çıkabilir. Bu sayıyı azaltmak için, sistemin kullanım durum modellerinden faydalanylabilir. Çünkü kullanım durum modelleri, sistem kapsamını kullanım durum'lar olarak bölmektedir. Aşağıdaki adımlar, nesne bulma ve tanımlama sürecinde kullanılır.

*Adım 1. Potansiyel nesneleri bul:* Bu adımda, kullanım durum'lara bakılacak ve iş varlıklarını ve olaylarını belirten isimler bulunacaktır. Bu isimlerden bir liste oluşturulacaktır.

*Adım 2. Önerilen nesneleri seç:* Oluşturulan listedeki tüm isimler bir nesne oluşturmaz. Her bir aday nesne analiz edilir, aşağıdaki sorular yöneltilir ve listede kalıp/kalma yacına karar verilir.

- Aday, başka bir nesne ile eş anlamlı mı? Ya da farklı bir isimle aynı nesne zaten var mı?
- Aday, sistemin kapsamı dışında mı?
- Aday, hiç bir davranışa sahip olmayan bir rol mü? Ya daharicibir rol mü?
- Aday, tam açık değil mi ve odaklanma ihtiyacı mı var?
- Aday, başka bir nesneyi tanımlayan bir eylem ya da özellik mi?

Eğer bu sorulardan herhangi birine verilen cevap evet ise o isim listeden çıkarılır. Eğer adaylardan özellik belirtenler varsa, onlar ayrı bir listede tutulabilir. Bu listeye, sınıf diyagramı oluşturulurken ihtiyaç duyulabilir. Eğer sorunun cevabından emin olunmazsa, aday listede bırakılır.

### Nesnelerin Düzenlenmesi ve İlişkilerinin Tanımlanması

Bir önceki aşamada aday nesneler belirlenmiş oldu. Bundan sonra, nesnelerin düzenlenmesi ve ilişkilerin tanımlanması gereklidir. Bu aşamada, sınıf diyagramları kullanılır ve aşağıdaki adımlar izlenir.

*Adım 1. İlişkileri ve çarpanı belirle:* Bu adımda, nesneler arasında bulunan ilişkiler tanımlanır. Bu ilişkiler, hangi nesnelerin arasında mesajlaşma olduğunu ve mesajın iletme yönü hakkında bilgi verir. İlişki belirlendikten sonra çarpanın da belirlenmesi gereklidir. Bir sınıf matrisi oluşturmak, ilişkilerin belirlenmesinde faydalı olacaktır. Matrisin satır ve sütunlarında nesneler bulunur ve bu matris nesneler arasında çapraz karşılaştırma yapma imkânı verir.

*Adım 2. Genelleştirme/Özelleştirme ilişkilerini tanımla:* Nesneler arasındaki temel ilişki ve çarpan belirlendikten sonra, genelleştirme ya da özelleştirme ilişkisi olup/olmadığı incelenir.

*Adım 3. Toplanma /bileşim ilişkilerinin belirlenmesi:* Bu kısımda, nesneler arasında toplanma ya da bileşim ilişkisi aranır.

*Adım 4. Sınıf diyagramı hazırlama:* Belirlenen sınıflar ve ilişkilerinin UML sınıf diyagramı olarak hazırlanması gereklidir.

**Nesnelerin saptanması ve tanımlanmasında ne yapılır? Açıklayınız.**



SIRA SİZDE

## Özet



### Nesne modellemeye ilişkin temel kavramları ve nesne modellemenin faydalarını açıklamak

Nesne tabanlı analiz ve tasarım yapmadan önce bazı kavramları bilmek gereklidir. Bu konuda en önemli kavram *nesnedir*. Nesneler, özniteliklere ve davranışlarına sahip olan yazılım bileşenleridir. Nesneler, bünyelerinde verileri ve işlemleri barındırır. Bu yapıya *kapsülleme* denir. Nesneler, *sınıf* olarak adlandırılan bir yapıda kodlanır. *Genelleştirme* ve *özelleşme* olarak adlandırılan hiyerarşik bir yapı ile sınıflar başka sınıflardan türetilebilir. Sınıflar arasında, bileşim ve toplanma olarak adlandırılan ve nesnelerin iş birliğini belirten ilişkiler tanımlanır. *Çarpan* olarak ifade edilen bir diğer kavram ise bir sınıfın bir başka sınıfın kaç tane örneği ile ilişkide olduğunu gösterir. *Çok biçimlilik*, nesnelerin ilişkisinin üst sınıflar üzerinden kurulmasını belirtir ve bu özellik ile kolay genişletilebilir yazılımlar ortaya çıkar. Nesne tabanlı programlama dillerinin kullanımı sürekli artmaktadır. Özellikle, büyük ölçüde yazılımların geliştirilmesinde birçok faydalı bulundur. Kodların, farklı sistemlerde tekrar kullanılabilmesi, hatasız kod yazmayı sağlayacak mekanizmalar sunması, bağımsız takımlarca kodlamayı mümkün kılmaktır. Nesne tabanlı tasarım yapılırken, UML olarak adlandırılan diyagramlardan faydalanailmaktadır. Bu diyagramlar, analiz, tasarım ve gerçekleme aşamalarında kolaylık ve verim sağlamaktadır.



### UML diyagram türlerini tanımlamak

Nesne tabanlı analiz ve tasarım sürecinde kullanılan önemli bir modelleme aracı, birleşik modelleme dilidir (UML). UML ile grafiksel olarak modelleme yapılır. Böylece, sistem geliştirme sürecinin paydaşları olan analistler, programcılar, yöneticiler ve müşteriler tarafından kolayca anlaşılabilecek ortak bir dil kullanılır. UML 2.5 sürümünde, yapısal ve davranışsal olarak iki grup altında toplanan 15 adet diyagram türü bulunmaktadır. Bu diyagramlardan en yaygın kullanılanları; kullanım durum diyagramları, sınıf diyagramları, dizge diyagramları, durum geçiş diyagramları ve etkinlik diyagramlarıdır. Kullanım durum diyagramları, bilgi sistemi ile bu sistemin kullanıcıları arasındaki etkileşimi modelllemek için kullanılır. Bilgi sistemindeki isterler, kullanım durum diyagramları ile modellenir. Sınıf diyagramları, sınıfların ve sınıflar arasındaki ilişkilerin gösterilmesinde kullanılır. Dizge diyagramları,

bir kullanım durum içindeki faaliyetleri gerçekleştiren nesneleri ve nesneler arasında geçen mesajları gösterir. Durum geçiş diyagramları, bir nesnenin var olduğu sürede gerçekleşen olaylara bağlı olarak değişen durumlarını gösterir. Etkinlik diyagramları, bir iş süreç faaliyetlerinin ardışık akışını göstermek için kullanılır.



### Nesne modelleme sürecini açıklamak

Nesne modelleme sürecinde gerçekleştirilen temel faaliyetler;

- sistem fonksiyonlarının modellenmesi,
- nesnelerin saptanması ve tanımlanması
- nesnelerin düzenlenmesi ve ilişkilerinin tanımlanması

olarak sıralanabilir.

Sistem fonksiyonlarının modellenmesinde, sistem gereksinimlerinin analizi yapılır. Gereksinimlere göre kullanım durum diyagramları oluşturulmaktadır. Kullanım durumlarının gerçeklenmesindeki faaliyetler, etkinlik diyagramları ile ortaya konur. Kullanım durumındaki faaliyetler gerçekleştirken, dizge diyagramları ile nesneler arası mesajlaşma gösterilmektedir. Nesnelerin saptanmasında, kullanım durumları, etkinlik diyagramları ve belgelerinden faydalananlarak, nesne olabilecek isimler belirlenip, eleme işlemleri gerçekleştirilmektedir. Son olarak, nesneler ve aralarındaki ilişkileri gösteren sınıf diyagramları çizilir.

## Kendimizi Sınavalım

- 1.** Aşağıdakilerden hangisi, nesne tabanlı programlamanın avantajlarından biri **değildir**?
- Kodların yeniden kullanımına olanak sağlama
  - Programlama maliyetlerini azaltma
  - Ayrı yazılım gruplarının coğrafi olarak farklı alanlarda bulunabilmesi
  - Farklı programcılar aynı yazılımın kod parçalarını bağımsız yazabilmesi
  - Küçük ölçekli yazılımların geliştirilme sürecini hızlandırması
- 2.** Nesne kavramı ile ilgili olarak aşağıda verilen ifadelerden hangisi **yanlıştır**?
- Özniteliklere sahip olabilir.
  - Sadece somut varlıklar olabilir.
  - Davranışlara sahip olabilir.
  - Diğer nesneler ile ilişkisi olabilir.
  - Başka nesnelere mesaj gönderebilir.
- 3.** Sınıflar arası tanımlanan ilişkiler için aşağıdakilerden hangisi doğrudur?
- Her sınıf, sadece bir sınıfın tek bir nesnesi ile ilişkili olabilir.
  - Toplanma ilişkisinde, bütün yok olduğunda parça da yok olur.
  - Bileşim ilişkisinde, bütün olmadan da parça var olmaya devam edebilir.
  - Miras ilişkisinde, alt sınıf üst sınıfın üye verileri ve fonksiyonlarına da sahiptir.
  - Çok biçimlilik kullanımında, miras hiyerarşisini kullanmak gerekmek.
- 4.** Pilot ve uçak arasındaki ilişki bağlamında aşağıdaki ifadelerden hangisi **yanlıştır**?
- Pilot, uçağa mesaj gönderebilir (mesaj gönderme, üye fonksiyon çağrıları)
  - Pilot, sıfır veya daha çok uçak nesnesi ile ilişkilendirilebilir (çarpan)
  - Pilot, uçağa toplanma ilişkisi ile bağlıdır.
  - Pilotun, aynı arayüze (pilot köşkü) sahip farklı tip uçakları ekstra eğitim almadan kullanabilmesi, çok biçimlilik kavramı ile özdeleşmektedir.
  - Pilot, uçağa bileşim ilişkisi ile bağlıdır.
- 5.** UML diyagramları ile ilgili olarak aşağıda verilen ifadelerden hangisi **yanlıştır**?
- UML diyagramları, sadece nesne tabanlı programlama için kullanılabilir.
  - UML diyagramları sistem modellemesinde kullanılır.
  - UML diyagramları, analizden tasarıma kadar kullanılabilecek kadar zengindir.
  - UML diyagramları, iki grupta toplamak mümkündür.
  - Yapısal diyagramlar, bilgi sistemindeki verileri ve statik ilişkileri temsil eder.
- 6.** Diyagramlar ve kullanılma sebepleri ile ilgili olarak aşağıda verilen ifadelerden hangisi **yanlıştır**?
- Dizge diyagramı, bir kullanım durum sürecinde nesnelerin davranışlarının zaman tabanlı sıralanmasını modeller.
  - Kullanım durum diyagramı, sistem için iş gerekliliklerini incelemek, sistem ve çevresi arasında etkileşimi göstermek için kullanılır.
  - Etkinlik diyagramı, sınıfların gerçekleştirdiği etkinliklerinin sıralanmasını göstermek için kullanılır.
  - Paket diyagramı, sınıflar ve diğer UML yapılarının mantıksal organizasyonunda kullanılır.
  - Sınıf diyagramı, sistemde modellenen sınıflar ve ilişkilerini gösterir.
- 7.** Sınıf diyagramları ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
- Sınıf diyagramları statik bir modeldir.
  - Sınıf diyagramı, sınıf adı, sınıf verileri ve fonksiyonları olmak üzere üç bölümden oluşan dikdörtgen şekli ile gösterilmektedir.
  - (+) işaretini ile sınıf üyesinin public tanımlı üye olduğu belirtilmektedir.
  - Sınıfların birbirine gönderdiği mesajlar açıkça gösterilmektedir.
  - Sistem yazılımında bulunan sınıflar ve ilişkileri grafiksel olarak gösterilmektedir.

**8.** Nesne modelleme süreci ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?

- İlk olarak gereksinimler doğrultusunda, sistem fonksiyonları modellenir.
- Kullanım durum diyagramları, sistem fonksiyonlarının modellenmesinde faydalı olur.
- Modelleme sürecinde ilk olarak nesneler belirlenir.
- Nesnelerin düzenlenmesi ve ilişkilerin tanımlanması temel etkinliklerdir.
- Etkinlik diyagramları, sistem fonksiyonlarının belirlenmesinde kullanılır.

**9.** Aşağıdakilerden hangisi sistemin işlevsel tanım modellemede, kullanım durum diyagramları için **yanlış** tanımlanmıştır?

- İş gereksinimlerinin kullanım durum modelleri, nesne tabanlı analiz esnasında yeni aktörler gerektirebilir.
- İş gereksinimlerinin kullanım durum modelleri, nesne tabanlı analiz esnasında yeni kullanım durumları gerektirebilir.
- Kullanım durumlarda karmaşık işlevsellikler saptanırsa, alt kullanım durumları oluşturulup, genişleme ya da ekleme simbolü ile ilişkilendirme yapılabilir.
- Kullanım durum diyagramları, tasarım aşamasında da oluşturulabilir.
- Kullanım durumlar, sistem işlevlerinin daha iyi anlaşılmasına ve nesnelerin belirlenmesine katkı sağlar.

