

BÖLÜM 1

SİSTEM ANALİZİNE GİRİŞ

ANAHTAR KAVRAMLAR

Sistem, Sistemin Özellikleri, Sistem Türleri, Bilgi Sistemi, Bilgi Sisteminin Bileşenleri, Bilgi Sisteminin Paydaşları, Sistem Analisti, İşletme Bilgi Sistemleri

ÖĞRENME HEDEFLERİ

- 1 Sistemin ve bilgi sisteminin tanımını yapar
- 2 Bir sistemin temel özelliklerini bilir
- 3 Bilgi sisteminin bileşenlerini açıklar
- 4 Bilgi sisteminin paydaşlarını açıklar
- 5 Sistem analistinin rolünü ve yeteneklerini bilir
- 6 İşletme bilgi sistemi türlerini açıklar

GİRİŞ



1. SİSTEM VE BİLGİ SİSTEMİ

Günlük hayatta sıkça duyduğumuz “sistem” kelimesi, “*bir araya getirmek*” anlamına gelen Yunanca “**systema**” kelimesinden gelen bir terimdir. Bu terim, farklı sektörlerde farklı şekillerde tanımlansa da özünde birlikte çalışma anlamını içermektedir. Buna göre basit bir ifade ile sistem, belirli bir amacı gerçekleştirmek için bir araya gelmiş birbirleri ile ilişkili bileşenler kümesi olarak tanımlanabilir. Bu kitap boyunca sistem için aşağıdaki tanımlama esas alınacaktır:

“Sistem, her biri açıkça belirlenmiş ve sınırlandırılmış yeteneklere sahip, bir kullanıcının belirli bir sonuç ve başarı olasılığı ile öngörülen bir işletim ortamında göreve yönelik operasyonel ihtiyaçları karşıla-

masını sağlamak için sinerjik bir şekilde çalışan, birlikte çalışabilir bir öğeler kümesidir”(Wasson, 2006).

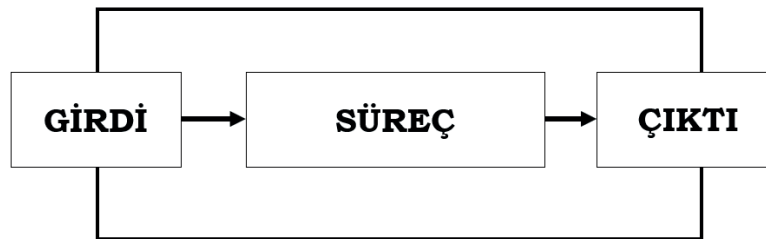
Yukarıdaki tanımlamaya bakıldığında iki önemli özellik belirginleşmektedir: Bütünleşik bir küme ve birlikte çalışabilir elemanlar. Bütünleşik küme ile, bir sistemin hiyerarşik fiziksel öğeler veya bileşenlerden oluştuğu kastedilir. Birlikte çalışabilirlik ile sistem yapısındaki elemanların biçim, uyum ve işlev açısından birbirleriyle uyumlu olması gerektiği ifade edilir.

Bu tanımlama doğrultusunda, eğitim sistemi, ekonomik ödeme sistem, işletim sistemi, güvenlik sistemi ve iletişim sistemi gibi birçok sistemin varlığından söz edilebilir.

1.1. Sistem Özellikleri

Tüm sistemlerde bir girdi ve bir çıktı bulunur. En yalın haliyle bir sistem bir girdiyi alır, işler ve

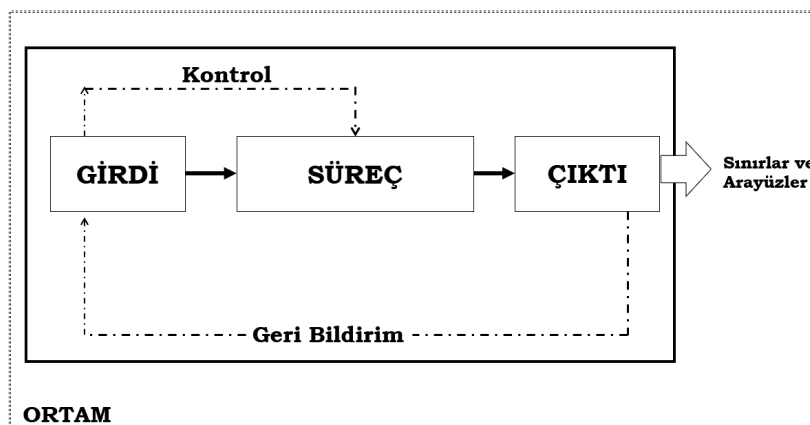
çıkıya dönüştürür. Şekil 1'de bir sistemin yalın hali gösterilmektedir.



Şekil 1- En yalın hali ile bir sistem girdi, süreç ve çıktı bileşenlerinden oluşur.

Bilgi sistemi açısından bakıldığında bir bilgi sistemi, belirli bir amaç doğrultusunda bilgiyi toplayan, depolayan, işleyen; insan, veri, süreçler ve bilgi teknolojilerinin etkileşimde bulundu-

ğu bir yapı olarak ele alınır (Parlaktuna, 2016). O halde bir bilgi sistemine; veriyi girdi olarak alan, işleyen ve sonuca dönüştüren bir mekanizma olarak bakılabilir.



Şekil 2- Bir bilgi sisteminin temel özellikleri

Bilgi sistemlerinde tutarlılık son derece hassas öneme sahip olabilir. Bu nedenle bir sistemin doğruluğunun geri bildirimler ile kontrol edilmesi gerekir. Buna göre genel bir ifade ile bir sistem Şekil2'de gösterilen özelliklerden oluşmaktadır.

- **Girdiler ve Çıktılar** : Bir sistemin esas amacı, kullanıcısı için anlamlı çıktılar üretmektir. Girdiler, işlenmek amacıyla sisteme verilen bilgiyi ifade eder. Çıktılar ise işlem sonucunda beklenen anlamlı sonuçlardır.

- **Süreç** : Girdinin çıktıya dönüşmesini sağlayan fonksiyonları ve iş adımlarını ifade eder.

- **Kontrol** : Sistemin yönlendirilmesini ve yönetilmesini sağlayan özelliktir. Sistemin girdisini, girdinin işlenmesini ve sonuç olarak çıktıya dönüştürülmesini kontrol eder.

- **Geri Bildirim** : Dinamik bir sistemin amacına uygun olarak çalışmasının denetimini sağlar.

Olumlu geri bildirimler sistemin çalışmasını teşvik ederken, olumsuz bildirimler süreçte iyileştirmeler ve güncellemeler yapılması gerektiğini ifade eder. Her iki durumda da geri bildirimler sistemin doğal olarak girdilerini oluşturur.

- **Ortam** : Bir sistemin içinde çalıştığı çerçeveyi ifade eder. Bu çerçeve, sistemin çalıştığı sektör, iş alanı ya da kurumsal yapısı ile ilişkilidir. Bu nedenle sistemin nasıl çalışacağını belirleyen dış faktörlerle de ilişkilidir. Örneğin bir işletmenin faaliyet alanı, rakipleri, tedarikçileri gibi etmenler, sistemin ne oranda ve nasıl çalışması gerektiğini değiştirebilir. Tüm bu faktörler bir sistemin performansına etki eden ortamını oluşturur.

- **Sınırlar ve Arayüzler** : Her sistemin kendi etki ve kontrol alanını belirleyen sınırları bulunur. Bunlar, başka bir sistemle arayüz oluşturduğunda bileşenlerini, süreçlerini ve aralarındaki ilişkiyi tanımlayan sınırlardır.

1.2. Sistem Türleri

Sistem kavramının farklı alanlarda farklı tanımlamalar ile gösterildiği daha önce ifade edilmişti. Bunun en önemli nedenlerinden biri, farklı türde sistemlerin varlığıdır. Bu başlıkta farklı şekillerde ayrılabilen sistem türlerine değinilecektir.

- **Soyut / Somut Sistemler** : Somut sistemler, fiziksel olarak hissedilebilen sistemler iken soyut sistemler fiziksel olmayan sistemlerdir. Örneğin bir bilgisayar sistemi somut sistemdir, oysa işletim sistemi bir soyut sistem örneğidir.

- **Açık / Kapalı Sistemler** : Açık sistemler çevresi ile etkileşim halinde olan, çevreden girdi alan ve çevreye çıktı veren sistemlerdir. İklimlendirme sistemleri açık sistemlere örnek olarak verilebilir. Kapalı sistemler çevresi ile etkileşime girmeyen, izole sistemlerdir. Gerçekte bu tür tamamen kapalı bir sistemin varlığı nadiren görülür.

- **Kalıcı / Geçici Sistemler** : Bir sistem uzun bir süre boyunca devam ediyorsa kalıcı sistem; be-

lirli bir amaç için geçici süreliğine varlığını sürdürüp görevi bittikten sonra kaldırılıyorsa geçici sistem olarak adlandırılır.

- **Uyarlanabilen / Uyarlanamayan Sistemler** : Uyarlanabilen sistemler, performanslarını artıracak ve varlığını sürdürebilecek şekilde çevresindeki değişimlere tepki verirler. Biyolojik sistemler uyarlanabilen sistemlerdir. Çevresindeki değişime tepki vermeyen dolayısıyla gelişim gösteremeyen sistemler ise uyarlanamayan sistemler olarak ele alınır.

- **Doğal / Yapay Sistemler** : Diğer bir sistem sınıflandırma türü de sistemin doğal ya da yapay sistemler olarak yer alması ile ilgilidir. Doğada var olan ve insan tarafından yapılmayan sistemler doğal sistemler iken, insanlar tarafından yapılan sistemler yapay sistemler olarak adlandırılır.

Bu sistem türlerine ait sınıflandırmalar keskin çizgiler ile ayrılmış türler değildir. Bir sistem

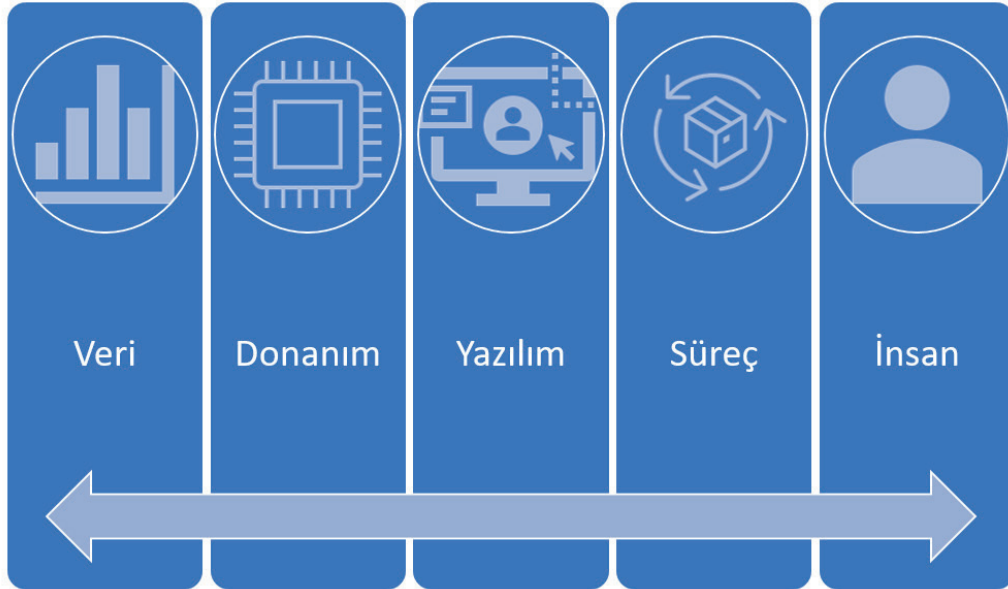
birden fazla kategoride yer alabilir. Örneğin bir sistem hem somut, hem açık, hem kalıcı hem de yapay bir sistem olabilir. Ayrıca bu bölümde yer

alan sınıflandırma dışında literatürde çeşitli sınıflandırma türlerini de görmek mümkündür.

1.3. Bilgi Sisteminin Bileşenleri

Günümüzde verinin yer aldığı hemen her alanda, teknolojik araçlarla desteklenen, veriden kullanıcılar için anlamlı bilgilerin çıkarılmasını sağlayan bilgi sistemleri görülebilir. Ayrıca bilgisayar destekli sistemlerin de sıklıkla kullanıldığını söylemek mümkündür. Bu nedenle günümüzde, bir bilgi sistemine atıfta bulunmak üzere bilgisayar uygulaması terimi kullanılmaktadır. Bir bilgisayar uygulaması, belirli bir işlevi veya ilgili bir

dizi işlevi yerine getirmek için bir bilgi işlem cihazında yürütülen bilgisayar yazılım programıdır (Satzinger et al., 2016). Çoğu zaman bilgisayar uygulaması ile bilgi sistemi kavramları birbirinin yerine kullanılsa da, bilgi sisteminin insanları ve insanların yaptığı manuel prosedürlerini içerdiği; oysa bir bilgisayar uygulamasının genellikle sadece yazılıma atıfta bulunduğu unutulmamalıdır.



Şekil 3. Bilgi sisteminin temel bileşenleri, veri, donanım, yazılım, süreç ve insandır.

Şekil 3'de gösterildiği gibi bir bilgi sistemi; insan, veri, verinin işlenmesini sağlayan teknolojik donanımlar ve yazılımlardan oluşur (Tilley, 2019).

Donanım : Donanım, bir bilgi sistemindeki tüm fiziksel yapıyı ifade eder. Sunucular, iş istasyonları, ağ altyapısı, sabit disk, tarayıcı, kamera gibi birçok teknoloji destekli fiziksel bileşenler bir bilgi sisteminin donanım bileşenini oluştururlar. Ayrıca, işlemci gibi hesaplama yeteneği olan donanımlar da bir bilgi sisteminin önemli bileşenlerindendir. Veri depolama, veri işleme, insan etkileşimi gibi bir bilgi sistemindeki görev-

ler, bu donanımların performansına doğrudan bağlıdır. Bu nedenle bir bilgi sisteminin tasarımında, beklenen performansı gösterebilmesi için fiziksel bileşenlerin doğru seçimi önemlidir.

Yazılım : Yazılım, bir bilgi sistemindeki donanımı kontrol eden ve istenen bilgi veya sonuçları üreten programları ifade eder. Genel olarak sistem yazılımları ve uygulama yazılımları olarak iki kategoride incelenir. İşletim sistemleri ve güvenlik sistemleri gibi yazılımlar sistem yazılımlarına örnektir. Sistem yazılımları donanımla etkileşimin yanı sıra, verinin akışını kontrol eder, veri güvenliğini sağlar ve ağ işlemlerinin gerçek-

leştirilmesini sağlar. Uygulama yazılımları, günlük iş fonksiyonlarını destekleyen ve kullanıcılara ihtiyaç duydukları bilgileri sağlayan programlardan oluşur. Kelime işlemci gibi yazılımlar, mobil uygulamalar ve internet tarayıcıları uygulama yazılımlarına örnek olarak verilebilir. Kimi uygulama yazılımları genel kullanıma uygun iken, kimi uygulama yazılımları ise belirli bir amaca yöneliktir. Birçok farklı işletme türünde kullanılmak üzere uyarlanabilen envanter veya bordro uygulaması gibi yazılımlar, *yatay sistemler* olarak adlandırılır. Hastane otomasyonu, online bankacılık gibi belirli bir sektöre yönelik yazılımlar ise *dikey sistemler* olarak adlandırılır (Tilley, 2019).

Bilgi sisteminin tasarımında, donanımın olduğu gibi yazılımın da önemi oldukça büyüktür.

Veri : Veri, faydalı bilgilere dönüştürülmesi beklenen bir bilgi sisteminin girdisi yani ham maddesidir. Bir bilgi sisteminde bu veriler toplanır, depolanır, işlenir ve kullanıcılar için anlamlı bilgilere dönüştürülür. Bu işlemler, donanım ve yazılımın desteği ile sağlanır. Bir bilgi sisteminde beklenen işlemlerden biri, verilerin güvenli ve güvenilir olarak saklanmasıdır. Verinin güvenli olması, yetkisiz erişime açık olmaması ile ilgili iken, güvenilir olması verinin tutarlı olmasını ifade eder. Bilgi sisteminden beklenen işlemlerden bir diğeri de, verinin işlenebilmesi ve kullanıcı açısından yorumlanabilir anlamlı bilgiye dönüştürülebilmesidir. Bilgiler grafiksel olarak gösterilebildiği gibi, verinin depolandığı yerden

(veri tabanları gibi) temel sorgulamalar ile alınabilir. Günümüzde büyük verinin (big data) de varlığı düşünüldüğünde, devasa veriler içerisinden bilginin çıkarılması (veri madenciliği gibi) bir bilgi sisteminden beklenen diğer bir işlem olarak düşünülebilir.

Süreç : Bir bilgi sisteminde süreç bileşeni, belirli bir amaca ulaşmak için, sistemin diğer bileşenlerinin uyumlu olarak görevlerini gerçekleştirmelerini ve bileşenlerin işlevlerini tanımlar. Şekil 2'de yer alan yapıda gösterildiği gibi, girdiden bir çıktının elde edilmesi tek seferlik bir işlem değil, kontrol edilen, geri bildirimlere göre değerlendirilen ve gerektiğinde tekrar güncellenen bir görevler zinciridir. Doğru tanımlanmamış bir sürecin çıktısına güvenilmez, elde edilen bilginin doğruluğu her zaman şüphelidir. Bu nedenle sistemin algoritmasını ifade eden süreç de diğer bileşenler gibi bir bilgi sisteminin önemli diğer bir bileşenidir.

İnsan : Bilgi sisteminin kullanıcısı, faydalancı ya da daha genel bir ifade ile bu sistemle ilgili olan insanlara bilgi sisteminin paydaşları denir. Paydaşlar, sistemden sorumlu yönetim grubu üyelerini, sistemle etkileşime girecek işletme içindeki ve dışındaki kullanıcıları (son kullanıcılar olarak adlandırılır) ve sistemi geliştiren ve destekleyen sistem analistlerini, programcıları ve ağ yöneticileri gibi BT personeli üyelerinden oluşur. Bilgi sisteminin paydaşları ayrı bir başlık altında detaylı olarak incelenecektir.

1.4. Bilgi Sisteminin Paydaşları

Her sistemin bir amacı, görevi yani varlığının bir nedeni vardır. İnsan yapımı sistemlerin çoğu, işletmenin misyonlarına ve hedeflerine katkıda bulunmaya ve bunları gerçekleştirmeye yöneliktir. O halde bir bilgi sisteminin sahibi, kullanıcısı, yararlanıcısı ve geliştiricisi gibi insana ilişkin rolleri ve paydaşları bulunur. Genel olarak bilgi

sisteminin paydaşları, sistem sahipleri, sistem kullanıcıları, sistem tasarımcıları, sistem kurucuları ve sistem analisti olmak üzere beş grupta incelenir (Whitten & Bentley, 2007). Her bir paydaşın sistemdeki rolü ve beklentileri de farklılık göstermektedir. Bu bölümde bilgi sisteminin paydaşları ele alınacaktır.

1.5. Sistem Sahipleri

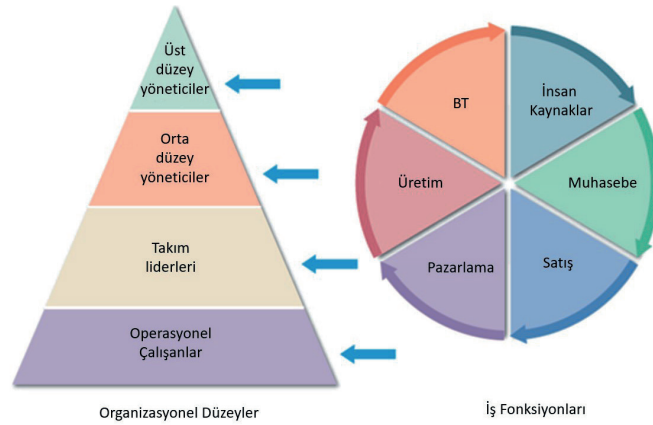
Bilgi sisteminin sahipleri, sistemin kurulduğu ortamın sorumluluğunu üstlenen ve genelde işletmenin yönetim kademesinde bulunan, teknik detaydan ziyade sistemin yönetiminden sorumlu olan paydaştır. Sistemin kuruluşu, işletilmesi ve yönetiminin esas sorumluluğu sistem sahiplerindedir. Sistem sahipleri, bir sistemde var olan diğer rolleri kendi işletmesinde bulunan diğer

kişilere dağıtabildiği gibi, işletme dışından kişi veya kuruluşlara da roller dağıtabilir. Örneğin bir sistemin kullanıcıları işletmenin kendi personeli olup, sistemin kurulumu ve tasarımı işletme dışından da olabilir. Aynı şekilde görevler ve roller işletme içi ve dışı olmak üzere karma bir şekilde de yapılandırılabilir.

1.6. Sistem Kullanıcıları

Bir sistemin amacına uygun bir şekilde görevlerini yerine getirmesi amacıyla sistemi kullanan ve çoğunlukla işletme içindeki son kullanıcıları ifade eder. Sistem kullanıcıları, sistemin teknik

detaylarına odaklanmadan, işlevselliğine, faydasına ve genelde sistemin kullanıcı dostu olması yönü ile ilgilenirler.



Şekil 4. Bir işletmenin hiyerarşik yapısı ve işletme fonksiyonları

Sistem kullanıcıları, bir işletmenin hiyerarşik yapısı ile bağlantılı olarak çeşitli rollerde kullanıcılar içerebilir. Temel olarak bu roller Şekil 4'te gösterilmiştir.

Üst Düzey Yöneticiler: Bir işletmedeki üst düzey yöneticiler, işletmenin misyonuna ve amacına göre stratejik planlamalar yapan ve bu doğrultuda kararlar alan yöneticilerdir. Bu kararlar bir işletmenin gelecekteki büyümesini ve varlığını sürdürmesini etkileyen kararlardır. Bilgi sistemi açısından bakıldığında, üst düzey yöneticilerin verdiği kararlar bilgi sistemine yapılacak yatırımı, bilgi sisteminin yönetsel olarak pozisyonunu belirler. Üst düzey yöneticiler bilgi sistemini stratejik planların ne oranda gerçekleştiğini görmek, işletmenin kâr / zarar durumunu

incelemek, gelecekteki verecekleri kararlara destek olmak ve işletmenin yönünü belirlemek gibi yönetsel amaçlarla kullanırlar. Bu nedenle bilgi sistemini daha genel düzeyde kullandıkları söylenebilir. Üst düzey yöneticiler, bilgi sisteminden elde ettikleri verileri stratejik planları ve kararları için girdi olarak kullanırlar.

Orta Düzey Yöneticiler: Bir işletmede bulunan orta düzey yöneticiler, işletmenin kısa vadeli taktiksel planların ve kararların sorumluluğunu üstlenen yöneticilerdir (Parlaktuna, 2016). Orta düzey yöneticiler, görece olarak daha kısa vadeye odaklandıkları için, üst düzey yöneticilere göre daha ayrıntılı bilgiye ihtiyaç duyarlar. Örneğin işletmenin aylık veya haftalık satış raporlarını incelemek orta düzey yöneticilerin ihtiyaç duy-

duğu bilgilerdendir. Orta düzey yöneticiler bu bilgilere dayanarak kısa vadeli taktiksel eylemleri gerçekleştirirler.

Takım Liderleri: Takım liderleri bir işletmede hiyerarşik olarak orta düzey yöneticiler ile operasyonel çalışanlar arasında yer alırlar. Üst ve orta düzey yöneticilere göre daha teknik personel olarak görev yaparlar. Bu nedenle bilgi sistemini aktif olarak kullanırlar. Orta düzey yöneticilerin taktiksel planlarına göre operasyonel çalışanları koordine eder, gerekli araç/gereç ve ekipmanların varlığından ve doğru bir şekilde kullanımından sorumludurlar. Bu sorumluluklarını yerine getirmek ve eyleme dönüştürmek için bilgi sistemini kullanırlar.

Operasyonel Çalışanlar: Operasyonel çalışanlar bir işletmedeki iş ve işlemlerin yapımı ile doğrudan ilgilenen birimlerdir. İşletmedeki görevlerini yerine getirmek için bilgi sistemini kul-

lanırlar. Bilgi sistemine veri girişini sağlar ve bilgi sistemindeki sürecin yürütülmesini sağlarlar.

BT Departmanı: Bilişim Teknolojileri bölümü bir işletmedeki BT ekipmanlarını efektif olarak kullanan ve yetenek gerektiren özelliklere sahip personelin bulunduğu birimdir. BT departmanı Şekil 4'teki hiyerarşik yapıda açıkça belirtilmemesine rağmen görev ve sorumlulukları itibarıyla önemli bir konumda yer aldıkları için ayrıca ele alınmıştır. BT departmanı kendi içerisinde ağ altyapısı, veritabanı, yazılım geliştirme, güvenlik, teknik destek gibi birimlerden oluşabilir. Bir işletmedeki bilgi sisteminin sağlıklı çalışması, BT departmanının performansı ile doğrudan ilişkilidir.

Bu başlıkta yer alan sistem kullanıcıları haricinde, işletme ile dolaylı olarak ilgili olan, bir işletmenin tedarikçileri ve müşterileri gibi bilgi sistemi kullanıcıları da var olabilir.

2. SİSTEM TASARIMCILARI

Bir işletmenin ihtiyaç duyduğu bilgi sisteminin tasarımını sağlayan paydaşlardır. Bir bilgi sistemi birden fazla uzmanlık gerektirdiğinden, sistem tasarımcıları da çeşitli uzmanlıklardan oluşmaktadır. Genelde bir bilgi sisteminin tasarımında aşağıdaki uzmanlıklara ihtiyaç duyulur:

Ağ altyapısı tasarımcıları: Bir bilgi sistemi çok büyük olasılıkla birden çok teknolojik donanımın bir araya getirilmesinden oluşur. Bu nedenle bilgi sisteminde bu bileşenlerin iletişimini sağlamak ya da internet bağlantısını gerçekleştirmek için bilgisayar ağlarına ihtiyaç duyulur. Ağ altyapısı tasarımcıları, işletmenin ihtiyacına ve bilgi sisteminin amacına uygun olarak bilgisayar ağlarını tasarlar.

Veritabanı tasarımcıları: Bilgi sistemindeki verilerin depolanması ve gerektiğinde sorgulanması gerekir. Verilerin bu amaç için depolanması bir veritabanının varlığını gerektirir. Veritabanı tasarımcıları, bilgi sistemine en uygun ve en verimli veritabanı tasarımını gerçekleştirirler.

Yazılım geliştiriciler: Bir bilgi sisteminin en temel bileşenlerinden biri yazılımdır. Yazılım geliştiriciler, yine işletmenin ihtiyaç duyduğu yazılımın (web ve/veya masaüstü) geliştirilmesini sağlarlar. Bir bilgi sisteminin geliştirilme aşamaları ilerleyen bölümlerde detaylı olarak ele alınacaktır.

Grafik tasarımcılar: Bilgi sisteminin kullanıcıları, sistemi kullanabilmeleri için arayüze ihtiyaç duyarlar. Bu arayüzün görsel tasarımını kullanıcı dostu ve işlevsel olarak yapmaktan sorumludurlar.

Güvenlik uzmanları: İşletmeler birçok nedenden dolayı bilgi sistemlerinin güvenliğini sağlamak zorundadırlar. Bu zorunluluk yasal bir zorunluluk olabildiği gibi, işletmenin ticari sırları, bilgi birikimi, müşterilerin veya çalışanları bilgileri gibi kritik öneme sahip veriler de olabilir. Güvenlik uzmanları, bilgi sisteminin güvenliğinin tasarımından sorumludurlar.

2.1. Sistem Kurucuları

Bir bilgi sisteminin tasarımının ardından, sistemin işletmeye kurulumu gerçekleştirilmelidir. Sistem kurucuları, bilgi sisteminin kurulumunu gerçekleştiren teknik personelden oluşan paydaşlardır. Bu teknik personel, genellikle sistem tasarımcılarından oluşur. Ancak işletme içi BT personeli ya da harici kaynaklar da bu amaç doğrultusunda kullanılabilir. Her iki durumda da teknik personel, bilgi sisteminin tasarımında olduğu gibi birden çok alt gruptan oluşur. Sistem

kurucuları, yazılım uzmanları, ağ altyapısı uzmanları, sunucu uzmanları, veritabanı uzmanları ve güvenlik uzmanlarından oluşur. Ayrıca, bilgi sistemi var olan bir sisteme entegre edilecekse, entegrasyon uzmanlarından oluşabilir. Günümüzde bulut tabanlı uygulamaların giderek arttığı göz önüne alınırsa, bulut bilişim entegrasyonu da başlı başına bir uzmanlık alanı olarak sistem kurucuları arasında yer alır.

2.2. Sistem Analistleri

Bir bilgi sisteminin tasarlanmasındaki en kritik öneme sahip olan ve tamamlayıcı özellikteki paydaş sistem analistleridir. Sistem analistleri işletmenin organizasyonel yapısını ve işleyişini iyi analiz eden ve sistemin diğer bütün paydaşları ile koordineli bir şekilde çalışan uzmanlardır (Den-

nis et al., 2019). Analist, işletmenin ihtiyaçlarını bilen ve bu ihtiyaçlara çözüm geliştiren uzmandır. Sistem analisti bu yönüyle hem teknik hem de yönetsel becerilere sahip bir paydaştır. Kritik önemi nedeniyle sistem analisti başlığında detaylı olarak ele alınacaktır.

2.3. Sistem Analisti

Sistem geliştirme programlarıyla ilgili en önemli sorunlardan biri, bir sistemi, ürünü veya hizmeti gizli kusurlar olmadan planlı, bütçe dahilinde sunabilmek ve kâr elde edebilmektir (Wasson, 2006). Böylesi bir sistemi geliştirmek, ihtiyaç duyulan sistemin kapsamlı ve planlı bir şekilde analiz edilmesini ve sonrasında tasarımını gerektirir. Sistem analizi ve tasarımı, belirli bir işletmedeki insanların veri girişini veya veri akışını sistematik olarak analiz edebilmesi, verileri işleyebilmesi veya dönüştürebilmesi, verileri depolaması ve

çıktı bilgilerini kullanabilmesi için neye ihtiyaç duyduğunu anlamak amacıyla kullanılır (Kendall & Kendall, 2019). Bu detaylı analiz, sistem analistleri tarafından yapılandırılmış programlar çerçevesinde gerçekleştirilir. Sistem analistleri, işletmenin sisteminin kapsamlı bir analizini yaparak doğru sorunları belirlemeye ve çözmeye çalışır. Bu analiz, gerçekleştirilecek olan bilgi sisteminin tasarımında ve gerçekleştirilmesinde kullanılan bir kaynak programdır.

2.4. Sistem Analistinin Rolü

Sistem analistinin rolü analiz sonucunda, geliştirilecek olan bilgi sistemine odaklanmıştır. Analist, BT'nin iş süreçlerini destekleyip iyileştirebileceği yöntemler için fikirler ve öneriler geliştirir. BT tarafından desteklenen yeni iş süreçlerinin tasarlanmasına yardımcı olur, yeni bilgi sistemini tasarlar ve tüm bilgi sisteminin standartlarının korunmasını sağlar.

Sistem analisti, analiz ve tasarım ve programlama konusunda önemli eğitim ve deneyime sahip olmalıdır. Bu özellikler ve analistin kritik öneme sahip olması, bir sistem analistinin birden çok role sahip olması gerektiği anlamına gelir. Sistem analisti aşağıda belirtilen bu rolleri dengeli bir şekilde yürütmelidir.

İş analisti: Bilgi sistemin yaratacağı iş değerini belirlemeye yardımcı olur, iş süreçlerini iyileştirmek için fikirler geliştirir ve yeni iş süreçleri ve politikaları tasarlamaya yardımcı olur (Dennis et al., 2019).

İhtiyaç analisti: Bu rol, işletmenin ve diğer paydaşların gereksinimleri ortaya çıkarmaya odaklanır. Bu gereksinimler işletmenin ihtiyaç duyduğu iş süreçleri olabildiği gibi, sistem kurucularının ve diğer paydaşların ihtiyaç duyacakları gereksinimleri de kapsar. İhtiyaç analisti rolü itibarıyla işletmenin iş sürecini iyi anlaması ve gereksinim belirleme tekniğinde oldukça yetenekli olması beklenir.

Altyapı analisti: Bu rolde sistem analisti, bilgi sisteminin teknik yönlerine odaklanır. Bu yönler, ağ altyapısı, donanım, yazılım, veritabanı ve güvenlik ihtiyaçları gibi teknolojik ihtiyaçlar ile ilişkilidir. Sistem analisti, bilgi sisteminin standartlara uygun olarak çalışabilirliğini sağlamak için

ihtiyaç duyulacak altyapı değişikliklerinin belirlenmesine yardımcı olur. Zamanla, deneyimli bir altyapı analisti, kuruluşun tüm BT ortamına bütünsel bir bakış açısı getiren ve bu bağlamda uygulama tasarım kararlarına rehberlik eden yazılım mimarı rolünü üstlenebilir (Dennis et al., 2019).

Değişim yönetimi: Her yeni sistem beraberinde bir değişim ve yenilik getirir. Bu rol itibarıyla bir sistem analisti, yeni sistem hakkında sistem kullanıcılarının destekler ve kullanıcıların eğitim sürecini yönetir. Ayrıca değişime karşı oluşan doğal dirence çözüm yöntemleri geliştirir. Bu nedenle organizasyonel davranış konusunda uzmanlığa ve deneyime sahip olması beklenir.

Proje yöneticisi: Bilgi sisteminin kurulması bir proje çerçevesinde gerçekleşir. Proje yöneticisi rolü, bilgi sistemi projesinin belirlenen zaman ve bütçe sınırları içerisinde projenin tamamlanmasını kapsar.

2.5. Sistem Analistinin Yetenekleri

Önceki başlıkta ele alınan sistem analistinin çeşitli rolleri, bir sistem analistinin geniş yelpazede yetenekler sergilemesi gerektiğini göstermektedir. Bir değişimi tasarladığından ve işletmenin problemini çözmeyi amaçladığından sistem analisti, bir problem çözücüdür (Kendall & Kendall, 2019). Ancak başarılı bir organizasyonel değişim başlatmak kolay bir süreç değildir. Gerektiğinde, araçların, tekniklerin ve deneyimin ustaca uygulanması yoluyla eldeki durumu sistematik olarak ele alabilmelidir (Wasson, 2006). Bu nedenle analistin birçok yeteneğe sahip olması beklenir. Bu yetenekler, teknik, iş süreçleri, analitik, kişilerarası iletişim, yönetim ve etik olmak üzere altı büyük kategoride toplanır (Dennis et al., 2019).

Teknik yetenekler: Sistem analisti, işletmenin var olan teknik ortamını, yeni sistemin teknoloji temelini ve bu ikisinin bütünleşmiş bir teknik çözüme nasıl uyabileceğini anlamak için teknik becerilere sahip olmalıdır. Bu teknik ye-

tenekler büyük çoğunlukla bilişim teknolojileri ile ilgilidir ve yazılım, donanım, veritabanı, ağ ve güvenlik gibi konuları kapsamaktadır. Özellikle yazılım geliştirme yeteneği kritik öneme sahip bir yetenektir. Çoğu sistem analistinin bir veya daha fazla üst düzey programlama dilini etkili kullanma yeteneğine sahip olması beklenir (Whitten & Bentley, 2007).

İş süreçleri yetenekleri: BT'nin işletme ortamına nasıl uygulanabileceğini anlamak ve BT'nin gerçek iş değeri sunmasını sağlamak için iş becerileri gereklidir. İş becerileri, işletmenin yönetiminde esas olan, muhasebe, finansman, pazarlama, üretim ve yönetim gibi konuları kapsamaktadır.

Analitik yetenekler: Analistler hem proje hem de işletme düzeyinde sürekli olarak problem çözerler. Bu durum analitik becerilerini düzenli olarak test etmelerini sağlar. Sistem analisti, bir işletmenin problem durumunu ele alır,

problemin olası nedenlerini ve etkilerini belirler. Çözüm geliştirmek amacıyla problemi parçalara ayırıp analiz eder ve çözüm önerisini analiz sonucunda geliştirir.

Kişilerarası iletişim yetenekleri: Analist diğer tüm sistem paydaşları ile iletişim halinde olması gereken tamamlayıcı bir paydaştır. Bu nedenle, işletme içerisindeki yöneticiler ve kullanıcılarla olduğu gibi, işletme dışından kişilerle, organizasyonlarla ve alan uzmanları ile de iletişim kurmak zorundadır. Bu iletişimler hem teknik hem organizasyonel boyutta olabilir. Ayrıca iletişim teknik bilgi ve beceri gerektirdiği gibi, son kullanıcılarla olduğu gibi düşük teknik düzeyde de gerçekleştirilir. Büyük ve küçük gruplara sunum yapabilmesi ve rapor yazabilmelidir.

Yönetim yetenekleri: Analistin yalnızca güçlü kişiler arası becerilere sahip olmaları değil, aynı

zamanda birlikte çalıştıkları insanları yönetmeleri ve belirsiz durumlarla ilişkili olan baskıları ve riskleri yönetebilmeleri gerekir.

Etik: Analistler diğer proje ekibi üyeleri, yöneticiler ve sistem kullanıcıları ile adil, dürüst ve etik bir şekilde ilgilenmelidir. Analistler genellikle çalışanlar arasında fikir ayrılığı gibi gizli bilgilere veya başkalarıyla paylaşıldığında zarara yol açabilecek bilgilere sahip olabilir. Analistlerin tüm insanlara karşı güveni sürdürmesi önemlidir. Ayrıca, tasarlanan bilgi sistemi bir işletmenin mahrem bilgilerini ya da bilgi birikimini içerebilir. Analist görevi itibarıyla bu bilgilere erişimi olduğundan, işletmeyi, işletme çalışanlarını ve diğer paydaşları zor durumda bırakmayacak şekilde ve adil davranması gerekir.

3. İŞLETME BİLGİ SİSTEMLERİ

Bilgi sistemleri işletmelerin ihtiyaçlarını çözmeyi amaçlayan sistemlerdir. Ancak dünyada yaşanan değişimler, teknolojik gelişmeler ve müşteri taleplerindeki değişimler, bilgi sistemlerinin değişmesini ve çeşitlenmesini sağlamıştır. Günümüz iş dünyası hızla artan küreselleşme, dizüstü bilgisayarlar ve akıllı telefonlar gibi çok çeşitli cihazlar ile sorunsuz bilgi erişimi sunan teknoloji entegrasyonu ve bulut tabanlı bilgi işlem ile yazılım hizmetlerinin hızlı büyümesi gibi trendler tarafından şekillenmektedir. Özellikle internetin etkisi işletme bilgi sistemlerinde temelde B2C (Business-to-Consumer), B2B (Business-to-Business) ve C2C (Consumer-to-Consumer) gibi kavramların şekillenmesine neden olmuştur.

B2C: Doğrudan tüketiciye veya işletmeden tüketiciye, ürünleri doğrudan müşterilere satmanın ve böylece herhangi bir üçüncü taraf perakendeciyi, toptancıyı veya diğer araçları atlayan iş modelidir.

B2B: İşletmelerin kendi aralarındaki pazarlama ve satış uygulamalarını ifade eden bir iş modelidir.

C2C: Tüketiciden tüketiciye ticaret uygulamalarını ifade eden bir iş modelidir. Tüketiciler genelde bir çevrimiçi bilgi sisteminde bir araya gelir ve ticari işlemlerini doğrudan gerçekleştirirler.

4. KURUMSAL BİLGİ SİSTEMLERİ

Kurumsal bilgi sistemleri, bir işletmenin işini yürütmesine ve belirli hedeflere ulaşmak için verilerini yönetmesini sağlayan bir yazılım sistemidir (Almutairi ve ark., 2022). Kurumsal bilgi sistemlerinin esas amacı, bir işletmenin üretim, satış, hizmet, envanter takibi ve muhasebe gibi

temel işlevlerinin entegrasyonunu sağlayarak verimliliği arttırmak, maliyetleri azaltmak ve yöneticilerin karar vermesine yardımcı olmaktır. Kurumsal bilgi sistemleri, veritabanı yönetimi ve ilişki yönetimi gibi çeşitli kurumsal yazılım türlerini kapsar. Envanter kontrol sistemleri, üretim

kontrol sistemleri ve otel rezervasyon sistemleri kurumsal bilgi sistemlerine örnek olarak verilebilir.

Birçok büyük işletmede Kurumsal Kaynak

Planlama (ERP- Enterprise Resource Planning) sistemi adı verilen uygulamalar, işletme genelindeki kullanıcılar ve yöneticiler için uygun maliyetli destek sağlar.

5. HAREKET İŞLEME SİSTEMLERİ

Hareket işleme sistemleri, bir işletmenin doğal fonksiyonlarını yerine getirirken üretilen verilerin işlenmesini ele alan sistemlerdir. İşletmenin müşterileri ile yaptığı etkileşimler (sipariş alma, ödeme sistemi, vb.) verilerin üretilmesini gerektirir. Bu sistemleri müşterinin mevcut durumunu, müşterinin kredi durumunu ve ürünlerin stok durumunu kontrol eder. Ayrıca muhasebeye ilişkin alacak hesaplarını görüntüleme ve yönetme imkânı sunar. Bu verilerin işletmenin temel fonksiyonları ile ilişkili kritik öneme sahip

veriler olduğundan, büyük miktarda veriler olması beklenir.

Hareket işleme sistemleri, başta satış olmak üzere işletmedeki bir dizi hareketten oluşur. Bu fonksiyonlar genellikle müşteri verilerini doğrulama, müşteri kredi durumunu doğrulama, ürün stok durumunu doğrulama, envanter kontrolü ve satış durumunun güncellenmesi ve raporlanması gibi işlevleri içerir. İşlevlerden herhangi birinde sorun yaşanırsa sistem hareket işlemlerini yerine getirmeyecektir.

6. İŞ DESTEK SİSTEMLERİ

İş destek sistemleri, bir işletmenin her düzeyinde yer alan kullanıcılarına yaptıkları işle ilgili bilgi desteği sunar. Bu sistemler, işlem verilerini analiz edebilir, iş süreçlerini yönetmek ve kontrol etmek için gereken bilgileri üretebilir ve daha iyi karar vermek için yeterli bilgileri sağlayabilir. Bir işletmede operasyonel çalışanlardan, üst düzey yöneticilerine kadar her düzeydeki kullanıcıların, görevlerini yerine getirebilmeleri ve destek almak için bilgiye ihtiyaçları vardır. İş Destek Sistemleri bu ihtiyaçları ele alır. İş Destek Sistemleri diğer sistemler ile entegre bir şekilde çalışır. Örneğin bir müşteriye gerçekleştirilen satış sonrasında, hareket işleme sistemleri satış işlemi kayıt altına alır, müşteri bakiyesini günceller ve stokta gerekli düzenlemeyi yapar. İş Destek Sistemleri hareket sisteminin verilerine göre hangi adımların yapılması gerektiğini, iş sürecinde gecikme durumlarını, üretim için ilgili ihtiyaç sürecine ilişkin tüm kullanıcılara destek sunar.

Bilgi sistemlerinde tutarlılık son derece hassas öneme sahip olabilir. Bu nedenle bir sistemin doğruluğunun geri bildirimler ile kontrol edil-

mesi gerekir. Buna göre genel bir ifade ile bir sistem Şekil2'de gösterilen özelliklerden oluşmaktadır.

- **Girdiler ve Çıktılar :** Bir sistemin esas amacı, kullanıcısı için anlamlı çıktılar üretmektir. Girdiler, işlenmek amacıyla sisteme verilen bilgiyi ifade eder. Çıktılar ise işlem sonucunda beklenen anlamlı sonuçlardır.

- **Süreç :** Girdinin çıktıya dönüşmesini sağlayan fonksiyonları ve iş adımlarını ifade eder.

- **Kontrol :** Sistemin yönlendirilmesini ve yönetilmesini sağlayan özelliktir. Sistemin girdisini, girdinin işlenmesini ve sonuç olarak çıktıya dönüştürülmesini kontrol eder.

- **Geri Bildirim :** Dinamik bir sistemin amacına uygun olarak çalıştığının denetimini sağlar. Olumlu geri bildirimler sistemin çalışmasını teşvik ederken, olumsuz bildirimler süreçte iyileştirmeler ve güncellemeler yapılması gerektiğini ifade eder. Her iki durumda da geri bildirimler sistemin doğal olarak girdilerini oluşturur.

• **Ortam:** Bir sistemin içinde çalıştığı çerçeveyi ifade eder. Bu çerçeve, sistemin çalıştığı sektör, iş alanı ya da kurumsal yapısı ile ilişkilidir. Bu nedenle sistemin nasıl çalışacağını belirleyen dış faktörlerle de ilişkilidir. Örneğin bir işletmenin faaliyet alanı, rakipleri, tedarikçileri gibi etmenler, sistemin ne oranda ve nasıl çalışması gerektiğini değiştirebilir. Tüm bu faktörler bir sistemin

performansına etki eden ortamını oluşturur.

• **Sınırlar ve Arayüzler :** Her sistemin kendi etki ve kontrol alanını belirleyen sınırları bulunur. Bunlar, başka bir sistemle arayüz oluşturduğunda bileşenlerini, süreçlerini ve aralarındaki ilişkiyi tanımlayan sınırlardır.

7. SİSTEM TÜRLERİ

Sistem kavramının farklı alanlarda farklı tanımlamalar ile gösterildiği daha önce ifade edilmişti. Bunun en önemli nedenlerinden biri, farklı türde sistemlerin varlığıdır. Bu başlıkta farklı şekillerde ayrılabilen sistem türlerine değinilecektir.

• **Soyut / Somut Sistemler:** Somut sistemler, fiziksel olarak hissedilebilen sistemler iken soyut sistemler fiziksel olmayan sistemlerdir. Örneğin bir bilgisayar sistemi somut sistemdir, oysa işletim sistemi bir soyut sistem örneğidir.

• **Açık / Kapalı Sistemler :** Açık sistemler çevresi ile etkileşim halinde olan, çevreden girdi alan ve çevreye çıktı veren sistemlerdir. İklimlendirme sistemleri açık sistemlere örnek olarak verilebilir. Kapalı sistemler çevresi ile etkileşime girmeyen, izole sistemlerdir. Gerçekte bu tür tamamen kapalı bir sistemin varlığı nadiren görülür.

• **Kalıcı / Geçici Sistemler:** Bir sistem uzun bir süre boyunca devam ediyorsa kalıcı sistem; belirli bir amaç için geçici süreliğine varlığını sürdürüp görevi bittikten sonra kaldırılıyorsa geçici sistem olarak adlandırılır.

• **Uyarlanabilen / Uyarlanamayan Sistemler:**

Uyarlanabilen sistemler, performanslarını artıracak ve varlığını sürdürebilecek şekilde çevresindeki değişimlere tepki verirler. Biyolojik sistemler uyarlanabilen sistemlerdir. Çevresindeki değişime tepki vermeyen dolayısıyla gelişim gösteremeyen istemler ise uyarlanamayan sistemler olarak ele alınır.

• **Doğal / Yapay Sistemler:** Diğer bir sistem sınıflandırma türü de sistemin doğal ya da yapay sistemler olarak yer alması ile ilgilidir. Doğada var olan ve insan tarafından yapılmayan sistemler doğal sistemler iken, insanlar tarafından yapılan sistemler yapay sistemler olarak adlandırılır.

Bu sistem türlerine ait sınıflandırmalar kesin çizgiler ile ayrılmış türler değildir. Bir sistem birden fazla kategoride yer alabilir. Örneğin bir sistem hem somut, hem açık, hem kalıcı hem de yapay bir sistem olabilir. Ayrıca bu bölümde yer alan sınıflandırma dışında literatürde çeşitli sınıflandırma türlerini de görmek mümkündür.

8. BİLGİ SİSTEMİN BİLEŞENLERİ

Günümüzde verinin yer aldığı hemen her alanda, teknolojik araçlarla desteklenen, veriden kullanıcılar için anlamlı bilgilerin çıkarılmasını sağlayan bilgi sistemleri görülebilir. Ayrıca bilgisayar destekli sistemlerin de sıklıkla kullanıldığını söylemek mümkündür. Bu nedenle günümüzde, bir bilgi sistemine atıfta bulunmak üzere bilgisa-

yar uygulaması terimi kullanılmaktadır. Bir bilgisayar uygulaması, belirli bir işlevi veya ilgili bir dizi işlevi yerine getirmek için bir bilgi işlem cihazında yürütülen bilgisayar yazılım programıdır (Satzinger et al., 2016). Çoğu zaman bilgisayar uygulaması ile bilgi sistemi kavramları birbirinin yerine kullanılsa da, bilgi sisteminin insanları ve

insanların yaptığı manuel prosedürlerini içerdiği; oysa bir bilgisayar uygulamasının genellikle sadece yazılıma atıfta bulunduğu unutulmamalıdır.

Şekil 3’de gösterildiği gibi bir bilgi sistemi; insan, veri, verinin işlenmesini sağlayan teknolojik donanımlar ve yazılımlardan oluşur (Tilley, 2019).

Donanım : Donanım, bir bilgi sistemindeki tüm fiziksel yapıyı ifade eder. Sunucular, iş istasyonları, ağ altyapısı, sabit disk, tarayıcı, kamera gibi birçok teknoloji destekli fiziksel bileşenler bir bilgi sisteminin donanım bileşenini oluştururlar. Ayrıca, işlemci gibi hesaplama yeteneği olan donanımlar da bir bilgi sisteminin önemli bileşenlerindendir. Veri depolama, veri işleme, insan etkileşimi gibi bir bilgi sistemindeki görevler, bu donanımların performansına doğrudan bağlıdır. Bu nedenle bir bilgi sisteminin tasarımında, beklenen performansı gösterebilmesi için fiziksel bileşenlerin doğru seçimi önemlidir.

