SPL
Sermaye Piyasası
Lisanslama Sicil ve Eğitim Kuruluşu

Lisanslama Sınavları
Çalışma Notları

Bilgi Sistemleri Geliştirilmesi ve Uygulanması

102

Hazırlayanlar:
Cem ERGÜL, Sakine YÜKSEL, Nezihe UYGUN


SPL
Sermaye Piyasası
Lisanslama Sicil ve Eğitim Kuruluşu

Bilgi Sistemleri Geliştirilmesi ve Uygulanması

Ders Kodu: 1021

- Bilgi Sistemleri Bağımsız Denetim Sınavı

30 Haziran 2023

Bu çalışma notu Cem ERGÜL, Sakine YÜKSEL, Nezihe UYGUN tarafından hazırlanmıştır.

SINAV ALT KONU BAŞLIKLARI
BİLGİ SİSTEMLERİ GELİŞTİRİLMESİ VE UYGULANMASI

1. Proje Yönetimi
- **1.1.** Proje Planlaması
- **1.2.** Proje İşletimi

2. Sistem Geliştirme Yaşam Döngüsü
- **2.1.** Sistem Geliştirme Süreçleri
- **2.2.** Sistem Bakım ve Destek Süreçleri
- **2.3.** Uygulama Kontrolleri

İÇİNDEKİLER

KISALTMALAR

BPR : Business Process Reengineering (İş Süreçlerinin Yeniden Yapılandırılması)
BT : Bilgi/Bilişim Teknolojileri
CASE : Computer Assisted Software Engineering (Bilgisayar Destekli Yazılım Mühendisliği)
CBSD : Component Based Software Development (Bileşen Tabanlı Yazılım Geliştirme)
CM : Configuration Management (Konfigürasyon Yönetimi)
CPM : Critical Path Method (Kritik Yol Metodu)
CSA : Cloud Security Alliance (Bulut Güvenliği İttifakı)
EVA : Earned Value Analysis (Kazanılan Değer Analizi)
GUI : Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
ITIL : Information Technology Infrastructure Library (Bilgi Teknolojisi Altyapı Kütüphanesi)
OOSD : Object Oriented Software Development (Nesneye Yönelik Yazılım Geliştirme)
PERT : Project Evaluation Review Technique (Program Değerlendirme ve Kontrol Tekniği)
PMBOK : Project Management Body of Knowledge (Proje Yönetimi Bilgi Birikimi Klavuzu)
PMI : Project Management Institute (Proje Yönetim Enstitüsü)
POC : Proof of Concept (Kavram Kanıtı)
PUKÖ : Planla-Uygula-Kontrol Et-Önlem Al (PDCA: Plan-Do-Check-Act)
RAD : Rapid Application Development (Hızlı Uygulama Geliştirme)
RFP : Teklif Talebi (Request For Purchase)
SDLC : Software Develeopment Life Cycle (Yazılım Geliştirme Yaşam Döngüsü (YGYD)
SEE : Software Efford Estimation (Yazılım Çaba Tahminleri)
SMART : Specic, Measurable, Attainable, Relevant, Timebased (Belirli, Ölçülebilir, Başarılabilir/Gerçekçi, İlgili/Uygun, Süreli)
SOAP : Service Oriented Architecture Protocol (Servis Yönelimli Mimari Protokolü)
SoD : Seperation of Duties (Görevler Ayrılığı)
WDSL : Web Services Description Language (Web Servisleri Tanımlama Dili)
UAT : User Acceptance Test (Kullanıcı Kabul Testi)
UDDI : Universal Description Definition and Integration (Evrensel Tanım Tanımlama ve Entegrasyon)
YGYD : Yazılım Geliştirme Yaşam Döngüsü

Bu kitapta; proje yönetimi ve sistem geliştirme yaşam döngüsü konularına yer verilmektedir.

1. PROJE YÖNETİMİ

Proje kelimesinin kökenine bakıldığında Latince pro ve jectum kelimelerinin birleşmesinden oluştuğu görülmektedir. "*Pro*" ön, ileri gibi anlamlara gelirken, "*jectum*" ise fırlatmak, çıkarmak anlamlarına gelmektedir. Projectum, ileriye doğru fırlatma, çıkarma anlamını taşımaktadır.

Proje Yönetimi Enstitüsü (Project Management Institute-PMI) projeyi "*Proje benzersiz bir ürün, hizmet ya da sonucun üretildiği tekrarlı olmayan bir çalışmadır*" şeklinde tanımlamaktadır. Kalite kavramının fikir babalarından Juran projeyi, çözüm için çizelgelenmiş bir problem olarak ifade etmektedir. Bir diğer kaynakta ise proje, belirli kaynaklarla belirli bir zaman içerisinde tamamlanması gereken ve tekrarlanmayan özel faaliyetler topluluğu olarak tanımlamıştır. En genel anlamda proje, kaynakların etkin kullanımı, belirlenen takvimin takip edilmesi ve projenin sponsorunun beklentilerinin karşılanması başlıklarını içermelidir. Tüm proje tanımlarından da görüleceği üzere bir çalışmanın proje olabilmesi için aşağıda ifade edilen 4 özelliğe sahip olması gerekmektedir. Buna göre proje;

- Emsalsiz (özgün) bir ürün, hizmet ya da sonuç elde edilen,
- Tekrarlanmayan,
- Kısıtları olan (başlangıç, bitiş, bütçe, işgücü vb.),
- Planlama, yönetim ve kontrol ihtiyaçları olan ve pek çok faaliyetten oluşan

bir süreçtir.

Yönetim kelimesi ise Türkçe'de "*yön*" ve "*etmek*" kelimelerinin birleşiminden, yön göstermek, hedef göstermek temel anlamlarından gelmektedir. Bu anlamlar kapsamında değerlendirildiğinde, "*proje yönetimi*" ileride yapılacak bir işin yönünün ve hedeflerinin belirlenmesi, izlenmesi ve kontrolünü içermektedir.

Akademik olarak anlamı ise; sadece bir kez yapılan, belirli bir başlangıç ve bitiş tarihleri olan, hedefi ve amaç gözeten, kendine özgü işleri yer alan ve tüm bu işlerin planlanması, yürütülmesi, izlenmesi ve kontrolü ile kapatılmasını barındıran süreçler bütünüdür.

Proje yönetimi; proje kapsamındaki gereksinim ya da beklentilerin karşılanmasına yönelik ortak bir hedefe yönelmiş, örgütlenmiş etkinlikler kümesinin bilgi, beceri, araç ve tekniklerin belirlenen proje amaçları doğrultusunda zaman, maliyet ve kalite kısıtları aracılığıyla uygulanmasıdır. Projelerin başarılı bir şekilde yürütülmesi için uygun teknoloji kullanımı ve gerekli kaynakların tahsisinden başka, etkin ve başarılı bir proje yönetiminin de gerçekleştirilmesi gerekmektedir (Çimen, 1994:4). Proje yönetiminde temel amaç, tespit edilen amaçlara, sınırlı kaynaklarla, belli bir zaman içinde ve belli bir bütçeyle optimum şekilde ulaşmaktır.

İşletmelerde gerçekleştirilerek bilgi sistemleri denetimlerinde bilgi sistemleri denetçisinin proje yönetimi kapsamında işletme tarafından kullanılan standart veya yapıya aşina olmalıdır. Proje yönetim süreçleri aşağıdakileri aşamaları içerir:

- ❖ Başlatma,
- ❖ Planlama,
- ❖ Yürütme,
- ❖ Kontrol Etme,
- ❖ Kapanış.

Bilgi sistemleri denetçisinin, bilgi sistemleri denetimini gerçekleştirdiği işletmenin hedeflerinin bilgi sistemlerinin yönetim uygulamaları tarafından yerine getirildiğine dair makul güvence sağlaması için, işletmenin bilgi sistemlerini ve ilgili bileşenlerini nasıl değerlendirdiğini, geliştirdiğini, uyguladığını, idame ettirdiğini ve elden çıkardığını bilmesi büyük önem taşır. Bir projenin içeriğini analiz ederken, bilgi sistemleri denetçisinin aşağıdakiler hususları da göz önünde bulundurması gerekir:

❖ Projenin kuruluştaki önemi,

❖ İşletmenin stratejisi ile proje arasındaki bağlantı,

❖ Proje ve diğer projeler arasındaki ilişki,

❖ Proje ve altında yatan iş olurluğu arasındaki bağlantı.

1.1. Proje Planlaması

Proje yönetimini yönetimden ayıran önemli bir faktör devam eden bir süreç olan yönetimden farklı olarak, proje yönetiminin nihai, teslim edilebilir ve sınırlı bir zaman dilimine sahip olmasıdır.

Her projeye başlarken şu temel soruların cevaplandırılması gerekir:

- Neden proje başlatılıyor? (Proje paydaşları bu sorunun cevabı konusunda tartışıp anlaşmalıdırlar.)

- Bu iş kimin için yapılıyor? Nihai ürün, hizmet ya da sonuçtan faydalanabilecek olan proje sponsoru ve kilit proje paydaşları kimlerdir?

- Ne teslim edilecek? Tamamlaması gereken işler nelerdir?

- Bu teslimatları yaratmak için hangi fiziki ve finansal kaynaklara ihtiyaç var?

- Ne zaman bu teslimatlar üretilecek?

- Proje paydaşları onları ne zaman inceleyecekler?

- Proje sponsoru projenin nihai sonucunu ne zaman kabul edip onaylayacak?

- Nerede teslimatlar kullanılacaklar?

- Nasıl projenin amaçlar ve hedeflerine ulaşılacak?

- Başarı nasıl ölçülecek?

1.1.1. Projelerde Yönetişim Yaklaşımı

Başarılı proje planlamasının genel özelliği, riske dayalı bir yönetim süreci ve doğası gereği tekrarlanabilir olmasıdır.

Bu kapsamda bilgi sistemleri denetçisi, aşağıdaki proje yönetimi faaliyetlerinin yeterliliğini gözden geçirmelidir:

- Proje komiteleri/kurulu gözetim seviyesi,

- Risk yönetim yöntemleri,

- Sorun yönetimi,

- Maliyet yönetimi,

- Planlama ve bağımlılık yönetimi süreçleri,

- Raporlama süreçleri,

- Değişim kontrol süreçleri,

- Paydaş yönetiminin katılımı,

- Onay süreci.

1.1.1.1. Proje Yönetim Uygulamaları

Proje yaşam döngüsü boyunca kullanılan birçok proje yönetim tekniği ve araçları bulunmaktadır. Söz konusu teknik/araçlar projenin boyutuna, karmaşıklığına göre manuel veya otomatize sistemler olabilmektedir. Bahsedilen teknikler yazılım boyutu tahmini, zamanlama, kaynak tahsisi ve üretkenlik vb. hususlarda ölçüme dayalı nitel/nicel yaklaşımlar sağlamaktadır.

Proje yönetiminin çıktı, zaman ve bütçe olmak üzere üç temel kısıtı önem arz etmektedir. Projenin bütçe ve zaman kısıtları çıktıların özellikleri ile ilişkili olmalıdır. Genelde beklentisi yüksek olan çıktılar, uzun süre ve yüksek bütçe arasında pozitif bir korelasyon oluşturmaktadır. Projenin ihtiyaç duyduğu kaynaklar belirli tahmin teknikleri ile projenin başlangıcında tahminlenir. Bu kapsamda gerçekleştirilen boyut tahmini kaynakların toplamına ilişkin bir hesaplama sağlar.

a. Proje Portföyü ve Program Yönetim Kavramları

Proje portföy yönetimi, işletmenin genel stratejisiyle uyum içerisinde, portföy dengelerini gözeterek ve projenin değerini maksimize ederek dahili ve harici kısıtlara tabi olan bir girişimin başarısını ve genel zenginliğini artırmak amacıyla organizasyona ait projelerin yönetimi olarak da açıklanır (Moustafaev, 2017: 5).

Proje portföy yönetimi, esasında projenin yaşam döngüsünün çok ötesinde bir iş olup, bir yandan olası bir yatırım projesinin işletmenin faaliyet sahası arasındaki geleneksel boşlukları doldurmaya çalışırken öte yandan sınırlı kaynaklardan maksimum değeri sunma arasındaki bağı kurmaya çalışmaktadır (Levine, 2005: 13).

Proje portföy yönetiminde, klasik bir proje yönetiminin dikkate aldığı gibi projelerin ne zaman bitirileceği veya ne kadara mal olacağının dışında aşağıdaki sorulara cevap aranmaktadır (Levine, 2005: 16):

- Ne gibi potansiyel projelerin birleşimi, işletme için uzun vadede büyümeyi sağlamak ve yatırım getirisini maksimize etmek için insan ve nakit kaynaklarının en iyi şekilde kullanılmasını sağlar?

- Bu projeler işletmenin stratejik girişimlerini nasıl destekler?

- Bu projeler işletme paylarının değerini nasıl etkiler?

Proje portföy yönetiminin sorguladığı hususlar dikkate alındığında, proje yönetiminden farklı olarak, doğası itibariyle kötü bir projenin mükemmel yönetilerek belli bir düzeye getirilmesinden çok, hangi projenin işletme tarafından yüklenilmesi gerektiğine odaklanılmaktadır. Proje portföy yönetimi, hali hazırda yürütülmekte olan projeler ile entegrasyon sağlamanın önemini vurgulamasının yanı sıra, potansiyel projelerin işletmenin gelecekteki kârlılığına nasıl bir etkide bulunacağı temelinde düşünmeye neden olması bakımından da önemlidir ve olası projelerin nasıl seçilip yönetileceğine odaklanmaktadır (Levine, 2005: 17).

Proje portföy yönetimi, temel olarak iki bölüme ayrılabilir. Bunlardan ilki önceliklendirme ve portföye alınacak yatırım projesinin seçimi, ikincisi ise portföye alınan projelerin yönetimi (Levine, 2005: 23). Potansiyel yatırım projelerinin seçimine geçmeden önce portföyün sınırlarını belirlemek için her bir portföy doğası itibariyle bileşenleriyle birlikte karşılıklı ilişkiye sahip olduklarından proje portföy yöneticilerinin aşağıdaki eylemleri gerçekleştirmeleri gerekir (Chatzipanos, 2018: 20; Levine, 2005: 24):

1. Portföyün kendi içindeki fizibilite, açık teslim süreleri, yarattığı katma değer, sınırlılıklar gibi bileşenlerini kavramalı,

2. İşletmenin stratejik hedefleri de dahil olmak üzere bu bileşenler arasındaki ilişkileri kavramalı,

3. Portföyün çevre şartlarını anlamalı,

4. Tüm portföy ekosistemini kavramalı,

5. Portföy içinde projelerin birbiriyle olan sinerji durumlarını analiz etmeli,

6. Stratejik değişim sürecini de takip ederek portföyü ekosistem değişimlerine hazırlamalı,

7. Değerlendirme ve önceliklendirme yapmalı ve ilgili değişimlerin varlığını kanıtlamalı,

a. Değerlendirme ve önceliklendirmelerde projenin değer ile fayda takdirini yapmalı ve bunlara ilişkin risk analizi gerçekleştirmeli,

b. Değerlendirme ve önceliklendirme için en uygun kaynak kullanımını belirlemeli,

c. Seçim kriterleri setine uygun olarak projeleri sıralamalı.

8. Performanslarını da öngörerek söz konusu değişimleri hayata geçirmeli.

Projelerin seçiminden sonra iki boyutta proje portföyünü izlemeye almak gerekir. Birinci boyut, projenin kendi hedeflerini; ikinci boyut ise portföyün kendi hedeflerini karşılayıp karşılamamasıdır. Bu bakımdan birinci boyut Geleneksel Proje Takibi ve Kontrol Süreçleriyle kolaylıkla gerçekleştirilebiliyorken, ikincisi için Kazanılmış Değer Analizi (Earned Value Analysis-EVA) ve Ürün Geliştirme Süreci (Stage-Gate Process) gibi çok iyi bilenen teknikten yararlanmayı gerektirmektedir (Levine, 2005: 24-25). Kısaca özetlemek gerekirse proje portföy yönetimi, işletmenin değerini maksimize eden, portföy dengesini gözeten ve nihai olarak işletmeninin genel iş stratejileriyle uyumlu olan projelerin seçilmesi sürecidir (Moustafaev, 2017: 8).

Birden çok projenin bir arada yönetilmesi ve koordinasyonu ise programdır. Programlar sayesinde çok karmaşık ve birbirine göre farklı konulara sahip çalışmalar yönetilebilmektedir.

PMI'ye göre, bir program "***bireysel olarak yönetilmesinden elde edilemeyecek faydalar elde etmek için koordineli bir şekilde yönetilen bir grup proje***"dir.

Program yönetimi, stratejik hedeflerin ölçülebilir iş sonuçlarına çevrilmesidir ve sonucun gerçekleşmesi için gereken birçok ilgili girişimin entegrasyonu ile birliktedir. Bir programın yönetiminden sorumlu olan kişiye genellikle program yöneticisi adı verilir. Program yöneticisinin rolü kuruma bağlı olarak değişir.

Bazı işletmeler rolün ticari yönlerini vurgular. Bazı işletmeler ise, Bilişim Teknolojileri (BT) veya teknolojiye odaklanır; belirli teknik ve proje yönetimi niteliklerini vurgular. Program yöneticileri, yaklaşımın şekillendirilmesinden istenen sonuç kümesinin sunulmasına kadar çapraz işlevli programın uçtan uca ücretini yönlendirir. Program yöneticisi aşağıdakilerden sorumludur:

- ❖ Girişimlerin önceliklendirilmesi ve finanse edilmesi,
- ❖ Kuruluşlar arası bir yol haritası tanımlanması,
- ❖ Kaynak kapasitesinin ve kullanılabilirliğinin sağlanması,
- ❖ Projeler arasındaki bağımlılıkların yönetilmesi,
- ❖ Program düzeyinde hedeflere ulaşmasının sağlanması.

Program yöneticileri genellikle bir işletmenin Proje Yönetim Ofisi'ne (PMO) veya Stratejik Planlama Ofisine rapor verir ve bölümleri ve iş birimlerini kapsayan stratejik girişimleri yönetme sorumluluğunu taşır.

Portföy ve program yönetimi arasındaki farklar;

- Program yönetiminde birbirine benzer projeler, portföy yönetiminde ise benzer olmayan projeler veya farklı programlar yönetilmektedir.

- Program yönetiminin kapsamı, proje yönetiminden daha geniştir. Portföy ise işletmelerin stratejik hedefleri ile değişen organizasyon çapında bir kapsamı olmaktadır.

- Proje yöneticisi bir proje içinde birden fazla görevi yönetirken, program yöneticisi tüm projeler arasında koordinasyonu sağlar ve aynı amaç ve hedefe sahip olabilecek birbirleriyle ilişkili bağımlılıklara bakar. Genel olarak programlar daha uzun süre, bütçe, risk ve yüksek stratejik öneme sahiptir.

b. Proje Yönetim Organizasyonel Yapıları

Bilgi sistemleri denetçisi proje yönetim sürecine dahil olan grupların/bireylerin genel rollerine ve sorumluluklarına hâkim olmalıdır. Proje organizasyonuna bilgi sistemleri denetçisi de dahil edilebilir. Bilgi sistemleri denetçisi, sorumlu tarafların katılım seviyesini değerlendirebilmek için bağımsız bir inceleme gerçekleştirebilir. Bu durumda bilgi sistemleri denetçisi projede denetçi olarak değil danışman olarak görev alır. Organizasyon içerisindeki yetki ve kontrolü ana hatlarıyla belirten üç tür proje yönetimi organizasyon yapısı bulunmaktadır. Bu yapılara aşağıda kısaca değinilmektedir.

1. Fonksiyonel Yapılı Organizasyon

İş, fonksiyonel bazda bölümler arasında paylaştırılır. Proje yöneticisinin resmi olarak yönetim yetkisi bulunmayıp, sadece ekip üyelerine tanımlanacak faaliyetler konusunda tavsiyede bulunmaktadır.

2. Proje Yapılı Organizasyon

İş proje bazında yürütülür. Proje yöneticisinin, proje bütçesi, proje programı, proje ekibi ve projede yer alan hususlara ilişkin resmi olarak yetkisi bulunmaktadır.

3. Matris Yapılı Organizasyon

İş hem bölüm hem de proje bazında takip edilir. Proje yöneticisi ve bölüm yöneticileri arasında yönetim yetkisi paylaşılmaktadır.

c. Proje Paydaşlarının Rol ve Sorumlulukları

Projede yer alan grup/bireylerin rol ve sorumlulukları aşağıda belirtilmektedir.

Proje Yönlendirme Komitesi

Proje yönlendirme komitesi genel yönlendirmeyi sağlar. Bu komite; proje kapsamındaki tüm nihai teslimatlardan, maliyetlerden ve programlardan sorumludur. Proje sponsoru komiteye başkanlık eder. Bu komitenin yeni sistem alımları ya da sistem geliştirmeleri kapsamında kritik bir rolde olması sebebiyle; proje yöneticisi ve ilgili iş alanlarında karar alma yetkisine sahip kıdemli temsilcilerin komiteye atanması gerekmektedir. Komite genel olarak; proje takibinin yapılması, gerektiğinde koordinasyon/danışmanlık hizmeti verilmesi ve gerekli düzeltici faaliyetlerin sağlanması konusunda görevlidir.

Üst Düzey Yönetim

İşletme içerisinde mevcut kaynakların kullanılması için daima bir rekabet ortamı vardır. Proje sürecinde var olan yüksek belirsizlik oranı projeyi bu yarışın dışında bırakabilir. Projeye üst düzey yönetiminin vereceği destek, projenin sürekliliği ve başarısı ile proje ekibi tarafından proje amaç ve öneminin anlaşılmasında önemli rol oynamaktadır. Bu farkındalık ve üst yönetim desteği krizler ve çatışmaları önlemekte ve belirsizlikleri ortadan kaldırmaktadır.

Proje Sponsoru

Proje sponsoru, projeyi finansal ve kritik noktalarda temsil eder. Projeyi stratejik olarak yönlendirecek üst düzey yöneticidir. Proje sponsoru, proje için finansal ve idari desteği işletmenin üst yönetimi veya genel müdüründen almaktadır. Bu nedenle projenin ilerleyişi ve durumu hakkında devamlı olarak proje yöneticisi ve proje lideri (varsa) dirsek temasında olması gerekir. Proje sponsoru, proje yönetimi ekibiyle çalışarak, proje finansmanı, faaliyet alanının belirginleşmesi, üçüncü kişilerin projeden faydalanması için teşvik edilmesi gibi konularda yol göstermektedir.

Kullanıcı Yönetimi

Kullanıcı yönetimi, proje ekibinin faaliyetlerini (gereksinimlerin tanımlanması, tasarım, test senaryolarının tanımlanması, kabul testi, kullanıcı eğitimi vb.) takibinden sorumludur.

Kullanıcı Proje Ekibi

Kullanıcı proje ekibi, projede çalışan, projenin başarısına katkıda bulunan kişilerdir. Tüm ekip üyeleri almış oldukları görevlerin durumlarının proje yöneticisine raporlanmasından sorumludur.

Proje Yöneticisi

Proje yöneticisi, projenin amaç ve hedeflerine ulaşmasının sağlanmasından sorumludur. Bir proje yöneticisinin başarısı, gereksinim ve beklentilerin doğru tanımlanması, ulaşılabilir hedeflerin belirlenmesi, etkili ve açık iletişim kurulması, kapsam, maliyet, zaman ve kalite kısıtları dahilinde planlamanın etkin ve verimli şekilde yapılabilmesine bağlıdır.

Proje yöneticisinin belirlenmesi sırasında aşağıda sıralanan nitelikler aranır:

- Altyapı ve Deneyim: Proje yöneticisinin proje yönetimi konusunda yeterli altyapı ve deneyime sahip olması gerekmektedir.

- Liderlik ve Stratejik Uzmanlık: Proje yöneticisi, proje ekibi üyelerinin doğrudan saha yöneticisi olmayabilir. Farklı bölümlerden proje ekibine katılan kişilerin yönetilebilmesi için etkin liderlik göstermesi beklenir. Proje yöneticisi, projenin kurumun stratejik planları ile bağlantılı gitmesine de özen göstermelidir.

- Teknik Uzmanlık: Proje yöneticisinin proje konusunda teknik uzman olması gerekmemektedir. Ancak proje sırasında olup biteni takip edebilecek, doğru soruları soracak ve verilen cevapları anlayacak derecede konuya hakim olmalıdır.

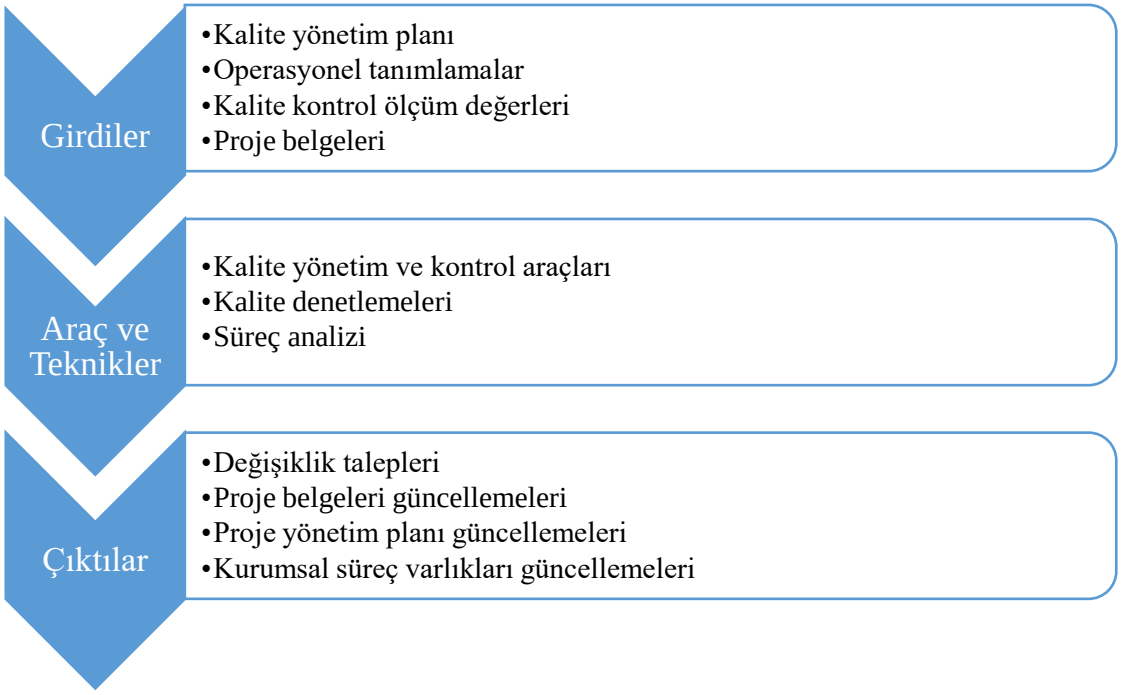
- İletişim Becerisi: Proje esnasında proje yöneticisi kendi ekibiyle, diğer ekiplerle ve üst yöneticilerle yoğun iletişim içinde olacaktır. Bu iletişimdeki etkinlik projenin başarısı için çok önemlidir.

- Yönetsel Beceri: Proje yöneticisinin stratejik planlama, bütçe, insan kaynakları yönetimi, kalite yönetimi vb. birçok konuda yeterli bilgi ve birikimi bulunmalıdır.

Kalite Güvence

Kalite güvence, proje çıktılarının incelenerek gereksinimlere uygunluğunun takibinin yapılması ve ilgili organizasyon süreçlerinin teslimatından sorumludur. Proje esaslarına bağlı kalarak kalite güvencesinin sağlanması ve kalite kontrolü gibi alt süreçler içermektedir.