**10.** Nesne modelleme süreci ile ilgili aşağıdakilerden hangisi **yanlıştır**?

- Eğer nesne adayları arasında, özellik belirten varsa silinir.
- Kullanım durum ve diğer gereksinim belgelerinde tüm isimler nesne adayı olarak listelenir.
- “Aday sistemin kapsamı dışında mı” sorusuna evet cevabına neden olan nesne adayı, nesne aday listesinden silinir.
- Nesnelerin belirlenmesinde, kullanım durum modelleri kullanılabilir.
- Nesneler belirlendikten sonra, nesne diyagramları üzerinde nesne ilişkileri gösterilir.

## Kendimizi Sınayalım Yanıt Anahtarı

- |       |   |
|-------|---|
| 1. e  | Yanınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.                                   |
| 2. b  | Yanınız yanlış ise “Nesne Modelleme İçin Sistem Kavramaları” konusunu yeniden gözden geçiriniz. |
| 3. d  | Yanınız yanlış ise “Nesne Modelleme İçin Sistem Kavramaları” konusunu yeniden gözden geçiriniz. |
| 4. e  | Yanınız yanlış ise “Nesne Modelleme İçin Sistem Kavramaları” konusunu yeniden gözden geçiriniz. |
| 5. a  | Yanınız yanlış ise “UML Diyagramları” konusunu yeniden gözden geçiriniz.                        |
| 6. c  | Yanınız yanlış ise “UML Diyagramları” konusunu yeniden gözden geçiriniz.                        |
| 7. d  | Yanınız yanlış ise “UML Diyagramları” konusunu yeniden gözden geçiriniz.                        |
| 8. c  | Yanınız yanlış ise “Nesne Modelleme Süreci” konusunu yeniden gözden geçiriniz.                  |
| 9. d  | Yanınız yanlış ise “Nesne Modelleme Süreci” konusunu yeniden gözden geçiriniz.                  |
| 10. a | Yanınız yanlış ise “Nesne Modelleme Süreci” konusunu yeniden gözden geçiriniz.                  |

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Miras sınıf ilişkisinde, üst sınıflardan türetilmiş alt sınıflar bulunmaktadır. Sınıflar arasında bir hiyerarşi vardır. Alt sınıflar, üst sınıf için tanımlanan üye verileri ve fonksiyonları kapsarlar. Ayrıca, eklenen veri ve fonksiyonlar ile gelişirler. Yeni bir sınıf oluşturulurken, mevcut sınıfın kodları kullanıldığından, kodların yeniden kullanımına imkân sağlanır. Sınıflar, kendi fonksiyonlarını yerine getiriken diğer sınıfların fonksiyonlarını da kullanabilmektedir. Bunun için, toplanma ve bileşim ilişkileri vardır. Toplanma ilişkisinde, bir nesne başka nesnelerin fonksiyonlarını kullanırken, onların örneklenmesi ya da yok olmasında rol almazlar. Bileşim ilişkisinde, bir nesne fonksiyonlarını kullanacağı nesneleri örnekler ve onlarla işi bittiğinde yok eder.

## Yararlanılan ve Başvurulabilecek Kaynaklar

### Sıra Sizde 2

Kullanım durum diyagramları, sistemle ilgili gereksinimlerin grafiksel bir gösterimle sunulmasını sağlar. Sistem kullanıcılarından alınan gereksinimler, öncelikle konulara göre ayrılır. Her konu kapsamında, sistem ile etkileşim içinde olan aktörler ve iş parçaları olarak tanımlanabilecek kullanım durum'lar belirlenir ve diyagram olarak çizilir. Böylece, analistler, kullanıcılar, sgeliştiriciler ve yöneticilerden oluşan sistem paydaşları tarafından gereksinimlerin ve sistem içinde yürütülecek işlerin analizi kolayca yapılır.

### Sıra Sizde 3

Analiz aşamasında, kullanım durum ve etkinlik diyagramları ve belgeleri nesne saptamada kullanılır. Bu belgeler taranarak, tüm isimler bir nesne adayı olarak listelenir. Daha sonra uygun nesnelerin belirlenmesi için listedeki nesne adaylarına bir dizi soru yöneltir. Bu sorulara olumsuz cevap alınan nesne adaylar listeden silinir. Listede kalan nesnelerin ayrıntılı tanımlamaları yapılır.

- Güler, Ç., (2007). **Sistem Analizi ve Tasarımı**, Nobel Yayın Dağıtım.
- Kendall, K.E. and Kendall, J.E., (2011). **System Analysis and Design**, 8th Edition, Prentice Hall.
- Roth, R.M., Dennis, A., and Wixom, B.H., (2013). **Systems Analysis and Design**, 5th Edition, John Wiley&Sons Inc.
- Tegarden, D., Dennis, A., and Wixom, B.H., (2013). **Systems Analysis and Design with UML**, 4th Edition, John Wiley&Sons Inc.
- Whitten, J. and Bentley, L., (2005). **Systems Analysis and Design Methods**, 7th Edition, McGraw-Hill.

# 8

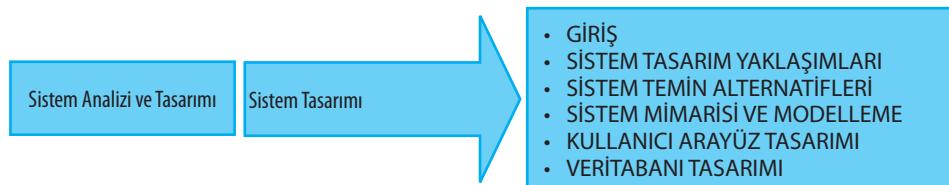
### Amaçlarımız

- Bu üniteyi tamamladıktan sonra;
- 🕒 Sistem tasarım yaklaşımlarını açıklayabilecek,
  - 🕒 Sistem temin alternatiflerini sıralayıp açıklayabilecek,
  - 🕒 Sistem mimarı tasarımını açıklayabilecek,
  - 🕒 Kullanıcı arayüz bileşenlerini açıklayabilecek,
  - 🕒 Veritabanı kavramlarını ve tasarımını tanımlayabilecek bilgi ve becerilere sahip olabileceksiniz.

### Anahtar Kavramlar

- Sistem Tasarımı
- Sistem Mimarisi
- Kullanıcı Arayüz
- Veritabanı
- Sistem Temini
- Girdi/Cıktı

### İçindekiler



# Sistem Tasarımı

## GİRİŞ

Tasarım aşamasında, mantıksal tasarımdan fiziksel tasarıma geçilmektedir. Bir başka ifadeyle, sistemin ne yapması gerekiği sorusundan, sistemin nasıl çalışacağı sorusuna geçilmiştir. Bu aşamada yapılması gereken birçok faaliyet vardır. Fiziksel tasarımını oluşturan bu faaliyetler, sistemin mimari tasarımını, veritabanı tasarımını ve kullanıcı arayüz tasarımını kapsamaktadır. Sistem analizi iş problemini ele alırken sistem tasarımını teknik ve gerçekleme üzerine odaklanır.

Sistem gereksinimlerini karşılayacak olan birkaç “sistem temin etme stratejisini” değerlendirip, hangi stratejinin uygulanacağına karar verilmesi, tasarımında önemli bir başlangıç aşamasıdır. Sistem sıfırdan oluşturulabilir, satın alınıp ihtiyaca göre özelleştirilebilir veya dışarıdan temin edilebilir. Proje ekibi, her bir alternatifin hayatı geçirilebilirlik anlamında değerlendirmelidir.

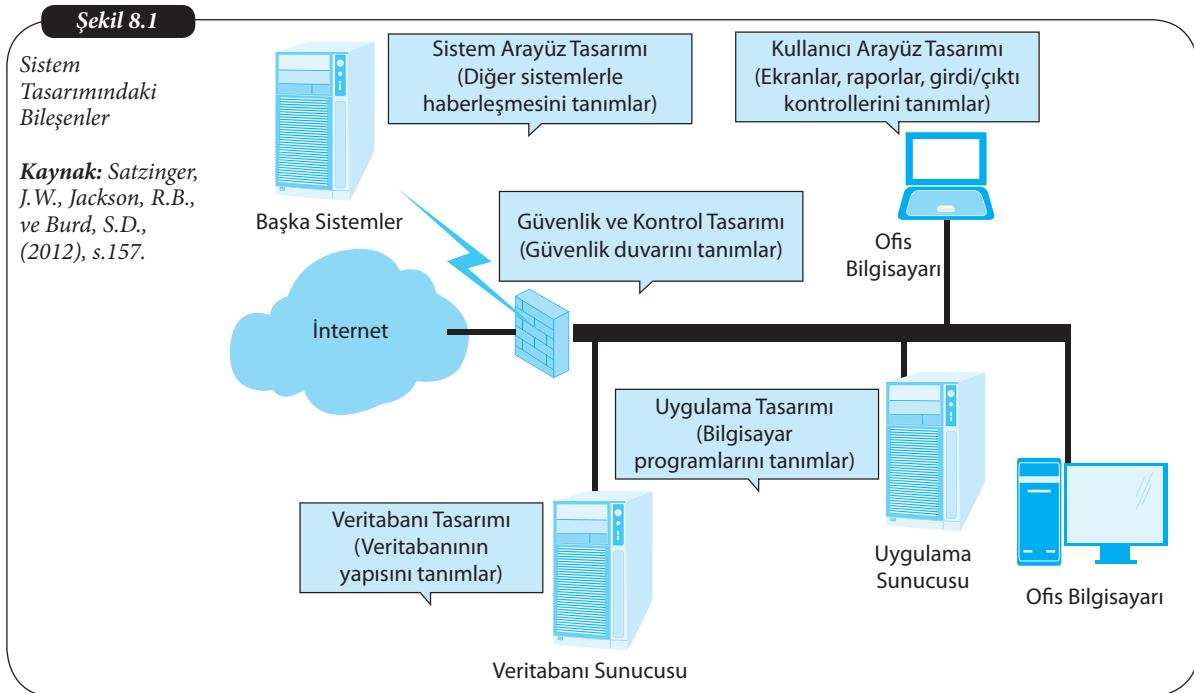
Analiz aşamasında belirlenen işlevsel olmayan iş gereksinimleri, sistemin mimari tasarımını şekillendirecek sistem gereksinimlerini etkilemektedir. Sistemin nasıl oluşturulacağına dair göz önünde bulundurulması gerekenler; işletim, performans, güvenlik, kültür ve politikadır. Örneğin, sistem ne kadar hızlı çalışacak? Kapasitesi ne olacak? Şifreleme ve virüs kontrol ihtiyacı var mı? vb. soruların cevaplandırılmasına ihtiyaç vardır. Bu şekilde belirlenen işlevsel olmayan gereksinimler daha sonra, mimari tasarım belgelerinde tanımlanan gereksinimlere dönüştürülür. Mimari kararlar, aynı zamanda yeni sistemi desteklemek üzere alınacak donanım ve yazılımlara yönelik yapılır. Bu kararlar, donanım ve yazılım özelliklerini kapsayan belgeler saklanır.

Kullanıcının, sistem ile etkileşiminin de tasarlanması gereklidir. Sistemin girdi ve çıktıları, bir plan ya da yol haritası doğrultusunda tasarlanır. Arayüzü dikkate alarak yapılan tasarım kararları, arayüz tasarımını olarak adlandırılan tasarım belgeleri biçiminde düzenlenir.

Mantıksal veri modelinde tanımlanan sistem veri bileşenlerinin, gerçekleme öncesiinde tasarım gereklidir. Veri depolamayı ilgilendiren tasarım kararları, veritabanı tasarım belgelerinde kaleme alınır.

Tasarım fazı sonunda, sistemin gerçekleştirilende kullanılacak olan, fiziksel süreç modelleri, fiziksel veri modelleri, mimari tasarım, donanım ve yazılım özellikleri, arayüz tasarımını, veritabanı tasarımını ve program tasarımını kapsayan belge elde edilir.

Günümüzün bilgi sistemleri için genel bir yapılandırmayı gösteren ağ diyagramı Şekil 8.1'de verilmiştir. Bu diyagram üzerinden, tasarımını yapılabilecek bileşenler; arayüz, uygunlama, veritabanı ve güvenlik tasarımları olarak görülmektedir.

**Şekil 8.1**

## SİSTEM TASARIM YAKLAŞIMLARI

Sistem tasarımda kullanılan birçok strateji ve teknik bulunmaktadır. Bunlar, model güdümlü yaklaşımalar, hızlı uygulama geliştirme (RAD-Rapid Application Development) ve ortak uygulama geliştirmeyi (JAD-Joint Application Development) kapsamaktadır. Bu lar, birbirine rakip alternatif sistem tasarım yaklaşımaları olarak görünür. Ancak gerçekte bazı durumlarda biri diğerinin tamamlayıcısı durumundadır.

### Model Güdümlü Yaklaşımalar

Yeni bir sistemin teknik ve gerçekleme açısından belgelenmesi amacıyla sistem modellerinin çizimine vurgu yapan sistem tasarım yaklaşımıdır. Modern yapısal tasarım, bilgi mühendisliği, prototipleme ve nesne tabanlı tasarım, model güdümlü yaklaşılara örnektir.

Model güdümlü analizde geliştirilen mantıksal modeller kullanılarak model güdümlü tasarım modelleri elde edilir. Sistem tasarım modelleri, yeni sistemin kurulması ve gerçekleme için bir tasarı niteliğindedir. Günümüzde, model tabanlı tasarım yaklaşımında bazı yazılım araçları kullanılmaktadır. Bazı tasarımcılar, model çizimlerini genel amaçlı grafik yazılımları ile oluşturmaktadır. Diğer taraftan bazı tasarımcılar da depo tabanlı Bilgisayar Destekli Yazılım Mühendisliği (CASE- Computer-Aided Software Engineering) ya da modelleme araçları kullanmaktadır. CASE araçları, güvenilirlik ve bütünlük sağlama kural tabanlı hata denetimi sunmaktadır.