Yazılım : Yazılım, bir bilgi sistemindeki donanımı kontrol eden ve istenen bilgi veya sonuçları üreten programları ifade eder. Genel olarak sistem yazılımları ve uygulama yazılımları olarak iki kategoride incelenir. İşletim sistemleri ve güvenlik sistemleri gibi yazılımlar sistem yazılımlarına örnektir. Sistem yazılımları donanım ile etkileşimin yanı sıra, verinin akışını kontrol eder, veri güvenliğini sağlar ve ağ işlemlerinin gerçekleştirilmesini sağlar. Uygulama yazılımları, günlük iş fonksiyonlarını destekleyen ve kullanıcılara ihtiyaç duydukları bilgileri sağlayan programlardan oluşur. Kelime işlemci gibi yazılımlar, mobil uygulamalar ve internet tarayıcıları uygulama yazılımlarına örnek olarak verilebilir. Kimi uygulama yazılımları genel kullanıma uygun iken, kimi uygulama yazılımları ise belirli bir amaca yöneliktir. Birçok farklı işletme türünde kullanılmak üzere uyarlanabilen envanter veya bordro uygulaması gibi yazılımlar, *yatay sistemler* olarak adlandırılır. Hastane otomasyonu, online banka-

cılık gibi belirli bir sektöre yönelik yazılımlar ise *dikey sistemler* olarak adlandırılır (Tilley, 2019).

Bilgi sisteminin tasarımında, donanımın olduğu gibi yazılımın da önemi oldukça büyüktür.

Veri : Veri, faydalı bilgilere dönüştürülmesi beklenen bir bilgi sisteminin girdisi yani ham maddesidir. Bir bilgi sisteminde bu veriler toplanır, depolanır, işlenir ve kullanıcılar için anlamlı bilgilere dönüştürülür. Bu işlemler, donanım ve yazılımın desteği ile sağlanır. Bir bilgi sisteminde beklenen işlemlerden biri, verilerin güvenli ve güvenilir olarak saklanmasıdır. Verinin güvenli olması, yetkisiz erişime açık olmaması ile ilgili iken, güvenilir olması verinin tutarlı olmasını ifade eder. Bilgi sisteminden beklenen işlemlerden bir diğeri de, verinin işlenebilmesi ve kullanıcı açısından yorumlanabilir anlamlı bilgiye dönüştürülebilmesidir. Bilgiler grafiksel olarak gösterilebildiği gibi, verinin depolandığı yerden (veri tabanları gibi) temel sorgulamalar ile alınabilir. Günümüzde büyük verinin (big data) de varlığı düşünüldüğünde, devasa veriler içerisinde bilginin çıkarılması (veri madenciliği gibi) bir bilgi sisteminden beklenen diğer bir işlem olarak düşünülebilir.

Süreç : Bir bilgi sisteminde süreç bileşeni, belirli bir amaca ulaşmak için, sistemin diğer bileşenlerinin uyumlu olarak görevlerini gerçekleştirmelerini ve bileşenlerin işlevlerini tanımlar. Şekil 2’de yer alan yapıda gösterildiği gibi, girdiden bir çıktının elde edilmesi tek seferlik bir işlem değil, kontrol edilen, geri bildirimlere göre değerlendirilen ve gerektiğinde tekrar güncellenen bir görevler zinciridir. Doğru tanımlanmamış bir sürecin çıktısına güvenilmez, elde edilen bilginin doğruluğu her zaman şüphelidir. Bu nedenle sistemin algoritmasını ifade eden süreç de diğer bileşenler gibi bir bilgi sisteminin önemli diğer bir bileşenidir.

İnsan : Bilgi sisteminin kullanıcısı, faydalancı ya da daha genel bir ifade ile bu sistemle ilgili

olan insanlara bilgi sisteminin paydaşları denir. Paydaşlar, sistemden sorumlu yönetim grubu üyelerini, sistemle etkileşime girecek işletme içindeki ve dışındaki kullanıcıları (son kullanıcılar olarak adlandırılır) ve sistemi geliştiren ve

destekleyen sistem analistlerini, programcıları ve ağ yöneticileri gibi BT personeli üyelerinden oluşur. Bilgi sisteminin paydaşları ayrı bir başlık altında detaylı olarak incelenecektir.

8.1. Bilgi Sisteminin Paydaşları

Her sistemin bir amacı, görevi yani varlığının bir nedeni vardır. İnsan yapımı sistemlerin çoğu, işletmenin misyonlarına ve hedeflerine katkıda bulunmaya ve bunları gerçekleştirmeye yöneliktir. O halde bir bilgi sisteminin sahibi, kullanıcısı, yararlanıcısı ve geliştiricisi gibi insana ilişkin rolleri ve paydaşları bulunur. Genel olarak bilgi

sisteminin paydaşları, sistem sahipleri, sistem kullanıcıları, sistem tasarımcıları, sistem kurucuları ve sistem analisti olmak üzere beş grupta incelenir (Whitten & Bentley, 2007). Her bir paydaşın sistemdeki rolü ve beklentileri de farklılık göstermektedir. Bu bölümde bilgi sisteminin paydaşları ele alınacaktır.

8.2. Sistem Sahipleri

Bilgi sisteminin sahipleri, sistemin kurulduğu ortamın sorumluluğunu üstlenen ve genelde işletmenin yönetim kademesinde bulunan, teknik detaydan ziyade sistemin yönetiminden sorumlu olan paydaştır. Sistemin kuruluşu, işletilmesi ve yönetiminin esas sorumluluğu sistem sahiplerindedir. Sistem sahipleri, bir sistemde var olan diğer rolleri kendi işletmesinde bulunan diğer

kişilere dağıtabildiği gibi, işletme dışından kişi veya kuruluşlara da roller dağıtabilir. Örneğin bir sistemin kullanıcıları işletmenin kendi personeli olup, sistemin kurulumu ve tasarımı işletme dışından da olabilir. Aynı şekilde görevler ve roller işletme içi ve dışı olmak üzere karma bir şekilde de yapılandırılabilir.

8.3. Sistem Kullanıcıları

Bir sistemin amacına uygun bir şekilde görevlerini yerine getirmesi amacıyla sistemi kullanan ve çoğunlukla işletme içindeki son kullanıcıları ifade eder. Sistem kullanıcıları, sistemin teknik detaylarına odaklanmadan, işlevselliğine, faydasına ve genelde sistemin kullanıcı dostu olması yönü ile ilgilenirler.

Sistem kullanıcıları, bir işletmenin hiyerarşik yapısı ile bağlantılı olarak çeşitli rollerde kullanıcılar içerebilir. Temel olarak bu roller Şekil 4'te gösterilmiştir.

Üst Düzey Yöneticiler: Bir işletmedeki üst düzey yöneticiler, işletmenin misyonuna ve amacına göre stratejik planlamalar yapan ve bu doğrultuda kararlar alan yöneticilerdir. Bu kararlar bir işletmenin gelecekteki büyümesini ve

varlığını sürdürmesini etkileyen kararlardır. Bilgi sistemi açısından bakıldığında, üst düzey yöneticilerin verdiği kararlar bilgi sistemine yapılacak yatırımı, bilgi sisteminin yönetsel olarak pozisyonunu belirler. Üst düzey yöneticiler bilgi sistemini stratejik planların ne oranda gerçekleştiğini görmek, işletmenin kâr / zarar durumunu incelemek, gelecekteki verecekleri kararlara destek olmak ve işletmenin yönünü belirlemek gibi yönetsel amaçlarla kullanırlar. Bu nedenle bilgi sistemini daha genel düzeyde kullandıkları söylenebilir. Üst düzey yöneticiler, bilgi sisteminden elde ettikleri verileri stratejik planları ve kararları için girdi olarak kullanırlar.

Orta Düzey Yöneticiler: Bir işletmede bulunan orta düzey yöneticiler, işletmenin kısa vadeli

taktiksel planların ve kararların sorumluluğunu üstlenen yöneticilerdir (Parlaktuna, 2016). Orta düzey yöneticiler, görece olarak daha kısa vadeye odaklandıkları için, üst düzey yöneticilere göre daha ayrıntılı bilgiye ihtiyaç duyarlar. Örneğin işletmenin aylık veya haftalık satış raporlarını incelemek orta düzey yöneticilerin ihtiyaç duyduğu bilgilerdendir. Orta düzey yöneticiler bu bilgilere dayanarak kısa vadeli taktiksel eylemleri gerçekleştirirler.

Takım Liderleri: Takım liderleri bir işletmede hiyerarşik olarak orta düzey yöneticiler ile operasyonel çalışanlar arasında yer alırlar. Üst ve orta düzey yöneticilere göre daha teknik personel olarak görev yaparlar. Bu nedenle bilgi sistemini aktif olarak kullanırlar. Orta düzey yöneticilerin taktiksel planlarına göre operasyonel çalışanları koordine eder, gerekli araç/gereç ve ekipmanların varlığından ve doğru bir şekilde kullanımından sorumludurlar. Bu sorumluluklarını yerine getirmek ve eyleme dönüştürmek için bilgi sistemini kullanırlar.

Operasyonel Çalışanlar: Operasyonel çalışanlar bir işletmedeki iş ve işlemlerin yapımı ile

doğrudan ilgilenen birimlerdir. İşletmedeki görevlerini yerine getirmek için bilgi sistemini kullanırlar. Bilgi sistemine veri girişini sağlar ve bilgi sistemindeki sürecin yürütülmesini sağlarlar.

BT Departmanı: Bilişim Teknolojileri bölümü bir işletmedeki BT ekipmanlarını efektif olarak kullanan ve yetenek gerektiren özelliklere sahip personelin bulunduğu birimdir. BT departmanı Şekil 4'teki hiyerarşik yapıda açıkça belirtilmemesine rağmen görev ve sorumlulukları itibarıyla önemli bir konumda yer aldıkları için ayrıca ele alınmıştır. BT departmanı kendi içerisinde ağ altyapısı, veritabanı, yazılım geliştirme, güvenlik, teknik destek gibi birimlerden oluşabilir. Bir işletmedeki bilgi sisteminin sağlıklı çalışması, BT departmanının performansı ile doğrudan ilişkilidir.

Bu başlıkta yer alan sistem kullanıcıları haricinde, işletme ile dolaylı olarak ilgili olan, bir işletmenin tedarikçileri ve müşterileri gibi bilgi sistemi kullanıcıları da var olabilir.

8.4. Sistem Tasarımcıları

Bir işletmenin ihtiyaç duyduğu bilgi sisteminin tasarımını sağlayan paydaşlardır. Bir bilgi sistemi birden fazla uzmanlık gerektirdiğinden, sistem tasarımcıları da çeşitli uzmanlıklardan oluşmaktadır. Genelde bir bilgi sisteminin tasarımında aşağıdaki uzmanlıklara ihtiyaç duyulur:

Ağ altyapısı tasarımcıları: Bir bilgi sistemi çok büyük olasılıkla birden çok teknolojik donanımın bir araya getirilmesinden oluşur. Bu nedenle bilgi sisteminde bu bileşenlerin iletişimini sağlamak ya da internet bağlantısını gerçekleştirmek için bilgisayar ağlarına ihtiyaç duyulur. Ağ altyapısı tasarımcıları, işletmenin ihtiyacına ve bilgi sisteminin amacına uygun olarak bilgisayar ağlarını tasarlar.

Veritabanı tasarımcıları: Bilgi sistemindeki verilerin depolanması ve gerektiğinde sorgulanması gerekir. Verilerin bu amaç için depolanması bir veritabanının varlığını gerektirir. Veritabanı tasarımcıları, bilgi sistemine en uygun ve en verimli veritabanı tasarımını gerçekleştirirler.

Yazılım geliştiriciler: Bir bilgi sisteminin en temel bileşenlerinden biri yazılımdır. Yazılım geliştiriciler, yine işletmenin ihtiyaç duyduğu yazılımın (web ve/veya masaüstü) geliştirilmesini sağlarlar. Bir bilgi sisteminin geliştirilme aşamaları ilerleyen bölümlerde detaylı olarak ele alınacaktır.

Grafik tasarımcılar: Bilgi sisteminin kullanıcıları, sistemi kullanabilmeleri için arayüze

ihtiyaç duyarlar. Bu arayüzün görsel tasarımını kullanıcı dostu ve işlevsel olarak yapmaktan sorumludurlar.

Güvenlik uzmanları: İşletmeler birçok nedenden dolayı bilgi sistemlerinin güvenliğini sağlamak zorundadırlar. Bu zorunluluk yasal bir

zorunluluk olabildiği gibi, işletmenin ticari sırları, bilgi birikimi, müşterilerin veya çalışanları bilgileri gibi kritik öneme sahip veriler de olabilir. Güvenlik uzmanları, bilgi sisteminin güvenliğinin tasarımından sorumludurlar.

8.5. Sistem Kurucuları

Bir bilgi sisteminin tasarımının ardından, sistemin işletmeye kurulumu gerçekleştirilmelidir. Sistem kurucuları, bilgi sisteminin kurulumunu gerçekleştiren teknik personelden oluşan paydaşlardır. Bu teknik personel, genellikle sistem tasarımcılarından oluşur. Ancak işletme içi BT personeli ya da harici kaynaklar da bu amaç doğrultusunda kullanılabilir. Her iki durumda da teknik personel, bilgi sisteminin tasarımında olduğu gibi birden çok alt gruptan oluşur. Sistem

kurucuları, yazılım uzmanları, ağ altyapısı uzmanları, sunucu uzmanları, veritabanı uzmanları ve güvenlik uzmanlarından oluşur. Ayrıca, bilgi sistemi var olan bir sisteme entegre edilecekse, entegrasyon uzmanlarından oluşabilir. Günümüzde bulut tabanlı uygulamaların giderek arttığı göz önüne alınırsa, bulut bilişim entegrasyonu da başlı başına bir uzmanlık alanı olarak sistem kurucuları arasında yer alır.

8.6. Sistem Analistleri

Bir bilgi sisteminin tasarlanmasındaki en kritik öneme sahip olan ve tamamlayıcı özellikteki paydaş sistem analistleridir. Sistem analistleri işletmenin organizasyonel yapısını ve işleyişini iyi analiz eden ve sistemin diğer bütün paydaşları ile koordineli bir şekilde çalışan uzmanlardır (Den-

nis et al., 2019). Analist, işletmenin ihtiyaçlarını bilen ve bu ihtiyaçlara çözüm geliştiren uzmandır. Sistem analisti bu yönüyle hem teknik hem de yönetsel becerilere sahip bir paydaştır. Kritik önemi nedeniyle sistem analisti başlığında detaylı olarak ele alınacaktır.

9. SİSTEM ANALİSTİ

Sistem geliştirme programlarıyla ilgili en önemli sorunlardan biri, bir sistemi, ürünü veya hizmeti gizli kusurlar olmadan planlı, bütçe dahilinde sunabilmek ve kâr elde edebilmektir (Wasson, 2006). Böylesi bir sistemi geliştirmek, ihtiyaç duyulan sistemin kapsamlı ve planlı bir şekilde analiz edilmesini ve sonrasında tasarımını gerektirir. Sistem analizi ve tasarımı, belirli bir işletmedeki insanların veri girişini veya veri akışını sistematik olarak analiz edebilmesi, verileri işleyebilmesi veya dönüştürebilmesi, verileri depolaması ve

çıktı bilgilerini kullanabilmesi için neye ihtiyaç duyduğunu anlamak amacıyla kullanılır (Kendall & Kendall, 2019). Bu detaylı analiz, sistem analistleri tarafından yapılandırılmış programlar çerçevesinde gerçekleştirilir. Sistem analistleri, işletmenin sisteminin kapsamlı bir analizini yaparak doğru sorunları belirlemeye ve çözmeye çalışır. Bu analiz, gerçekleştirilecek olan bilgi sisteminin tasarımında ve gerçekleştirilmesinde kullanılan bir kaynak programdır.

9.1. Sistem Analistinin Rolü

Sistem analistinin rolü analiz sonucunda, geliştirilecek olan bilgi sistemine odaklanmıştır. Analist, BT'nin iş süreçlerini destekleyip iyileştirebileceği yöntemler için fikirler ve öneriler geliştirir. BT tarafından desteklenen yeni iş süreçlerinin tasarlanmasına yardımcı olur, yeni bilgi sistemini tasarlar ve tüm bilgi sisteminin standartlarının korunmasını sağlar.

Sistem analisti, analiz ve tasarım ve programlama konusunda önemli eğitim ve deneyime sahip olmalıdır. Bu özellikler ve analistin kritik öneme sahip olması, bir sistem analistinin birden çok role sahip olması gerektiği anlamına gelir. Sistem analisti aşağıda belirtilen bu rolleri dengeli bir şekilde yürütmelidir.

İş analisti: Bilgi sistemin yaratacağı iş değerini belirlemeye yardımcı olur, iş süreçlerini iyileştirmek için fikirler geliştirir ve yeni iş süreçleri ve politikaları tasarlamaya yardımcı olur (Dennis et al., 2019).

İhtiyaç analisti: Bu rol, işletmenin ve diğer paydaşların gereksinimleri ortaya çıkarmaya odaklanır. Bu gereksinimler işletmenin ihtiyaç duyduğu iş süreçleri olabildiği gibi, sistem kurucularının ve diğer paydaşların ihtiyaç duyacakları gereksinimleri de kapsar. İhtiyaç analisti rolü itibarıyla işletmenin iş sürecini iyi anlaması ve gereksinim belirleme tekniğinde oldukça yetenekli olması beklenir.

Altyapı analisti: Bu rolde sistem analisti, bilgi sisteminin teknik yönlerine odaklanır. Bu yönler, ağ altyapısı, donanım, yazılım, veritabanı ve güvenlik ihtiyaçları gibi teknolojik ihtiyaçlar ile ilişkilidir. Sistem analisti, bilgi sisteminin standartlara uygun olarak çalışabilirliğini sağlamak için ihtiyaç duyulacak altyapı değişikliklerinin belirlenmesine yardımcı olur. Zamanla, deneyimli bir altyapı analisti, kuruluşun tüm BT ortamına bütünsel bir bakış açısı getiren ve bu bağlamda uygulama tasarım kararlarına rehberlik eden yazılım mimarı rolünü üstlenebilir (Dennis et al., 2019).

Değişim yönetimi: Her yeni sistem beraberinde bir değişim ve yenilik getirir. Bu rol itibarıyla bir sistem analisti, yeni sistem hakkında sistem kullanıcılarının destekler ve kullanıcıların eğitim sürecini yönetir. Ayrıca değişime karşı oluşan doğal direnç çözüm yöntemleri geliştirir. Bu nedenle organizasyonel davranış konusunda uzmanlığa ve deneyime sahip olması beklenir.

Proje yöneticisi: Bilgi sisteminin kurulması bir proje çerçevesinde gerçekleşir. Proje yöneticisi rolü, bilgi sistemi projesinin belirlenen zaman ve bütçe sınırları içerisinde projenin tamamlanmasını kapsar.

9.2. Sistem Analistinin Yetenekleri

Önceki başlıkta ele alınan sistem analistinin çeşitli rolleri, bir sistem analistinin geniş yelpazede yetenekler sergilemesi gerektiğini göstermektedir. Bir değişimi tasarladığından ve işletmenin problemini çözmeyi amaçladığından sistem analisti, bir problem çözücüdür (Kendall & Kendall, 2019). Ancak başarılı bir organizasyonel değişim başlatmak kolay bir süreç değildir. Gerektiğinde, araçların, tekniklerin ve deneyimin ustaca uygulanması yoluyla elde edilen durumu sistematik olarak ele alabilmelidir (Wasson, 2006). Bu nedenle analistin birçok yeteneğe sahip olması beklenir. Bu yetenekler, teknik, iş süreçleri, analitik, kişilerarası iletişim, yönetim ve etik olmak üzere altı büyük kategoride toplanır (Dennis et al., 2019).

Teknik yetenekler: Sistem analisti, işletmenin var olan teknik ortamını, yeni sistemin teknoloji temelini ve bu ikisinin bütünleşmiş bir

teknik çözüme nasıl uyabileceğini anlamak için teknik becerilere sahip olmalıdır. Bu teknik yetenekler büyük çoğunlukla bilişim teknolojileri ile ilgilidir ve yazılım, donanım, veritabanı, ağ ve güvenlik gibi konuları kapsamaktadır. Özellikle yazılım geliştirme yeteneği kritik öneme sahip bir yetenektir. Çoğu sistem analistinin bir veya daha fazla üst düzey programlama dilini etkili kullanma yeteneğine sahip olması beklenir (Whitten & Bentley, 2007).

İş süreçleri yetenekleri: BT'nin işletme ortamına nasıl uygulanabileceğini anlamak ve BT'nin gerçek iş değeri sunmasını sağlamak için iş becerileri gereklidir. İş becerileri, işletmenin yönetiminde esas olan, muhasebe, finansman, pazarlama, üretim ve yönetim gibi konuları kapsamaktadır.

Analitik yetenekler: Analistler hem proje hem de işletme düzeyinde sürekli olarak problem çözerler. Bu durum analitik becerilerini düzenli olarak test etmelerini sağlar. Sistem analisti, bir işletmenin problem durumunu ele alır, problemin olası nedenlerini ve etkilerini belirler. Çözüm geliştirmek amacıyla problemi parçalara ayırıp analiz eder ve çözüm önerisini analiz sonucunda geliştirir.

Kişilerarası iletişim yetenekleri: Analist diğer tüm sistem paydaşları ile iletişim halinde olması

gereken tamamlayıcı bir paydaştır. Bu nedenle, işletme içerisindeki yöneticiler ve kullanıcılarla olduğu gibi, işletme dışından kişilerle, organizasyonlarla ve alan uzmanları ile de iletişim kurmak zorundadır. Bu iletişimler hem teknik hem organizasyonel boyutta olabilir. Ayrıca iletişim teknik bilgi ve beceri gerektirdiği gibi, son kullanıcılarla olduğu gibi düşük teknik düzeyde de gerçekleştirilir. Büyük ve küçük gruplara sunum yapabilmesi ve rapor yazabilmelidirler.

Yönetim yetenekleri: Analistin yalnızca güçlü kişiler arası becerilere sahip olmaları değil, aynı zamanda birlikte çalıştıkları insanları yönetmeleri ve belirsiz durumlarla ilişkili olan baskıları ve riskleri yönetebilmeleri gerekir.

Etik: Analistler diğer proje ekibi üyeleri, yöneticiler ve sistem kullanıcıları ile adil, dürüst ve etik bir şekilde ilgilenmelidir. Analistler genellikle çalışanlar arasında fikir ayrılığı gibi gizli bilgilere veya başkalarıyla paylaşıldığında zarara yol açabilecek bilgilere sahip olabilir. Analistlerin tüm insanlara karşı güveni sürdürmesi önemlidir. Ayrıca, tasarlanan bilgi sistemi bir işletmenin mahrem bilgilerini ya da bilgi birikimini içerebilir. Analist görevi itibarıyla bu bilgilere erişimi olduğundan, işletmeyi, işletme çalışanlarını ve diğer paydaşları zor durumda bırakmayacak şekilde ve adil davranması gerekir.

10. İŞLETME BİLGİ SİSTEMLERİ

Bilgi sistemleri işletmelerin ihtiyaçlarını çözmeyi amaçlayan sistemlerdir. Ancak dünyada yaşanan değişimler, teknolojik gelişmeler ve müşteri taleplerindeki değişimler, bilgi sistemlerinin de değişmesini ve çeşitlenmesini sağlamıştır. Günümüz iş dünyası hızla artan küreselleşme, dizüstü bilgisayarlar ve akıllı telefonlar gibi çok çeşitli cihazlar ile sorunsuz bilgi erişimi sunan teknoloji entegrasyonu ve bulut tabanlı bilgi işlem ile yazılım hizmetlerinin hızlı büyümesi gibi trendler tarafından şekillenmektedir. Özellikle internetin etkisi işletme bilgi sistemlerinde temelde B2C

(Business-to-Consumer), B2B (Business-to-Business) ve C2C (Consumer-to-Consumer) gibi kavramların şekillenmesine neden olmuştur.

B2C: Doğrudan tüketiciye veya işletmeden tüketiciye, ürünleri doğrudan müşterilere satmanın ve böylece herhangi bir üçüncü taraf perakendeciyi, toptancıyı veya diğer aracılara atlayan iş modelidir.

B2B: İşletmelerin kendi aralarındaki pazarlama ve satış uygulamalarını ifade eden bir iş modelidir.

C2C: Tüketiciden tüketiciye ticaret uygulamalarını ifade eden bir iş modelidir. Tüketiciler genelde bir çevrimiçi bilgi sisteminde bir araya

gelir ve ticari işlemlerini doğrudan gerçekleştirirler.

11. KURUMSAL BİLGİ SİSTEMLERİ

Kurumsal bilgi sistemleri, bir işletmenin işini yürütmesine ve belirli hedeflere ulaşmak için verilerini yönetmesini sağlayan bir yazılım sistemidir (Almutairi ve ark., 2022). Kurumsal bilgi sistemlerinin esas amacı, bir işletmenin üretim, satış, hizmet, envanter takibi ve muhasebe gibi temel işlevlerinin entegrasyonunu sağlayarak verimliliği arttırmak, maliyetleri azaltmak ve yöneticilerin karar vermesine yardımcı olmaktır. Kurumsal bilgi sistemleri, veritabanı yönetimi ve

ilişki yönetimi gibi çeşitli kurumsal yazılım türlerini kapsar. Envanter kontrol sistemleri, üretim kontrol sistemleri ve otel rezervasyon sistemleri kurumsal bilgi sistemlerine örnek olarak verilebilir.

Birçok büyük işletmede Kurumsal Kaynak Planlama (ERP- Enterprise Resource Planning) sistemi adı verilen uygulamalar, işletme genelindeki kullanıcılar ve yöneticiler için uygun maliyetli destek sağlar.

12. HAREKET İŞLEME SİSTEMLERİ

Hareket işleme sistemleri, bir işletmenin doğal fonksiyonlarını yerine getirirken üretilen verilerin işlenmesini ele alan sistemlerdir. İşletmenin müşterileri ile yaptığı etkileşimler (sipariş alma, ödeme sistemi, vb.) verilerin üretilmesini gerektirir. Bu sistemleri müşterinin mevcut durumunu, müşterinin kredi durumunu ve ürünlerin stok durumunu kontrol eder. Ayrıca muhasebeye ilişkin alacak hesaplarını görüntüleme ve yönetme imkânı sunar. Bu verilerin işletmenin temel fonksiyonları ile ilişkili kritik öneme sahip

veriler olduğundan, büyük miktarda veriler olması beklenir.

Hareket işleme sistemleri, başta satış olmak üzere işletmenin bir dizi hareketten oluşur. Bu fonksiyonlar genellikle müşteri verilerini doğrulama, müşteri kredi durumunu doğrulama, ürün stok durumunu doğrulama, envanter kontrolü ve satış durumunun güncellenmesi ve raporlanması gibi işlevleri içerir. İşlevlerden herhangi birinde sorun yaşanırsa sistem hareket işlemlerini yerine getirmeyecektir.

13. İŞ DESTEK SİSTEMLERİ

İş destek sistemleri, bir işletmenin her düzeyinde yer alan kullanıcılarına yaptıkları iş ilgili bilgi desteği sunar. Bu sistemler, işlem verilerini analiz edebilir, iş süreçlerini yönetmek ve kontrol etmek için gereken bilgileri üretebilir ve daha iyi karar vermek için yeterli bilgileri sağlayabilir. Bir işletmede operasyonel çalışanlardan, üst düzey yöneticilerine kadar her düzeydeki kullanıcıların, görevlerini yerine getirebilmeleri ve destek almak için bilgiye ihtiyaçları vardır. İş Destek Sistemleri bu ihtiyaçları ele alır. İş Des-

tek Sistemleri diğer sistemler ile entegre bir şekilde çalışır. Örneğin bir müşteriye gerçekleştirilen satış sonrasında, hareket işleme sistemleri satış işlemini kayıt altına alır, müşteri bakiyesini günceller ve stokta gerekli düzenlemeyi yapar. İş Destek Sistemleri hareket sisteminin verilerine göre hangi adımların yapılması gerektiğini, iş sürecinde gecikme durumlarını, üretim için ilgili ihtiyaç sürecine ilişkin tüm kullanıcılara destek sunar.

14. BİLGİ YÖNETİM SİSTEMLERİ

Bir bilgi tabanı ve bilginin nasıl uygulanacağını belirleyen çıkarım kurallarını birleştirerek bir alan uzmanının düşünce tarzını simüle eden sistemlerdir. Bu nedenle bu sistemler uzman sistemler olarak adlandırılır. Bilgi tabanı tüm kullanıcıların sorgulamalar yapabildiği geniş bir veritabanından oluşur. Bu veritabanında yer alan veriler arasındaki ilişkilerden ve desenlerden

mantıksal çıkarımlar yapılır. Örneğin bir işletme, müşterileri için sorunları sorgulayıp olası çözüm önerilerini sunduğu sistemler bilgi yönetim sistemlerine örnek olarak verilebilir. Çoğu bilgi yönetim sistemlerinde, veriler arasındaki belirgin olmayan ilişkilerin çıkarımı için bulanık mantık denilen bir teknik kullanılır. (Shelly & Rosenblatt, 2012)

15. KULLANICI VERİMLİLİK SİSTEMLERİ

İşletmelerde her düzeydeki kullanıcıların verimliliğinin arttırmak amacıyla kullanılan sistemlerdir. Bu sistemler kullanıcıların iş ile ilgili yaptıkları görevlerin daha verimli bir şekilde yerine getirilmesini sağlar. Örneğin iletişim için e-posta sistemleri, hesaplama için muhasebe sistemleri, yazışmalar için kelime işlemci yazılımlarını içeren sistemler ve ofis uygulamaları kullanıcı verimlilik sistemlerine örnek olarak verilebilir.

Veri tabanı sistemleri, elektronik tablolar, masaüstü yayıncılık uygulamaları, grafiksel analiz yazılımları gibi yazılımlardan oluşan sistemler de kullanıcı verimlilik sistemlerine örnek olarak verilebilir. Ayrıca bir işletmenin çevrimiçi olarak kullanıcılarının ortak çalışma yapmalarını, iş birliği yapmalarını sağlayan sistemler de bu kategoride yer alan sistemlerdir.



BÖLÜM ÖZETİ

Sistem, belirli bir amacı gerçekleştirmek için bir araya gelmiş, birbirleriyle ilişkili bileşenler kümesidir. Tüm sistemlerde bir girdi ve bir çıktı bulunur. En yalın hâliyle bir sistem, bir girdiyi alır, işler ve çıktıya dönüştürür.

Sistemler, açık ve kapalı sistemler, soyut ve somut sistemler, kalıcı ve geçici sistemler, uyarlanabilen ve uyarlanamayan sistemler ve doğal ve yapay sistemler olmak üzere çeşitli türlerde bulunabilir.

Bilgi sistemi, belirli bir amaç doğrultusunda bilgiyi toplayan, depolayan, işleyen; insan, veri, süreçler ve bilgi teknolojilerinin etkileşimde bulunduğu bir yapıdır. Genel olarak bir bilgi sistemi, girdiler ve çıktılar, süreç, kontrol, geri bildirim, ortam ve sınırlar ve arayüzlerden oluşur. Bilgi sisteminin temel bileşenleri, veri, donanım, yazılım, süreç ve insandır.

Bilgi sisteminin paydaşları; sistem sahipleri, sistem kullanıcıları, sistem tasarımcıları, sistem kurucuları ve sistem analistleri olmak üzere beş grupta incelenir. Sistem sahipleri, sistemin kurulduğu ortamın sorumluluğunu üstlenen ve genellikle işletmenin yönetim kademesinde bulunan paydaştır. Sistem kullanıcıları, sistemin amacına uygun bir şekilde görevlerini yerine getirmesi amacıyla sistemi kullanan ve çoğunlukla işletme içindeki son kullanıcıları ifade eder. Üst düzey yöneticiler, orta düzey yöneticiler, takım liderleri ve operasyonel çalışanlar sistem kullanıcılarıdır. Sistem tasarımcıları, bilgi sisteminin tasarımını yapan paydaştır. Ağ altyapısı tasarımcıları, veritabanı tasarımcıları, yazılım geliştiriciler, grafik tasarımcılar ve güvenlik uzmanları, sistem tasarımcılarının ihtiyaç duyduğu uzmanlıklardır. Sistem kurucuları, bilgi sisteminin kurulumunu gerçekleştiren teknik personelden oluşan paydaşlardır. Sistem kurucuları, yazılım uzmanları, ağ altyapısı uzmanları, sunucu uzmanları, veritabanı uzmanları ve güvenlik uzmanlarından oluşur. Bir bilgi sisteminin tasarlanmasındaki en kritik öneme sahip olan ve tamamlayıcı özellikteki paydaş ise sistem analistleridir.

Sistem analistleri, işletmenin sisteminin kapsamlı bir analizini yaparak doğru sorunları belirlemeye ve çözmeye çalışır. Sistem analistinin birden çok rolü vardır ve bu rolleri dengeli bir şekilde yürütmelidir. Bu roller; iş analisti, ihtiyaç analisti, altyapı analisti, değişim yöneticisi ve proje yöneticisidir.

Bir sistem analistinin geniş yelpazede yetenekler sergilemesi gerekir. Bu yetenekler; teknik yetenekler, iş süreçleri yetenekleri, analitik yetenekler, kişiler arası iletişim yetenekleri, yönetim yetenekleri ve etik'tir.

Bilgi sistemleri çok farklı türlerde sınıflandırılabilir. Kurumsal bilgi sistemleri, hareket işleme sistemleri, iş destek sistemleri, bilgi yönetim sistemleri ve kullanıcı verimlilik sistemleri işletme bilgi sistemleri türlerindendir.

GÖZDEN GEÇİRELİM

1. Hangisi bir sistemin en basit haliyle gösterimidir?
 - a) Girdi-Süreç-Çıktı
 - b) Donanım – Yazılım – İşletim Sistemi
 - c) Geri bildirim ve kontrol
 - d) İnsan- Veri – Donanım – Yazılım
2. Hangisi bir sistemdeki girdinin çıktıya dönüşmesini sağlayan fonksiyonları ve iş adımlarını ifade eder?
 - a) Süreç
 - b) Kontrol
 - c) Veri
 - d) Girdi/çıkıtı
3. Sistemin yönlendirilmesini ve yönetilmesini sağlayan özellik hangisidir?
 - a) Süreç
 - b) Kontrol
 - c) Veri
 - d) Girdi/çıkıtı
4. sistemler, fiziksel olarak hissedilebilen sistemler iken sistemler fiziksel olmayan sistemlerdir.
Yukarıdaki boşluklara sırasıyla hangi ifade gelmelidir?
 - a) Açık – Kapalı
 - b) Kalıcı – Geçici
 - c) Doğal – Yapay
 - d) Somut – Soyut
5. Hangisi bir bilgi sistemindeki tüm fiziksel yapıyı ifade eder?
 - a) Süreç
 - b) İnsan
 - c) Veri
 - d) Donanım
6. Hangisi sistemin kurulduğu ortamın sorumluluğunu üstlenen ve genelde işletmenin yönetim kademesinde bulunan, teknik detaydan ziyade sistemin yönetiminden sorumlu olan paydaştır?
 - a) Sistem kurucuları
 - b) Sistem sahipleri
 - c) Sistem kullanıcıları
 - d) Sistem analistleri
7. Sistemin teknik detaylarına odaklanmadan, işlevselliğine, faydasına ve genelde sistemin kullanıcı dostu olması yönü ile ilgilenen paydaşlar hangileridir?
 - a) Sistem kurucuları
 - b) Sistem sahipleri
 - c) Sistem kullanıcıları
 - d) Sistem analistleri
8. Bilgi sistemindeki bileşenlerin iletişimini sağlamak ya da internet bağlantısını gerçekleştirmek hangi sistem tasarımcısının gerektirir?
 - a) Veritabanı tasarımcısı
 - b) Ağ altyapısı tasarımcısı
 - c) Yazılım geliştirici
 - d) Güvenlik uzmanı

9. İş süreçlerini iyileştirmek için fikirler geliştirme, yeni iş süreçleri ve politikaları tasarlamaya yardımcı olma hangi analist rolü ile sağlanır?

- a) İş analisti
- b) İhtiyaç analisti
- c) Değişim yöneticisi
- d) Proje yöneticisi

10. Envanter kontrol sistemleri, üretim kontrol sistemleri ve otel rezervasyon sistemleri hangi tür bilgi sistemlerine örnek olarak verilebilir?

- a) Kurumsal bilgi sistemleri
- b) Hareket işleme sistemleri
- c) İş destek sistemleri
- d) Kullanıcı verimlilik sistemleri

Yanıt Anahtarı: 1-B, 2-C, 3-A, 4-D, 5-E

Kaynakça

- Almutairi, A., Naeem, M. A., & Weber, G. (2022). Understanding enterprise systems adaptability: An exploratory survey. *Procedia Computer Science*, 197, 743–750. <https://doi.org/https://doi.org/10.1016/j.procs.2021.12.196>
- Dennis, A., Wixom, B. H., & Roth, R. M. (2019). *System Analysis and Design* (7th ed.). Wiley.
- Kendall, K. E., & Kendall, J. E. (2019). Systems Analysis and Design. In *A Guide to Medical Computing* (10th ed.). Pearson.
- Parlaktuna, O. (2016). Sistem Analizi ve Tasarımına Giriş. In H. Durucasu & Ş. Lezki (Eds.), *Sistem Analizi ve Tasarımı* (Issue 1, pp. 1–23). Anadolu Üniversitesi.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). System Analysis and Design in a Changing World. In *Course Technology Cengage Learning* (7th ed.). Cengage Learning.
- Shelly, G. B., & Rosenblatt, H. J. (2012). System Analysis and Design. In *News.Ge* (10th ed.). Shelly Cashman Series.
- Tilley, S. (2019). Systems Analysis and Design. In *Shelly Cashman Series*.
- Wasson, C. S. (2006). System Analysis, Design, and Development Concepts, Principles, and Practices. In *A John Wiley & Sons, Inc., Publication*.
- Whitten, J., & Bentley, L. (2007). *Systems Analysis and Design Methods* (7th ed.). McGraw-Hill Irwin.

BÖLÜM 2

PROJE YÖNETİMİ

ANAHTAR KAVRAMLAR

Sistem, Sistemin Özellikleri, Sistem Türleri, Bilgi Sistemi, Bilgi Sisteminin Bileşenleri, Bilgi Sisteminin Paydaşları, Sistem Analisti, İşletme Bilgi Sistemleri

ÖĞRENME HEDEFLERİ

- 1 Projeyi bir sistem olarak tanımlar
- 2 Proje yönetiminin ve proje yöneticisinin niteliklerini bilir
- 3 Proje yönetim araçlarını kullanarak bir projenin şebekesini ve çizelgesini hazırlar

İKİNCİ BÖLÜM

GİRİŞ



1. BİR SİSTEM OLARAK PROJE

Bir amacı gerçekleştirmek için var olan ve birbirleri ile etkileşimde bulunan parçalardan oluşan bütünlere sistem adı verilmektedir. (Sarıaslan, 1984) Sistemlerin varlığının temel nedeninin bütünün parçalarından daha fazlası olması olarak gösterebiliriz. Sistem yaklaşımı, insanların çevrede bulunan olgu ve faaliyetleri ilişkilendirmesine yardımcı olmaktadır. Bu ilişkileri hiyerarşik bir yapıda irdelemesini de mümkün kılan sistem yaklaşımı, herhangi bir bütün ve onun bileşenlerini daha anlaşılır kılmaktadır. Sağlanan bu anlaşılabilirlik içerisinde sebep, süreç, sonuç dizilimi ile bütünün ve bileşenlerin işlevleri aydınlatılmaktadır. Sonucunda sistemler, anlaşılması ve iyileştirilmesi mümkün bir bünye halini almaktadır.

Sistemler alt sistemlerden oluşmaktadır. Alt sistemler de çeşitli parçaları ihtiva etmektedir. Sistemin karmaşıklığına dayalı olarak alt sistemlerin sayısı artmaktadır. Bir diğer değişle her sistemin bir başka sistemin alt sistemi olduğunu söyleyebiliriz. Değerlendirildiği noktaya göre hiyerarşik yapının en tepesinde bulunan sisteme süper sistem adı verilmektedir.

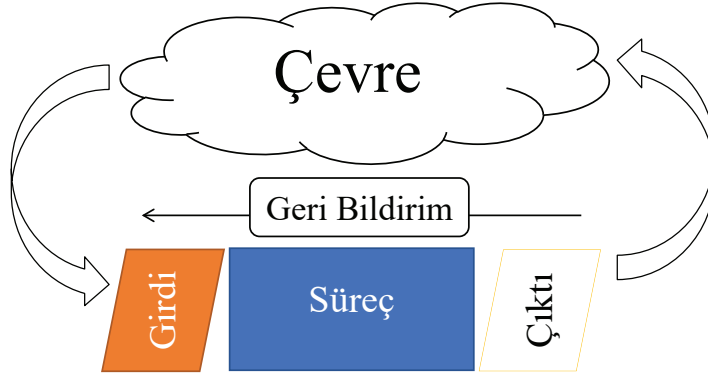
Sistemlerin bir diğer özelliği belli bir çevrede bulunmaları ve bu çevreyle etkileşim içerisinde varlıklarını sürdürmeleridir. Çevresi ile etkileşimde bulunan sistemler açık sistem olarak anılmaktadır. Kapalı sistemler ise çevresiyle etkileşimde bulunmayan sistemlerdir. Kapalı sistemlere örnek olarak kurmalı bir çalar saat verilebilmektedir. Belirli bir süre sonra çalmak üzere kurulan bir çalar saat dışarıdan başka bir girdiye ihtiyaç duymadan amacına ulaşana (çalana) kadar görevini yapabilir. Doğal olarak yeniden amacını gerçekleştirmesi için yaşam döngüsüne yeniden başlaması gerekmektedir. En tepede bulunan süper sistemlerin bile dışarıyla bir alışverişini olduğunu gözlemleyebiliyoruz.

Sistemler, çevreden girdiler alan, bu girdileri bir süreçten geçirerek çıktıları çevreye ve

ren bütünlere dir. Açık sistemlerde bu akışın tersi yönünde işleyen geri besleme süreci de yer almaktadır. Geri beslemenin açık sistemleri kapalı sistemlerden ayıran temel özellik olduğunu söyleyebiliriz. Açık sistemlerde geri besleme sürekli gerçekleşerek sistemin kendi kendini düzeltme ve iç dengesini koruma işlevlerini görür. Geri besleme süreci sistem çıktılarının sonuçlarını değerlendirme, öz düzeltme, özdengeye kavuşma olarak gerçekleşir. Geri besleme süreci ile sistem çevresine uyum sağlayarak varlığını sürdürme başarısını artırır.

Sistemlere farklı örnekler verilebilir. Güneş Sistemi astronomik bir sistemdir. Amacı daha üst sistem olan Samanyolu Galaksisi içerisinde devinim göstererek var olmaktır. Kendisi alt sistemlerden -Güneş ve gezegenler ve çok sayıda asteroitten- oluşmaktadır. İnsan vücudu da bir sistemdir. Bir amacı vardır -yaşamak-, çeşitli parçalardan oluşmuştur -organlar- ve bu organlar birbiriyle elektromekanik ve kimyasal etkileşim içerisinde dir. Bu organlarda olan bir değişiklik diğer organları ve insanı etkiler yani sistemi etkiler. Güneş Sistemi gibi fizik kurallarıyla işleyen sistemlerin yanında insan vücudu gibi otomatik biyolojik işleyişe sahip sistemlerde vardır. Bireysel bir bilinç dayalı sistemlerin yanında ekonomik sistem gibi kolektif bir bilinç dayalı sistemler de bulunmaktadır.

Sistemlerin, bir doğası ve varlık nedeni bulunmaktadır. Bu doğayı ve varlık nedenini dış etkenlere karşı sürdürdüğünü görmekteyiz. Bilinçli sistemlerde ise varlık amacını gerçekleştirmek için fark edilir bir çaba gözlemlenmektedir. Amaç, sistemin varlık nedenini oluşturmaktadır. Bu daha önce anılan sistemlerin iç dengesini koruma, geri bildirim yapma ve kendi kendini düzeltme özelliğinden kaynaklanmaktadır.



Şekil 1 - Sistem yapısı

Sistem, belirli bir amaç çerçevesinde birbiriyle etkileşimde bulunan parçalardan oluşan bir bütündür. Sistemi oluşturan parçalar sistemin varlık nedenine yönelik etkileşimli bir çalışma gösterir. Bu, parçaların bir yığına dönüşmesinin önüne geçen nedendir.

Sistem anlayışına göre bir proje de bir sistemdir. Bir amacı vardır, çeşitli parçalardan oluşmuştur -takımlar, etkinlikler, olaylar- ve bunlar arasında etkileşim yer almaktadır. Takımların

faaliyet sonuçları hem diğerlerini hem de bütün projeyi etkilemektedir. İnsan ihtiyaçlarını verimli bir biçimde karşılamak için oluşmuş ekonomi sisteminin ya da bir işletmenin bir alt sistemidir. Projenin amaçlarını gerçekleştirmek için kullandığı insan kaynağı ile maddi ve maddi olmayan varlıkları da onun girdileridir. Proje kullandığı kaynakları kendi süreçlerinde kullanarak bulunduğu üst sistemlere çıktılar sağlar.

2. PROJE

2.1. Projenin Tanımı

Proje, belirli bir sorunu çözmek üzere özgün olarak oluşturulan sistem şeklinde tanımlanabilir. Projeler benzersiz bir ürün ya da hizmet sağlamak üzere girişilen süreli uğraşlar olarak da ta-

nımlanmaktadır. Projelerin belirli bir zaman ve maliyet kısıtı bulunmaktadır. Bu kısıtlar dahilinde bir hedefe yönelik tasarlanan proje gerçekleştirilmesi ile son bulur.

2.2. Projenin Özellikleri

Projeler odaklı ve özgün çözümler getirerek çeşitli avantajlar doğurmaktadır. Diğer iş yapış biçimlerinden farklılıklar barındıran projelerin özellikleri şu şekilde listelenebilir:

- Projeler bir kapsam, süre ve bütçe kısıdına sahiptir
- Projeler birden fazla işlev ya da örgütün katılımıyla oluşturulan takımlar tarafından gerçekleştirilir
- Projeler özgün, bir kerelik, sıra dışı görevlerden oluşur

- Projeler beşerî, aynı ve mali kaynaklar kullanılarak tamamlanır

- Projeler kendine özel örgütlenme, yönetim ve bilgi işlem tekniklerine sahiptir

Bir projenin başarısı onun kapsam, süre ve maliyet hedeflerine uyumu ile tanımlanmaktadır. Bunların doğru belirlenmesi iyi bir yapılabirlik (fizibilite) çalışması gerektirmektedir. Yapılabirlik çalışmasının doğru bir biçimde yapılması kapsam, süre ve maliyetin doğru belirlenmesini sağlayacaktır. Yapılabirlik çalışması,

projenin finansal kaynak bulabilmesi açısından da önem arz etmektedir.

Projelerde bilgi işlem teknolojileri yoğun bir biçimde kullanılmaktadır. Projelerin kapsam, süre ve maliyet sınırlaması, kritik hatta sahip olması, proje örgütünün takımlardan oluşması bilgi işlem teknolojilerinin projeler açısından gerekliliğini arttırmaktadır.

Projelerin örgütlemesi yatay bir örgüt yapısına sahiptir. Proje örgütleri hiyerarşik yapıda

değil takımlardan oluşan matris biçiminde oluşturulmaktadır. O nedenle proje yönetimi üst yönetimden ziyade merkezi yönetim olarak anılır. Merkezi yönetim projenin aksamadan tamamlanabilmesi takımların ve görevlerin koordinasyonuna odaklanmaktadır. Koordinasyonun rahatlıkla sağlanabilmesi için proje görevlerinin takibi yapılmaktadır.

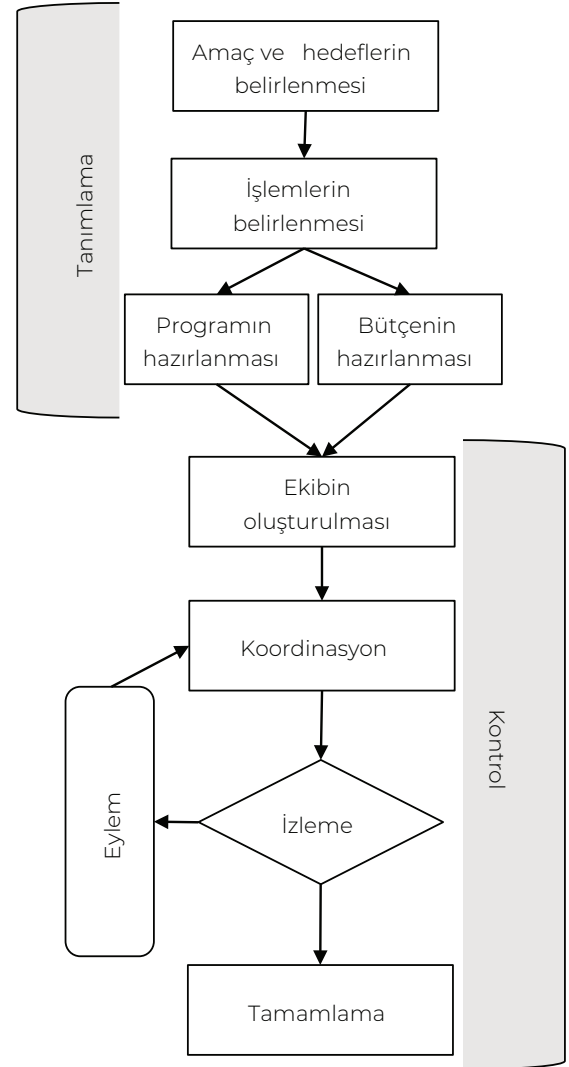
2.3. Proje Çeşitleri

Projelerin amaçlarına göre çeşitlerine ayrılmaktadır (Albayrak, 2005):

- A. İş organizasyonu ve yönetim projeleri.
- B. Nesneye ilişkin projeler.
- C. Kurulum projeleri.