Bir kalite güvence uygulamasının yapılması aşağıdaki gibidir:



Kalite yönetimi ve proje yönetimi arasında benzer özellikler bulunmaktadır. Aşağıda sıralanan kavram ve yaklaşımlar her ikisi için de kullanılabilir:

- Müşteri Memnuniyeti: Proje, müşterinin ihtiyaçlarının giderilmesi konusundaki beklentilerini söz verilen biçimde karşılamalıdır.

- Önlem Alma: Kalite proje içinde planlanır ancak denetlenmez. Hatanın önlenmesi düzeltilmesinden her zaman daha az maliyetlidir.

- Bazı işletmelerde kalite faaliyetleri kalite güvence bölümü tarafından yapılmaktadır. Kalite güvence, hataların bulunması ve giderilmesi ile ilgilidir. İki tür kalite güvence vardır:

- İç Kalite Güvence: Yönetime ve proje ekibine sağlanan güvencedir.

- Dış Kalite Güvence: Proje müşterilerine sağlanan güvencedir.

Kalite güvence uygulamasının yapılması sürecinin, proje yöneticisi ve proje ekibi tarafından hazırlanması gereken üç temel girdisi bulunmaktadır:

- Kalite Yönetim Planı: Proje ekibinin kalite politikasını nasıl uygulayacağı tanımlayan plandır.

- Kalite Kontrol Ölçüm Değerleri: Bu değerler kalite kontrol testlerinden sağlanmaktadır. Değerler karşılaştırılabilir ve analiz edilebilir nitelikte olmalıdır.

- Operasyonel Tanımlamalar: Projenin süreçlerini tanımlayan ölçütler, bunların değerleri ve ölçüm birimleri kalite güvence için gerekmektedir. Kalite güvence gerekliliklerini kalite güvence bölümü veya bazı durumlarda proje yöneticisi hazırlamaktadır. Kalite güvence için maliyet kazanç analizi, akış diyagramı, deney tasarımı vb. gibi yaklaşımlar kullanılmaktadır.

Kalite kontrolleri öğrenme amaçlıdır. Kalite kontrollerindeki temel fikir proje süresince oluşan hatalardan öğrenilenlerin, bu proje ve ilerleyen diğer projeler için işletme içinde içselleştirilerek hayata geçirilmesiyle hataların tekrarını önlenmektedir. Kalite kontrolleri proje süresince önceden belirlenmiş sıklıklarla veya haber verilmeden de yapılabilmektedir. Kalite kontrollerini, iç denetçiler yapabileceği gibi dış kaynaklardan da hizmet alımı yapılabilmektedir.

Proje ekibinin kalite kontrol için kullanacağı bazı beceriler ve teknikler aşağıdaki şekilde sıralanmaktadır:

- İstatistiksel kalite kontrol teknikleri,

- Hatalı ürünlerin müşteriye ulaşmasına engel olacak kontroller,

- Kalitede anormalliklere ve dalgalanmalara yol açan nedenlerin tespiti,

- Kalitenin istenen seviyede olması için gerekli kontrol sınırlarının belirlenmesi.

Sistem Geliştirme Yönetimi

Sistem geliştirme yönetimi, proje kapsamında gereksinim duyulan sistem/uygulamanın geliştirilerek, işlerliğinin sağlanması, stratejik BT bakış açısıyla uyumluluğun güvencesinin verilmesi, bakım hizmetleri gibi teknik destek faaliyetlerinin yürütülmesinden sorumludur.

Sistem Geliştirme Proje Ekibi

Sistem geliştirme proje ekibi, projede sistem geliştirme yönetimi tarafından atanan görevlerin yürütülmesinden sorumludur.

Güvenlik Yöneticisi/Güvenlik Ekibi

Güvenlik yöneticisi veya güvenlik ekibi, sistem kontrollerinin ve destek süreçlerinin, kurumsal güvenlik bakış açısına uygun şekilde tesis edilmesinden sorumludur.

Bilgi Sistemi Güvenlik Mühendisi

Bilgi sistemi güvenlik mühendisi, güvenlik açıklarının belirlenmesi ve bu açıklıklara ilişkin riskin en aza indirilmesi veya sınırlamak için bilimsel ve mühendislik ilkelerini uygulamaktadır. Bu rolü yerine getirmenin anahtarı hem geniş alanda savunulması hem de derinlemesine güvenlik ilkelerine göre ağ, ortam ve uygulama yapılarını inşa edilmesi için ihtiyaçların, gereksinimlerin, mimarilerin ve tasarımların tanımlanmasıdır.

d. Proje Yönetim Ofisi (PMO)

Proje Yönetim Ofisi, projeyle ilgili yönetişim süreçlerini standartlaştıran, kaynak, metodoloji, araç ve tekniklerin paylaşılmasını kolaylaştıran bir yönetim yapısıdır. PMO, yürürlükteki standart ve prosedürlerin korunması, geliştirilmesine yönelik yeterli kaynağa sahip olmalıdır. PMO'nun amacı yönetilen süreçlerin kalitesinin iyileştirilmesi ve başarının güvenceye alınmasıdır. PMO proje/program içeriğinden daha çok faaliyet/görevlere odaklanır.

Bilgi sistemleri denetçisi, denetime ilişkin projelerin içeriğinin yanı sıra prosedüre ilişkin yönleri de değerlendirmelidir.

Proje portföy yönetiminin hedefleri aşağıdakilerden oluşmaktadır:

- Bireysel olmayan proje portföy sonuçlarının iyileştirilmesi,

- Projelerde önceliklendirme ve zaman planlamasının yapılması,

- İç ve dış proje kaynak koordinasyonlarının sağlanması,

- Proje süresince bilgi akışının sağlanması.

Proje portföy yönetimi için bir proje portföy veritabanı zorunludur. Veritabanı, projelerin sahiplik, program, hedef, tür, durum ve maliyet gibi verileri içermelidir. Çubuk grafik, kar ve risk matrisi ve proje portföyü gibi ilerleme grafikleri gibi tipik proje portföy raporları proje portföy yönetimi kapsamında hazırlanmaktadır.

e. Proje Faydalarının Tanımlanması

Proje faydalarının gerçekleştirilmesinin hedefi, BT'nin ve iş birimlerinin değer yönetimi sorumluluklarının yerine getirilmesini sağlamaktır. Hem programa hem zamana duyarlı pazara, sektör ve yasal gerekliliklere göre zamanında, bütçe dahilinde bulunan yetenekler daha büyük katkı sağlamaktadır. Söz konusu BT hizmetleri ve varlıkları iş değerine de katkıda bulunmaktadır.

1.1.2. Proje Planlanma Metotları

Proje planlaması kapsamında, proje yöneticisi tarafından proje kapsamı (paydaşlar ile mutabakat sağlanarak), hedeflenen iş uygulamasının gerçekleştirilebilmesi için yapılacak görevlendirmeler, görevlerin sırası, düzeni, süresi ve önceliği, BT ve BT dışı kaynaklar, bütçe ve maliyet, proje kapsamındaki (işçilik, hizmet, materyal, tesis ve ekipman vb.) giderlerin fonlama kaynağı belirlenmelidir.

Doğru kararların alınması ve buna uygun faaliyetlerin yürütülebilmesi için projelerin mutlaka planlama aşamasından geçmesi gerekmektedir. Planlama, proje süresi boyunca yalnızca bir kere yapılan ve proje sonuna kadar hiç değişikliğe uğramadan uygulanan bir nitelik taşımaz. Proje planı, proje süresince yapılan kontroller sonucunda elde edilen bilgiler ve değişen dış etkenler uyarınca, devamlı olarak değerlendirme, gözden geçirme ve yenilenme süreçleri içerisinde bulunmaktadır. Böylece plan, zamanla uygulanamaz bir nitelik kazanmak yerine, sürekli güncellenerek proje tamamlanana kadar uygulanabilirliğini korumaktadır. Planlama yapılmadığı takdirde gelecekteki fırsatları ve tehlikeleri görmek mümkün olmayacağından, bu konuda gerekli önlemler de alınamayacaktır (Barutçugil, 1984:162). Neyin, niçin, nasıl ve ne zaman yapılacağını tanımlayan, projedeki işlerin yürütülmesini ve projedeki çalışanların yönetimini sağlayan planlama çalışmaları yapılmaksızın, projenin başarılı bir şekilde yürütülmesi ve sonuçlandırılması mümkün değildir. Proje planının geliştirilmesinde, görev ve sorumlulukların belirlenmesi, proje zaman cetvelinin hazırlanması ve proje bütçesinin çıkarılması en önemli çalışmalar arasındadır (Barutçugil, 1988:239-240).

Proje planlamada nihai hedef üç farklı şekilde olabilir;

- Eldeki kaynaklar çerçevesinde projenin en kısa zamanda bitirilmesi,

- Daha önceden belirlenmiş bir proje süresi dahilinde en az kaynak kullanımı ile projenin bitirilmesi,

- Proje toplam maliyetini en az yapacak bir proje süresinde projenin bitirilmesi.

Projenin planlama aşaması, belirlenen amaçlara göre projeyi teşkil eden faaliyetlerin birbirleriyle olan mantıksal ilişkileri, tamamlanma süreleri, kaynak gereksinimleri ve maliyetleri göz önünde bulundurularak gerçekleştirilir. Dolayısıyla projenin planlama aşamasında gereksinim duyulan bilgi ve verilerin elde edilme sürecinde çalışmaya ışık tutacak soruların sorulması önemlidir. Örneğin; "*Proje ne gibi faaliyetlerden oluşmaktadır?*", "*Faaliyetlerin birbirleriyle olan ilişkileri nasıldır?*", "*Her faaliyetin ne tür bilgi sistemi gücüne gereksinimi vardır?*", "*Her faaliyetin hangi miktarda işgücüne*

ihtiyacı vardır?", "*Faaliyetlerin ve projenin bütününün maliyet unsurları nelerdir?*" gibi soruların sorulması ve cevap bulması bir projenin etkin bir şekilde planlanabilmesi için gerekmektedir.

Özet olarak, planlama safhasında proje, ana faaliyet gruplarına ayrılır ve her grup içindeki temel faaliyetler tespit edilir. Her bir faaliyetin tamamlanma süresi ve gerektirdiği kaynak miktarı belirlenir. Daha sonra faaliyetlerin birbirleriyle olan mantıksal ilişkileri göz önünde bulundurularak faaliyet sıraları belirlenerek tüm proje bu mantıksal ilişkiler ve faaliyet öncelik sıralaması ışığında bir bütün olarak ifade edilir. Böylece proje şebekesi kurularak proje programlama aşamasına zemin hazırlanmış olur.

Proje programlama, kaynak gereksiniminin ve tahmin edilen süre içinde projenin gidişatının programlanmasıdır. Proje programlamada ilk aşama, her bir faaliyet için gerekli süreyi belirlemektir (Grow, 1975:185). Ayrıca bu aşamada, her faaliyetin başlama ve bitiş zamanını gösteren bir zaman diyagramı hazırlanmaktadır. Proje programı, proje açısından önem arz eden kritik faaliyetleri göstererek, faaliyetlerin serbestlik süresi ve gecikme miktarı hakkında bir fikir vermelidir (Monks, 1996:352; Halaç, 1995:184). Bu şekilde projenin zamanında bitirilebilmesi için dikkatle takibi gereken ve zaman açısından kritik olan faaliyetlerin tanımlamaları yapılmaktadır. Ayrıca kritik olmayan faaliyetlerin sahip oldukları boş zamanların tespiti sınırlı kaynakların mümkün olduğu kadar projenin daha kritik olan işlemlerine dağıtılabilmesini sağlamaktadır.

Programlama, her faaliyetin yalnızca ne zaman tamamlanması gerektiğini göstermez, aynı zamanda "*boşluk zamanları*" da göz önüne alınarak, bir faaliyetin "*en erken başlama*", "*en geç başlama*", "*en erken bitirme*", "*en geç bitirme*" tarihlerini de belirtmektedir. Böylece var olan insan gücü ve kaynaklardan birlikte yararlanacak faaliyetler arasında sıralamalar yapmak suretiyle, sınırlı insan gücü ve kaynaklardan en çok faydalanma gerçekleştirilmiş olur.

Bir proje programı aşağıdaki unsurlar hakkında bilgi vermelidir (Jensen, 1994):

- Proje aşamalarının ayrıntıları,

- Faaliyet sayısı,

- Faaliyet süreleriyle ilgili minimum ve maksimum tahminler,

- Hedef süreyle ilgili kısıtlamalar ve mantıksal ilişkiler,

- Kaynak ve maliyet kısıtlamaları,

- İyileştirme yöntemleri.

Programlama aşamasında, projenin hedeflenen süre içerisinde bitirebilmenin mümkün olup olmadığı belirlenebilmekte ve proje programının ön gördüğü tamamlanma süresinin kısaltılması ihtiyacı doğduğu zaman hangi faaliyetlere hızlandırma işlemi uygulanması gerektiği ve bu işlemi gerçekleştirmenin maliyeti tespit edilebilmektedir.

Proje programının sağladığı avantajlar aşağıdaki gibi ifade edilebilir (Monks, 1996):

- Tüm projeyi ve birbirleriyle ilişkili faaliyetleri koordine etmektedir.

- Tüm faaliyetlerin mantıklı bir biçimde planlanmasını sağlayarak faaliyetlerin organize edilmesini kolaylaştırmaktadır.

- Öncelik ilişkilerini ve özelikle kritik olan faaliyetlerin sırasını tanımlamaktadır.

- Gerçek değerlerle karşılaştırma yapabilmek için projenin tamamlanma süresinin (veya maliyetinin) tahminiyle ve bu konudaki standartlarla ilgili bilgileri sağlamaktadır.

- İnsan gücü, malzeme ve mali alanda yer değiştirebilecek kaynakları tanımlayarak kaynakların daha iyi kullanılmasını sağlamaktadır.

Projenin programlama aşaması tamamlandıktan sonra projenin kritik yolu üzerinde önemle durulması gerekmektedir. Proje yöneticisinin projeyi olası en kısa sürede ve en düşük maliyetle tamamlayabilmesi için kritik yolun üzerindeki faaliyetlere hızlandırma işlemi uygulanması ve projenin tamamlanma süresinin hedef süreye uygun şekilde ayarlanması gerekmektedir.

1.1.2.1. Proje Kapsamının Belirlenmesi ve Fizibilite Çalışması

Proje başlangıcındaki onay sonrasında ihtiyacı ve bu kapsamdaki alternatif çözümleri belirlemek amacıyla fizibilite analizi gerçekleştirilmektedir. İş olurluk incelemesi genelde fizibilite analizine dayanmaktadır. Üretilen çözümün ihtiyaca ve tanımlanan bütçe, zaman kısıtlarına uygunluğunu analiz edebilmek amacıyla gerçekleştirilen ön çalışmadır.

Fizibilite çalışması aşağıda belirtilen altı maddeyi içerecektir:

1- Proje Kapsamı: İş sorununun tanımı veya ele alınması gereken fırsattır. Açık, kısa ve doğru olmalıdır.

2- Mevcut Analiz: Sistemin/yazılım ürününün vb. bir anlayışın tanımlanması ve oluşturulmasıdır. Bu analize dayanarak, mevcut sistem veya yazılım ürününün doğru çalıştığı, bazı küçük değişikliklere ihtiyaç olduğu veya tam bir yükselme/değiştirme gerektiği belirlenebilir.

3- Gereksinimler: Paydaşların ihtiyaçlarına ve kısıtlamalarına ilişkin proje gereksinimlerinin tanımlanmasıdır. Yazılım ve sistem için gereksinimler farklı şekillerde belirlenmektedir. Örneğin;

- İş, sözleşme ve mevzuat süreçleri,

- Son kullanıcı fonksiyonel ihtiyaçları,

- Operasyonel ve mühendislik parametrelerini tanımlayan teknik ve fiziksel öz nitelikler.

4- Yaklaşım: Önerilen sistem/yazılım çözümü için gereksinimleri karşılamak üzere eylem planının tanımlanmasıdır. Bu adım, dikkate alınan alternatifleri ve tercih edilen çözümün seçilme gereksinimini açıkça tanımlamaktadır. Bu, mevcut yapıların ve ticari alternatiflerin kullanımının dikkate alındığı süreçtir.

5- Değerlendirme: Fizibilite çalışmasında önceden tanımlanmış öğelere dayanarak, projenin maliyet etkinliğinin incelenmesidir. Nihai rapor, seçilen yaklaşımın maliyet etkinliğini ele almaktadır. Nihai raporun öğeleri şunları içerir:

- Tercih edilen çözüm seçilirse, projenin tahmini toplam maliyeti ve aşağıdakiler dahil bir maliyet karşılaştırması sağlama alternatifleri:

- Tamamlanması için gerekli çalışan saatleri,

- Materyal ve olanak maliyetleri,

- Tedarikçi ve üçüncü taraf yüklenici maliyetleri,

- Proje zaman programı başlangıç ve bitiş tarihleri

- Maliyet-fayda analizini, yatırım getirisini vb. kapsayan bir maliyet değerlendirme özeti.

6- İnceleme: Fizibilite çalışmasının daha önce tanımlanmış öğelerinin hem fizibilite çalışmasının eksiksizliğini ve güvenilirliğini doğrulamak hem de nihai kararı vermeden önce projeyi onaylama/reddetme/düzeltme/isteme kararı vermek için gözden geçirilmesidir. İnceleme ve rapor kilit paydaşlarla birlikte yapılmaktadır. Fizibilite çalışmasının reddedilmesinin gerekçesi açıklanmalı ve gelecekteki proje çalışmalarında kullanılmak üzere derse alınanların listesinin parçası olarak belgeye eklenmelidir.

Fizibilite çalışması sırasında, bilgi sistemleri denetçisi aşağıdakileri yapmalıdır:

- Uygun olduğundan emin olmak için faz belgelerinin gözden geçirilmesi,

- Tüm maliyet gerekçelerinin/faydalarının doğrulanabilir olup olmadığının ve beklenen maliyetleri ve beklenen faydaları gösterip göstermediklerinin belirlenmesi,

- İhtiyacın kritikliğinin belirlenmesi ve tespit edilmesi,

- Hali hazırda mevcut olan sistemlerle bir çözüm elde edilip edilmeyeceğinin belirlenmesi, çözüme ulaşılamamışsa, alternatif çözümlerin makul olup olmadığına ilişkin değerlendirmenin gözden geçirilmesi,

- Seçilen çözümün uygunluğunun tespit edilmesi.

Fizibilite çalışmasında ihtiyaç tanımına ilişkin uygulanması sebebiyle bilgi sistemleri denetçisi aşağıdaki işlevleri yerine getirmelidir:

- Ayrıntılı gereksinimler tanımı belgesinin edinilmesi ve ilgili bölümler ile görüşmeler yaparak doğruluğunun ve tamlığının incelenmesi,

- Proje ekibindeki kilit üyelerin belirlenmesi ve etkilenen tüm kullanıcı gruplarının uygun temsile sahip olduğunun doğrulanması,

- Projenin maliyeti ve başlatılmasına ilişkin onayların alındığının teyit edilmesi,

- Kullanıcının gereksinimlerini karşıladığından emin olmak için kavramsal tasarım spesifikasyonlarının incelenmesi,

- Kontrol spesifikasyonlarının tanımlandığından emin olmak için kavramsal tasarımın incelenmesi,

- Makul sayıda tedarikçiden proje kapsamını ve kullanıcı gereksinimlerini kapsayan teklifin alınıp alınmadığının incelenmesi,

- Kullanıcı kabul testi spesifikasyonlarının incelenmesi,

- Uygulamanın gömülü bir denetim rutini kullanımına aday olup olmadığının belirlenmesi, eğer adaysa rutinin sistemin kavramsal tasarımına dahil edilmesinin istenmesi.

1.1.2.2. Sistem Geliştirme Projesi Maliyet Tahminleri

Bilgi sistemi projelerinde maliyet tahmini için kullanılan dört farklı yöntem bulunmaktadır. Bunlar:

1- Benzer Tahmin: Benzer tahmin, proje yöneticisinin geçmiş projelerine yönelik tahminleri ile maliyetin planlanmasıdır. En hızlı yöntem olarak değerlendirilmektedir.

2- Parametrik Tahmin: Parametrik tahmin, proje yöneticisinin benzer tahminlerde kullanılan geçmiş veri analizine dayalı olarak gerçekleştirdiği planlamadır. Bu yaklaşım benzer tahminden daha doğru olarak değerlendirilmektedir.

3- Aşağıdan Yukarıya Tahmin: Bu yöntemde, projedeki her bir faaliyetin maliyeti en büyük ayrıntıya kadar tahminlenir ve maliyete eklenmektedir. En çok zaman alan bu yaklaşım en doğru tahmin yöntemidir.

4- Gerçek Maliyet: Gerçek maliyet, geçmiş projeler sırasında aynı sisteme ilişkin gerçekleşen gerçek maliyetlerin değerlendirilmesi yöntemidir.

Proje maliyetlerinin tahminine ilişkin olarak, aşağıda da açıklamalarına yer verilen, yazılım boyutu tahminleri, fonksiyon nokta analizi ve yazılım maliyet tahminlerine ilişkin çalışmalar yapılır.

a. Yazılım Boyutu Tahminleri

Yazılım boyutu tahmini, bir yazılım ürünü geliştirmek için gereken çabayı belirlemek için önemli bir özelliktir. Yetersiz, sorgulanabilir ve usulsüz veriler ışığında kalkınma görevlerini oluşturulması veya sürdürülmesi için gerekli olan en pratik efor ölçüsünün (bireysel saat ya da sermaye olarak iletilen) öngörülmesini sağlayan metodolojidir. Yazılım Çaba Tahminleri (Software Effort Estimation-SEE), yazılımı geliştirmek veya sürdürmek için gereken çabanın en makul şekilde kullanımını öngören prosedürdür.

Yazılım boyutlandırma veya yazılım boyutu tahmini, diğer yazılım proje yönetimi faaliyetlerinin (tahmin etme veya izleme gibi) uygulayabilmesi için bir yazılım uygulaması veya bileşeninin boyutunun belirlenmesi veya tahmin edilmesi için kullanılan yazılım mühendisliğindeki bir faaliyettir. Boyut, tıpkı ağırlığın somut bir malzemenin doğal özelliği olması gibi, bir yazılım parçasının doğal bir özelliğidir.

Geliştirilecek uygulama yazılımının göreceli fiziksel büyüklüğünü belirlenmesini sağlamaktadır. Tahminler, kaynakların tahsisine rehberlik etmesi, geliştirme için gereken zaman ve maliyetin değerlendirilmesi ve kaynakların gerektirdiği eforun karşılaştırılması için kullanılabilmektedir.

b. Fonksiyon Nokta Analizi

Büyük iş uygulamalarının geliştirilmesinde karmaşıklığı tahmin etmek için kullanılan çok noktalı bir tekniktir. Bu analiz bir bilgi sistemi boyutunun, kullanıcının etkileşimde bulunduğu girdi, çıktı, arayüz, dosya ve sorguların sayı ve karmaşıklığına bağlı ölçüsüdür. Bu doğrudan boyut odaklı ölçümlere karşı yazılım boyutunun ve geliştirme sürecinin dolaylı ölçüsüdür. Fonksiyon noktaları ilk önce belirli bir girişin basit, ortalama veya karmaşık olup olmadığının belirlenmesi için bir tablo tanımlanarak hesaplanmaktadır. Kullanıcı girişi, kullanıcı çıktı, kullanıcı sorgu, dosya ve harici arayüz sayıları dahil olmak üzere beş fonksiyon noktası sayım değeri tanımlanmaktadır.

Tablo girişlerinin tanımlanmasının ardından, fonksiyon noktasının türetilmesindeki toplam sayı güvenilirlik, kritiklik, karmaşıklık, tekrar kullanılabilirlik, değiştirilebilirlik ve taşınabilirlik gibi konulara ilişkin sorulara verilen yanıtlara dayalı olarak karmaşıklık ayarlama değerlerini (derecelendirme faktörleri) dikkate alan bir algoritma ile hesaplanmaktadır. Bu denklemden türetilen fonksiyon noktaları daha sonra maliyet, program, üretkenlik ve kalite kriterleri için bir ölçüt olarak Source Line of Code (Kaynak Kod Satır Sayısı-SLOC) sayılarına benzer şekilde kullanılmaktadır.

Fonksiyon nokta analizi, iş uygulamalarının tahminlenmesinde oldukça başarılı bir yöntemdir. Ancak diğer yazılım türleri (işletim sistemi, süreç kontrolü, iletişim ve mühendislik vb.) için yeterli değildir. Diğer tahmin yöntemleri bu tür yazılımlar için daha uygundur ve Constructive Cost Model (Yapıcı Maliyet Modeli-COCOMO), De Marco ve Watson-Felix'in fonksiyon nokta analizi özelliklerini içermektedir.

Fonksiyon Nokta Analizi Ölçütlerinin Hesaplanması					
Ölçüm Parametresi	**Sayı**	**Ağırlık Faktörü**			
		Basit	**Ortalama**	**Karmaşık**	**Sonuçlar**
Kullanıcı girdi sayısı		3	4	6	…..
Kullanıcı çıktı sayısı		4	5	7	…..
Kullanıcı sorgu sayısı		3	4	6	…..
Dosya sayısı		7	10	15	…..
Harici arayüz sayısı		5	7	10	…..
Toplam sayı	FP yöntemlerini kullanan kuruluşlar, belirli bir girişin basit, ortalama veya karmaşık olup olmadığını belirlemek için kriterler geliştirir.				

c. Yazılım Maliyeti Tahminleri

Yazılım maliyeti, yazılım boyutu tahmininin sonucudur. Proje kapsamının en uygun şekilde belirlenmesini sağlamaktadır Bilgi sistemi geliştirme projelerinin her aşamasında maliyet tahmini yapılmaktadır. Bu tahminleme için kullanılan otomatik teknikler bulunmaktadır. Söz konusu otomatik maliyet tahmini tekniklerinin kullanılabilmesi için bilgi sistemi genellikle ana bileşenlerine ayrılmakta ve bir dizi maliyet sürücüsü oluşturulmaktadır. Ayrılan bileşenlerin içerikleri aşağıda belirtildiği gibidir:

- Kaynak kod dili,
- Yürütme süresi kısaltmaları,
- Ana depolama kısaltmaları,
- Veri depolama kısaltmaları,
- Bilgisayar erişimi,
- Geliştirme için kullanılan hedef makine,

- Güvenlik ortamı,

- Personel deneyimi.

Tüm sürücüler belirlendikten sonra program bilgi sistemi ve toplam projenin maliyet tahminlerini geliştirecektir.

1.1.2.3. Zaman Çerçevesinin Oluşturulması ve Çizelgelenmesi

Bütçeleme, her göreve dahil olan insan ve makine eforunun toplamı olup, zamanlama ve görevler arasındaki sıralı ilişkidir. Bu görevlerin belirlenmesini sağlayan iki öğeye göre düzenlenmektedir. Bunlar:

- Görevler arası mantıksal sıralı ilişkiye dayalı ve görevlerin paralel yürütülmesi halinde en erken başlangıç tarihi,

- Çalışan dahil tüm kaynakların uygunluğuna göre (izin/tatil süreçleri, işe alım süresi vb. dahil edilerek) en son beklenen bitiş tarihi.

Program, aşağıda da belirtilen teknikler (Gantt şeması, Critical Path Method (Kritik Yol Metodu-CPM), Project Evaluation Review Technique (Program Değerlendirme ve Kontrol Tekniği PERT) vb.) kullanılarak grafiksel olarak da gösterilebilir. Proje sürecince durumun analiz edilmesi önemlidir. Kilit noktalar/ kritik faaliyetler ve kilometre taşları analiz edilerek program ve bütçedeki sapmalar analiz edilmektedir. Varyanslar ve varyans analizi yönetime zamanında bildirilmelidir.