Bazı model tabanlı tasarım yaklaşımaları şu şekilde tanımlanabilir:

**Modern Yapısal Tasarım:** Bu yaklaşım, programların boyutu ve karmaşıklığı ile ilgilenmesine yardımcı olur. Süreç tabanlı bir yaklaşım olan modern yapısal tasarım, sistemin süreçlerini, hiyerarşik olarak yönetilebilir bileşenlere ayırır. Böylece, gerçekleme ve yazılım güncellemesi daha kolay olur. Bu yaklaşım, yukarıdan aşağıya doğru hiyerarşik bileşenler olarak tasarlanır. Bir bileşen, bir grup koddan oluşur. Bu kod grubu, bir paragraf, bir blok, alt-program ya da alt-yordam olabilir.

*Bilgi Mühendisliği:* Bu yaklaşım, bilgi sistemlerinin planlama, analiz ve tasarımları için kullanılan, model güdümlü ve veri merkezli ancak süreçle duyarlı bir yaklaşımdır. Bilgi mühendisliği modelleri, sistemin veri ve süreçlerini gösteren ve bunları eş uyumlu hâle getiren resimlerdir. Bu yaklaşımında, veri model diyagramı en çok kullanılan araçtır.

*Prototipleme:* Küçük ölçekli, tamamlanmamış ancak sistemin çalışan bir örneğidir. Tasarımcı ve kullanıcı arasında yakın bir çalışma ilişkisini gerektiren yinelemeli bir süreçtir. Geleneksel fiziksel tasarımda, analistler tarafından girdiler, çıktılar, veritabanlarının yerleşimini ya da yapısını gösteren resimler, diyalog ve yordamların aksı çizilir. Bu tasarımlar, öngörülememiş gereksinimler neticesinde hatalara açık olan ve zaman alan bir süreçte gerçekleştirilir. Sonuçta elde edilen çizimler, uygun, tam ya da doğru olmayabilir.

Günümüzde, birçok analist ve tasarımcı, tasarımda modern mühendislik tabanlı yaklaşım olan prototiplemeyi tercih etmektedir. Bu yaklaşımın ana avantajları şunlardır:

- Son kullanıcıların aktif katılımını gerektirir. Bu durum, son kullanıcının moralini yükseltir ve projeye desteğini arttırmır.
- Bu yaklaşımda yineleme ve değişim, sistem tasarımının doğal bir sonucudur. Böylece, tasarım sırasında son kullanıcı fikrini değiştirebilir.
- Çok sıkıkla, sistem gerçeklemesi tamamlanmadan önce son kullanıcı isteklerini tam olarak bilmez. Bu yaklaşımla, son kullanıcı prototipler üzerinden isteklerini belirleyebilir. Bir sonraki adımda, yeni istekler uygulanabilir.
- Prototipler aktif olduğu için, son kullanıcı sistemi görebilir, hissedebilir ve deneyimleyebilir.
- Kabul edilen prototip, kâğıt üzerinde yapılan tasarımlar ile denktir. Tek fark vardır; hatalar önceden tespit edilmiştir.

Ayrıca, bu yaklaşımın aşağıda sıralanan dezavantajları ya da tehlikeleri bulunmakta- dir. Ancak uygun bir disiplin ile bu tehlikeler ortadan kaldırılabilir.

- Sistem analiz fazlarına olan ihtiyacı sanıldığı gibi ortadan kaldırılmaz. Prototipleme, yanlış problemleri ve fırsatları, geleneksel olarak geliştirilmiş sistemler kadar kolay çözer.
- Bir prototipleme, hiçbir zaman kâğıt üzerinde yapılan spesifikasyonların yerini alamaz. Örneğin, kâğıt üzerinde bir tasarım olmadan hiç kimse bir araba motoru prototipleyemez. Bu nedenle, prototipleme, diğer yöntemlerin yerine değil onların tamamlayıcısı olarak kullanılmalıdır. Kâğıt üzerindeki tasarımın ayrıntıları azaltılabilir ama tamamen elenemez.
- Çok sayıdaki tasarım bileşenleri prototiplemede ele alınmaz. Bu bileşenler, dikkat edilmezse unutulabilir.
- Prototiplemede, genellikle geliştirilen ilk tasarım olgunlaşmamıştır.
- Prototiplemede, sistem kapsamı genişleyebilir ve karmaşıklığı birden artabilir. Bu da kontrolü kaybetmeye neden olabilir.
- Prototip, tasarımlarda yaratıcılığı azaltabilir. Analistlerin, tasarımcıların ve son kullanıcıların, daha iyi çözümleri aramasını engelleyebilir.

*Nesne Tabanlı Tasarım:* Bu yaklaşım, en yeni tasarım stratejisidir. Bu yaklaşımında, veri ve işlem gibi kavram ayrimı ortadan kaldırılır. Bilgi sisteminin problem çözümü, nesnelerde dağıtilır.

## Hızlı Uygulama Geliştirme

Bir başka yaygın tasarım yaklaşımı, hızlı uygulama geliştirme (RAD)'dır. Bu yaklaşım, sistem geliştirmeyi hızlandırmak için, bazı tekniklerin bir arada kullanılmasını sağlar. Veri güdümlü bilgi mühendisliği gibi yapısal teknikler, prototipleme ve ortak uygulama geliştirme teknikleri ile birlikte kullanılmaktadır. Çeşitli yapısal tekniklerin birlikte kul-

lanılmasıyla, sistemlerin geliştirme süreci hızlandırılmaktadır. Kullanıcı gereksinimlerini tanımlamak ve nihai tasarım elde etmek üzere, yapısal teknikler ve prototiplemenin et-kileşimli kullanımı söz konusudur. Yapısal teknikler kullanılarak, geliştirici öncelikle iş gereksinimlerinin veri ve süreç modellerini oluşturur. Daha sonra prototipler, analist ve kullanıcıların gereksinimlerini doğrulamalarına ve süreç ile veri modellerini resmî olarak düzeltmelerine yardım eder. Süreç içinde, model hazırla, prototip yap, model hazırla, prototip yap, vb. döngü olarak devam eder. Tasarım çabasının seyri, kullanıcıların katıldığı **ortak uygulama geliştirme** oturumları ile gelişir.

### **Ortak Uygulama Geliştirme**

Ortak uygulama geliştirme (JAD), diğer sistem analiz ve tasarım yaklaşımlarının tamamlayıcısı olan bir yaklaşımındır. Bu yaklaşımında, sistem sahipleri, kullanıcıları, tasarımcıları ve kurucularının geliştirme sürecine katılımları üzerine odaklanılır. Sistem tasarım için ortak uygulama geliştirme oturumları esnasında, tasarımcılar birkaç tam gün süren atölye çalışmalarında yönetici rolünü üstlenir. Bu atölye çalışmalarında, farklı tasarım konuları ele alınır. Ortak uygulama geliştirme, hızlı uygulama geliştirme tekniğine katkı sağlayan başlıca bileşendir.

SIRA SİZDE

1

**Prototiplemenin avantajları nedir? Kisaca açıklayınız.**

### **SİSTEM TEMİN ALTERNATİFLERİ**

Tasarım aşamasına kadar yapılan çalışmalar, sistemin proje ekibi tarafından tasarlanıp, geliştirilip, uygulanacağı düşünülerek yapılmıştır. Ancak birçok projede, sistem yazılımının bir kısmı ya da tamamı dış tedarikçilerden sağlanmaktadır. Bu tercihin yapılması organizasyonlar için daha avantajlı olabilir. Bununla birlikte, bu seçenekin tercih edilmesi, bu noktaya kadar yapılan çalışmaların yok sayılacağı anlamına gelmez. Analiz aşaması hâlâ, projenin başarısı için çok önemlidir. Bu aşamada, kullanıcı ve iş gereksinimleri belirlenmiş, tanımlanmış ve netleştirilmiştir. Artık, temin edilecek türünden nelerin beklentiği bilinmektedir.

Yeni sistemin temin edilmesi için, üç yol vardır. Bunlar kurum içi geliştirme, ticari paket yazılım alımı ve hizmet alımı biçiminde sıralanabilir.

### **Kurum İçi Geliştirme**

Kurum, yeni sistemi sıfırdan kurmak isteyebilir. Bazı durumlarda ise bu tercihin yapılması kaçınılmazdır. Yeni sistem, ciddi sistem mühendisliği gerektirecek derecede, önemli olabilir ve çeşitli zorluklar içerebilir. Bu durumda, geliştiriciler proje gereksinimlerini karşılayabilecek hazır paket bir çözüm bulamazlar. **Kurum içinde geliştirmek** tek uygun seçenekktir.

Ayrıca, kurum içi sistem geliştirme, teknik yeteneklerin ve fonksiyonel bilginin işletme içinde olmasını sağlar. Kullanıcılar ile birlikte çalışan geliştiricilerin, işi anlamaları kolaylaşır. Teknoloji öğrenim eğrileri arttığı için, gelecek projelerde benzer teknolojileri daha kolay uygularlar.

Ancak diğer taraftan, bu tercih işletme için çok olumsuz sonuçlar da ortaya çıkarabilir. Kurum içi uygulama geliştirme, uzun saatler ve ağır çalışma içeren büyük bir gayret gerektirebilir. Birçok işletmede, sistem geliştirme personeli farklı projelerde istihdam ediliyor olabilir ve bu proje için zamanları olmayabilir.

Sistemi kurum içi sıfırdan geliştirmenin birçok riskleri vardır. Projenin başarılacığıının bir garantisidir. Sistem geliştiriciler diğer projelerde çalışmak üzere görevlendirilebilir. Teknik engeller, projede gecikmeye neden olabilir. Sistem kullanıcıları, uzayan zaman çizelgesinden dolayı tahammül gösteremeyebilir.

**Kurum içi geliştirme,**  
bilgi sisteminin kurum  
içinde bulunan kaynaklar  
kullanılarak geliştirilmesi olarak  
tanımlanabilir.

## Ticari Paket Yazılım Alımı

Birçok iş ihtiyacı, sadece bir işletme için geçerli değildir. Bundan dolayı da bu ihtiyaçları giderecek paket yazılımlar ticari olarak satılmaktadır. Organizasyonlar da kendi çözümlerini oluşturmak yerine, bu **ticari yazılımları** satın almayı tercih edebilmektedir. Çok sayıda amaca hizmet edecek binlerce ticari yazılım bulunmaktadır. Bu durumda, benzer bir yazılımı geliştirmek yerine satın almak, çok daha düşük maliyetli olacaktır.

Birçok işletmenin, bordro ve muhasebe ile ilgili ihtiyaçları bulunmaktadır. Bu ihtiyaçları giderecek, çok iyi paket yazılımlar bulunmaktadır. Bu yazılımlar, tedarikçinin deneyim ve uzmanlığı ile geliştirildiği için kurum içi geliştirilecek yazılıma göre her zaman daha iyi olacaktır.

Paket yazılımlar, yükleme yöneticisi gibi küçük bir fonksiyon aracından, kurumsal kaynak planlama (ERP) uygulamaları gibi büyük ölçekli sistemlere kadar geniş yelpazede olabilmektedir. Ancak büyük ölçekli yazılımların kurulumu, küçük uygulama paketlerinin kurulumuna göre daha zordur. Bu nedenle, bu yazılımların kurulumu için de işletmeler büyük maliyetlere katlanmaktadır.

Bu temin yaklaşımındaki bir problem, paket sistemleri kullanacak olan işletmenin sistem tarafından sağlanan fonksiyonellikleri kabul etmesinin gerekliliği ve mükemmel uyumun seyrek gerçekleşmesidir. Paket yazılımının kapsamı çok geniş ise sistemin işletmede kurulması ile işletmenin iş yapma sürecinde önemli değişiklikler gerekecektir. Bu nedenle, teknolojinin yapılan işi zorlaması tehlikeli olabilir.

Birçok paket uygulama, bazı özelleştirmelere ya da belli özellikleri çalışırmak için parametreleri yönetmeye imkân vermektedir. Örneğin, paket yazılım işletmenin ya da işletme logosunun ekranda görünmesini sağlayabilir. Muhasebe yazılımları, nakit akışının kaydını tutmak için farklı seçenekler sunabilmektedir. Böylece, farklı işletmelerde muhasebe uygulamaları yazılımla desteklenmektedir. Eğer yazılımın bazı özellikleri işletmenin ihtiyaçlarını karşılamıyorsa işletme içi bir çalışma sonucu paket program ile arayüze sahip bir eklienti programı yazılabilir. Ancak paket programın tedarikçileri tarafından eklienti programın desteklenmemesi nedeniyle, yazılım güncellemelerinde sorun yaşanabilir, eklienti kullanışsız duruma gelebilir.

Paket yazılım almak, kurum içi geliştirmeye göre daha kolaydır. Ancak paket yazılım temini için de başlangıçta yapılan analizlerden faydalana bilir. Satın alınacak paket yazılım için, ayrıntılı gereksinimler analiz aşamasında tanımlanmış olur. İşletme, satın alacağı paket yazılımda nelerin olması gerektiğini bilir. Böylece ihtiyaçlarına, en uygun alım tercihini yapabilir.