İş organizasyonu ve yönetim projeleri iş yapım projeleri olarak da adlandırılmaktadır. Bu tip projelere örnek olarak bilgisayar destekli yönetim ve bilişim sistemlerinin oluşturulması, üretim planlama ve kurulumu, yeniden örgütlenme faaliyetleri verilebilir. Bu projeleri diğerlerinden ayıran iş yapış biçimini tanımlaması ve genellikle bir örgüt içerisindeki çalışma şeklini oluşturmasıdır. Bir diğer deyişle işletme sistemi içerisindeki bir sistemin oluşumunu kapsar ve genellikle bir örgüt dahilinde, örgüt kaynaklarıyla gerçekleştirilir.

Nesneye ilişkin projeler ve kurulum projeleri ise genellikle birden fazla örgütün dahil olduğu projeler olarak karşımıza çıkmaktadır. Bu projelere ürün odaklı teknik projelerdir. Bunlara örnek olarak araştırma geliştirme faaliyetleri, yeni bir ürünün tasarımı ya da piyasa araştırmaları verilebilir. Bu projelerde proje tasarımı ve gerçekleştirilmesi farklı örgütler tarafından üstlenilebildiği gibi bu tasarım ve gerçekleştirme aşamalarında da birden fazla örgüt devreye girebilmektedir.



Şekil 2 - Projenin Tanımlama ve Kontrol akış şeması (Albayrak, 2005)

Kurulum projeleri en kapsamlı projeler olarak karşımıza çıkmaktadır. Yeni bir işletmenin kurulumu bu tip projelerden olmakla birlikte fabrika ya da bina gibi bir inşaatın yapımından uzay araçlarının yapımına kadar uzanan çeşitleri

barındırmaktadır. Birden fazla seneye yayıldığı görülen bu projelerin hemen her zaman birden fazla örgüt katılımıyla, yüksek uzmanlık ve emek düzeyiyle diğer projelerden ayrıştığını görmekteyiz.

2.4. Proje Yapısı

Projenin yapısını tanımlama ve kontrol aşamaları olarak ikiye ayırabiliriz. Tanımlama ve kontrol projenin başarıyla sonuçlanmasını sağlayacak adımları içeren iki unsurdur. Biri projenin tasarımı, diğeri ise uygulama ve sonuçlanma aşaması olarak projenin başındaki ve sonundaki süreçleri tarif etmektedir. (Albayrak, 2005)

Tanımlama aşamasında projenin amaçları, hedefleri, projenin adımları, programı ve bütçesi ortaya konulur. Amaçlar projeye neden ihtiyaç duyulduğuna verilen cevap olarak tanımlanabilir. Hedefler ise bu amaçlara ulaşıldığının nasıl ölçüleceğini belirlemek için belirlenmelidir. Amaçlar nitel olarak belirlenir, hedeflerin ölçülebilir olması için nicel olarak belirlenmesi gerekmektedir. Projenin adımları hedeflere ulaşmak için gerek duyulan görevlerden oluşmaktadır. Görevler benzer işleri içeren paketler olarak tanımlanabilir. Bir projede bulunan görevler birbiri ile ilişkilidir ve çoğunlukla birbirini takip eder biçimdedir. O nedenle projenin etkin bir biçimde yürütülmesi için bunlar bir program dahilinde gerçekleştirilir. Program görevlerin hangi sırayla gerçekleştirileceğini gösteren ve görevlerde kullanacakları kaynakları da içeren çizelgedir. Projeyi gerçekleştirmek için kullanılacak beşeri ve maddi tüm kaynakların bir bedeli bulunmaktadır. Projede kullanılacak kaynaklar için gerek-

li mali kaynağın planlanması projenin etkin bir biçimde başarılması için bir gerekliliktir. Bütçe, projede kullanılacak mali kaynağın planı olarak tanımlanabilir.

Kontrol proje ekibinin uyum içerisinde çalışması için çabaların koordine edilmesi ve izlenmesini içerir. Bir ya da birden fazla örgütten oluşabilecek proje ekibinin projeyi planlandığı biçimde yürütüp tamamlaması gerekmektedir. Projenin planlandığı biçimde yürütülmesi ve tamamlanması için proje yönetimi ekipleri ve kaynaklar üzerinde gerekli koordinasyonu sağlar. Projenin program ve bütçeye göre ilerlediğinden emin olmak için süreci izler ve sonuçları değerlendirerek gerekirse düzenleyici eylemde bulunur.

Kontrol aşamasında projenin tanımlandığı şekilde gerçekleştirilmesi sağlanır. Proje ekibini oluşturan takımlar gerekli ve yeterli uzmanlardan oluştuğunda ve bu takımlara gerekli yetki göçerimi yapılırsa dahi kontrolden vazgeçilemez. Projenin planlandığı gibi gittiğinden emin olmak ve koordinasyonu sağlamak için kontrole ihtiyaç duyulmaktadır. Kontrol görevlerin ardı ardına tamamlanarak projenin tamamlanması işlevini de üstlenmektedir. Görevlerin tamamlanarak istenilen kapsamda bir sisteme ulaşmasını proje yönetimi kontrol işleviyle elde etmektedir.

2.5. Proje Hedefleri

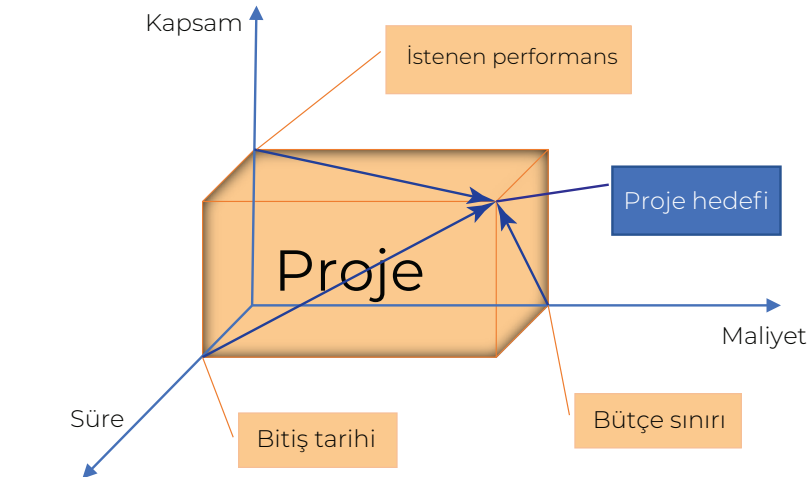
Başarılı bir proje genellikle *zamanında, bütçe dahilinde ve doğru kalitede (nitelikte)* tabirleriyle tanımlanmaktadır. (Pinto & Slavin, 1988) Projelerin her biri kendine özgü olsa da tüm projeler için ortak olan hedefleri *kapsam, süre ve maliyet* ola-

rak listelenmektedir. (Meredith & Samuel, 2000) Bu hedefler her projenin ortak yönünü oluşturmaktadır. Projenin etkinliği bu hedeflere ne ölçüde uyulduğuna bağlıdır. Kapsam her projeden beklenen etkinin bir sonucu, olarak tarif edile-

bilir. Projenin kapsamı sınırlıdır ve bu kapsam dahilinde yapılan işler tamamlandığında proje sonlanır. Süre etmeni istenilen hedefe ulaşılması için belirlenen takvim ile oluşur. Kapsamı belirli olan projelerin belirli bir zaman diliminde gerçekleştirilmesi esastır. Belirlenen sürede ulaşmak istenilen hedefe ulaşılması projeler için bir gerekliliktir. Maliyet etmeni projenin kapsamı ve süresi ışığında belirlenir ve kullanılacak çeşitli kaynakların bedeli olarak tarif edilebilir. Kapsam ve süre boyutları değiştikçe projenin maliyeti de değişecektir. Bu nedenle projenin planlara uygun ilerlemesi projenin istenen maliyetle gerçekleştirilebilmesi için bir gerekliliktir.

Projenin istenen etkiye etkin bir biçimde ulaşması proje hedeflerine uyumuna bağlıdır. Projenin başta yapılan tanımlamaya uygun olarak gerçekleştirilmesi ancak projenin kapsam, süre ve maliyet hedeflerine uyumuyla mümkündür. Kapsamın elde edilmesi proje için yeterli değil-

dir. Kapsam sonucu yeterli etki elde edilmiş olsa da etkin bir proje için süre ve dolayısıyla maliyet hedeflerinin de tutturulması gerekmektedir. Maliyet kapsam ve sürenin bir fonksiyonu olduğu için diğer iki hedefte gerçekleşecek bir değişiklik maliyeti doğrudan etkileyecektir. Projenin beklenenden uzun bir sürede tamamlanması sabit maliyetleri artırırken kimi işlerin yeniden yapımını da gerektirebilir. Bu nedenle, istenilen süreyi aşan projelerin maliyeti de yükselecektir. Projeyi planlanandan daha kısa sürede bitirmek sabit maliyetleri düşürüp birim değişken maliyetlerin yükselmesine neden olurken dış kesimlerden hizmet alımlarının artmasına sebep olabilir. Genellikle bir projenin en düşük maliyet düzeyi planlanan kapsam ve sürede olanıdır. Bunun istisnası proje süresince çıkacak fırsatların değerlendirilmesi ile planlama aşamasında öngörülme-yen iyileştirmelerdir.



Şekil 3 - Proje hedefleri, kapsam, Maliyet ve Süre (Meredith & Samuel, 2000)

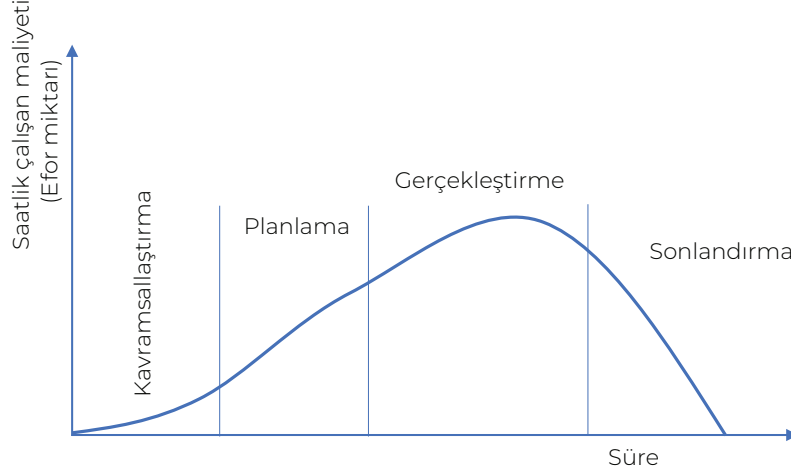
2.6. Proje Yaşam Döngüsü

Her proje belirli bir süre ile sınırlıdır. Süreklilik ilkesi gereği işletmelerde devamlılık esassen proje yaşam döngüsü bir sona sahiptir. Projenin yaşam döngüsündeki bu farklılık onun seyrini de şekillendirir. Proje yaşam döngüsü kavramsallaştırma, planlama, gerçekleştirme ve sonlandırma aşamalarından meydana gelmektedir. (Pinto & Slavin, 1988)

Kavramsallaştırma, üst yönetim tarafından stratejik bir ihtiyacın ortaya konulması ile başlayan süreci ifade eder. Kavramsallaştırma aşamasında problemin ne olduğu, bir proje geliştirme-nin bu probleme çözüm getirip getiremeyeceği, geliştirilecek projenin hedefleri, ile proje için yeterli kaynak bulunup bulunmadığı sorularına yanıt aranır. (Pinto & Slavin, 1988)

Planlama aşamasında, yapımına karar verilen projenin üst yönetim tarafından başlanması talimatı verilir. Belirlenen hedeflere ulaşılması için gerekli adımların ve kaynağın belirlenmesi, büt-

çeleme ve görevlerin dağılımı planlama aşamasının adımlarındandır. (Pinto & Slavin, 1988) Proje yöneticisinin bu aşamada projeye dahil olması beklenir.



Şekil 4 - Proje yaşam döngüsü (Pinto & Slavin, 1988)

Gerçekleştirme proje kapsamında belirlenen işlerin yapıldığı adımdır. Bu aşamada kaynaklar elde edilerek planlanan sonuçlara ulaşmak için kullanılır. Bu aşamada performans sürekli gözlenerek projenin planlandığı gibi seyrettiğinden emin olunur. (Pinto & Slavin, 1988) Projede birim zamanda en fazla harcama bu dönemde yapılır.

Son aşama sonlandırma aşamasıdır. Kritik hattaki görevler ve diğer görev tamamlandıkça sonlandırma aşamasına yaklaşılr. Sonlandırma aşamasında proje çıktıları teslim edilerek proje-

nin hedeflerine ulaşılmış olur. Proje hedeflerine ulaşıldığında proje takımı ya da takımları dağıtılır. Dağıtım sonucu geldikleri örgütteki konumlarına geri dönebildikleri gibi başka bir projeye de dahil olabilirler. Benzer durum beşeri kaynak dışındaki kaynaklar için de geçerlidir. Aynı ve mali kaynaklar da hami örgüte geri dönebileceği gibi bir başka projede kullanılmak üzere yeniden değerlendirilebilir. (Pinto & Slavin, 1988) Proje sonlandıktan sonra çoğunlukla proje çıktıları hami örgüt içerisinde sıradan işleyişe dahil olur.

2.7. Proje Görevleri

Bir işletmenin amaç ve hedeflerine bir işleve bağlı sıradan (rutin) işlerin tekrarlanması şeklinde ulaşılır. Bir projede ise birbiriyle ilintili sıra dışı (rutin olmayan) görevler tamamlanarak proje amaç ve hedeflerine ulaşılır. Her projenin kendine özgü olması unsurunu göz önüne aldığımızda projedeki görevlerin neden sıradan olmadığını daha iyi anlayabiliriz.

İşletmelerin işlevleri dahilinde yapılan gündelik işler işletmenin amaç ve hedeflerine ulaşmayı sağlamaktadır. Bir projede ise proje hedeflerine ulaşmak için yapılması gereken işler planlama aşamasında paketler halinde belirlenmektedir.

Bu iş paketlerinin projeye özgü belirlenmesi onları sıra dışı kılmaktadır.

Proje görevlerinin sıra dışı olmasının bir diğer nedeni projelere ayrılan sürenin kısıtlı olmasından kaynaklanmaktadır. Belirli bir sürede tamamlanması gereken projelerde görevler birbirini takip eder şekilde ilerlemektedir. Kimi görevler paralel bir biçimde ilerlese de projeler birbirini takip ederek kritik bir hat oluşturan görevler dizini ile tamamlanmaktadır. İşletme planları bir ya da birden fazla faaliyet dönemi kapsayacak biçimde yapılmaktadır. Bu esnada örgüt içerisinde yapılan işler bu planların hedef-

lerine hizmet ederek sürmektedir. Projelerde ise faaliyet dönemi anlayışının yerine proje bütçesi, süresi ve kapsamında yapılacak işler bulunmaktadır.

Proje görevlerine işlevlerin katkıları farklı dönemlerde gerçekleşebilmektedir. Hami örgütün ya da örgütlerin bölümlerinden farklı adımlarda katılımlar sağlanmaktadır. Yönetim kademesinin kavramsallaştırma aşamasında yoğun katkı

mı bulunurken proje yönetimi oluştuktan sonra sonlandırma aşamasına kadar bir katkısı bulunmaz. Projenin kapsamına bağlı olarak pazarlama, üretim işlevlerinin de farklı adımlarda katkısı bulunabilir. Finans işlevi projeye mali kaynak bulmak üzere gerçekleştirme aşamasından önce katkı sağlamaktadır. Muhasebe işlevi ise gerçekleştirme ve sonlandırma aşamalarında alınan raporlar ve yapılan denetimle varlığını hissettirir.

2.8. Proje Örgütü

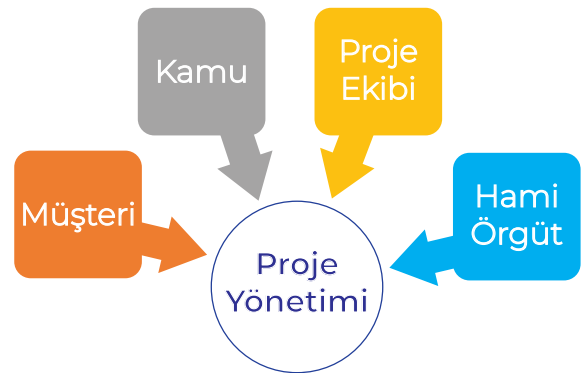
Projelerin örgütlenmesi çeşitli nedenlerle diğer örgüt yapılarından ayrılmaktadır. Projelerin kendine özgü çözümler gerektirmesi nedeniyle proje örgütlemesi takımlarla yapılır. Takımlar, bir ya da birden fazla alanda çalışanlar içeren iş gruplarıdır. Takımlar farklı işlevlerden ya da farklı örgütlerden oluşturulabilmektedir. Bunun yanında proje örgütü farklı ülke vatandaşlarından da oluşabilmektedir. Tüm bu nedenler yeniden ve özgün bir örgüt yapısını gerektirmektedir. Normalde üst yönetimin bir görevi olan dış iletişim proje yönetimi tarafından yapılmaktadır.

Proje örgütüne dair bu farklılıklar daha esnek ve yetkin bir proje yönetimi gerektirmektedir.

Projelerin nitelikleri göz önüne alındığında hiyerarşik örgüt yapısının projelerle uyumlu olmadığı görülmektedir. Üst, orta ve alt kademeler barındıran hiyerarşik örgütler yerine projelerde matris örgüt yapısının kullanıldığını görmekteyiz. Proje örgütü takımların koordinasyonunu, görevlerin takibini ve dış iletişimi yürüten bir merkezi yönetim ve takımlardan oluşan bir matris biçiminde oluşur.

2.9. Projenin Çıkar Sahipleri

Projenin çıkar sahipleri müşteri, hamî örgüt, proje ekibi ve kamudan oluşmaktadır. Çıkar sahiplerinin her biri farklı olan beklentilerini karşılama görevi proje yönetimine düşmektedir. Müşteri projeden büyük bir performans beklerken hamî örgüt kârlılık beklentisi içerisindedir. Proje ekibi proje hedeflerine odaklanarak projeyi tamamlamak isterken kamu ise projenin mevcut durumda fazla değişikliğe neden olmamasını isteyebilmektedir. Bu ve bunun gibi birçok beklenti altında gerçekleştirilmeye çalışılan projenin yönetimi sürekli bir optimizasyon problemine dönüşmektedir.



Şekil 5 - Proje çevresi

3. YÖNETİM

Yönetim, sistemin amacının gerçekleştirilmesi için planların yapılması, planları gerçekleştirecek örgütün kurulması, kullanılacak kaynakların elde edilmesi, kaynakların koordine edilmesi ve amaca doğru yöneltilmesi, en son ise amacın gerçekleşip gerçekleşmediğini belirlemek amacıyla denetimin yapılmasını kapsayan bir faaliyetler dizisidir. Planlamadan, denetime uzanan yönetim faaliyetleri sürekli bir karar verme sürecidir. Kaynakların nerede, ne zaman ve nasıl kullanılması gerektiğini belirleyerek en verimli çalışma biçimi sağlanmaya çalışılır. Bu çaba yönetim faaliyetlerinin her aşamasında gerçekleştirilen koordinasyon ile gerçekleştirilir.

Sistem yönetiminin geleneksel yönetim anlayışından farkı işlev temelinde değildir. Planlama, örgütlenme, yöneltme, eşgüdümleme, denetim işlevlerini barındıran yönetim sistem yönetiminde de aynıdır. Sistem yönetiminde fark bu işlevlerin

etkileşiminin ve üst sisteme katkısını daha gözle görünür hale getirmede yatar.

Sistemlerin yönetiminde sistemin yaşam döngüsü ve işlevlerin tekrarı bir ayrıştırma yaratmaktadır. Geleneksel yönetimin temelinde işletmenin sürekliliği anlayışı yer almaktadır. Proje yönetiminde ise projenin başlangıç ve bitişinin planlanması bu anlayışı değiştirmektedir. Geleneksel yönetim tüm işletmelerde bulunan yönetim, üretim, pazarlama, muhasebe ve finansman işlevlerine odaklı yürütülmektedir. Proje yönetiminin odağında ise -yine bu işlevlerin kısmen veya tamamen yer aldığı- görevlerden oluşan bir proje bulunmaktadır.

İşletme yönetiminde sürekli tekrarlanan işlev yönetimi hakimdir. Proje yönetiminde kimi tekrarlanan işlevler olabilirken birçoğu bir kerelik görevler bulunmaktadır. Bu iki farklı yapının yönetiminde de farklılıklar bulunmaktadır.

3.1. Sistem Analizinin Yönetimdeki Önemi

Sistem analizi sistemin, alt sistemlerin ve üst sistemin/sistemlerin beraber göz önüne alınmasını sağlayarak seçenekler arasında en uygun olanın belirlenmesinde daha ussal bir yöntem sunar. Sonucunda sistem daha etkin bir hal alırken diğer sistemlerin de iyileşmesi yönünde katkı sağlar.

Sistem Analizinin Aşamaları

1. Sistemin çözümlenmesi: Aşağıdaki faaliyetlerden oluşur:

a. Sistem işleyişinin gözlenmesi. Sistemin girdileri, nasıl işlediği ve çıktıları irdelenir.

b. Sistemin çevresinin belirlenmesi. Sistemi çevreleyen sistemler, çevreden alınan girdiler, çevreye verilen çıktılar irdelenir.

c. Sistemin alt sistemlerinin belirlenmesi. Sistemin alt sistemlerini oluşturan bileşenler, bunların amaca katkısı irdelenir.

d. Bileşenler arasındaki ilişkinin belirlenmesi. Sistem bileşenlerinin girdileri ve çıktıları ile birbirleriyle olan neden ve nasıl etkileşimde bulundukları irdelenir.

e. Verilerin toplanması, işlenmesi ve yorumlanması. Sistemin çevresine ve işleyişine ilişkin veriler toplanır, işlenir ve yorumlanır. Veriler ham halde kullanılmasından ziyade analiz ve senteziyle yeni veri üretilmesi de gerekebilir.

2. Sorunun saptanması: Sistemin tüm adımları (girdi, işleyiş ve çıktı) ve tüm bileşenleriyle gerektiği gibi işleyip işlemediği irdelenir. Sorunlar bu aşamada ortaya konulmaya çalışılır. Sorunları ortaya koymak için bunların mevcut durum ile arzu edilen durum arasındaki boşluğu dolduracak problemler olarak ifade edilmesi gerekir.

3. Amaçların belirlenmesi: Sorunun saptanması sırasında ortaya konan problemler sistemin

amaçlarını oluşturma ya da iyileştirme imkânı doğuracaktır. Amaçlar belirlenirken aşağıdaki özellikleri gözlemlenmelidir. Amaçlar,

a. Açık ve ölçülebilir olmalı. Amaçlar ölçütler olarak ifade edilmeli. Sistemin çözümlenmesi sırasında verilerden performans göstergeleri ortaya konulmalı. Bu performans göstergeleri sistemin işleyişi esnasında gözlenecek ve istenilen seviyede tutulmaya çalışılacaktır.

b. Anlaşılabilir ve uyumlu olmalı. Sistemin amaçları alt sistemler tarafınca benimsenip gerçekleştirilebilir olması için anlaşılabilir olmalı. Amaçlar üst ve alt sistemlerin amaçlarıyla uyumlu olmalı.

c. Gerçekçi ve ulaşılabilir olmalı. Amaçların gerçekleştirilebilir olmasının önemli bir özelliği de ulaşılabilir olmasıdır. Bir performans yaratması beklenen ölçütlerin ulaşılabilir olması gerekmektedir. Amaçların analizin sonuçlarına göre verimlilik sağlaması gereken düzeyde belirlenmesi beklenmektedir. Amaçlar alt düzeyde belirlenirse istenen verimlilik yakalanamazken ve çok üst düzeyde belirlenirse gerçekleştirilemez.

4. Tüm bu süreç sonunda belirlenen amaçların sisteme yön gösterebilir olduğundan emin olunmalıdır. Eğer amaçlar yön gösterebilme yeterliğine sahip değilse ortaya konulan problemlere yanıt aranmaya devam edilmelidir.

5. Seçeneklerin Araştırılması: Amaçları gerçekleştirecek yolların tespit edilmesi ve bu seçenekler hakkında veri toplanması ve bilgilerin araştırılması aşamasıdır.

6. Seçeneklerin Değerlendirilmesi : Belirlenen seçeneklerin sonuçlarının ve sistem üzerine etkilerinin sistematik bir biçimde değerlendirme sürecidir. Seçeneklerin amaçları gerçekleştirme bakımından değerlendirilerek filtre edilmesinin ardından seçeneğin uygulanması haline sistemi nasıl etkileyeceği de irdelenmelidir. İleride yaratacağı olasılıkları mümkün mertebe değerlendirilmesini sağlayacak bu irdeleme süreciyle ge-

lecekte karşılaşılabilecek olasılıklar kestirilmeye çalışılır.

Sağlıklı seçeneklerin bu şekilde değerlendirilebilmesi için modellenmesi gerekmektedir. Seçeneklerin değişkenleriyle oluşturulacak modelde seçeneklerin sonuçları ve etkilerini değerlendirmek mümkün olabilecektir. Bu aşamada nitel analiz yöntemi kullanılarak seçeneklerin çevre ve sistem açısından değerlendirmesi yapılabilir. Seçeneklerin çevre ve sisteme etkisinin sonuçları değişkenlere dönüştürülebilir. Seçeneklerin değişkenleri belirlendikten sonra nicel analiz yöntemleri kullanılarak sonuçlar net bir biçimde değerlendirilebilir.

7. En iyi seçeneğin seçilmesi: En iyi (optimum) seçenek en düşük maliyetle en fazla faydanın çevre ve sistem açısından elde edildiği seçenektir. En iyi seçenek ancak zıt kriterler değerlendirmeye alınarak belirlenebilir. Örnek olarak, optimum seçenek genellikle minimum maliyetli ya da maksimum faydalı seçenek değildir fakat en düşük maliyetle en fazla faydayı alabildiğimiz seçenektir.

8. Tasarım: Tercih edilen seçeneğin nasıl uygulanacağı konusundaki tasarımın yapıldığı aşamadır. Bu aşamada uygulama öncesindeki hazırlıklar, uygulama güzergahı ve uygulama sonrası faaliyetlerin belirlenir. Bunların sistemin diğer faaliyetleriyle ilişkilendirilmesi de bu aşamada yapılmaktadır. Faaliyetler ve ilişkileri ortaya koyan PERT (Program Evaluation and Review Technique – Program Değerlendirme ve Denetleme Tekniği) ve CPM (Critical Path Method – Kritik Hat Yöntemi) gibi şebeke analiz teknikleri ile oluşturulan proje yönetim araçlarından bu aşamada faydalanılmaya başlanır.

9. Uygulama ve İzleme: Hazırlıkları tamamlanan seçenek uygulamaya konur. Adım adım gerçekleştirilmeye çalışılan programın istendiği gibi gittiğinden emin olmak için faaliyetlerin ve aşamaların izlenmesi gerekmektedir. Bu sayede

sistemin istenen yönde ilerlediğinden emin olunabilir. Eğer, uygulama aşamasında tasarımdan farklı bir sonuç ortaya çıkıyorsa bu farklılıklara karşın önlem alınarak sistemin düzeltilmeye çalışılır.

10. Genel Değerlendirme: Uygulama süreci tamamlandığında seçeneğin sonuçları değerlendirilir. Sistemin amaçlarının ne derece elde edildiği, seçenek ya da seçeneklerin katkısı irdelenir. Uygulama başarıyla tamamlanmış olsa da sistemin amaçlarına ulaşılmamış ya da yeterli derecede ulaşılamamış olabilir. Sistemin genel olarak istenilen noktada olup olmadığı anlaşılmalıdır. Eğer yanıt olumsuz ise nedenini ortaya koyacak analiz süreci tekrarlanmalıdır. Eğer yanıt olumlu ise seçeneklerin ortaya çıkardığı yeni problemler, sistemin etkililiği ve etkinliği gözden geçirilmelidir.

Sistem analizinin geleneksel problem çözme yöntemlerinden temel farkı diğer sistemleri de değerlendirmeye katan, sistematik bir yöntem olmasıdır. Yapılan analiz sonucunda amaçların neler olması gerektiği, bu amaçlara ulaşabilecek en iyi yöntemin ne olduğu, yöntemlerin ne dere-

ce başarılı olduğu anlaşılmış olur. Sistem analizinin her uygulanmasında sistem daha etkili ve etkin bir hale getirilebilecektir. Sonucunda sistemler toplamı için iyileştirme elde edilmeye daha fazla yaklaşılabacaktır.

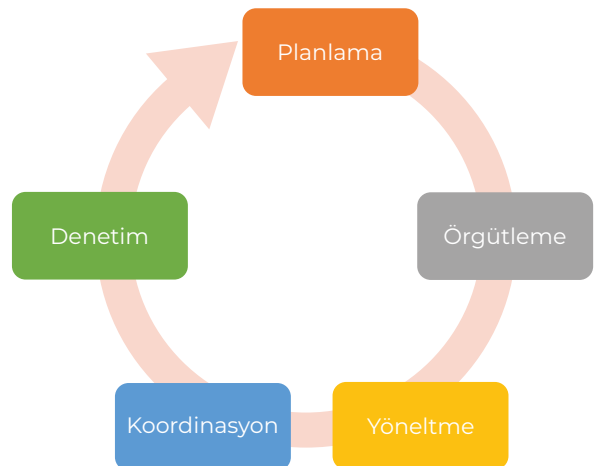
Sistem analizi yöneylem araştırması, matematiksel modelleme yöntemleri, maliyet-etkililik analizi, maliyet-fayda analizi gibi analiz yöntemlerini akla getirebilmektedir. Sistem analizi bu yöntemleri kapsamakla birlikte bunlardan biri değildir. Bu yöntemler seçeneklerin ve sistemin değerlendirilmesi için kullanılabilir olsa da sistem analizi sistemin üst sistemler ve alt sistemler de göz önüne alınarak değerlendirilmesidir. Sistem analizi yoğunlukla bilgi işlem sistemlerinin düzenlenmesinde kullanıldığı için bu alana özgü olduğu anlaşılabilmektedir. Bu sistem analizinin anlamını ve kullanım alanını daraltacak bir anlayışı doğurmaktadır. Sistem analizi tüm sistemler için kullanılabilecek bir araştırma yaklaşımıdır. Doğru yönetim ile etkililiğinin ve etkinliğinin artırılması mümkün olan tüm sistemler için kullanılabilmektedir.

3.2. İşlev Yönetimi

Geleneksel işletme yönetimi hiyerarşik bir yapıda işlevlerin yönetimi biçiminde gerçekleşmektedir. Sıradan işlerin gerçekleştirilmesine odaklanan geleneksel işletme yönetimi genel itibarıyla işlemlere dayalı bir örgüt yapısını benimser. İşleve dayalı bölümlerden oluşan örgüt işletmenin işlevlerin daha etkin ve etkili bir biçimde yönetilmesini sağladığı için tercih edilir. Seçili işlev konusunda uzman bir yöneticinin idaresine bırakılan işlevler üst düzey yönetim tarafından denetlenerek işletme amaçlarının gerçekleşmesi sağlanır.

Bölümlerin işletmenin amaçlarına hizmet ettiğinden emin olmak için hiyerarşik bir örgüt yapısı gerekmektedir. Sistem anlayışı ışığında işletme sisteminin bir alt sistemi olan bölümlerin

amaçlarının işletmenin amaçlarının önüne geçmemesi için hiyerarşi teşkil edilmektedir.



Şekil 6 - Yönetimin işlevleri

İşletme faaliyetleri büyüdükçe farklı örgütlenme yapılarına ihtiyaç duyulabilmektedir. İşletmenin büyümesine göre bölgeye, piyasa yapısına veya ürün çeşidine göre yeni örgütlemeler yapılabilmektedir. Bahsi geçen bu yeni örgütler de bir işlevin çoğaltılması, bir ya da birkaç işlevin farklılaştırılması veya yine işleve dayalı örgütlemenin tekrarlanması olarak gerçekleşir. Yeni örgüt her ne kadar farklılaşmış olsa da yine hiyerarşik işlevler yönetimi biçiminde gerçekleşmektedir.

4. PROJE YÖNETİMİ

İşletme içerisindeki sıra dışı işlerin sayıca artması ve büyümesi, bu işlerin birden fazla örgütün ortak çabasıyla gerçekleştirilir hale gelmesi işlev yönetimini yetersiz kılmıştır. Askeri ve bayındırlık alanları başta olmak üzere devlet yönetiminde sıklıkla kullanılan yönetim yaklaşımı işletmeler tarafından da kullanılmaya başlanmıştır.

Pazarın ve ürünlerin çeşitlenmesi, üretim teknolojilerindeki gelişimin yardımıyla daha özgün çözümlere olan eğilimi artırmıştır. Artan rekabet, karmaşılaşan örgüt yapıları daha odaklı çözümler için çözüme odaklı bir ekibin çalışmalarına ihtiyaç duyulmasına neden olmaktadır. Bilgi ve iletişim teknolojilerindeki gelişimler ise proje ekibi kurulumu ve takibini kolaylaştırmaktadır. Tüm bu nedenlerle proje yönetimi daha fazla sektörde ve daha fazla işletmede görülür hale gelmiştir.

Projeler ve bunların yönetimi çok eski zamanlara dayanmaktadır. Modern çağda proje yönetiminin başlangıç noktası olarak Manhattan Projesi anılmaktadır. Proje yönetimi olarak adlandırılan bu yönetim anlayışının işlev yönetiminden farklı olmasına neden olan temel dayanaklar bulunmaktadır.

Projelerin belirli bir ömrü bulunmaktadır. İşletmenin sonsuz kabul edilen ömrü ve işletmeden beklenen ürünlerin yeknesak olması nedeniyle işletme işlevlerinin sürekli tekrarlanacak

İşlev yönetiminde işletmenin sürekliliği ilkesine dayalı olarak işletme ömrü süresiz olarak kabul edilir. Buna dayalı olarak işletme ömrü boyunca planlama, örgütlenme, yöneltme, koordinasyon ve denetimden oluşan yönetim süreci sürekli tekrarlanır. Bir alanda uzman olan bir yöneticinin bölümü işletmenin amaçlarına hizmet edecek şekilde yönetmesi ile işlev amaçları gerçekleştirilir.

biçimde yönetimi esastır. Projeler ise çoğu bir seferlik farklı görevlerden oluşmaktadır. Proje süresince tekrarlanan işlerden oluşan kimi görevler de bulunmaktadır. Fakat işlev yönetiminden farklı olarak bu görevler de diğerleri gibi planlanır.

Projeler birden fazla işlevin yönetimini içerir. Geleneksel yönetim anlayışında işlevlerin o alanda uzman bir yönetici tarafından yönetimi esastır. Projeler ise birden fazla işlevin katkısıyla tamamlanabilmektedir. Bu nedenle birden fazla alandan gelen çalışanların koordine edilmesi ve uyumlaştırılması esastır. Projelerde birden fazla örgütün kaynakları kullanıldığı olabilmektedir. Bu koordinasyon esnasında görevlerde yapılacak işler ve kullanılacak kaynaklar bir işlevin koordinasyonundan çok daha karmaşık olmaktadır. Bütün bu nedenler doğrultusunda proje yönetimi hiyerarşik yönetimden ziyade katılımcı bir yönetim biçiminde gerçekleştirilir.

İşletmelerde yönetimin farklı bir görev olarak kabul edilmesinde sanayi devrimi sonrası kurulan fabrikalar ve büyüyen örgütler önemli bir yer almaktadır. Daha öncesinde çalışanların idaresi de bir meşgaleiyken bir tesis içerisinde bir örgütün dahilindeki çalışanların yönetimi kendi zorluklarını doğurmuştur. Makinelerin performansına uyum sağlaması beklenen çalışanların bilimsel yöntemlerle idaresi bir gereklilik ha-

lini almıştır. Fabrikalarda bir anda zayıf halka durumuna gelen çalışanların doğru idaresi için Frederick Taylor'un (1910) yayınladığı Bilimsel Yönetim İlkeleri benimsenmiştir. İş yapışta *görev* terimini ön plana çıkartan Taylor işlerin verimli bir biçimde yapılabilmesi için çalışanların ve yönetimin ayrılmasını, görevlerin tanımlanmasını, çalışanların bu görevleri yapabilir nitelikte seçilmesini ve örgüt içinde eşgüdümün sağlanması gerektiğini belirtmiştir. Bilimsel Yönetim İlkelerine katkılarıyla bilinen Henry Gantt 1917 yılında görevlerin şematik takibi ile verimlilik sağlayan Gantt şemasını geliştirmiştir. Gantt şeması, CPM ve PERT yöntemleriyle birlikte halen proje yönetiminde kullanılmaktadır.

Günümüzde bilgisayar ve internet teknolojilerinin sağladığı imkânlarla proje yönetimi daha yaygın bir biçimde kullanılmaktadır. Görevlerin takibi ve kaynakların eşgüdümü ile verimlilik sağlayan proje yönetimi teknikleri sıradan bütün işletmelerde kullanılmaya başlamıştır. Sıradan işlerin takibini de kolaylaştıran bu yöntemler gittikçe daha fazla örgütte benimsenmektedir. Proje yönetimi tüm işletmeler için çeşitli avantajlar sağlamaktadır:

- Geliştirme süreçlerinde ekonomiklik
- Kaynakların verimli kullanımı
- Etkin kontrol
- Kalite ve güvenlik temini
- Eşgüdüm ve motivasyonda iyileşme
- Müşteriler ve diğer dış paydaşlarla daha etkili etkileşim

Proje yönetimi işlev yönetimiyle aynı işlevleri içermektedir: Planlama, örgütlenme, yöneltme, eşgüdümleme ve kontrol. Proje yönetimi işlev yönetiminden ayıran bu işlevlerin uygulanış biçimidir. Planlama proje kapsamına göre görevlerin, sürenin ve maliyetin belirlenmesi biçiminde gerçekleşmektedir. Örgüt görevlerin tamamlanmasını sağlayacak bir ya da birden fazla alandan

çalışanlardan oluşan takımlardan oluşmaktadır. Farklı bir kesite sahip takımların yetkisinin daha yüksek, yönetiminin daha az merkeziyetçi olması esastır. Takımların proje amaç ve hedeflere yöneltimi de örgütlemenin bir sonucu olarak şekillenmektedir. Yetki göçeriminin çok daha yüksek gerçekleştiği proje yönetiminde hedeflerin belirlenmesi ve görev sürelerinin takibi ile yöneltim gerçekleştirilmektedir. Görevlerin ve görevlerin tamamlanması için gerekli kaynakların planlama aşamasında belirlenmesi ve bunların şematik bir biçimde ortaya konulmasıyla koordinasyon çok daha belirgin bir biçimde yapılmaktadır. Görevlerin süre ve kaynaklarıyla etiketlemesi hangi zaman aralığında hangi işlerin yapılacağı ve buna bağlı olarak hangi kaynağın nerede kullanılacağı kolayca tespit edilebilmektedir. Planlama aşamasında ve sonrasında eşgüdümü kolaylaştıran bu yapı projeleri daha verimli kılmaktadır. Merkezden uzak, hiyerarşik olmayan örgüt ve yöneltim proje yönetiminde kontrolü daha da kritik hale getirmektedir. Projenin hedeflerinin tutturulabilmesi için görevlerin istenilen niteliklerde yapıldığından ve kritik hattın takibinden emin olunması gerekmektedir. Kapsam, süre ve maliyet hedeflerine ulaşmak için kontrolü kullanan proje yönetimi görevleri sırasıyla takip ederek projeyi tamamlamaktadır.

Projelerin belirli bir soruna yönelik bir çözüm olarak belirli hedeflerle tasarlanması proje yönetimini kolaylaştırmaktadır. Proje yönetiminin avantajları proje yönetimi adımlarında yatmaktadır. Proje yönetimi işleyişi şu şekilde gruplanabilir:

- Belirli bu yapı içerisinde proje görevleri ve görevler dahilindeki işler belirlenir,
- Görevler birbiri ile ilişkilendirilerek kritik hat ortaya konulur,
- Kritik hat ve diğer görevler takip edilerek projenin istenildiği gibi gittiğinde emin olunur,
- Projede aksayan noktalar bulunuyorsa bunlar kolayca tespit edilebilir.

5. PROJE YÖNETİCİSİ

Proje yöneticisi, belirlenen amaçları gerçekleştirmek için proje sistemini kuran ve bu sistemi sonuçlandıran kişidir. Projenin tanımlanma ve kontrol işlevlerini gerçekleştirmek proje yöneticisinin görevleridir. Proje paydaşları ve proje yönetimi arasında iletişimi tesis etmek proje yöneticisinin sorumluluğundadır. Bunun yanında paydaşların taleplerinin proje kapsamında karşılandığından emin olmak ve talepler arasında çıkan çatışmaların çözümünü geliştirmek de proje yöneticisinin görevleri arasında yer almaktadır.

Projeden beklentilerin karşılanmasını sağlayacak olan proje yöneticisi sistem oluşturma ve yönetme bilgi ve becerisine sahip olmak durumundadır. Proje beklentilerini proje hedeflerine uygun bir biçimde karşılandığında emin olması gereken proje yöneticisinin proje yönetimi konusunda da birikime sahip olması gerekmektedir. Başarılı bir proje yöneticisinin akademik başarı, yöneticilik tecrübesi ve proje üzerine teknik birikimin yanında kendini geliştirme yönünde bir motivasyonu da bulunmalıdır.

5.1. Proje Yöneticisinin Rolü

Proje yöneticisi, projenin hedeflerini iyi bir biçimde özümsemiş olarak, kaynakların doğru zamanda temin edildiğinden ve takımların koordinasyonundan emin olmalıdır. Proje ilerleyişi esnasında çıkan sorunlar karşısında projenin bütçe ve plan kısıtlarından şaşmadan hangi ödünlerin verilebileceğini proje yöneticisinin belirlemesi gerekecektir.

Proje yöneticilerinin karşılaştıkları en büyük problem yetki konusunda sınırlarla karşılaşırken proje sonuçları bakımından tam sorumluluk yüklenmesi olarak tarif edilmektedir. (Meredith & Samuel, 2000) Hami örgüt tarafından projeden beklenenler doğrudan proje yöneticisinden sorulacağı için yetki ve sorumluluk dengesinin projenin daha ilk (kavramsallaştırma) safhasında doğru kurulması önemlidir.

Projeye özgün olarak oluşturulan proje örgütü birkaç işlevden ya da birkaç farklı örgütten çalışanları barındırabilmektedir. Proje örgütü

dahilinde bulunan takımların ve takım çalışanlarının uyumlaştırılması ve eşgüdümlemesi proje yöneticisine düşmektedir. Proje ekibinin yanında üst yönetim, müşteri ve kamunun projeden taleplerini karşılamak yine proje yöneticisine düşmektedir. Proje hedeflerinden uzaklaşmadan hangi ödünlerin verilebileceğine, taleplerin nasıl uzlaştırılabileceği proje yöneticisinin vermesi gereken kararların arasında yer almaktadır.

Proje hedefleri ve paydaş talepleri altında yüksek bir sorumluluğa sahip proje yöneticisinin bunları karşılayacak yetkilerle donatılması bir zorunluluktur. Belirli bir süre içerisinde çok çeşitli işlerle haşır neşir olan proje yöneticileri -risk almaya ve problem çözmeye yatkınlarsa- yüksek bir iş tatmini duymaktadır. İş tatmininin yanında tamamlanmış projelerin sağladığı saygınlık proje yöneticisi olmanın maddi olmayan ödülleri olarak sayılabilir.

5.2. Proje Yöneticisinin Nitelikleri

Bilgi sistem projelerinde sistem analizi ve işlevsel gereklilikler projenin teknik boyutunu oluşturmaktadır. (Meredith & Samuel, 2000)

Bir işlev yöneticisinden yüksek düzeyde teknik birikim ve analiz yeteneği beklenirken bir proje yöneticisinden genel kültür ve sentez yete-

neği beklenmektedir. Proje yöneticisinin sistem yaklaşımını iyi anlaması bu nedenle önemlidir. Proje yöneticisi projenin süreçlerini bir sentez becerisiyle derleyerek proje sisteminden beklenen çıktının elde edilmesini sağlamalıdır.

Proje yöneticisi eğitim alarak ve tecrübe ederek proje yönetimi bilgi ve becerisine sahip olması gerekmektedir. Fakat bu, proje yöneticisinin herhangi bir projeyi yönetebilmesini sağlamaz. Proje yöneticisinin devraldığı projenin konusuyla ilgili teknik bir birikime sahip olması da gerekmektedir.

Proje yöneticisi üst yönetim tarafından projenin ilk aşamalarında projeye dahil edilmektedir. Kimi projelerde projenin teknik boyutu göz önüne alınarak proje yöneticisinin değiştiği de olmaktadır. Projeye dahil olduğu noktadan başlayarak örgütlenme, kadrolama, bütçeleme, planlama, yöneltme ve kontrol görevlerini proje yöneticisi üstlenmektedir. Yönetim proje yöneticisine devredildikten sonra üst yönetimin projeye müdahil olmaması proje yöneticisinin işini yapabilmesi açısından bir gerekliliktir. Projenin hedeflerinden ayrılmadan özgün ilerleyişini sürdürebilmesi açısından proje ekibinde yer alan işlevsel yönetici ve çalışanların baskın çıkmaması da benzer biçimde dikkat edilmesi gereken bir diğer husustur. Projenin özgün bir süreç olduğu ve hami örgütün diğer faaliyetlerine benzediğinin anlaşılması önem arz etmektedir. Üst yönetimin yapabileceği bir mikro-yönetim (proje yöneticisinin sürekli ne yapması gerektiğinin belirlenmesi) ve bir işlevin uzmanlarının projede baskın olması projenin özgünlüğünü ortadan kaldırarak diğer faaliyetlerle aynı sonucu almasına neden olacaktır.

Proje yöneticisinde olması gereken ilk özellik güvenilirliktir. Güvenilirliğin ilk boyutu teknik güvenilirliktir. Proje yöneticisi projenin tüm kesimlerince -üst yönetim, müşteri, proje ekibi- teknik açıdan yeterli bulunmalıdır. Teknik yeterlilik proje ekibinde teknik anlamda daha nitelikli kişilerin bulunmaması anlamını taşımamaktadır. Projenin teknik boyutunu anlayarak bunu idare etmek proje yöneticisinin yeterliliğini belirlemektedir. Güvenilirliğin ikinci boyutu yönetsel yeterliliklerdir. Proje ekibini ve projeye aktarılan

kaynakları yönetme, proje paydaşlarının talepleri arasında dengeyi bulma sorumluluğu proje yöneticisindedir. Projeyi tüm etkenler altında başarıyla tamamlamak için yönetsel becerileriyle güven sağlamalıdır.

Proje yöneticisinin başarılı olmasını sağlayacak bir özellik duyarlılıktır. Teknik ve yönetsel birikim ile iş tecrübesi sorunların nerede ne zaman çıkabileceği konusunda proje yöneticisine bir duyarlılık kazandıracaktır. Ön-etkin bir yönetim becerisi kazandıracak bu duyarlılık sayesinde problemler oluşmadan ya da büyümeden çözmek mümkün hale gelecektir.

Projeyi başarıyla sonlandıracak bir proje yöneticisinde önderlik bulunması gerekmektedir. Önderlik tanımı en zor bulunan terimlerden birisidir. Bunun nedeni önderlik özelliklerinin kaynağının çeşitli olmasıdır. Önderlik teknik bilgi ve beceri, daha önce alınan sonuçlar, aranan tutum ve davranışlar, iletişim becerisi, dış görünüş, makam gibi birçok etken sonucu atfedilmektedir. Yapısal ve yapısal olmayan etkenlerden ortaya çıkan önderlik insanları belirli bir yöne sevk etme yeteneğidir. Tüm kesimlere projeyi sonlandırabileceğine inandırması için bir proje yöneticisinde önderlik niteliğinin bulunması gerekmektedir. Proje yöneticisinin etik ilkelere bağlılığı proje ekibi başta olmak üzere tüm kesimlerin projeye olan inancı artırarak projenin yönetimini kolaylaştıracaktır. Projenin başında ulaşmak istenilen nokta, projeden beklenenler belirliken oraya ulaşmayı proje yöneticisinin önderliği sağlamaktadır.