1.1.2.4. Gantt Şemaları

Gantt şemaları, bir projenin tamamlanması için gerekli aktivitelerin planlanmasını desteklemektedir. Projelerin planlanması, projenin zamanlamasını ayarlanması, kaynakların belirlenmesi ve projenin yönetimini sağlanması amacıyla tasarlanan Gantt şeması proje yöneticileri için kurtarıcı bir diyagramdır.

Gantt şeması bir bakışta, bireysel faaliyetlerin ne olduğunu, hangi aşamada kim tarafından yapılacağını, bu faaliyetlerin her birinin ne zaman başlayıp bitmesi gerektiğini, planlanan zaman ve geçen fiili zamanın karşılaştırılması ve buna göre aksiyon alınması gibi durumların planlanmasını, faaliyetler arasında herhangi bir çakışma olup olmadığını ve projenin belirli bir süre içinde nasıl tamamlanabileceğini göstermektedir. Aynı zamanda hangi aktivitelerin eş zamanlı ve hangi sırayla yürütüleceğini de göstermektedir. Gantt şeması, bir projede ihtiyaç duyulan tüm özelliklerin tek bir çizelgede gösterilmesini sağlar. Bu özellikler aşağıdaki gibidir:

- Kilometre Taşları (Milestones): Projeyi bir aşamadan başka bir aşamaya taşıyan olay, tarih, karar veya teslimlerdir.

- İş Paketleri (Work Package): Birbiriyle alakalı yapılacak işler bütünüdür. Herhangi bir eylem barındırmaz.

- Görev Listesi (Task List): Yapılacak işlerin listesini gösterir. İş paketlerinin altında yer alan eylem gerektiren listedir.

- Zaman Çizelgesi (Timeline): Çizelgenin üstünde bulunan projenin zamanlarını gün, hafta, ay ve yıllık bazda gösteren bir özelliktir.

- Şimdi Çizgisi (Dateline): Genellikle kırmızı renkli dik bir çizgi olarak gösterilir. Bu çizginin solunda kalan işler tamamlanmış olması gerekmektedir.

- Bağlantılar (Dependencies): Bir işe başlamadan önce bir başka işin bitirilmesi gerektiği durumlarda kullanılan terimdir. Gantt şemasında bu ilişkileri gösterebilmek için bağlantıları da kullanmak mümkündür. Bağlantılar ile görevler arasındaki ilişkiyi görebilir, bu görevlerin erken bitmesi/gecikmesi durumunda bir sonraki görevlerin nasıl etkilenebileceğine dair tarih hesaplaması otomatik olarak yapılabilir.

Gantt şemaları aynı zamanda göreve atanan kaynakları ve atamaların yüzde kaçını yansıtabildiği gibi, bir taban çizgisine kıyasla erken ya da geç tamamlanan faaliyetleri belirlemeye yardımcı olabilir.

Gantt şemaları tüm projenin izlenmesini sağlayacağı gibi, projenin belirli bir aşamasının ya da kritik bir sürecinin başarısının izlenmesini de sağlar.

Gantt şemasının diğer avantajları aşağıdaki gibidir:

- İletişim: Güncellenmiş bir Gantt şeması, bir bakışta projenin nasıl ilerlediğini gösterebilir, bu da aynı bilgileri paylaşmak için birden fazla toplantıya olan ihtiyacı ortadan kaldırabilir.

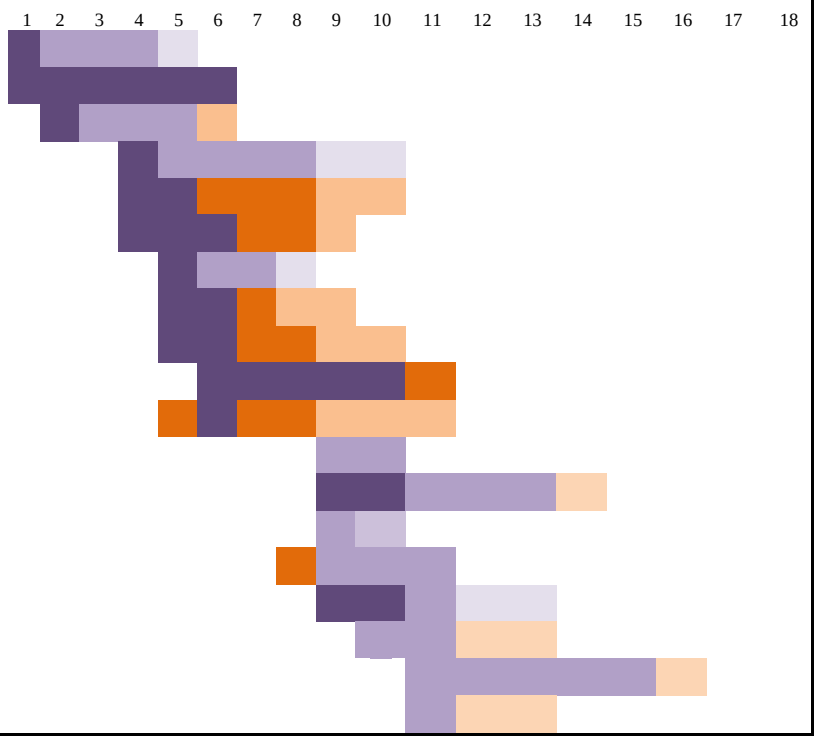
- Motivasyon: Bir projenin büyük resminin ve nerede durduğunu görülmesi proje ekibini tamamlanmaya zorlamak için motive edebilir. Takımın bir sonraki adımı anlamasını sağlar.

- Esneklik: Bir proje başladıktan sonra bir Gantt şemasında değişiklik yapmak kolaydır.

- Verimlilik: Büyük bir projeye giren tüm faaliyetlerin önceden bilinmesi zamanın daha verimli kullanılmasına izin verebilir, diğer ekip üyelerinin yanıtlarını veya geri bildirimlerini beklerken diğer faaliyetler başlatılabilir.

Gantt şemalarının proje yönetiminde kullanmanın en büyük dezavantajı, şemaların çok ayrıntılı ve dolayısıyla hantal olabilmesidir. Gantt şemasını yönetmek kendi başına bir proje haline gelebilir. Bir diğer büyük dezavantaj Gantt şemalarının öncelikleri göstermemesidir. Yeterli zaman varsa, gerçekleşmesinin daha iyi olacağı bir faaliyete karşı yüksek öncelikli bir başka faaliyeti göstermek zordur. Bir Gantt şemasında, bir bütün olarak projeyle ilişkili maliyetleri veya bireysel faaliyetleri belirtmek de zordur.

Etkinlik	Plan Başlangıcı	Plan Süresi	Fiili Başlangıç	Fiili Süre	Tamamlanma Yüzdesi
Etkinlik 01	1	5	1	4	25
Etkinlik 02	1	6	1	6	100
Etkinlik 03	2	4	2	5	35
Etkinlik 04	4	8	4	6	10
Etkinlik 05	4	2	4	8	85
Etkinlik 06	4	3	4	6	85
Etkinlik 07	5	4	5	3	50
Etkinlik 08	5	2	5	5	60
Etkinlik 09	5	2	5	6	75
Etkinlik 10	6	5	6	7	100
Etkinlik 11	6	1	5	8	60
Etkinlik 12	9	3	9	3	0
Etkinlik 13	9	6	9	7	50
Etkinlik 14	9	3	9	1	0
Etkinlik 15	9	4	8	5	1
Etkinlik 16	10	5	10	3	80
Etkinlik 17	11	2	11	5	0
Etkinlik 18	12	6	12	7	0
Etkinlik 19	12	1	12	5	0
Etkinlik 20	14	5	14	6	0



Gantt Şeması

1.1.2.5. Program Değerlendirme ve Gözden Geçirme Tekniği (PERT)

PERT, belirsizliğin fazla olduğu araştırma-geliştirme projelerinde kullanılmak üzere geliştirilmiş, fakat daha sonra farklı sektörlerde de yaygın olarak kullanılmıştır. Dunne ve Klementowski (1982:77), yapılan araştırmalara göre, araştırma-geliştirme projeleri yöneticilerinin, PERT'i en iyi proje planlama tekniği olarak nitelendirdikleri belirtmektedir.

Yönetim açısından PERT, planlamanın nasıl yapılması gerektiğini belirtmektedir. Yönetime, koşullar değiştiğinde planlamanın güncel kalmasını sağlayacak gerekli araçlar sunmaktadır. Plandan

sapmanın yaratacağı etkileri yönetimin önceden görmesini sağlayarak, olası problemler ortaya çıkmadan önce düzeltici önlemlerin alınmasına imkan tanımaktadır (Thierauf, 1978:173; Stevenson, 1996:784).

PERT analizinde projenin başlangıcından bitimine giden yollardan en yüksek toplam beklenen (ortalama) süreye sahip yol, kritik yoldur. Projenin beklenen tamamlanma süresi, kritik yolun toplam beklenen süresidir. Ancak gerçek yaşam problemlerinde bir projeyi oluşturan faaliyetlerin kesin sürelerini bilmek mümkün değildir. Faaliyet süreleri bir olasılık dağılıma sahip rassal değişkenlerdir. Eğer projenin faaliyetlerinin tamamlanma süresi kesin olarak bilinmiyorsa, projelerin verilen bir termin süresi içinde tamamlanıp, tamamlanamayacağının olasılık tahmini için PERT kullanılabilmektedir. Diğer bir eleştiri de, projenin başlangıcı ile sonu arasında bulunan faaliyetlerin süre ve varyanslannın art arda toplanarak bulunmasının, bu faaliyetlerin bağımsız olduğu ile ilgilidir. (Öcal, 1991:97; Sauls, 1978:30). Cottrell'in (1999:17), PERT'le ilgili olarak belirttiği diğer bir önemli sorun, her faaliyetin, en iyimser, en kötümser ve en olası süre tahminlerini doğru bir biçimde yapmanın oldukça zor olduğu ile ilgilidir, bu tahminler sübjektif yargıya dayanmaktadır. Karmaşıklığı nedeniyle PERT'in uygulanmasının güç ve maliyetli olması bir diğer olumsuzluktur.

Birtakım eksikliklerine rağmen, PERT, yine de yaygın olarak kullanılmaktadır. Proje yöneticilerine, kritik olmayan faaliyetlerden, projenin zamanında tamamlanmasını etkileyen kritik faaliyetlere, kaynakların nasıl aktarılabileceği konusunda yardımcı olmakta ve projedeki belirsizliklerle baş edebilmek için çoklu süre tahmini yapmayı sağlamaktadır.

PERT ve CPM tekniklerinin proje yönetimini daha basit hale getiren en önemli araçlar olduğu bilinen bir gerçektir. Bu araçların günümüzde çok kullanılır hale gelmesi bilişim sektöründeki hızlı gelişim ile birlikte; bu teknikleri uygulayan paket programların projelerde aktif olarak kullanılmasının bir sonucudur. Paket bilgisayar programlar 1970'li yıllarda özellikle büyük askeri projelerde kullanılmaya başlanmıştır. Ancak o yıllarda bilgisayarların çizim kapasitelerinin yeterli olmaması ve çizici araçların oldukça pahalı olması bu araçların projelerde kullanımını sınırlı kılmıştır. Fakat günümüzde bilgisayarların hem maliyetlerinin düşmesi, yaygınlaşması hem de kapasitelerinin artması ile birlikte birçok farklı endüstride proje yönetimi ile ilgili paket programlar kullanılır hale gelmiştir.

PERT ve CPM'de bu yaklaşımlara özgü çeşitli terimler kullanılır. Bunlar genelde günlük hayatta kullanılan anlamlarından daha fazlasını ifade eder. Bu nedenle, bu terimlerin bilinmesinde yarar bulunmaktadır. Bu terimler;

- Faaliyet: Projenin gerektirdiği görev veya görev grubudur. Faaliyetler için kaynak ve zaman kullanılmaktadır.

- Olay: Bir veya daha fazla faaliyetin tamamlanması sonucu ulaşılan tanımlanabilir durumdur. Olaylar için zaman veya kaynak kullanmaz. Bir olayın ortaya çıkabilmesi için bunun öncesinde tamamlanması gereken faaliyetler bitirilmelidir.

- Mihenk Noktası: (Kilometre Taşı) Projede dikkate değer gelişmeyi gösteren tanımlanabilir ve önemli olaydır.

- Ağ: Faaliyet ve olayları gösteren, birbirine oklarla bağlı düğüm/kutucukların bulunduğu şekildir. Oklar, faaliyetleri veya teknik olarak faaliyetlerin birbirine bağımlılığını göstermektedir. Ağlar, genellikle solda bir "*başlangıç*" kutucuğu ve sağda "*bitiş*" kutucuğu ile çizilmektedir. Okların yönü bağımlılığı göstermektedir. Okun yönü öncül faaliyetten ardıl faaliyete doğrudur.

- Yol: Ağ içindeki herhangi iki olayı birbirine bağlayan faaliyettir.

- Kritik Yol: Projenin başlangıcından sonuna kadar giden ve herhangi bir gecikme olduğunda tüm projenin gecikmesine yol açacak olan faaliyetler serisidir.

- Kritik Zaman: Kritik yol üzerindeki faaliyetlerin tamamlanması için gerekli olan süredir. Faaliyetlerin hangisinin öncül ve hangisinin ardıl olduğunun bilinmesi çok önemlidir. Öncüllük veya ardıllık ilişkisi faaliyetler arasındaki teknik bağıntıları da tanımlamaktadır.

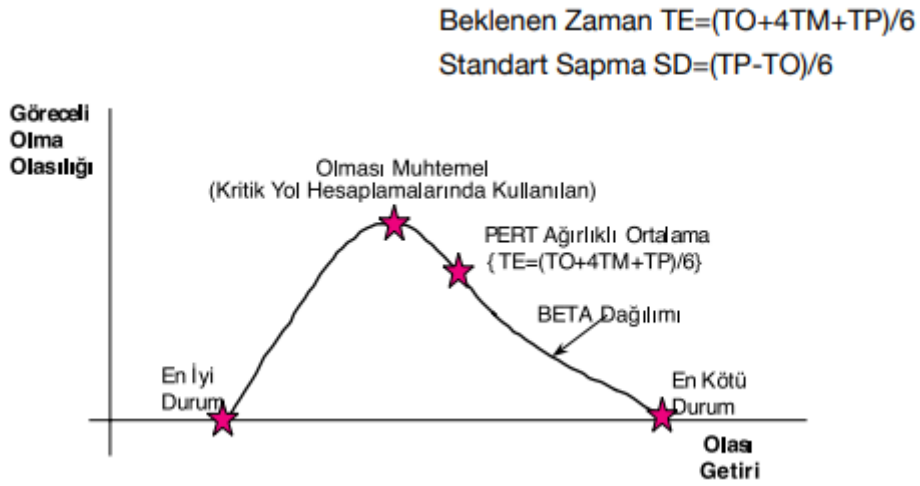
Bu bağıntıların üç temel tipi vardır:

- Zorunlu Bağıntılar: Bu bağıntıların değiştirilmesinin imkânı yoktur. Örneğin boya alınmadan boyama işlemine başlanamaması vb gibi.

- İsteğe Bağlı Bağıntılar: Bu bağıntılar proje yöneticisinin isteğine bağlı olan veya projeden projeye değişebilecek bağıntılardır. Örnek olarak evin boyanması öncesi evden çıkılıp başka bir yere geçilmesi isteğe bağlı bir bağıntıdır. Eğer istenirse boyama esnasında evde kalınabilir.

- Dış Bağıntılar: Bu bağıntılar proje yöneticisinin kontrolünde olmayanlardır. Kritik yol üzerinde boyacıların çalıştığını düşünelim. Proje yöneticisi boyacıların işinin kaç gün süreceğini (eğer bu konuda özel bir uzmanlığı yok ise) bilemez. Bunu belirleyecek olan boyacıların kendileri olacaktır.

Her aktivite süresi için üç farklı tahmin kullanan CPM türü bir tekniktir. Bir görevin tamamlanmasını belirlemek için her bir aktiviteyi tamamlamak için en az üç tahmin gösterilir. Bu tahminler matematiksel formüller kullanılarak tek bir sayıya indirgenir. Ardından klasik bir CPM algoritması uygulanır. PERT ağ yönetimine ilişkin diyagram aşağıda belirtilmiştir. Burada olaylar zaman içindeki noktalar veya faaliyetlerin başlatılması ve tamamlanması için kilometre taşlarıdır. Birincisi en iyi durum (iyimser/TO) senaryodur. Üçüncüsü en kötü durum (kötümser/TP) senaryosudur. İkincisi olması muhtemel (TM) senaryodur. Bu tahmin, büyüklük ve kapsamda benzer projelerden elde edilen deneyimlere dayanmaktadır. Verilen her aktivite için PERT zaman tahminini hesaplamak için aşağıdaki hesaplama uygulanır:



PERT Ağ Yönetimi Şeması

PERT kullanılarak kritik bir yol türetilir. Kritik yol ağdaki en uzun yoldur. Kritik yol, projeyi kısaltabildiği (hızlandırıldığı) veya uzatabildiği (geciktirildiği) yoldur.

Sağlanan örnekte PERT'in CPM'ye göre avantajı formülün üç tahmininin bir beta istatistik dağılımını izlediği ve buna bağlı olarak olasılıkların (ilişki güven seviyeleri ile) toplam proje süresi ile ilişkilendirilebileceği makul varsayımına dayanmaktadır.

Sistem geliştirme projeleri için bir PERT ağı tasarlarken ilk adım, projenin tüm faaliyetlerini ve ilgili olaylarını/kilometre taşlarını ve bunların göreceli sırasını tanımlamaktadır. Örneğin bir olay veya sonuç operasyonel fizibilite çalışmasının sonuçlanması veya kullanıcının ayrıntılı tasarımı kabul ettiği nokta olabilmektedir. Analist herhangi bir faaliyeti göz ardı etmemeye dikkat etmelidir. Buna ek olarak, program kodlama başlamadan önce analiz ve tasarım gibi bazı faaliyetlerin öncesinde diğerleri olmalıdır. Faaliyetlerin listesi PERT ağının detayını belirlemektedir. Analist giderek daha ayrıntılı zaman tahminleri sağlayan birçok diyagram hazırlayabilir.

PERT ile ilgili dikkat edilmesi gereken hususular:

- PERT tahmininin kesinliği kullanılan tahminlerin güvenilirliğine bağlı olarak değişkenlik göstermektedir.

- Yapılacak güncelleme ve düzenleme zaman ve efor gerektirmektedir.

- CPM gibi tüm kaynakların proje için uygun olduğu varsayılarak ilerlenmektedir.

- Kritik yol değişiklikleri tam olarak görülemeyebilir.

1.1.2.6. Kritik Yol Metodu (CPM)

Kritik Yol Metodu (CPM) 1950'li yılların sonlarında DuPont'tan Morgan R. Walker ve Remington Rand'dan James E. Kelley Jr. tarafından geliştirilen bir proje modelleme tekniğidir. Proje faaliyetlerinin planlanması için kullanılan yöntemlerden biri olan kritik yol metodu, bir projenin tamamlanması için gerekli faaliyetleri tek bir planda toplayarak, faaliyetlerin öncelik ilişkilerini ve sürelerini gösterir ve aynı zamanda projenin kontrolüne yardımcı olur. Kritik yol ise projenin başlangıcı ile bitişi arasındaki en uzun mesafedir. Kritik yol üzerinde bulunan faaliyetlerin sürelerindeki değişim, proje süresinin değişmesine sebep olur. Basit ve hesaplaması kolay olduğu için dolayı birçok sektörde yapılan projelerde kullanılmaktadır.

Genellikle manipüle edilmemiş proje zaman çizelgelerinde en az bir kritik yol vardır. Projenin mümkün olan en kısa sürede tamamlanması için kritik yollar önemlidir. Kritik yollara dahil edilmeyen faaliyetlerin, sarkma süresi oluşturma riski bulunmaktadır. Sarkma süresi, tüm projenin tamamlanmasını geciktirmeyecek her faaliyetin mümkün olan en son tamamlanma süresi ile öncü tüm faaliyetlere dayanarak mümkün olan en erken tamamlanma süresi arasındaki farktır. Kritik bir yoldaki faaliyetlerin sarkma süresi sıfırdır. Ağ üzerinde ileri- geri çalışılarak, faaliyetler ve proje için mümkün olan en hızlı tamamlanma süresi belirlenir.

Proje yöneticisi dikkatini özellikle kritik yol üzerindeki faaliyetlere vermelidir. Proje yöneticisi projenin başlangıcında hangi faaliyetlerin kritik yol üzerinde olduğunu hangilerinin olmadığını dikkatle kaydetmelidir. Bu durumda iş gücü vb. kaynaklar kritik yol üzerinde olmayan bir faaliyetten kritik yol üzerindeki faaliyetlere aktarılarak kritik süre kısaltılabilir. Bu kararların verilebilmesi için her faaliyetin sarkma süresinin de bilinmesi gereklidir. Projede sarkma süresi olması proje yöneticisinin işini oldukça kolaylaştıran bir durumdur.

PERT yöntemi ile CPR yönteminin farklılıkları aşağıdaki gibidir:

- CPR aktivite odaklı, PERT olay odaklıdır.

- CPR'da süreler bellidir, PERT'te kesin değildir.

- CPR'da aktiviteler düğüm üzerinde PERT'te ok üzerindedir. PERT'te kilometre taşları düğüm şeklindedir.

- PERT sadece bitiş-başlangıç ilişkilerini (başlangıç-başlangıç, bitiş-bitiş vb.) içerir.

- Düğümler CPR'da dikdörtgen, PERT'te daire şeklindedir.

Kritik yolu kısaltacak önlemler aşağıda belirtilmiştir:

- Kritik etkinlikler üzerine yoğunlaşılması,

- Seri işlerin paralel olarak yeniden planlanması (hızlı izleme),

- Kritik etkinliklerin sürelerinin kısaltılması,

- Fazla mesailerle çalışma sürelerinin uzatılması,

- Erken başlayan işlerin kısaltılması

- En kolay işlerin kısaltılması,

- Çok kaynak kullanan etkinliklerin kısaltılması,

- Maliyeti yüksek işlerin kısaltılması,

- İşletmenin kontrolü altındaki işlerin kısaltılması.

1.1.2.7. Zaman Kutulama Yöntemi

Zaman kutulama yöntemi, yazılım teslimatlarını nispeten kısa ve sabit bir süre içinde ve önceden belirlenmiş spesifik kaynaklarla tanımlamak ve dağıtmak için bir proje yönetim tekniğidir. Zaman kutulama tekniğinin temel amacı projenin bilinen bir zaman dilimde sistemin her parçasını zaman kutularına yerleştirerek ele almak ve bu yönüyle zamana, sonuca ya da ürüne odaklı bir sistem geliştirme

yaklaşımı sağlamaktadır. Zaman kutusu yönetimi, temel özelliklerin kısa sürede sunulması gereken prototipleme veya hızlı uygulama geliştirme türü yaklaşımları gerçekleştirmek için kullanılabilmektedir. Temel özellikler gelecekteki entegrasyonlar için arayüzler içermektedir. Bu yaklaşımın en büyük avantajı, proje maliyet aşımlarının ve programlı teslimattan kaynaklanan gecikmelerin önlenmesidir. Proje, kalite sürecine olan ihtiyacı ortadan kaldırmaz. Yeni geliştirme araç ve tekniklerinin kullanılması nedeniyle tasarım ve geliştirme aşaması kısaltılmıştır. Test senaryolarının hazırlanması ve test gereklilikleri, son kullanıcı katılımının bir sonucu olarak kolayca yazılmaktadır. Sistem testi ve kullanıcı kabul testi (User Acceptance Test-UAT) normalde birlikte yapılmaktadır.

1.2. Proje İşletimi

Planlama çalışmaları tamamlandıktan sonra, program yöneticisi proje yönetim ofisi ile koordineli yürütülen planlarda, süreçlerde ve prosedürlerde belirtildiği gibi planlanan görevlerin fiili yürütülmesini sağlar.

1.2.1. Proje Açılışı

Proje, sponsor ya da proje yöneticisi tarafından başlatılır. Proje onayının alınması için gerekli bilgiler sponsor ya da proje yöneticisi tarafından toplanmaktadır. Genellikle referans veya projenin amacını, üretilecek sistemdeki paydaşları, proje yöneticisini ve sponsorunu belirten bir proje tüzüğü olarak derlenmektedir.

Proje başlatma belgesinin veya proje talep belgesinin onaylanması, projenin başlaması için yetkilendirmeyi belirtmektedir.

a. Proje Hedeflerinin Belirlenmesi

Proje hedefleri, kaynak kullanımını iyileştirerek stratejik işletme hedeflerine uygun çıktılar oluşturularak hedefe ulaşılmasının sağlanması ve bütçe, maliyet, zaman optimizasyonunu sağlayacak spesifik eylem faaliyetleridir. Tüm proje amaçlarının bu amaca ulaşmak için gereken bir veya daha fazla hedefi olacaktır. Proje hedefi, proje amacına ulaşma yoludur.

Projenin üç kısıtını içeren bir model olan proje üçgeni başarılı bir proje yönetiminin vazgeçilmez unsurudur. Demir üçgen, sihirli üçgen ve proje saç ayağı olarak da bilinen model; zaman, maliyet ve kapsam sabitlerini kalite çemberi içerisinde yönetmektedir.

Kapsam, projenin hangi hedefe ulaşmayı amaçladığı sorusunu cevaplar. Sponsor veya müşteri proje sonucunda hangi ürün veya hizmeti elde etmek amacında olduğu belirlenmektedir. Zaman, projenin tamamlanmasının ne kadar zaman alacağı sorusunu cevaplamaktadır. Diğer bir deyişle ürün veya hizmetin gerçekleştirilmesine ilişkin takvimi ortaya koymaktadır. Başlangıçta kararlaştırılan takvime uyulmadığı takdirde müşteri veya sponsora tazminat ödenmek zorunda kalınabilir. Bu da maliyetlerin artmasına neden olacaktır. Proje yöneticisinin görevi her aşamada takvim ile gerçekleşen işin uyumunun takip edilmesidir. Maliyet, projeyi tamamlamanın maliyetinin ne olacağı sorusunu cevaplamaktadır. Yani eldeki kaynakların çalışır bir sisteme dönüştürülmesinin maliyetini ifade etmektedir. Burada maliyet kelimesi, analist veya programcıların ücretleri, özel ekipmanların satın alınma, kira giderleri, yazılım donanım ve bilgisayar ağları için yapılan harcamalar, yönetim ve kırtasiye giderleri gibi projenin yürütülmesi ile doğrudan ilgili konu başlıklarını kapsamaktadır.

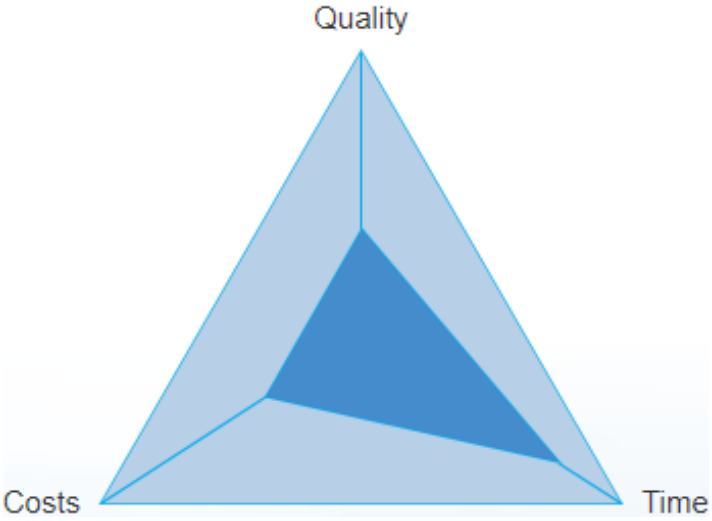
Kapsam, zaman ve maliyet kısıtlarının her üçünü kalite çemberi içerisinde yönetilebilmesi iyi bir proje yönetimi için doğru yolda olunduğunu gösterir. Bu üçgende herhangi bir sabit değiştiği zaman diğer iki sabitinde değişmesi gerekir. Bir projeyi üç ayaklı (kapsam, zaman, maliyet) bir tabureye benzetmek mümkündür. Ayaklardan bir tanesinin olmaması taburenin devrilmesine yol açacaktır. Örneğin; bir projenin kapsamı genişletilirse bu ölçüde zaman ve maliyetin de artması gerekir.