## Hizmet Alımı

En az kurum içi kaynak gerektiren sistem temin alternatifisi, **hizmet alımı**dır. Hizmet alımı, sistemin oluşturulması ya da temini için dış tedarikçi, geliştirici ya da hizmet sağlayıcıya işin verilmesidir. Bu yaklaşım son zamanlarda çok yaygınlaşmış durumdadır. Bu amaçla, çalışan birçok hizmet sağlayıcısı bulunmaktadır.

Hizmet alımı farklı şekillerde gerçekleştirilebilir. Hizmet alımı sağlayan işletmeler, uygulama hizmet sağlayıcıları olarak adlandırılmasında, yazılım uygulamaları ya da yazılımla ilgili hizmetleri internet üzerinden sunmaktadır. Yazılım teminindeki bu yaklaşımında, hizmet sağlayıcılarının yazılım uygulamasına ev sahipliği yapıp yönetmesi söz konusudur. Sahip oldukları, yönetikleri ve bakımını yaptıkları sunucular üzerinde bu yazılımlar çalıştırılmaktadır. İşletmeler, bu hizmet sağlayıcılara uygulama başına ya da aylık hizmet kullanım bedeli olarak ödeme yapmaktadır.

**Ticari paket yazılım alımı**  
birimindeki temin yaklaşımı, bilgi sistemi gereksinimlere uygun ve ticari olarak satılan paket yazılımın satın alınmasıdır.

**Hizmet alımı** birimindeki temin yaklaşımında, bilgi sistemi dışarıdan bir işletmeye yaptırılmakta ya da Internet üzerinden hizmet olarak kullanılmaktadır.

Bir uygulama hizmet sağlayıcısından, bir paket yazılıma erişim sağlamanın birçok avantajı vardır. İlk temin sırasında maliyeti çok düşüktür ve kısa sürede kurulumu gerçekleşir. Kullandıkça ödeme, hizmeti sık kullananlar için daha ucuzdur. Bilgi sistem personeli için yatırım ihtiyacı azalır. Ayrıca, uzmanlaşmış bilgi teknolojileri altyapı yatırımı ihtiyacı da olmayabilir.

Hizmet alımı sağlayan işletmeler, müşterilerinin ihtiyaçları doğrultusunda onlara özel uygulamalar da geliştirmektedir. İşletmenin, sistemi geliştirecek birilerini bulmasının da birçok faydası olabilir. Hizmet alımı sağlayan işletmeler, teknolojide deneyimlidir, daha çok kaynakları vardır ve deneyimli programcılara sahiptir.

Sonuçta, hizmet alımı yeni bir sistem geliştirme sürecinde iyi bir alternatifdir. Ancak kusursuz bir seçenek olarak da düşünülmemelidir. Yeni bir sistemin kurulumunda dış teşkilatçılar tercih edilirse işletmeye özel bilgilerinin paylaşılması ya da gelecekteki geliştirme kontrolünün kaybedilmesi anlamında ödünlendirilmelidir. Kurum içi uzmanlar, bu projeden kazanacakları yeteneklerden mahrum kalacaklar ve uzmanlık dış işletmelere kayacaktır.

SIRA SİZDE



**Hizmet alımı yaklaşımı ile bilgi sisteminin temin edilmesinin, avantaj ve dezavantajları nedir? Açıklayınız.**

## SİSTEM MİMARİSİ VE MODELLEME

Mimari tasarımın amacı, bilgi sisteminin yazılım bileşenlerinin sistemin donanım cihazlarına nasıl tesis edileceğinin belirlenmesidir. Mimari tasarımda, aşağıda sıralanan tasarım kararlarının verilmesine ihtiyaç vardır:

- Bilgi sistemi, merkezi mi yoksa dağıtık mı olacak? (Birçok sistem, ağ üzerinden dağıtık çalışmaktadır.)
- Bir ağ üzerinden depolanan verinin dağıtımını nasıl olacak? (Birçok modern veritabanı, dağıtık ya da ağ üzerinde çok kopyali olarak bulunabilmektedir.)
- Geliştirilecek yazılım için uygulama teknolojileri ne olacak? Hangi programlama dili ve araçları kullanılacaktır?
- Satın alınabilen ticari yazılımların bütünlendirilmesi nasıl olacak? (Ticari yazılımın gereksinimlere göre düzenlenme ihtiyacı vardır.)
- Kullanıcı arayüz uygulamasında kullanılacak teknolojiler neler olacaktır?
- Diğer sistemlerle arayüz oluşturmak için kullanılacak teknolojiler neler olacaktır?

### Mimari Tasarım Bileşenleri

Bir bilgi sisteminin ana mimari bileşenleri, yazılım ve donanımdır. Bütün yazılım sistemleri, dört temel fonksiyon yerine getirir. Bunlardan ilki *veri depolamadır*. Coğu bilgi sistemi, depolamak ve yeniden kullanmak üzere veriye ihtiyaç duyar. Bu veri, küçük bir dosyada ya da büyük veritabanlarında kaydedilebilir. Bunlar, veri ilişkili diyagramlarında belgelenen veri varlıklarıdır. İkinci fonksiyon, *veri erişim mantığıdır*. Bu, veriye erişim için gereken işlemlerdir. Yapışal Sorgu Dilinde (SQL-Structured Query Language), veritabanı sorguları bu fonksiyonu oluşturur. Üçüncü fonksiyon, *uygulama mantığıdır*. Veri akış diyagramları, kullanım durumları ve fonksiyonel gereksinimleri kapsayan belgelerdeki mantık olarak tanımlanabilir. Dördüncü fonksiyon, *sunum mantığıdır*. Kullanıcı komutlarının alınması ve bilginin kullanıcıya gösterimini kapsar (kullanıcı arayüzü). Bu dört fonksiyon, bilgi sisteminin temel yapı bloklarını oluşturmaktadır.

Sistemin donanımı ise üç temel bileşenden oluşur. Bunlar; istemci bilgisayarlar, sunucu bilgisayarlar ve bunların bağlantısını sağlayan ağ yapısıdır. İstemci bilgisayarlar, kullanıcılar tarafından kullanılan giriş ve çıkış cihazlarıdır. Masaüstü bilgisayarlar, dizüstü bilgisayarlar, tabletler ve akıllı telefonlar istemci bilgisayarlara verilebilecek örneklerdir. Sunucu bilgisayarlar ise tanımlanan haklar dâhilinde her kullanıcı tarafından erişilebilen, veri ve yazılım depolamakta kullanılan büyük bilgisayarlardır. Bilgisayarları birbirine bağlayan ağ yapısı, hız, protokol ve bağlantı topolojisi anlamında farklılık gösterebilir.

## Uygulama Mimarisi

Mimari tasarım bir dizi kararın verilmesini gerektirir. Bu kararlardan biri, bilgi sisteminin merkezi mi yoksa dağıtık mı yapılacağıdır. Günümüzde birçok çağdaş sistem, ağ üzerinden dağıtık olarak oluşturulmaktadır. Ağ üzerinden depolanan verinin dağıtımını da karar verilmesi gereken bir konudur. Veritabanları, ağ üzerinde dağıtık ya da birden fazla kopyali, müşteri/sunucu ya da ağ hesaplama örtütüsünde olabilir. Bütün yazılım kurum içinde geliştiriliyorsa gerçekleme teknolojisi seçiminde hangi programlama dili ve araçların kullanılacağı da seçilmelidir. Ticari olarak temin edilen yazılımın bütünlendirilmesi ve yazılımın özelleştirilmesi için ihtiyaçlar belirlenmelidir. Kullanıcı arayüzünün gerçekleştirilmesi için kullanılan teknoloji seçilecektir. Diğer sistemlere arayüz oluşturulmasında kullanılacak teknolojinin belirlenmesi gerekmektedir. Söz konusu bu kararlar, bilgi sistemi için uygulama mimarisini tanımlar. Buna göre, bilgi sistemini gerçeklerken kullanılan teknoloji özellikleri de uygulama mimarisini oluşturur.

Sistem mimarisinin tasarımında cevap aranan sorular nelerdir? Açıklayınız.



SIRA SİZDE

3

## KULLANICI ARAYÜZ TASARIMI

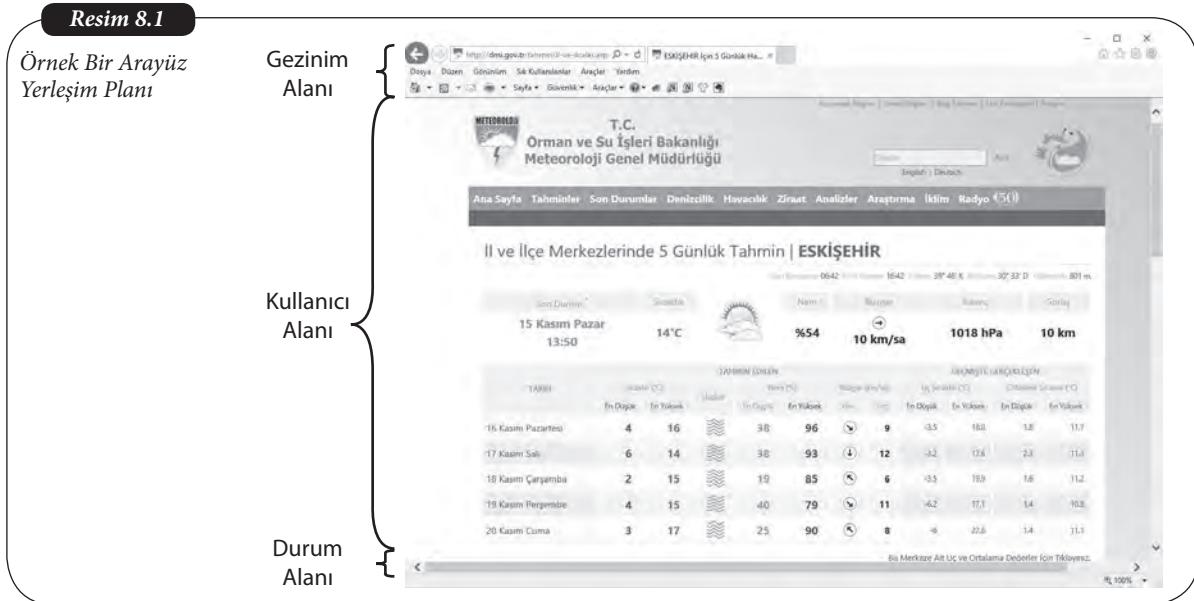
Arayüz tasarımı, haricî varlıkların (müşteriler, tedarikçiler, diğer sistemler) sistem ile nasıl etkileşimde olacağını belirleyen bir süreçtir. Arayüzler, kullanıcı arayüzü ve sistem arayüzü olarak iki grupta toplanabilir. Sistem arayüzü, sistem bütünlertirmesinin bir parçası olarak tasarılanmaktadır. Sistem arayüzü, program tasarımını ve veritabanı tasarımında ayrıntılı biçimde tasarılanmaktadır.

Kullanıcı arayüz tasarımı, kullanıcının sistemle etkileşimi ve sistemin kabul ettiği ve ürettiği girdi ve çıktıları tanımlar. Böylece, kullanıcı arayüz üç temel yapı içerir. Bunlardan ilki, *gezinim mekanizması*dır. Kullanıcı, sisteme talimatlar verir ve ne yapması gerektiğini söyler (örneğin; düğmeler, menüler vb.). İkincisi, *girdi mekanizması*dır. Sistem bilgiler alır (örneğin, yeni müşteri eklemek için formlar gibi). Üçüncüüsü, *çıktı mekanizması*dır. Sistem, kullanıcıya ya da diğer sistemlere bilgi sağlar (örneğin; raporlar, web sayfaları vb.). Tüm bunlar, kavramsal olarak ayrik olmasına rağmen, aslında iç içelerdir.

Kullanıcı arayüz tasarımında bazı temel ilkelerin yerine getirilmesi önem arz eder. Bu ilkeler doğrultusunda, tasarımını yapılan arayüz, göze hoş görünmeli, kullanımını basit olmalıdır. Aynı zamanda, kullanıcıların işlerini gerçekleştirirken sarf ettikleri çabın azaltılması da hedeflenmektedir.

*Yerleşim planı:* Kullanıcı arayüz tasarımının ilk prensibi, ekran form ve raporun yerleşim planı ile ilgilidir. Yerleşim planı, ekran ya da belge alanının düzenini belirtmektedir. Kişisel bilgisayarlar için geliştirilen birçok yazılım, Microsoft ya da Macintosh standartlarını izler. Bu yaklaşımada, ekran üç ana alana bölünür. En üst alan, kullanıcıya sistemde gezinim sağlar. Orta alan, kullanıcının çalışmalarını göstermek içindir. En alt alan, kullanıcının yaptıkları ile ilgili durum bilgisi içerir. Resim 8.1'de bir Internet tarayıcı ekran görüntüsü verilmektedir. Ekrana ait üç ana alan Resim 8.1'den izlenebilmektedir.

Resim 8.1



*İçerik bilinci:* Kullanıcı, her zaman sistemde neyin nerede olduğunu ve hangi bilginin gösterileceğinin bilincinde olmalıdır. Bu içerik bilinci, arayüze, forma ve raporlardaki alanlara, form ve raporlarda bulunan bilgilere uygulanabilir.

*Estetik:* Arayüzler tasarılanırken göze hoş gelecek şekilde olmalıdır. Ancak sanatsal objeler ile doldurulmamalıdır. Fonksiyonel ve kullanıcı için çekici olmalıdır.