Sürekli karşımıza çıkan projenin paydaşları ve proje hedefleri kolay yönetilen unsurlar değildir. Proje dahilindeki günlük uğraşların yanında bu unsurları gözetmek yüksek strese neden olabilmektedir. O nedenle, bir proje yöneticisinde olması gereken bir diğer nitelik stresle başa çıkabilme özelliğidir. Projenin sıra dışı işler bütünü olması sürekli bir problem çözme sürecine dönüşmesini doğurmaktadır. Sıradan işlerden fark-

lı olarak yaratıcılık ve problem çözme yeteneği gerektiren proje yöneticiliği ödüllendirici olabilir gibi uygun olmayan bireyler için sarsıcı ola-

bilmektedir. Farklı kültürlerle ve beklentilere maruz kalacak proje yöneticisinin bunlarla çalışacak esnekliği kendinde bulması önemlidir.

6. PROJE YÖNETİM ARAÇLARI

“You can’t produce a baby in one month by getting nine women pregnant” Warren Buffet

6.1. Programlama

Proje yönetiminde programlama proje kapsamında yapılacak işlerin sırasıyla takip edilebilmesini sağlayan bir takvim oluşturma sürecidir. Proje yöneticisinin projeyi izlemesi ve kontrol edebilmesi için bir programın oluşturulması elzem görülmektedir. Bunun yanında, proje ekibin bütünü görebilmesi ve daha rahat eşgüdüm yakalayabilmesi için de program fayda sağlamaktadır. Projelerde istenilen katılımcı yönetim için program önemli bir yer tutmaktadır.

Proje kapsamında yapılacak her iş programda aynı düzeyde yer almamaktadır. Farklı düzeylerdeki görevleri programlamak için farklı seviyede programlar yapılmaktadır. İyi uygulama örneği olarak bir asıl programın yapılması asıl programda yer alan görevlerin iş paketlerini detaylandırılan programların da ayrıca yapılması verilmektedir. Programlama esnasında detaylarda boğulmamak için programlamanın belli bir seviyede tutulması gerekmektedir. (Meredith & Samuel, 2000) Örnek olarak, proje ekibi ya da takımlar tarafından bilinen adımları (prosedürleri) programa aktarmak gereksiz bir kalabalık yaratacaktır.

Tanımlama aşamasında belirlenen işler programlama esnasında, bir şebeke oluşturacak biçimde, sıraya konulur. Oluşan şebeke takip edilebilir bir şemanın oluşmasını sağlayarak proje yönetimine birçok avantaj sağlar: (Meredith & Samuel, 2000)

- Planlama, takvimleme, izleme ve kontrolün tutarlı bir biçimde yapılabilmesi kolaylaşır,
- Görevlerin, iş paketlerinin ve iş bileşenlerinin arasındaki bağımlılığı görselleştirir,
- Kimlerin hangi görevde hangi zaman aralığında rol alacağını belirler,
- İşlev bölümleri arasında gerekli iletişimin kurulmasını sağlar,
- Projenin beklenen tamamlanma tarihinin tayin edilmesini sağlar,
- Gecikme yaşandığında projenin bitiş tarihini öteleyecek olan kritik hat görevlerini belirginleştirir,
- Bir gecikme yaşandığında projeye zararı dokunmayacak gevşekliliğe sahip görevlerin görülmesini sağlar,
- Projede gecikme olmaması için görevlere hangi tarihte başlanması gerektiğini gösterir,
- Görevler arasında zaman ve diğer kaynakların koordinasyonunun yapılmasına görsel açıdan yardımcı olur,
- Hangi görevlerin eşlenik hangi görevlerin sırayla yapılabileceğini gösterir,
- Görevlerin bağımlılığını göstererek çatışmaların çıkmasını engeller,
- Kullanılan bilgiye dayanarak projenin farklı tarihlerde bitme olasılıklarını gösterir.

6.2. Şebeke Teknikleri: Program Değerlendirme ve Gözden Geçirme Tekniği (PERT) ve Kritik Hat Yöntemi (CPM)

Program Değerlendirme ve Gözden Geçirme Tekniği (PERT¹) 1958’de ABD Deniz Kuvvetleri, Booz-Allen Hamilton ve Lockheed firması tarafından Polaris füze/denizaltı geliştirme projesi için geliştirilmiştir. Kritik Hat Yöntemi (CPM²) aynı dönemde DuPont firmasınınca geliştirilmiştir. PERT daha çok araştırma geliştirme projelerinde kullanılırken CPM ise inşaat sektöründe benimsenmiştir. İki teknik arasındaki bir diğer fark ise proje ilerleyişine yaklaşımlarıdır. PERT projenin zaman boyutuna odaklanarak belirlenen takvimde tamamlanma olasılığını ortaya koymaktadır. CPM ise PERT şebekesinden farklı olarak görevlerin zaman boyutunu belirli kabul eder. Zaman ve maliyeti odağına alan CPM zaman maliyet takası yaparak projenin daha yüksek maliyetle daha kısa sürede bitirilip bitirilemeyeceğinin tahlil edilmesini sağlar. CPM faaliyetlerin çarpışma testinden geçirilerek daha fazla harcama yapılarak proje süresinin ne kadar kısaltılabileceğini görmemizi sağlar. Her iki yöntem de kritik hatta bulunan görevleri ve diğer görevleri ortaya koymaktadır. (Meredith & Samuel, 2000)

Projenin programlanması esnasında kullanılan önümüze çıkacak kavramlar etkinlik, olay, şebeke, hat, kritik olarak listelenebilir. Etkinlik projede gerek duyulan bir görev veya görevler kümesini ifade etmektedir. Bir veya birden fazla etkinliğin sonuçlanmasıyla ortaya çıkan sonuca olay adı verilmektedir. Projedeki tüm etkinliklerin ve olayların birleştirilmesi ile ortaya şebeke çıkmaktadır. Şebeke etkinlik ve olayları soldan sağa oklarla birleştirilmiş bir şekilde dizerek projenin örüntüsünü ortaya koyar. Böylece, etkinliklerin birbirini takip eder bir biçimde gerçekleştirilmesi kolaylaşır. Hat şebekede birleştirilen etkinliklerin ortaya koyduğu etkinlikler dizisini ifade etmektedir. Kritik, bir etkinlikte ya da bir hatta yaşanan gecikmenin projenin tamamlanma tarihini ötelemesidir. Projenin omurgasını oluşturan kritik hat içerisindeki etkinliklerin proje yönetimi tarafından titizlikle takip edilmesi gerekmektedir. Projeyi daha erken bir tarihte bitirilip bitirilemeyeceği de kritik hat üzerinde yapılan çarpışma testiyle anlaşılır. (Meredith & Samuel, 2000)

6.3. Şebeke Çizimi

Şebeke çiziminin nasıl yapıldığını bir örnek üzerinden görmek istersek aşağıdaki gibi bir plandan yarın alabiliriz.

Örnek Etkinlik Planı				
Hedef:				
Performans kriteri:				
Sınırlamalar:				
Görev	Halef	Süre (gün)	Maliyet	Takım
A	-	4		
B	-	3		
C	A	7		
D	B	5		
E	B	2		
F	C, D	6		

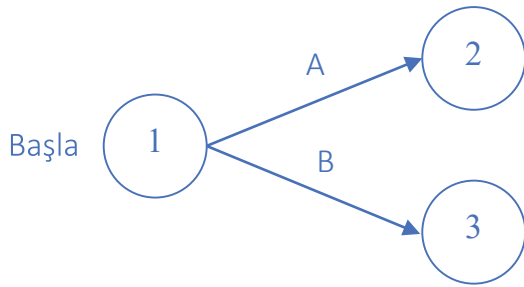
1 Program Evaluation and Review Technique

2 Critical Path Method

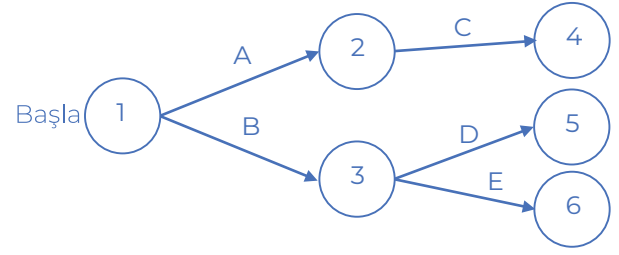
PERT ve CPM’le özdeşleşmiş şebeke çizim yöntemi olan AOA³ etkinlikler oklar üzerinde olaylar ise boğumlar ile gösterilir. Etkinlikler şebeke içerisinde üç farklı yapıda karşımıza çıkmaktadır: 1. Yalnız selefli olanlar 2. Hem halef hem de selefli bulunanlar 3. Yalnız halefli bulunanlar. Verilen örnekte A ve B birinci tür etkinliklere girmektedir. Bunlar projenin başlama-sıyla devreye alınabilen etkinliklerdir. İkinci tür etkinliklere C ve D gösterilebilir. Bu etkinliklere halefli olan etkinlikler tamamlanmadan başlanamaz. Bunları takiben ise örneğimizdeki selef etkinlikler E ve F başlayacaktır. Son iki etkinlik tamamlanınca proje de sonlanma olayıyla tamamlanmaktadır. Bu örneğin AOA şebekesi aşağıdaki gibi görülecektir.

Örneğimizde A ve B görevlerinin halefli olmadığı için proje başlangıcı (birinci olay) itibariyle her iki etkinliğe de başlanabilmektedir. Bu etkinliklerin tamamlanması ikinci ve üçüncü olaylara sonuç vermektedir. Başlangıç olayı projenin ilk boğum noktası olurken ikinci seviye boğumlarımız 2 ve 3’tür.

Etkinlik A’nın selefli C olduğu için A etkinliği tamamlanınca C etkinliğine başlanmaktadır. B etkinliği sonuçlanınca ise ona bağımlı olan D ve E etkinlikleri başlayabilecektir.

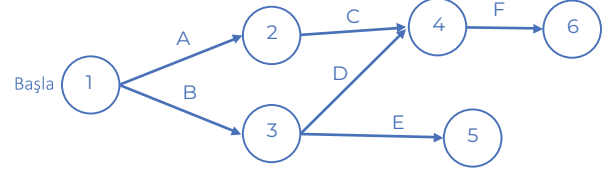


Şekil 7 -AOA adım 1



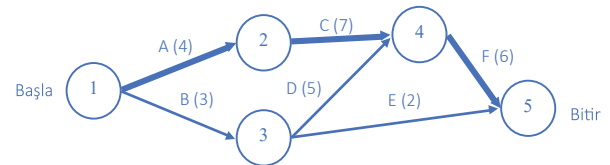
Şekil 8 -AOA adım 2

Görevlerin takibini yaptığımızda E ve F etkinliklerinin selefli olmadığı görülebiliyor. B etkinliği tamamlanınca E etkinliğine başlayabilirken F etkinliğine başlamak için C ve D etkinliklerinin tamamlanması gerekmektedir. O halde C ve D etkinliklerinin aynı olaya sonuç verdiği çıkarımını yapabiliriz. Bu haliyle ikinci adımın yeniden ele alınarak şebekenin aşağıdaki gibi çizilmesi doğru olacaktır.



Şekil 9 -AOA adım 3

Etkinlik F olay 6 ile sonuçlandığında projenin bütün görevleri tamamlanmış olmaktadır. Artık proje tamamlandığına göre son boğum olarak şebekeye bitiş boğumu eklenerek F ve E etkinlikleriyle projenin sona erdiği gösterilmelidir. Bu düzenlemeyle şebekenin son hali aşağıdaki gibi oluşacaktır.

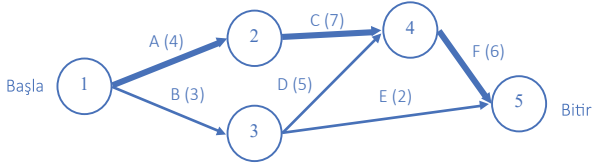


Şekil 10 -AOA adım 4

3 Activity-On-Arrow (AOA) Etkinliklerin ok üzerinde gösterildiği şebeke çizim yöntemidir. Etkinlikler ok üzerinde gösterildiği için olaylar boğum noktalarında gösterilmektedir. Bir diğer şebeke çizim yöntemi AON (Activity On Node -Etkinlik Boğum Üzerinde) bulunmaktadır. AOA daha yaygın kullanıma sahip olduğu için bu yöntemin uygulaması kullanılmıştır. Aralarında çok fark olmamakla birlikte AOA daha esnek bir yapıya sahip olduğu için tercih edilmektedir. Öte yandan AOA’da iki etkinlik aynı olayla sonuçlandığında boş olay kullanımı gerekebilmektedir. AOA zaman yönetiminde AON ise maliyet yönetiminde tercih edilmektedir. (Meredith & Samuel, 2000)

PERT/CPM yöntemini kullanarak bu projenin ne kadar sürede bitirilebileceğini hesaplayabiliriz. Verilen örnekteki işlerin toplam süresi 27 gündür. Fakat bütün işler sırasıyla yapılmak zorunda değildir ve doğru koordine edildiğinde kimi görevler paralel yürütülebilmektedir. Bunun için hatları birbirine bağlayan boğumları takip etmemiz gerekmektedir.

Örnek şebekeye süreleri dahil ederek aşağıdaki yapıya dönüştürebiliriz. Olaylara uzanan hatları ve hat süresince bulunan etkinliklerin süresini göz önüne alarak her bir hattın süresini hesaplayabiliriz. Bununla her bir olaya ulaşmak için harcanması gereken süreleri değerlendirebiliriz.



Olay	Etkinlik	Süre
1	-	-
2	A	4
3	B	3
4	A,C B,D	11 8
5	A,C,F B,D,F B,E	17 14 5

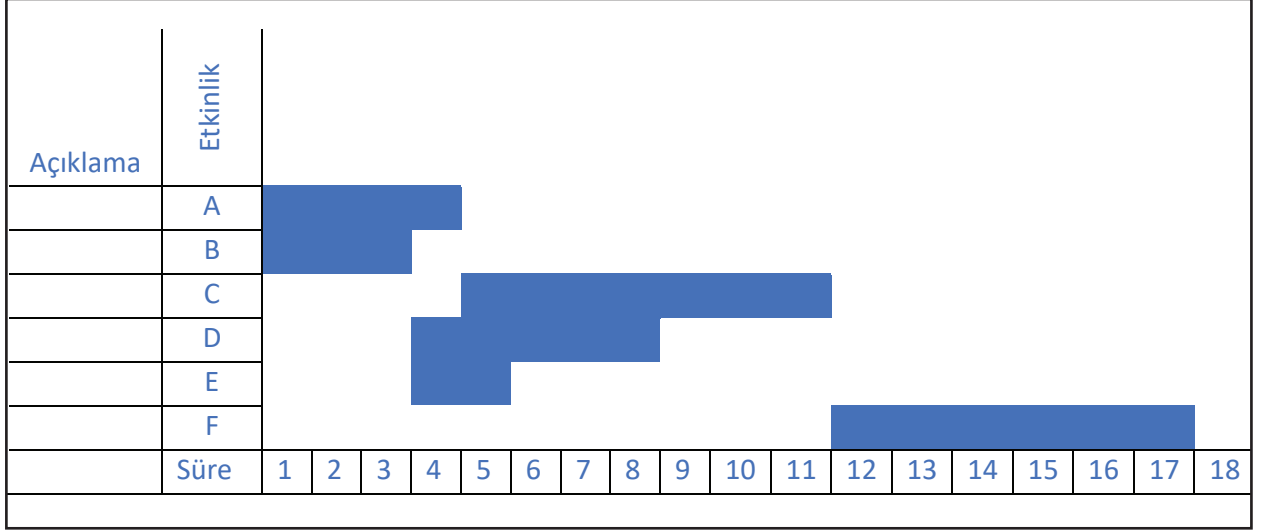
Tabloda görülebileceği gibi birinci olay başlangıç olduğu için herhangi bir zaman almamaktadır. İkinci olaya yalnız A etkinliği ile ulaşıldığından onun süresini göz önüne almamız yeterlidir. Olay 3 için de benzer bir durum B etkinliği bağlamında geçerlidir. Olay 4'e geldiğimizde iş biraz daha karmaşıklaşmaktadır. Olay 4'e sebep olan iki hat bulunmaktadır. Birisi A ve C etkinliklerinden oluşan ve 11 toplamda 11 gün süren hat. Diğeri ise B ve D etkinliklerinden oluşan ve 8 gün süren hat. Bu durumda olay 4'e ulaşmak için gerekli tüm etkinliklerin tamamlanmasının 11 gün süreceğini kestirebiliyoruz. Olay 5 son olay olarak tüm etkinliklerin tamamlanmasıyla ulaşılabilir. Bu olaya ulaşmak için A, C, F; B, D, F; ile B, E hatlarının tamamlanması gerekmektedir. Bu hatların sırasıyla 17 gün, 14 gün ve 5 gün süreceğini de ele alırsak projenin en erken 17 günde tamamlanabileceğini çıkartabiliriz.

Şebekenin yardımıyla toplamda 27 günlük bir iş yüküne sahip projeyi 17 günde tamamlayabileceğimizi görmüş olduk. Bunun yanında A, C, F hattının (kalın çizgiyle gösterilen) kritik hat olduğunu da görmüş oluyoruz. Bu etkinlikler tamamlanmadan projenin bitmesi mümkün olmadığı için proje yöneticisi bu hattın gecikme olmaksızın sürdürülmesine odaklanabilir. Ayrıca, kaynakları tahsis ederken diğer hatların boşluklarını değerlendirerek daha kolay eşgüdüm sağlayabilir.

6.4. Gantt Çizelgesi

Gantt çizelgesi en eski takvim ve halen kullanılan oluşturma yöntemidir. Görevleri yatay bir çizelgede bağımlılıklarına bağlı olarak sıralı sunmak-

tadır. Örnek projemizin görevlerini Gantt çizelgesine dökersek aşağıdaki yapıya ulaşırız.



Etkinlik A ve B herhangi bir halefe sahip olmadıkları için sıfır noktasından başlayabilmektedir. C etkinliğine başlanabilmesi için A etkinliğinin tamamlanması gerektiği için ancak A etkinliğinin tamamlandığı 4 noktasından başlanabilmektedir. Bu esnada B etkinliğini takip eden D ve E etkinlikleri devreye alınabilmektedir. PERT/CPM şebekesinde bahsedildiği gibi F etkinliğine başlanabilmesi için diğer etkinliklerin tamamlanması gerektiği için ancak 11 noktasından bu etkinliğe başlanabilmektedir. Projenin toplam süresi 17 gün olarak çizelgede gösterilmektedir.

Gantt çizelgesi yaygın bir kullanıma ve birçok çeşitlemeye sahiptir. Yaygın kullanımının nedeni Gantt çizelgesinin birçok bilgiyi barındıracak

biçimde kolay bir biçimde inşa edilip projeyi resmedebilmesidir. Proje güncellemeleri kolay bir biçimde yansıtılabilmektedir ve projenin gidişatı kolay bir biçimde gözlemlenebilmektedir.

Gantt çizelgesinin dezavantajlı olduğu noktalar etkinliklerin ilişkilerini ve değişkenliklerini ortaya koymakta zorlanmasıdır. Proje süresinin tayini ve çarpışma testleri gibi iyileştirme çalışmaları için Gantt çizelgesi uygun değildir. Projeyi bir bütün olarak gösteren Gantt çizelgesi kritik hattın göz önünde bulundurulmasını zorlaştırmaktadır. Çeşitli varyasyonlarla ve PERT/CPM şebekelerinin kullanımı ile bu dezavantajların önüne geçilmektedir.

BÖLÜM ÖZETİ

Bir amacı gerçekleştirmek için var olan ve birbirleri ile etkileşimde bulunan parçalardan oluşan bütünlere sistem adı verilmektedir. Sistemlerin varlığının temel nedeninin bütünü parçalarından daha fazlası olması olarak gösterebiliriz. Sistem yaklaşımı, insanların çevrede bulunan olgu ve faaliyetleri ilişkilendirmesine yardımcı olmaktadır. Sistemlerin, bir doğası ve varlık nedeni bulunmaktadır. Bu doğayı ve varlık nedenini dış etkenlere karşı sürdürdüğünü görmekteyiz. Bilinçli sistemlerde ise varlık amacını gerçekleştirmek için fark edilir bir çaba gözlemlenmektedir. Amaç, sistemin varlık nedenini oluşturmaktadır. Bu sistemlerin iç dengesini koruma, geri bildirim yapma ve kendi kendini düzeltme özelliğinden kaynaklanmaktadır.

Proje, belirli bir sorunu çözmek üzere özgün olarak oluşturulan sistem şeklinde tanımlanabilir. Projeler benzersiz bir ürün ya da hizmet sağlamak üzere girişilen süreli uğraşlar olarak da tanımlanmaktadır. Projelerin belirli bir zaman ve maliyet kısıtı bulunmaktadır. Bu kısıtlar dâhilinde bir hedefe yönelik tasarlanan proje gerçekleştirilmesi ile son bulur. Projeler odaklı ve özgün çözümler getirerek çeşitli avantajlar doğurmaktadır. Diğer iş yapış biçimlerinden farklılıklar barındıran projelerin özellikleri şunlardır: Projeler bir kapsam, süre ve bütçe kısıdına sahiptir. Projeler birden fazla işlev ya da örgütün katılımıyla oluşturulan takımlar tarafından gerçekleştirilir. Projeler özgün, bir kerelik, sıra dışı görevlerden oluşur. Projeler beşerî, aynı ve mâli kaynaklar kullanılarak tamamlanır. Projeler kendine özel örgütlenme, yönetim ve bilgi işlem tekniklerine sahiptir.

Projeler amaçlarına göre çeşitlerine ayrılmaktadır. Bunlar; İş organizasyonu ve yönetim projeleri, nesneye ilişkin projeler ve kurulum projeleridir. İş organizasyonu ve yönetim projeleri iş yapım projeleri olarak da adlandırılmaktadır. Bu tip projelere örnek olarak bilgisayar destekli yönetim ve bilişim sistemlerinin oluşturulması, üretim planlama ve kurulumu, yeniden örgütlenme faaliyetleri verilebilir. Nesneye ilişkin projeler ve kurulum projeleri ise genellikle birden fazla örgütün dâhil olduğu projeler olarak karşımıza çıkmaktadır. Bu projeler ürün odaklı teknik projelerdir. Bunlara örnek olarak araştırma geliştirme faaliyetleri, yeni bir ürünün tasarımı ya da piyasa araştırmaları verilebilir.

Projeler birden fazla işlevin yönetimini içerir. Geleneksel yönetim anlayışında işlevlerin o alanda uzman bir yönetici tarafından yönetimi esastır. Projeler ise birden fazla işlevin katkısıyla tamamlanabilmektedir. Bu nedenle birden fazla alandan gelen çalışanların koordine edilmesi ve uyumlaştırılması esastır. Proje yönetimi tüm işletmeler için çeşitli avantajlar sağlamaktadır. Bu avantajlar; geliştirme süreçlerinde ekonomiklik, kaynakların verimli kullanımı, etkin kontrol, kalite ve güvenlik temini, eşgüdüm ve motivasyonda iyileşme ve müşteriler ve diğer dış paydaşlarla daha etkili etkileşimdir.

GÖZDEN GEÇİRELİM

1. Projeleri bir sistem olarak değerlendirdiğimizde etkinlikler ne isim almaktadır?
 - a. Süreç
 - b. Olay
 - c. Başarı
 - d. Analiz
 - e. Yönetim
2. Projelerde hangi örgüt tipi benimsenmektedir?
 - a. Hiyerarşik örgüt
 - b. Coğrafi örgüt
 - c. Müşteri temelli örgüt
 - d. Matris örgüt
 - e. Ürün temelli örgüt
3. İş organizasyonu ve yönetim projelerine aşağıdakilerden hangisi verilebilir?
 - a. İnşaat projeleri
 - b. Askeri tatbikatlar
 - c. İşletme kurulumu
 - d. Yönetim bilişim sisteminin oluşturulması
 - e. Araştırma çalışmaları
4. Proje yöneticisinin proje ekibini oluşturması, koordine etmesi ve izlemesi projenin hangi aşamasında gerçekleşir?
 - a. Planlama
 - b. Tanımlama
 - c. Kavramsallaştırma
 - d. Kontrol
 - e. Tamamlama
5. Bir projeden beklenen performans hangi proje hedefi ile doğrudan ilişkilidir?
 - a. Proje süresi
 - b. Proje maliyeti
 - c. Proje kapsamı
 - d. Proje yönetimi
 - e. Proje örgütü
6. Bir projenin yaşam döngüsü boyunca birim zamanda en yüksek harcama hangi dönemde yapılmaktadır?
 - a. Kavramsallaştırma
 - b. Planlama
 - c. Gerçekleştirme
 - d. Sonlandırma
 - e. Tanımlama
7. Proje yönetimi ve geleneksel yönetim arasındaki farklılıklardan birisi hangi seçenekte verilmektedir?
 - a. Proje yönetiminde katılımcı, geleneksel yönetiminde hiyerarşik yönetim anlayışı yer almaktadır
 - b. Geleneksel yönetim işlevleri proje yönetiminde yer almaz
 - c. Proje yönetimi kontrole sahipken geleneksel yönetim bundan yoksundur
 - d. Geleneksel yönetim bir örgüte hitap ederken proje yönetiminde örgüt yer almaz
 - e. Geleneksel yönetimde teknik uzmanlık gerekirken proje yönetimi teknik uzmanlık aranmaz

8. Bir projede aynı zaman diliminde yapılabilecek projeler hangi seçenekte verilmektedir?

- a. Sıralı görevler
- b. Aynı zaman diliminde yapılabilecek görevler
- c. Bir görevin halefi olan görev
- d. Bir görevin selefi olan görev
- e. Farklı zaman diliminde yapılması gereken görevler

10. Bir yönetim bilişim sisteminin oluşturulması hangi tür projeye örnek olarak verilebilir?

- a. Kurulum projeleri
- b. Nesneye ilişkin projeler
- c. İş organizasyonu ve yönetim projeleri
- d. Kapalı projeler
- e. Eşik değerli proje

9. Bir projenin görevleri şu şekilde verilmektedir: Etkinlik A 6 günde, etkinlik B 7 günde, etkinlik C 4 günde tamamlanmaktadır. Etkinlik A halef, etkinlik B bağımsız, etkinlik C ise A selefi durumundadır. Bu projenin tamamlanma süresi ne kadardır?

- a. 10 gün
- b. 11 gün
- c. 13 gün
- d. 15 gün
- e. 17 gün

Yanıt Anahtarı: 1-A, 2-D, 3-D, 4-D, 5-C, 6-C, 7-A, 8-B, 9-A, 10-C

Kaynakça

- Albayrak, B. (2005). *Proje Yönetimi*. Nobel Yayın Dağıtım.
- Frederick, W. T. (1910). *The Principles of Scientific Management*. 7.
- Meredith, J. R., & Samuel, J. M., Jr. (2000). *Project Management A Managerial Approach* (4. bs). John Wiley & Sons.
- Pinto, J. K., & Slavin, D. P. (1988). Critical success factors across the project life cycle definitions and measurement techniques. *Project Management Journal*, 19(3), 67-75.
- Sarıaslan, H. (1984). Sistem Analizinin Temelleri. *Ankara Üniversitesi SBF Dergisi*, 39(1), 1. https://doi.org/10.1501/SBFder_0000001447

BÖLÜM 3

BİLGİ SİSTEMLERİNİN GELİŞTİRİLMESİ

ANAHTAR KAVRAMLAR

Yazılım Geliştirme Yaşam Döngüsü, Doğrusal Ardişik (Şelale) Model, V Modeli, Prototip Model, Spiral Model, Aşırı Programlama Modeli, Scrum Modeli

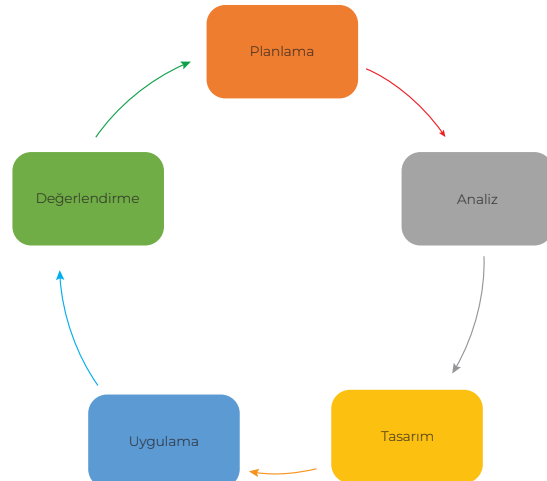
ÖĞRENME HEDEFLERİ

- 1 Sistematik yazılım geliştirme hakkında bilgi sahibi olacaksınız
- 2 Yazılım Geliştirme Yaşam Döngüsünü ve aşamalarını açıklayabileceksiniz
- 3 Yazılım Geliştirme Modelleri hakkında bilgi sahibi olacaksınız
- 4 Geleneksel ve Çevik yazılım geliştirme modelleri hakkında bilgi sahibi olacaksınız

GİRİŞ

Günümüzde bilgi sistemleri eğitim, sağlık, finans, seyahat ve eğlence gibi yaşamımızın neredeyse tüm alanlarında kullandığımız sistemlerdir. Bilgi sistemleri sayesinde birçok iş ve işlem zahmetsiz ve hızlı bir şekilde tamamlanabilmektedir. Davis (1999, s. 3) bilgi sistemini, doğru zamanda, doğru kişiye, doğru veri ve bilgiyi sağlamayı amaçlayan donanım, yazılım, veri, insan ve prosedürel bileşenler kümesi olarak tanımlamaktadır. Tüm sistemlerin geliştirilme sürecinde olduğu gibi bilgi sistemlerinin geliştirilmesinde de birçok farklı model rehber alınmaktadır. Bu bölümde bilgi sistemlerinin geliştirilmesinde kullanılan modeller hakkında bilgi verilecektir.

Sarıdoğan (2008, s. 63) her sistemin uygun bir yaşam döngüsü (Life Cycle) metodu kullanılarak proje şeklinde gerçekleştirilmesinin etkinlik açısından çok önemli olduğu vurgulamaktadır. Sistematik olarak geliştirilen projelerin geçirdiği süreçleri açıklayan birçok modelin ortak noktalarını tanımlamak için sıkça “Sistem Geliştirme Yaşam Döngüsü (System Development Life Cycle-SDLC)” kavramının kullanıldığı görülmektedir. Bu ortak model temel olarak planlama, analiz, tasarım, uygulama ve değerlendirme adımlarını içermekte ve bir döngü şeklinde uygulanmaktadır. SDLC Şekil 1’de görsel olarak sunulmuştur.



Şekil 1- Sistem Geliştirme Yaşam Döngüsü (SDLC)

1. SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ

1. Adım: Planlama

Planlama SDLC' nin ilk adımıdır. Bu adımda; projeye neden ihtiyaç duyulduğu, ne kadar zamanda tamamlanması gerektiği, ne kadar bütçe gerektiği ve projenin kazanımlarının neler olacağı gibi soruların yanıtlanması gerekmektedir. Planlama adımının en kritik yanının projeye devam edip edilmeyeceğine dair kararın çoğunlukla bu aşamanın sonunda verilmesi olduğu düşünülmektedir. Bu adımda alınacak kararlar projeden vazgeçilebilir, uygulanabilir veya genişletilerek uygulanabilir. Silahtaroglu (2010, s. 31) bu adımda yapılacak işlemleri beş başlıkta toplamaktadır. Bunlar:

1. Problemin tanımlanması,
2. Fizibilite raporlarının hazırlanması,
3. Proje zaman çizelgelerinin hazırlanması,
4. Projede çalışacak personelin zamana bağlı olarak belirlenmesi ve
5. Projenin başlatılmasıdır.

Planlama adımının doğru olarak ele alınması projenin başarıya ulaşması açısından çok önemlidir. Çünkü hatalı hesaplara başlayan bir işin doğru sonuçlandırılması olası olmadığı düşünülmektedir. Bu adımın sonunda yapılan işlemler raporlanır ve projeye devam kararı verilirse ikinci adıma geçilir.

2. Adım: Analiz

Analiz adımında projenin ihtiyaçları belirlenmeye çalışılmaktadır. Sistemin girdi ve çıktıların neler olacağı, kimler tarafından kullanılacağı, kullanıcıların beklentilerinin neler olduğu gibi soruların cevaplanması gerekmektedir. Bu tip soruları cevaplamak için de bilgi toplanması, tanımlar yapılması, örnek senaryolar hazırlanması, alternatif çözüm önerilerinin getirilmesi ve tüm bu verilerin değerlendirilmesi gerekmektedir. Bu adımda da yapılan tüm işlemler raporlanarak

kayıt altına alınmaktadır. Foster (2022, ss. 90-91) yazılımsal bir sistem geliştirilirken bu adımda yapılması gerekenleri şu şekilde sıralamaktadır:

- a) Katkı sağlayacak kişi veya kurumların tanınması,
- b) Problemin özetlenmesi,
- c) Sistemin genel durumunun çıkarılması,
- d) Sistem bileşenlerinin belirlenmesi,
- e) Gereksinimlerin detaylı olarak belirtilmesi,
- f) Verilerin nasıl depolanacağını belirlenmesi,
- g) İşletim için gereksinimlerin belirtilmesi ve
- h) Sistemin kurallarının tanımlanması.

3. Adım: Tasarım

Bu adım, analizde elde edilen veriler temelinde geliştirilecek sistemin tasarlandığı adımdır. Bu adımda sistemin nasıl görüneceği ve nasıl çalışacağına dair eskizler hazırlanmaktadır. Kullanıcı ve sistem arayüzünün nasıl olacağı, bileşenlerin birbiriyle nasıl etkileşime gireceği, veritabanı ve sistemle nasıl entegre olacağı, sistem kontrollerinin nasıl yapılacağı gibi başlıklar tasarlanıp görüştürilmektedir.

Yapılan tasarımda en çok üzerinde durulan konuların, sistemin tutarlı ve bütünlüğe sahip olması olduğu belirtilmektedir (Tsui vd., 2017, s. 170). Tutarlı bir sistem tasarımı ekranlarda, raporlarda, veritabanı öğelerinde ve süreç mantığında ortak terminolojinin kullanılmasını sağlamaktadır. Bütünlüğe sahip bir tasarım ise tüm gereksinimlerin tasarlandığı ve hiçbirinin dışarıda bırakılmadığı tam bir yapı sunmaktadır.

4. Adım: Uygulama

Uygulama adımında şekillenen sistemi kurmak hedeflenmekte ve tasarım adımında belir-

lenen arayüzler, ilişkiler, veritabanı ve kontroller gerçekleştirilmektedir. Bu adımda sistem belirlenen yazılım dilinde kodlanır ve veritabanı yönetim sistemleri işe koşulur. Uygulama adımının başarıya ulaşması için geliştirilen sistemin şu özellikleri taşıması gerektiği belirtilmektedir (Tsui vd., 2017, s. 192):

- a) Okunabilirlik: Kodlar başka programcılar tarafından da anlaşılabilirliklidir.
- b) Sürdürülebilirlik: Kodlar kolayca değiştirilebilir ve bakımı yapılabilirliklidir.
- c) Performans: Diğer her şey sabitken mümkün olduğu kadar hızlı çalışan kod üretmelidir.
- d) İzlenebilirlik: Tüm kod öğeleri bir tasarım ögesine karşılık gelmelidir. Kod, tasarıma kadar izlenebilir olmalıdır.
- e) Doğruluk: Sistem, analiz ve tasarım adımlarında amaçlanan şeyi yapmalıdır.
- f) Bütünlük: Tüm sistem gereksinimleri karşılanmalıdır.

2. YAZILIM GELİŞTİRME MODELLERİ

Sistem Geliştirme Yaşam Döngüsünde bahsedilen adımların özellikle yazılım sistemi geliştirmek için tasarlanmış birçok farklı model bulunmaktadır. Bu modellerin amacı yazılım sistemi geliştirilirken zaman, para, işgücü gibi kaynakların doğru kullanımını sağlamak ve elde edilen ürünün hem üretici hem de müşteri için

5. Adım: Değerlendirme

Bu adım bakım (Silahtaroglu, 2010), destek (Saridoğan, 2008), entegrasyon, test ve değerlendirme (Wasson, 2006) gibi farklı isimlerle de anılmaktadır. Değerlendirme adımının amacı ortaya konulan sistemin doğru çalıştığından emin olmak ve varsa eksik ya da hatalı yönlerinin tespit edilerek düzeltilmesini sağlamaktır. Bu adımda elde edilen veriler sistemin yeniden ele alınması için birinci adımdan itibaren girdi olarak kullanılabilirliktedir. Bunun yanında başka adımlara da doğrudan veri sağlayıp, oradaki eksikliklerin giderilmesinde de kullanılabilirliktedir. Bu adımda yapılan işlemler şu şekilde özetlenebilir:

- a) Kullanıcı desteği verilmesi,
- b) Hataların giderilmesi,
- c) Versiyon veya sürüm yükseltmelerinin yapılması,
- d) Mevzuat değişikliklerinin sisteme entegre edilmesi,
- e) Teknoloji değişikliklerinin yapılması.

tatmin edici olmasını garanti etmektir. Bu başlık altında yazılım geliştirme modelleri iki kategoride ele alınmakta ve yaygın kullanılan modeller tanıtılmaktadır.

- A. Geleneksel Yazılım Geliştirme Modelleri
- B. Çevik (Agile) Yazılım Geliştirme Modelleri.

A. Geleneksel Yazılım Geliştirme Modelleri

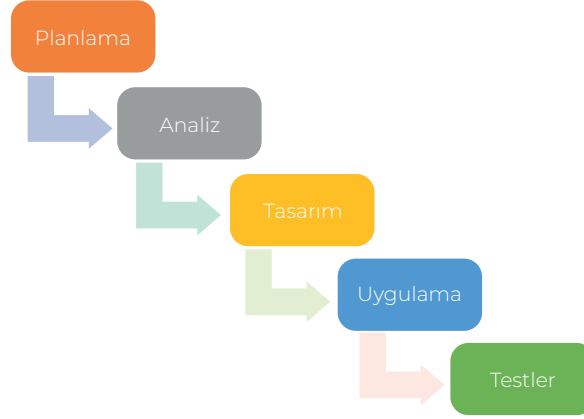
Bu kategoride yer alan yazılım geliştirme modellerinin, çoğunlukla işletmeler için kullanılan sistem geliştirme adımlarının, yazılım geliştirme adımlarına uyarlanması sonucu ortaya çıktığı görülmektedir. Bu sebeple kategori adı Geleneksel Yazılım Geliştirme Modelleri olarak seçilmiştir. Bu başlıkta tanıtılacak modeller şunlardır:

- 1. Doğrusal Ardışık (Şelale) Model
- 2. V Modeli
- 3. Prototip Model
- 4. Spiral Model

1. Doğrusal Ardışık (Şelale) Model

Doğrusal Ardışık Model (DAM) katı bir dokümantasyon ve sürece sahiptir. Her aşaması sıra ile takip edilir, biri tamamlanmadan diğerine geçilmez ve biri tamamlandıktan sonra önceki aş-

malara dönülmez. Bu sebeple hedefleri çok net bir şekilde tanımlanmış ve süreç içerisinde değişiklik beklenmeyen projeler için uygulanabilir. DAM beş aşamadan oluşmaktadır ve bunlar Şekil 2’de gösterilmiştir.



Şekil 2- Doğrusal Ardışık (Şelale) Model

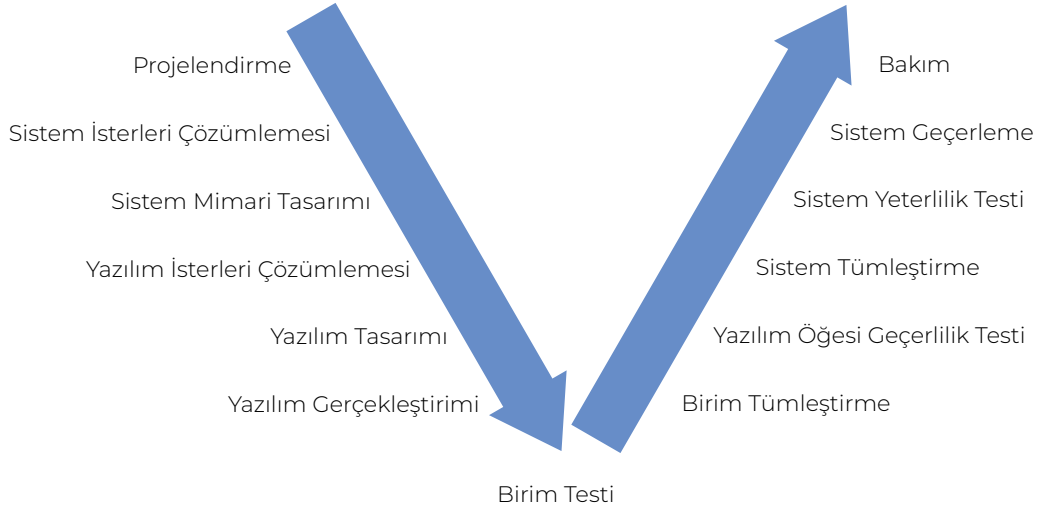
DAM Planlama aşaması ile başlar. Bu aşamada projenin bütçesi, süresi, ve paydaşları gibi hususlar belirlenir ve Analiz aşamasına geçilir. Analiz aşaması genel olarak durum ve ihtiyaç analizi olmak üzere ikiye ayrılır. Durum analizinde var olan kaynaklar ortaya konur ve yazılım geliştirme ortamına karar verilir, ihtiyaç analizinde ise projeden ne beklendiği açıklanır ve bu bilgilere göre Tasarım aşamasına başlanır. Tasarım aşaması en detaylı çalışmanın yapıldığı aşamadır. Bu aşamada yazılım ekranları, akış diyagramları, veri tabanı gibi yazılım bileşenlerinin detaylı tasarımları yapılır. Uygulama aşaması, tasarım aşamasında karar verilen tasarımların kodlara döküldüğü aşamadır. Yazılımın kendisi önceki aşamalarda verilen kararlar doğrultusunda kodlanır. Ardından son aşama olan Testler aşamasında elde edilen yazılımın varsa hataları tespit edilerek giderilir ve proje tamamlanır. Her aşamada

yapılan işlemler detaylı olarak yazılır ve rapor haline getirilir. (Şeker, 2015)

2. V Modeli

V modeli süreç gösteriminden dolayı bu ismi almaktadır. V şeklinin sol tarafı üretimi, sağ tarafı ise test işlemlerini göstermek için kullanılmaktadır. Modelin aşamaları Şekil 3’te gösterilmektedir.

V Modelinde sağ tarafta yer alan süreç, DAM başlığında değindiğimiz Planlama, Analiz ve Tasarım aşamalarında yapılan işlemlerin, yazılım projelerine özel olarak isimlendirilmiş aşamalarından oluşmaktadır. Model görselinin sol tarafı ise test aşamalarından oluşmaktadır. Son aşama olan Bakım ise test aşamalarında elde edilen veriler ışığında yazılım projesinde yapılması gereken iş ve işlemlerin yapıldığı aşamadır.



Şekil 3- V Modeli

3. Prototip Model

Prototip Model, proje paydaşlarının bir araya gelerek yazılımın girdi ve çıktılarına, verilerin nasıl yönetileceğine ve korunacağına karar vermesiyle başlamaktadır. Bu toplantı sonucunda verilen kararlar hızlıca yapılmış bir tasarıma uygulanarak yazılımın ilk örneği elde edilir. Bu ilk örnek “prototip” olarak isimlendirilir ve kullanıcı-

cılara sunulur. Kullanıcı deneyim ve değerlendirmelerine göre prototip üzerinde değişiklikler yapılır. Prototipin yeni hali de kullanıcılara sunulularak yazılımın beklentileri karşılayacak işlevlere sahip ilk sürümü geliştirilmiş olur. Prototip Modelin simgesel gösterimi Şekil 4’te gösterilmektedir.



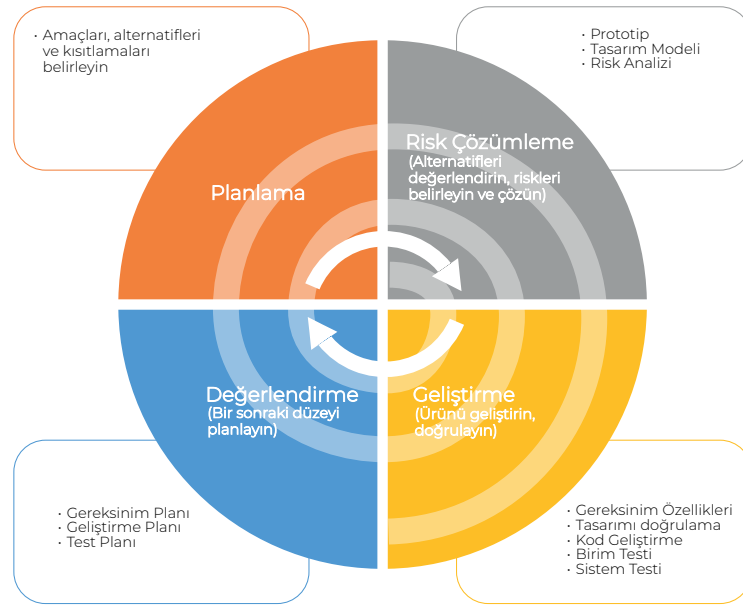
Şekil 4- Prototip Model

Prototip geliştirilmesi esnasında yazılımın gereksinimleri karşıladığı kabul edilene kadar müşteri değerlendirmesine sunulur. Prototip geliştirme tamamlandıktan sonra yazılıma yönelik testler uygulanır ürün tamamlanır. Bu modeli özel yapan nokta müşteri ile geliştirilmesidir.

4. Spiral Model

Spiral model, yazılım geliştirmedeki risklerin azaltılmasına vurgu yapmaktadır. Bu nedenle

model, yazılım sürecine risk odaklı bir yaklaşım sergilemektedir. Model proje geliştirme döngülerinden geçerek, proje riskini azaltırken yazılım sistemini aşamalı olarak geliştirmek için döngüsel bir yaklaşım sağlamaktadır. Spiral modelin dört çeyreği vardır ve yazılım projesi aşamalı olarak döngüsel geçiş sergiler. Spiral Modelin simgesel gösterimi Şekil 5’te gösterilmektedir.



Şekil 5- Spiral Model

Şekil 5'te gösterildiği gibi, spiral yol çok düzgün olmayabilir. Her döngü, endişelerin, bileşenlerin veya yapıların her biri için aynı adım dizisini içerir. Yazılım geliştirme ve yazılım geliştirme projelerine eşit derecede uygulanabilir olan spiral model, hedef odaklıdır. Spiral süreç, nihai sonuca ulaşılan veya ulaşılamaz olduğu gösterilene kadar, bu hedefin veya gereksinimin sürekli olarak test edilmesini veya yinelenmesini içerir. Dört kadrandan tipik bir geçiş aşağıdaki gibidir:

1. Spiralin her bir döngüsü için hedefleri, alternatifleri veya kısıtlamaları tanımlanmalıdır.
2. Hedeflere ve kısıtlamalara göre alternatifler değerlendirilmelidir.

3. Tanımlanan risklerin miktarına ve türüne bağlı olarak, bir prototip geliştirilmeli, daha ayrıntılı bir değerlendirme yapılmalı, evrimsel bir gelişme veya belirlenen hedefe ulaşma riskini daha da azaltmak için gerekirse başka bir adım geliştirilmelidir. Öte yandan, risk önemli ölçüde azaltılırsa, bir sonraki adımda gereksinimler, tasarım veya kodlama gibi bir görev belirlenmelidir.

4. Hedefe ulaşıldığı doğrulanmalı ve bir sonraki döngü için plan yapılmalıdır. (Abrahamsson vd., 2002)

B. Çevik Yazılım Geliştirme Modelleri

Bu kategoride yer alan yazılım geliştirme modellerinde yazılım geliştirme süreçlerinin ihtiyaçlarının daha çok dikkate alındığı görülmektedir. Çevik yazılım geliştirme modelleri ayrıca yeni nesil yazılım geliştirme modelleri olarak da adlandırılmaktadır (Saridoğan, 2008). Birçok çevik yazılım geliştirme modeli olmakla birlikte bu başlıkta tanıtılacak modeller şunlardır:

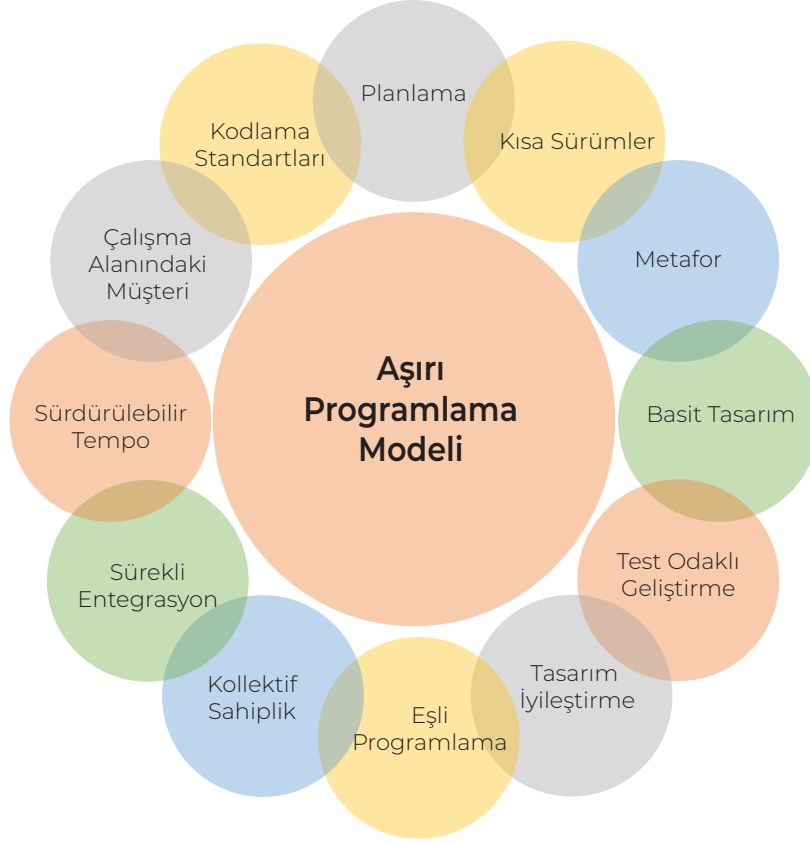
1. Aşırı Programlama Modeli
2. SCRUM Modeli

1. Aşırı Programlama Modeli

Aşırı Programlama (Extreme Programming) geleneksel modellerin uzun geliştirme döngülerinin yarattığı sorunlara bir çözüm getirmek amacıyla yapılan başarılı uygulamalardan doğmuş bir model olarak sunulmaktadır. Aşırı programlama modelinin bireysel uygulamalarının yeni olmadığı belirtilmektedir. Başarılı bir dizi uygulamanın adımları işlevsel olacak şekilde bir araya getirilerek yazılım üreticileri için yeni bir

model olarak sunulmuştur (Abrahamsson vd., 2002, s. 18). Aşırı Programlama Modeli kullanılırken ekip üyeleri ve müşteri arasında sık iletişim kurulmalı, tasarım ve kodda sadelik esas olmalı, birçok farklı düzeyde geri bildirim sağlanmalı

ve proje için uygun görünmüyorsa bu modeli kullanmaktan kaçınılmalıdır. Aşırı Programlama Modelinin 12 anahtar kavramı bulunmaktadır (Abrahamsson vd., 2002) ve bunlar Şekil 6'da gösterilmektedir.