Projelerde öncelikler çoğu durumda birbirine bağımlıdır. Bir önceliği değiştirmek diğerlerini etkilemektedir. Örneğin;

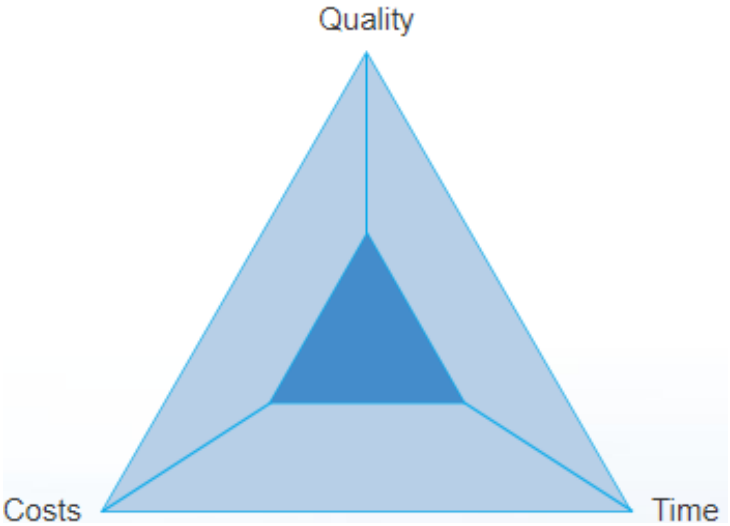
- Proje daha önce bitirilmeli: Etkiler daha fazla maliyet veya daha küçük kapsam/daha az kalite,

- Projenin kapsamı daha büyük olmalı: Projenin daha fazla zamana veya daha fazla maliyete ihtiyacı var,

- Projenin maliyeti daha düşük olmalı: Projenin daha fazla zamana ihtiyacı var veya kapsam daha küçük/daha az kaliteye sahip.



Orta kalite (endüstri standartı), zaman öncelikli ve maliyetin önemsiz olduğu senaryo

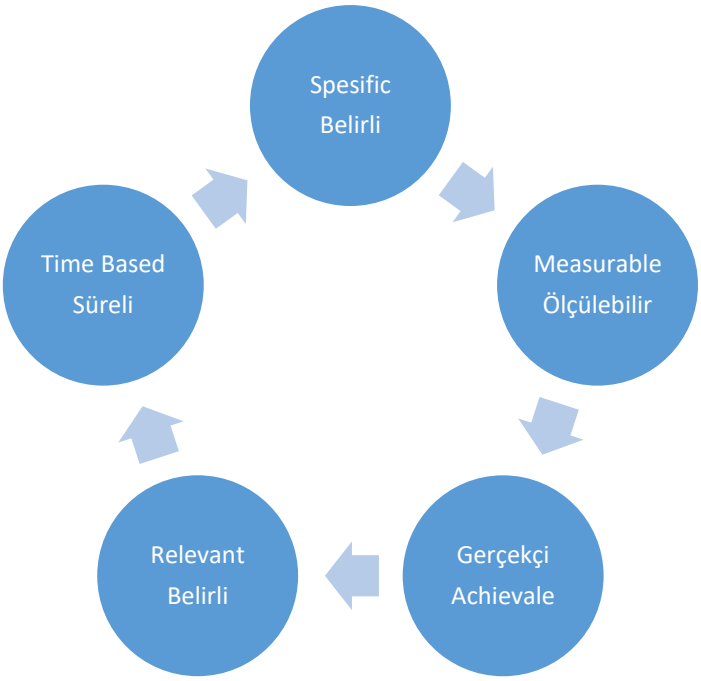


Zaman, kalite ve maliyetin orta seviyede tutulduğu senaryo

Maliyetin öncelikli olduğu senaryo

Zaman + Maliyet + Kapsam = Kalite

Bir projenin belirli (spesific), ölçülebilir (measurable), başarılabilir/gerçekçi (attainable), belirli, ilgili/uygun (relevant) ve süresi (time based) net olarak tanımlanmış sonuçları olmalıdır. Bu beş sonuç SMART metodunu oluşturur. SMART uygulandığında, ölçülebilir ve doğrulanabilir hedefleri göstermektedir.



SMART Şeması

- Belirli (spesific), ulaşılmak istenen hedefin ortaya kesin bir şekilde konulması gereklidir. Genel ifadelerin yerini, özel ifadeler almalıdır. Bunun yanı sıra, neden önemli olduğu ve hangi kaynaklara ihtiyaç duyulduğu belirlenmelidir. Örneğin, bir firmanın kar marjını arttırmayı hedeflemesi genel bir düşüncedir. Hedefi belirli bir hale getirmek için rakamsal ifadelerle nicelik kazandırmak gerekecektir.

- Ölçülebilir (measurable), ulaşılmak istenen hedefe ulaşılıp ulaşılamadığını ya da ulaşılması için ne kadar zaman veya emek harcanacağını her an bilmek için koyulan hedef ölçülebilir olmalıdır. Örneğin, kişinin işini büyütmek gibi genel bir hedefi olmamalıdır. Bir yıl içinde her 3 ay içinde ölçülebilir bir büyüme oranının belirlenmesi, işinin ileri bir boyuta taşınması açısından yararlı olacaktır.

- Başarılabilir/Gerçekçi (achievale), SMART analizi yapılırken hedefte olması gereken diğer bir özelliktir. Mevcut imkanlarla ulaşılabilecek gerçekçi bir hedef olmalıdır. Örneğin; bir işletmenin 5 aylık cirosunu 1 ayda elde etmesi zordur. Çalışan kanaatlerini de göz önünde bulundurarak firmanın başarabileceği hedefler ortaya konmalıdır. Ütopik düşüncelerden kaçınılmalıdır.

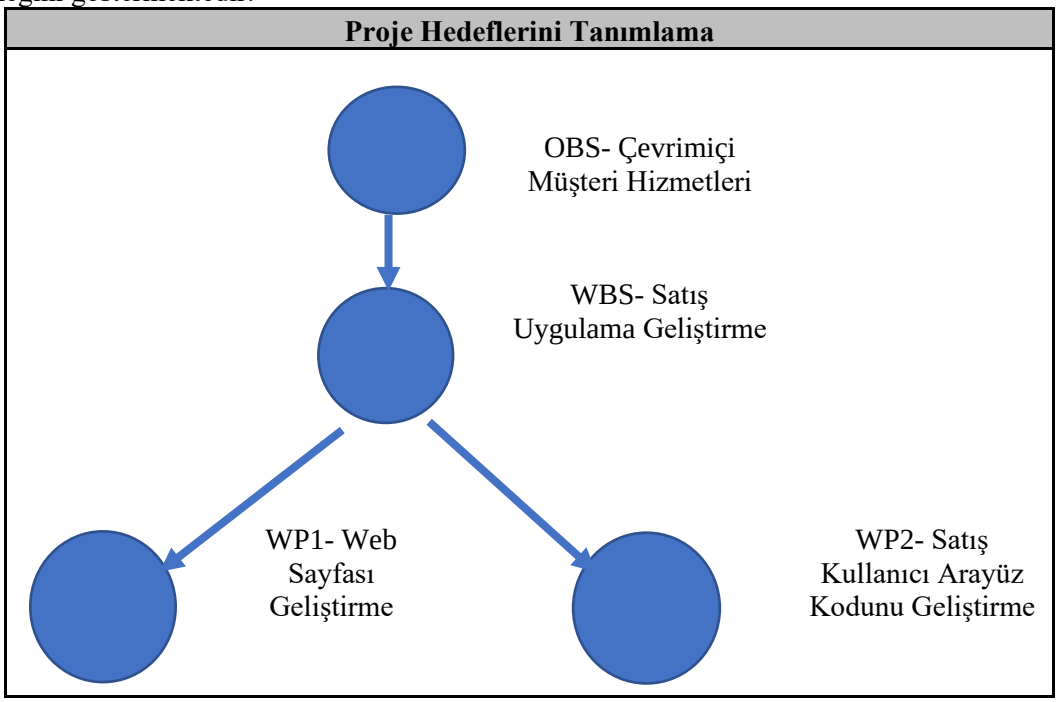
- İlgili/Uygun, hedefleri belirlerken ana amaca uygun hedefler belirlenmelidir. Örneğin, bir işletme ticaretini Çin ile yapıyorken ve başka bir sahaya girmeyi düşünmüyorken, personellerine İspanyolca öğretmeye çalışması hedefleriyle alakasız olmaktadır.

- Süreli (time based), hedefler belirlenirken belli bir zaman sınırı koyulmalıdır. Eğer zaman sınır koyulmazsa hedefe ulaşmak için doğru bir planlama yapılamamaktadır. Örneğin, bir işletmenin satış hedefinin 1 yıla yayılarak genel olarak incelenmesi basit bir yönetim şekli olacaktır. Gelir-gider dengesinin sağlanması amacıyla rakamların aylık takip edilmesi gerekmektedir.

Proje hedeflerinin tanımlanması için yaygın kabul gören bir yaklaşım, nesne kırılım yapısı ile başlamaktır. Çözümün bireysel bileşenlerini ve birbirleriyle olan ilişkilerini hiyerarşik bir şekilde grafiksel olarak veya bir tabloda göstermektedir. Bir nesne kırılım yapısı, özellikle organizasyonel gelişim gibi somut olmayan proje sonuçları ile uğraşırken teslim edilebilir bir materyalin göz ardı edilmemesini sağlamaya yardımcı olabilmektedir.

Bir nesne kırılım yapısı derlendikten veya bir çözüm tanımlandıktan sonra, proje sırasında nesne kırılım yapısı öğelerini oluşturması için gerekli olan tüm görevleri yapılandırılması için bir iş kırılım yapısı tasarlanmaktadır. Bir iş kırılım yapısı, projeyi yönetilebilir iş birimleri açısından temsil edilmekte, projede merkezi bir iletişim aracı olarak hizmet etmekte ve maliyet ve kaynak planlamasının taban çizgisini oluşturmaktadır.

Bir nesne kırılım yapısının aksine iş kırılım yapısı oluşturulacak çözümün temel öğelerini içermez. Bunun yerine münferit çalışma paketlerini göstermektedir. Bir iş kırılım yapısı süreç odaklı ve aşamalıdır. İş kırılım yapısının ayrıntı düzeyi, proje sponsoru, proje yöneticisi ve proje ekibi üyeleri arasında ayrıntılı hedeflerin müzakere edilmesi için temel oluşturmaktadır. Aşağıdaki şekil bu sürecin bir örneğini göstermektedir:



Proje Hedefleri Tanımlama Şeması

İş kırılım yapısı, gerekli tüm görevleri listelemekte ve bunları yönetilebilir ve kontrol edilebilir birimler halinde gruplandırmaktadır. Her bir münferit çalışma paketinin ayrı bir sahibi ve ana hedeflerin bir listesi ile ek hedefleri olmalı ve kapsam dışı hedeflerin bir listesi olmalıdır. Münferit çalışma paketlerinin spesifikasyonları diğer çalışma paketlerine bağımlılıkları, performans ve amaç başarısının nasıl değerlendirileceğinin bir tanımını içermelidir. Bir iş kırılım yapısı örneği aşağıdaki şekilde gösterilmektedir.

İş kırılım yapısı ve ilgili çalışma paketleri ile hatırlanması gereken temel konular:

- En üst iş kırılım seviyesi nihai teslimatı veya projeyi temsil etmektedir.

- Alt teslimatlar bir işletmenin bölümüne veya birimine atanan çalışma paketini içermektedir.

- Bir iş kırılımının tüm öğelerinin aynı seviyede tanımlanması gerekmemektedir.

- Münferit çalışma paketleri alt teslimatları üretmek için gereken görevlerin işini, süresini ve maliyetlerini tanımlamaktadır.

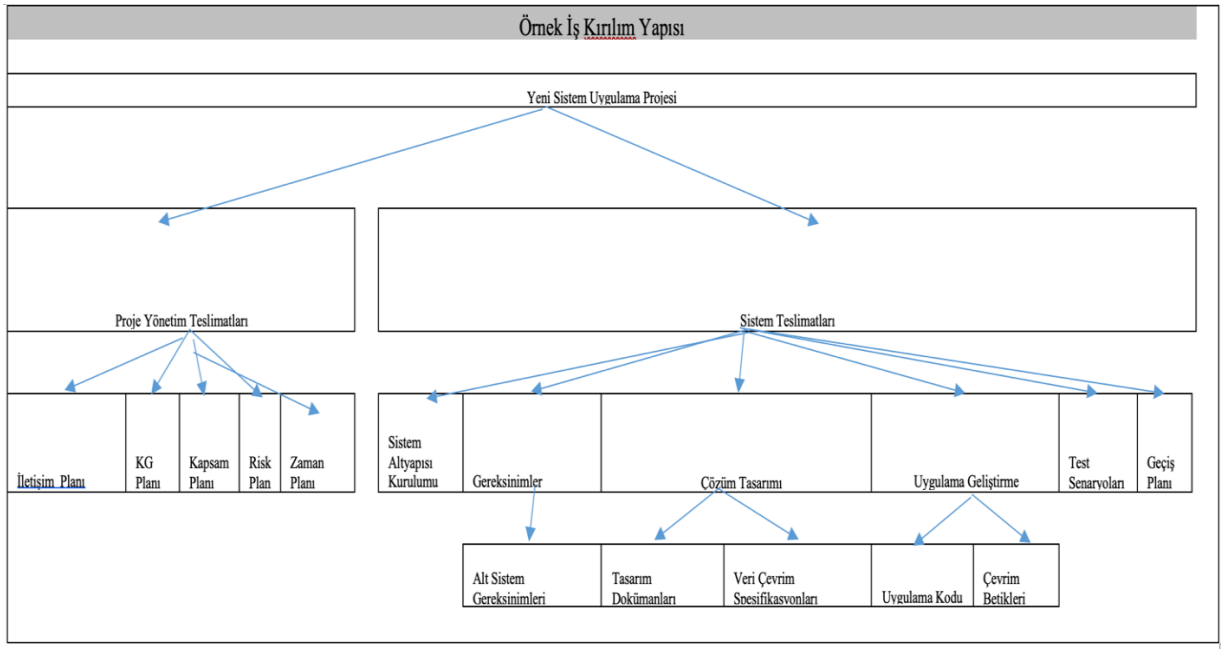
- Münferit çalışma paketleri ortalama 10 günü aşmamalıdır.

- Münferit çalışma paketlerinin iş kırılımında birbirinden bağımsız olması gerekmektedir.

- Münferit çalışma paketleri benzersizdir, iş kırılımı genelinde çoğaltılmamalıdır.

İletişimi desteklemek için genelde görev listeleri kullanılmaktadır. Görev listesi münferit çalışma paketleri ile ilgili olarak gerçekleştirilecek eylemlerin listesidir. Atanmış sorumlulukları ve son tarihleri içermektedir. Görev listesi, proje ekibi üyelerine operasyonel planlama ve anlaşmalarda

yardımcı olmaktadır. Bu görev listeleri genellikle bir projenin planlama aşamasında bir proje zaman çizelgesinde derlenmektedir. Projenin kontrol aşamasında, çalışma paketlerinin ilerlemesi ve tamamlanmasının takibi için kullanılmaktadır. Proje zaman çizelgeleri yaşayan belgelerdir ve bir çalışma paketinin görevleri, başlangıç ve bitiş tarihleri, tamamlanma yüzdesi, görev bağımlılıkları ve bu görevler üzerinde çalışması planlanan kişilerin kaynak adları belirtilmektedir.



İş Kırılım Şeması

b. İş Olurluk Analizi

İş olurluk incelemesi, bir işletmenin söz konusu projenin devam edip etmeyeceğine karar vermesi için gerekli bilgileri sağlamaktadır. İşletme ve yatırım büyüklüğüne bağlı olarak iş olurluk incelemesinin geliştirilmesi projenin ilk adımı ya da öncüsüdür. İş olurluk incelemesi, projenin başlatılma ve işletilme gerekçeleri ile faydalarını açıklayacak ayrıntıyı kapsamalıdır. İş olurluk analizi projenin her safhasında önemli bir karar ölçütüdür. Proje yaşam döngüsü içerisinde değişen koşullar karşısında iş olurluk analizi geçerliliğini yitirdiğinde proje sponsoru veya BT yönlendirme komitesi tarafından projenin devam edip etmeyeceği değerlendirilmelidir. Eğer iş olurluk analizinin değişmesi durumunda proje bölüm planlama ve onay süreci işletilmelidir.

Bilgi sistemleri denetçisi, işletmenin fizibilite çalışmaları sırasında kullanılan iş olurluk incelemelerini nasıl tanımladığını ve bilgi sistemleri geliştirmesiyle ilgili projeler için yatırım getirisi ile ilgili sonuç tespitlerini anlamalıdır. İşletme yatırım getirisi hedeflerini tutarlı bir şekilde karşılayamazsa, bu durum sistem geliştirme yaklaşımında ve ilgili proje yönetimi pratiklerinde zayıflıklara işaret edebilir.

c. Projenin Yürütülmesi

Projenin yürütülmesi kapsamında dokümantasyon, toplantılar, gözetim faaliyetleri ele alınmaktadır. Proje büyüklüğüne, karmaşıklığına ve etkilenen taraflara göre proje yöneticisi/sponsoru tarafından; proje ekibi ve proje yöneticisi arasında gerçekleştirilen birebir toplantılar, proje ekibinin iş birliği ve katılımıyla gerçekleştirilen çalıştaylar, proje yöneticisinin proje ekibine aktarımlarını ve proje iletişiminin yürütülmesini sağlayan başlangıç toplantıları yollarından en az birini seçerek başlatılabilir.

1.2.2. Projenin Kontrolü ve İzlenmesi

Proje çalışmalarının izlenmesi, proje yönetimi planında tanımlanan performans hedeflerine ulaşılması için projenin ilerlemesinin izlenmesi, gözden geçirme ve düzenleme sürecidir. Kontrol ise düzeltici ya da önleyici eylemleri belirlemeyi ve gerçekleştirilen eylemlerin performans sorununun

çözülüp çözülmediğinin belirlenmesi amacıyla eylem planlarının izlenmesini ya da yeniden yapılanmasını içermektedir.

Planlama aşamasında yükleniciler ve tedarikçilere ilişkin ölçütler izlenmektedir. Projenin BT sistemi gereksinimleri, mimarisi, tasarımı, geliştirilmesi, test edilmesi, uygulanması ve üretim operasyonlarına geçişi konusunda entegre ekibin gözetimi önemli bir başarı faktörüdür.

1.2.2.1. Projelerde Kontrol ve İzleme Kavramı

Bir projenin kontrol ve izleme faaliyetleri kapsam, kaynak, kullanım ve risk yönetimini içermektedir.

Proje kontrolü, projedeki faaliyetlerin durumun değerlendirilmesi, proje durumunun planlanan durumla karşılaştırılması ve eğer gerekiyorsa düzeltici önlemlerin alınması için yapılan faaliyetlerdir. Proje kontrolü sayesinde projenin yürütülmesi sırasında sorun yaratabilecek, kritik veya yan kritik faaliyetler üzerinde yoğunlaşmak mümkündür (Monks, 1996:354).

Gerçekleşen ilerlemeyle ilgili sürekli bilgi akışının olması proje yöneticisinin belirsizliklerle baş edebilmesini sağlayacak bir geri-besleme mekanizmasıdır. Gerçekleşen ilerlemeyi planla karşılaştırmakla sapmalar belirlenmekte, problemler önceden tespit edilmekte ve düzeltici hareketler yapılabilmektedir. Çizelge ve teknik alanlarda beklenenden az bir başarı ve maliyetlerdeki sapmaların erkenden tespit edilmesi, yöneticilere önemli konular üzerinde odaklanma imkânı vermektedir. Projenin kontrolü ve güncellenmesi neticesinde alınan sağlıklı kararlar ile projenin başarısındaki sapmalar minimize edilebilmektedir.

Proje kontrolü, etkili yönetimi ve kararı destekleyen formatlarda bilgi iletilerek bir proje veya programın zaman ve maliyet sonuçlarını tahmin edilmesi ve yapıcı bir şekilde etkilemesi için kullanılan veri toplama yönetimi ve analiz süreçlerinden oluşmaktadır.

İyi bir proje yöneticisi, proje kontrolünün ne olduğunu tam olarak kavrayabilmek için üç unsur üzerinde durmalıdır. Bu unsurlar önleme, tespit etme ve eylem olarak sıralanabilir.

- Önleme: Önleme, projenin istenildiği biçimde ilerleyebilmesini sağlamak için proje planından olası sapmaların ortaya çıkmasının engellenmesidir. Bunun gerçekleştirilebilmesi için proje yöneticisinin tüm bilgi birikimini kullanması ve gerektiğinde diğer paydaşlar ile bilgi paylaşımına gitmesi gerekir. Ek olarak iyi bir proje yöneticisinin planlama aşamasında iyi bir yatırım yapması, etkin bir iletişim gücüne sahip olması, risk faktörlerini sürekli olarak izliyor olması, sorunların çözümünde mümkün olduğunca saldırgan bir tavır takınmaması ve yapılacak işleri mümkün olduğunca açık bir dille anlatması gereklidir.

- Tespit Etme: Tespit etme, proje kontrolünün bu boyutunu bir erken uyarı sistemi olarak görmek mümkündür. Proje kontrol sistemi olası sapmalar için bir erken tespit sistemi olmalıdır. Herhangi bir plan sapmasına ne kadar çabuk müdahale edilirse problemin ortadan kaldırılması ve tekrar temel plana dönülmesi de o denli kolay olacaktır. Erken tespit için en önemli faktörler iyi bir izleme sistemi kurulması ve proje sonuçlarının zamanında ölçülmesine olanak verecek iş parçacıklarının en iyi biçimde geliştirilmesidir. Kullanılabilecek tespit sistemlerine performans raporları ve değerlendirme toplantıları örnek olarak gösterilebilmektedir. Burada üzerinde durulması gereken bir diğer nokta da bir sapmayı tespit edebilmek için ilgili faktörün önceden belirlenmiş temel bir değerinin olması gerektiğidir. Sapmalar herhangi bir başarı faktöründe ortaya çıkabilir, paydaşların beklentileri ve kalite söz konusu bu başarı faktörlerinden bazılarıdır.

- Eylem: Eylem, genelde önleme işleminin gerçekleştirilebilmesi için de birçok eylemin hayata geçirilmesi gerekmektedir. Ancak eylemi tespit aşaması ile omuz omuza ilerleyen bir unsur olarak görmek gerekir.

Proje kontrolünün mümkün olduğunca etkin olabilmesi için herhangi bir sapmanın tespitinin çözüme yönelik uygun ve zamanında bir tepkiyi doğurması gerekir. Bu noktada proje yöneticisinin uygulayabileceği üç farklı eylem tipi öne çıkmaktadır. Bu eylem tipleri; düzeltici eylemler, değişim kontrol prosedürleri ve kazanılmış dersler olarak sıralanabilir. Bir proje zaman, maliyet ve kalite hedeflerinden sapmaya başladığında ve proje yöneticisinin çeşitli eylemleri uygulaması kaçınılmaz hale

geldiğinde "*Yöneticinin eylem seçenekleri nelerdir?*" sorusu cevaplanmalıdır. Temel bazı değişiklik ve eylem seçenekleri aşağıdaki gibi sıralanabilir:

- Projenin yeniden tahmin edilmesi: Daha önceden tanımlanan tüm tahmin değerleri ikinci kez kontrol edilir. Projenin paydaşlarının maliyet ve zaman kısıtları konusundaki isteklerini karşılamak ve projenin temel amacına ulaşmak için kullanılabilecek alternatif yollar araştırılarak projeye dâhil edilebilir.

- Projeye yeni bireyler eklenmesi: İlk bakışta projeye yeni bireyler eklenmesi hızlı ve kolay bir çözüm gibi görünmekle birlikte, yeni bireyler kendi fikir ve görüşlerini projeye aktarmak istediğinde fikir çatışmaları ortaya çıkabilecektir. İşin gerçekten daha hızlı bir biçimde tamamlanmasına destek olacak bireylerin projeye eklenmesi daha faydalı bir yaklaşımdır.

- Görevlerin kapsamının daraltılması: Bazı durumlarda projenin başarı ile tamamlanması için plan içinde yer alan bazı iş parçacıkları kaldırılabilir. Böylece tamamlanması nerede ise imkânsız düzeye ulaşan iş parçacıkları yerine daha altından kalkılabilir iş parçacıklarına ulaşılması sağlanır. Ancak hangi iş parçacıklarından vazgeçileceği belirlenirken proje hedeflerinden bir sapmanın ortaya çıkmaması gerektiği de göz önünde bulundurulmalıdır.

- Üretkenliğin verimli personel ile arttırılması: Kimi bireyler diğerlerine göre daha üretken ve verimlidir. Yakın çevrede proje ekibinde yer almayan daha üretken elemanlar bulunduğunda, bu elemanların kısa süreli olarak projede yer almasının sağlanması yoluna gidilmelidir. Ayrıca proje ekibi içinde yer alan üretken üyeler de kısa süreli olarak projenin başka aşamalarında çalıştırılabilir.

- Dış kaynak kullanımı: Yoğun çalışmalar sonucu hazırlanan özgün plana göre yapılması gereken bir iş parçacığını veya proje fazını proje ekibinden daha hızlı, daha verimli ve kalite standartlarına daha uygun yapabileceğine inanılan bir dış kaynak bulunduğunda, o iş veya fazın dış kaynağa verilmesinde bir sakınca yoktur.

- Fazla mesai kullanımı: Proje yöneticisi ölçülü ve ihtiyatlı davranmayı ihmal etmeden fazla mesai seçeneğine başvurabilir. Ancak fazla mesai uygulaması proje ekibinin beklendiğinden daha önce yorulmasına sebep olacağından proje ekibinin verimliliğinde bir düşmeye yol açabilir. Ölçüyü aşmaksızın gerektiğinde kullanıldığında projeye katkı yapacak bir eylem biçimidir.

- Bazı işlerin müşteriye aktarılması: Proje bir müşteri için gerçekleştirildiğinde ve proje yüksek maliyetli ama insan kaynakları bakımından kısıtlı olduğunda, bazı işler müşterinin de kabul etmesi hâlinde müşterinin kendisine bırakılabilir.

- Proje amaçlarının yeniden düzenlenmesi: En tehlikeli eylem türüdür. Projenin beklenen sonuçları sağlaması için kapsam daraltması ilk bakışta iyi bir yaklaşım gibi gözükmekle birlikte projenin kalitesinden ödün verme yönünde atılacak bir adım çok kötü sonuçlar doğurabilir. Kalite standartlarını yakalamadan tamamlanan bir proje, projeyi yürüten ekip veya işletme için kötü bir reklam haline gelebilir.

- Proje ekibinin uyumlu çalışması: Tüm işletmelerin süreçlerinde olduğu gibi izleme ve kontrol süreçlerinde de en önemli unsur proje ekibinin uyum içinde çalışmasıdır. Proje yöneticisinin, gerekli durumlarda proje ekip elemanlarına, kendilerinin arkasında olduğu güvenini vermesi elemanların işe daha iyi sarılmalarını sağlayabilir.

Proje kontrolü, proje yönetiminin birçok unsuruna uygulanabilir. Ancak, aşağıdaki unsurlar proje kontrolünün ana alanlarını temsil etmektedir:

- Planlama

- Risk yönetimi

- Maliyet yönetimi

Proje Kontrolünün Avantajları: Proje kontrolünün aşağıdaki faydaları, onu proje yönetiminin vazgeçilmez bir parçası haline getirir:

- Bir projenin toplam maliyetleri, örneğin temel performans göstergelerine (KPI) dayalı etkili karar verme yoluyla düşük tutulabilir.