*Kullanıcı deneyimleri:* Kullanıcı deneyimi, arayüz tasarılanırken kullanıcının bilgisayar bilgi düzeyinin göz önünde tutulmasını ifade eder. Bir bilgisayar sistemi, deneyimli olan ve olmayan insanlar tarafından kullanılabilir. Arayüz her iki tip kullanıcı için tasarlanmalıdır. Deneyimsiz kullanıcılar, nasıl hızlı ve kolay bir şekilde sistemi öğrenecekleriyle ilgilenirken uzman kullanıcılar, işlerini nasıl hızlıca ve kolayca tamamlayabilecekleri ile ilgilenmektedir. Çoğunlukla bu iki amaç birbirinin tamamlayıcısı durumundadır ve benzer tasarım kararları gereklidir. Ancak bazen ödünlendirme de söz konusu olabilir. Deneyimsiz kullanıcı için, menülerde tüm sistem fonksiyonlarına yer verilirken uzman kullanıcı menülerde sadece en sık kullanılan fonksiyonlardan oluşan daha az seçenekli menü tercih edebilir.

*Tutarlılık:* Tasarımda, sistemin daha basit olarak kullanılabilmesi için belki de en önemli faktördür. Kullanıcının ne olacağını kestirebilmesi gereklidir. Şöyle ki sistemin bir bölümü ile etkileşim kurduğunda, kalan bölümleri ile nasıl etkileşim kuracağının da bilir. Aynı sistemin bütün parçaları, aynı şekilde çalışır. Ayrıca, sistem diğer bilgisayar sistemleri ile organizasyonda uyumlu olmalıdır. Örneğin, birçok kullanıcı web ile tanışmıştır. Bundan dolayı, web benzeri arayüzlerin kullanımı, öğrenme için geçen zamanı azaltır.

*Kullanıcı çabasını azaltma:* Arayüz, kullanıcının işlerini tamamlamak için ihtiyaç duyduğu çaba miktarını en aza indirecek biçimde tasarlanmalıdır. Kullanıcı, en az sayıda fare tıklaması ya da klavye kullanımı ile bir noktadan diğerine erişebilmelidir. Burada, genelde üç tık kuralı kullanılmaktadır.

## Gezinim Tasarımı

Bir bilgisayar sisteminde en zor şeylelerden biri, kullanıcının sistemde gezinimi için komut girmesini ve bilginin girilmesi ve izlenmesi için eylemler gerçekleştirmesini mümkün kılmaktır. Gezinim bileşenleri, kullanıcının eylemlerinin başarı ya da başarısızlık durumu ile ilgili mesaj gösterir. Gezinim sisteminin amacı, sistem kullanımını olabildiğince kolaylaştırılmaktır. Gezinim tasarımda temel prensipler şunlardır:

Üç tık kuralı, arayüzde en fazla üç fare tıklaması ile istenen içeriğe ulaşılabilmesidir.

**Hataların önlenmesi:** İlk prensip, kullanıcının hata yapmasını engellemektir. Kullanıcı hataları, zaman kaybına ve düş kırıklığına neden olur. Sıklıkla karşılaşılan hatalar, kullanıcının sistemi kullanmaktan vazgeçmesine neden olabilir. Hataları azaltmak için, komutlar ile eylemlere uygun etiketlemelerin yapılması ve sunulan seçeneklerin sınırlandırılması gereklidir. Eğer, birden çok benzer seçenek aynı menüde bulunuyorsa ikinci düzey menü oluşturulabilir.

Kullanılamayacak komutların gösterilmemesi gereklidir. Bulunan içerikte kullanılamayacak seçenekler varsa gri tonlamalı ve seçilemez olarak gösterilebilir. Böylece, kullanıcıya, bu seçenekin bulunduğu ama içinde bulunulan durumda bunun seçilemeyeceği belirtilebilir.

Kullanıcı kritik, geriye dönüsü olmayan bir işlem seçerse kullanıcıya bu durumu belirtmek ve onay istemek gereklidir.

**Hata tafafisini basitleştirmek:** Kullanıcı, mutlaka hata yapacaktır. Sistem, kolayca bu hatayı düzeltme imkânı sunmalıdır. Sistem “geri al” düğmesine sahip olabilir ve bu düğme ile hatalı işlem öncesine dönülebilir. Ancak bazı işlemlerde bu çok daha karmaşık olabilir.

**Tutarlı gramer düzeni kullanmak:** Dil bilgisi düzeninin seçilmesi önemli bir karardır. Birçok komut, kullanıcıdan nesneyi (dosya, kayıt, kelime vb.) ve nesne üzerinde gerçekleştirilen eylemi (kopyala, sil vb.) belirtmesini gerektirir. Arayüz, önce nesne sonra eylem seçimi ya da önce eylem sonra nesne seçimini gerektirebilir. Hangisi tercih edilirse edilsin, sistemin tamamında tutarlı bir biçimde aynı düzene başvurulması gerekmektedir. Birçok kullanıcı, nesne-eylem düzenine alışktır ve günümüzde tasarımlarda bu düzen tercih edilmektedir.

## Gezinim Kontrol Türleri

Kullanıcı arayüzlerini kontrol etmek üzere kullanılan geleneksel donanım öğeleri, klavye ve işaretleme cihazlarıdır. Son zamanlarda ses tanıma sistemleri de bu amaçla kullanılmıştır. Ancak henüz, yeterince yaygınlaşmış değildir. Buna göre, kullanıcı komutlarını tanımlayan üç temel kontrol türünden bahsedilebilir: Diller, menüler, doğrudan yönetim.

**Diller:** Kullanıcı, özel bir bilgisayar dilinde komut girişini yapabilir. Örneğin, Unix ve SQL özel komut dilleri kullanır. Kullanıcı geliştiricinin önceden tanımladığı yolları kullanmadan işlemlerini yürütebildiği için, komut dili, çoğu zaman büyük esneklik sağlar. Ancak diğer taraftan çok saydaki tip ve sözdizimine sahip komutların öğrenilme güçlüğü vardır. Doğal dil arayüzleri, kullanıcının sesli komutlarını anlayarak işlem yürütür.

**Menüler:** Çok genel bir sistem gezinim türü, menülerdir. Menüler, seçenekleri kullanıcıya düzenlenmiş fonksiyonel gruplar olarak sunar. Kullanıcı, menüde yer alan seçenekleri tıklayarak ya da seçenekle ilişkilendirilen kısayolu tuşlayarak kolayca işlem yürütebilir. Bundan dolayı, yaygın olarak tercih edilen yaklaşımındır.

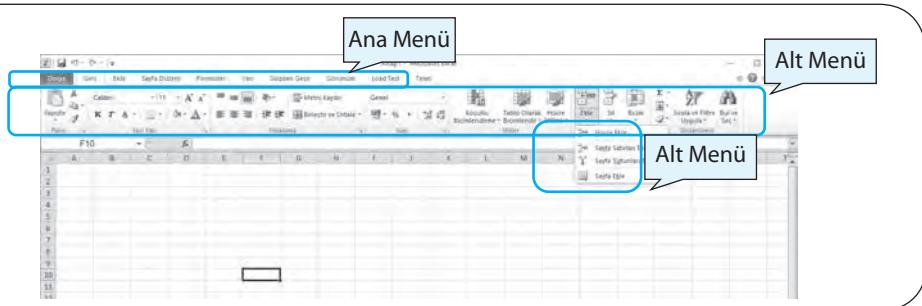
Menüler dikkatli bir şekilde tasarlanmalıdır. Alt menüler, ana menü arkasında saklıdır. Kullanıcı ana menüdeki seçiminden sonra alt menüdeki seçenekleri görebilmektedir. Ana menüde sekizden fazla seçenek olmamalıdır. Ayrıca, kullanıcının en fazla üç tıklama ile isteği eylemi gerçekleştirmiş olması beklenir. Ayrıca, menülerin kategorilere göre ayrıştırılarak oluşturulması da kullanıcının aradığı seçeneğe kolayca ulaşmasında önemlidir. Resim 8.2'de bir programda yer alan menüler görülmektedir. Ayrıca, menülerin kategorilere göre ayrıldığı yine Resim 8.2'den izlenebilmektedir. Örneğin dosya ile ilgili işlemler, “Dosya” adlı menü altında toplanmıştır.

**Doğrudan yönetim:** Bu kontrol türü ile kullanıcı, komut girişini doğrudan arayüz nesneleri üzerinde çalışarak gerçekleştirir. Örneğin, Microsoft Word programında, bir nesnenin üzerine tıklanıp, kenarlarından çekerek boyutunun değiştirilmesi ya da dosya

gezgininde bir dosyanın bir dizinden diğerine fare ile taşınması gibi. Bu yaklaşımında iki problem söz konusudur. Kullanıcılar menü kullanım alışkanlıklarını nedeniyle bu yaklaşımın kullanımını tercih etmeye bilir. Diğer taraftan, bütün komutlar için bu yaklaşım uygulanabilir olmamıştır.

Resim 8.2

Menü Örneği



### Girdi Tasarımı

Girdi mekanizmaları, sisteme veri girişini kolaylaştırır. Sisteme girilen veri, sipariş bilgisi (ürün numarası, miktarı, fiyatı vb.) gibi yapısal veri ya da yorumlar gibi yapısal olmayan veri olabilir. Girdi tasarımlı, bilgi girişi yapılan ekranın ya da bilgi yazılan belgelerin tasarımını kapsamaktadır.

Girdi denildiği zaman, genellikle klavye ve fare gibi girdi cihazları akla gelmektedir. Ancak, girdi bu cihazlara gelmeden çok önce başlar. İş verilerini bilgisayara girmek üzere, sistem analistleri kaynak belgelerini, girdi ekranlarını, veriyi bilgisayara aktarmak için metot ve yordam tasarlamak zorundadır. Bu amaçla aşağıda açıklanan temel prensipler gerçekleştirilebilir:

*Çevrimiçi ve toplu işlem, uygun olarak kullanılır.* Girdileri sisteme aktarmak üzere iki genel yaklaşım kullanılır: Çevrimiçi işleme ve toplu işleme. Çevrimiçi işleme ya da başka bir ifadeyle hareket işleme, her bir girdi kaleminin (müşteri siparişi gibi) sisteme ayrı ayrı girilmesidir. Örneğin, kütüphaneden ödünç bir kitap alırken, süreci destekleyen bilgisayar sistemi çevrimiçi işlemeye başlayarak, hareketi hemen uygun veritabanına kaydeder.

Diğer taraftan, toplu işleme yaklaşımında, belli bir zaman aralığında toplanan bütün girdiler bir araya getirilir ve sisteme bir kerede toplu olarak girilir. Örneğin bir işletmede saatlik ücret bordroları toplu işlem ile yapılır. Çünkü zaman kartları bir yığın olarak toplanır ve bir kerede işlenir.

*Veri, kaynağından tutulur.* Veri elektronik formatta, kaynağında ya da mümkün olduğunda kaynağına yakın tutulur. Birçok iş sürecinde, hâlâ bilgisayara aktarılan bilgiler aynı zamanda kağıt formlarda da tutulmaktadır. Örneğin, organizasyonlar, harcama talep formlarını elle doldurur; muhasebe bölümünde teslim eder; bu bölüm tarafından onaylanıp toplu olarak sisteme girilir. Bu sürecin, birçok problemli yönü vardır. İki kere iş yapıldığı için hem maliyetler hem de işlem süresi artmaktadır. Ayrıca, hataılma olasılığı da artmaktadır.

Günümüzde birçok hareket işleme sistemi, veriyi kaynağından almak üzere geliştirilmektedir. Hiç kimsenin formdaki bilgileri yazmasını gerektirmeden, özel cihazlarla otomatik olarak veri sisteme aktarılabilmektedir. Örneğin; barkod okuyucular, ürünü otomatik olarak tarayıp, bilgiyi sisteme aktarmaktadır. Bu durumda, kağıt formlara ihtiyaç duyulmamaktadır. Bir başka teknoloji ise optik karakter tanımadır. Bu teknoloji ile formdaki sayılar ve metin, sisteme aktarılabilmektedir. Yaygın olarak kullanılan akıllı kartlar da veri aktarımında kullanılan araçlardandır. Taşınabilir bilgisayar ve tarayıcılar ile herhangi bir konumda formlar taranarak sisteme aktarılabilmektedir.

*Klavye tuşlamaları azaltılır.* Bir başka önemli prensip, klavye tuşlamalarının azaltılmasıdır. Tuşlamalar, zaman kaybına ve maliyet artışına neden olmaktadır. Sistem, kullanıcıdan mümkün olduğunca az bilgi girmesini istemelidir. Sistem girdiyi, veritabanından çekmek ve hesaplama yapmak gibi farklı yollardan temin etmelidir. Kullanıcıdan sadece tercih yapması ve onaylamasını beklemelidir. Çoğu durumda da varsayılan değerler ile kullanıcının klavyeden bilgi girmesi azaltılabilir. Örneğin, müşteriden istenecek güncel tarih, alan kodu, fatura adresi gibi otomatik üretilebilecek ya da kullanıcıdan daha önce alınan bilgiler, müşteriye girmesi istenmeden varsayılan olarak sunulabilir.