Şekil 6- Aşırı Programlama Modelinin Anahtar Kavramları

Aşırı Programlama Metodu temel olarak üç ana adımdan oluşur (Saridoğan, 2008). Kullanıcıdan alınan bilgiler doğrultusunda gereksinimler belirlenir ve ilk aşama olan “Sürüm Planlaması” yapılır. Bu aşamada yazılıma yönelik kestirimler ortaya çıkar ve bu öngörülere yönelik araştırma yapılarak bu kestirimler doğrulanmaya çalışılır. Aynı zamanda yazılım projesinin mimarisi de ortaya konur.

İlk aşamadan elde edilen verilere göre ilerleme hızı sürüm planı ve ek kullanıcı bilgileri ile “Yineleme” aşamasına geçilir. İkinci aşama kodlamanın yapıldığı aşamadır ve sürekli olarak ilk aşamaya dönülerek yazılımda yapılması gereken değişikliklere yönelik olarak Sürüm Planlaması aşamasında da değişiklikler yapılır.

İkinci aşamanın tamamlanmasının ardından “Kabul Testleri” aşamasına geçilir. Üçüncü aşamada yapılan testler sonucu bulunan hatalar veya eksikler ışığında Yineleme aşamasına dönülerek gerekli düzeltmeler yapılır ve düzeltilen yazılım yeniden üçüncü aşamadaki testlere tabi tutulur. Bu aşamada ilk aşamaya girdi olarak alınan kullanıcı gereksinimleri karşılanıp karşılanmadığı da denetlenmektedir. Nihai olarak müşteri onayı alındıktan sonra sürüm kullanılmaya başlanır. Aşırı Programlama Modelinin simgesel gösterimi Şekil 7’de verilmektedir.



Şekil 7- Aşırı Programlama Modeli

2. SCRUM Modeli

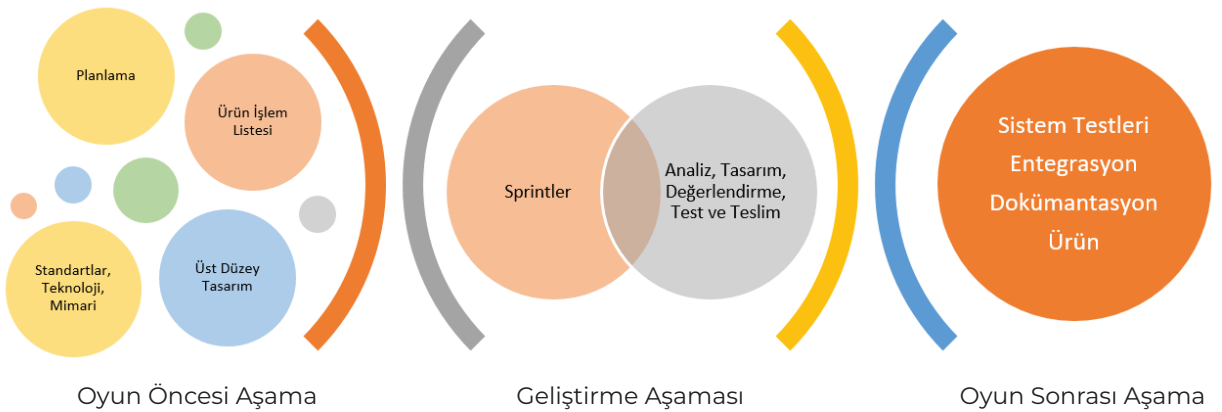
Scrum Modeli takım rolleri, olaylar, ürünler ve kurallardan oluşan bileşenlere sahiptir. Bu modelle geliştirilecek bir yazılımda, bileşenlerin tümünü kullanmak gerekir, ancak başka teknikler veya bileşenler de eklenebilmektedir. Scrum Modeli yinelemeli bir yöntemdir ve sprint adı verilen kısa yinelemelere dayanır. Sprint'ler, toplam yazılım ürününün işlevlerine yönelik küçük geliştirme birimleridir. Süreleri kısadır (bir ay veya daha az) ve zaman sınırlıdır (süreleri sabit tutulur, ancak gerekirse kapsam ayarlanabilir). Modelin ideal uygulamalarında, her sprint sonunda potansiyel olarak kullanılabilir bir ürün ortaya çıkar. Scrum Modeli ile geliştirilen bir yazılım projesi paydaşları için üç temel rol tanımlanmakta ve ihtiyaç duyulduğunda bu rollere başka rollerin de eklenebileceği belirtilmektedir (Abrahamsson vd., 2002). Bu roller:

1. Müşteriyi temsil eden ve ekibin işletmeye değer katmasını sağlayan ürün sahibi,
2. Yazılımı üreten, genellikle üç ila on geliştiriciden oluşan geliştirme ekibi,
3. Bir proje lideri olarak değil, bir koç olarak ekibi yönlendiren ve Scrum'un takip edilmesini sağlayan bir Scrum Üstadı.

Bunlara ek olarak paydaşlar ve yöneticiler yardımcı rol olarak tanımlanabilir. Scrum, üç temel direk üzerine kurgulanmalıdır, bunlar:

1. Şeffaflık (süreci görünür kılmak),
2. Denetleme (ürün ve ilerlemenin)
3. Adaptasyon (önemli bir sapma tespit edildiğinde, düzeltin).

Scrum Modeli Oyun Öncesi, Geliştirme ve Oyun Sonrası olmak üzere üç ana aşamadan oluşmaktadır (Abrahamsson vd., 2002). Aşamaların simgesel gösterimi Şekil 8'de verilmiştir.



Şekil 8- Scrum Modeli

BÖLÜM ÖZETİ

Günümüzde bilgi sistemleri; eğitim, sağlık, finans, seyahat ve eğlence gibi yaşantımızın neredeyse tüm alanlarında kullandığımız sistemlerdir. Bilgi sistemleri sayesinde birçok iş ve işlem zahmetsiz ve hızlı bir şekilde tamamlanabilmektedir. Bilgi sistemi; doğru zamanda, doğru kişiye, doğru veri ve bilgiyi sağlamayı amaçlayan donanım, yazılım, veri, insan ve prosedürel bileşenler kümesi olarak tanımlanmaktadır.

Sistematik olarak geliştirilen projelerin geçirdiği süreçleri açıklayan birçok modelin ortak noktalarını tanımlamak için sıkça “Sistem Geliştirme Yaşam Döngüsü” kavramının kullanıldığı görülmektedir. Bu ortak model temel olarak planlama, analiz, tasarım, uygulama ve değerlendirme adımlarını içermekte ve bir döngü şeklinde uygulanmaktadır.

Planlamada projeye neden ihtiyaç duyulduğu, ne kadar zamanda tamamlanması gerektiği, ne kadar bütçe gerektiği ve projenin kazanımlarının neler olacağı gibi soruların yanıtlanması gerekmektedir. Planlama adımının en kritik yanının projeye devam edip edilmeyeceğine dair kararın çoğunlukla bu aşamanın sonunda verilmesi olduğu düşünülmektedir.

Analiz adımında projenin ihtiyaçları belirlenmeye çalışılmaktadır. Sistemin girdi ve çıktıların neler olacağı, kimler tarafından kullanılacağı, kullanıcıların beklentilerinin neler olduğu gibi soruların cevaplanması gerekmektedir. Bu tip soruları cevaplamak için de bilgi toplanması, tanımlar yapılması, örnek senaryolar hazırlanması, alternatif çözüm önerilerinin getirilmesi ve tüm bu verilerin değerlendirilmesi gerekmektedir.

Tasarım; analizde elde edilen veriler temelinde geliştirilecek sistemin tasarlandığı adımdır. Bu adımda sistemin nasıl görüneceği ve nasıl çalışacağına dair eskizler hazırlanmaktadır. Kullanıcı ve sistem arayüzünün nasıl olacağı, bileşenlerin birbiriyle nasıl etkileşime gireceği, veritabanı ve sistemle nasıl entegre olacağı, sistem kontrollerinin nasıl yapılacağı gibi başlıklar tasarlanıp görselleştirilmektedir.

Uygulama adımında şekillenen sistemi kurmak hedeflenmekte ve tasarım adımında belirlenen arayüzler, ilişkiler, veritabanı ve kontroller gerçekleştirilmektedir. Bu adımda sistem belirlenen yazılım dilinde kodlanır ve veritabanı yönetim sistemleri işe koşulur.

Değerlendirme; destek, entegrasyon ve test gibi farklı isimlerle de anılmaktadır. Değerlendirme adımının amacı ortaya konulan sistemin doğru çalıştığından emin olmak ve varsa eksik ya da hatalı yönlerinin tespit edilerek düzeltilmesini sağlamaktır. Bu adımda elde edilen veriler sistemin yeniden ele alınması için birinci adımdan itibaren girdi olarak kullanılabilir. Bunun yanında başka adımlara da doğrudan veri sağlayıp, oradaki eksikliklerin giderilmesinde de kullanılabilir.

Kaynakça

- Abrahamsson, Pekka., Salo, Outi., Ronkainen, Jussi., & Warsta, J. (2002). Agile Software Development Methods : Review and Analysis. VTT.
- Davis, W. S., & Davis, W. S. (Ed.). (1999). The Information System Consultant's Handbook. CRC Press.
- Foster, E. C. (2022). Software Engineering; A Methodical Approach (2nd bs). Taylor & Francis.
- Sarıdoğan, E. (2008). Yazılım Mühendisliği (2nd bs). Papatya Yayıncılık Eğitim.
- Şeker, S. E. (2015). Yazılım Geliştirme Modelleri ve Sistem/Yazılım Yaşam Döngüsü. YBS Asiklopedisi, 2(3). <https://ybsansiklopedi.com/>
- Silahtaroglu, G. (2010). Bilgisayar ve Yazılım Mühendisliğinde Sistem Analizi ve Tasarımı. Papatya Yayıncılık Eğitim.
- Tsui, F., Karam, O., & Bernal, B. (2017). Essentials of Software Engineering (4th bs). Jones & Bartlett Learning.
- Wasson, C. S. (2006). System Analysis, Design, and Development : Concepts, Principles, and Practices. John Wiley & Sons.

BÖLÜM 4

SİSTEM GEREKSİNİMLERİNİN BELİRLENMESİ

ANAHTAR KAVRAMLAR

Sistem Gereksinimi, Gereksinim Çeşitleri,
Gereksinimin Analizi, Gereksinimlerin
Yönetimi, Olguları Belirleme Teknikleri



ÖĞRENME HEDEFLERİ

- 1 Projenin sistem gereksinimlerine ilişkin kavramları açıklar
- 2 Projenin sistem gereksinimlerine ilişkin süreç adımlarını kavrar
- 3 Gerçekleri tespit etme tekniklerini sıralar

GİRİŞ

Sistem analizi önemli miktarda çaba ve maliyet gerektirir ve bu nedenle ancak

yönetim söz konusu sistem geliştirme projesinin değerli olduğuna ve bu aşamadan geçmesi gerektiğine karar verdikten sonra gerçekleştirilir. Çoğu gözlemci, geliştirilen sistemlerdeki hataların çoğunun yaşam döngüsünün analiz ve tasarım aşamalarındaki yetersiz çabalardan kaynaklandığı konusunda hemfikirdir. Sektörde yapılan araştırmalar, sistem sorunlarının %56'sının yetersiz gereksinim tanımlamasından, %7'sinin ise yetersiz kodlamadan kaynaklandığını göstermektedir. Bakım alanında, yetersiz kodlama için harcanan çabanın %1'ine karşılık, çabanın %82'si yetersiz gereksinimlerden kaynaklanmaktadır (Matthews, 1995).

Sistemle ilgili problem analizi yapıldıktan hemen sonra sistem analistinin, sistemin ne yapması gerektiğiyle ilgili gereksinimleri ortaya koyarak, hangi işlevlerin gerçekleştirileceğiyle ilgili planlar yapması gerekir.

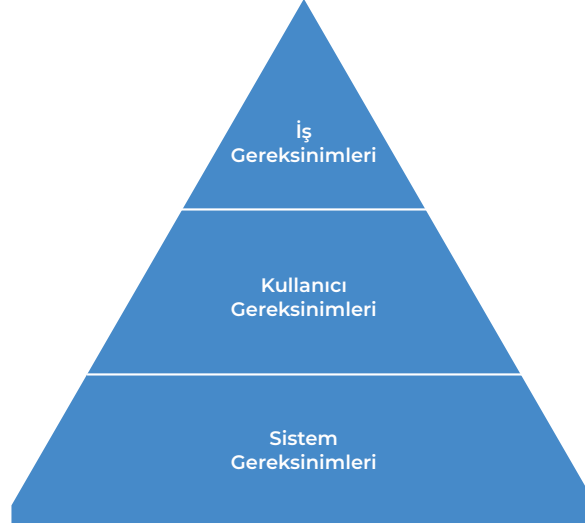
Herhangi bir sistemin tasarımı için kullanıcı gereksinimleri ile birlikte, geliştirilecek olan sistemin sahip olması gereken kabiliyetlerin de analiz edilmesi ve belirlenebilmesi gerekmektedir.



1. SİSTEM GEREKSİNİMLERİ

Sistem gereksinimleri, müşterinin ihtiyaç ve gereksinimlerini tam anlamıyla karşılayabilmek için bir sistemin sahip olması gereken kabiliyetlerin açıkça ifade edilmesidir. Sistem gerek-

sinimleri, Şekil 1'de ifade edilen Gereksinimler Hiyerarşisine göre İş Gereksinimleri ve Kullanıcı Gereksinimleri kavramlarından türetilmiştir.



Şekil 1. Gereksinimler Hiyerarşisi

İş Gereksinimleri bölümünde, neden projeye ihtiyaç duyulmaktadır sorusuna cevap aranırken, Kullanıcı Gereksinimleri bölümünde kullanıcıların, sistemin ne yapmasına ihtiyacı olduğu sorgulanmaktadır. Sistem Gereksinimleri bölümünde ise 'Sistemin Ne Yapması Gerekıyor' sorusu irdelenmektedir.

Sistem gereksinimleri; Fonksiyonel Gereksinimler ve Fonksiyonel Olmayan Gereksinimler olmak üzere iki kategoride tanımlanmaktadır. Fonksiyonel gereksinimler; sistemin gerekli davranış ve fonksiyonlarını tanımlarken, Fonksiyonel olmayan gereksinimler ise performans, güvenlik ve kullanılabilirlik gibi sistemin çalışmasını değerlendirebilmek için kullanılabilecek ölçütleri tanımlamaktadır.

Literatürde tanımlanan en yaygın fonksiyonel gereksinim türleri; Sertifika Gereksinimleri, İş Düzenlemeleri, İdari Fonksiyonlar, Raporlama Gereksinimleri, Denetim İzleme, Yetki Düzeyleri, Veri Yönetimi, Dış Arayüzler, Yasal ve Düzenleyici Gerekliliklerdir.

Projede fonksiyonel olmayan gereksinimlerin kapsamlı bir listesini tanımlamak, sistemin beklendiği gibi bir performans göstermemesi riskini olabildiğince azaltır ve performans standartlarının tanımlanabilmesine olanak sağlar.

Fonksiyonel Gereksinimler İçin Verilebilecek Örnekler:

Öğrenci Bilgi Sisteminde öğrencinin ders kaydı onaylandığında öğrenciye SMS ve eposta gönderilmesi.

Öğrenci Bilgi Sisteminde öğrencinin kullanıcı adı ve şifresi ile sisteme giriş yapabilmesi.

E-ticaret sisteminde müşterilerin tükenen ürünleri görebilmesi ve satıcıdan talep edebilmesi.

Otel Rezervasyon Sisteminde müşterilerin belirli tarih aralıklarında otelin boş odalarını ve fiyatlarını görerek rezervasyon yapabilmeleri.

Online Kurs Sisteminde öğrencinin sanal pos sistemi ile kredi kartından ödeme yaparak ilgili kursa kayıt hakkı kazanabilmesi.

Fonksiyonel Olmayan Gereksinimler İçin Verilebilecek Örnekler:

Öğrenci Bilgi Sisteminde öğrencinin ders kaydı onaylandığında öğrenciye SMS ve eposta 1 dakika içerisinde gönderilmelidir.

Öğrenci Bilgi Sisteminde öğrencinin şifresinin en az 8 karakter uzunluğunda olması, şifre karakterlerinin harf ve rakamlardan oluşması gerekir.

Otel Rezervasyon Sisteminin yalnızca Linux serverlarda çalışabilmesi.

Otel Rezervasyon Sisteminde arıza olması durumunda sistem 5 dakika içerisinde ayağa kaldırılabilir.

Fonksiyonel olmayan gereksinimlerin belirlenmesinde başvurulabilecek, uluslararası kabul görmüş standart örnekleri Tablo 1’de gösterilmektedir.

Tablo 1. Fonksiyonel olmayan gereksinimleri için Uluslararası kabul görmüş standartlar

Kategori	Alt Kategori	Standart
Teknik	Bilgisayar Teknolojisi- Bilgisayar Ağları	ISO/IEC/IEEE 8802
	Yazılım Kalite Yönetim Sistemi	ISO 9001:2000
	Biyometri	ISO/IEC 19784/5
	Telekomünikasyon	ISO ICS 33.040
Güvenlik	Bilgi ve Kayıt Yönetimi	ISO 15489
	Bilgi Güvenlik Yönetimi	ISO/IEC 27002
Gizlilik	Veri Koruma	ISO/IEC 27001
	Biyometri	ISO/IEC 19794/5
Denetim	Bilgi ve Belge Yönetimi	ISO 15489

2. GEREKSİNİM TESPİTİNDE KULLANILAN BAŞLICA FAALİYETLER

Gereksinim Öngörüsü

Yeni bir sistem tasarımı için belirli sorunları veya özellikleri ve gereksinimleri içeren önceki tecrübelerle dayalı olarak sistemin barındırması gereken özelliklerinin tahmin edilmesidir. Bu faaliyetle deneyimsiz bir analist tarafından fark edilmeyecek alanların analiz edilmesi sağlanabilir. Ancak araştırmanın yürütülmesinde kestirme yollara başvurulur ve önyargılı davranılırsa, gereksinim öngörüsü başarısızlıkla sonuçlanabilir.

Gereksinim Araştırması

Mevcut sistemin incelenerek daha ileri analizler için özelliklerinin belgelenmesi işlemidir. Sistem analizinin merkezinde analistin olgu bulma teknikleri, prototip oluşturma ve bilgisayar destekli araçlar kullanarak sistem özelliklerini belgelenmesi ve tanımlaması yer almaktadır.

Gereksinim Özellikleri

İhtiyaç özelliklerini belirleyen verilerin analizini, yeni oluşturulacak sistem için teknik özelliklerin tanımlanmasını ve hangi bilgi gereksinimlerinin sağlanacağını belirlemesini içermektedir. Gereksinim özellikleri, gerçek verilerin analiz edilmesini, temel gereksinimlerin tanımlanmasını ve gereksinim karşılama stratejilerinin seçimini içerir.

Sistem gereksinimlerinin doğru belirlenmemesi zaman ve maliyet açısından kayıplara neden olabilmekte ve kullanılabilirliği olumsuz yönde etkileyebilmektedir. Bu nedenle sistem gereksinimleri tanımlanırken aşağıdaki ölçütler dikkate alınmalıdır.

Tutarlılık: Gereksinimler kendi içerisinde tutarlı olmalı, herhangi bir belirsizliğe fırsat verilmemelidir.

Tamlık: Gereksinimler sistemin tüm girdi ve çıktılarını tanımlamalıdır.

Gerçeklenebilirlik: Gereksinimler, mevcut şart ve koşullarda gerçekleştirilebilir olmalıdır.

Gereklilik: Gereksinimler gerçek bir ihtiyacı karşılamalıdır.

Doğruluk: Gereksinimler doğru bir şekilde ifade edilmelidir.

Takip Edilebilirlik: Gereksinimler geliştirilecek sistemin işlev ve özelliklerine karşılık gelmelidir.

Onaylanabilirlik: Gereksinimler gözlemlenebilir olmalıdır.

3. OLGULARI (VERİLERİ) BELİRLEME YÖNTEMLERİ

Bilgi toplama, sistem analizinin merkezinde yer alır. Bilgi gereksinimlerinin belirlenmesi sıklıkla ve ikna edici bir şekilde bilgi sistemi geliştirme- nin en kritik aşaması olarak sunulur ve birçok bilgi sistemi başarısızlığı eksik ve yanlış bilgi gereksinimlerine bağlanmıştır (Pitts, 1997) Sistem analistleri, mevcut sistem ve kullanıcıların yeni bilgi sistemi ile performanslarını nasıl artırmak istedikleri hakkında bilgi toplamalıdır. Kullanıcıların gereksinimlerinin doğru bir şekilde anlaşılması, sistem geliştirme ekibinin sınırlı zaman ve sınırlı bütçe içinde son kullanıcılara uygun bir sistem sunmasına yardımcı olacaktır.

Olgular nicel biçimde ifade edildiğinde veri olarak adlandırılır. Geliştirilen herhangi bir projenin başarısı, mevcut bilgilerin doğruluğuna bağlıdır. Doğru bilgi de literatürde kabul görmüş belirli metot ve teknikler kullanılarak toplanabilir. Sistem hakkında bilgi toplamak için kullanılan bu yöntemler olgu bulma teknikleri olarak adlandırılır. Anket, Görüşme, Mevcut Belge, Form ve Dosyaların Analiz Edilmesi ve Kullanıcıları Doğrudan Gözlemleme gibi bazı geleneksel yöntemler vardır. Günümüzde, Ortak Uygulama Tasarımı ve Prototip oluşturma gibi bazı modern olgu toplama yöntemleri ortaya çıkmıştır.

Görüşme

Görüşme, bir bilgi sistemi hakkında bilgi toplamanın en temel yollarından biridir. İyi bir sistem analisti görüşme yapma konusunda iyi olmalıdır ve hiçbir proje görüşme yapılmadan yürütülemez. Etkili bir görüşme düzenlemenin

birçok tanımlı yolu vardır ve hiçbirisi diğerlerinden üstün değildir. Ancak deneyimli analistler, etkili bir görüşme için aşağıda tanımlı bazı en iyi uygulamaları yaygın olarak kullanmaktadır:

- ✓ Randevu, ön hazırlık sorusu, kontrol listesi, gündem ve sorular dahil olmak üzere görüşmeyi dikkatlice hazırlayın.

- ✓ Görüşme sırasında müşteriyi dikkatle dinleyin ve not alın (mümkünse ses kaydediciye kaydedin).

- ✓ Görüşmeden sonraki ilk 48 saat içinde mutlaka notları gözden geçirin.

- ✓ Görüşme yaparken sistemin gidişatıyla ilgili tarafsız olun.

- ✓ Farklı görüşler arayın.

Anketler

Anketler, çok sayıda kişiden nispeten kısa bir sürede bilgi toplama ve sonuçlarının yorumlanmasında daha az önyargılı olma avantajına sahip bir yöntemdir. Ankete, o alanı ilgilendiren doğru anket katılımcılarının seçilmesi ve etkili anketlerin tasarlanması bu bilgi toplama yöntemindeki kritik konulardır. İnsanlar genellikle bir sistemin işlevlerinin sadece bir kısmını kullanırlar, bu nedenle her zaman sistem fonksiyonlarının veya süreçlerinin sadece bir kısmına aşinadırlar. Çoğu durumda, anketlerin bir kopyasının tüm kullanıcılara uymayacağı açıktır. Etkili bir anket yürütmek için analist kullanıcıları uygun şekilde gruplandırmalı ve farklı gruplar için farklı anketler tasarlamalıdır. Bununla birlikte, iyi anketler

oluşturma becerisi pratik ve deneyimle gelişen bir beceridir. Anketleri tasarlarlarken sistem analisti en azından aşağıdaki konularla ilgilenmelidir.

- ✓ Soruların tutarlılığı
- ✓ Katılımcıların cevaplarının tutarlılığı
- ✓ Ne tür bir soru uygulanmalıdır, açık uçlu mu kapalı uçlu mu?
- ✓ Anketlerin uygun uzunluğu nedir?

Kullanıcıları Doğrudan Gözlemleme

İnsanlar, güvenilir olmaya ve doğru olduğunu düşündükleri şeyleri söylemeye çalışsalar bile, her zaman çok güvenilir kaynak kişiler değildir. İnsanlar genellikle ne yaptıklarını ya da nasıl yaptıklarını tam olarak bilmezler. Bu durum özellikle nadir görülen olaylar, geçmişten gelen meseleler ya da insanların büyük bir tutkuyla bağlı oldukları konular için geçerlidir. İnsanların kendi eylemlerini güvenilir bir şekilde yorumlamalarına ve rapor etmelerine her zaman güvenilemeyeceğinden, sistem analisti insanların ne yaptıklarını izleyerek veya insanların iş ortamında nasıl davrandıklarına dair nispeten objektif ölçümler elde ederek söylediklerini tamamlayabilir ve doğrulayabilir. Ancak gözlem, insanların normal çalışma rutinlerini değiştirmelerine neden olabilir. Bu da toplanan bilgiyi taraflı hale getirir (Jeffrey ve ark.,2001).

Mevcut Belge, Form ve Dosyaların Analiz Edilmesi

Sistem analisti mevcut sistem ve kurumsal dokümantasyonu inceleyerek mevcut sistem ve bu sistemlerin desteklediği organizasyon hakkındaki detayları öğrenebilir. Bu dokümanlarda sistem analizi, mevcut sistemlerle ilgili sorunları, sistemin eksikliklerini ve yeni ihtiyaçları ne ölçüde karşılayabildiğini, yeni sistem ihtiyacını etkileyebilecek sorun veya fırsatları, mevcutta kullanılan

sistemin neden o şekilde tasarlandığı gibi teferuatları öğrenerek yeni sistemin tasarımı için bir örnekleme yapabilir (Jeffrey ve ark.,2001). Ayrıca organizasyon şemasına bakılarak proje sahipleri ve kullanıcıları arasındaki ilişki tanımlanabilir. Ancak, bu resmi belgeleri analiz ederken, analistler resmi belgelerde tanımlanan sistemler ile gerçek dünyadaki pratik sistemler arasındaki farka dikkat etmelidir. Resmi prosedürlerin yetersizlikleri, bireysel çalışma alışkanlıkları ve tercihleri, kontrole karşı direnç ve diğer faktörler nedeniyle, resmi sistem ve gayri resmi sistem olarak adlandırılan sistem arasındaki fark yaygın olarak mevcuttur.

Ortak Uygulama Tasarımı

Ortak Uygulama Tasarımı (OUT), yeni veya değiştirilmiş bilgi sistemlerinin gereksinimlerini tanımlamak için kolaylaştırılmış, ekip tabanlı bir yaklaşımdır. OUT, 1970'lerin sonunda IBM'de başlatılmıştır. OUT'un arkasındaki ana fikir, mevcut bir sistemin analizinde yer alan kilit kullanıcıları, yöneticileri ve sistem analistlerini bir araya getirmektir. Analiz aşamasında OUT kullanmanın birincil amacı, sistemle ilgili kilit kişilerden eş zamanlı olarak sistem gereksinimlerini toplamaktır. Neticede yoğun ve yapılandırılmış, ancak son derece etkili bir süreç ortaya çıkar. Tüm kilit kişilerin aynı anda tek bir yerde bir araya gelmesi, analistlerin nerede anlaşma alanları olduğunu ve nerede çatışmalar olduğunu görmelerini sağlar (Jeffrey ve ark.,2001).

Bir OUT'daki tipik katılımcılar şunlardır: OUT oturum lideri, son kullanıcılar, iş yöneticileri, sponsor, sistem analistleri, bilgi sistemi personel ve raportör. OUT ekibi, hepsi de yüksek kaliteli bir sistemin tasarlanmasında pay sahibi olan altı ila on altı kişiden oluşan bir gruptur. Grup üyelerinin yaklaşık üçte ikisi işlevsel uzmanlar, diğer üçte biri ise sistem profesyonelleridir. OUT oturumları genellikle katılımcıların normalde çalış-

tıkları yerlerin dışında bir yerde gerçekleştirilir ve genellikle katılımcıların at nalı şeklindeki masaların etrafında oturduğu özel amaçlı odalarda yapılır. Bir atölye çalışmasına bu kadar çok farklı türden insanın dahil edilmesi, OUT oturumunun etkili ve verimli bir şekilde nasıl organize edileceğini büyük bir zorluk haline getirmektedir (Zhu, 2003).

Bir OUT tamamlandığında, nihai sonuç, bir değiştirme sistemi çalışmasıyla ilgili mevcut sistemin işleyişini detaylandıran bir dizi belgedir. Bu gereksinim tanımlama belgeleri genellikle iş faaliyeti modelini ve tanımlarını, veri modelini ve tanımını, veri giriş ve çıkış gereksinimlerini içerir. Aynı zamanda arayüz gereksinimlerini, ekran ve rapor düzenlerini, geçici sorgu özelliklerini, menüleri ve güvenlik gereksinimlerini de içerebilir. Sistem geliştirme yaşam döngüsünün daha sonraki bir noktasında kullanıldığında, bir OUT oturumu bir sistem prototipini iyileştirmek, sistem kullanıcıları için yeni iş profilleri geliştirmek veya bir uygulama planı geliştirmek için de kullanılabilir (Zhu, 2003).

Ancak, OUT'un tüm potansiyelinden faydalanmak için, grup yazılımı araçları OUT çalışma oturumlarında uygulanmalıdır. Ortak Uygulama Geliştirme tekniğini desteklemek için grup yazılımı araçlarının kullanılması, bu tekniğin değerini önemli ölçüde artırır. Grup yazılımı araçları otomatik bir OUT çalışması sırasında kullanıldığında, bilginin üretilmesini, analiz edilmesini ve belgelenmesini büyük ölçüde kolaylaştırır. Bu, özellikle yeni sistemlerin gereksinimlerini tanımlamak ve bunlar üzerinde fikir birliği oluşturmak için yürütülen OUT çalışma grupları için değerlidir (Leventhal ve ark., 1995).

Prototip Oluşturma

Prototip oluşturma, fikirlere yatırım yapmadan önce onları keşfetmenin bir yoludur. Çoğu sistem geliştiricisi, erken kullanılabilirlik verilerinden elde edilen faydaların geç kullanılabilirlik verilerinden elde edilenlerden en az on kat daha fazla olduğuna inanmaktadır (Nielsen, 2003& Berkun, 2000).

Prototip oluşturma, sistem analistlerinin kullanıcılara temel gereksinimleri istenen bilgi sisteminin çalışan bir versiyonunu hızlı bir şekilde göstermelerini sağlar. Prototipi gördükten ve test ettikten sonra, kullanıcılar genellikle mevcut gereksinimleri yenilerine göre ayarlar. Gereksinimlerin belirlenmesini desteklemek için prototip oluşturma kullanıldığında amaç, nihai sistem için somut özellikler geliştirmektir, nihai sistemi prototip oluşturarak inşa etmek değildir.

Prototip oluşturma, kullanıcı gereksinimleri net olmadığında veya iyi anlaşılmadığında, bir veya birkaç kullanıcı ve diğer paydaşlar sistemle ilgilendiğinde, olası tasarımlar karmaşık olduğunda ve tam olarak değerlendirmek için somut bir yapı gerektirdiğinde, kullanıcılar ve analistler arasında geçmişte iletişim sorunları yaşandığında ve çalışan sistemleri hızlı bir şekilde oluşturmak için araçlar ve veriler hazır olduğunda gereksinimlerin belirlenmesi için en yararlı yöntemdir (Jeffrey ve ark., 2001).

Analistler, Prototip oluşturmaya uygularken bu gereksinim belirleme yöntemiyle ilgili resmi olmayan dokümantasyon, göz ardı edilen ince ama önemli gereksinimler vb. gibi potansiyel sorunlarla ilgilenmelidir.

BÖLÜM ÖZETİ

Herhangi bir sistemin tasarımı için kullanıcı gereksinimleri ile birlikte geliştirilecek olan sistemin sahip olması gereken kabiliyetlerin de analiz edilmesi ve belirlenmesi gerekmektedir. Sistem gereksinimleri, müşterinin ihtiyaç ve gereksinimlerini tam anlamıyla karşılayabilmek için bir sistemin sahip olması gereken kabiliyetlerin açıkça ifade edilmesidir.

Sistem gereksinimleri; Fonksiyonel Gereksinimler ve Fonksiyonel Olmayan Gereksinimler olmak üzere iki kategoride tanımlanmaktadır. Fonksiyonel gereksinimler, sistemin gerekli davranış ve fonksiyonlarını tanımlarken, Fonksiyonel Olmayan Gereksinimler ise performans, güvenlik ve kullanılabilirlik gibi sistemin çalışmasını değerlendirebilmek için kullanılabilecek ölçütleri tanımlamaktadır.

Gereksinim tespitinde kullanılan başlıca faaliyetler; gereksinim öngörüsü, gereksinim araştırması ve gereksinim özellikleridir.

Sistem gereksinimleri tanımlanırken tutarlılık, tamlık, gerçekleştirilebilirlik, gereklilik, doğruluk, takip edilebilirlik ve onaylanabilirlik ölçütleri dikkate alınmalıdır.

Sistem hakkında bilgi toplamak için kullanılan yöntemler, olgu bulma teknikleri olarak adlandırılır. Anket, görüşme, mevcut belge, form ve dosyaların analiz edilmesi ve kullanıcıları doğrudan gözlemleme gibi bazı geleneksel yöntemler vardır. Günümüzde, ortak uygulama tasarımı ve prototip oluşturma gibi bazı modern olgu toplama yöntemleri ortaya çıkmıştır.

Kaynakça

- Berkun, S. (2000). The art of UI prototyping. Microsoft Corp. ESSAYS NovDec.
- Jeffrey A. Hoffer, Joey George, Joseph Valacich (2001). Modern Systems Analysis and Design (3rd Edition)
- Matthews, Joy. (1995). Computerworld. Framingham JAD to the Rescue, Vol. 29, Iss. 49; pg. 93, 1 pgs
- Leventhal, Naomi S. (1995). Using groupware tools to automate Joint Application Development. Journal of Systems Management. Cleveland: Sep/Oct 1995. Vol. 46, Iss. 5; pg. 16, 6 pgs
- Nielsen, J. (2003). Paper Prototyping: Getting User Data before you code, Nielsen Norman Group.
- Pitts, M. G. (1997). The Use of Stopping Rules in Information Requirements Determination.
- Zhu, Z. (2003). Requirements Determination and Requirements Structuring. Elektronik adres: http://www.umsl.edu/~sauter/analysis/6840_f03_papers/zhu.

BÖLÜM 5

SİSTEM GEREKSİNİMLERİNİN BELİRLENMESİ

ANAHTAR KAVRAMLAR

Kullanım Durum, Kavramsal Veri Modeli,
Veri Modelleme, Mantıksal Veri Modeli,
Aktörler- Durumlar- İlişkiler, Fiziksel Veri Modeli



ÖĞRENME HEDEFLERİ

-  Projenin kullanım durumlarına ilişkin kavramları açıklar
-  Veri modelleme yönetimlerini kavrar
-  Geliştireceği sistemin kullanım durumlarını oluşturabilir

GİRİŞ

Sistem analistleri, sistem geliştirmenin analiz aşamasında verilerin nasıl oluşturulacağı, nasıl depolanacağı, veriler üzerinde hangi işlemlerin gerçekleştirileceği gibi teknik ayrıntıya girmeden, sistemi genel hatlarıyla ifade edebilmek için model oluştururlar. Veri modelinde sistemin görsel bir temsili oluşturulur. Benzer biçimde, sistemin gereksinimlerini modelleyebilmek için kullanım durum kullanılmaktadır. Kullanım durumlar, sistem mimarisini görselleştirmenin önemli bir unsuru olarak karşımıza çıkmaktadır. Veri modelleme ise veri noktaları ve yapıları arasındaki bağlantıları iletmek için sistem parçalarının görsel bir temsili oluşturma sürecidir. Bu ünite, kullanım durumun ve bileşenlerinin neler olduğu, ne amaçla kullanıldığı, veri modelleme kavramı, veri modeli çeşitleri ve veri modelleme türleri konularına değinilmiştir.



1. KULLANIM DURUMLARI (USE CASES)

Sistem tasarımında bir müşterinin bir ürün, hizmet veya sistemle nasıl etkileşeceğini ifade etmek, gereksinimlerin toplanması ve üst düzey paydaş iletişimi için önemlidir. Kullanım durumları, bir sistemin gereksinimlerini yakalamak, modellemek ve belirtmek için kullanılan bir tekniktir (Bittner, 2003). Bir kullanım durumu, sistemin aktörleriyle etkileşim içinde gerçekleştirebileceği ve hedeflerine katkıda bulunan gözlemlenebilir bir sonuç üreten bir dizi davranışa karşılık gelir. Aktörler, insan kullanıcıların veya diğer sistemlerin etkileşimdeki rolünü temsil eder. Gereksinim analizinde, bir kullanım durumu tanımlanırken, birincil aktörü için temsil ettiği belirli kullanıcı hedefine göre adlandırılır. Vaka, özel koşullar, istisnalar veya hata durumları gibi varyantların yanı sıra genel faaliyet ve olay sırasını açıklayan metinsel bir açıklama

veya ek grafik modellerle daha da detaylandırılır. Kullanım durum diyagramı bir ürünü veya hizmeti kullananların o ürün veya hizmetle nasıl etkileşime gireceklerinin ve ürün veya hizmetin ne yaptığının görsel olarak temsilidir. Bir kullanım durumu, başarı senaryolarını, başarısızlık senaryolarını ve kritik varyasyonları veya istisnaları oluşturabilir.

Kullanım Durumuyla ilgili ilk çalışma, İsveçli bilgisayar bilimcisi Ivar Jacobson tarafından 1987 yılında yapılmıştır. Ivar Jacobson senaryoyu sistem ile kullanıcı arasında gerçekleştirilen etkileşimli diyalog olarak tanımlamaktadır. Jacobson daha sonra iki yazılım mühendisi ile birlikte bir sistem tasarımını görselleştirmenin standart yolunu sunan Birleşik Modelleme Dili'ni (UML) oluşturmuştur.

1.1. Kullanım Durumlarının Amaçları

- Kapsam yönetimi
- Gereksinimlerin belirlenmesi
- Kullanıcının sistemle etkileşime gireceği yolların ana hatlarıyla belirtilmesi
- Sistem mimarisini görselleştirme
- Teknik gereksinimleri iş paydaşlarına iletme
- Risk yönetimi

1.2. Kullanım Durum Modeli

Kullanım durum modeli, bir sistem ile bir aktör arasındaki etkileşimlerin görsel olarak sunumudur. Kullanım durum modelleri iş süreçlerini temsil eder. Bu da ön koşulların ve tetikleyicilerin daha fazla ifade edilmesine yardımcı olur.

Bir kullanım durum modeli genellikle UML ile ifade edilir. Bu modelde dört ana bileşen bulunmaktadır: aktörler, sistem, kullanım durumları ve ilişkiler.

UML içerisinde sistem, bir dikdörtgen ile gösterilir. Aktörler dikdörtgenin dışında çubuktan insanlar olarak gösterilirken, kullanım durumları ise dikdörtgenin içerisinde ovalerde metin olarak sunulurlar. Düz ve kesikli çizgiler sistem

ile aktörlerin kullanım durumları arasındaki ilişkiyi temsil etmektedir (Daly, 2022).

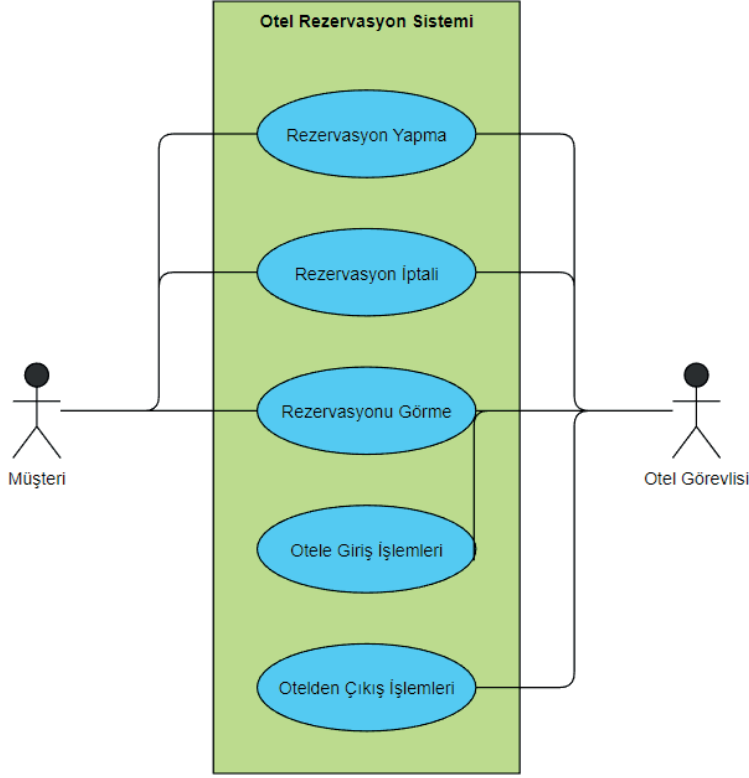
Kullanım durum diyagramı ise, temel olarak bir tür kullanım durum modelidir. Kullanım durum diyagramları aşağıda belirtilen durumlar için kullanılır:

- Sistemin akışını ve davranışını görselleştirmek
- Sistemin işlevselliğini örneklemek
- Temel sistem-kullanıcı etkileşimlerini temsil etmek.

Kullanım durum diyagramının amacı, oluşturulan sistemin hangi işlevinin hangi aktörler

tarafından gerçekleştirileceğini ve kullanım durumlarının kendi arasındaki ilişkilerini göstermektedir.

Kullanım durum diyagramına ait bir örnek Şekil 1'de gösterilmiştir:



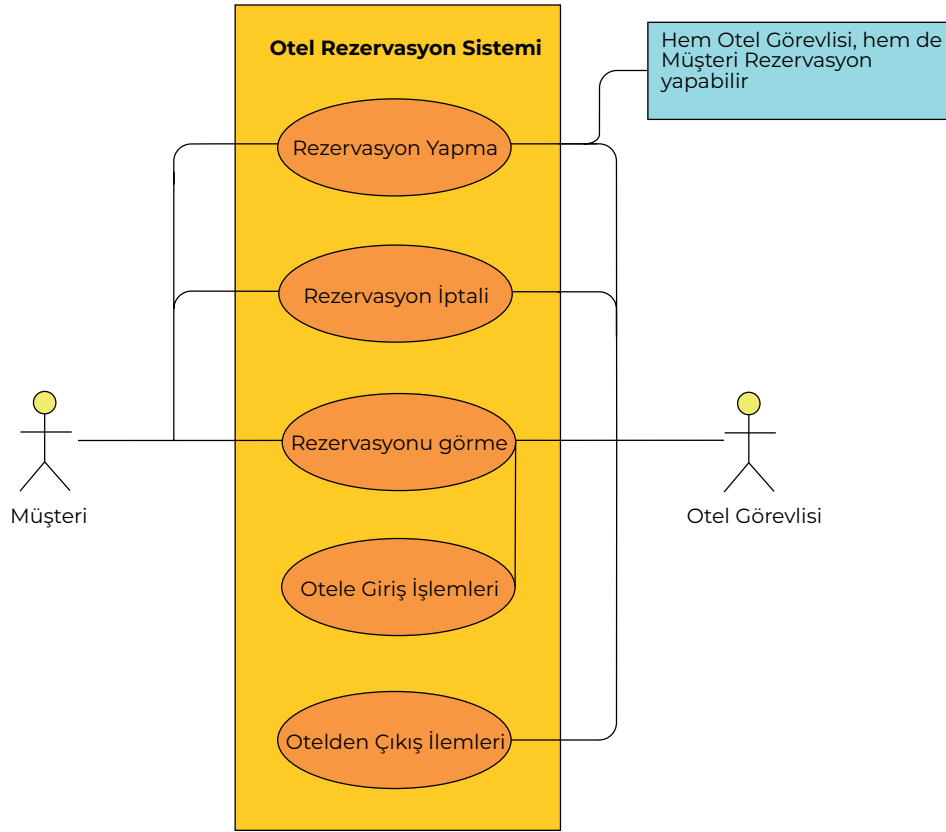
Şekil 1. Otel Rezervasyon Sistemine Ait Kullanım Durum Diyagramı

Aktör, bir sistemin dışında onunla etkileşime giren herhangi bir şeydir. Örneğin, insanların internet üzerinden mal satın almalarını sağlayan bir sistemde, insan kullanıcılar önemli aktörler olacaktır, ancak kullanıcıların alışverişleri için ödeme yapmalarını sağlayan kredi kartı sistemi de aynı şekilde bir aktör olacaktır (Jackson, 1995).

Kullanım durumları, sistemin gerçekleştirdiği bazı görevleri veya tutarlı işlevsellik birimlerini temsil eder. Bir aktör figürü ile bir kullanım durumu ovalini birleştiren çizgi, aktör ile kullanım durumu arasındaki iletişimi temsil eden bir iliş-

kiyi gösterir. Bu, aktörün görevin yerine getirilmesine dahil olabileceği ama dahil olmak zorunda olmadığı anlamına gelir.

Kullanım durum diyagramları yalnızca fonksiyonel gereksinimlerle (sistemin yapması gereken şeyler) ilgilenir, ancak bir kullanım durum diyagramı geliştirirken ve işlevsel olmayan gereksinimlerle karşılaşıldığında, kullanım durum diyagramına açıklayıcı bir not eklenerek bunların kaydedilmesi gerekir. Bu durumla ilgili bir örnek, Şekil 2'de gösterilmiştir.



Şekil 2. Kullanım Durum Diyagramında Açıklayıcı Not Kullanımı

UML’de kullanım durumlarını çevreleyen bir dikdörtgenle gösterilen sistem sınırı, sistemi diğer yapılardan ayıran önemli bir kavramsal çizgidir. Bir kullanım durum diyagramında temsil edilen sistemin etrafına sınır çizilerek çözümün kapsamı belirlenmiş olur. Genel kullanım durum diyagramlarında sınır dikdörtgeni çoğunlukla gösterilmez.

Şekil 1 ve Şekil 2 aynı zamanda yaygın bir problemi de ortaya koymaktadır. Yalnızca diyagramdan, iki aktörden hangisinin (Müşteri veya Otel Görevlisi) rezervasyon yapma kullanım senaryosunu başlattığı net değildir. Kullanım durumu başarıyla tamamlamak için her iki aktöre de ihtiyaç duyuluyor olabilir. UML bu sorunu ortadan kaldırmak için çeşitli yollar sunmaktadır. Bunların en basit olanı, UML içerisinde açıklayıcı not kullanmaktır.

Senaryo, kullanım durumlarını gösteren bir dizi eylemin açıklamasıdır. Bir sistemin kullanıcıları (aktörleri) ile sistemin kendisi arasındaki

etkileşimi ve diyalogu tanımlar. Aşağıda iki örnek senaryo görülmektedir:

1. Yunus isimli müşteri, 30 Mart için X Otel’de bir oda ayırtmak istemektedir. O tarih için bir oda müsaittir ve otel görevlisi Yunus için rezervasyon yapar.

2. Azize isimli müşteri, Haziran ayının ilk haftası için Y Otel’de bir oda ayırtmak istemektedir. Haziran ayında yedi gün için boş oda yoktur, ancak dört gün için bir oda ve ardından üç gün için aynı tipte başka bir oda mevcuttur. Otel görevlisi bu seçeneği Azize’ye sunar fakat Azize bu teklifi reddeder.