- Maliyetler ve son tarihler için daha fazla öngörülebilirlik sağlar.

- Bir projenin tüm aşamalarında finansal durumu hakkında daha fazla ve daha iyi içgörüler sağlar.

- Sonuçların ertelenmesini daha kolay önlemeyi sağlar.

- Daha yüksek marjlar sağlar.

- Proje yönetimi ve sonuç garantileri söz konusu olduğunda daha iyi bir itibar sağlar.

- Artan bir çalışan memnuniyeti duygusu sağlar.

- Etkili proje yönetimi, daha az olgun proje yönetimi yeteneklerine sahip proje ekiplerine göre rekabet avantajı sağlar.

Bilgi sistemleri denetçisi, aşağıdaki proje yönetimi faaliyetlerinin yeterliliğini gözden geçirmelidir:

- Proje komiteleri/kurulu gözetim seviyesi,

- Risk yönetim yöntemleri,

- Sorun yönetimi,

- Maliyet yönetimi,

- Planlama ve bağımlılık yönetimi süreçleri,

- Raporlama süreçleri,

- Değişim kontrol süreçleri,

- Paydaş yönetiminin katılımı,

- Onay süreci.

1.2.2.2. Kapsam Değişikliklerinin Yönetimi

Projelerin kapsamını yönetmek, bir iş kırılım yapısı şeklinde dikkatli bir belgeleme gerektirir. Bu belgeler proje planının veya proje taban çizgisinin bir parçasını oluşturur. Kapsamdaki değişiklikler nerdeyse sürekli gerekli faaliyetlerde değişiklikleri ve son teslim tarihlerini ve bütçeyi etkiler. Bu nedenle, resmi değişiklik talepleri de dahil olmak üzere değişiklik yönetim sürecinin bir parçası olmalı ve dokümante edilerek saklanmalıdır. Değişiklikleri sadece proje paydaşları önerebilmektedir. Proje yöneticisi her değişiklik talebini kapsam, bütçe ve program açısından değerlendirmektedir. Ardından değişiklik danışma kurulu da değişiklik talebini değerlendirmektedir. Değişiklik kabul edilirse proje planı güncellenir. Güncellenen proje planı proje sponsoru tarafından resmi olarak onaylanmalıdır.

1.2.2.3. Zaman Yönetimi

Zaman, bir projedeki temel kıstaslardan biridir. Zaman yönetimi; zaman çizelgesi yönetiminin planlanması, aktivitelerin tanımlanması, aktivitelerin sıralanması, kaynaklarının belirlenmesi, aktivite sürelerinin tahmin edilmesi, zaman çizelgesinin geliştirilmesi ve zaman çizelgesinin kontrolü süreçlerinden oluşur.

1. Aktivitelerin Tanımlanması: Bu süreçle Aktivite Listesi, Kilometre Taşı Listesi oluşturulur.

2. Aktivitelerin Sıralanması: Belirlenen proje aktiviteleri arasındaki ilişkileri belirleme ve kayıt altına alma sürecidir.

3. Aktivite Kaynaklarının Tahmin Edilmesi: Her bir aktiviteyi gerçekleştirmek için gerekli malzeme, insan, araç ya da gereçlerin türünü ve miktarlarını tahmin etme sürecidir.

4. Aktivite Sürelerinin Tahmin Edilmesi: Belirlenen kaynaklarla her bir aktivitenin tamamlanması için gerekli çalışma sürelerinin belirlenmesi sürecidir. Belirleme tahmin etme teknikleri (Parametrik, 3 Nokta, Örneksel vb.) kullanılarak yapılır.

5. Zaman Çizelgesinin Geliştirilmesi: Proje zaman çizelgesi için, aktivite sıralamalarının, sürelerinin, kaynak gereksinimlerinin ve zaman çizelgesi kısıtlarının analiz edilme sürecidir. CPM, Kaynak Dengeleme, Zaman Sıkıştırma teknikleri kullanılarak yapılabilir.

6. Zaman Çizelgesinin Kontrolü: Projedeki ilerlemeyi takip etmek ve güncellemek için projenin mevcut durumunun izlenmesi ve zaman çizelgesindeki değişiklerin yönetilmesi sürecidir. Varyans Analizi, Önden Gitme ve Beklemelerin yönetilmesi teknikleri uygulanır.

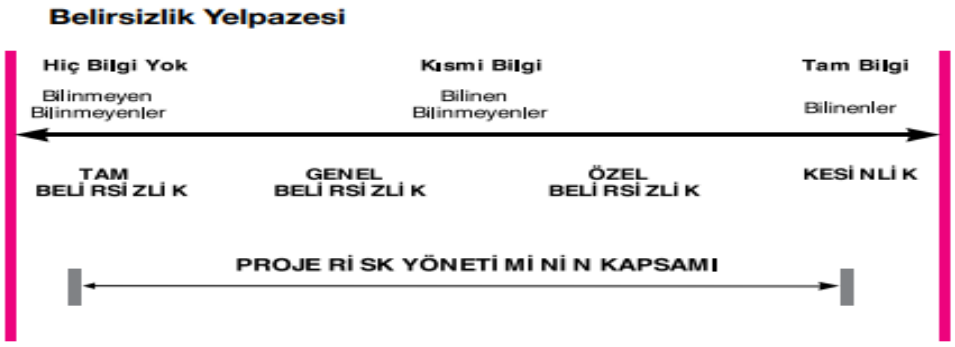
Bu süreçler birbirleriyle ve diğer alanlardaki süreçlerle etkileşim halindedir. Her süreç, projenin ihtiyaçlarına bağlı olarak, çalışan sayısında değişkenlik gösterebilir. Her süreç her projede en az bir kez gerçekleştirilir ve proje fazlara ayrılıyorsa bir ya da birkaç fazda gerçekleştirilebilir.

1.2.2.4. Proje Risk Yönetimi

İstenmeyen olayların etkisinin/olasılığının kontrol altına alınması, izlenmesi ve en aza indirilmesi için kaynakların koordineli ve ekonomik uygulanabilmesi amacıyla risklerin tespiti, değerlendirilmesi ve önceliklendirilmesidir (Hubbard, 209).

PMI'a göre, proje riskleri gerçekleştiklerinde; zaman, maliyet, kapsam veya kalite gibi en az bir proje hedefini olumsuz etkileyen belirsiz olay veya durumdur.

Geniş anlamda riske iki farklı yaklaşım söz konusudur: Birinci yaklaşımda risk, belirsizlik anlamına gelir. Bu durumda hem olumlu hem de olumsuz sonuçlar içermektedir. İkinci yaklaşımda risk tehdit/tehlike anlamına gelir. Bu durumda yalnızca olumsuz sonuçlar içerir. Risk, proje hedeflerini olumlu/olumsuz etkileyebilen belirsiz olayların yığılımlı etkisi olarak tanımlanmaktadır.



Belirsizlik Yelpazesi Şeması

Belirli Proje Riskleri

Belirli proje riskleri, kaynağı ve niteliklerine göre beş ana sınıfa ayrılır. Bu risklere aşağıda yer verilmektedir.

a) Dış kaynaklı, öngörülemeyen, kontrol edilemeyen;

- Mevzuat kaynaklı,
- Doğal afetler,
- Varsayılabilecek olaylar,
- Dolaylı etkiler,
- Tamamlanmama eksiklikleri.

b) Dış kaynaklı, öngörülebilen, kontrol edilemeyen;

- Pazar riskleri,
- İşletme,
- Çevresel etkiler,
- Sosyal etkiler,

- Parasal değişiklikler,
- Enflasyon,
- Vergilendirme.

c) İçten kaynaklanan, teknik olmayan, genellikle kontrol edilebilen;

- Yönetim,
- Program,
- Maliyet,
- Para akışı,
- Yeterlilik kaybı.

d) Teknik, genellikle kontrol edilebilen;

- Teknolojik değişiklikler,
- Performans,
- Projenin teknolojisine özgü risk,
- Tasarım,
- Projenin büyüklüğü ya da karmaşıklığı.

e) Hukuksal, genellikle kontrol edilebilen;

- Lisanslar/Patent hakları,
- Sözleşme,
- Örgüt dışından gelen hukuk davası,
- Örgüt içinden gelen hukuk davası,
- Mücbir sebepler

Risk ne zaman alınır?

Risk sadece, beklenen yararın başarısızlığın maliyetini ve kazanma şansını da kaybetme olasılığını aştığı durumlarda alınmalıdır. Risk alan aşağıdaki soruların gerçek cevabını aramalıdır:

- Risk niçin alınmalıdır?
- Ne kazanılacak?
- Ne kaybedilebilir?
- Başarı (ve kaybetme) şansı nedir?
- İstenen sonuç elde edilemez ise ne yapılabilir?
- Beklenen ödül maruz kalınacak riske değer mi?

Risk ne zaman alınmaz?

Proje yöneticisinin risk almayacağı durumlar:

- İşletme bir kayba tahammül edemediğinde,
- Maruz kalınan riskin sonucu çok yüksek olduğunda,
- Durum/projenin alınacak riske değmediği durumlarda,
- Proje lehinde herhangi bir avantaj söz konusu değilse,
- Yarışma dürüst olmayacaksa,
- Sağlanacak yararlar belirlenmemiş ise,

- Kabul edilebilir seçenek sayısı çok ise (belirsizlikte artar),
- Alınacak risk bir proje hedefine ulaşmasında yetersiz kalıyorsa,
- Varsayımların beklenen değeri negatif ise (ya da küçük bir değişiklik negatif yapıyorsa),
- Eldeki değerler düzensiz bir dağılım oluşturuyorsa,
- Sonucu hesaplamak için yeteri veri yoksa,
- Sonucun tatminkar olmaması halinde uygulanacak bir olasılık/maliyet planı yok ise.

Risk yönetimi, projenin amacına ulaşmasına yardımcı olması için bir projenin yaşam döngüsü boyunca ortaya çıkabilecek potansiyel ve beklenmedik sorunların belirlenmesi ve yönetilmesi çalışma çabalarını ifade etmektedir. Projelerde yaşanan problemlerin %90'ı öngörülebilmektedir. Reaktif yaklaşım (problem olduğunda müdahale) kayıpların yaşanmasına sebep olabilmektedir. Proaktif yaklaşım olan proje risk yönetimi, kayıpların önlenmesi veya olumsuz etkilerini en aza indirilmesini sağlamaktadır.

Proje risk yönetiminin hedefleri projede olumlu olayların olasılık ve etkilerinin arttırılması, olumsuz olayların olasılık ve etkilerinin azaltılmasıdır. Risk yönetimi, farklı proje türlerinde farklı anlamlara gelebilmektedir. Büyük ölçekli projelerde, risk yönetimi stratejileri, sorunlar ortaya çıktığında azaltma stratejilerinin uygulanmasını sağlamak için her risk için kapsamlı ayrıntılı planlama içerebilmektedir. Daha küçük projeler için risk yönetimi, yüksek, orta ve düşük öncelikli risklerin basit, önceliklendirilmiş bir listesi anlamına gelebilmektedir.

Başarılı bir risk yönetiminin anahtarı erken tanımlama, planlama ve kararlı bir uygulamadır. İyi planlama; kapsamlı ve yinelenen bir yaklaşımla risk tanımlama, değerlendirme ve tepki geliştirmeyi mümkün kılar. Risk yönetimi başlı başına bir yönetim disiplinidir, ancak belirsizlikleri ve riskleri tamamen ortadan kaldıracak sihirli bir yönetim disiplini değil, potansiyel risklerin sistematik olarak değerlendirilerek, olası zararlarının etkisini azaltıcı yönde, verilere dayalı karar vermeyi sağlayan bir disiplindir ve diğer disiplinlerle bir bütünlük içerisinde uygulanması gerekir (Özkılıç, 2014).

Risk yönetimi ikilemi grafik modelinde gerçekleşen bir risk olayındaki değişiklikler (zaman tahminlemesindeki, maliyet kestirimindeki veya dizayn teknolojisindeki bir hata), projenin konsepti, planlaması ve başlangıç evresi açısından çok önemlidir. Projede bir risk olayının maliyete etkisi, risk olayı ne kadar geç meydana gelirse o kadar çoktur. Projenin başlangıç safhaları potansiyel risk olaylarının belirlenmesi ve etkilerinin en küçüklenmesi için en elverişli safhalardır. Proje yarılandıktan sonra meydana gelen risk olaylarının maliyeti de hızla artar. Projenin risk olaylarını proje başlamadan önce belirlemek ve bu muhtemel risklerin karşılıklarını hazırlamak, bu risk olaylarını proje süreci içinde yönetmekten daha ihtiyatlı ve mantıklı olacaktır.

Risk yönetimi reaksiyonel bir yaklaşım değil, riske mantıklı yanıt geliştiren bir yaklaşımdır. Aslında risk yönetimi, sürprizleri azaltmayı ve istenmeyen olayların olumsuz sonuçlarını minimize etmeyi sağlayan bir önleyici süreçtir. Risk yönetimi, proje yöneticisine gerektiğinde zaman, maliyet veya teknik avantajlar söz konusu olduğunda risk alabilme olanağı vermektedir. Proje risklerinin başarılı yönetimi, proje yöneticisine gelecek üzerinde daha iyi bir kontrol imkânı vermekte ve projenin amaçlarına zamanında, belirlenen bütçe ile ve teknik gereklilikleri karşılayacak şekilde ulaşma şansını önemli ölçüde artırmaktadır.

Proje riskinin iki ana kategorisi vardır:

1- İş yararlarını etkileyen kategori,

2- Projenin kendisini etkileyen kategori.

Proje sponsoru ilk risk kategorisini hafifletmekten, proje yöneticisi ise ikinci kategoriyi hafifletmekten sorumludur.

Proje riskine ilişkin adımlara aşağıda yer verilmektedir.

Adım 1 Riskin Belirlenmesi:

Risk yönetimi süreci proje üzerinde etkili olabilecek tüm olası risklerin bir listesinin oluşturulmaya çalışılması ile başlamaktadır. Tipik olarak planlama süreci boyunca proje yöneticisi, risk yönetimi takımını ve diğer ilgilileri bir araya getirmektedir. Bir araya gelen takım beyin fırtınası ve diğer problem belirleme teknikleri ile potansiyel problemleri belirlemektedir. Değerlendirme süreci boyunca takım üyeleri belirlenen riskleri analiz etme ve uygun olmayanları eleme şansı bulmaktadırlar. Risk belirleme sürecinin ön safhalarında çok sık karşılaşılan bir hata, olabilecek olayların sonuçlarından çok olmuş olayların sonuçları üzerinde odaklanmaktır. Örneğin takım üyeleri bu yüzden projedeki asıl riski gözden kaçırabilirler. Asıl üzerinde odaklanılması gereken riskin gerçekleşmesine neden olabilecek olaylardır.

Başlangıçtaki odak noktası, projenin spesifik noktalarındaki risklerden çok projenin tamamı üzerinde etkili olan risklerdir. Makro riskler tanımlandıktan sonra projenin spesifik noktalarındaki riskler ele alınabilmektedir. Mikro risklerin belirlenmesi için etkili bir araç, iş kırılım yapısıdır. İş kırılım yapısının kullanımı risklerin gözden kaçırılma şansını azaltmaktadır. Büyük projelerde spesifik noktalarda çoklu risk takımları organize edilmekte ve bunların raporları toplanarak proje yöneticisine iletilmektedir.

Risk profili, yöneticilerin riski tanımlamaları ve analiz etmeleri için diğer bir araçtır. Risk profili, geleneksel olarak proje üzerindeki belirsizlikleri arayan bir soru listesidir. Bu sorular daha önceki benzer projelerden elde edilen deneyimler yardımıyla belirlenmektedir. İyi bir risk profili, ele alınan projenin tipine göre yeniden biçimlendirilmektedir. Risk profilleri firmanın yalnızca güçlü ve zayıf yönlerini ortaya koyar. Sonuç olarak risk profilleri, firmanın hem teknik hem yönetsel riskleri belirlemektedir.

Risk belirleme süreci çekirdek kadro ile sınırlandırılmamalıdır. Müşterilerden, sponsorlardan, taşeronlardan, satıcılardan ve diğer katılımcılardan sürekli bilgi istenmelidir. İlgili katılımcılarla resmi olarak görüşülmekte veya risk yönetimi takımına alınabilmektedir. Amaç olayları meydana gelmeden önce tespit etmektir. İş kırılım yapısı ve risk profilleri, ele alınmamış olay kalmadığından emin olmak için önemli iki araçtır. Aynı zamanda, iyi yapılmış bir risk tanımlama işlemi ile risk bir miktar önlenebilmektedir. Proje yöneticisinin doğru bir tavır alması ve yönetim sürecini tamamlaması önemlidir.

Risk belirlemede yardımcı olabilecek bazı sorular aşağıdaki gibidir:

- Amaca ulaşma yolunda neler yanlış gidebilir?
- Nerelerde sorun yaşanır?
- Başarısız olmaya neden olabilecek işler nelerdir?
- Hangi alanlarda ya da yerlerde zayıflıklar var?
- Hangi varlıklar daha çok korunmalı?
- Hırsızlık veya yolsuzluk alanları neler olabilir?
- Faaliyetler hangi durum ya da olaylar karşısında aksayabilir, durabilir?
- En kritik bilgi kaynakları nelerdir?
- En fazla harcama yapılan alanlar hangileridir?
- Takdire dayanan, bir kişi tarafından alınan kritik kararlar nelerdir?
- Hangi faaliyet veya süreçler daha karmaşıktır?
- İdari, mali ve cezai yaptırımlara maruz kalınan alanlar hangileridir?

Adım 2 Risk Ölçme:

Adım 1 ile potansiyel risklerin bir listesi oluşturulmaktadır. Ancak bu risklerin tümü dikkate değer değildir. Bazı riskler aşikar ve görmezden gelinebilecek boyuttayken, bazıları projenin başarısı açısından çok ciddi olabilmektedir. Bu aşamada proje yöneticisi, listeden önemsiz ve gereğinden fazla önem verilmiş olan riskleri elemek için bir yöntem geliştirmelidir.

Senaryo analizleri, risk analizlerinde en çok kullanılan ve en kolay uygulanan yöntemdir. Takım üyeleri her bir riski aşağıdaki durumlar açısından değerlendirmelidir.

- Kabul edilemez olaylar,
- Olayların meydana gelişlerinin tüm sonuçları,
- Olayların etkilerinin büyüklüğü ve şiddeti,
- Olayların meydana gelme olasılıkları,
- Olay projenin hangi aşamasında meydan gelebileceği,
- Ele alınan projenin diğer kısımları veya başka projelerle olan etkileşimi.

Örneğin belirli bir materyalin temininde sıkıntı yaşanması olasılığı yüzde 80 olsun. Bu sıkıntının yaşanması; projenin ertelenmesi, sıkı program, düşük esneklik ve maliyet artışı gibi sonuçlara yol açabilir. Olayın etkisi ile maliyet %10 ve proje süresi %5 artabilir. Bu sıkıntı, projenin dizayn aşamasında meydana gelecektir. Bu projenin gecikmesi belki de başka projeleri de geciktirecek veya önceliklerini değiştirecektir. Bu bilginin elde edilebilmesi riskin ölçülmesini kolaylaştıracaktır.

İşletmeler genellikle farklı risk büyüklüklerini tek bir risk ölçüm matrisinde kategorize etmeyi kullanışlı bulmaktadırlar. Matris tipik olarak risk olaylarının olabilirlikleri ve olumsuz etkilerini dikkate almaktadır.

Risk şiddet matrisinde hücreler sırasıyla büyük, orta ve düşük riskleri ifade edecek şekilde kırmızı, yeşil ve sarı renkli bölgelere ayrılmaktadır. Kırmızı bölge matrisin sağ üst köşesine, yeşil bölge sol alt köşesinde, sarı bölge ise orta bölgede yer almaktadır. Riskin etkisi olabilirliğinden daha önemli kabul edildiği için kırmızı bölge olabilirlik ekseninde en aşağılara kadar inmektedir. Risk şiddet matrisi riskler arasında bir öncelik sıralaması oluşturmak için bir temel oluşturmaktadır. Birinci öncelik kırmızı bölgedeki risklere ikinci öncelik sarı bölgedeki risklere ayrılmaktadır. Yeşil bölgedeki riskler ise en son ele alınmaktadır.

Olasılık					
Çok Yüksek	5	10	15	20	25
Yüksek	4	8	12	16	20
Orta	3	6	9	12	15
Düşük	2	4	6	8	10
Çok Düşük	1	2	3	4	5
Etki	Çok Hafif	Hafif	Orta	Ciddi	Çok Ciddi

Risk Şiddet Matrisi

Etki x Olasılık x Belirlenememe = Risk Değeri

Yukarıdaki formüldeki her üç bileşen de beş üzerinden ölçüme göre derecelendirilir. Örneğin, belirleme (detection), proje takımının risk olayının ne kadar yakında gerçekleşeceğinin farkına

varabilme becerisi olarak tanımlanır. Bu bağlamda, 1 puan bir şempanzenin bile yaklaştığını kolaylıkla anlayabileceği türden risklere verilir. En yüksek puan 5 ise gerçekleşeceği fark edildiğinde artık çok geç denecek türden risklere verilir (örneğin sistem freezing). Benzer bir derecelendirme olayların meydana gelme olasılıklarına ve etkilerine de uygulanabilir. Risklerin ağırlıklandırılması tüm riskler için hesaplanacak olan bu "*risk değerl*eri" temel alınarak yapılır. Örneğin 1 etki puanı olan, çok düşük olasılıklı ve önceden belirlenebilmesi çok kolay olan bir olay için risk değeri 1 (1x1x1=1) olacaktır. Diğer taraftan, etki puanı 5, meydana gelme olasılığı çok yüksek ve önceden belirlenebilmesi imkânsıza yakın bir olayın risk değeri 125 (5x5x5=125) olur. Risk değerleri için belirlenen bu 1-125 aralığı risk olaylarını sınıflandırmaya yardımcı olur.

Adım 2.1 Olasılık Analizi:

Proje riskinin ölçülmesi sürecinde proje yöneticilerinin yararlanabileceği birçok istatistiksel yöntem vardır. Karar ağaçları, alternatif eylem biçimlerini beklenen değerlerini dikkate alarak değerlendirir. Net bugünkü değerin (NPV) istatistiksel değişimi, projedeki nakit akışında karşılaşılabilecek risklerin ölçümünde kullanılır. Geçmiş projelerin nakit akışları ile *S*-eğrileri (*S*-curves) (proje süresince herhangi bir andaki toplam maliyeti gösteren eğri) arasındaki korelasyon projenin nakit akış riskinin belirlenmesinde kullanılır.

PERT ve PERT simülasyon yapılan işleri gözden geçirmek ve proje riskini ölçmekte kullanılır. PERT ve diğer benzer teknikler projedeki tüm maliyetleri dikkate alarak daha makro açıdan riskleri sıralar. Burada tek tek olaylar üzerinde değil, projenin belirlenen zamanda ve belirlenen bütçe ile tamamlanabilmesinin olabilirliği üzerinde odaklanılacaktır. Bu teknikler projenin tüm risklerini ve gerekli olan ek sermaye, kaynak ve zamanın ölçülmesinde oldukça kullanışlıdır. PERT simülasyonun kullanımı gittikçe artmaktadır. Çünkü PERT için gerekli olan aynı verileri kullanır ve simülasyonu gerçekleştirmek için gerekli yazılımlar mevcuttur.

Adım 2.2 Yarı Nicel Analiz:

Proje yöneticileri risk analizi için olasılık üretilmesinde ve kullanılmasında çoğu zaman isteksizlerdir. Çünkü proje takımına riski telaffuz ettirmektir. Bu bilgi çok pratik olabilir ve aynı zamanda, olasılık ve fayda (utility) teorisinin yararlarını da beraberinde getirmektedir.

Proje yöneticileri tarafından kullanılan yaklaşımlardan biri yarı nicel senaryo analizidir. Birçok risk türünün zamana bağımlı olması, projenin ertelenmesini etkilemesi ve risk takımı üyeleri tarafından kolayca anlaşılması nedeniyle bu yaklaşım süreleri kullanılmaktadır.

Yaklaşık olarak nicel senaryo yaklaşımı, bir sonraki adım için senaryo analizini temel almaktadır. Etkilerin doğrulanmasında numaraların kullanılması tanımlanan risklerin ve analizlerinin kontrolünde bir gerçeklik sunmaktadır.

Adım 3 Riske Tepki Geliştirme:

Bir risk olayı tanımlandığında ve değerlendirildiğinde, bu olay için hangi tepkinin uygun olduğuna kararı verilmelidir. Riske verilecek tepkiler; riski azaltma, riskten kaçınma, riski transfer etme, riski paylaştırma ve riski önleme gibi sınıflara ayrılabilmektedir.

Adım 3.1 Riskin Azaltılması:

Riskin azaltılması, ilk olarak dikkate alınan ve potansiyel kayıpları azaltmakla ilgili alternatiftir. Riskin azaltılması için iki strateji vardır. Bunlar risk olayının meydana gelişinin olabilirliğinin azaltılması ve/veya risk olayının proje üzerindeki olumsuz etkisinin azaltılmasıdır. İlk strateji için bir bilgi sistemi projesi örnek olabilir. Konu proje süresinin ve maliyetinin gerçek değerinin altında kestirilmesi olduğundan proje yöneticileri bu belirsizlikleri telafi etmek için kestirimlerini arttırmaktadırlar. Proje süresinin ve maliyetinin ayarlanabilmesi için geçmiş benzer bir proje ile ele alınan proje arasındaki oran oldukça sık kullanılır. Oran tipik olarak sabittir. Örneğin, eğer eski bir projede her bir satır bilgisayar programının kodunun yazılması 10 dakika sürmüşse, 1.10 oranı yeni projede bu sürenin 11 dakika olacağı anlamına gelmektedir.

Adım 3.2 Riskten Kaçınma:

Riskten kaçınma, riski ya da etkilerini tamamen ortadan kaldırmaya yönelik geliştirilen stratejidir. Riskin ortadan kaldırılması amacıyla proje planının değiştirilebilir. Bu yöntemle tüm risklerin ortadan kaldırılması mümkün olmasa da, proje başlamadan önce bazı belirli risklerden kurtulunabilir. Örneğin, deneme aşamasında olanlar yerine doğruluğu ispatlanmış teknolojilerin kullanılması teknik riskleri ortadan kaldıracaktır. Endonezyalı bir tedarikçi yerine Avustralyalı bir tedarikçinin seçilmesi kritik materyallerin sağlanması konusunda karşılaşılabilecek politika kaynaklı riskleri ortadan kaldırabilir.

Bazı riskler kaçınılmazdır. Örneğin, iflas riski, takım sorumluluğu riski ya da işletmeler ve insanlar için erken ölüm riski kaçınılmazdır. Bu kaybetme riski genellikle azaltılabilir ama yok edilemez. Kaçınmak diğer riskler için yalnızca kabul edilebilir bir alternatiftir. Temel kural; kaybetme riski yüksekse ve kaybetme şiddetti de yüksekse kaçınmak çoğu kez en iyisidir ve bazen tek pratik seçenektir.

Adım 3.3 Riskin Transferi:

Riskin transferi, riskin gerçekleşmesi durumunda oluşabilecek zararı karşılayacak çözümler bularak (örneğin sigorta yaptırmak) riskin başkalarına aktarılmasıdır. Riskin diğer bir partiye taşınması çok karşılaşılan bir yöntemdir ancak bu riski azaltmamaktadır. Ancak bu aktarım hemen hemen her zaman ek bir ödemeyle sonuçlanmaktadır. Sabit ödemeli anlaşmalarda riskin proje sahibinden bir yükleniciye transfer edilmesi tipik bir örnektir.