## Girdi Türleri

Girdi yapılacak veri alanları, grafik kullanıcı arayüzdeki form üzerine yerleştirilir. Bu alanlar, veri girişinde kullanılır. Bu alanların yakın çevresinde, bir etiket bulunur. Bu etiket verinin türü ile ilgili bilgiyi kullanıcıya verir. Çok çeşitli girdi ve girdi alanları bulunmaktadır. Resim 8.3'te bazı girdi türü örnekleri izlenebilmektedir. Bunlara ilişkin örnekler aşağıda verilmiştir:

*Metin Kutusu;* adından da anlaşılacağı gibi, metin girmek için kullanılır. Bunlar, sabit boyutta olabileceği gibi, kaydırma yapılabilecek şekilde sanal olarak sınırsız metin uzunluğu da olabilir. Bir veya daha çok satırlı metinsel bilgi içerebilir. Seçme kutusu kullanılabilecekse hiçbir zaman metin kutusu kullanılmamalıdır. Metin kutuları, kes, kopyala, yapıştır gibi standart grafik kullanıcı arayüz fonksiyonlarını desteklemelidir.

*Sayı Kutuları;* sayıların girilmesi için kullanılır. Bazı yazılımlarda, girilen değer otomatik olarak biçimlendirilebilmektedir. Örneğin, 10 girdığınızda yazılım bu değeri 10.00TL olarak biçimlendirmektedir. Seçim kutularının kullanılmasının mümkün olduğu durumda sayı kutularının kullanımına başvurmamak gereklidir.

*Seçim Kutusu;* kullanıcının önceden tanımlı olan bir listeden bir değer seçmesini sağlamaktadır. Seçenekler, anlamlı bir düzende tutulur. Ayrıca, başlangıçta varsayılan değer vardır. En yaygın tercih edilen seçenek varsayılan bir değer olarak kullanılabilir.

Genel olarak kullanılan altı farklı seçim kutusu türü vardır:

*Seçme Kutusu (Check Box);* seçeneklerin tam listesini sunar ve her bir seçenekin önündeki kare kutu vardır. Seçenekler arasında birkaç tanesi seçilebileceği durumlarda kullanılabilir. Bir seçenek grubu içerisinde, 10 taneden fazla seçenek kutusu olmamalıdır.

*Seçenek Düğmesi (Radio button);* birden çok seçenek arasından sadece birinin seçilebileceği seçenek listesidir. Her bir seçenekin önünde daire bulunur. Seçeneklerden sadece birinin seçilebileceği durumda kullanılabilir. Herhangi bir listede, altıdan fazla seçenek düğmesi olmamalıdır.

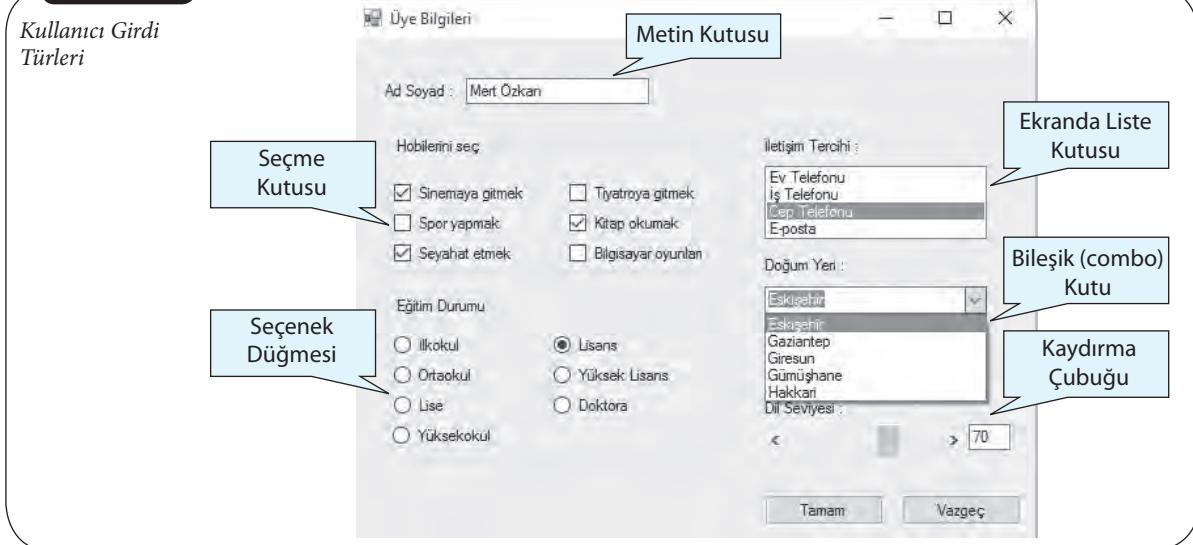
*Ekranda Liste Kutusu (On-screen list box);* bir kutu içerisinde seçeneklerin listesini sunar. Seçme kutusu ya da seçenek düğmesi için yeterli yer olmadığındada kullanılır. Seçme kutusu yerine kullanıldığında, yalnızca bir seçenekin işaretlenmesine izin verir. Seçenek düğmesi yerine kullanıldığında ise birden çok seçenekin seçilmesi için kullanılır. Kaydırma çubuğu ile birlikte çok sayıda seçeneği listede bulundurabilir.

*Açıılır Liste Kutusu (Drop-down list box);* seçilen seçeneği bir satır kutuda gösterir. Bu satır altında, bir liste açılır. Bu listeden yapılan seçim, tek satır kutuya düşer. Bütün seçeneklerin gösterileceği yeterli bir alan olmadığındada kullanılır. Seçenek düğmesine benzer ama daha sıkıştırılmış bir biçimdedir. Kullanıcı tarafından açılana kadar seçenekler saklanır.

*Bileşik Kutu (Combo Box);* açılır liste kutusunun özel bir türüdür. Kaydırılan listeden seçilebileceği gibi, kullanıcıya klavyeden yazma imkânı da sunar. Deneyimli kullanıcılar için kısayol sağlar.

**Kaydırma Çubuğu**, -dikey ya da yatay- (*Scroll Bar*); bir sayı seçmek için, kaydırma çubuğu bulunan bir grafik ölkectir. Geniş sürekli bir ölçekte, yaklaşık bir sayısal değer girmek için kullanılır. Kaydırma, hassas olarak bir sayı seçmek istendiğinde, kullanıcı için güçlüğe neden olabilir. Bazılarında sayı kutusu da bulunur. Kullanıcı doğrudan sayısal değeri de girebilir.

Resim 8.3



### Girdi Doğrulama

Sisteme girilen bütün verilerin doğruluğunun onaylanması gereklidir. Sistemin, geçersiz bir veri girildiğinde bunu engellemesi beklenir. Geçersiz veri girişi olduğunda sistem ya uyarı düzeltilir ya da verideki problemi çözmeli kullanıcısını uyarır. Girdi doğrulama amacıyla kullanılan altı farklı çeşitte onaylama kontrolü vardır.

**Tamlık Kontrolü**; gereken bütün verilerin girilmesini temin eder. Veriler üzerinde bir işlem uygulamadan önce, formda girilmesi zorunlu alanlar varsa girilmesi sağlanır.

**Biçim Kontrolü**; girilen verinin doğru tipte ve biçimde olmasını temin eder. Veri alanı sayısal ya da kodlanmış veri içeriyorsa kullanılır.

**Aralık Kontrolü**; sayısal verinin doğru minimum ve maksimum değerler arasında olmasını temin eder.

**Tutarlılık Kontrolü**; veri kombinasyonunun geçerli olduğundan emin olunmasını sağlar. Veri alanları birbirleriyle ilişkilidir. Örneğin, doğum tarihinden sonra evlilik yılı gelmelidir. Hangisinin hatalı olduğunu bilme güçlüğüne rağmen hata kullanıcıya düzeltme için raporlanabilir.

**Veritabanı Kontrolü**; verinin, veritabanı ile karşılaştırılarak doğruluğundan emin olunmasını sağlar. Veritabanı kontrolü diğer kontrollere göre maliyetli olabilir. Bu nedenle, öncelikli olarak diğer kontroller tercih edilmelidir.

### Çıktı Tasarımı

Çıktılar, sistem tarafından üretilen raporlardır. Ekranda, kâğıt üzerinde ya da daha farklı ortamlardan kullanıcıya sunulur. Aslında, sistemin en çok görünen kısmıdır. Amaç, bilginin en az çaba ile kullanıcının doğru anlayabilmesine imkân sağlayacak biçimde, kullanıcıya sunulmasıdır.

Cıktılar için, çeşitli uygulama yaklaşımları vardır.

**Basılı Çıktı:** Bilgisayar çıktısi için en yaygın ortam kâğıttır. Hâlihazırda, kâğıt en yaygın ortam olmaya devam etmektedir. Ancak henüz elde edilememiş kâğıtsız bir ofis beklenisi hep vardır. Bunun gerçekleşmemesi psikolojik de olabilir. Her durumda, kâğıtlar bir çıktı aracı olarak kullanılmaya devam etmektedir.

Kâğıt üzerinde çıktı oluşturmak için yazıcılar kullanılmaktadır. Yazdırma işlemi yapılan boş kâğıt üzerine tüm işaretlemeler ve bilgiler aktarılabilen gibi, önceden bastırılmış yerleşim düzenebine sahip formlara sadece değişken bilgiler bastırılarak da çıktı üretilmektektir.

Baskı çıktısı için en genel biçim, tablodur. **Tablo çıktı**, bilgiyi metin ya da sayı sünterlerinde sunar. Birçok bilgisayar programı, bu biçimde çıktılar üretmektedir. Bir başka alternatif ise **bölgelenmiş çıktı** olarak adlandırılmaktadır. Bölgelenmiş çıktı, metin ve sayıların tasaranmış alan ya da kutular içerisinde formattı sunulmasıdır. Bu çıktı tipi de tablo çıktı ile birlikte sıkılıkla kullanılmaktadır. Form üzerinde her iki çıktı için de alanlar bulunmaktadır.

**Ekran Çıktısı:** Bilgisayar çıktıları için en çok gelişen ortam, bilginin çevrimiçi olarak gösterildiği bilgisayar ekranıdır. Ekran çıktısı, sistem kullanıcısına bilgiye uygun erişim imkânı sunmaktadır. Ancak bilgi ekranda geçicidir. Bilgi ekrandan kaldırıldığında, kullanıcının tekrar görüntülemesine kadar kaybolur. Bundan dolayı, basılı çıktı seçeneği genellikle ekran çıktı ile birlikte sunulur. Ekran çıktısı üzerinden onaylanarak, basılı çıktı üretilmeli.

Ekran çıktısı, grafik biçimde sunulabilir. **Grafik çıktı** olarak adlandırılabilen bu çıktı biçiminde bilgi, tablo çıktıya göre daha verimli bir şekilde kullanıcıya sunulabilir.

**Çoklu ortam:** Çoklu ortam terimi, sayı, kod ve kelimeler gibi geleneksel biçim dışında sunulan bilgiyi tanımlar. Bu bilgi, grafikler, ses ve video içerebilir. Çoklu ortam çıktılarının kullanımı bilgi sistem uygulamalarının internete geçiş ile artmıştır.

Çoklu ortam, daha çok grafik çıktı olarak algılansa da diğer çoklu ortam biçimleri, geleneksel ekran tasarımda yer almaktadır. Animasyonlar, filmler vb. çoklu ortam çıktılarıdır. Örneğin, ürün tanımlamaları için çoklu ortam destekli ürün katalogları sunulmaktadır. Elbette, çoklu ortam sadece ekran üzerinden sunulan bir çıktı türü değildir. Telefon üzerinden ses, bilgi sisteminin çıktıları olarak sunulabilmektedir.

**Elektronik Posta (e-posta):** Elektronik posta, modern iş dünyasında bir çıktı tipi olarak kullanılmaktadır. Yeni bilgi sistemlerinin, e-posta üzerinden çıktı üretme imkânı bulunuş beklenmektedir. Özellikle web tabanlı alışveriş sistemlerinde, sipariş alındığında, sipariş bilgisi otomatik olarak elektronik posta çıktısi olarak müşteriye ulaştırılmaktadır.

Ayrıca sistem içerisinde üretilen çıktılar, e-posta desteği ile geliştirilebilmektedir. Örneğin, yeni üretilen bir rapor için ilgili kişilere haber vermek üzere otomatik e-posta gönderilmektedir. Kişi, rapora erişerek raporum basılı çıktısını alabilmektedir.

**Köprüler (Hyperlinks):** Günümüzde birçok çıktı, web tabanlı olarak sunulmaktadır. Birçok veritabanı ve tüketici sipariş sistemleri, web destegine sahiptir. Web bağlantıları, kullanıcının kayıt listelerine erişmesine, ilgilendiği kaydı aramasına ve farklı düzeylerde ayrıntılı bilgiye erişmesine imkân vermektedir.

Sistem tarafından oluşturulan raporları İnternet üzerinden dağıtmak için, kolayca HTML ve XML formatlarına dönüştürecek birçok teknoloji bulunmaktadır. Böylece, basılı rapora ve işletim sistemine göre değişkenlik gösteren ekran çıktısına bağımlılık ortadan kalkmaktadır. Çünkü web tarayıcıları, her bilgisayarda çalışmakta ve rapor hepsinde aynı şekilde sunulabilmektedir.

**Tablo çıktı** (Tabular output), bilginin metin ve sayı sütunları olarak basılı çıktısının üretilmesidir.

**Bölgelenmiş çıktı** (Zoned output), bilgiyi oluşturan metin ve sayıların önceden belirlenmiş ekran ya da form üzerindeki alanlara bastırılmasıdır.