Her iki senaryo da rezervasyon yapma kullanım durumunun olası örnekleridir. Etkileşimleri ve sonuçları farklıdır. İlkinde, kullanım durumunun başarılı bir sonuca götüren bir açıklaması vardır. İkincisinde ise ana başarı senaryosunda bir istisna vardır. Bir kullanım durumu için normal davranışın istisnaları, özellikle aktörlerin bir kullanım durumunu tamamlamadan iptal etme-

ye karar verdiği durumlarda yaygındır. Bununla birlikte, tüm senaryolar arasındaki ortak tema, bir aktörün bir kullanım senaryosu tarafından tanımlanan hedefe ulaşma niyetidir. Yukarıdaki başarısız senaryoda Azize, Y Otel’de rezervasyon yaptırmaya çalışıyordu. Belki de kaldığı süre boyunca oda değiştirme fikrinden hoşlanmamıştı. Bu nedenle, bir kullanım durumu olağandışı veya alternatif senaryoları içermelidir.

Bir araştırmaya sadece bir kullanım durumu ve onun ana başarı senaryosu belirlenerek başlanabilir ve daha sonra bu durum geliştirilebilir. Sadece ana başarı senaryosu için değil, ilgili

tüm senaryolar için her bir kullanım durumuna ilişkin bilgilerin nasıl kaydedileceğine karar verilmesi gerekecektir. En basitinden, her bir kullanım durumu için her bir senaryoyu sonucuyla birlikte detaylandıran metinsel bir açıklama kaydedilebilir.

Her gereksinim, geliştirici ile müşteri arasındaki sözleşmenin bir parçasıdır. Her iki tarafın da her bir kullanım senaryosunda neyin ifade edildiği ve ne anlama geldiği konusunda net bir anlayışa sahip olması gerekir. Aşağıdaki tablo, bir kullanım senaryosunu yapılandırmanın ve kaydetmenin bir yolunu göstermektedir.

Tablo 1. Otel Rezervasyon Sisteminin Metinsel Açıklaması

Kullanım Durum Adı	Rezervasyon Yapma İşlemi
Aktörler	Otel Görevlisi
Amaç	Otel odasının rezerve edilmesi
Ön Koşul	Yok (yani, bu kullanım senaryosunu gerçekleştirmeden önce yerine getirilmesi gereken hiçbir koşul yoktur).
Son Koşul	İstenilen tipte bir oda, talep edilen dönem için müşteri için rezerve edilmiş olacak ve oda o dönem için dolu olacaktır.
Varsayımlar	Bir müşteri internet üzerinden de rezervasyon yapabilir.

1.3. Temel Başarı Senaryosu

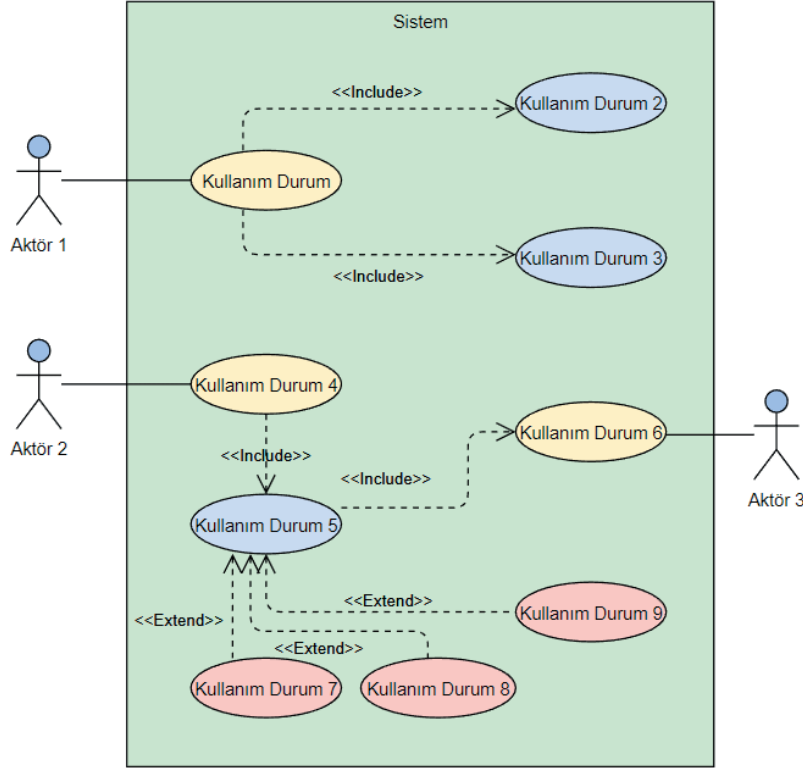
1. Müşteri rezervasyon talebinde bulunur.
2. Müşteri istediği oteli, tarihleri ve oda tipini seçer.
3. Otel Görevlisi, talep için müsaitlik durumunu ve fiyatı sağlar.
4. Müşteri teklife devam etmeyi kabul eder.
5. Müşteri, otelin kayıtları için kimlik ve iletişim bilgilerini verir.
6. Müşteri, ödeme bilgilerini verir.
7. Otel Görevlisi bir rezervasyon oluşturur ve ona bir numara verir.
8. Otel Görevlisi rezervasyon numarasını müşteriye açıklar.
9. Otel Görevlisi rezervasyonun onayını oluşturur ve müşteriye gönderir.

Bazı durumlarda kullanım durumların, işlevini tamamlayabilmesi için başka kullanım durumlarla birlikte kullanılması gerekir. Yani mevcut kullanım durum, başka kullanım durumlar olmaksızın tamamlanamaz. Örneğin, bir alışveriş uygulamasından sipariş verebilmesi için öncelikle sipariş vermek istediği firmayı seçip, o firmaya ait ürünleri görüntülemeli ve daha sonra satın almak istediği ürünleri sepete eklemelidir. Yani “Sipariş Ver” isimli kullanım durumunun tamamlanabilmesi için “Firma Seç” ve “Sepete Ekle” kullanım durumların çalışması gerekir. Include (içerme) olarak isimlendirilen bu ilişki türü, Şekil 3’de gösterildiği şekilde kullanılmaktadır.

Kullanım durumları arasındaki başka bir ilişki türü ise Extend (genişletme) olarak adlandırılır ve iki kullanım durumu arasında opsiyonel olarak var olabilecek bir durumu ifade etmek için kul-

lanılır. Bu ilişki türü, bir kullanım durumunun kapsamının, başka bir kullanım durum tarafından genişletilebilmesinin mümkün olduğu durumlar için geçerlidir. Extend (genişletme) ilişki türü, Şekil 3’de gösterildiği şekilde kullanılmaktadır.

Diyagramlar üzerindeki okların yönü, kullanım durumlar arasındaki bağımlılığın yönünü gösterir.



Şekil 3. Kullanım Durumlar için Include ve Extend İlişkisinin Kullanımı

2. KULLANIM DURUM DİYAGRAMINDA DİKKAT EDİLMESİ GEREKEN UNSURLAR

- Kullanım durum diyagramı oluştururken dikkat edilmesi gereken unsurlardan birisi, kullanım durum isimlendirilmesinde tercih edilen kelimelerdir. Kelimeler sıradan değil, sistemin söz konusu davranışını ve işlevini yansıtacak şekilde seçilmelidir. Kullanım durumlar isimlendirilirken genellikle emir kipi kullanılır.
- Aktörler için de ilgili kullanıcı gruplarını yansıtan isimler tercih edilmelidir.

- Include ve Extend ilişkileri sadece gerekli olduğu durumlarda kullanılmalı, diyagram oluşturma temel amacının gereksinimleri belirlemek olduğu unutulmamalı, basitliğin korunmasına mümkün olduğunca dikkat edilmelidir.
- Gerektiğinde bazı hususlara açıklık getirebilmek için diyagrama açıklayıcı notlar eklenmelidir (Alagöz, 2014).

3. VERİ MODELLEME

Veri modelleme, veri noktaları ve yapıları arasındaki bağlantıları iletme için tüm bir bilgi sisteminin veya sistem parçalarının görsel bir temsiliyi oluşturma sürecidir. Amaç, sistem içinde kullanılan ve depolanan veri türlerini, bu veri türleri arasındaki ilişkileri, verilerin gruplandırıl-

ma ve organize edilme yollarını, biçimlerini ve niteliklerini göstermektir.

Veri modelleri ihtiyaçlar etrafında oluşturulur. Kurallar ve gereksinimler, yeni bir sistemin tasarımına dahil edilebilmeleri veya mevcut bir

sistemin yinelenmesi sırasında uyarlanabilmeleri için iş paydaşlarından alınan geri bildirimler yoluyla önceden tanımlanır.

Veriler çeşitli seviyelerde modellenenebilir. Süreç, gereksinimler hakkında bilgi toplanmasıyla başlar. Bu iş kuralları daha sonra somut bir veritabanı tasarımı formüle etmek için veri yapılarına dönüştürülür.

Veri modellemesinde standartlaştırılmış şemalar ve teknikler kullanılır. Bu, veri kaynaklarını tanımlamak ve yönetmek için ortak, tutarlı ve öngörülebilir bir yol sağlar.

Veri Modeli Çeşitleri

Her tasarım sürecinde olduğu gibi, sistem tasarımı da soyut seviyede başlar ve giderek daha somut ve spesifik hale gelir. Veri modelleri genellikle soyutlama derecelerine göre değişen üç kategoriye ayrılabilir. Süreç kavramsal bir modelle başlar, mantıksal bir modele doğru ilerler ve fiziksel bir modelle sonuçlanır.

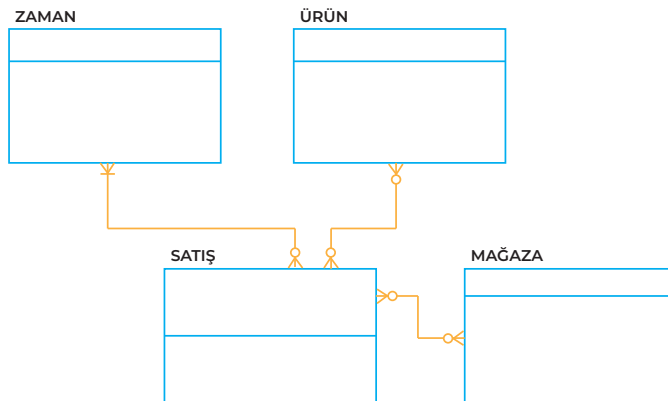
Kavramsal Veri Modeli

Etki alanı modelleri olarak da adlandırılırlar ve sistemin neleri içereceği, nasıl organize edile-

ceği ve hangi iş kurallarının söz konusu olduğuna dair büyük resimli bir görünüm sunarlar. Kavramsal modeller genellikle ilk proje gereksinimlerinin toplanması sürecinin bir parçası olarak oluşturulur. Tipik olarak, varlık sınıflarını, bunların özelliklerini ve kısıtlamalarını, aralarındaki ilişkileri ve ilgili güvenlik ve veri bütünlüğü gereksinimlerini içerirler. Kavramsal veri modelleri, teknik ayrıntılara girmeden modellemekte olduğunuz iş süreçlerini yöneten kavramları ve kuralları görselleştirir (IBM,2020).

Tipik olarak, kavramsal bir model sistemin içeriğinin, organizasyonunun ve ilgili iş kurallarının üst düzey bir görünümünü gösterir. Örneğin, bir e-ticaret işletmesi için veri modeli satıcıları, ürünleri, müşterileri ve satışları içerecektir. Bir iş kuralı, her satıcının en az bir ürün tedarik etmesi gerektiği şeklinde olabilir.

Kavramsal modeller için standart bir format yoktur. Önemli olan, hem teknik hem de teknik olmayan paydaşların veri projelerinin amacı, kapsamı ve tasarımı üzerinde uyum sağlamalarına ve anlaşmalarına yardımcı olmasıdır. Aşağıda örnek bir kavramsal veri modeli sunulmuştur.



Şekil 4. Kavramsal Veri Modeli Örneği

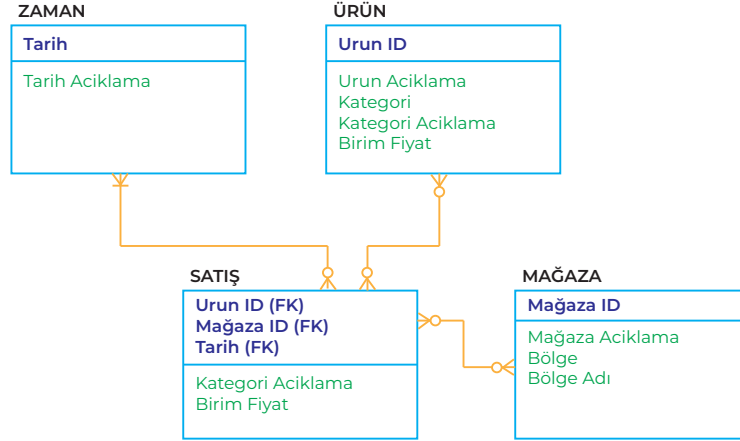
Mantıksal Veri Modeli

Daha az soyutlar ve söz konusu etki alanındaki kavramlar ve ilişkiler hakkında daha fazla

ayrıntı sağlarlar. Mantıksal veri modeli, kavramsal modeli temel alır ve projenin veri öğelerini ve ilişkilerini tanımlar. Mantıksal veri modelleri

herhangi bir teknik sistem gereksinimi belirtmez. Son derece prosedürel uygulama ortamlarında veya veri ambarı tasarımı ya da raporlama sistemi geliştirme gibi doğası gereği veri odaklı olan projeler için faydalı bir modeldir.

Veri mimarları ve iş analistleri, bir veritabanı yönetim sisteminin (bir veritabanındaki verileri depolayan, alan, tanımlayan ve yöneten yazılım) uygulanmasını planlamak için mantıksal veri modelini kullanırlar. Aşağıda mantıksal veri modeline ait bir görsel yer almaktadır.



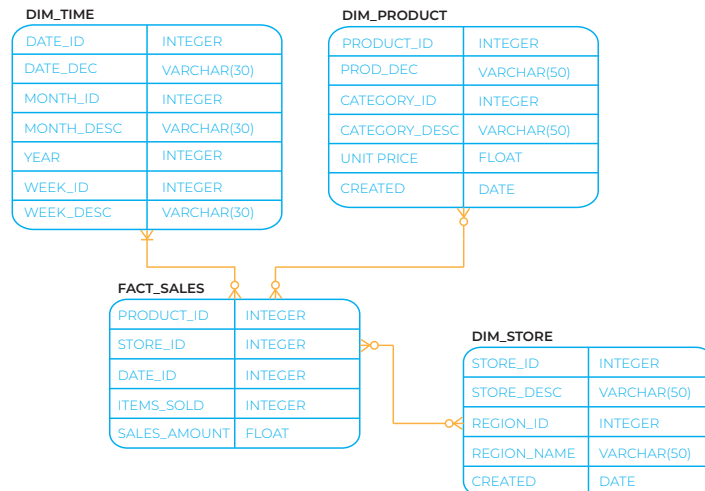
Şekil 5. Mantıksal Veri Modeli Örneği

Fiziksel Veri Modeli

Fiziksel veri modeli tekniktir. Veritabanı analistleri ve geliştiricileri bunu veritabanının ve ilgili veri yapılarının tasarımı için kullanır. Model, teknik veri gereksinimleriyle birlikte depolanacak veri türlerini belirtir.

Veri türü belirtilmelerine örnek olarak, bir verinin tamsayı mı yoksa ondalıklı mı olacağı, veya metinsel bir tipte mi olacağı verilebilir. Teknik gereksinimler arasında depolama ihtiyaçları, erişim hızı ve veri yedekliliği ile ilgili ayrıntılar yer alır.

Verilerin bir veritabanı içinde fiziksel olarak nasıl saklanacağına ilişkin bir şema sağlarlar. Bu nedenle, veri modelleri içerisinde en az soyut olanıdır. Varlıklar arasındaki ilişkileri gösteren ilişkisel tabloların yanı sıra bu ilişkileri sürdürmek için kullanılacak birincil anahtarlar ve yabancı anahtarlar da dahil olmak üzere ilişkisel bir veritabanı olarak uygulanabilecek nihai bir tasarım sunarlar. Aşağıda fiziksel veri modeline ait bir görsel yer almaktadır.



Şekil 6. Fiziksel Veri Modeli Örneği

3.1. Veri Modelleme Türleri

Veri modelleme, veritabanı yönetim sistemleri ile birlikte gelişmiştir ve işletmelerin veri depolama ihtiyaçları arttıkça model türleri de karmaşıklaşmıştır. Veri modelleme, bir veritabanı içindeki verilerin mantıksal yapısının bir diyagramıdır. Veri modelleme, insanların verileri daha iyi anlamalarına ve gelecekteki sonuçları tahmin etmek için verileri kullanmalarına yardımcı olabilir. En sık kullanılan veri modelleme türleri aşağıda özetlenmiştir.

1. Hiyerarşik Veri Modeli

Hiyerarşik veri modeli, verileri ebeveyn-çocuk ilişkisi olarak da bilinen ağaç benzeri bir hiyerarşide düzenlemeye yönelik bir yapıdır. Hiyerarşik bir veri modelinde her kayıt, hiyerarşide aynı seviyedeki her kayıt için aynı değer olan bir anahtarla benzersiz bir şekilde tanımlanır. Bu model, 1966 yılında tanıtılan ve özellikle bankacılıkta hızla yaygın kullanım alanı bulan IBM Bilgi Yönetim Sisteminde (IMS) uygulanmıştır. Bu yaklaşım daha yeni geliştirilen veritabanı modellerine göre daha az verimli olsa da, Genişletilebilir Biçimlendirme Dili (XML) sistemlerinde ve coğrafi bilgi sistemlerinde (GIS) hala kullanılmaktadır.

2. İlişkisel Veri Modeli

İlişkisel veri modelleri ilk olarak 1970 yılında IBM araştırmacısı E.F. Codd tarafından önerilmiştir. Bugün hala kurumsal bilgi işlemde yaygın olarak kullanılan birçok farklı ilişkisel veritabanında uygulanmaktadır. Bu modelde, veri segmentleri tablolar kullanılarak açıkça birleştirilir ve veritabanı karmaşıklığı azaltılır. İlişkisel bir model, ilişkisel veriler içeren bağlantılar aracılığıyla birbirleriyle ilişkili düğümler içerir. Bu modeller genellikle bilgileri hızlı ve kolay bir şekilde depolamak ve almak için veritabanları

oluşturmakta kullanılır. İlişkisel veri modeline bir örnek, insanlar hakkında bilgi depolamak için kullanılan bir tablo olabilir. Tabloda ad, soyad, sosyal güvenlik numarası, doğum tarihi vb. için sütunlar olacaktır.

3. Varlık- İlişki (ER) Veri Modeli

Varlık-ilişki (ER) veri modelleri, bir veritabanındaki varlıklar arasındaki ilişkileri temsil etmek için şekilsel diyagramlar kullanır. Veri mimarları tarafından veritabanı tasarım hedeflerini aktaran görsel haritalar oluşturmak için çeşitli ER modelleme araçları kullanılır.

ER modeli verileri aşağıdaki kategorilere ayırır:

Varlıklar: Birlikte çalıştığınız nesneler, eylemler veya kavramlar. Örneğin, müşteriler, ürünler ve satışlar birer varlıktır.

İlişkiler: Varlıklar arasındaki bağlantılar. Bunlar bire bir veya bire çok ilişkiler olabilir.

Öznitelikler: Bir varlığı veya ilişkiyi tanımlayan veriler. Örneğin, bir ürünün adı o ürünün bir özneliğidir.

4. Nesne Yönelimli Veri Modeli

Nesne yönelimli veri modeli, bilgileri tanımlamak ve açıklamak için nesneleri kullanan kavramsal bir veri modelidir. Bu, bilgiyi ilişkilerle birbirine bağlı varlıklar olarak tanımlayan varlık-ilişki modelinin tersidir.

Nesne yönelimli veri modelleri, nesne yönelimli programlama olarak ilgi görmüş ve 1990'ların ortalarında popüler hale gelmiştir. İlgili “nesneler” gerçek dünya varlıklarının soyutlamalarıdır. Nesneler sınıf hiyerarşileri içinde gruplandırılır ve ilişkili özelliklere sahiptir. Nesne yöne-

limli veritabanları tablolar içerebilir, ancak daha karmaşık veri ilişkilerini de destekleyebilir.

5. Boyutsal Veri Modeli

Boyutsal veri modelleri Ralph Kimball tarafından geliştirilmiştir ve bir veri ambarında analitik amaçlar için veri alma hızlarını optimize etmek üzere tasarlanmıştır. İlişkisel ve ER modelleri verimli depolamayı vurgularken, boyutsal modeller raporlama ve erişim için bilgi bulmayı kolaylaştırmak amacıyla veri fazlalığını artırır.

Boyutsal veri modelleri, iş zekası (BI) ve çevrimiçi analitik işleme (OLAP) sistemlerinin te-

melini oluşturur. Bu modeller genellikle geçmiş işlem verilerini içeren veri ambarları için uygulanır ancak daha küçük veri kümelerine de uygulanabilir.

Boyutsal bir modelin temel amacı, kullanıcıların iş tahminleri, tüketim eğilimleri ve diğer ilgili sorularına hızlı bir şekilde yanıt bulmalarına yardımcı olmaktır. Boyutsal modelleme, iş zekası raporlaması için organize bir yöntem sağlar. Kullanıcıların etkili işbirliği ve karar verme için bir kuruluş içindeki farklı bölümler arasında bilgi paylaşmasına olanak tanır.

3.2. Veri Modellemenin Faydaları

- Veri modelleme, geliştiricilerin, veri mimarlarının, iş analistlerinin ve diğer paydaşların bir veri tabanındaki veya veri ambarındaki veriler arasındaki ilişkileri görmesini ve anlamasını kolaylaştırır.
- Yazılım ve veritabanı geliştirmedeki hataları azaltır.
- Kurum genelinde dokümantasyon ve sistem tasarımında tutarlılığı artırır.
- Uygulama ve veritabanı performansını iyileştirmede önemli bir rol oynar.
- Sistem organizasyonunda veri haritalamayı kolaylaştırır.
- Geliştiriciler ve iş zekası ekipleri arasındaki iletişimi iyileştirir.
- Kavramsal, mantıksal ve fiziksel seviyelerde veritabanı tasarımı sürecini kolaylaştırır ve hızlandırır.

BÖLÜM ÖZETİ

Sistem analistleri, sistem geliştirmenin analiz aşamasında verilerin nasıl oluşturulacağı, nasıl depolanacağı, veriler üzerinde hangi işlemlerin gerçekleştirileceği gibi teknik ayrıntıya girmeden, sistemi genel hatlarıyla ifade edebilmek için model oluştururlar. Veri modelinde sistemin görsel bir temsili oluşturulur. Benzer biçimde, sistemin gereksinimlerini modelleyebilmek için kullanım durum kullanılmaktadır. Kullanım durumları, sistem mimarisini görselleştirmenin önemli bir unsuru olarak karşımıza çıkmaktadır. Veri modelleme ise veri noktaları ve yapıları arasındaki bağlantıları iletmek için sistem parçalarının görsel bir temsilini oluşturma sürecidir.

Sistem tasarımında bir müşterinin bir ürün, hizmet veya sistemle nasıl etkileşeceğini ifade etmek, gereksinimlerin toplanması ve üst düzey paydaş iletişimi için önemlidir. Kullanım durumları, bir sistemin gereksinimlerini yakalamak, modellemek ve belirtmek için kullanılan bir tekniktir. Kullanım Durumlarının Amaçları; Kullanım Durum Modeli, Kapsam yönetimi, Gereksinimlerin belirlenmesi, Kullanıcının sistemle etkileşime gireceği yolların ana hatlarıyla belirtilmesi, Sistem mimarisini görselleştirme, Teknik gereksinimleri iş paydaşlarına iletme ve risk yönetimidir.

Kullanım durum modeli, bir sistem ile bir aktör arasındaki etkileşimlerin görsel olarak sunumudur. Kullanım durum modelleri iş süreçlerini temsil eder. Bu da ön koşulların ve tetikleyicilerin daha fazla ifade edilmesine yardımcı olur. Bir kullanım durum modeli genellikle UML ile ifade edilir. Bu modelde dört ana bileşen bulunmaktadır: aktörler, sistem, kullanım durumları ve ilişkiler. Kullanım durum diyagramı ise, temel olarak bir tür kullanım durum modelidir. Kullanım durum diyagramları sistemin akışını ve davranışını görselleştirmek, sistemin işlevselliğini örneklemek, Temel sistem-kullanıcı etkileşimlerini temsil etme durumlarında kullanılır.

Kullanım durum diyagramı oluştururken dikkat edilmesi gereken unsurlardan birisi, kullanım durum isimlendirilmesinde tercih edilen kelimelerdir. Kelimeler sıradan değil, sistemin söz konusu davranışını ve işlevini yansıtacak şekilde seçilmelidir. Kullanım durumlar isimlendirilirken genellikle emir kipi kullanılır.

Veri modelleme, veri noktaları ve yapıları arasındaki bağlantıları iletmek için tüm bir bilgi sisteminin veya sistem parçalarının görsel bir temsilini oluşturma sürecidir. Amaç, sistem içinde kullanılan ve depolanan veri türlerini, bu veri türleri arasındaki ilişkileri, verilerin gruplandırılma ve organize edilme yollarını, biçimlerini ve niteliklerini göstermektir.

Kaynakça

- Alagöz, Emre (2014), Use Case Analizi #1. Elektronik Adres: <https://tr.linkedin.com/pulse/use-case-analizi-1-emre-alag%C3%B6z>
- Bittner, Kurt (2003). Use case modeling. Spence, Ian. Addison Wesley. ISBN 0-201-70913-9. OCLC 50041546
- Daly, Nicky (2022). What Is a Use Case?. Elektronik adres: <https://www.wrike.com/blog/what-is-a-use-case/>
- IBM Cloud Education (2020). Data Modeling. Elektronik Adres: <https://www.ibm.com/cloud/learn/data-modeling>
- Jackson, Michael (1995), Software Requirements & Specifications, Addison-Wesley. ISBN 0-201-87712-0.

BÖLÜM 6

SÜREÇ MODELLEME

ANAHTAR KAVRAMLAR

Süreç Modelleme, İş Süreç Modelleme, Veri Akışı Diyagramı, Veri Akış Diyagramı Düzeyleri

ÖĞRENME HEDEFLERİ

-  1 Süreç modelleme ihtiyacını açıklar yapar
-  2 Temel kavramları kullanarak bir iş sürecini modeller
-  3 Veri akış diyagramının tanımını yapar
-  4 Bir süreci veri akış diyagramı ile ifade eder
-  5 Veri akış diyagramı türlerini açıklar

GİRİŞ

Bir bilgi sisteminin, kullanıcısı, tasarımcısı ya da genel ifade ile tüm sistem paydaşları tarafından doğru anlaşılması sistemin analizini, tasarımını ve kullanımını kolaylaştırır. Bu bölümde bir sistemin standart bir şekilde ifadesini sağlayan süreç modelleme konusu ele alınacaktır. Süreç modellemede kullanılan semboller, veri akış diyagramlarının süreç modelindeki kullanımı ve örnekler yer alacaktır.



1. SÜREÇ MODELLEME İHTİYACI

Sistemler, kolay ifade edilebilmesi, doğru analiz edilebilmesi ve ilgili tüm paydaşlar tarafından doğru algılanabilmesi için matematiksel ya da grafiksel gibi çeşitli gösterimler ile ifade edilir. Süreç modelleme, bir modelde birlikte sınıflandırılan aynı nitelikteki süreçlerin bu gösterimler ile kavramsallaştırılması olarak belirtilebilir. Geliştirilecek olan bir bilişim sisteminin iş süreçleri de analistler tarafından doğru tanımlanabilmesi için modellenmesi önemli bir husustur.

Bir bilişim sisteminin geliştirilmesi birden çok iş parçacıklarının, birden çok bileşenin ve alt sistemlerin bir araya getirilmesini içerdiği önce-

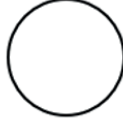
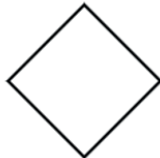
ki ünitelerde ele alınmıştır. Bu nedenle, sistemin paydaşlarının geliştirilecek olan ya da geliştirilen bir sistemin ortak dille betimlenmesi hem sistemin geliştirilmesini kolaylaştıracak hem de kavram karmaşasının önüne geçecek bir standart yapı sağlayacaktır. Örneğin geliştirilecek bir sistemdeki adımlar verinin elde edilmesini, verinin işlenmesini, depolanmasını ve çıktıya dönüştürülmesini kapsayacaktır. Bu iş adımlarının modellenmesi, ortak dil ve standart sağlayacaktır. Böylece insan kaynaklı farklı algılamaların önüne geçilecektir.

2. İŞ SÜRECİ MODELLEME

Bilişim sistemleri, bir işletmenin ya da kurumun ihtiyaçlarını gidermek için geliştirilen sistemlerdir. Dolayısıyla böyle bir bilişim sisteminin özünde işletmenin çalışması için gereken görevlerin yapısını da içermesi doğaldır. Yani, bir işletme bir tedarikçi firmadan ürün alabilir, bir müşteriye ürün satabilir, müşteriden bilgi / veri alabilir, bu verileri işleyebilir, depolayabilir ve bir çıktıya dönüştürebilir. Tüm bu görevler, basit kurallar ve adımlar içerebildiği gibi daha karmaşık iş adımlarını da içerebilir. Bu durum dolayısıyla işletme için geliştirilecek olan bilgi sisteminin analizi de zorlaştıracaktır. Bu sistemin, kullanıcısı ya da diğer sistem paydaşları tarafından doğru algılanabilmesi gerekir. Süreç modelleme bu karmaşayı kolaylaştırması beklenen bir modelleme sürecidir. Aslında süreç modelleme, süreç hakkında hiç bilgisi olmayan kişilerin de anlayabileceği bir yapıda olması gerekir.

İşletmenin iş sürecini modelleme için veri akış diyagramları ve aktivite diyagramları gibi çeşitli yöntemler vardır. Nesne yönelimli programlama standartlarından sorumlu olan Nesne Yönetim Grubu (OMG – Object Management Group), iş süreçleri için özel bir modelleme yaklaşımı oluşturmuştur. Bu model İş Süreci Mo-

delleme Gösterimi (BPMN - Business Process Modeling Notation) olarak adlandırılır. İş Süreci Modelleme Gösteriminde dört temel kavram yer almaktadır. Bu temel kavramlar Şekil 1'de gösterildiği gibi olay, etkinlik, karar noktası ve akıştır.

Şekil	İsim
	Olay
	Etkinlik
	Karar noktası
	Akış

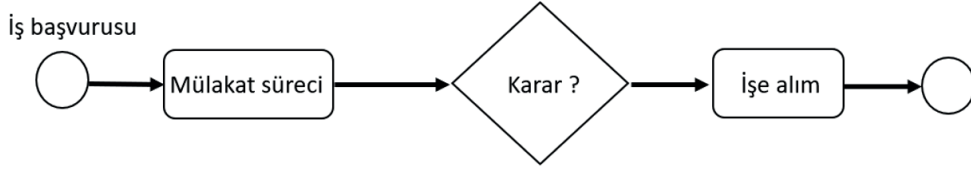
Şekil 1. İş süreci modelleme gösteriminde kullanılan dört temel kavram.

Olay : İş süreci modellemede olay, bir süreci başlatan bir tetikleyicidir.

Etkinlik : Bir sürecin tamamlanması için gerçekleşmesi gereken bir eylemdir.

Karar Noktası : İş süreci modellemede karar noktasını ifade eder.

Akış : Bir süreçteki eylemlerin sırasını gösterir.



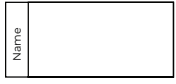
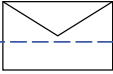
Şekil 2. İş süreci modelleme gösterimi örneği

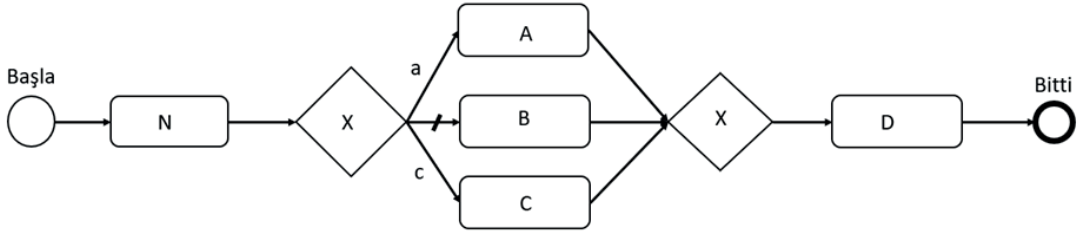
Şekil 2'deki örnekte bir iş başvuru süreci temel kavramlar kullanılarak gösterilmiştir. Model **akış** her zaman soldan sağa olacak şekilde okunur. Bir işletmenin İnsan Kaynakları bölümü için süreci başlatan tetikleyici **olay**, iş başvurusudur. İş başvurusu sonrasında mülakat süreci **etkinliği** başlar. Etkinlik sonrasında işe alımın gerçekleşip gerçekleşmeyeceği bir **karar noktası** yer alır. Karar noktası “Evet” veya “Hayır” ile dallanma gös-

terecektir. Örnekte sadece “Evet” kararı için akış gösterilmiştir. Karar noktası sonrasında işe alım süreci etkinliği gerçekleşir.

Örnekte gösterilen modelini bir şekilde ifade edemediği açıktır. Bu nedenle İş Süreci Modelleme Gösterimi diğer bileşenler ile genişletilmiştir. Tablo 1’de bu ek bileşenlerin bazıları ve açıklamaları yer almaktadır.

Tablo 1 İş süreci modelleme gösteriminin diğer bazı bileşenleri

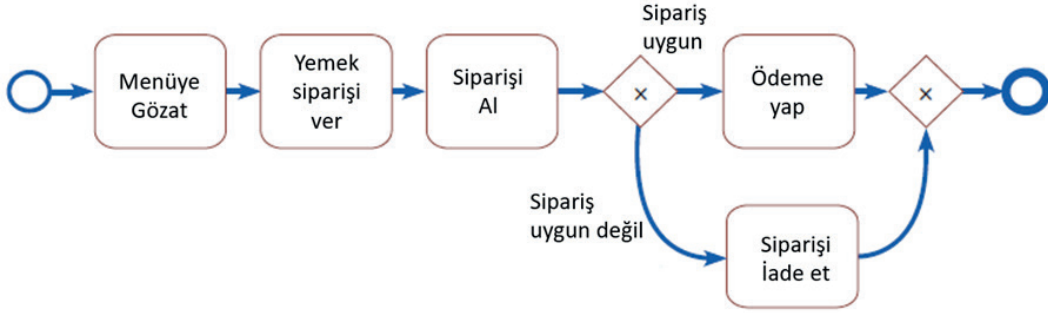
Gösterim / İsim	Açıklama
○.....> Mesaj akışı	Mesaj Akışı, mesajı göndermeye ve almaya hazırlanan iki katılımcı arasındaki mesaj akışını göstermek için kullanılır.
.....> İlişkilendirme	İlişkilendirme, bilgileri Akış Nesneleri ile ilişkilendirmek için kullanılır.
 Havuz	Havuz, bir süreçteki bir katılımcıyı temsil eder, genellikle B2B durumları bağlamında, diğer Havuzlardan bir dizi aktiviteyi bölümlenmek için kullanılır
 Veri Nesnesi	Veri Nesneleri, Sürecin Sıra Akışı veya Mesaj Akışı üzerinde herhangi bir doğrudan etkiye sahip değildir. Ancak hangi etkinliklerin gerçekleştirilmesi gerektiği ve/veya ne ürettikleri hakkında bilgi sağlayan yapılardır.
◇-----> Şartlı akış	Akışın kullanılıp kullanılmayacağını belirlemek için değerlendirme koşulu ifadelerine sahip bileşendir.
↘-----> Varsayılan akış	Varsayılan akış, yalnızca diğer tüm koşullu akış çalışma zamanında doğru değilse kullanılan akışı belirtir.
 İletişim / Bilgi	Kaynaklar ya da havuzlar arasındaki haberleşmeyi veya bilgi aktarımını ifade eder.



Şekil 3. Örnek iş akış diyagramı

Şekil 3'te yer alan örnek bir süreci göstermektedir. N faaliyeti bittikten sonra, hangi eylemin gerçekleştirileceği bir karar noktası (X) ile belirlenir. Karar noktasındaki değer "a" ise A etkinliği gerçekleştirilir. Değer "c" ise, C etkinliği gerçekleştirilir. Aksi durumda varsayılan akış gerçekleştirilir.

şir ve B etkinliği gerçekleştirilir. Ancak bu yollar-dan sadece biri takip edilebilir. A, B veya C etkinliği gerçekleştirildiğinde, tüm yollar ikinci karar noktasında birleşir ve D etkinliği gerçekleştirilir. Sürecin bittiğini ifade eden sembolün başlama sembolüne göre koyu olduğuna dikkat ediniz.



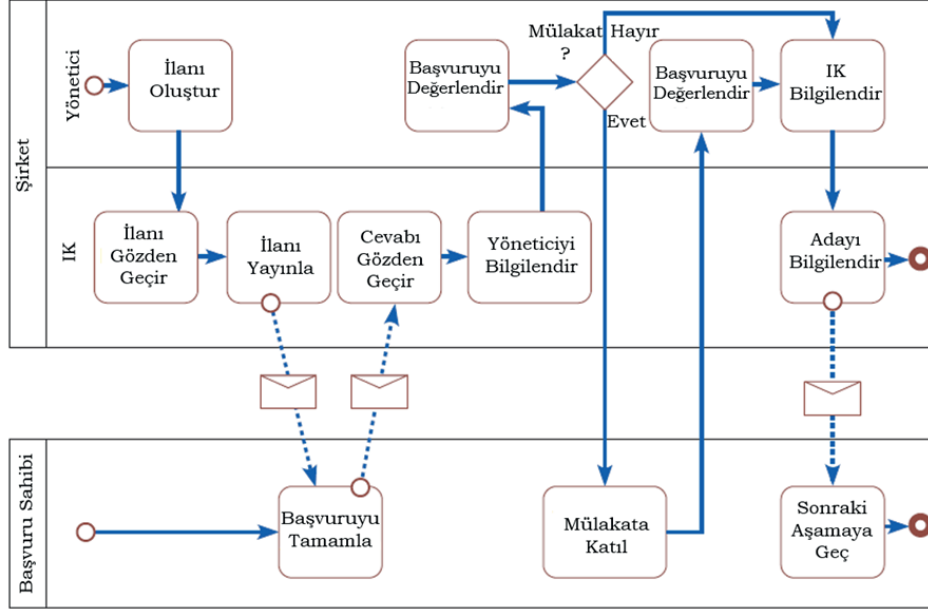
Şekil 4. Yemek siparişi iş akışı örneği

Şekil 4'teki örnekte yemek sipariş ve ödeme sürecinin basit bir modellemesi gösterilmektedir. İş sürecinin modellenmesi, sistem geliştirilmesinde analistler için kolaylık sağlayacaktır (Valacich & George, 2021).

Tablo 1'de bazı bileşenleri gösterilen İş süreci modelleme gösteriminin haricinde birçok bileşen yer almaktadır. Detaylı liste için Object Management Group'un ilgili sayfasını ziyaret edebilirsiniz.

3. BİR İŞ SÜRECİ MODELLEME ÖRNEĞİ

Şekil 5'te yer alan modelde bir iş başvurusu ve alım süreci Tablo 1'de yer alan semboller kullanılarak modellenmiştir.



Şekil 5. Örnek bir iş başvuru ve alım sürecinin modellenmesi

Gösterilen işe alım sürecinde üç katılımcı yer almaktadır. Bu katılımcılar başvuru sahibi (aday), Yönetici ve İnsan Kaynakları (İK) bölümüdür. Başvuru sahibi aday farklı bir havuzda gösterilmiştir. Yönetici ve İK bölümü aynı şirkettedir ve bu nedenle her ikisi de aynı havuzda gösterilmiştir. Şirket ile aday arasındaki tüm iletişim mesajlaşma yoluyla yapılmaktadır. Başvuru sahibi ile kuruluş arasındaki iletişimi gösteren çizgilerin noktalı çizgiler olduğuna ve orta hatta zarflara yer verdiğine dikkat edin. Zarflar mesajları veya bilgileri sembolize eder. Süreci Şirket içerisinde yer alan Yönetici başlatır. Yöneticinin bir iş için işe alım yapması gerekiyor, bu nedenle bir iş ilanı hazırlar. Akış kontrolü daha sonra İK bölümüne geçer. İK bölümü bu ilanı gözden geçirir ve ilanı yayınlar. Bu noktada, bir iş başvurusunda bulunan kişi (aday) ilanı görür ve bu başvuruyu tamamlar ve şirkete gönderir. Başvuru İK tarafından alınır, burada değerlendirilir ve yöneticiye iletir. Yönetici başvuruyu değerlendirir ve ardından başvuru sahibiyle mülakat yapıp yapılmayacağına karar vermelidir. Bu durum karar noktası sembolü ile gösterilmiştir. Eğer ka-

rar “evet” ise başvuru sahibiyle görüşülür. Karar hayır ise, başvuru sahibiyle görüşme yapılmaz. Kararın hayır olması durumunda İK’ya bu durum bildirir. İK başvuru sahibini bilgilendirir ve başvuru sahibinin bundan sonra ne yapacağına karar vermesi gerekir. Süreç hem şirket hem de başvuru sahibi için sona erer. Bu durum aday için sonraki aşamaya geç olarak gösterilmiştir. Karar “evet” ise, başvuru sahibi bir görüşmeye katılır. Sonuçlar yönetici tarafından değerlendirilir. Bu noktada yönetici başvuruları işe almaya karar verse de vermese de bu kararını İK’ya bildirir ve ardından İK adayı bilgilendirir. Başvuru sahibi bir sonraki adımına karar vermelidir ve süreç ilgili herkes için sona erer.

Şekil 5’te gösterilen bu iş sürecinde birçok adım kolay anlaşılabilirlik için kısaltılmıştır. Aslında süreç içerisinde birçok noktanın detaylandırılmasına ihtiyaç duyulabilir. Örneğin adayın bilgilerinin nasıl alınacağı, nerede saklanacağı gibi veri üzerinde yapılacak işlemlerin de belirlenmesi, süreç modellemenin önemli bir parçasıdır. Bu nedenle süreç modellemede veri akışının da betimlenmesi gerekir.

4. VERİ AKIŞ DİYAGRAMLARI

Bilgi sisteminin tanımına bakıldığında, veriler toplandığı, depolandığı, işlendiği ve anlamlı bilgilere dönüştürüldüğü görülür. Bu nedenle verinin hareket ettiği ve şekil değiştirdiği söylenebilir. Bu nedenle verinin akışı da bir diyagram ile modellenenebilir. Sistem analistleri, bir bilgi sistemini tanımlamak için birçok grafik tekniği kullanır. En yaygın yöntemlerden biri bu süreci veri akış diyagramı (VAD) ile ifade etmektir. Sistem Geliştirme Yaşam Döngüsü'nün analiz aşamasında, sistem analisti, bir dizi veri akış diyagramı kullanarak bilgi sisteminin görsel bir modelini oluşturur (Satzinger et al., 2016). Veri akış diyagramları bu görsel modellerden biridir. Veri akış diyagramla-

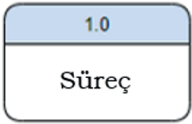
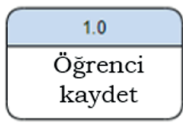
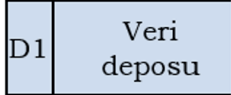
rında, verilerin bir bilgi sisteminde nasıl hareket ettiği gösterilir, ancak program mantığı veya işleme adımlarını gösterilmez. Yani, sistemin nasıl yaptığını değil ne yaptığını gösteren mantıksal bir modeldir. Mantıksal model, fiziksel olarak nasıl uygulanacağından bağımsız olarak sistemin ne yapması gerektiğini gösterir. Daha sonra sistem tasarım aşamasında, sistemin nasıl kurulacağını açıklayan fiziksel bir model oluşturulur (Tilley, 2019). Özetle veri akış diyagramları sistemin girdilerini yararlı bilgilere nasıl dönüştürdüğünü göstermek için çeşitli semboller kullanan bir grafiksel gösterim aracıdır.

5. VERİ AKIŞ DİYAGRAMININ SEMBOLLERİ

Veri akış diyagramları süreçleri, veri akışlarını, veri depolarını ve varlıkları temsil eden dört temel sembol kullanır. VAD sembollerinin birkaç farklı versiyonu bulunmaktadır, ancak hepsi aynı amaca hizmet eder. Bu bölümde Gane & Sarson gösterimi dikkate alınacaktır. Bu gösterimi, Chris Gane ve Trish Sarson'un Structured Systems

Analysis: Tools and Techniques adlı kitabında tanıtmıştır. Bu tanımlamada kullanılan semboller, isimleri ve örnek gösterimleri Tablo 2'de yer almaktadır. Bütün bir bilgi sistemi veya çok sayıda alt sistem, bu dört sembol kombinasyonu ile grafiksel olarak gösterilebilir.

Tablo 2. Gane & Sarson gösterimine göre temel veri akış diyagramı sembolleri

Sembol	İsim	Örnek gösterim
	Varlık (Entity)	
	Veri akışı (Data flow)	
	Süreç (Process)	
	Veri deposu (Data store)	
	Harici varlık (External Entity)	

Varlık ve Harici Varlık : Varlıklar bir bilgi sisteminin dışında yer alan ve sistemle etkileşime giren insanları, kuruluşları ve yapıları ifade eder. Sistemin dışında yer almasına rağmen sistemin birer parçalarıdır. Yani sisteme bilgi sağlarlar ve sistemden bilgi alırlar. Bir veri akış diyagramında yer alan Varlık, dikdörtgen ile gösterilir ve içine varlığın ismi yazılır.

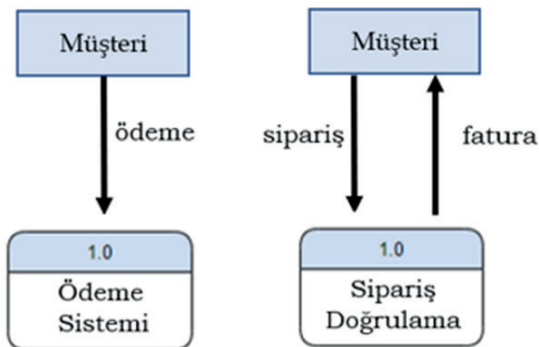
Veri akış diyagramları, sistemin sınırlarını ve sistemin dış dünya ile nasıl arayüz oluşturduğunu gösterir. Örneğin, bir müşteri varlığı, bir sipariş işleme sistemine bir sipariş gönderir. Ya da bir öğrenci varlığı, sisteme not girişi yapar. Bu müşterinin ya da öğrencinin sistemle etkileşimi arayüz aracılığıyla gerçekleştirilir.

Benzer şekilde, bir hasta kayıt sistemine veri girişi yapan bir hasta, bir online alışveriş bilgi sisteminden fatura alan bir kullanıcı veya şirketin satın alma sisteminden veri alan bir muhasebeci varlık örneği olarak gösterilebilir.

Varlıklar, verinin kaynağı olabildikleri gibi verinin hedefi de olabildiklerinden bazen sonlandırıcılar (terminators) olarak da adlandırılırlar. Ayrıca her varlık, bir veri akışıyla bir sürece bağlanmalıdır. Şekil 6'da doğru gösterim örnekleri yer almaktadır.

Şekil 6. Her varlık bir veri akışıyla bir sürece bağlanmalıdır.

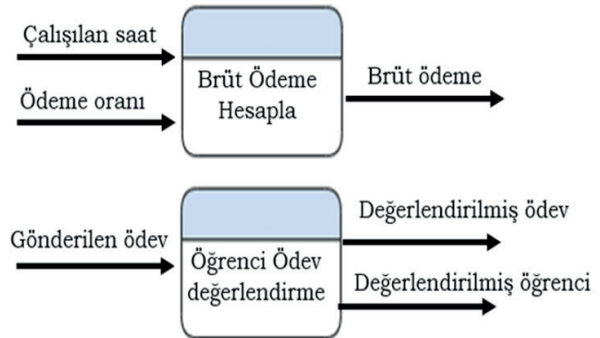
Şekilde iki farklı doğru örnek gösterilmektedir.



Veri akışı : Verilerin nereye doğru hareket ettiğini belirtmek için kullanılan oklu çizgilerdir. Veri akışının simgesi, tek veya çift ok başlı bir

çizgidir. Veri akış çizgileri, tüm bilgileri birbirine bağladıkları için veri akış şemasında önemli bir bileşendir. Ayrıca, her veri akışının hangi görevi yaptığını kısaca tanımlayan etiketlerle işaretlenirler. Şekil 5'te "ödeme", "sipariş" ve "fatura" ile bu etiketleme örnekleri gösterilmiştir.

Veri akışı simgesi gibi birden fazla giden veri akışına veya birden fazla gelen veri akışına sahip olabilir. Şekil 7'de bu iki duruma örnekler gösterilmiştir.

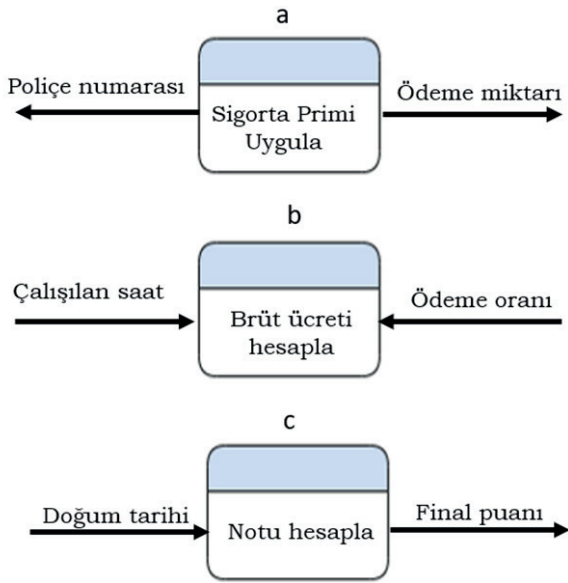


Şekil 7. Veri akışı birden fazla gelen ya da giden şekilde olabilir.