Adım 3.4 Riskin Paylaştırılması:

Riskin paylaştırılması, riskin belli oranlarla farklı partilere bölüştürülmesidir. Bunun bir örneği olarak Airbus A340 verilebilir. Araştırma ve geliştirme riskleri İngiltere ve Fransa'yı da içeren Avrupa ülkeleri arasında bölüştürülmüştür.

Adım 3.5 Riskin Kabul Edilmesi:

Bu aşamada risk ihtimalini veya etkisini etkilemek için herhangi bir önlem alınmamaktadır. Bazı durumlarda meydana gelen bir olayın riskini kabul etmek bilinçli bir yaklaşımdır. Bazı risk olayları transfer edilemeyecek ya da azaltılamayacak kadar büyük olabilir (örneğin deprem, su baskını vs). Proje sahibi bu tip riskleri kabul eder çünkü bu tip olayların meydana gelme olasılıkları oldukça zayıftır. Diğer durumlarda ek bütçe yapılırken belirlenmiş olan risklerin etkisi kolayca bu bütçeden karşılanabilecektir.

Proje başlamadan önce riske tepki geliştirme işine daha büyük çaba harcanması projede karşılaşılabilecek sürprizlerin olumsuz etkilerinin minimize edilme olasılığını artırır.

1.2.2.5. Kaynak Kullanımı Yönetimi

Kaynak kullanımı yönetimi, proje bütçesinin harcandığı süreçtir. Gerçekleşen harcamanın planlanan harcamaya uygunluğunun belirlenmesi amacıyla kaynak kullanımı ölçülmeli ve raporlanmalıdır. Harcamalara ek olarak üretkenlik izlenmelidir. Bu kapsamda kazanılan değer analizi (EVA) adı verilen teknik kullanılmaktadır.

EVA, proje sırasında aşağıdaki ölçütlerin düzenli aralıklarla karşılaştırılmasını içerir:

- Bugüne kadar gerçekleşen bütçe,
- Bugüne kadar gerçekleşen fiili harcama,
- Tamamlanacak tahmin,
- Tamamlanma süresi tahmini.

EVA, yaygın biçimde kullanılan bir performans ölçüm tekniğidir. Projenin kapsamı, maliyeti ve zaman çizelgesi ölçümlerini bütünleştirerek proje yönetim ekibinin proje performansını ve ilerlemesini değerlendirmesine ve ölçmesine yardımcı olan grafiksel bir araçtır.

Projenin yürütülmesi sırasında sapmalar ile karşılaşması hâlinde proje yöneticisinin cevabını araştırması gereken beş temel soru bulunmaktadır:

- Sapmaya neden olan problem nedir?
- Sapmanın zaman, maliyet ve performans üzerindeki etkisi nedir?
- Sapmanın, eğer var ise diğer çabalar üzerindeki etkisi nedir?
- Planlanan veya uygulanan düzeltici hareket nedir?
- Düzeltici hareket ya da hareketlerin beklenen sonuçları nedir?

Proje ekibi bu soruların cevaplarına göre gerekli düzenlemeleri yaparak projenin başarı şansını arttırabilir. Bilgi sistemleri denetçisinin proje yöneticisi tarafından bu hususların değerlendirilip değerlendirilmediğini incelemesi beklenir.

1.2.2.6. Proje Faydalarının İzlenmesi

Projeler için proje yaşam döngülerinden öte yeni oluşturulan sistem için tüm iş fayda ve toplamdaki iş maliyetlerini değerlendiren uzun planlamalara yönelik fayda geliştirecek şekilde planlanmış bir yaklaşıma ihtiyaç bulunmaktadır. Faydalar nadiren planlarda öngörüldüğü şekillerde ortaya çıkar. Proje faydalarının gerçekleştirilmesinin temel unsurları aşağıdaki gibidir:

- Faydaların yönetiminin veya faydalarının gerçekleştirilmesinin açıklanması,
- Bir ölçüm ve hedef atama,
- Bir izleme/ölçüm rejiminin oluşturulması,
- Varsayım belgelemesi,
- Gerçekleştirme için anahtar sorumluluklarının oluşturulması,
- İşte öngörülen faydaların doğrulanması,
- Gerçekleştirilmesi gereken fayda planlanması.

Projede faydanın gerçekleşme aşamaları;

- Anlama,
- Planlama,
- Gerçekleştirme,
- Raporlama,

şeklindedir.

Proje faydalarının gerçekleştirilmesi maliyet, kalite, geliştirme/teslim süresi, güvenilirlik ve bağımlılık gibi önemli faktörler arasında uzlaşmadır. Stratejistler kapsamlı bir çalışma yürütür ve elemeyi geçen/kazanan faktörleri değerlendirir. Ardından bu faktörlerin sistemleri tamamlamak ve bakımını yapmak için mevcut hizmetlerin güçlü, zayıf yönleri ile karşılaştırır. Birçok işletme bilgi sistemlerindeki değişiklikleri desteklemek için yapılandırılmış proje yönetimi ilkelerini kullanmaktadır. Bilgi sistemleri denetçisi işletmenin geliştirmeye ilişkin projeler için değeri veya yatırım getiri hedeflerini tutarlı bir şekilde yerine getirememesi, yazılım geliştirme yaşam döngüsündeki (YGYD-Software Develeopment Life Cycle-SDLC) ve ilgili proje yönetimi pratiklerindeki zayıflığını gösterebilir.

1.2.2.7. Maliyet Yönetimi

Proje büyüklüğüne bakılmaksızın maliyetin kontrol altında tutulması ve bütçenin aşılmaması, proje yöneticisinin her zaman titizlikle üzerinde durması gereken bir husustur. Maliyet izleme ve kontrol işlemleri, projenin tamamlanması amacına doğru gidilen yolda, yöneticinin projeyi plana uygun bir biçimde tamamlamasına yardımcı olmalıdır. Etkin bir maliyet yönetim sisteminde aşağıdaki unsurların yer alması gerekmektedir:

- Maliyet tahmini,

- Maliyet muhasebesi,

- Proje nakit akışı,

- Proje paydaşları nakit akışı,

- Direkt iş gücü maliyeti,

- Cezalar, kâr paylaşımı vb.

Maliyetin etkin kontrolünün sağlanması için izleme ve kontrol süreçlerinin aşağıdaki unsurları içermesi gerekir:

- Projenin tamamlanması için yapılacak tüm işlerin planlamasının iyi bir biçimde yapılmış olması,

- İş gücü ve maliyetlere ilişkin güçlü ve güvenilir tahminlerin yapılmış olması,

- Görevler ve kapsam arasındaki ilişkilerin iyi belirlenmiş olması,

- Disiplin altına alınmış bir bütçe ve harcama yetkisinin bulunması,

- Fiziksel ilerleme ve maliyet harcamaları için muhasebe işlemlerinin zamanında yürütülmesi,

- Kalan işin tamamlanması için gereken zaman ve maliyet için periyodik olarak tahminlerin yürütülmesi,

- Projenin doğasına uygun aralıklarla gerçekleşen iş ve maliyetler ile planlanan iş ve maliyetler arasında karşılaştırmalar yapılması.

Maliyetlerin kontrolü için oluşturulan bir izleme ve kontrol sistemi sayesinde proje yöneticisi, planlanan işler ile gerçekleşen işleri karşılaştırabilme imkânını elde etmektedir. Bu karşılaştırmalar sayesinde;

- Amaçların başarılı biçimde performans standardı hâline gelip gelmediği,

- Gerçek değerler ve planlanan değerler arasındaki farkın ortaya çıkmasına olanak verecek bir bütçenin hazırlanıp hazırlanmadığı,

- Performans standartlarının proje faaliyetlerinin güvenilir birer temsilcisi olup olmadığı.

Plan dâhilinde yer alan faaliyetin tamamlanması veya ek maliyet talebi ile ortaya çıkan maliyet değerlerinin derlenmesi maliyet kontrol sisteminin en önemli işlevidir. Gerçekleşen maliyet (PMBOK'a göre AC) ve tamamlanan işin bütçelenen maliyeti (PMBOK'a göre EV) her bir faaliyet için ayrıntılı olarak ele alınır ve planlanan değer (PMBOK'a göre PV). Maliyet Veri Derleme ve Raporlama Akış Diyagramı'na göre raporlanır (Ş 7.2; S.183).

Maliyet kontrolünün içermesi gereken ögeler aşağıda sıralanmıştır:

- Onaylanan maliyette değişikliklere yol açan faktörlerin ele alınması,

- Tüm değişiklik taleplerinin zamanında işleme konulması,

- Gerçekleşen değişikliklerin, gerçekleştikleri sırada yönetilmesi,

- Maliyet harcamalarının gerek dönem gerekse proje bazında onaylanan bütçeyi aşmamasının sağlanması,

- Onaylanan maliyete göre farklılıkların belirlenmesi ve bunların anlaşılması için maliyet performansının izlenmesi,

- Çalışma performansının, harcanan fonlarla karşılaştırılarak izlenmesi,

- Onaylanmayan değişikliklerin, raporlanan maliyet ya da kaynak kullanımına dâhil edilmesinin önlenmesi,

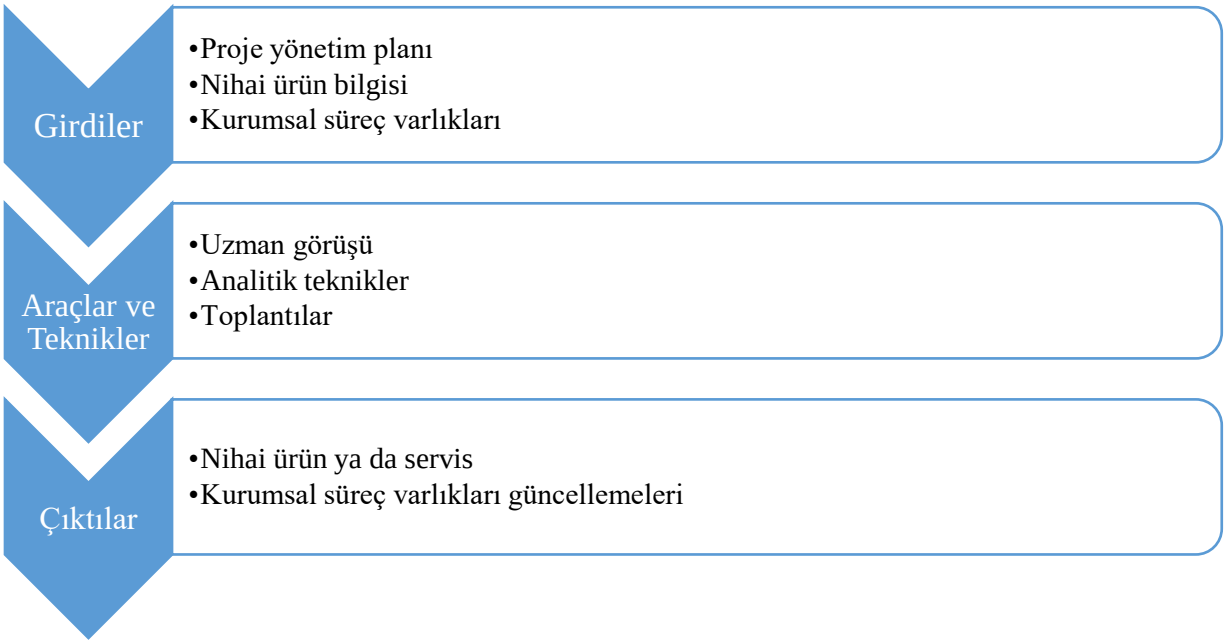
- Paydaşların, onaylanan tüm değişiklikler ve bunlarla ilgili maliyetler konusunda bilgilendirilmesi,

- Beklenen maliyet aşımlarını, kabul edilir sınırlar içinde tutmaya gayret gösterilmesi.

1.2.3. Projenin Kapatılması

Projenin kapatılması, proje akışı içerisinde yer alan son süreçtir. Projenin kapatılması aşamasına verilecek önem, proje yöneticisinin ve paydaşların daha sonraki projelerinde daha başarılı olmaları olasılığını arttıracaktır. Proje yöneticisi, projenin yaşam döngüsünün başlaması ile birlikte projenin kapatılması için gerekli belgeleri de hazırlamaya başlamış olmaktadır. Projenin gerçekleştirilmesi sırasında ortaya çıkan problemlerin nasıl çözümlendiğine ilişkin raporlar, kapsamda meydana gelen değişimlere ilişkin notlar, paydaş katkıları, kalite raporları, bütçe değişikliklerine ilişkin bilgi ve belgeler, proje süresince proje takımının edindiği deneyimler/kazanımlar/dersler bir bütün olarak projenin kapatılması aşamasının öğelerini oluşturmaktadır. Projenin kapatılması aşamasının başarılı olması için proje yöneticisinin bu aşamayı da bir süreç olarak yönetmesinde büyük fayda sağlayacaktır.

Projenin kapatılması sürecinin girdileri, araç ve teknikleri ve çıktıları aşağıda belirtilmiştir:



Proje Kapatma Girdi, Araç ve Teknikler ve Çıktılar Şeması

Proje bir noktada kapatılmalı ve bu aşamada kullanıcılara ve/veya sistem destek personeline yeni veya değiştirilmiş bir sistem teslim edilmelidir. Bu noktada, tespit edilen sorunların atanması gerekir. Proje sponsorunun, üretilen sistemi kabul edilebilir olması veya teslime hazır olduğundan memnun olmasının sağlanması önemlidir. Bu kapsamda, dikkate alınması gereken temel sorular aşağıdaki gibidir:

- Proje yöneticisi, nihai proje kapanış bildirimini ne zaman yayımlayacaktır?

- Nihai proje bildirimini kim yayımlayacaktır?

- Proje yöneticisi, proje ekibinin yeni projelere geçişine ya da düzenli atanmış görevlerine bırakılmasına nasıl yardımcı olacaktır?

- Proje yöneticisi açık kalan eylemler, riskler ve sorunlar için ne yapacaktır? Bu eylemleri kimler üstlenecek ve bunlar nasıl finanse edilecektir?

Bu aşamada ilgili belgeleme ve görevlerin teslimi gerçekleşecek, proje sahibinin sonucu onaylaması istenecektir. Onay için performans, test ve kontrol sonuçları dikkate alınacaktır. Proje kapanışında projenin başarısı veya projeden çıkarılan dersler proje ekibi ve ilgili paydaşlar tarafından belirlenecektir.

Projenin kapatılabilmesi için proje yaşam çevrimi boyunca yapılan tüm anlaşmaların kapatılması gerekmektedir. Aslında tedariklerin kapatılması alt süreci projenin başlaması ile başlayarak,

projenin kapatılması ile sona ermektedir. Tedarik kapatma alt sürecine, müşteriden ürün ile ilgili onayın alınması da dâhildir. Tedarik kapatılırken müşterinin de onayı alınmaktadır. Müşteri onayı alınırken genellikle iki durum ortaya çıkmaktadır. İlk olarak müşteri ve proje ekibi arasında toplantı düzenlenmeli ve toplantıda müşterinin ürünü kabul etmesi önemlidir. Müşteri ürünün beklentilerini karşıladığını ve belgede belirtilen tarihte ürünü teslim aldığını ifade eden metni imzalanır. Eğer proje herhangi bir neden ile ürün ortaya çıkmadan sonlandırılıyor ise yine proje yöneticisi ve müşteri arasında projenin sonlandırıldığına dair bir belge imzalanır. Bu durumda proje sonlandırma nedenleri belgeye eklenir. Son aşamada ortaya çıkabilecek problemlerin çözümlerinin maliyetinin yüksek olma ihtimali göz önüne alınmalı ve bütçe planlamasında son aşamada ortaya çıkabilecek öngörülemeyen problemler için bir harcama kalemi bulundurulmalıdır. Projenin kapatılması ile beraber tüm sözleşmeler, muhasebe kayıtları arşive aktarılır.

Öğrenilmiş dersler belgesi oluşturulurken, proje yöneticisi temel olarak kendisi ve ekibinin karşılaştığı problemleri belirlemekte ve bu problemlerden gelecek projelerde nasıl kaçınabileceğini kurgularsa çok daha faydalı bir belge ortaya çıkmaktadır. Proje yöneticisi karşılaştığı her problem için izleyen kesimde sıralanan soruları sorarak gelecekte benzer sorunlar altında kalmayacak veya benzer sorunlar ile karşılaşıldığında ne yapacağını bilecektir. Proje kapsamında tanımlanmış bir problem için sorulabilecek sorular:

- Problem ve etkisi nedir?

Bu problemden dolayı projede ne tür sorunların ortaya çıktığının öğrenilmesi, problemin tüm açılardan ele alınması gerekmektedir.

- Problemin ortaya çıkma nedeni ne olabilir?

Problemin nedenini bulmak için algılanan veya bilinen etkinin ne olduğu araştırılmalıdır. Problemin nedeni araştırılırken harcanacak zaman ve maliyetlerin problemin çözümü ile elde edilerek faydaya değip değmeyeceği araştırılmaktadır.

- Problem neden daha önce tespit edilemedi?

Projenin izleme ve kontrol süreci doğru kurgulanmamış olabilir. Bu nedenle izleme ve kontrol sistemlerinin doğru çalıştığından emin olunmalıdır. Ayrıca performans raporlarının doğru yazıldığı ve ilgili bireylere zamanında ulaştırılıp ulaştırılmadığı araştırılmalıdır.

- Problem kişisel performansa bağlı olabilir mi?

Söz konusu problem sadece bir proje çalışanının performans düşüklüğünden kaynaklanmış olabilir.

- Gelecekte bu problem bertaraf edilebilir mi?

Bazen söz konusu problemin tekrar ortaya çıkmasını engellemek imkânsız olabilir. Bu durumda gelecekte aynı problem ile karşılaşmamak için projede yapılması mümkün tüm değişikliklerin neler olduğu tespit edilmelidir.

- Bertaraf edilemez bir problem ise erken tespit mümkün müdür?

Bu aşamada proje ekibi ile beraber izleme, kontrol, raporlama ve kalite kontrol süreçleri yardımıyla problemin ortaya çıkışının mümkün olan en erken sürede anlaşılabilmesi için alınması gereken tedbirler gözden geçirilmelidir.

Proje süresince ortaya çıkan problemlerin çözüme kavuşturulması için sıralanan bu soruların cevapları yardımıyla öğrenilmiş dersler ortaya çıkmaya başlamaktadır. Projenin kapatılması aşamasında da bu dersler bir belge içerisinde bir araya getirilerek gelecekte aynı problemler ile karşılaşıldığında çözüme giden yol kısaltılmış olmaktadır. Öğrenilmiş dersler belgesi hazırlanırken projenin büyüklük ve tipine göre kategoriler oluşturulabilmektedir. Bu kategoriler içerisinde yer alan her bir öğrenilmiş ders bilgisi tek tek sıralanmaktadır.

Nihai proje raporu, proje yaşam çevrimi sürecince ortaya çıkan tüm gelişmeleri içeren bir belgedir. Rapor yardımıyla projenin üzerinden çok uzun bir süre geçse dahi proje süresince yaşananların açık ve anlaşılır bir şekilde hatırlanabilmektedir. Aynı şekilde benzer projelere başlayacak olan proje

yöneticileri için de nihai proje raporu yol gösterici bir belge olmaktadır. İyi hazırlanmış bir nihai proje raporunun projeye ilişkin birçok bilgiyi içermesi beklenmekle birlikte temel olarak rapor içerisinde yer alması gereken ana başlıklar izleyen biçimde sıralanabilmektedir.

- Projenin Genel Başarısı: Paydaşlardan, müşteriden ve proje çalışanlarından elde edilen geri bildirimler göz önüne alındığında proje ne kadar başarıya ulaşmıştır? Başarı için kullanılan ölçütler nelerdir ve proje kapsamında bu proje ölçütlerine ne kadar ulaşılmıştır?

- Projenin Genel Organizasyonu/Düzeni: Projenin tamamlanması ile beraber proje yöneticisi, projenin genel organizasyonunu nasıl gerçekleştirdiğini gözden geçirme şansını yakalamaktadır. Proje yöneticisi gerçekten iyi bir organizasyon gerçekleştirilip gerçekleştirilmediğini araştırmaktadır ve bir sonraki projede nelere dikkat edilmesi gerektiğini sıralamaktadır.

- Projenin Güçlü ve Zayıf Yanları: Proje yöneticisi, bir güçlü ve zayıf yönler, fırsatlar ve tehditler (SWOT) analizi veya proje yaşam çevrim süreci boyunca tutulan not ve raporlar yardımıyla projenin güçlü ve zayıf yanlarını ortaya çıkararak raporuna ekler. İleride raporun hazırlandığı proje konusunda yapılacak yeni çalışmalar için varsa öneriler ayrıntılandırılarak listelenmektedir.

- Proje Ekibine İlişkin Tavsiyeler: Projenin başlatılmasından kapatılmasına kadar geçen süre zarfında proje ekibine ilişkin bilgi alışverişleri olması kaçınılmazdır. Proje ekibi içerisinde bireyselliği ön planda tutanlar ile takım çalışmasına yatkın olanlar belirlenerek çeşitli tavsiye kararları sıralanabilmektedir.

- Sonuç Veren Teknikler: Projede sonuç veren tekniklere ilişkin özet bir liste hazırlanır. Bu liste hazırlama işine projenin başlaması ile hayat verilir. Sonuçların veya çıktıların elde edilebilmesi için ortaya konan çabaların neler olduğuna dair bilgiler bu liste içerisinde yer almalıdır.

Nihai proje raporunun hazırlanmasında paydaş ve müşteri görüşlerinin de yer almasına özen gösterilmelidir. Projenin kapatılması sürecinde proje nihai raporu oluşturulması amacıyla izlenen içindekiler yapısı oluşturulabilir. Projenin türüne ve büyüklüğüne göre rapor daha da ayrıntılı hâle getirilebilir. Bu raporda aşağıdaki hususlara yer verilir.

- Proje adı ve tarih: Projenin açık ismi, varsa proje numarası ve raporun hazırlanma tarihi belirtilir.

- Projenin kapatılmasına ilişkin betimleyici istatistikler: Gerçekleşen maliyet değeri, plana göre projenin tamamlanma tarihi, ekibin performansı ve kalite başarım miktarı belirtilmektedir.

- Varsa ilk faydalar: Proje sayesinde elde edinilen faydaların birçoğu ilk anda ortaya çıkmayabilir. Eğer proje kapatılması sırasında kazanılmamış faydalar olduğu düşünülüyor ise proje yöneticisinin proje kapatıldıktan ve bu faydalar ortaya çıktıktan sonra bir rapor hazırlanması beklenir. Bu rapora proje tespit raporu adı verilir.

- Öğrenilmiş dersler: Hazırlanmış öğrenilmiş dersler belgesi nihai rapor içerisine eklenir.

- Proje yöneticisinin projeye ilişkin görüş ve önerileri: Proje sürecinde nelerin yolunda gittiği, nelerin yolunda gitmediği ve kazanılmış dersler kapsamına girmeyen ama projeye ilişkin olarak değerlendirilmesinde fayda olan konular ele alınır.

Proje yöneticisi, projenin başlangıcından itibaren iyi bir evraklama sistemine ihtiyaç duyar. Bu evraklama sistemi bilgisayar ortamında olabileceği gibi dosya içerisinde basılı belgelerin ve faturaların bir kütüphanede tasnif edilmesi biçiminde de gerçekleşebilir. Önemli olan proje için üretilen tüm belge ve faturaların kronolojik ve ilgi alanlarına göre indekslenip bir arada tutulmalarıdır. Evraklama sistemi sayesinde projenin yaşam çevrimi süresince projenin nasıl geliştiği ve sona doğru ilerlendiği görülecektir. Eğer proje çeşitli sebeplerden dolayı sonlandırıldı ise sonlandırma nedenine kaynak olabilecek proje tabanlı gelişmelere ilişkin izler de eski evrakların incelenmesi ile tespit edilebilir. Düzgün ve kolay anlaşılır bir arşiv sistemi oluşturulması projenin başarısına katkı sağlayacaktır.

Bir değişiklik yönetim sistemi izleme bilgileri aşağıdaki hususları içermelidir:

- Programcıların oturum açma kapama geçmişi, program silme tarihçesi, görevler ayrılığı (SoD) ve kalite güvence faaliyetlerinin geçmişi ile atanan programcı, iş emri tarihi, yapılan değişiklikler ve kapanış tarihi gibi detayların yer aldığı iş emri faaliyetlerinin geçmişi,

- Bütünlüğün sağlanması amacıyla üretim kütüphanesi güvenliğinin yeterliliğinin belirlenmesi amacıyla mevcut kontrollerin incelenmesi,

- Kontrol hedeflerinin sağlandığından emin olunması için sistem bakım süreçlerinin değerlendirilmesi, test sonuçları ve denetim kanıtlarının dokümante edilmesi,

- Sistem bakım standartları ve prosedürlerin kilit personel ile gözden geçirilmesi,

- Uygulama sonrası gözden geçirmelere katılınması.

Proje yönetiminde bilgi sistemleri denetçisinin rolü proje süresince veya proje kapanışı tamamlandıktan sonra gerçekleştirilebilir. Bu kapsamda;

- Kontrol gerektiren alanların belirlenmesi amacıyla sistemin ana bileşenleri, hedefleri, kullanıcı gereksinimlerinin belirlenmesi için anahtar sistem geliştirme ve proje paydaşları ile tanışılması,

- Sistemin risklerinin ve zayıf yönlerinin belirlenmesi ve sıralanması için sistem geliştirme ve proje paydaşları ile uygun kontrol yöntemlerinin belirlenmesi,

- Sistem riskinin azaltılması için önleyici kontrollerin belirlenmesi,

- Mevcut kontrollerin değerlendirilmesi ve işlerliğinin proje paydaşları ile incelenmesi,

Kontrollerin uygulandığı, gereksinimlerin karşılandığından ve sistem geliştirme/satın alma metodolojisinin takip edildiğinden emin olunabilmesi amacıyla tüm sürecin, belge ve teslimatların incelenmesi, tüm değişiklik, düzenleme ve işleme işlemlerine ilişkin kontrollerin tanımlandığından emin olunması.

Örnek Sorular

Soru 1: Etkin maliyet yönetimi aşağıdaki unsurlardan hangilerini içermez?

A) Maliyet tahmini

B) Proje nakit akışı

C) Direkt iş gücü maliyeti

D) Cezalar, kâr paylaşımı vb.

E) Proje analiz süreçleri

Cevap: E

Soru 2: Bir bilgi sistemleri denetçisi, aşağıdakilerden hangisinin proje yöneticisi tarafından izlenip kontrol altına alındığına bakmalıdır?

A) Proje planı ile karşılaştırmalı olarak hâli hazırda yürütülmekte olan faaliyet ya da faaliyetlerin durumunun tespiti

B) O ana kadar tamamlanan iş miktarı ve kalitesi

C) Proje planı ile karşılaştırmalı olarak maliyetlerin ve harcamaların durumu

D) Proje ile ilgili tüm paydaşların (proje çalışanları, işverenler, müşteriler vb.) projeye karşı olan genel tavır ve hareketleri

E) Hepsi

Cevap: E

Soru 3: Bir bilgi sistemleri denetçisi Proje ekibinde kalite kontrol için nelerin yapıldığına bakabilir?

A) İstatistiksel kalite kontrol teknikleri

B) Hatalı ürünlerin müşteriye ulaşmasına engel olacak denetlemeler

C) Kalitede anormalliklere ve dalgalanmalara yol açan nedenlerin tespiti

D) Kalitenin istenen seviyede olması için gerekli kontrol sınırlarının belirlenmesi

E) Hepsi

Cevap: E

Soru 4: Bir bilgi sistemleri denetçisi kazanılan değer analizi (EVA) tekniği ile proje sırasında aşağıdaki ölçütlerden hangisinin düzenli aralıklarla karşılaştırılmasını içerdiğinden emin olmalıdır?