**Grafik çıktı**, bilginin resimsel çizgiler ile sunulmasında kullanılan çıktı tipidir.



## VERİTABANI TASARIMI

Bilgi sistemlerinin, mutlaka veri depolama ihtiyacı bulunmaktadır. Bu durumda, analistin dikkatini vermesi gereken noktalardan biri sistemin ihtiyaç duyduğu veri depolamanın nasıl tasarlanacağıdır. Veri depolama biçiminin seçilmesi gerekmektedir. Uygulama sistemlerinin kullandığı, temelde iki tip veri depolama biçimi vardır: Dosyalar ve veritabanları. Ayrıca, her bir depolama biçiminin çok sayıda türü vardır. Örneğin, veritabanları nesne tabanlı ya da ilişkisel olabilir. Her bir tip, belirli özelliklere sahiptir ve belli koşullarda tercih edilebilir.

Veri depolama biçimine karar verildikten sonra, analiz aşamasında elde edilen veri modeli, uygulama kararları dikkate alınarak güncellenir. Mantıksal veri modelleri, fizikselleşmiş modellerine dönüştürülür.

### Veritabanı Kavramları

Bilgi sistemi tasarımda, veritabanı kavramları ve konuları, sistem analistinin bilmesi gerekenler arasında bulunmaktadır.

**Alanlar (Fields):** Bu kavram hem dosyalar hem de veritabanları için ortaktır. Alan bir dosya ya da veritabanında depolanan en küçük anlamlı veri birimidir. Dört tür alan vardır: *Birincil anahtar, ikincil anahtar, yabancı anahtar ve tanımlayıcı alan*.

**Birincil anahtar**, değeri sadece bir kaydı belirten alandır. Örneğin müşteri numarası, veritabanında sadece tek bir müşteri kaydını belirtir. Bu alandaki değer, her kayıt için diğerlerinden farklıdır.

**İkincil anahtar**, veritabanı için alternatif bir belirteçtir. Bu alan, birincil anahtar gibi sadece bir alanı belirleyebileceğini gibi, bütün kayıtların ilişkili bir alt kümelerini de belirtebilir.

**Yabancı anahtar**, bir veri tabanındaki farklı bir dosyanın kaydını işaret eder. Bir türdeki kayıtların, başka bir türdeki kayıtlara bağlantısını oluşturur. Örneğin, sipariş kaydında, müşteri numarası için yabancı anahtar bulunur ki bu müşteri kaydını belirtir.

**Tanımlayıcı alan**, veri depolayan bir başka alandır. Örneğin, bir çalışan dosyası, çalışan ismi, işe giriş tarihi gibi bazı tanımlayıcı alanlar içerir.

Bu alanlar, sistem analizi içinde veri modelleme sırasında belirlenir.

**Kayıtlar (Records):** Kayıtlar, önceden tanımlanan biçimde düzenlenmiş alanlardan oluşur. Kayıtlar da dosya ve veritabanları için ortak kullanılan bir kavramdır. Örneğin, müşteri kaydı, numara, isim, soyisim, posta numarası, adres, kredi limiti gibi alanlardan oluşabilir.

Sistem tasarımda, kayıtlar sabit boyutta ya da değişken boyutta tanımlanmaktadır. Birçok veritabanı teknolojisi, sabit boyutlu kayıt yapısını desteklemektedir. Her kayıt, aynı alana, aynı sayıda alana ve aynı mantıksal boyuta sahiptir.

Bilgisayar programı veritabanından kayıt okudüğunda, bir defada bir grup ya da blok kayıt almaktadır. Bu yaklaşım, disk erişim sayısını azaltmakta ve performansın artmasını sağlamaktadır. Tek bir okuma ya da yazma işleminde, söz konusu kayıt sayısı, bloklama faktörü olarak adlandırılır. Performansı artırmak adına bu faktör, veritabanı yöneticisi tarafından değiştirilir.

**Dosyalar ve Tablolar:** Benzer kayıtlar, dosya olarak düzenlenebilir. Veritabanı sistemlerinde, bir dosya tablo olarak adlandırılır. Bir tablo, bir dosyaya denk olan ilişkisel veritabanıdır. Bazı geleneksel dosya ve tablolar şunlardır.

**Master dosyalar ya da tablolar**, kalıcı kayıtlar içerir. Bir kayıt bu dosyaya eklendiğinde, sistem içerisinde uzun süre kalır.

**Hareket dosyaları ya da tablolar**, iş olaylarını tanımlayan kayıtlar içerir. Bu olayları tanımlayan verilerin, kısıtlı kullanım süreleri vardır. Örneğin, bir fatura kaydı, ödeme yapılmaya kadar ya da tahsil edilemeye nedeniyle silinmeye kadar kullanışlıdır.

**Belge dosyaları ve tabloları**, belgeleri tekrar oluşturma yükü olmaksızın, belgelere hızlı erişim ve gözetme için geçmiş verilerin kopyasını depolar.

*Arşiv dosyaları ve tabloları*, çevrimiçi depolama alanından silinmiş olan master ve hareket kayıtlarını içerir. Böylece kayıtlar çok seyrek silinme ile sonlanır. Kayıtlar, çevrimiçi depolama alanından çevrim dışı alana aktarılır.

*Tablo sorgu dosyaları*, tutarlılığı sağlamak ve performansı geliştirmek üzere uygulamalar ile paylaşılabilen statik verileri taşır.

*Denetim dosyaları*, özellikle master ve hareket olmak üzere diğer dosyalara güncellemlerin kaydını tutar. Kayıp verileri kurtarmak için arşiv dosyaları ile birlikte kullanılır.

**Veritabanları:** Bir veritabanı, gruplanmış bilgiler topluluğudur. Bu bilgiler bir şekilde, birbirleri ile ilişkilidir. Örneğin, bu ilişki ortak alanları üzerinden olabilir. Bilginin mantıksal gruplamaları, müşteri verisi, bir sipariş ile ilgili bilgi ve ürün bilgisi gibi kategoriler içerebilir.

*Veri Mimarisi*; veritabanı ortamında, veri bir iş kaynağıdır. Bilgi sistemleri, bilgisayar programcılara sunulmak ve son kullanıcıların esnek bir şekilde veriye erişimini sağlamak üzere bu kaynak üzerine kurulmaktadır. Veri mimarisi, bir kısım sorunun cevabını oluşturur. Bu sorular şunlardır: İş nasıl geliştirilecek ve verilerin depolanmasında dosyalar ve veritabanları nasıl kullanılacak? Hangi dosya ya da veritabanı teknolojisi kullanılacak? Veri kaynağını yönetmek üzere ne çeşit bir yönetimsel yapı kurulacak?

*Veritabanı mimarisi*; veritabanı teknolojilerini belirtir. Bu teknolojileri, veritabanı motoru, veritabanı kullanım araçları, analiz ve tasarım için CASE araçları, veritabanı uygulama geliştirme araçlarını içerir. Veritabanı mimarisinin, kontrol merkezi veritabanı yönetim sistemidir.

*Veritabanı yönetim sistemi (VTYS)*; özel bir bilgisayar yazılımıdır. Veritabanını, oluşturmak, erişmek, kontrol etmek ve yönetmek için kullanılır. Bu yazılımın çekirdeğini, *veritabanı motoru* oluşturur. Motor, aldığı belirli komutlara karşılık veritabanının yapılarını oluşturulması ve sonrasında, veritabanındaki kayıtların oluşturulması, okunması, güncellenmesi ve silinmesi işlemlerini yürütür. VTYS'ler, Oracle, IBM, Microsoft gibi veritabanı yönetim teknolojisi tedarikçilerinden satın alınır. Sistem analisti ya da veritabanı analisti, veri yapılarını tasarılar. Bu yapıarda, kayıt tipleri, o kayıt tiplerinde bulunacak alanlar ve o kayıt tipleri arasındaki ilişkileri tanımlanır. Bu yapılar, veri tanıtım dili kullanılarak veritabanı yönetim sistemine tanıtırlar. Veri tanıtım dili, veritabanı yönetim sistemi tarafından fiziksel olarak bu yapıların kurulmasını sağlar. Ayrıca veri tanıtım dili, veri tabanına farklı kullanıcılar ve programlar tarafından erişim kısıtlarını tanımlamakta da kullanılır.

Ayrıca, VTYS, uygulamalarda depollanmış veriye erişim ve kullanım için veri yönetim dili sunmaktadır. Veri yönetim dili, veritabanındaki kayıtları oluşturma, okuma, güncelleme ve silme amacıyla farklı kayıtlar ve kayıt tipleri arasında geçinmekte kullanılır. Örneğin, müşteri kaydından, sipariş kaydına geçme gibi. VTYS ve veri yönetim dili, kayıtların disk üzerinde yerlesimi ve organizasyonu gibi ayrıntıları uygulamalardan saklar. Veri yönetim dili, veritabanı üzerindeki işlemlerin çok esnek ve kolay bir şekilde yapılmasını sağlar. Ayrıca, Visual Basic, Java gibi farklı programlama dilleri ile komutlar çağrılabilir.

İlişkisel VTYS'ler günümüzde uygulama geliştirme için, en yaygın kullanılan veritabanı türüdür. Makine etkinliği düşük olmasına rağmen, geliştirme açısından bu veritabanı ile çalışmak çok kolaydır. İlişkisel veritabanları, veriyi bir dizi iki boyutlu tablolar olarak oluşturmaktadır. İki boyutlu tablolarda, bir tablo diğerine yabancı anahtar ile bağlıdır. Her bir tablo, isimli sütunlar (alanlar) ve bir dizi isimsiz satırlardan (kayıtlarla ilgili) oluşmaktadır.

**Veri tabanında kaç tür alan vardır? Her birini kısaca açıklayınız.**



SIRA SİZDE

## Özet



### Sistem tasarım yaklaşımlarını açıklamak

Sistem tasarımında kullanılan çeşitli stratejiler ve teknikler bulunmaktadır. Bunlardan yaygın olarak bilinenler, model güdümlü yaklaşımalar, hızlı uygulama geliştirme ve ortak uygulama geliştirmidir. Bu yaklaşımın bir kaçının birlikte kullanılması da mümkündür. Modern yapısal tasarım, bilgi mühendisliği, prototipleme ve nesne tabanlı tasarım, model güdümlü yaklaşımlara örnektir. Popüler tasarım stratejisi, hızlı uygulama geliştirmede, sistem geliştirmeyi hızlandırmak için, bazı yaklaşımın bir arada kullanılmasını sağlar. Veri güdümlü bilgi mühendisliği gibi yapısal yaklaşımalar, prototipleme ve ortak uygulama geliştirme teknikleri ile birlikte kullanılmaktadır. Ortak uygulama geliştirme, diğer sistem analiz ve tasarım tekniklerinin tamamlayıcısı olan bir yaklaşımındır. Bu bu yaklaşımında, sistem sahipleri, kullanıcıları, tasarımcıları ve kurucularının geliştirme sürecine katılımlına odaklanılır.



### Sistem temin alternatiflerini sıralayıp, açıklamak

Sistem geliştirme yaşam döngüsü sürecinde, tasarım aşamasında sistemin temin edilmesi için çeşitli seçenekler vardır. Bu seçenekler, sistemin kurum içinde geliştirilmesi, ticari paket yazılım olarak satın alınması ve hizmet alımı biçiminde sıralanabilir. Her seçenek, artı ve eksi özelliklere sahiptir. Sisteme ihtiyaç duyan organizasyon, kendisi için en uygun tercihi yapmalıdır. Bu tercih, önceki evrede yapılan analiz çalışmalarının çıktıları kullanılarak daha kolay yapılabilir. *Kurumiçi geliştirmede*, kurum içi kaynaklar kullanılarak sistem tasarımı ve gerçeklemesi yapılmaktadır. *Ticari paket yazılım satın alma* alternatifisi, sistemin piyasada satılan çok sayıdaki paket yazılımlardan birinin satın alınması ve kullanılması olarak tanımlanabilir. *Hizmet alımı* ise, ihtiyaç duyulan sistemin alanında uzman bir işletmeye yaptırılması olarak tanımlanabilir. Ayrıca, sistemi sıfır dan tasrarlatıp, yaptırmak yerine; sistem yazılımı, bakım sorumluluğu, sistem alt yapısı hizmet sunan tedarikçi işletmeye ait olmak üzere, sadece kullanım süresinde kullanım bedeli de ödenebilir.



### Sistem mimari tasarımını açıklamak

Mimari tasarımın amacı, bilgi sistem yazılım bileşenlerinin sistem donanımına nasıl tahsis edileceğinin belirlenmesidir. Buna göre, sistem merkezi mi dağıtık mı olacak, ağ üzerindeki verinin dağılımı nasıl olacak, uygulama teknolojileri ne olacak gibi bir dizi sorunun

cevabını vermek gerekmektedir. Mimari tasarımında, donanım ve yazılım olmak üzere iki ana bileşen vardır. Yazılım sistemleri, veri depolama, veri erişim mantığı, uygulama mantığı ve sunum mantığı olmak üzere dört temel fonksiyona bölünebilir. Sistemde, istemci, sunucu ve bağlantı sağlayan ağ olmak üzere üç temel donanım bileşeni de bulunmaktadır.



### Kullanıcı arayüz bileşenlerini açıklamak

Arayüz tasarımı, sistem dışındaki varlıkların sistem ile nasıl etkileşim kuracağını belirler. İki tip arayüzden bahsedilebilir. Bunlar, kullanıcı arayüzü ve sistem arayüzüdür. Sistem arayüzü, sistemlerin bütünlüğünü sağlar. Kullanıcı arayüzü, kullanıcının sistemle etkileşimi ve sistemin kabul ettiği ve ürettiği girdi ve çıktıları tanımlar. Kullanıcı arayüzü üç temel bileşenden oluşur. Gezinim mekanizması, sisteme talimatlar vermek ve yapması gerekenleri belirtmek için kullanılır. Girdi mekanizması, sisteme bilgi aktarılmasını sağlar. Çıktı mekanizması ise, kullanıcılarla değişik biçimlerde bilgi aktarılmasını sağlar.