Süreç : Verileri doğrudan değiştiren eylemlerdir. Bir eylem gerçekleştiğinde, bu eyleme dayalı olarak yeni bir çıktı oluşturulur. Tipik süreçler, bir işletmenin izleyebileceği hesaplamaları, veri sıralamayı veya belirli kuralları içerir. Süreç, sürecin ne yaptığını tanımlayan tek bir kelime veya basit cümle ile adlandırılmalıdır. Genellikle "Kontrol Et", "Hesapla" gibi bir cümleden oluşur. Süreçlere üstte bir numara verilir. Bu numara bir tanımlayıcıdır, sıra anlamını taşımaz. Şekil 6 ve Şekil 7'de doğru kullanım örnekleri yer almaktadır. Şekil 8'de hatalı süreç kullanım örnekleri bulunmaktadır. Sıklıkla yapılan hatalı kullanım örneklerinden biri, "spontane üretim" olarak adlandırılan hatadır. Spontane üretimde girdi olmadan kendiliğinden çıktılar yer alır. Şekil 8(a)'da yer alan örnekte, Sigorta Primi Ödeme sürecinin girdisi olmamasına rağmen Poliçe numarası ve ödeme miktarı çıktısı yer almaktadır. Bu hatalı bir gösterimdir. Şekil 8 (b)'de yer alan diğer hatalı gösterim, çıktısı olmayan bir süreci

ifade eden “kara delik” olarak adlandırılan hatadır (Dennis et al., 2019). Şekil 8(c)’de yer alan hatalı gösterim, “gri delik” olarak adlandırılır. Bu hatalı gösterimde ilgisiz veri girdisine dayalı veri çıktısı yer alır. Örnekte, öğrenci notunun hesaplanmasında doğum tarihinin etkisi olmayacağı açıktır.

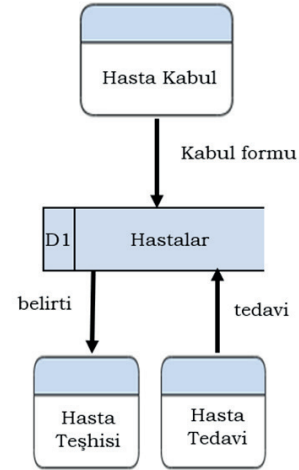
Şekil 8. *Hatalı süreç kullanım örnekleri. (a) Spontane üretim, (b) Kara delik, (c) Gri delik örnekleridir.*



Veri deposu : Veri depoları, ileride çağrılabilir bilgileri saklayan dosyalar veya depolama alanlarıdır. Genellikle veritabanı tabloları veya elektronik tablolar olarak görünürler. Süreçler gibi, veri depoları da amaçlarını veya düzenlerini açıklayan basit etiketler kullanır. Örneğin, bir öğrenci otomasyon sisteminde öğretmenlerin dönem sonunda final notlarını verebilmeleri için öğrencilerin dönem boyunca yaptıkları testlere ve ödevlere ilişkin puanları saklamaları gerekir. Veya bir şirket, yaptığı harcamaları kaydedip, yıl sonunda ortalama maliyet hesaplarını yapmak isteyebilir. Veri depoları bu amaç için kullanılan gösterimlerdir. Gane & Sarson gösteriminde, sağ tarafı açık ve sol tarafı kapalı olan düz bir dikdörtgendir. Veri deposunun adı bu dikdörtgenin içine yazılır ve içerdiği verileri tanımlar.

Veri deposunda verinin ne tür bir veri olduğu, verinin uzunluğu gibi detaylar yer almaz. Her ne kadar bu bilgiler sistemin fiziksel geliştirilmesi aşamasında önemli bilgiler olsa da veri akış diyagramında yer almazlar.

Şekil 9. *Veri deposunun kullanımına ilişkin bir örnek.*

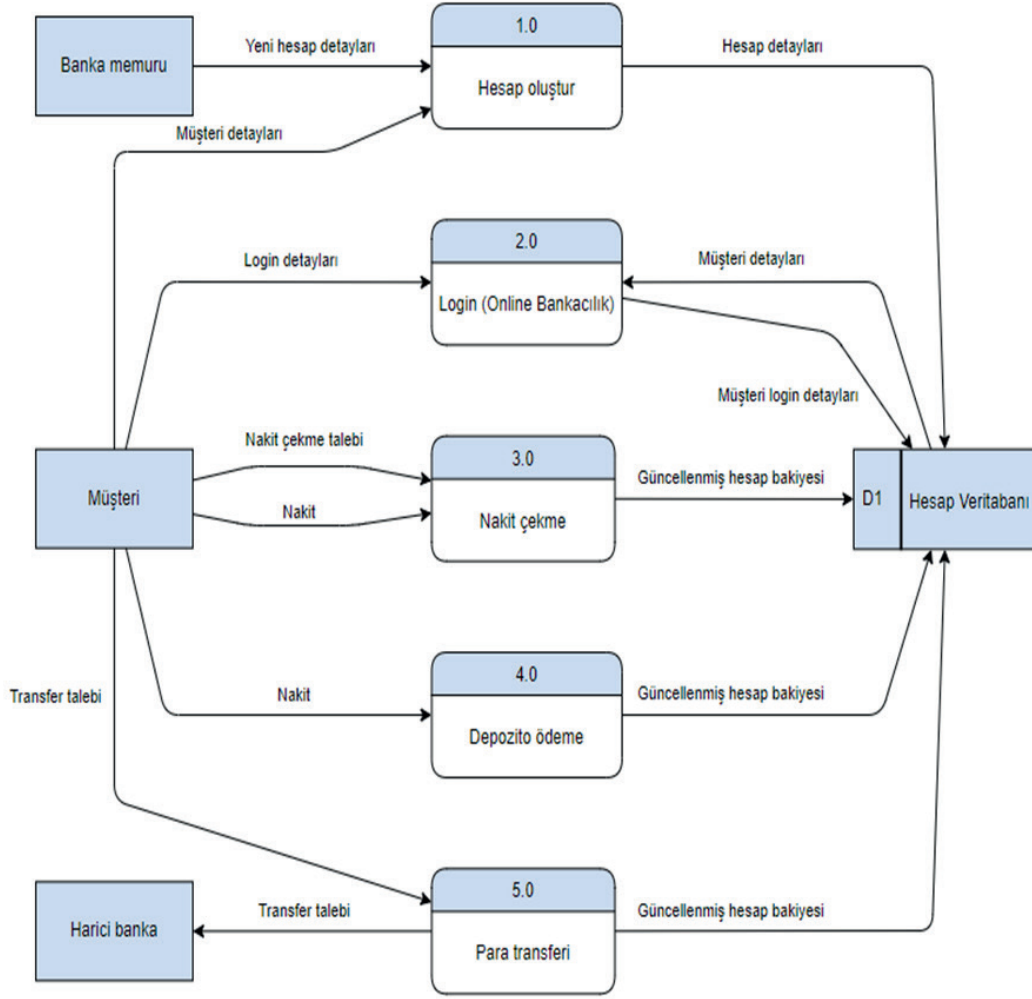


Şekil 9’da yer alan örnekte bir hasta için süreç ve veri deposunun nasıl kullanılabileceği örneklenmiştir. Örnekte hasta kabul süreci sonrasında hasta kabul formuna göre “hastalar” veri deposuna kayıt işlemi gerçekleştirilir. Veri deposunda yer alan belirtilere göre hastalık teşhisi süreci başlayacaktır. Tedavi sürecinde yapılan işlemler de yine veri deposuna aktarılır.

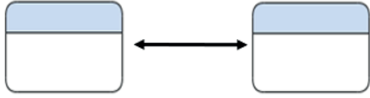
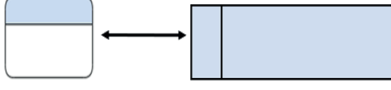
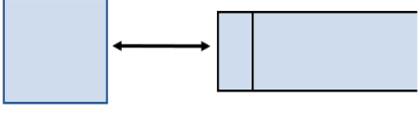
Veri akış diyagramı sembollerinin doğru anlaşılmasının ardından bir bilgi sistemini doğru tanımlayan süreç modellemesi yapılabilir. Şekil 10’da iş süreci doğru şekilde modellenmiş bir veri akış diyagramı örneği yer almaktadır. Örnekte bir banka müşterisinin süreç modellemesi, modellemede kullanılan süreçler, veri akışları, varlıklar ve veri depoları örneklerini görebilirsiniz (İnternet Sayfası, Visual Paradigm Online).

Süreç modellemesi yapılırken yaygın yapılan hatalar sürecin anlaşılmasını güçleştirir. Bu nedenle model oluşturulurken bu hatalardan kaçınmak gerekir. Tablo 3’te hatalı ve doğru bazı kullanım örnekleri yer almaktadır (Shelly & Rosenblatt, 2012).

Şekil 10. Veri akış diyagramlarının doğru kullanımını gösteren örnek bir süreç modellemesi



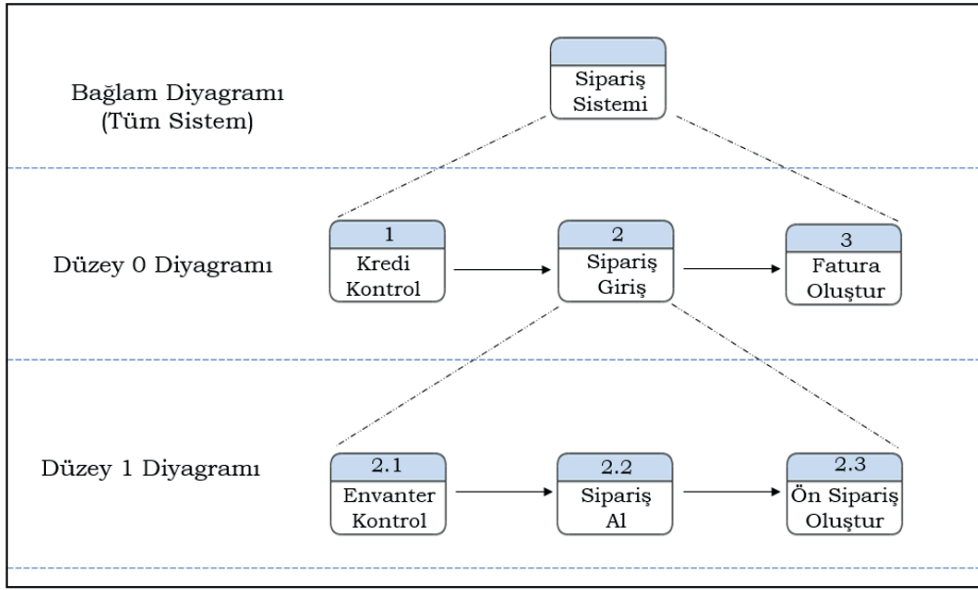
Tablo 3. Veri akış diyagramında doğru ve yanlış kullanım örnekleri

Gösterim	Açıklama	Doğru / Yanlış
	Süreçten sürece akış	Doğru
	Süreç ile harici varlık arası akış	Doğru
	Süreç ile veri deposu arası akış	Doğru
	Harici varlıklar arası akış	Yanlış
	Harici varlık ile veri deposu arasındaki akış	Yanlış
	Veri depoları arasındaki akış	Yanlış

6. VERİ AKIŞ DİYAGRAMI DÜZEYLERİ

Veri akış diyagramları bir işletmenin iş sürecini modellemede kullanılabilir. Süreç modellemesi hiyerarşik bir şekilde katmanlı yapıda gerçekleştirilir. Süreç modellemesinde öncelikle bütün sistemi temsil eden bağlam diyagramı ile başlar. Bağlam diyagramı en üst düzeydir. “Ödeme Sistemi”, “Öğrenci Otomasyon”, “Hastane Otomas-

yon Sistemi” gibi isimlendirmelere örnek olarak verilebilir. Bağlam diyagramının ardından bağlamın alt sistemlerini temsil eden, Düzey 0 diyagramı çizilir. Düzey 0 diyagramı bir veya daha fazla süreci içerebilir. Düzey 0 diyagramının altında Düzey 1 diyagramı çizilir.



Şekil 11. Çoklu düzeyden oluşan örnek bir veri akış diyagramı

Şekil 11’de çoklu düzeyden oluşan örnek bir veri akış diyagramı gösterilmektedir. Diyagramın en başında tüm sistem tanımlanmıştır. Örnekte bağlam diyagramı “Sipariş Sistemi” olarak gösterilmiştir. Sipariş Sistemi’nin altında Düzey 0 diyagram yer almaktadır. Bu düzeyde var olan süreçler akış çizgileri ile bağlanmış ve her bir süreç benzersiz bir numara ile kimliklendirilmiştir. Düzey 1 diyagramında, üst düzeyde yer alan süreçler detaylandırılmıştır. Yine örnekte 2.süreç olan “Sipariş Giriş” süreci hiyerarşik yapıda Envanter Kontrol, Sipariş Al ve Ön Sipariş oluş-

turma süreçlerinden oluştuğu belirtilmiş ve veri akışları ifade edilmiştir. Düzey 1’de yer alan süreçler Sipariş Giriş sürecinin alt süreçleri olduğu düşünülürse, numaralandırma 2.1, 2.2 şeklinde devam eder. Benzer şekilde tüm süreç hiyerarşik bir yapıda veri akış diyagramları ile ifade edilebilir. Eğer Düzey 1’de yer alan herhangi bir süreç de alt süreçlerden oluşacaksa, bir sonraki Düzey 2 diyagram çizilmelidir. Örneğin Envanter Kontrol süreci Düzey 2’de tanımlanacaksa oluşturulacak alt süreçlerin numaralandırmaları 2.1.1, 2.1.2 şeklinde devam ettirilmelidir.

BÖLÜM ÖZETİ

Bilgi sistemleri; kolay ifade edilebilmesi, doğru analiz edilebilmesi ve ilgili tüm paydaşlar tarafından doğru algılanabilmesi için grafiksel gösterimler ile ifade edilir. Süreç modelleme, bir modelde birlikte sınıflandırılan aynı nitelikteki süreçlerin bu gösterimler ile kavramsallaştırılması olarak tanımlanabilir. Geliştirilecek bir sistemdeki adımlar, verinin elde edilmesini, verinin işlenmesini, depolanmasını ve çıktıya dönüştürülmesini kapsar. Bu iş adımlarının modellenmesi, ortak dil ve standart sağlamaktadır.

Nesne yönelimli programlama standartlarından sorumlu olan Nesne Yönetim Grubu, iş süreçleri için özel bir modelleme yaklaşımı oluşturmuştur. Bu model, İş Süreci Modelleme Gösterimi olarak adlandırılır. İş Süreci Modelleme Gösteriminde dört temel kavram yer almaktadır. Bunlar; olay, etkinlik, karar noktası ve akıştır. Bu temel dört kavram dışında modellemede kullanılan çeşitli semboller yer almaktadır.

Veri üzerinde yapılacak işlemlerin belirlenmesi, süreç modellemenin önemli bir parçasıdır. Bu nedenle süreç modellemede veri akışının da betimlenmesi gerekir. Veri akış diyagramları bu görsel betimleme modellerinden biridir.

Varlıklar, bir bilgi sisteminin dışında yer alan ve sistemle etkileşime giren insanları, kuruluşları ve yapıları ifade eder. Sistemin dışında yer almasına rağmen sistemin birer parçalarıdır. Yani sisteme bilgi sağlarlar ve sistemden bilgi alırlar. Bir veri akış diyagramında yer alan varlık, dikdörtgen ile gösterilir ve içine varlığın ismi yazılır.

Veri akışı: Verilerin nereye doğru hareket ettiğini belirtmek için kullanılan ok işaretli çizgilerdir. Veri akış çizgileri, tüm bilgileri birbirine bağladıkları için veri akış diyagramında önemli bir bileşendir.

Süreç: Verileri doğrudan değiştiren eylemlerdir. Bir eylem gerçekleştiğinde, bu eyleme dayalı olarak yeni bir çıktı oluşturulur. Tipik süreçler, bir işletmenin izleyebileceği hesaplamaları, veri sıralamayı veya belirli kuralları içerir. Süreç, sürecin ne yaptığını tanımlayan tek bir kelime veya basit cümle ile adlandırılmalıdır.

Veri deposu: Bilgileri saklayan dosyalar veya depolama alanlarıdır. Genellikle veritabanı tabloları veya elektronik tablolar olarak görünürler.

Veri akış diyagramları, bir işletmenin iş sürecini modellemede kullanılabilir. Süreç modellemesi hiyerarşik bir şekilde katmanlı yapıda gerçekleştirilir.

GÖZDEN GEÇİRELİM

1. Hangisi iş süreci modellemede kullanılan temel kavramlardan biri değildir?

- a. Olay
- b. Etkinlik
- c. Varlık
- d. Karar noktası

2. Bir süreç modelindeki eylemlerin akış sırasını gösteren kavram hangisidir?

- a. Akış
- b. Olay
- c. Etkinlik
- d. Girdi/çıkış

3. Bir iş sürecini başlatan tetikleyici hangi kavram ile ifade edilir?

- a. Akış
- b. Olay
- c. Etkinlik
- d. Havuz

4. Bir süreçteki bir katılımcıyı temsil eder. Genellikle B2B durumları bağlamında, diğer bir dizi aktiviteyi bölümlenmek için kullanılır.

Yukarıdaki tanımlama aşağıdakilerden hangisine aittir?

- a. Karar noktası
- b. Olay
- c. Akış
- d. Havuz

5.



Şekilde gösterilen veri akış diyagramı bileşeni aşağıdakilerden hangisidir?

- a. Süreç
- b. İnsan
- c. Akış
- d. Varlık

6.



Şekilde gösterilen veri akış bileşeni aşağıdakilerden hangisinin bir örneğidir?

- a. Süreç
- b. Varlık
- c. Veri deposu
- d. Harici varlık

7. Nesne Yönetim Grubu (OMG) İş Süreci Modelleme Gösterimine göre hangisi varsayılan akışı gösterir?



8. Sistemin girdilerini yararlı bilgilere nasıl dönüştürdüğünü göstermek için çeşitli semboller kullanan bir grafiksel gösterim aracıdır.

Yukarıdaki ifade hangisini tanımlar ?

- a. Süreç modelleme
- b. Veri akış diyagramı
- c. Proje yönetimi
- d. Algoritma

10. Hiyerarşik veri akış diyagramı çiziminde en tepede hangisi yer alır?

- a. Düzey 0 diyagramı
- b. Düzey 1 diyagramı
- c. Bağlam diyagramı
- d. Üst düzey diyagram

9. Yukarıdaki veri akış diyagramı gösterimlerinden hangileri hatalıdır?

- a. Yalnız I
- b. I ve II
- c. II ve III
- d. I ve III

Yanıt Anahtarı: 1-C , 2-A, 3-B, 4-D, 5-D, 6-C, 7-B, 8-B, 9-D, 10-C

Kaynakça

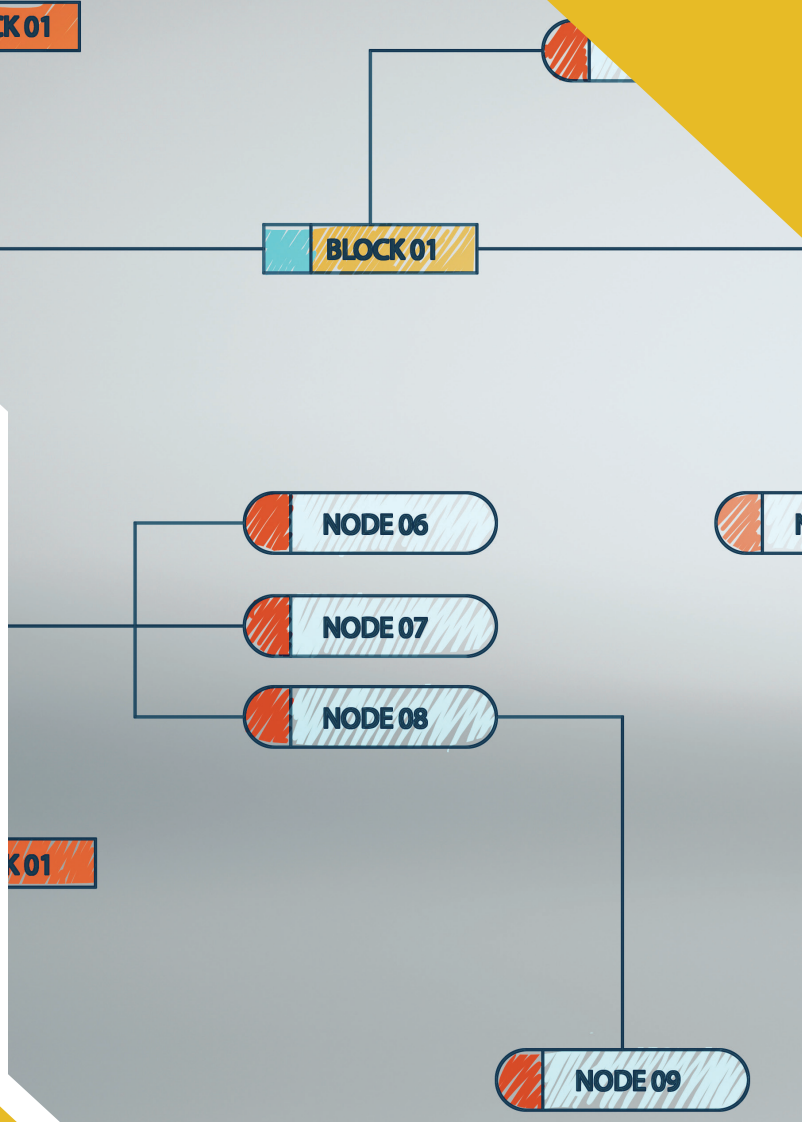
- Dennis, A., Wixom, B. H., & Roth, R. M. (2019). System Analysis and Design (7th ed.). Wiley.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). System Analysis and Design in a Changing World. In In Course Technology Cengage Learning (7th ed.). Cengage Learning.
- Shelly, G. B., & Rosenblatt, H. J. (2012). System Analysis and Design. In News.Ge (10th ed.). Shelly Cashman Series.
- Tilley, S. (2019). Systems Analysis and Design. In Shelly Cashman Series.
- Valacich, J. S., & George, J. F. (2021). Modern Systems Analysis and Design, 9th Edition.
- İnternet Sayfası, Visual Paradigm Online, <https://online.visual-paradigm.com/knowledge/software-design/gane-sarson-dfd-tutorial> Son erişim: 01.11.2022

BÖLÜM 7

NESNE YÖNELİMLİ MODELLEME

ANAHTAR KAVRAMLAR

Nesne, Sınıf, Katılım, Çok Bizimlilik,
Kapsülleme, UML

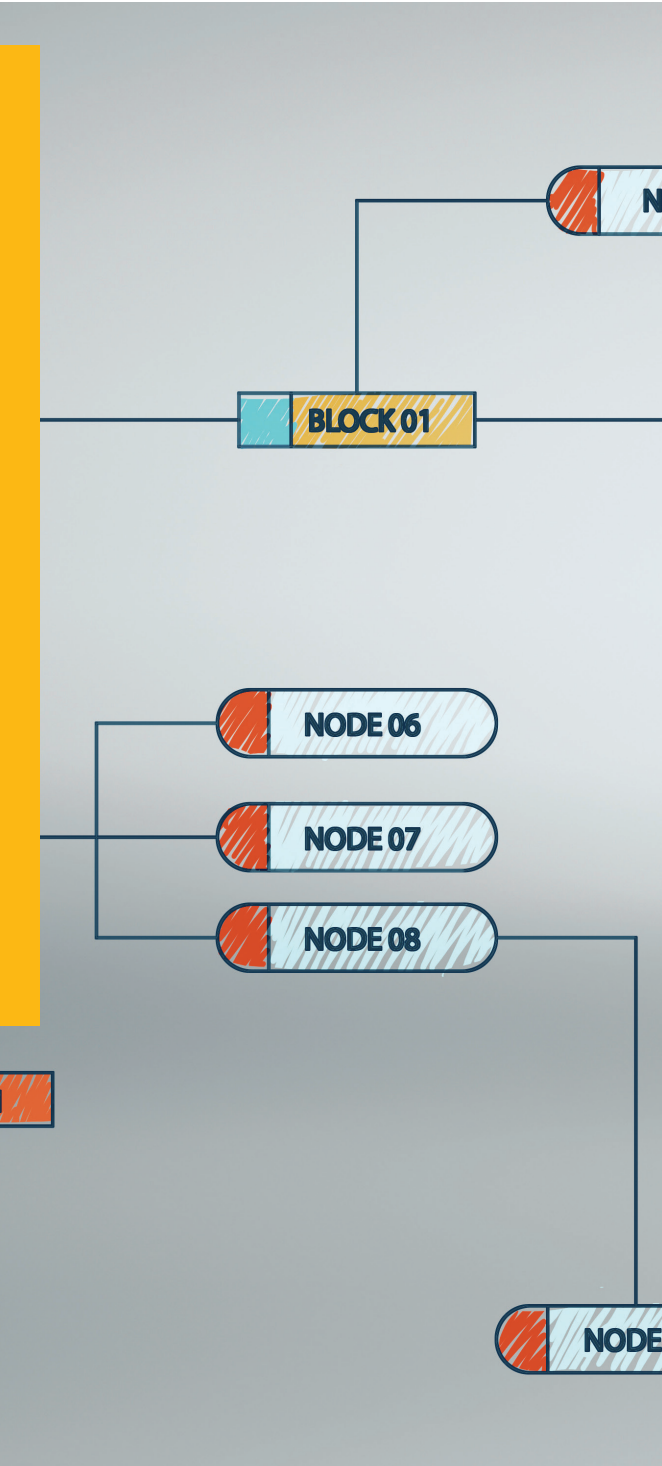


ÖĞRENME HEDEFLERİ

-  1 Nesne tabanlı modelleme kavramını açıklayabilecek
-  2 Nesne tabanlı modelleme temel kavramlarını açıklayabilecek
-  3 UML diyagramlarını ve çeşitlerini tanımlayabilecek
-  4 Nesne tabanlı modelleme sürecini açıklayabilecek

GİRİŞ

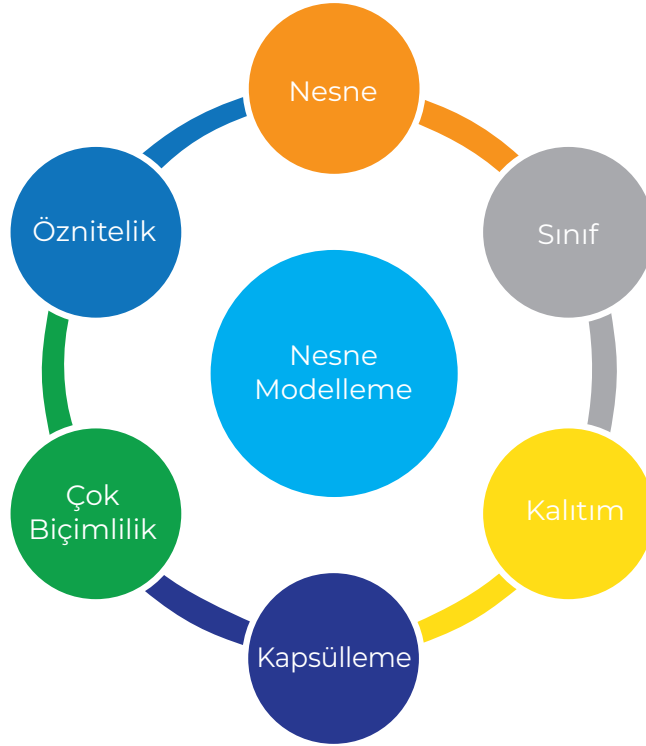
Dijital ortamlar ve uygulamalar hayatımızın vazgeçilmez bir parçası olmuştur. Teknolojideki hızlı gelişim ile birlikte kurum-kuruluşlar, özel şirketler, sivil toplum kuruluşları gibi birçok yapı için bahsi geçen ortamları ya da uygulamaları kullanma zorunluluğu oluşmuştur. Dolayısıyla sistem ve yazılım geliştirme süreçleri çok önemli bir hale gelmiştir. Nesne tabanlı modelleme de yazılım geliştirme ve sistem tasarımı yaklaşımlarının temelini oluşturan bir kavramdır. Nesne yönelimli modelleme, nesne ve sınıflarının tanımlanmasıyla karakterize edilen bir görselleştirme sürecidir. Başka bir ifade ile nesne ve sınıfları temel alan bir yaklaşımdır. Bahsi geçen yaklaşımın daha hızlı yürütülmesi, daha kolay bir yapı sağlaması, tasarlanacak sistemin bakımının, güncellenmesinin ya da hata ayıklamasını kolaylaştırması, daha kısa sürede kullanılabilir uygulamalar geliştirmeye olanak sunması en önemli avantajlarından. Bu avantajlardan dolayı geleneksel modelleme yaklaşımlarına alternatif bir seçenek olmuştur.



1. NESNE MODELLEMeye İLİŞKİN TEMEL KAVRAMLAR

Nesne tabanlı tasarım ve modelleme süreçlerini gerçekleştirmek için bazı temel kavramları öğrenmek gerekmektedir. Bu kavramların bilinmesi, ilgili sürecin anlaşılması için önemlidir. Nesne tabanlı modelleme geleneksel yöntemle-

rin aksine, davranış modelleri oluşturmaktadır. Davranış modellerini oluşturmak adına gerekli temel kavramlar Şekil 1'de verilmiş olup takip eden bölümde tanımlamaları yapılmıştır.



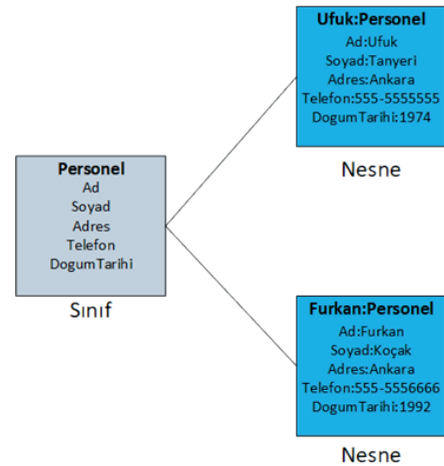
Şekil 1. Nesne tabanlı modelleme temel kavramları

Sınıf:

Nesne yönelimli tasarımın temelini sınıflar oluşturmaktadır. Kendi veri üyelerini ve üye fonksiyonları olan bir veri türüdür. Sınıf, bir nesnenin planı gibidir. Bir başka ifade ile sınıf, veri üyelerine ve üye işlevlerine sahip olan veri türüdür. Sınıflar genel olarak nesnelerin genel özelliklerini tanımlarlar.

Nesne:

Nesneye yönelik tasarımlarda, nesneler sınıflardan türetilirler. Nesne, bir sınıfın örneğidir. Sınıfın tüm veri üyelerine ve üye işlevlerine nesneler yardımıyla erişilebilir. Aynı sınıftan türetilmiş nesnelerin sahip olduğu özellikler benzerdir. Şekil 2'de nesne ve sınıf kavramlarının ilişkileri verilmiştir.



Şekil 2. Sınıf ve Nesne ilişkisi

Söz konusu şekil incelendiğinde Personel adında bir sınıf tasarımı yapılmıştır. Personel sınıfı genel olarak bir şablon tasarımıdır. Ad, soyad

vb. gibi değişkenler Personel sınıfının veri üyelerini oluşturmaktadırlar. Ufuk ve Furkan yapıları ise Personel sınıfından türetilen nesneleri temsil etmektedirler. Her iki nesnede sınıfa ait özellikleri taşımaktadırlar. Dolayısıyla nesneler sınıfların şablon tasarımını kullanan yapılardır. Sınıflar soyut bir veri tipi iken, nesneler ise onun somutlaşan halidir. Nesneler ancak sınıflar türetildikten sonra oluşturulurlar.

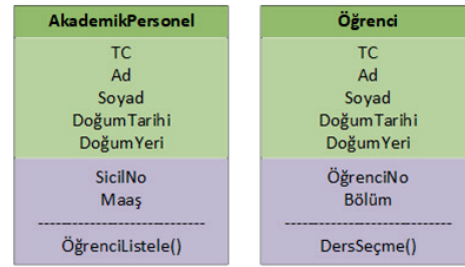
Öznitelik:

Sınıflardan oluşturulan nesneleri tanımlamaya yarayan özelliklere öznitelik denir. Şekil 2’de verilen Furkan ve Ufuk nesnelerinin öznitelikleri ad, soyad, adres, telefon ve doğum tarihidir.

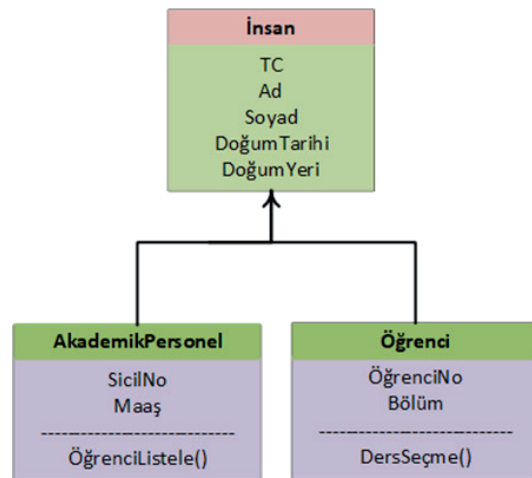
Kalıtım:

Nesne yönelimli modellemede kalıtım, bir nesneyi veya sınıfı, başka bir nesnenin veya sınıfın özelliklerini kullanarak alt sınıf ya da alt nesneler oluşturma mekanizması olarak tanımlanmaktadır. Temel sınıf gibi mevcut sınıflardan yeni sınıflar-alt sınıflar türetmek ve ardından bunları bir sınıf hiyerarşisi halinde oluşturmak olarak da tanımlanır. Kalıtım yoluyla oluşturulan bir sınıf, temel sınıfın tüm özelliklerini ve davranışlarını alır. Kalıtımın ana avantajı, sistemin yeniden kullanılabilirliğine yardımcı olmasıdır. Sınıflar yalnızca bir kez tanımlanır ve birden çok kez kullanılabilir. Kalıtım yoluyla zaman tasarrufu sağlanmaktadır ve okunabilirlik yapısı geliştirilmektedir. Kalıtım ile oluşturulan sınıfların ortak veri üyeleri temel sınıf ya da üst sınıfta bulunmaktadır. Bu yapıya genelleşme denir. Alt sınıflar ayrıca temel sınıfların veri üyelerinin dışında farklı öznitelik ya da fonksiyonlara sahip olabilmektedir. Bu durum ise özelleşme olarak tanımlanmaktadır. Örneğin bir İnsan temel sınıfından türetilmiş Öğrenci ve AkademikPersonel isimlerinde alt sınıflar düşünebiliriz. Öğrenci ve AkademikPersonel alt sınıfları İnsan temel sınıflarının özelliklerini taşımaktadır. Bu durum ge-

nelleşme kavramına örnek teşkil ederken, bahsi geçen Öğrenci ve AkademikPersonel alt sınıflarının temel sınıftan farklı veri üyeleri ve fonksiyonlara sahip olabilmesi özelleşmeye örnektir. Şekil 3 AkademikPersonel ve Öğrenci sınıflarını göstermektedir. Her iki sınıfta ortak özellikleri bulunmaktadır. Şekil 4 ise kalıtım yapısını temsil etmektedir. Şekil 4’te AkademikPersonel ve Öğrenci sınıfları alt sınıfları gösterirken, İnsan sınıfı ise temel sınıfı temsil etmektedir. Alt sınıfların ortak özellikleri temel sınıf altında toplanmıştır. Kalıtım yoluyla ise alt sınıflara farklı özellikler ve fonksiyonlar eklenmiştir. Şekillerden de anlaşılacağı üzere ortak öznitelikleri bir defa tanımlamak yeterli olacaktır. İlaveten İnsan temel sınıfı üzerinde yapılacak herhangi bir güncelleme işlemi her iki alt sınıfta da uygulanacaktır. Dolayısıyla bu işlem zaman ve hesaplama maliyetini azaltan bir işlemdir.

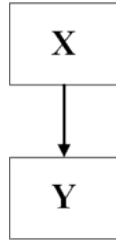


Şekil 3. Kalıtım kullanılmadan oluşturulan sınıflar

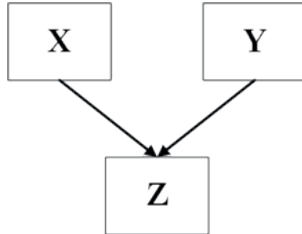


Şekil 4. Kalıtım yoluyla oluşturulan sınıflar

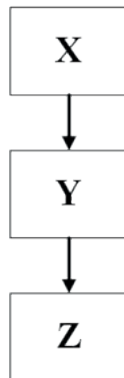
Kalıtım işlemi, **tekli kalıtım**, **çoklu kalıtım**, **çok seviyeli kalıtım**, hiyerarşik kalıtım ve hibrit kalıtım olmak üzere 5 çeşittir. Tekli kalıtımda alt sınıf tek bir üst sınıfın özelliklerini taşıırken, çoklu kalıtımda birden fazla üst sınıfın özelliklerini taşımaktadır. Bir diğer kalıtım çeşidi olan çok seviyelide ise bir sınıfın bir alt sınıfı oluşturulduktan sonra, bu alt sınıftan başka bir alt sınıf oluşturulur. Hiyerarşik kalıtım ise bir sınıfın birden fazla alt sınıfa temel sınıf yapmasıdır. Son olarak hibrid kalıtım ise yukarıda bahsi geçen kalıtım mimarilerinden iki ya da daha fazlasının kullanılmasıdır. Şekil 5, Şekil 6, Şekil 7 ve Şekil 8 sırasıyla tekli, çoklu, çok seviyeli ve hiyerarşik kalıtım çeşitlerini simgelemektedir.



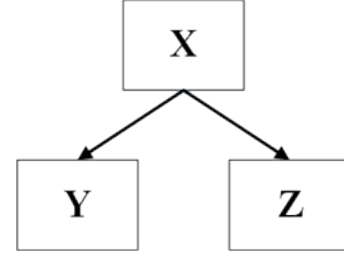
Şekil 5. Tekli kalıtım



Şekil 6. Çoklu kalıtım



Şekil 7 Çok seviyeli kalıtım



Şekil 8 Hiyerarşik kalıtım

Kapsülleme

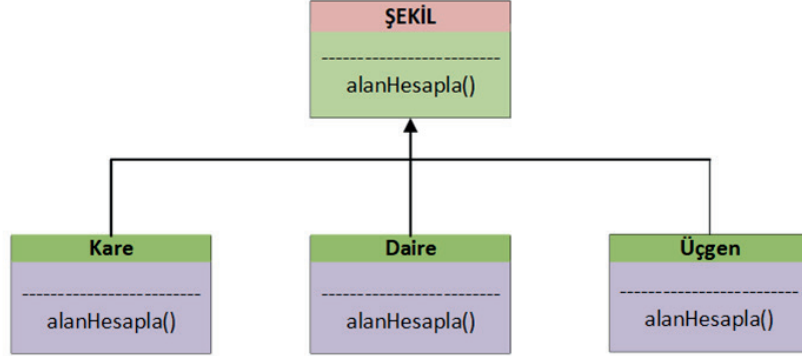
Nesne tabanlı modellemenin temel kavramlarından biri olan kapsülleme, kullanıcıların sınıf, metot verilerin ne kadarına erişebileceğini belirler. Başka bir ifade ile sınıf, metot ya da verilere doğrudan erişimlerin kısıtlanması ve izin verilmesi gibi işlemleri gerçekleştirmek için kullanılır. Erişim izinleri erişim belirleyici anahtar kavramları ile oluşturulurlar. Bu kavramlar **public**, **private**, **protected**, **internal** olarak isimlendirilir. Public olarak tanımlanan bir nesne modelleme bileşeninin (sınıf, metot ya da veri) erişim kısıtlaması yoktur ve her yerden erişilebilirler. Bir üye private olarak tanımlanmış ise; o üyeye sadece kendi sınıfı içerisinde ulaşılır. Protected erişim belirleyicisi model içerisindeki bir üyeye bulunduğu sınıftan veya o sınıftan türetilen diğer sınıflar içinden erişebileceğini gösterir. Internal ise bir üyeye aynı program içinden erişime izin verirken, başka programdan erişimi kısıtlaması için başvurulmuş bir erişim belirleyicisidir.

Çok Biçimlilik:

Çok biçimlilik herhangi bir verinin birden fazla biçimde kullanılmasıdır. Çok biçimlilik, nesne yönelimli modellemede en önemli kavramlarından biridir. İlgili kavram, herhangi bir mesajın birden çok sınıftaki nesnelere gönderilebildiği bir özelliktir ve her nesne, sınıf özelliklerine bağlı olarak uygun bir şekilde yanıt verme eğilimindedir. Yani bir fonksiyonun birçok türde ve birçok biçimde temsil edilebilmesidir. Çok

biçimliliğin gerçek hayattan bir örneğinde, herhangi bir kişinin aynı anda farklı rollere sahip olmasıdır. Bir birey, aynı zamanda bir ebeveyn, bir eş, bir çalışan ve bir evlattır. Yani aynı kişi birçok

özellığe sahip olabilir ama her birini duruma ve koşula göre uygular. Şekil 9 çok biçimlilik kavramını temsil etmektedir.



Şekil 9 Çok biçimlilik

Şekil 9'da ŞEKİL isimli temel sınıftan Kare, Daire ve Üçgen olmak üzere 3 sınıf türetilmiştir. Temel sınıftaki alanHesapla fonksiyonu her üç alt sınıfta da bulunmaktadır. Ancak ŞEKİL sınıfı bir Kare sınıfı nesnesinin yerine geçti ise alanHesapla fonksiyonu karenin alanını, Daire sınıfının

nesnesinin yerine geçti ise daire alanını hesaplayacaktır. Çok biçimlilik ile bir kez tasarlanan, test edilen ve uygulanan sınıfların yeniden yazılmasına gerek kalmaz ve dolayısıyla nesne yönelimli modellemede işlemlerinde zaman kazandırır.

2. UML DİYAGRAMLARI

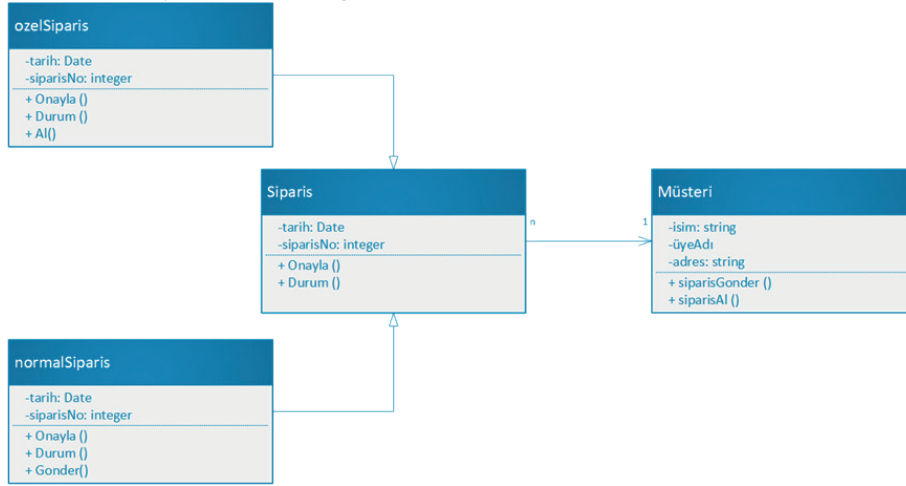
Birleşik modelleme dili (UML: Unified Modeling Language), yazılım sistemlerini, görselleştirmek, oluşturmak ve belgelemek için kullanılan standart bir dildir. Yazılım sistemlerinin geliştirilmesi genel olarak analiz, dizayn, kodlama ve uygulama aşamalarından oluşmaktadır. Özellikle analiz ve dizayn adımlarında UML diyagramları sıklıkla tercih edilmektedir. Geliştirilecek olan sistemin görselleştirilmesi kodlama ve uygulama aşamalarında hata riskini azaltmaktadır. UML Java, C#, Python gibi programlama dillerinden farklıdır. UML, yazılım sistemlerinin tasarımlarını oluşturmak adına kullanılan resimsel bir dildir. UML genellikle sistemleri modellemek için kullanılmaktadır. UML nesne yönelimli analiz ve tasarımın vazgeçilmez bir unsurudur. UML yapısının amacı tasarlanacak sistemi kolay anlaşılır ve basit hale getirmektir. İlaven tüm kullanıcılar ya da tasarımcılar için genel amaçlı modelleme oluşturmaktır.

UML diyagramları yapısal ve davranışsal olmak üzere genel olarak 2 kategoride toplanmaktadır. Yapısal model, sistemin çerçevesini temsil eder. Yapısal model bileşenleri verileri ve onları bir araya getiren mekanizmayı temsil etmektedir. İlgili model sistemin dinamik davranışından ziyade statik davranışlarını tanımlamaktadır. Sınıf, nesne, paket, dağılım, bileşen, birleşik gibi diyagramlar yapısal modeli oluşturan UML diyagramlarıdır. Sınıf diyagramı ise en yaygın kullanılan yapısal diyagramdır. Davranışsal model ise sistemdeki etkileşimi tanımlamaktadır. Davranışsal modelleme, sistemin dinamik doğasını gösterir. Etkinlik diyagramları, etkileşim diyagramları, kullanım-durum diyagramları dinamik modele örnektir. İlerleyen bölümlerde nesne tabanlı modelleme için en çok kullanılan UML diyagramları tanımlanacaktır.

2.1. Sınıf Diyagramları (Class Diagram)

Sınıf diyagramı statik bir diyagramdır. Bir sistemin statik görselini temsil eder. Sınıf diyagramı, aynı zamanda yazılım uygulamasının çalıştırılabilir kodunu oluşturmak için de kullanılmaktadır. İlgili diyagram ile bir sınıfın özniteliklerini, fonksiyonlarını ve ayrıca varsa eğer sisteme uygulanan kısıtlamalar ifade edilebilir. Sınıf diyagramları, doğrudan nesne yönelimli programla-

ma dilleriyle örtüşen tek UML diyagramları olduklarından, nesne yönelimli sistemlerin modellenmesinde yaygın olarak kullanılmaktadır. Sınıf diyagramı, sınıfların, ara yüzlerin, ilişkilerin, işbirliklerinin ve kısıtlamaların bir koleksiyonunu gösterir. Yapısal diyagram olarak da bilinir. Örnek bir sınıf diyagramı Şekil 10 ile gösterilmiştir.

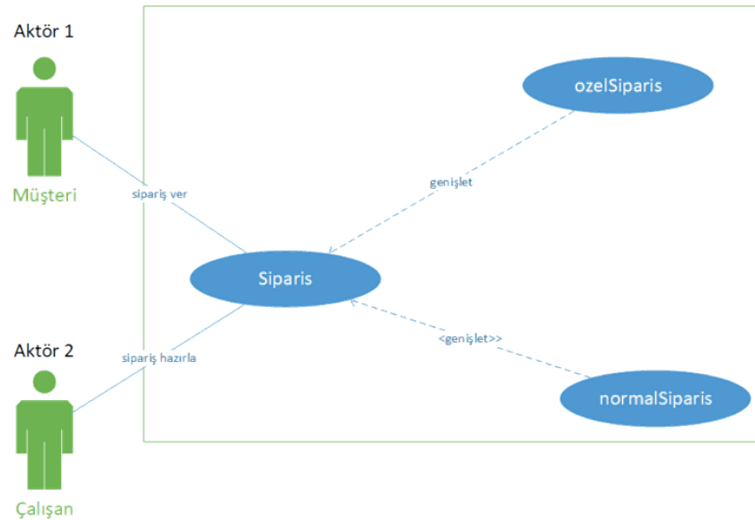


Şekil 10. Sınıf diyagramı

2.2. Kullanım Durum Diyagramları (Use Case Diagram)

Kullanım durum diyagramının temel amacı, bir sistemin dinamik yönünü simgelemektir. İlgili diyagramlar, iç ve dış etmenler de dâhil olmak üzere bir sistemin gereksinimlerini toplamak için kullanılır. Bu gereksinimler genellikle tasarım gereksinimleridir. Bu nedenle, kullanım

senaryoları hazırlanır ve aktörler belirlenir. Aktörler, sistemle etkileşime giren kullanıcıların rolleri, dâhili veya harici uygulamaları olarak tanımlanmaktadır. Kullanım durum diyagramını Şekil 11 özetlemektedir.