A) Bugüne kadar gerçekleşen bütçe

B) Bugüne kadar gerçekleşen fiili harcama

C) Tamamlanacak tahmin

D) Tamamlanma süresi tahmini

E) Hepsi

Cevap: E

Soru 5: Riskin yönetilmesi sorumluluğunun başka bir tarafa aktarılmasını sağlayan riske yanıt stratejisi aşağıdakilerden hangisidir?

A) Riski aktarma

B) Riskten kaçınma

C) Riski kabullenme

D) Riski azaltma

E) Duruma göre davranma

Cevap: A

Soru 6: Bir bilgi sistemleri denetçisi projenin SMART metoduna uygunluğunu incelerken aşağıdaki bileşenlerden hangisini göz önünde bulundurmaz?

A) Ölçülebilir

B) Belirli bir amaca hizmet eden

C) Süreli

D) Gerçekçi

E) Belirsiz

Cevap: E

Soru 7: Projelerde zaman, maliyet ve kapsam yönetimi önemli üç önemli kısıttır. Bir projede kapsam sabit kalmak şartı ile proje maliyeti düşürülmek istenirse, proje süresi genellikle nasıl değişim gösterir?

A) Proje süresi kısalır.

B) Proje süresi değişmez.

C) Proje süresi uzar.

D) Proje maliyeti hiçbir zaman düşürülemez.

E) Kapsam hiçbir zaman sabit kalmaz.

Cevap: C

Soru 8: Aşağıdakilerden hangisi BS denetçisinin iş kırılım yapısı ve ilgili çalışma paketlerine ilişkin göz önünde bulundurması gereken temel konulardan değildir?

A) En üst iş kırılım seviyesi nihai teslimatı veya projeyi temsil etmektedir.

B) Bir iş kırılımının tüm öğelerinin aynı seviyede tanımlanması gerekmemektedir.

C) Münferit çalışma paketleri alt teslimatları üretmek için gereken görevlerin işini, süresini ve maliyetlerini tanımlamaktadır.

D) Münferit çalışma paketleri ortalama 10 günü aşmamalıdır.

E) Münferit çalışma paketleri birbirine benzer, iş kırılımı genelinde çoğaltılabilir.

Cevap: E

Soru 9: Aşağıdakilerden hangisi kritik yolu kısaltacak önlemler arasında sayılamaz?

A) Kritik etkinlikler üzerine yoğunlaşılması

B) Seri işlerin paralel olarak yeniden planlanması (Hızlı izleme)

C) Kritik etkinliklerin sürelerinin uzatılması

D) Çok kaynak kullanan etkinliklerin kısaltılması

E) Maliyeti yüksek işlerin kısaltılması

Cevap: C

Soru 10: Bir bilgi sistemleri denetçisi aşağıdakilerden hangisinin proje yönetim süreçleri arasında olmasını beklemez?

A) Başlatma

B) Planlama

C) Yürütme

D) Kontrol Etme

E) Bekleme

Cevap: E

Soru 11: Aşağıdakilerden hangisi tipik bir projede olması gereken özelliklerden biri değildir?

A) Her bir projenin sonucunda benzersiz bir ürün, hizmet veya sonuç ortaya çıkar.

B) Projeler geçici bir süre içinde gerçekleştirildikleri için her projenin bir başlangıcı ve bitişi vardır.

C) Projelerin sürekli kontrol edilmesine gerek yoktur.

D) Projelerin tanımlanmış bir bütçeleri ve hedefleri vardır.

E) Projeleri oluşturan faaliyetler benzersizdirler.

Cevap: C

Soru 12: Aşağıdakilerden hangisi projede yer alan grup/bireylerin rol ve sorumlulukları arasında yer alır?

A) Proje sponsoru

B) Kalite güvence

C) Proje Yöneticisi

D) Üst Düzey Yönetim

E) Hepsi

Cevap: D

Soru 13: Aşağıdakilerden hangisi bir bilgi sistemleri denetçisinin proje yönetimi denetimleri sırasında proje yönetim ekibinin kalite kontrol için kullanmasını bekleyeceği beceri ve teknikler arasında yer alır?

A) İstatistiksel kalite kontrol teknikleri

B) Hatalı ürünlerin müşteriye ulaşmasına engel olacak denetlemeler

C) Kalitede anormalliklere ve dalgalanmalara yol açan nedenlerin tespiti

D) Kalitenin istenen seviyede olması için gerekli kontrol sınırlarının belirlenmesi

E) Hepsi

Cevap E

Soru 14: Aşağıdakilerden hangisi proje portföy yönetiminin hedefleridir?

A) Bireysel olmayan proje portföy sonuçlarının iyileştirilmesi,

B) Projelerde önceliklendirme ve zaman planlamasının yapılması

C) İç ve Dış proje kaynak koordinasyonlarının sağlanması

D) Proje süresince bilgi akışının sağlanması

E) Hepsi

Cevap E

Soru 15: Aşağıdakilerden hangisi fizibilite çalışmasının alt maddesidir?

A) Proje Kapsamı

B) Mevcut analiz

C) Gereksinimler

D) Yaklaşım

E) Hepsi

Cevap: E

2. SİSTEM GELİŞTİRME YAŞAM DÖNGÜSÜ (SYSTEM DEVELOPMENT LIFE CYCLE, SDLC)

Sistem geliştirme yaşam döngüsü (SDLC) bilgi sistemlerinin planlanması, kontrollü bir şekilde geliştirilmesi, test edilmesi ve uygulanmasına ilişkin süreçtir. Sistemlerin zamanla karmaşıklaşması ve iş ihtiyaçlarının değişmesi sonucunda sistem geliştirme metodolojileri yaygın bir şekilde kullanılmaktadır.

2.1. Sistem Geliştirme Süreçleri

Sistem geliştirme süreçlerinin her aşaması bir sonraki aşama için temel teşkil etmektedir. Bu sayede iş uygulamalarına ilişkin yürütülen geliştirme aşamaları daha etkin bir yönetim kontrolü sağlamaktadır. İş ve sistemlerin entegrasyonu yönetişimin en önemli unsurlarındandır. İş uygulaması geliştirme süreci, üretime alma ve yaygınlaştırma, bakım faaliyetleri ve ömrünü tamamlamış sistemlerin üretim ortamından kaldırılması aşamalarını içermektedir.

Bir iş uygulaması geliştirme projesi, aşağıdaki durumlardan biri veya birkaçı sonucunda başlatılır:

- Yeni/mevcut iş süreciyle ilgili oluşan yeni fırsat,

- Mevcut teknolojiye ilişkin bir sorun,

- İş uygulamalarının iş ortakları/endüstri standardı sistemler ve ilgili ara yüzlerle uyumlu hale getirilmesi,

- Mevcut iş süreciyle ilgili gelişen sorun,

- İşletmenin teknolojiden faydalanmasını sağlayacak yeni bir fırsat.

İş uygulamaları geliştirilmesi iki farklı kategoride değerlendirilmektedir:

1) Organizasyon Odaklı (Merkezli): Organizasyon odaklı uygulamaların temel amacı, bilinmesi gerekenler ilkesine yönelik kullanıcıların ve destek fonksiyonlarına ilişkin bilgilerin konsolide edilmesi, depolanıp arşivlenmesi ve paylaşılmasıdır. Çoğunlukla bu tür projelerde geliştirme için SDLC veya detaylı yazılım mühendisliği yaklaşımları da kullanılmaktadır.

2) Son Kullanıcı Odaklı (Merkezli): Son kullanıcı odaklı uygulamaların amacı ise, performansın optimize edilmesini sağlamak için farklı veri görünümleri sağlamaktır. Bu amaç, karar destek sistemlerini (DSS), coğrafi bilgi sistemlerini ve ilgili teknikleri vb. içermektedir. Bu tür uygulamaların çoğu alternatif yaklaşımlar kullanılarak geliştirilmektedir.

SDLC modelleri aşağıdaki 3 başlıkta verilebilir.

1) Geleneksel Şelale (Waterfall) Modeli

Şelale modeli proje yönetim süreci; sıralı ve doğrusal bir proje yönetim süreci olup, BT projeleri ve yazılım mühendisliği için SDLC'nin en popüler versiyonudur. Kullanımı karmaşık olmayan projeler için daha uygundur. Geleneksel bir yöntemdir. Süreçler tıpkı bir şelale gibi yukarıdan aşağıya doğrusal olarak işlemektedir. Analiz, tasarım, yazılım, test, yayın gibi fazlardan oluşur. Bir faz tamamlanıp yenisine geçildiğinde, bir önceki faza geri dönülmez. Söz konusu aşamanın yeniden gözden geçirilmesi ancak birinci aşamadan başlanmasıdır. Bir aşamanın çıktısı bir sonraki aşamanın girdisidir. Proje sonunda elde edilen ürün müşteriye teslim edilmektedir.

Söz konusu model potansiyel hataların yalnızca nihai kabul testi öncesinde düzeltilmesini sağlayan bir yaşam döngüsü doğrulama yaklaşımını kapsamaktadır. Bu durum projenin gereksinimlerinin net ve iyi tanımlanmış olması durumunda son derece efektiftir. Ancak iletişimin tek yönlü olması nedeniyle gereksinimlerin belirlenmesi veya analiz aşamasında müşteri ihtiyacının net olarak anlaşılmaması hatalı geliştirmelere neden olabilecektir. Bu sebeple müşteri tamamlanan yazılım sistemini tüm artı ve eksileriyle kabullenmek ve kullanmak zorundadır. Geleneksel yaklaşım, gereksinimlerin ve tasarımın tamamlanmasına yardımcı olmak için ekran prototiplerinin gerekli olduğu web uygulamalarında kullanışlıdır.

Proje yönetim modelleri genelde şelale modeli ile başlar. Yazılım mühendisliğinde önemli bir modeldir. Şelale modeli çok daha statiktir. Geleneksel yaklaşımın en önemli avantajı, iş, tasarım ve programlama gereksinimleri için bir şablon sağlamasıdır.

Bu yaklaşımın avantajları aşağıdaki gibidir:

- Projenin başlangıcında gereksinimlerin açık, anlaşılır ve tam olarak tanımlanması, daha doğru zaman ve maliyet tahmini yapılmasını sağlayarak projedeki belirsizlikleri azaltır.

- Proje yönetim ve ekip rollerinin belli olması; kimin ne zaman ne yapması gerektiği katı bir şekilde tanımlandığı ve takip edildiği için kendi başına iş yapma yeterliliği ve tecrübesi olmayan proje elemanları, proje yöneticileri ve performansı iniş çıkışlı proje ekipleri için idealdir.

- Bir safhaya başlanmadan önce bir önceki safha tamamlandığı için proje ilerleyişinin takibi kolaydır.

- Kullanımı ve yönetimi kolaydır.

- Projede müşteri kaynak ihtiyacı daha azdır.

- Müşterinin projeye ne zaman ve ne kadar katkı yapacağı bellidir ve proje boyunca zaman ayırması beklenmez. Fonksiyonel organizasyonlarda birden fazla projede görev alan proje elemanlarının işlerinin üst üste binmesini engeller. Örneğin analiz çalışması tamamlandıktan sonra başka bir projede görev almaya devam eden bir elemanın geri dönüşü gerekmez.

- İşletmenin klasik yöntemlere aşina olması nedeniyle genel olarak proje teknikleri ile ilgili olarak eğitim ihtiyacının agile tekniklere göre daha azdır.

Bu yaklaşımın dezavantajları aşağıdaki gibidir:

- Müşterinin ürünü projenin sonlarına doğru, geliştirme aşamasından sonra gördüğünde çok sayıda değişiklik isteğinin çıkabilmektedir.

- Değişiklik, geliştirme veya düzeltme amaçlı geriye dönüşler şelalede akıntıya karşı yüzmeye benzetilir, zor ve maliyetlidir. Bu yüzden değişikliklere cevap veremez ve süreç içerisinde gelen istekler çoğunlukla kabul edilmez. Bu nedenle müşteri ihtiyaçlarına cevap verilmediği ölçüde projede kapsamın kayması maliyet artışı getireceği ve süreyi uzatacağı için başarısızlık olasılığı yükselebilmektedir.

- Yoğun ve güncel dokümantasyon zaman ve emek gerektiren bir faaliyettir ve genel olarak geliştiriciler tarafından gereksiz bir yük olarak görülür.

- Ürün, kullanıcılar tarafından yeterince sahiplenilmediği için kullanımında direnç gösterilebilmektedir.

- Bir safhanın tamamlanmadan sonraki safhanın başlamaması nedeniyle, tüm proje elemanlarını ilgilendirmeyen gecikmeler dahi bütün proje elemanları için zaman israfına neden olabilmektedir.

- Yazılım geliştirme faaliyetleri yaratıcılık gerektiren faaliyetlerdir ve şelale modelinin yüksek disiplinli yaklaşımı yaratıcılığı olumsuz etkileyebilmektedir.

2) V-Şekil Modeli

Adımlar V şeklini oluşturduğu için V model ya da V şekil model denilmiştir. Şelale modelinde olduğu gibi katı bir disipline sahip olan bu modelde de bir sonraki aşamanın başlangıcı, önceki aşamanın bitişi ile mümkündür. Ancak şelale modelinden farklı olarak proje basamakları lineer olarak aşağıya doğru gitmek yerine, her modelin eşleniği olarak bir test evresini barındırmak sureti ile V şeklini çizen bir şemaya sahiptir. Model, ilk önce şelale modelinde olduğu planlama evresi ile başlar ve şelale modelindeki adımları kodlama aşamasına kadar aynı şekilde takip eder. Ancak kodlama aşamasından sonra yukarıya doğru yönelen lineer gelişim evresi, çoğunlukla kontrol ve test süreçlerini kapsayan ve test süreçleri sonucunda uygulamaya geçen bir süreci temsil eder. Her test süreci, aslında gelişim evresinde karşılık bulduğu adımın bir testi niteliğindedir.

Bu model, aslında şelale modeline benzerdir. Planlama, ihtiyaç, belirleme, üst seviye ve alt seviye tasarımlar vardır. Ancak bazı farkları bulunmaktadır. Üst seviye de daha genel bakış açısına göre

tasarım yapılır. Üst seviye tasarımların daha sonra daha detaylı alt seviye tasarımlarının yapıldığı görülür. Daha alt seviyelerinin açıldığı, girdi-çıktıları ve beklentilerinin yazıldığı söylenebilir. Sonrasında kodlamaya geçilir. Buraya kadar şelale modeli ile benzerlikler söz konusudur. Aşağı doğru akan bir proje yönetimi söz konusudur. Buradan sonra yukarı doğru çıkılmaya başlanır. Birim testleri yapılır. Her üretilen modülün testi yapılır. Daha sonra bu modüllerin/alt seviye tasarım ürünlerinin birbiri ile çalışma durumu, entegrasyon testleri yapılır ve kabul testleri aşamasında müşterinin kabul edip etmemesi ile ilgili müşteriye gidilir. Müşteri gereksinimler kapsamında uygulamayı test eder. Sonra bakım başlar. Bu da proje geliştirme modeli olarak V şekil modeli diğer modellere göre daha güvenilir hale getirir. Ancak bu güvenilirlik, proje oluşumunda yapılan araştırmaların derinliğine ve kalitesine bağlı kalır.

Önceden öngörülemeyen risklerin sonraki aşamalarda probleme dönüşümü, bu proje modeli için en temel risk unsurunu oluşturur. Bu risk unsurunun nedeni olan kısıtlı esneklik yapısı, Çevik model ile karşılaştırıldığında bu model için en temel zayıf noktadır. Bu nedenle iyi planlanmış büyük sistemler veya planlanması çok karmaşık olmayan küçük sistemler için tercih edilebilen bir yöntem olmasına rağmen, her zaman öngörülemeyen risk unsurunu barındırır. Bunun yanında Çevik modeli ile benzer olan bir yanı mevcuttur. Bazı durumlarda Çevik modelin testleri, tasarımın oluşturulma aşamasında belirlenebilir. Bu gibi çevik model projeleri, testlerin oluşturulma zamanları ile V şekil modele benzerlik göstermektedirler (Radack, 2009).

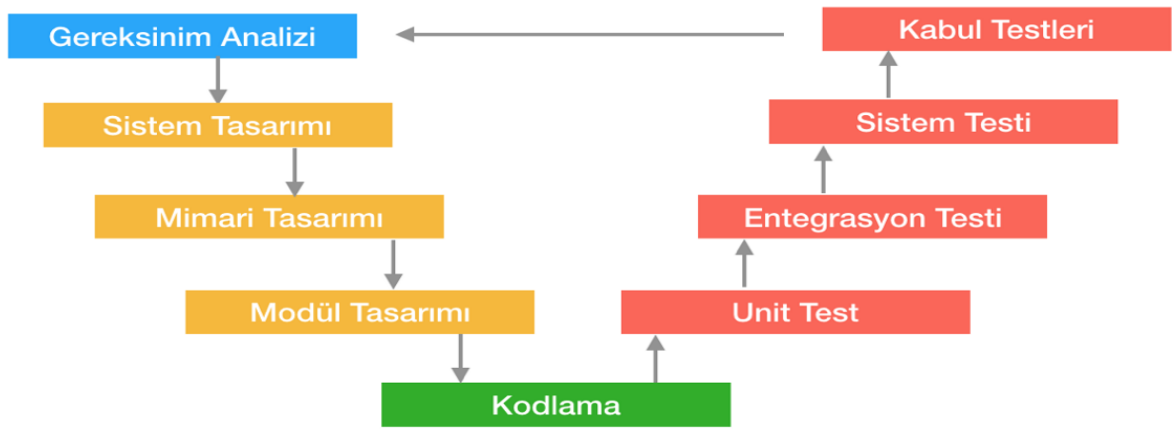
Bilgi sistemleri denetçisinin bakış açısından, V şekil modeli aşağıdaki avantajları sağlar:

- Bilgi sistemleri denetçisinin etkisi, iş uygulaması yaşam döngüsündeki her bir aşamayı ve katılımının kapsamını tanımlayan resmi prosedürler ve talimatlar olduğunda önemli ölçüde artar.

- Bilgi sistemleri denetçisi, iş uygulamasının geliştirme aşamaları boyunca uygulanan yöntem ve tekniklere ilişkin değerlendirme sağlayabilir.

- Bilgi sistemleri denetçisi, sistem geliştirme projesinin tüm ilgili alanlarını ve aşamalarını gözden geçirerek planlanan hedef ve organizasyonel prosedürlere bağlılık konusunda yönetime bağımsız olarak rapor verebilir.

- Bilgi sistemleri denetçisi, sistemin seçilmiş kısımlarını belirleyerek beceri ve yeteneklerine yönelik teknik yönlerini değerlendirebilir.



V Şekil Model Şeması

3) Yinelemeli (İteratif) Model

Yinelemeli model, şelale modelinde görülen esnek olmayan proje disiplini yüzünden oluşan bir ihtiyaç sonucu ortaya çıkmış modeller arasındadır. Bu modelde, projenin müşteriye gösterilmesi ve dağıtımının gerçekleşmesi için projenin tamamlanması gerekmez. Bunun yerine proje küçük parçalara bölünür ve bu parçalar sonuca ulaştırılmak üzere geliştirilmeye başlanır. Projenin tanımı daha çok projenin geliştirilmesi aşamasında gerçekleşmektedir. Bu model bir nevi şelale ve çevik modellerinin bir karışımı olarak değerlendirilebilir.

Modelde bir döngü söz konusudur. Planlama aşaması ile başlar. Planlamalardan geçip bir döngüye girilir. İstek analizleri, analizler, tasarımlar yapılır ve uygulamaya geçilir. Uygulamadan sonra iki durum söz konusudur. Birincisi canlıya geçiş durumudur. Deployment edilir ve tabi problemler her aşamada olduğu gibi bu aşamada da yaşanır. Ama bu problemler testlerin birer parçası olarak görülebilir. Bir yandan da testler devam eder. Bu problemler sistemi besler. Testlerden ve canlıdan gelen bilgilere göre tekrar değerlendirilmeye tabii tutulur. Sonra tekrar istek analizler, canlıya bir bilgi verilmesi ve tekrar testler gibi suren bir döngü vardır. Bu sırada planlamada değişiklikler olabilir. Her yineleme sırasında, geliştirme süreci gereksinimlerden teste kadar her aşamadan geçer ve sonraki her döngü, süreci aşamalı olarak iyileştirir.

Bu model bütün bir projeyi geri dönülmez bir disiplin içerisinde geliştirmek yerine, parçalara ayırarak şelale modelini parçalarda uygulamayı tercih eder. Böylece şelale modelindeki öngörülemeyen hatalar riskini düşürmüş olur. Ancak bu yönü ile de esnekliğini kaybederek çevik modelden ayrılır. Talepleri toplanmış bir proje üç kısma bölünerek kurgulanır. Bu ayrımdan sonra ikinci aşama olan geliştirme aşamasına geçilir. Geliştirme aşamasında oluşum olarak ayrılan her kısım ayrı ayrı geliştirilerek testlere tabi tutulur. Testlerden sonra gerekli görülmesi halinde eksiklikler, uyarlama ve bakımlar ile sağlanır ve müşteriye teslim gerçekleşir. Yapı bakımından esneklik ve ihtiyaçlara sonradan cevap verebilme imkânına sahip olan bu proje modelinde yaşanan sorun, şelale modelinde olduğu gibi, ayrılmış kısımların tesliminde oluşacak problemlerde geri dönüş zorluğudur. Ancak bu zorluk bütün projenin yalnızca bir kısmına hitap ettiği için getireceği masraf klasik şelale modelinden çok daha az olacaktır. Bu özelliği ile riski ve maliyeti düşürebilse de, projeler için çevik modelde olduğu kadar geniş bir esneklik sağlamamaktadır. Sahip olduğu bu özelliği nedeni ile modelde eksik olan kısım daha sonra artırımlı model ile giderilmeye çalışılmıştır (Radack ve ark., 2009).

2.1.1. Sistem Geliştirme Yaşam Döngüsü Yaklaşımı

SDLC yaklaşımı her biri farklı aktiviteler içeren ve birbirlerini izleyen aşamalardan oluşur. Her aşama için amaç ve hedefler, ulaşılmak istenen sonuçlar ve bu kapsamda belirlenmiş sorumluluklar vardır. SDLC yeni bir proje kapsamında işletme içi geliştirme yapmak için kullanılabildiği gibi üçüncü partiden sistem satın alındığı zaman da kullanılabilir. Bu durumda izlenecek fazlar da değişir.

SDLC'nin başarı faktörleri aşağıdakileri içerir:

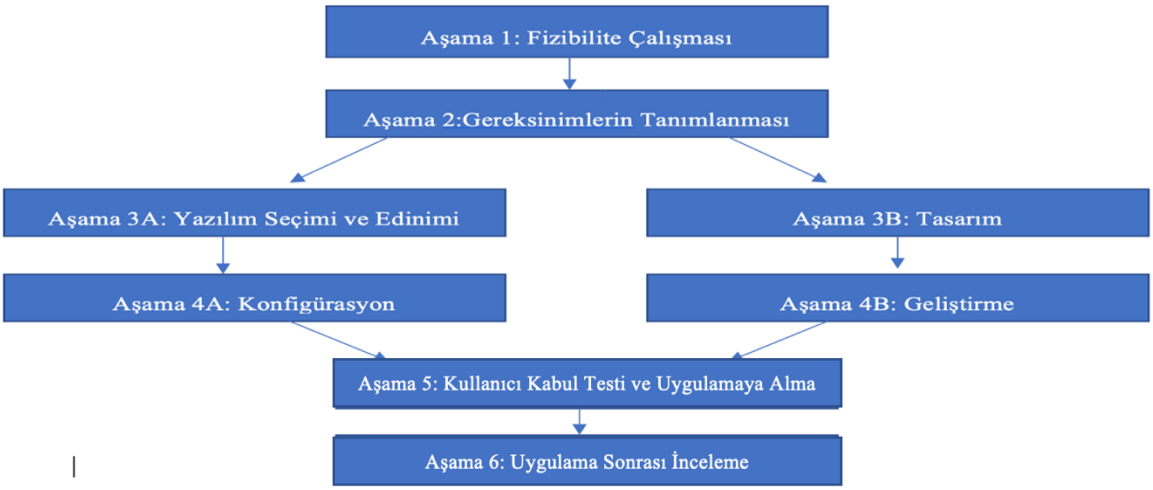
- Müşteriye sunulan hizmetin seviyesi,
- Kalitesi,
- İşin verimi,
- Sağladığı ekonomik değer.

2.1.2. Sistem Geliştirme Yaşam Döngüsü Yaklaşımının Aşamaları

SDLC kapsamında her bir aşama, bu aşamaların amacı, süreçteki diğer aşamalar ile ilişkisi, aşama kapsamında gerçekleştirilecek faaliyetler tamamlandığında beklenen sonuçlar açısından detaylandırılmıştır.

SDLC; bilgisayar endüstrisinde, yeni yazılım ve donanımların oluşturulmasını, oluşturulan yazılım ve donanımların kullanılmasını, tamamlanan ürünlerin elden çıkarılması sürecini tanımlar, düzenler ve yönetilmesini sağlar.

SDLC'nin bir fizibilite çalışmasıyla başlayan ve gereksinimlerin tanımlanması, tasarımı, geliştirilmesi, uygulanması ve uygulama sonrası boyunca ilerleyen yazılım geliştirmeye yönelik sistematik, sıralı bir yaklaşım ile devam eder. Ayrıca, bilgi sistemleri denetçisi, SDLC aşamalarını iyi bir şekilde bilmeli ve fizibilite çalışmasını detaylı olarak değerlendirebilmelidir.



SDCL'nin Aşamaları Şeması

Aşama 1-Fizibilite Çalışması:

Fizibilite çalışması, bir proje yapmaya karar verme süreci boyunca hem teknik hem finansal açılardan gerekli araştırmalar ile değerlendirme yapılmasıdır. Olası problemler ve tehditler araştırılarak yatırımın istenilen hedefe ulaşıp ulaşamayacağı, kar getirip getiremeyeceği öngörülmeye çalışılır. Fizibilite çalışması ile sistemin üretkenlik kazanımlarında veya gelecekteki maliyetlerden kaçınmada uygulanmasının stratejik faydalarını belirlemek, yeni bir sistem maliyet tasarruflarını ve miktarını belirlemek ve sistem uygulanmasında ortaya çıkan maliyetler için bir geri ödeme planı tahmin etmek. Ayrıca iş kullanıcılarının hazır bulunmaları ve iş süreçlerinin olgunluğu gibi somut olmayan faktörleri de göz önüne alacak ve değerlendirilecektir. Bu iş olurluk incelemesi bir sonraki aşamaya geçmenin gerekçesini sağlar.

Fizibilite çalışmasında aşağıdaki aşamaların gerçekleştirilmesi önemlidir:

- İşletmenin hedefleri ve eğer sorun ile ilgili ise sorunun incelenmesi ve kapsamının belirlenmesi,

- Proje kapsamının tanımlama,

- Mevcut durum analizi yapma,

- Paydaş ihtiyaçlarına dayalı gereksinimleri belirleme,

- Önerilen bir yaklaşım sağlama,

- Yaklaşımın maliyet etkinliğini değerlendirme,

- Paydaşlarla resmi bir gözden geçirme yapma.

Fizibilite çalışması sırasında yapılacak hususlar:

- Çözüme ulaşmak için gereken zaman çerçevesi belirlenir.

- İş gereksinimlerinin karşılanabilmesi için en iyi risk tabanlı çözüm belirlenir. Sonuçta süreç kolayca SDLC ya da hızlı uygulama geliştirme/(RAD) ile eşleştirilebilir.

- Mevcut sistemde küçük bir değişiklik ya da geçici çözüm ile çözüm bulunup bulunamayacağı belirlenir.