### Veritabanı kavramlarını ve tasarımını tanımlamak

Bilgi sistemlerinin, mutlaka veri depolama ihtiyacı bulunmaktadır. Sistemin ihtiyaç duyduğu veri depolamanın tasarımını yapılır. Bu amaçla öncelikle veri depolama biçiminin seçilmesi gereklidir. Sistemlerde temelde iki tip veri depolama biçimi kullanılır: Dosyalar ve veritabanları. Veri depolama biçimine karar verildikten sonra, analiz aşamasında oluşturulan veri modeli, uygulama kararları dikkate alınarak güncellenebilir. Mantıksal veri modelleri, fiziksel veri modellerine dönüştürülür. Veritabanı ile ilgili önemli kavramlar; alanlar, kayıtlar, dosyalar, tablolar ve veritabanlarıdır. Alan dosya ya da veritabanında depolanan en küçük anlamlı veri birimidir. Kayıtlar, önceden tanımlanmış biçimde düzenlenmiş alanlardan oluşan yapılardır. Benzer kayıtlar dosyalar ve tablolar, biçiminde düzenlenir. Veritabanları ise, ilişkilerine göre gruplandırılmış bilgiler topluluğudur. Verilerin yapısal olarak, oluşturulması, değiştirilmesi, erişilmesi amacıyla veritabanı sistemleri kullanılır. Veritabanı yönetim sistemi, uygulama programlarında depollanmış veriye erişim ve kullanım için veri yönetim dili sunmaktadır. Bu dil, veritabanındaki kayıtları oluşturma, okuma, güncelleme ve silme amacıyla, farklı kayıtlar ve kayıt tipleri arasında geçinmekte kullanılır.

## Kendimizi Sınayalım

- 1.** Sistem Geliştirme Yaşam Döngüsü içinde, tasarım fazı ile ilgili aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. Mantıksal tasarım yapılmaktadır.
  - b. "Sistem nasıl çalışacak?" sorusuna cevap aranır.
  - c. Analiz aşamasından sonra, kurum içinde sistem tasarımı tek alternatif değildir.
  - d. Sistemin ne kadar hızlı çalışacağı ile bu aşamada ilgilendirilir.
  - e. Bu fazda, donanım ve yazılım özellikleri de belirlenmektedir.
- 2.** Sistem tasarım yaklaşımı ile ilgili aşağıdaki ifadelerden hangisi doğrudur?
  - a. Modern yapısal tasarımda, sistem süreçleri bir bütün olarak ele alınır.
  - b. Prototipleme, sistem analiz fazlarına olan ihtiyaçları reddetmez.
  - c. Ortak uygulama geliştirmede, kullanıcı ve tasarımcılar sistem analistleri aracılığı ile haberleşir.
  - d. Hızlı uygulama geliştirme, prototipleme içermez.
  - e. Prototiplemede, sistemin kontrolü kaybedilmez.
- 3.** Sistem temin alternatiflerinden ticari paket yazılım alımı ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. Sistemi kurumiçi geliştirmeye göre daha düşük maliyetlidir.
  - b. Satın alınan yazılımın kurulumu için de para ödemek gerekebilir.
  - c. Paket yazılımlar özelleştirme imkanı sunmaktadır.
  - d. Bu alternatifin seçilmesi, analiz fazının gereksiz yere yapıldığı anlamına gelir.
  - e. Karşılanmayan ihtiyaçlar için, kurumiçi ekleni geliştirme mümkündür.
- 4.** Sistem temin alternatiflerinden hizmet alımı ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. En fazla kurumiçi kaynak kullanılan alternatifdir.
  - b. Sistemin oluşturulması ve temini dış tedarikçilerden sağlanır.
  - c. Uygulama hizmet sağlayıcılarından yazılımın kullanımı için hizmet alımı yapılabilir.
  - d. Şirketin özel bilgilerinin dış işletmelerle paylaşılması gerekebilir.
  - e. Gelecekte olacak geliştirmeler için kontrol başka işletmeletmelerin eline bırakılır.
- 5.** Sistem mimarisi ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. Kullanıcı arayüz uygulamasında, kullanılacak teknolojiler belirlenir.
  - b. Ana mimari bileşenler, yazılım ve donanımdır.
  - c. Donanım bileşenleri; istemci, sunucu ve ağdır.
  - d. Bilgi sistemi merkezi mi dağıtık mı olacak kararı verilir.
  - e. Ticari yazılımlar ile entegrasyon, mimari tasarımda ele alınan bir konu değildir.
- 6.** Kullanıcı arayüz tasarımı ile ilgili olarak aşağıdaki ifadelerden hangisi doğrudur?
  - a. Çıktı mekanizmaları içermez.
  - b. Estetik olması beklenmez, önemli olan işlevselolisidir.
  - c. Sadece deneyimsiz kullanıcılarla yönelik tasarlanmalıdır.
  - d. İşlemleri yapabilmek için gereken fare tiklama sayılarında bir kısıt yoktur.
  - e. Estetik görüntü için sanatsal objelere çok fazla yer vermemelidir.
- 7.** Kullanıcı arayüz tasarımindan gezinim tasarımları ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. Sistem kullanımını kolaylaştırır.
  - b. Kullanıcı hatalarını önleyecek tasarımlar oluşturulur.
  - c. Menüler gezinim kontrol tiplerindendir.
  - d. Komut dili esneklik sağlar ve öğrenilmesi kolaydır.
  - e. Menülerin sınıflarına göre ayrılması, işlem seçimlerine kolaylık sağlar.
- 8.** Kullanıcı arayüz tasarımindan girdi ve çıktı tasarımları ile ilgili olarak aşağıdaki ifadelerden hangisi **yanlıştır**?
  - a. Basılı çıktı sadece boş kağıt üzerine üretilir.
  - b. Geçersiz bir veri girildiğinde, engellenmesi gereklidir.
  - c. Seçenek düğmesi, birden çok seçenekten sadece birinin seçilmesi için kullanılır.
  - d. Basılı çıktı, sistem bilgilerin kağıt üzerine dökülmesi ile oluşur.
  - e. Günümüzde web destekli çıktılar da yaygın olarak kullanılmaktadır.

### Kendimizi Sınayalım Yanıt Anahtarları

**9.** Veritabanı tasarımlı ile ilgili olarak aşağıdaki ifadelerden hangisi yanlışdır?

- a. Birincil anahtar değeri, her kayıt için tektir.
- b. Uygulama sistemlerinin kullandığı tek tür veri depolama formatı vardır.
- c. Kayıtlar, önceden tanımlanan biçimde düzenlenmiş alanlardan oluşur.
- d. Kayıt okuması sırasında bir blok ya da grup kayıt almaktadır.
- e. Dört tür alan vardır.

**10.** Veritabanı ile ilgili olarak aşağıdaki ifadelerden hangisi yanlışdır?

- a. Belge dosyaları ve tabloları, belgeleri tekrar oluşturma yükü olmadan, belgelere hızlı erişim ve göz atma için geçmiş verilerin kopyalarını depolar.
- b. Veri tanıtım dili, veri yönetim dili ile benzer fonksiyonlara sahiptir.
- c. Veritabanı yönetim sisteminin çekirdeğini veritabanı motoru oluşturur.
- d. Veri yönetim dili, farklı programlama dillerinde aynı komutun çağrılabilmesini sağlar.
- e. Veritabanı mimarisinin kontrol merkezi veritabanı yönetim sistemidir.

- |       |  |
|-------|--|
| 1. a  | Yanınız yanlış ise “Giriş” konusunu yeniden gözden geçiriniz.                        |
| 2. b  | Yanınız yanlış ise “Sistem Tasarım Yaklaşımları” konusunu yeniden gözden geçiriniz.  |
| 3. d  | Yanınız yanlış ise “Sistem Temin Alternatifleri” konusunu yeniden gözden geçiriniz.  |
| 4. a  | Yanınız yanlış ise “Sistem Temin Alternatifleri” konusunu yeniden gözden geçiriniz.  |
| 5. e  | Yanınız yanlış ise “Sistem Mimarisi ve Modelleme” konusunu yeniden gözden geçiriniz. |
| 6. e  | Yanınız yanlış ise “Kullanıcı Arayüz Tasarımı” konusunu yeniden gözden geçiriniz.    |
| 7. d  | Yanınız yanlış ise “Kullanıcı Arayüz Tasarımı” konusunu yeniden gözden geçiriniz.    |
| 8. a  | Yanınız yanlış ise “Kullanıcı Arayüz Tasarımı” konusunu yeniden gözden geçiriniz.    |
| 9. b  | Yanınız yanlış ise “Veritabanı Tasarımı” konusunu yeniden gözden geçiriniz.          |
| 10. b | Yanınız yanlış ise “Veritabanı Tasarımı” konusunu yeniden gözden geçiriniz.          |

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

Son kullanıcıların aktif katılımını gerektirir. Bu, son kullanımının moralini ve projeye desteğini artırır. Yinelemeli ve değişim, bu yaklaşımında sistem tasarıminın doğal sonucudur. Böylece, tasarım sırasında son kullanıcı fikrini değiştirebilir. Son kullanıcı prototipler üzerinden isteklerini belirleyebilir. Bir sonraki yinelemede, yeni istekler uygulanabilir. Prototipler aktif olduğu için son kullanıcı sistemi görebilir, hissedebilir ve deneyimleyebilir. Kabul edilen prototip, kağıt üzerinde yapılan tasarımlar ile denktir. Tek fark vardır; hatalar önceden tespit edilmiştir.

### Sıra Sizde 2

Bir uygulama hizmet sağlayıcısından, bir paket yazılıma erişim sağlamanın birçok avantajı vardır. İlk temin sırasında maliyeti çok düşüktür ve kurulumu kısa sürede kurulumu gerçekleşir. Kullandıkça ödeme, hizmeti sık kullananlar için daha ucuzdur. Bilgi sistem personeli için yatırım ihtiyacı azalır. Ayrıca, uzmanlaşmış bilgi teknolojileri altyapı yatırımı ihtiyacı da olmayabilir. Hizmet alımı sağlayan işletmeler, teknolojide deneyimlidir, daha çok kaynakları vardır ve deneyimli programcılara sahiptir. Bunula birlikte bu yaklaşımın dezavantajları da vardır. Yeni bir sistemin kurulumu dış teşkilatçılere bırakılırsa, işletmeye özel bilgilerin paylaşılması ya da gelecekteki geliştirme kontrolünün kaybedilmesi anlamında ödünlendirilmiş olunacaktır. Ayrıca, kurumiçi uzmanlar, bu projeden kazanacakları yeteneklerden mahrum kalaçaklar ve uzmanlık dış işletmelere kayacaktır.

### Sıra Sizde 3

Sistem mimarisinin tasarımında cevapları aranan sorular; bilgi sistemi, merkezi mi yoksa dağıtık mı olacak? Bir ağ üzerinden depolanan verinin dağıtımını nasıl olacak? Geliştirilecek yazılım için uygulama teknolojileri ne olacak? Satın alınabilen ticari yazılımların bütünlendirilmesi nasıl olacak? Kullanıcı arayüz uygulamasında, kullanılacak teknolojiler ne olacak? Diğer sistemlerle arayüz için kullanılacak teknolojiler neler olacak? biçiminde sıralanabilir.

### Sıra Sizde 4

Sisteme girilen bütün verilerin doğruluğunun onaylanması gereklidir. Sistemin, geçersiz bir veri girildiğinde bunu engellemesi beklenir. Altı farklı çeşitte onaylama kontrolü yapılabilir. Bunlar; tamlılık kontrolü, formal kontrolü, aralık kontrolü, tutarlılık kontrolü, veritabanı kontrolü biçiminde sıralanabilir.

### Sıra Sizde 5

Veri tabanında dört tür alan vardır. *Birincil anahtar*, değeri sadece bir kaydı belirten alandır. Bu alandaki değer, her kayıt için diğerlerinden farklıdır. *Ikincil anahtar*, veritabanı için alternatif bir belirteçdir. Bu alan, birincil anahtar gibi sadece bir alanı belirtebileceği gibi, bütün kayıtların bir ilişkili alt kümesini de belirtebilir. *Yabancı anahtar*, bir veri tabanındaki farklı bir dosyanın kaydını işaret eder. Bir tipteki kayıtların, başka bir tipteki kayıtlara bağlantısını oluşturur. *Tanımlayıcı alan*, bir başka iş verisini depolayan alandır.

## Yararlanılan ve Başvurulabilecek Kaynaklar

- Güler, Ç., (2007). **Sistem Analizi ve Tasarımı**, Nobel Yayın Dağıtım.
- Roth, R.M., Dennis, A., and Wixom, B.H., (2013). **Systems Analysis and Design**, 5th Edition, John Wiley&Sons Inc.
- Satzinger, J.W., Jackson, R.B., and Burd, S.D., (2012). **Systems Analysis and Design in Changing World**, 6th Edition, Course Technology CENGAGE Learning.
- Tegarden, D., Dennis, A., and Wixom, B.H., (2013). **Systems Analysis and Design with UML**, 4th Edition, John Wiley&Sons Inc.
- Whitten, J. and Bentley, L., (2005). **Systems Analysis and Design Methods**, 7th Edition, McGraw-Hill.