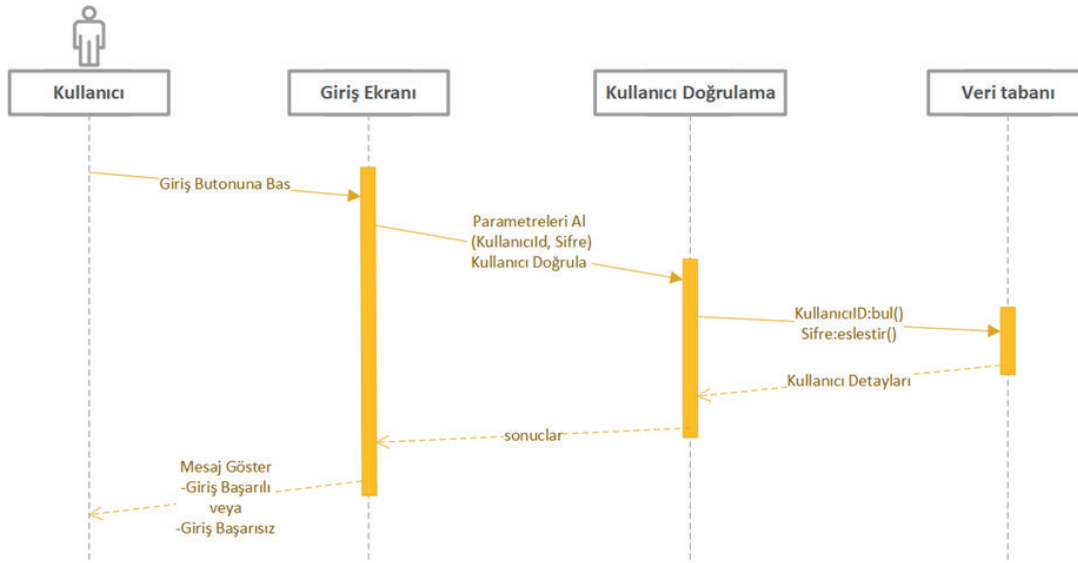


Şekil 11 Kullanım durum diyagramı

2.3. Dizge Diyagramları (Sequence Diagram)

Dizge diyagramları, nesneler arasındaki etkileşimi sıralı bir şekilde gösteren diyagramlardır. Dizge diyagramları, bir sistemdeki nesnelerin ve olayların nasıl ve hangi sırayla gerçekleştiğini açıklar. Bu bağlamda dinamik bir bakış açısı sağlamaktadır. Bu diyagramlar sistem ya da yazılım geliştiriciler tarafından yeni ve mevcut sistemlerin gereksinimlerini belgelemek ve ifade etmek adına yaygın olarak kullanılmaktadır. Zaman ve sıra dizge diyagramlarında önemli kavramlardır.

Aktörler diyagramların en tepesine yerleştirilir ve aşağı indikçe zaman geçmektedir. Yani bir sistemdeki gerçekleşecek işlemler yukarıdan aşağı sırasıyla yazılır. Şekil 12 bir kullanıcının herhangi bir sisteme giriş işleminden başlayan süreci gösteren bir dizge diyagramını göstermektedir. Söz konusu şekilde giriş butonuna basılmasıyla süreç başlamaktadır. Sonraki her bir adım sistemin içerisindeki işlemleri tanımlamaktadır ve belirli bir sıraya göre ilerleme gerçekleşmektedir.

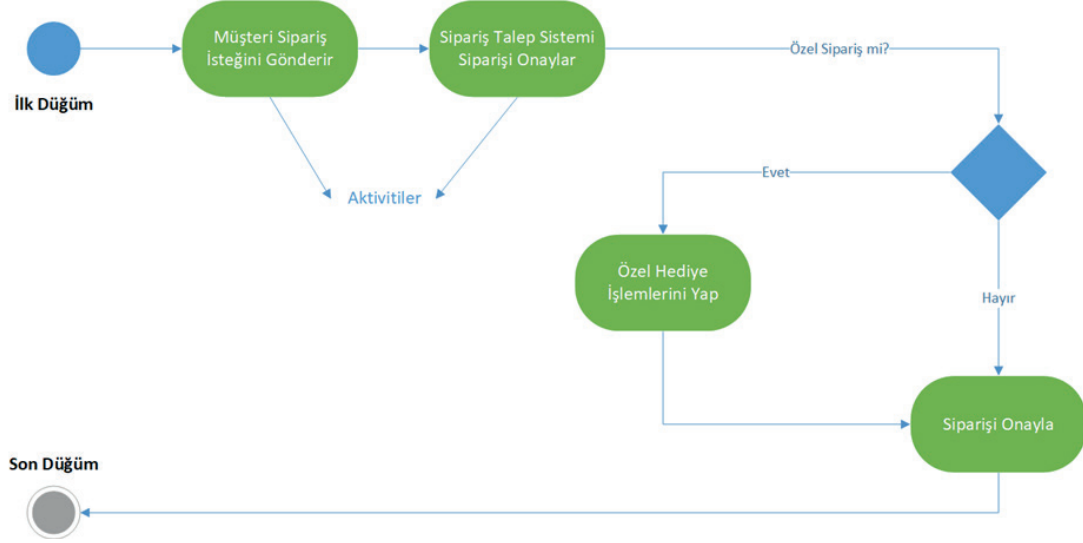


Şekil 12. Dizge diyagramı

2.4. Etkinlik Diyagramları (Activity Diagram):

Etkinlik diyagramlarının temel amacı tasarım yapılacak olan sistemin dinamik davranışlarını yakalamasıdır. Dizge diyagramları bir nesneden diğerine mesaj akışını göstermek için kullanılırken, etkinlik diyagramı bir etkinlikten diğerine mesaj akışını göstermek için kullanılır. Etkinlik, sistemin içerisinde belirli bir işlemdir. Etkinlik diyagramları sadece bir sistemin dinamik

doğasını görselleştirmek için kullanılmaz, aynı zamanda ileri ve tersine mühendislik teknikleri kullanılarak yürütülebilir sistemi oluşturmak için de kullanılır. Bir etkinlik diyagramı, bir sistemin etkinlik akışını görselleştirmek, etkinliklerin sırasını açıklamak, sistemin paralel, dallanmış ve eşzamanlı akışını tanımlamaktadır. Şekil 13 örnek bir etkinlik diyagramını ifade etmektedir.



Şekil 13. Etkinlik diyagramı

3. NESNE YÖNELİMLİ MODELLEME

Sistemlerin karmaşıklığının artması, proje maliyetlerinin ve iş gücünün önemi modelleme tekniklerinin önemini arttırmaktadır. Nesne yönelimli modelleme, gerçek dünyanın kavramları ile organize edilen bir çözümü görselleştirme tekniğidir. Söz konusu modelleme yaklaşımı, problemleri ifade etmeye, sistemlerin çalışma süreçlerini ifade etmeye ya da genel olarak tasarlamaya yardımcı olmaktadır. Bir sistem mekanizmasını geliştirmeden önce bir model geliştirmek çok önemlidir. Çünkü yapılacak hataları indirgeyerek zaman ve maliyet tasarrufu sağlar. Dolayısıyla nesne yönelimli modelleme popüler bir yaklaşımdır. Nesne yönelimli modeller diyagramlarla temsil edilir. Optimum düzeyde hazırlanmış bir model, ekipler arasındaki iletişim ve mimari olarak iyi bir yapılanmaya olanak sağlar.

Nesne Yönelimli modellemenin hedefi, tasarım sırasında sistemin detaylarını içeren bir uygulama modeli oluşturmaktır. Bu süreçte geliştiricinin, sistemin doğasında bulunan kavramları (Nesne, sınıf, özellik, fonksiyon vb.) tanımlaması gerekmektedir. İlk adımda, geliştirici sistemi net olarak organize eder. Devamında veri yapısı ve fonksiyonların detaylarını etkin bir şekilde tespitini yapar.

Nesne yönelimli modelleme aşağıda belirtilen 3 aşamadan meydana gelir:

- Sistem Analizi
- Sistem Dizaynı
- Nesne Dizaynı

İlk aşama olan sistem analizinde problem net bir şekilde belirlenir ve veriler toplanır. Analiz aşamasından sonra, istenen sistemin nasıl geliştirilmesi gerektiğine dair anlaşılır ve kesin bir durum ortaya çıkar. İkinci aşamada, sistem mimarisi tasarlanır. İlgili aşama, hem sistem analiz modeline hem de sistemin önerilen mimarisine dayalı olarak tüm sistemin alt sistemlere ayrıldığı aşamadır. Nesne tasarımı aşamasında, analiz modelinde verilen sistem detaylarının aracılığıyla sistemdeki bileşenlerin (sınıfların) yapılarına karar verilir.

Nesne yönelimli modellemenin yukarıda bahsi geçen aşamalarını tasarlamak için nesne, dinamik ve fonksiyon modelleme kullanılmaktadır. Nesne modelleme, sisteminin statik yapısını nesneler açısından tasarlar. Nesneleri, nesnelerin sınıflarını ve nesneler arasındaki ilişkileri tanımlar. Ayrıca, her bir sınıfı özelleştiren temel niteliklerini ve işlemlerini tanımlar. Nesne modelleme aşamaları aşağıdaki gibidir.

- ✓ Nesneleri tanımlanması ve sınıflara ayrılması
- ✓ Sınıflar arasındaki ilişkilerin belirlemesi
- ✓ Kullanıcı durum modeli diyagramı oluşturulması
- ✓ Kullanıcı nesne özniteliklerini tanımlanması

Sistemin statik tavrı analiz edildikten sonra dinamik davranışlarının incelenmesi gerekir. Dinamik modelleme, bir nesnenin, diğer nesneler tarafından tetiklenen sistem içi olaylar veya sistem dışından tetiklenen harici olaylar gibi durumlarda nasıl tepki göstereceğini açıklar. Dinamik modelleme süreci:

- ✓ Olayların tanımlanması
- ✓ Etkinlik diyagramları ile olayların ifade edilmesi
- ✓ Her durumun nesne özniteliklerinin tanımlanması

aşamalarından oluşmaktadır. Son olarak fonksiyonel modelleme, nesne yönelimli analizin son aşamasıdır. Fonksiyonel model, gerçekleşen işlemler ve yöntemler verilerin nasıl değiştiğini gösterir. Nesne modelleme işlemlerinin ve dinamik modelleme eylemlerinin sonuçlarını ifade eder. Fonksiyonel model, geleneksel analizin veri akış diyagramına benzemektedir.

BÖLÜM ÖZETİ

Dijital ortamlar ve uygulamalar hayatımızın vazgeçilmez bir parçası olmuşlardır. Teknolojideki hızlı gelişim ile birlikte kurum-kuruluşlar, özel şirketler, sivil toplum kuruluşları gibi birçok yapı için bahsi geçen ortamları ya da uygulamaları kullanma zorunluluğu oluşmuştur. Dolayısıyla sistem ve yazılım geliştirme süreçleri çok önemli bir hâle gelmiştir. Nesne tabanlı modelleme de yazılım geliştirme ve sistem tasarımı yaklaşımlarının temelini oluşturan bir kavramdır. Nesne yönelimli modelleme, nesne ve sınıflarının tanımlanmasıyla karakterize edilen bir görselleştirme sürecidir. Başka bir ifade ile nesne ve sınıfları temel alan bir yaklaşımdır. Bahsi geçen yaklaşımın daha hızlı yürütülmesi, daha kolay bir yapı sağlaması, tasarlanacak sistemin bakımının, güncellenmesinin ya da hata ayıklamasını kolaylaştırması, daha kısa sürede kullanılabilir uygulamalar geliştirmeye olanak sunması en önemli avantajlarından biridir.

Nesne tabanlı tasarım ve modelleme süreçlerini gerçekleştirmek için bazı temel kavramları öğrenmek gerekmektedir. Nesne yönelimli tasarımın temelini sınıflar oluşturmaktadır. Kendi veri üyeleri ve üye fonksiyonları olan bir veri türüdür. Sınıf, bir nesnenin planı gibidir. Bir başka ifade ile sınıf, veri üyelerine ve üye işlevlerine sahip olan veri türüdür. Nesneye yönelik tasarımlarda, nesneler sınıflardan türetilirler. Nesne, bir sınıfın örneğidir. Sınıfın tüm veri üyelerine ve üye işlevlerine nesneler yardımıyla erişilebilir. Sınıflardan oluşturulan nesneleri tanımlamaya yarayan özelliklere öznitelik denir. Nesne yönelimli modellemede kalıtım, bir nesneyi veya sınıfı, başka bir nesnenin veya sınıfın özelliklerini kullanarak alt sınıf ya da alt nesneler oluşturma mekanizması olarak tanımlanmaktadır. Nesne tabanlı modellemenin temel kavramlarından biri olan kapsülleme, kullanıcıların sınıf, metot verilerin ne kadarına erişebileceğini belirler. Başka bir ifade ile sınıf, metot ya da verilere doğrudan erişimlerin kısıtlanması ve izin verilmesi gibi işlemleri gerçekleştirmek için kullanılır. Çok biçimlilik herhangi bir verinin birden fazla biçimde kullanılmasıdır. Çok biçimlilik, nesne yönelimli modellemede en önemli kavramlardan biridir. İlgili kavram, herhangi bir mesajın birden çok sınıftaki nesnelere gönderilebildiği bir özelliktir ve her nesne, sınıf özelliklerine bağlı olarak uygun bir şekilde yanıt verme eğilimindedir.

Nesne yönelimli modelleme, gerçek dünyanın kavramları ile organize edilen bir çözümü görselleştirme tekniğidir. Söz konusu modelleme yaklaşımı, problemleri ifade etmeye, sistemlerin çalışma süreçlerini ifade etmeye ya da genel olarak tasarlamaya yardımcı olmaktadır. Nesne Yönelimli modellemenin hedefi, tasarım sırasında sistemin detaylarını içeren bir uygulama modeli oluşturmaktır.

Nesne yönelimli modelleme 3 aşamadan meydana gelir. Bunlar; Sistem Analizi, Sistem Dizaynı ve Nesne Dizaynıdır. İlk aşama olan sistem analizinde problem net bir şekilde belirlenir ve veriler toplanır. Analiz aşamasından sonra, istenen sistemin nasıl geliştirilmesi gerektiğine dair anlaşılır ve kesin bir durum ortaya çıkar. İkinci aşamada, sistem mimarisi tasarlanır. İlgili aşama, hem sistem analiz modeline hem de sistemin önerilen mimarisine dayalı olarak tüm sistemin alt sistemlere ayrıldığı aşamadır. Nesne tasarımı aşamasında, analiz modelinde verilen sistem detaylarının aracılığıyla sistemdeki bileşenlerin (sınıfların) yapılarına karar verilir.

GÖZDEN GEÇİRELİM

1. Kitap ödünç yazılımı için bir sistem tasarlayınız ve bu sistemin sınıflarını, nesnelerini, aktörlerini belirleyiniz.
2. Sınıf diyagramları dinamik bir süreci modellemek için kullanılır (D/Y)
3. diyagramı bir nesneden diğerine mesaj akışını göstermek için kullanılırken, diyagramı bir etkinlikten diğerine mesaj akışını göstermek için kullanılır.
4. Kullanım Durum diyagramı özellikleri hakkında bilgi veriniz.
5. Erişim kısıtlaması olmayan ve her yerden erişilebilen bir nesne modelleme bileşeninin erişim belirleyicisi nedir ?

Kaynakça

- Roth, R.M., Dennis, A., and Wixom, B.H., (2013). Systems Analysis and Design, 5th Edition, John Wiley&Sons Inc.
- Tilley, S. (2019).Systems analysis and design. Cengage Learning
- <https://www.geeksforgeeks.org/software-engineering-object-modeling-technique-omt/>
- Kenneth E. Kendall Julie E. Kendall, (2010). Systems Analysis and Design,Pearson Education
- Robert Lafore, (2009). Nesne Yönelimli Programlama Kılavuzu, 4. Baskı, ALFA yayıncılık

BÖLÜM 8

SİSTEM TASARIMI

ANAHTAR KAVRAMLAR

Sistem Tasarımı, Model Güdümlü Yaklaşım (MGY), Mimari, Katmanlar, Standartlar



ÖĞRENME HEDEFLERİ

-  Model Güdümlü Yaklaşımı (MGY) anlama
-  MGY modellerinin kullanılmasını, otomatikleştirilmesini ve analizini yapma
-  MGY mimarisini ve katmanlarını tanıma
-  Modelleme dillerini öğrenme
-  MGY modellerini özel gereksinimlere uygulama

GİRİŞ

Sistem tasarımı için günümüzde pek çok metot bulunmaktadır. Model Güdümlü

Yaklaşım (MGY), Hızlı Uygulama Geliştirme (HUG) ve Ortak Uygulama Geliştirme (OUG) en çok bilinen ve tercih edilen tekniklerdir. Bizler kitabımızın bu bölümünde sistem tasarımı, MGY üzerinden öğreneceğiz. Ayrıca, iletişim aracı olarak modellerin kullanılması, modellerin otomatikleştirilmesi, analizi, model olarak benzetim araçlarının kullanılması ve bilginin elde edilmesi süreci öğreneceğimiz diğer konular arasındadır.

Sistem tasarımı yapılırken MGY modellerinin yapısı ve ne modellediğimiz çok önemlidir. Bu bağlamda bilinmesi gereken temel kavramlar, sistem türleri, modelleme dilleri ve mimari katmanlar ayrıntılı bir şekilde işlenmektedir. Ünite tamamlandığında sizlere MGY modelleri hakkında bir bakış açısı kazandırılması hedeflenmektedir. Söz konusu bakış açısı, modellenecek sistemin mimarisini anlamada ve temel endişelerin giderilmesinde fikir sahibi olmamızı sağlar. MGY modelleri platforma özel veya bağımsız olarak üretilebilmektedir. Bu kapsamda örnek bir dönüşüm modeli incelenmiştir. Son olarak, MGY standartları, gerçekleştirilen yaşam döngüsü, temel model yönetimi ve MGY'nin özel gereksinimlere uygulanması işlenen konular arasındadır.



1. SİSTEM TASARIMI

Model GÜDÜMLÜ Yaklaşım

Model GÜDÜMLÜ Yaklaşım (MGY), yazılım sağlayıcıların tasarım, geliştirme ve uygulamada, yaklaşımın adında da geçtiği üzere, model kullanarak bir dizi şartnamenin hazırlanmasını sağlayan bir kılavuздur. Söz konusu metodun mimarisi, etki alanı (domain) mühendisliği olarak da bilinmektedir. MGY, ilk kez 2001 yılında bilgisayar endüstrisi standartlarını belirleyen ve bir şirketler birliği olan Object Management Group (OMG) tarafından önerilmiştir.

Bilindiği üzere Bilişim Teknolojileri (BT) günümüzde geçerliliğini koruyan ve hali hazırda devam eden önemli bir yapıdır. İlgili sistemde diğer sistemlerde olduğu gibi bir yaşam döngüsü olup, tüm işler bir çevrim içerisinde gerçekleştirilmektedir. MGY bu döngüye ürünlerin kalitesini artırmak ve sürdürülebilirliğini sağlamak için en küçük uygulama detayına kadar destek sağlayıp, model adı verilen bir yapı ile değer üretmeye çalışmaktadır. Bu bağlamda, MGY'nin asli işi endüstri standartları aracılığıyla modellerin yapısını, dönüştürülmesini, paydaş belgelerin üretilmesini ve yürütülmesi noktasındaki aşamalarını belirlemektedir. Ayrıca MGY modelleri, tedarik ve sistem şartnameleri hazırlamada, yazılım tasarım modellerini kaynak kodlara dönüştürmede ve yürütülebilir diğer sistemler için de tercih edilmektedir.

Yukarıda bahsi geçen MGY modeli aslında bir sistemin soyutlaması yani temsildir. OMG üç tür soyutlama düzeyinden bahseder:

- Hesaplamadan Bağımsız Model – HBM (Computation-independent Model),
- Platformdan Bağımsız Model – PBM (Platform-independent model),
- Platforma Özgü Model – PÖM (Platform-specific model).

HBM, sistemi kavramsal olarak tanımlarken, PBM onu gerçekleştirmek adına gerekli olan teknolojilere atıfta bulunmadan sistemin hesaplamalı yönlerini belirler. PÖM ise ilgili sistemin gerekli teknik boyutları ile ilgilenmektedir. MGY modelleri sistemin mimarisini ve teknolojik gereksinimlerini, HBM, PBM ve PÖM soyutlama düzeylerindeki durumları ve farklı bakış açıları ile temsillerindeki bağlantı noktalarını bularak birleştirmektedir. Böylece karmaşık sistemler ile başa çıkma noktasında bir değer sağlamaktadır. Elde edilen değerler konusu hakkında daha fazla bilgi bölümün izleyen başlıklarında incelenmektedir.

İletişim Aracı Amaçlı Modeller

MGY ile modellerin kurulmasında fikir birliğine varmak esastır. Bu anlamda MGY için kullanılan bir takım standartlar söz konusudur. Bu standartlar:

- Belirlenen bir konu üzerinde uzlaşa sağlayarak terim, simge ve notasyonları oluşturmak
 - İşlemlerin sürümlendirilmesi (bir konuyla ilgili değişik metinlerden her biri, versiyon) ve paylaşılması için kullanılacak temelleri inşa etmek
 - Ortak sözlük ve kurallar, tekrarlanan işlemler, iş nesne modelleri ve mimari tasarım örnekleri için kütüphaneyi hazırlamak
- şeklinde dir. Böylece organizasyonun iletişim kurmada gerekli bilgilerin, süreçlerin ve hizmetlerin tasarımı daha kolay hale gelmektedir.



EK BİLGİ

Notasyon kelimesi TDK güncel Türkçe sözlükte notalama olarak geçmektedir. Türk işaret dilinin parmak alfabesi ile gösterilişi, matematikte tam sayıların Z olarak ifade edilmesi veya satranç tahtasındaki karelerin a-h ve 1-8 arası bölünmesi ile oluşan koordinat sistemi buna örnek olarak verilebilir.

Otomatikleştirilmiş Modeller

MGY modelleri kısmen veya tamamen otomatik hale getirilebilmektedir. Amaç bir otomasyonun gerçekleştirme, değişim ve bakım maliyetlerinin azaltılmasıdır. Böylece her tekrarda daha tutarlı sonuçlar elde edilmektedir. OWL ve ortak mantık gibi ontoloji dilleri, ilgili süreçte kullanılmaktadır. Bu bağlamda modellerden kod üretmek şeklinde bir değer üretilmektedir.



ARAŞTIRILIM

Ontoloji dilleri MGY modellerinden sürekli bir değer önermek için kullanılmaktadır. W3C Web Ontoloji Dili (OWL), nesne grupları ve nesneler arasındaki ilişkiler hakkında zengin ve karmaşık bilgileri temsil etmek için tasarlanmış bir Semantik Web dilidir.

dır. Modelin doğrulanması, istatistiki işlemler ve çeşitli diğer ölçümler modelin sunmuş olduğu semantik (anlamsal) veriler üzerinden yapılır.

Model Simülasyonu (Benzetim) ve Yürütülmesi

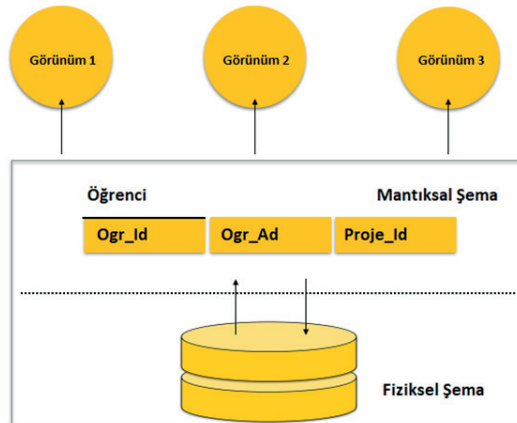
Simülasyon bir sistemin nasıl çalıştırıldığı noktasında tasarlanan örnek bir modeldir. Böylece analiz ve yürütme basamağında bilgi sahibi olunmaktadır. Veritabanı Yönetim Şeması (VTYŞ) ve süreç modelleri de yürütülebilir modeller için bir örnektir.

VTYŞ, verilerin nasıl düzenlendiğini ve aralarındaki ilişkilerin nasıl olacağını belirleyen bir iskelet yapıdır ve şema diyagramları aracılığıyla ayrıntılı bir gösterim içermektedir. Fiziksel ve mantıksal olarak iki tür şema oluşturulmaktadır. Fiziksel şema verilerin nasıl depolanacağı ile ilgilenirken, mantıksal şemalar mantıksal kısıtlamaları belirlemede kullanılmaktadır. Şekil 1'de örnek bir VTYŞ gösterilmektedir.

Model Analitiği

MGY'de modellerden karar verme, izleme ve kalite değerlendirme amaçlı da yararlanılmakta-

Şekil 1. Örnek bir veritabanı yönetim şeması



Modellerden Bilgi Elde Etme

İyi tanımlanmış MGY modelleri ile diğer bilgilerin türetilmesi mümkündür. Bu tarz modeller kaynak model olarak ifade edilmektedir. Söz konusu modeller;

- Oyun kitapları (Amerikan futbolunda ve basketbolda takımların strateji ve taktiklerinin açıklandığı saha üzerinde hareketleri gösteren şemaların olduğu deftere verilen isimdir)

- Edinilen sezgiler
 - Belgeler
 - Şartnameler (Örneğin Teklif Talepleri)
 - Bilişim Sistemleri
- olarak ifade edilebilir.



ARAŞTIRALIM

Teklif talebi, bir ürün veya hizmet satın almak isteyen bir kuruluştan potansiyel satıcılara verilen resmi, anket tarzı bir belgedir. Teklif talebi, potansiyel satıcılara arka plan bilgileri sağlar, önemli sorular sorar ve onlara ihtiyacı karşılamak için bir teklif sunmaya davet eder.

İş dünyasında ise alıcılar tarafından satıcılardan önemli bilgileri organize bir şekilde toplamak için kullanılır. Örneğin, kuruluşunuzun yeni yazılım satın alması gerekiyorsa, büyük olasılıkla birkaç seçeneği araştırmak ve bunları karşılaştırmak istemektedir.

Bu şekilde erişebileceğimiz Teklif Talebi belgelerini inceleyin.

Yapılandırılmamış Bilginin Yapılandırılması

Belgelerde, web sayfalarında ve şemalarda çok sayıda yapılandırılmamış bilgi olup, MGY modelleri ile bunların yapılandırılmış temsili sağlanabilir. Yapılandırılmamış bilgilerden semantik veri elde eden yazılımlar ise bu tarz MGY modellerine örnek olarak verilebilir.

Modellerin Yapısı ve Anlamı

MGY modellerinin temelinde; sorgulama, analiz etme, raporlama, doğrulama, benzetim ortamına aktarma ve yürütülebilir kodlarda olduğu gibi diğer yararlı biçimlere dönüştürme işlemi yapılmaktadır. Bu süreçte verilerin temsiline, anlamına ve farklı gösterimlerde ifade edilmesine uygun MGY standartları vardır.

MGY modellerinin yapısını ve anlamını kavramak için bir şirket toplantısından örnek verelim. İlgili toplantıda tartışma konusu bir beyaz tahta üzerinden yapılsın. Anlatıcının durumu iyi bahsetmesi adına kullandığı simgeler önemlidir

ve çizdiği diyagramların herkes tarafından anlaşılabilir olması gerekmektedir. Ayrıca yazılan bilgilerin kaybolmaması için de tahtanın bir fotoğraf makinesi aracılığıyla kaydedilmesi lazımdır. Bu yöntem gayri resmi bir modeldir. İşte MGY modelleri bu beyaz tahtada sunulan diyagramların arkasında yatan verileri anlamsal hale getirmektedir. Buna verinin semantiği (anlamı) denir. Yapısı ise verinin tanımlanmasıdır. MGY modelleri ile elde edilen ve istenilen şekilde tanımlanmış semantik veriler ile

- Yönetme ve geliştirme,
- Sorgulama, doğrulama ve analiz etme,
- Paylaşma,
- Diğer ortamlarda yeniden kullanma,
- Bağlantılar kurma,
- Başka diyagramlar ve görseller elde etme ve
- Yazılım araçları ile başka verilere dönüştürme işlemleri yapılabilmektedir.

Ne Modelliyoruz?

MGY modelleri farklı amaçlar için kullanılabilir ve farklı alanlarda bir bağlam ve kapsam oluşturan alanlara sahiptir. Bahsi geçen alanlar;

- Telekomünikasyon
- Sağlık Hizmeti
- Perakende Yapılan Ticari Alanlar
- İmalat Sanayi
- Toplu Taşıma
- Savunma Sistemleri

- Her Türden Devlet (Kamu) İşleri

olabilir. Yukarıda listelenmeyen daha özel alanlarda model konusu olabilir. Bu modeller bir tasarı olup, gerçeğe dönüştürülmek istenmektedir. Modeller oluşturulurken farklı bakış açılarına sahip olunabilir. Bu durum farklı paydaşlar için bir anlam ifade edebilmektedir. Dolayısıyla MGY modelleri oluşturulurken farklı düşünce tarzlarına göre birden fazla model oluşturulabilmektedir.

2. MGY TEMEL KAVRAMLAR

Bu bölümde, MGY'ye ait temel kavramlar olarak sistem, model, modelleme dili, mimari, bakış açıları, soyutlama, katmanlar, dönüştürme işlemleri, endişeler ve platformlardan bahsedilecektir.

Sistem

Sistem kelime anlamı olarak bir hedefe ulaşmak için kullanılan organize edilebilen parçalar ve onlar arasındaki ilişkilerin bütünüdür. MGY için sistem kavramı bilgi işlem sistemlerine atıfta bulunmaktadır. Ancak daha genel anlamlar için de kabul edilmektedir. Bu bağlamda sistem;

- Donanım
- Yazılım
- İnsanların oluşturduğu bir topluluk/federasyon
- İşletme
- Program
- Tek bir bilgisayar
- Bir makineye ya da bir bilgisayar ait gömülü bir yapı

şeklinde olabilmektedir. Burada amaç yukarıda maddeler halinde listelenen yapı için MGY modelleri ile bir temsil ve anlam oluşturmaktır.

Model

MGY için model, belirlenen kaygılar ışığı altında sistemin bazı yönlerini temsil etmektir. Burada modelin açık ya da örtülü bir şekilde sisteme entegre olması önemlidir. Model, sistemi temsil edecek bilgi kümesini, sistemin geçerliliğini bozmayacak şekilde tanımlı kurallarını ve sisteme ait bütün terimlerin anlamını bozmadan vermelidir.

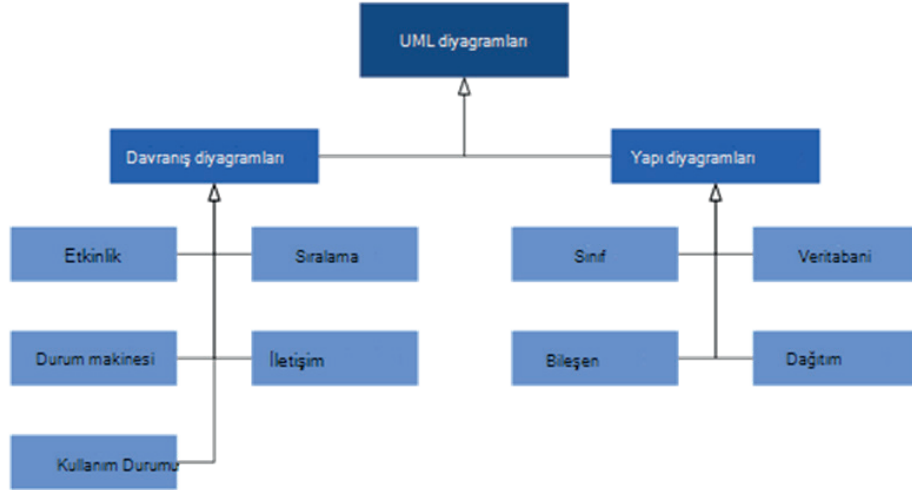
Örneğin yazılım sistemlerine ait bir model;

- Birleşik Modelleme Dili (UML) diyagramları (Şekil 2)

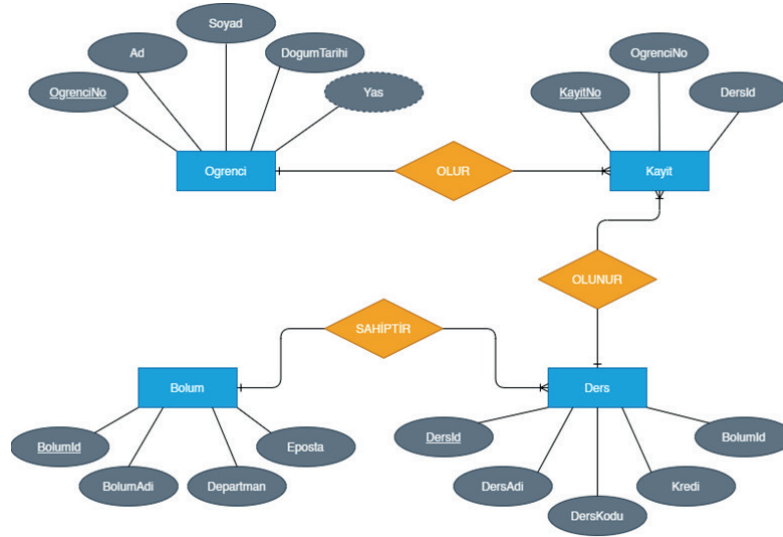
- Varlık-İlişki diyagramları (Şekil 3) ve

- Kullanıcı Arayüzü görüntüleri

içerirken, fiziksel bir sistem donanım, fiziksel ortamların temsili ve simülasyon ortamlarını içerebilmektedir.



Şekil 2. UML diyagramı



Şekil 3. Varlık-İlişki diyagramı

Bir kuruluşa ait model, insanların okuyabildiği veya elektronik ortama aktarılmış iş süreçlerini, hizmetleri, bilgileri ve kaynakları içerebilir. Modellere zamansal anlamlar yüklenerek bugünkü halini, onun bir türevini veya gelecekte olması gereken hali temsil edilebilmektedir.

Modelleme Dili

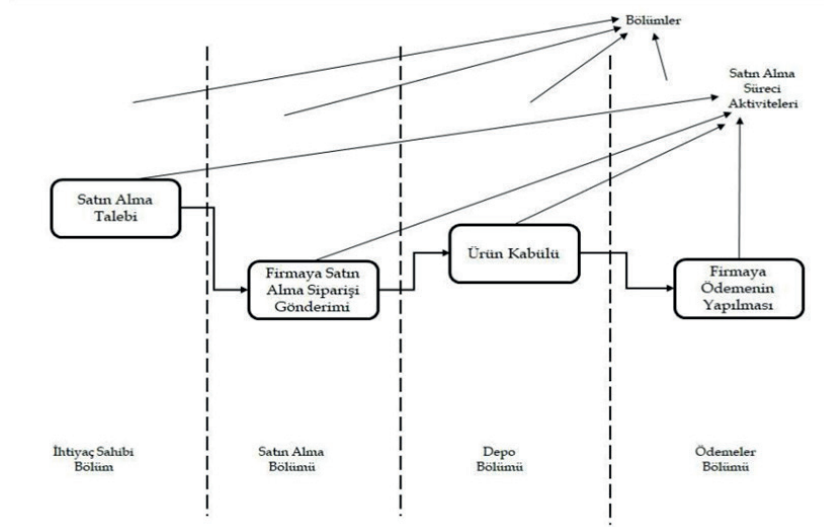
Modelleme dilinden bahsedilirken sisteme ait tüm paydaşların anlayabileceği, teknolojik olarak desteklenebilecek ve sistem hakkındaki tüm bilgileri kapsayan unsurlar akla gelmektedir. Bu bağlamda sistemin yapısının, terimlerin, notasyonların, söz dizimlerin ve bütünlüğü sağlayıcı

tüm kuralların iyi bir şekilde temsil edilmesi gerekmektedir.

İyi bilinen modelleme dilleri;

- UML,
- SQL şemaları,
- İş Süreci Modeli ve Gösterimi (BPMN) (Şekil 4),
- Varlık-İlişki,
- OWL ve
- XML şemaları

akla gelmektedir.



Şekil 4. İş süreci modeli ve gösterimi (BPMN) [2]

Modeller ve modelleme dili arasında bir bağlantı olmak zorundadır. Ancak gayri resmi modellerde (beyaz tahtada yapılan toplantı örneği) bir uygunluk aramaya gerek yoktur.

Mimari

MGY için mimari, sistemleri tanımlamak ve iyileştirmek için kullanılan bir kavramdır. Mimari yapı bahsedilirken,

- İlgili sistemi temsil eden bir dizi model ve
- Model kümelerinin oluşturulduğu etkinlik/uygulama

akla gelmelidir. Bu bağlamda mimari yapı sistemin gerçekleştirilmesi, iyileştirilmesi ve daha az maliyet ve risk taşıması anlamına gelmektedir.

Görünüm ve Bakış Açıları

Sistemde paydaşların endişelerinin ele alındığı bilgilerin oluşturulması, seçilmesi ve sunulması adına gerekli bir dizi kıstas, modelin görünüm ve bakış açısını belirlemektedir (Örneğin bir bordro sistemindeki güvenlik endişeleri). Ayrıca bilginin paylaşıldığı bir sisteme ait bir MGY modeli oluşturulacaksa sisteme ait görünüm,

- Paylaşılacak bilginin anlamını,
- Bilginin nasıl yapılandırıldığını,
- Bilgilerin kiminle ve nasıl paylaşılacağını,
- Tüketicinin istediği bilgiyi,
- Dürüstlük ve iş kaygılarını,
- Bilgi iletiminde hangi teknolojilerin kullanılacağını ve
- Verilerin nasıl elde edildiğini

ifade etmelidir. Yukarıda bahsi geçen görünüm-lerin arasında görünen veya görünmeyen bir bağlantı olacağı unutulmamalıdır. Tüm bu unsurlar, bizim bakış açılarımızı geliştirmek adına önemlidir.

Soyutlama

Soyutlama bir sistemi daha genel bir şekilde kavramadır. Daha genel bir ifadeyle belirli unsurlar tanımlanan kapsamdan çıkarılmaktadır. Bu ayrıntıların çıkarılması daha üst düzey bir bakış açısı bize kazandırmaktadır. Dolayısıyla çıkarılan unsurlar ile sistem daha soyut hale getirilmektedir. Eğer sistem kısıtlı hale getirilirse daha az soyut olacaktır.

Mimari Katmanlar

Bir mimariyi soyutlama düzeyine göre tanımlamak faydalıdır. Mimari katman herhangi bir sayıda olabilmektedir. Genel sınıflandırma,

- İş veya etki alanı modelleri
- Mantıksal sistem modelleri
- Uygulama modelleri

Dönüşüm

Dönüşüm bir modelden farklı modeller, bakış açıları veya eserler üretmekle ilgili bir durumdur. Bir gösterimi diğerinden elde etmek için dönüşüm kullanıldığında kavramsal içerik aynı kalsa da bilgi başka bir biçimde sunulmaktadır. Dönüşümler çoklu girdi modellerini bir modelde birleştirmektedir. Örneğin teknolojiden bağımsız bir model J2EE ile birleştirilerek teknolojiye özgü hale getirilebilmektedir. Örneğin somut bir XML şeması ile hazırlanan bir modelden soyut bir UML şemasına geçmek mümkündür.

Endişelerin Ayrılması

Sistemin daha çevik ve esnekliğinin sağlanması adına oluşturulacak MGY modellerinde endişelerin kurulacak modelden ayrılması gerekmektedir. Bu bize “böl ve fethet” yaklaşımını sunmaktadır. Böylece daha dinamik bir ortam söz konusu olmaktadır.

Platform

Platform sistemi desteklemek ve uygulamak adına bir işlemin gerçekleştirildiği kaynaklar kümesi olarak ifade edilmektedir.

Platform türü olarak,

- İş veya Etki Alanı Platform türleri

o Organizasyon Yapısı

o Fiziksel tesisler

o İnsan Kaynakları

o Anlaşmalar

- Bilgisayar Donanımı ve Yazılım Platform Türleri

o Toplu İş

o Veri Akışı

o Gömülü Sistemler

o AJAX

o CORBA

o Eclipse RCP

o OSGi

o J2EE Uygulama Sunucusu

o Microsoft .NET

o RTOS

o Athlon 64 X2, ARM11, PowerPC G5, 8051

Yukarıda bahsedilen platformlar daha sonra diğer platformları kullanan MGY modelleri için kullanılabilmektedir.



EK BİLGİ

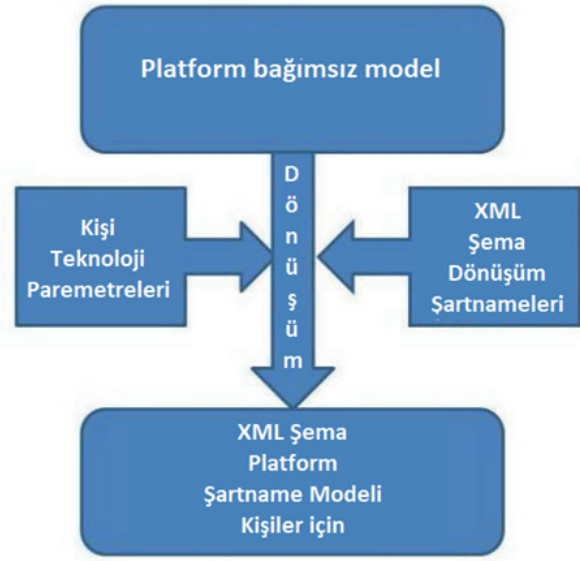
Java EE (Java Enterprise Edition, eski adıyla J2EE), özellikle büyük çaplı projelerin ihtiyaçları için geliştirilmiş Java teknolojilerinin bütünün oluşturduğu çerçevenin ve standardın ismidir.

3. ÖRNEK MODEL

Bir kişi ile ilgili model platformdan bağımsızdır. Yani herhangi bir sayıda platform teknolojisi ile gerçekleştirilebilmektedir. Model dönüşümü platform olarak XML şeması kullanarak yapılandırılabilir. Sağdaki diyagram bu dönüşümü modellemektedir. Burada:

- Platform bağımsız model, kişi bilgilerini temsil eden mantıksal bilgi modelini
- XML şema dönüşüm şartnamesi, dönüşümü yürüten yeniden kullanılabilir bileşeni
- Kişi teknoloji parametreleri, istenen teknoloji olarak XML şemasını
- Dönüşüm, dönüşüm işlemini
- Kişiler için XML şema platform şartname modeli, kişi için temsili

ifade etmektedir. Söz konusu modelde XML şeması yerine Java ve C# gibi birden çok teknoloji ile gerçekleştirilebilir.



BÖLÜM ÖZETİ

Sistem tasarımı için günümüzde pek çok metot bulunmaktadır. Model GÜdümlü Yaklaşım, Hızlı Uygulama Geliştirme ve Ortak Uygulama Geliştirme en çok bilinen ve tercih edilen tekniklerdir. Model GÜdümlü Yaklaşım, yazılım sağlayıcıların tasarım, geliştirme ve uygulamada, yaklaşımın adında da geçtiği üzere, model kullanarak bir dizi şartnamenin hazırlanmasını sağlayan bir kılavuzdur. Söz konusu metodun mimarisi, etki alanı mühendisliği olarak da bilinmektedir.

Model GÜdümlü Yaklaşım'a ait temel kavramlar olarak sistem, model, modelleme dili, mimari, bakış açıları, soyutlama, katmanlar, dönüştürme işlemleri, endişeler ve platformlardır. Sistem kelime anlamı olarak bir hedefe ulaşmak için kullanılan organize edilebilen parçalar ve onlar arasındaki ilişkilerin bütünüdür. Model GÜdümlü Yaklaşım için model, belirlenen kaygılar ışığı altında sistemin bazı yönlerini temsil etmektir. Burada modelin açık ya da örtülü bir şekilde sisteme entegre olması önemlidir. Modelleme dilinden bahsedilirken sisteme ait tüm paydaşların anlayabileceği, teknolojik olarak desteklenebilecek ve sistem hakkındaki tüm bilgileri kapsayan unsurlar akla gelmektedir. Bu bağlamda sistemin yapısının, terimlerin, notasyonların, söz dizimlerinin ve bütünlüğü sağlayıcı tüm kuralların iyi bir şekilde temsil edilmesi gerekmektedir.

Model GÜdümlü Yaklaşım için mimari, sistemleri tanımlamak ve iyileştirmek için kullanılan bir kavramdır. Mimari yapı bahsedilirken, ilgili sistemi temsil eden bir dizi model ve Model kümelerinin oluşturulduğu etkinlik veya uygulama akla gelmelidir. Bu bağlamda mimari yapı sistemin gerçekleştirilmesi, iyileştirilmesi ve daha az maliyet ve risk taşıması anlamına gelmektedir. Sistemde paydaşların endişelerinin ele alındığı bilgilerin oluşturulması, seçilmesi ve sunulması adına gerekli bir dizi kıstas, modelin görünüm ve bakış açısını belirlemektedir. Soyutlama, bir sistemi daha genel bir şekilde kavramadır. Daha genel bir ifadeyle belirli unsurlar tanımlanan kapsamdan çıkarılmaktadır. Bu ayrıntıların çıkarılması daha üst düzey bir bakış açısı bize kazandırmaktadır. Dolayısıyla çıkarılan unsurlar ile sistem daha soyut hâle getirilmektedir. Eğer sistem kısıtlı hâle getirilirse daha az soyut olacaktır. Bir mimariyi soyutlama düzeyine göre tanımlamak faydalıdır. Mimari katman herhangi bir sayıda olabilmektedir. Genel sınıflandırma; iş veya etki alanı modelleri, mantıksal sistem modelleri ve uygulama modelleri şeklindedir. Dönüşüm bir modelden farklı modeller, bakış açıları veya eserler üretmekle ilgili bir durumdur. Bir gösterimi diğerinden elde etmek için dönüşüm kullanıldığında kavramsal içerik aynı kalsa da bilgi başka bir biçimde sunulmaktadır. Dönüşümler çoklu girdi modellerini bir modelde birleştirmektedir. Sistemin daha çevik ve esnekliğinin sağlanması adına oluşturulacak Model GÜdümlü Yaklaşım modelinde endişelerin kurulacak modelden ayrılması gerekmektedir. Bu bize “böl ve fethet” yaklaşımını sunmaktadır. Böylece daha dinamik bir ortam söz konusu olmaktadır. Platform sistemi desteklemek ve uygulamak adına bir işlemin gerçekleştirildiği kaynaklar kümesi olarak ifade edilmektedir.

GÖZDEN GEÇİRELİM

1. Sistem tasarımında kullanılan yaklaşımlar nelerdir?

4. Model güdümlü yaklaşımlarda temel kavramlar nelerdir?

2. Model Güdümlü Yaklaşımın soyutlama düzeyleri nelerdir?

5. Bilgisayar yazılımı için kullanılan platformlara örnek veriniz?

3. Kaynak modeller ne için kullanılmaktadır?

Yanıt Anahtarı:

1. Sistem tasarımında kullanılan pek çok yaklaşım bulunmaktadır. Ancak en çok kullanılan yöntemler Model Güdümlü Yaklaşım, Hızlı Uygulama Geliştirme ve Ortak Uygulama Geliştirme yaklaşımları en bilindik metotlardır.
2. Soyutlama bir modelin temsil edilmesidir. Hesaplamadan Bağımsız Model, Platformdan Bağımsız Model ve Platforma Özgü Model MGY'nin kullandığı soyutlama türleridir.
3. Kaynak modeller ile diğer bilgilerin türetilmesi mümkün olup, oyun kitapları, edinilen sezgiler, belgeler, şartnameler ve bilişim sistemleri kaynak model olarak kullanılabilir.
4. Sistem, Model, Modelleme Dilleri, Mimari, Bakış Açıları, Endişeler, Dönüşüm, Soyutlama ve Platform MGY'nin anlaşılması adına önemli temel kavramlardır.
5. Toplu İş, Veri Akışı, Gömülü Sistemler, AJAX, CORBA, Eclipse RCP, OSGi, J2EE, Microsoft .NET, RTOS, Athlon 64 X2, ARM11, PowerPC G5 ve 8051 bilgisayar yazılımlarının gerçekleştirilmesinde en yaygın kullanılan platformlardır. Bu platformların özelliği birinden bir diğerine geçebilme esnekliğidir.

Kaynakça

- OMG MDA Guide revision 2.0, 18 Haziran 2014, Boston, <https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf>
- Yıldız, D. (2021). İş Süreçlerinin Modellenmesi, İyileştirilmesi, Performansının Ölçülmesi ve Yönetilmesinin İşletme Verimliliğine Katkısı: Bir Uygulama. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, (59), 91-118.
- Karata, E., & İyidir, B. (2009). Yazılım Ürün Hattı Yaklaşımında Model Güdümlü Uygulama Mühendisliği.
- Büşra, İ. Ç. Ö. Z., & KALIPSIZ, O. (2020). Model Güdümlü Yazılım Geliştirme Yaklaşımı Kullanılarak Mikro Servis Geliştirilmesi. Mühendislik Bilimleri ve Tasarım Dergisi, 8(5), 142-148.
- Uysal, M. P., & Mergen, A. E. (2015). Yazılım Yeniden Yapılamaya Yönelik Model Güdümlü ve Kaliteye Yönelimli Süreç Modeli. In UYMS.
- Yigit, I. O., Turhan, N. K., Yilmaz, A. E., & Durak, B. (2015). Yazılım Mimari Tasarımından Yazılım Geliştirme Çatısının Üretilmesinde Model Güdümlü Bir Yaklaşım. In UYMS.

Faydalı Kaynaklar ve İnternet Kaynakları

- Owl. OWL - Semantic Web Standards. (n.d.). 13 Aralık 2022, <https://www.w3.org/OWL/>
- DBMS - Data Schemas. (n.d.). https://www.tutorialspoint.com/dbms/dbms_data_schemas.htm
- Microsoft. (n.d.). Visio'da UML diyagramları oluşturma - Microsoft Desteği. <https://support.microsoft.com/tr-tr/office/visio-da-uml-diyagramlar%C4%B1-olu%C5%9Fturma-ca4e3ae9-d413-4c94-8a7a-38dac30cbcd6>
- New AI Technology DDoS Protection System. (n.d.). <https://serdarsari.dev/erd-varlik-iliski-diyagrami-nedir/>