- Üçünü partiden alınacak bir ürünün çözüm bulup bulmayacağı belirlenir.

- Çözüm için gerekli maliyet belirlenir.

- Çözümün iş stratejisine uyumu değerlendirilir.

Fizibilite çalışması sırasında, bilgi sistemleri denetçisi aşağıdakileri yapmalıdır:

- İhtiyacın kritikliğini belirlemeli ve tespit etmeli,

- Mevcut sistemlerle bir çözüm elde edilip edilmeyeceğinin değerlendirilip mümkün değilse, alternatif çözümlerin makul olup olmadığının değerlendirildiğini gözden geçirmeli,

- Tüm aşamalardaki çıktıları gözden geçirip uygunluğunu değerlendirmeli,

- Proje kapsamında maliyetlere ilişkin gerekçelerinin/faydalarının gerçeği yansıtıp yansıtmadığını ve beklenen maliyetleri ve beklenen faydaları gösterip göstermediklerini değerlendirip güvence verebilmeli,

- Tercih edilen çözümün uygunluğunu değerlendirmelidir.

İş Olurluğu (Business Case)

Bir iş olurluğu sonucunda, bir işletmenin bir projenin devam edip etmemesi gerektiğine karar vermesi için gereken bilgiler sağlanır. Bir projedeki tamam ya da devam kararının alındığı adımdır. Maliyetlerin ve iş faydalarının karşılaştırılması sonucunda bir projenin kurulması veya sürdürülmesi için gerekçe sağlar. Uygun olmayan projeler bu aşamada sonlandırılabilir.

Aşama 2- Gereksinimlerin Tanımlanması:

Gereksinim analizi, projede hayati bir rol oynar. Çünkü projenin başarısı ve başarısızlığı gereksinim analizine bağlıdır. Projeye özel ihtiyacın ve çözüm gerektiren sorunun gereksinimler ile tanımlanır. Bu tanımlama sırasında kullanıcının katılımı çok önemlidir. Gerek özelleşmiş bir yaklaşım gerekse tanımlanmış ve belgelenmiş bir tedarik sürecini takiben satıcı tarafından sağlanan yazılım paketinde olsun kullanıcıların aktif olarak dahil olması beklenir.

Gereksinim tanımları, bir sistemin ne yapması gerektiği, kullanıcıların bir sistemle nasıl etkileşime gireceği, sistemin hangi koşullarda işleyeceği ve sistemin yerine getirmesi gereken bilgi kriterleri hakkında açıklamalar içermelidir.

Gereksinim tanımlarını başarıyla tamamlamak için proje ekibi aşağıdaki gibi görevleri yerine getirecektir:

- Paydaşları belirlemek,

- Gereksinimleri yapısal bir formatta kaydetmek ve paydaşlara danışmak,

- Gereksinimlerin eksiksiz, tutarlı, açık, doğrulanabilir, değiştirilebilir, test edilebilir ve izlenebilir olduğunu doğrulamak,

- Gereksinimleri ile beklentiler arasındaki farkları tespit etmek ve düzeltmek,

- Sistemsel sınırlamaları belirlemek,

- İlgili güvenlik gereksinimlerini belirlemek,

- Kullanıcı gereksinimlerini sistemsel gereksinimlere dönüştürmek (Bu aşamada prototip bir kullanıcı arayüzü bu aşamada hazırlanabilir.),

- Gereksinimlerin tarihsel bir şekilde saklanabilmesini sağlamak (Bu amaçla hazırlanmış birçok paket program temin edilebilir.),

- Gereksinimler kapsamında herhangi bir kısıtlamayı belirlemek,

- Gereksinimlerin eksiksiz, tutarlı açık doğrulanabilir, değiştirilebilir ve izlenebilir olduğunu doğrulamak (Bu şekilde etkin bir iş olurluğu takibi yapılabilir.),

- Proje paydaşları arasındaki uyuşmazlıkları (çatışmaları) çözmek,

- Gereksinimler ve kaynaklar arasındaki uyumsuzlukları çözmek.

Bu adımda, iş analizi, teknik analiz ve güvenlik analizi yapılır. Bu analizlerin kapsamı kısaca aşağıda açıklanmaktadır.

İş Analizi

İş gereksinimleri analizi olarak da adlandırılır. İş ve BT arasında bir köprü olacak şekilde konumlandırılır. İşin sınırlarının kesin olarak belirlendiği süreçtir.

Teknik Analiz

Teknik gereksinimler analizi olarak da değerlendirilir. Teknik dizayn gereksinimleri bu aşamada ele alınır. İş gereksinimleri analizine göre teknik ihtiyaçların belirlendiği süreçtir.

Güvenlik Analizi

Güvenlik gereksinimleri analizi olarak da ifade edilmektedir. Proje boyunca güvenlik kapsamında güvence verebilmek için gereklilikler bu kapsamda değerlendirilir.

Tipik olarak, bilgi sistemleri denetçisi, aşağıdaki hususların gereksinim analizi kapsamında yeterliliğini gözden geçirmelidir:

- Çözümün düzenleyici, yasalara ve yasal gerekliliklere uygun olduğunu,

- Denetim izlerinin yeterli açıklıkta sistemin bir parçası olarak yer alıp almadığını (sorunların bu denetim izleri ile takip edilmesine imkân verilip vermeyeceği),

- Proje dahilinde veri oluşturma işleme, depolama, iletim ve imha süreçlerine kadar verilerin geçerliliği gizliliği bütünlüğü ve erişilebilirliği ile ilgili kontrollerin mevcut olduğunu,

- İşletmenin sistem güvenliği ekibinin, iş uygulaması içindeki veri yaşam döngüsü boyunca güvenlik kontrollerinin geliştirilmesinde ne derece yer aldığını.

Bu aşamada, genel olarak aşağıdaki adımlar gerçekleştirilir:

- Sistemin fonksiyonel gereksinimleri, müşteri gereksinimleri, geliştirilecek sistem ile ilgili standartlar, yönetmelikler ile oluşan gereksinimler dokümante edilir.

- Müşteri gereksinimleri ile sistem arasındaki izlenebilirlik oluşturulur.

- Gereksinimlerin doğrulanması ve öncelik nitelikleri belirlenir.

- Kalite gereksinimleri belirlenir.

- Gereksinimler gözden geçirme sürecine uygun olarak gözden geçirilir.

Aşama 3A- Yazılım Seçimi ve Edinimi (Satın Alınan Sistemler):

Yazılım edinimi aslında SDLC'de bir aşama olarak değerlendirilmez. Yazılım geliştirme yerine edinme kararına varılırsa, gereksinimlerin belirlenmesinden sonra bu süreç izlenir. Bu süreçte bir satıcıya bağlı olmak yerine yeni sistem üzerindeki tam sahiplik ve kontrol sahibi olmanın faydaları da düşünülerek yazılımın temin edilmesi ön planda tutulur.

Gereksinimlerin tamamlanması sonrasında bu gereksinimleri karşılamak için, tedarikçilerden veya çözüm sağlayıcılardan sağlanacak sistemler için, potansiyel tedarikçilerden teklif almak amacı ile işletme gereksinimlerini özetleyen bir teklif talebi oluşturulması gerekir.

Bu aşamada en önemli unsur, bir teklif talebi (RFP) hazırlamak ile başlar. Bir teklif talebinde aşağıdaki hususların belirtilmesi önemlidir.

- Ürün – sistem gereksinimleri,

- Ürün ölçeklenebilirliği ve birlikte çalışabilirliği,

- Müşteri referansları,

- Firma yaşayabilirliği/finansal istikrarı,

- Tam ve güvenilir dokümantasyon varlığı,

- Firma desteği,

- Kaynak kod mevcudiyeti,

- Teklif edilen ürün üzerine yıl deneyimi,

- Tarihleri ile birlikte yeni ve planlanmış iyileştirme listesi,

- Mevcut kullanıcı listesi ile ürünü kullanan müşteri sayısı,

- Ürünün kabul testi.

Aşama 3B- Tasarım (İşletme İçi Geliştirme):

İşletme içinde yapılacak geliştirmeler için tasarım süreci gerçekleştirilir. Bu süreçte tamamlanmış olan gereksinimlere dayanarak, işletme içi geliştirmelerin gereği olarak sistem bölümlerini arayüzlerini ve seçilen donanım, yazılım ve ağ olanaklarını kullanarak sistemin nasıl uygulanacağını tanımlayan bir sistem ve alt sistem spesifikasyonları için bir taban çizgisi oluşturulur. Tasarım genellikle program ve veri tabanı özelliklerini de içerir ve gerekli tüm güvenlik hususlarını ele alır. Bununla beraber, yeni gereksinimlerin kontrolsüz bir şekilde geliştirme sürecine girmesini engellemek için değişim kontrol süreci oluşturulur.

Bu aşamada yazılım geliştirme ile ilişkili risklerinde göz önüne alınması gerekir. Bu riskler başlıca aşağıdaki kategorilerde değerlendirilir.

Stratejik Risk

Kurumsal stratejiler dikkate alınmadan iş amaçları tanımlandığında ve önceliklendirildiği zaman ortaya çıkar.

İş Riski

Oluşturulan sistemin kullanıcı ihtiyaçlarına cevap verememesi, gereksinimlerini karşılayamaması riski.

Proje Riski

Proje için planlanan finansal kaynakların planlananı aşması sonucu projenin öngörülen zaman dilimi içinde tamamlanamama riski.

Bilgi sistemleri denetçisi, sadece bir SDLC yaklaşımı izlemenin, bir geliştirme projesinin başarılı bir şekilde tamamlanmasını sağlamadığını bilmelidir. Bilgi sistemleri denetimi kapsamında aşağıdaki hususlarında değerlendirilmesi gerekmektedir.

- Yönetim tarafından yazılım tasarım ve geliştirme faaliyetlerinin takip edildiği ve bu faaliyetlere destek verildiği,

- Proje aşamalarında periyodik gözden geçirildiği ve risk analizinin yapıldığı,

- Projenin amaç ve hedefi sağlayıp sağlamadığı,

- Kaynak bütçe ve zamanlama planlamasının doğru bir şekilde yapıldığı,

- Planlamalardan sapmaların kontrol edilip edilmediği imkânsız sapmaların önüne geçmek amacıyla yazılım için bir taban çizgisinin oluşturulup oluşturulmadığı.

Yapısal analiz, tasarım ve geliştirme tekniklerinin kullanımı klasik bir SDLC için vazgeçilmezdir. Bilgi sistemleri denetçisi de özellikle geleneksel bir SDLC yaklaşımı kullanırken detaylı bir şekilde tanımlandığını, belgelendiğini ve takip edildiğini değerlendirmelidir.

Ayrıca genel bir ön tasarımın oluşturulmasında varlık ilişkisi diyagramlarının kullanılması son derece önemlidir.

Yazılım tasarım aşamasında yazılım taban çizgisi (tasarım dondurma) de oluşturulur. Yazılım taban çizgisi tasarımdaki kesme noktası anlamına gelir. Kullanıcı gereksinimleri zaman, fayda etki ve maliyet açısından değerlendirilerek belirlenmiş noktadır. Sistem gereksinimlerini taban çizgisi üzerinden yönetememek proje kapsamında risklerin gerçekleşmesine neden olur.

Tasarımın başarılı olması için kısıtlı da olsa kullanıcıların katılımı önemlidir. Katılım kısıtlıdır, çünkü kullanıcılardan teknik detayda bilgi sahibi olmaları beklenmez.

Tasarımda bilgi sistemleri denetçisi rolü aşağıdaki gibi özetlenebilir:

- Kalite güvence sonuçlarının değerlendirilmesi,

- Risk değerlendirmelerinin tamlığı bütünlüğü ve doğruluğuna güvence verilmesi,

- Sistem girdi ve çıktılarının incelenmesi,

- Sistem akış diyagramlarının değerlendirilmesi,

- Denetim izleri yeterliliğine güvence verilmesi,

- Hesaplamaların ve işlemde bütünlüğe güvence verilmesi,

- Hatalı verilerin sistem tarafından doğru ve eksiksiz bir şekilde tespit edilip ve işlenebildiğine güvence verilmesi.

Aşama 4A- Yapılandırma, Konfigüre Etme (Satın Alınan Sistemler):

Bir paket program sisteminin temin edildikten sonra işletmenin gereksinimlerine göre uyarlaması için yapılandırılması gerekmektedir. Bu amaçla genel olarak programlarda yazılımsal değişiklikler yapmak yerine genel olarak sistem kontrol parametreleri kullanılarak sistemin ihtiyaca uygun şekilde kontrol edilmesi sağlanır. Günümüzün yazılım paketleri son derece esnek oldukları için paketin işlevselliğini ilgili parametreleri açıp kapatarak veya tablolardaki parametreleri ayarlayarak işletmeye adapte edilmesi sağlanır. Eğer sadece parametre ayarları yeterli olmuyorsa bu aşamada ayrıca satın alınan sistemin mevcut programlara ve veri tabanlarına bağlanacak arayüz programlarının da geliştirmesi gerekebilir. Bu aşamada iş surecilerinde istenmeyen sonuçları en aza indirmek için BT sistemlerindeki değişiklikler dikkatlice değerlendirilmeli, planlanmalı, test edilmeli, onaylanmalı, belgelenmeli ve anlatılmalıdır.

Bilgi sistemleri denetçisinin, geliştirme personeli ile üretim ortamı arasında görevler ayrılığını (SoD) sağlamak için konfigürasyon, değişim ve sürüm yönetimi ve mevcut kontrollerin yönetimi için mevcut araçların farkında olması gerekir.

Konfigürasyon yönetim araçları, aşağıdakileri kullanarak değişiklik yönetimini ve sürüm yönetimini destekler:

- Fonksiyonel, operasyonel ve güvenlik açısından etki değerlendirmesi yapılabilmesini teminen önerilen bir değişiklikten etkilenen ögelerin belirlenmesi,

- Değişikliklerin yetkilendirme kayıtlarına uygun olarak uygulanması,

- İnsan hatalarından ve kaynak maliyetlerinden kaçınmak için genel bir sürüm hazırlanması,

- Yetkili değişikliklerden etkilenen yapılandırma ögelerinin doğru olarak belirlenip, sürümler devreye alındığında yapılandırma öğesindeki tüm değişikliklerinin kaydedilmesi,

- Uygulanan bir değişiklik başarısız olursa bir işletmenin geri döneceği sürümlerle ilgili (öncesinde doğru olarak çalıştığı bilinen) referans noktasının kaydedilmesi.

Aşama 4B- Geliştirme (İşletme İçi Geliştirme):

Geliştirme, proje sahibinin talepleri doğrusunda yapılmış olan tasarıma uygun yazılım geliştirme araçlarıyla vasıtasıyla ürüne dönüştürme işlemidir. Bu aşamada veri tabanının oluşturulması, kodlamanın yapılması, kullanıcı ara yüzlerinin oluşturulması ve raporların yazılması gibi işlemler yapılır. Projeye ait ilk çıktılar ortaya çıkmaya başlar. Geliştirilen modüllerle ilgili birim testleri ve sistem testleri de bu aşamada yapılabilir.

Programda daha kaliteli bir sonuç elde etmek ve bakımların etkin ve efektif bir şekilde yapılabilmesini teminen kodlama standartları uygulanmaktadır. Bu durum kodların daha basit yazılmasını okunmasını ve anlanmasına imkân vermektedir.

Programların daha kolay yazılabilmesini teminen sadece programlama standartları değil çevirim için programlama olanakları da değerlendirilmektedir. Çevirim içi programlama olanakları bilgisayar iş

istasyonlarında kullanılır. Çevrim içi sistemler programcının sorun çözme yeteneğini geliştirir ama yetkisiz erişimlerden oluşan güvenlik açıkları oluşturur.

Sistem geliştirme işlemi sırasında birçok programlama hatası tespit edilebilir. Bu kapsamda hata ayıklama araçları kullanılır. Hata ayıklama araçları üç ana kategoriye ayrılır. Bunlar:

1) Mantık yolu monitörleri,

2) Bellek dökümleri,

3) Çıktı analizörleri.

Aşama 5- Kullanıcı Kabul Testi ve Uygulamaya Alma:

Bu aşamada gerçekleştirilen kullanıcı kabul testinin son tekrarı ve takibinde kullanıcıdan alınan onay ile yeni bilgi sisteminin fiili olarak devreye alınarak çalıştırılması sağlanır. Bu süreçte riskleri kabul edilebilir bir seviyeye indirgemesi ve sistemin etkinliği üzerinde yönetimin sorumluluğunun sağlanması, sistemin amaçlanan hedeflerini karşılaması ve uygun bir iç kontrol seviyesinin oluşturulması konusunda iş uygulamasının etkinliğinin değerlendirilmesi için bir spesifikasyon ve akreditasyon sürecinden de geçebilmektedir.

Aşama 6- Uygulama Sonrası İnceleme:

Bakım sureci, yazılım kullanılmaya başlandıktan sonra sistem üzerinde değişiklik yapılması surecidir. Yazılım proje sahibi ve paydaşları tarafından yapılan isteklere göre sürekli değişir. Bu aşamada sisteme yeni eklentiler eklenir ya da hatalar giderilir.

Teslimden sonra değişiklik yapmanın temelde 3 nedeni vardır. Bunlar:

1) Doğrulayıcı bakımlar, sistemde bulunan herhangi bir hatayı düzeltmek için (gereksinim, tasarım, kodlama, dokümantasyon hatası gibi) yapılan bakımlar,

2) Kusursuzlaştırma bakımları ise, değişiklikler ile ürününün etkinliğinin iyileştirilmesi,

3) Ürünün çevresinde yapılan değişikliklere cevap verebilmesi için yapılan değişiklikler.

Yeni ve yaygın olarak geliştirilmiş bir sistemin başarılı bir şekilde oluşturulmasını takiben, fizibilite aşaması için öngörülen değerlerin doğrulanmasına geçilir. Buradaki sapmalar karşısında sistemin yeterliliğini ve öngörülen maliyet-fayda ve yatırım getirisi ölçümlerini değerlendiren resmi bir süreç uygulanmaktadır. Bu yolla, bilgi sistemleri projesi ve son kullanıcı yönetimi, sistem eksiklerini ele almak için öğrenilen dersleri ve gelecekteki projeler için sistem geliştirme ve proje yönetimi için önerileri sağlayabilir.

Bilgi sistemleri denetçisi SDLC sürecini gözden geçirirken, çeşitli aşamalardan belgeler/dokümanlar almalı ve proje geliştirme toplantılarına katılarak sistem geliştirme sürecini gözlemlemelidir. Bilgi sistemleri denetçisi, proje ekibinin söz verilen/taahhüt edilen tarihlere kadar anahtar çıktıları üretme yeteneğini de değerlendirmelidir.

Tipik olarak, bilgi sistemleri denetçisi, aşağıdaki proje yönetimi faaliyetlerinin yeterliliğini gözden geçirmelidir:

- Proje komitesi/kurulu tarafından gözetim seviyeleri,
- Proje dahilindeki risk yönetimi yöntemleri,
- Sorun yönetimi,
- Maliyet yönetimi,
- Planlama ve bağımlılık yönetimi süreçleri,
- Üst yönetime raporlama süreçleri,
- Kontrol süreçlerinin değiştirilmesi,
- Paydaş yönetimi katılımı,

- İmzalama süreci (En azından, proje maliyetinden ve/veya sistemin kullanımından sorumlu sistem geliştirme ve kullanıcı yönetiminden imzalı onaylar).

Ek olarak, SDLC surecinin tüm aşamalarının yeterli ve eksiksiz bir şekilde dokümantasyonu olmalıdır. Tipik dokümantasyon türleri, bunlarla sınırlı olmamak üzere aşağıdakileri içerir:

- Her aşamada neyin gerçekleştirileceğini tanımlayan hedefler,

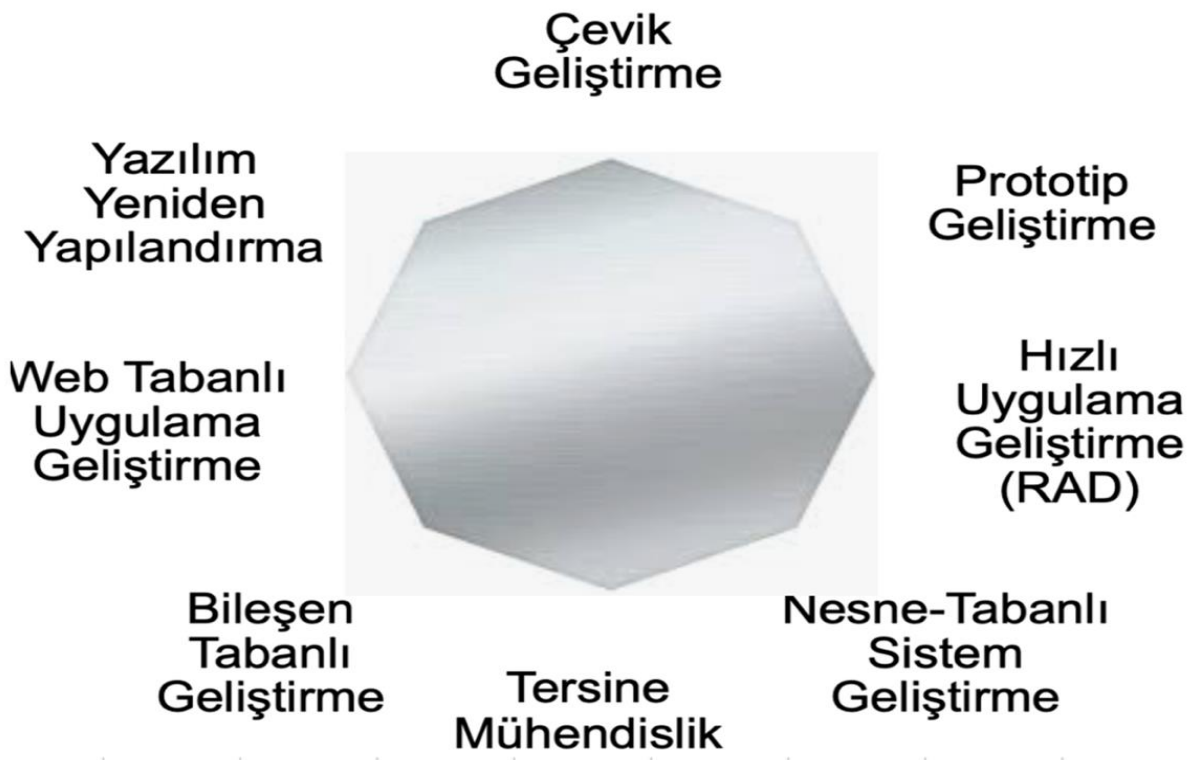
- Aşamalara göre anahtar teslimatlar ile bu teslimatlara proje personelinin atandığı doğrudan sorumluluklar,

- Anahtar teslimatların tamamlanması için vurgulanmış tarihleri olan bir proje takvimi,

- Her aşamanın tamamlanması için gereken kaynakları ve kaynak maliyetlerini tanımlayan ekonomik bir tahmin.

2.1.3. Yazılım Geliştirme Metodolojileri

Yazılım geliştirme metodolojileri aşağıdaki şemada verildiği üzere bünyesinde birçok unsuru barındırır.



Yazılım Geliştirme Metodolojisi Şeması

Çevik Geliştirme:

Çevik geliştirme, karmaşık sistemler geliştirmenin geleneksel olmayan bir yolunu benimseyen benzer geliştirme süreçleri ailesidir. Bu süreçler, geliştirilmekte olan sistemdeki veya geliştirmeyi gerçekleştiren projedeki değişiklikleri esnek bir şekilde ele almak üzere tasarlandıkları için "*çevik*" olarak adlandırılır. Çevik yazılım geliştirme; yinelemeli (iterational) olarak yazılım geliştirme sürecinin aktivitelerinin uygulanması ve tedrici (incremental) olarak yazılım ürününün doğru bir şekilde, değişime açık, Agile (Çevik) Birliği'nin oluşturduğu değerler ışığında prensiplerini uygulayarak geliştirilmesidir.

Çevik yazılım geliştirmenin avantajları:

- Kısa toplantılar ve yüz yüze iletişim felsefesi ile çevik yöntemlerde uygulanan insan ve iletişim merkezli süreçler sayesinde proje elemanlarının moral ve motivasyonu artar, değerli bilgi ve görüşlerin paylaşımı ile üretkenlik seviyeleri yükselir.

- Kısa süreli yinelemeler ve yineleme sonucunda teslim edilen çalışır ürün sayesinde projenin ilerlemesi daha sağlıklı bir şekilde gözlemlenir ve proje riski azalır, muhtemel hatalar daha erken tespit edilir, teslimat öngörülebilirliği artar ve ürün pazara daha çabuk ulaşır.

- Çevik yöntemlerin felsefesi gereği proje süresince değişikliklerin hoş karşılanması ve hatta teşvik edilmesi nedeniyle ve yinelemeli süreç yapısı sayesinde esneklik yüksektir ve değişen şartların yönetilmesi kolaydır.

- Müşteri veya kullanıcıların proje sürecinde proje elemanları ile yüz yüze iletişim kurabilmeleri sayesinde iş süreçleri ile yazılım geliştirme süreçlerinin entegrasyon seviyesi yükselir, istenen özelliklerin tam olarak ifade edilebilmesiyle yazılımın fonksiyonel kalitesi artar ve müşteri memnuniyeti sağlanır.

- Genel olarak üretilen yazılımın sadece ürün özellikleri dokümantasyona çevrilir ve diğer süreçler hariç tutulur. Hafif dokümantasyon iş yükünü azaltır.

- Çevik yöntemler, uygulanan safhalar arası geçişlere ve geri dönüşlere müsait olduğu için süreç devam ederken tespit edilen geriye dönük aksaklıklar kolaylıkla düzeltilebilir.

Çevik yazılım geliştirmenin dezavantajları:

- Değişime açık olması nedeniyle projenin kaynak ve kapsam planlaması zordur.

- Çevik yöntemlerde süreçler iletişim merkezli olduğu için proje elemanlarının birbirleriyle veya müşteriyle aynı ortamda bulunmaması, proje gelişimini olumsuz etkiler. Video konferans gibi uzaktan iletişim kurabilecek teknolojiler kullanılabilir fakat bu da gerekli altyapı ve teknoloji nedeniyle maliyetleri arttırır.

- Alt yüklenicinin teklif verebilmesi ve bir kontrat yapılabilmesi için istenenlerin kesin olması ve değişmemesi gerekir. Çevik yöntemlerin belirsiz ve değişken ortamı alt yüklenici uygulamalarını zorlaştırır.

- Fonksiyonel olarak bütünlük gerektiren büyük ve karmaşık sistemler yinelemeler için uygun parçalara bölünemeyebilir.

- Zaman ayıramama veya isteksiz olma gibi nedenlerle müşteri veya son kullanıcıların proje içerisinde olma oranı düşükse, gereksinimler doğru biçimde tanımlanamayabilir.

- "*Gerektiği kadar*" anlayışı ile hazırlanan dokümantasyonun yetersiz olması durumunda projeye sonradan katılan geliştiriciler ve bakım safhaları olumsuz etkilenir.

- Takımlar kendi kendini organize ettiği için bireylerin yetenekli, tecrübeli ve çevik süreçlere hâkim olması gerekir. Yeteneksiz, tecrübesiz çalışanlar takım performansını düşürerek proje gelişim sürecini olumsuz etkiler.

- Önceliklerini tam olarak belirleyemeyen ve yapılacaklar listesindeki öncelikleri sık sık değiştirmek isteyen müşteriler planlamaları olumsuz etkileyebilir.

- Çevik yöntemlerle, spesifik problemler veya fonksiyonlara göre tanımlanan gereksinimleri karşılayacak yeterlikte çözüm geliştirildiği için, ürünlerin diğer projelerde yeniden kullanılabilirliği olmaz.

- Yazılım hatasının insan hayatını tehlikeye attığı veya büyük maddi zararlar doğurduğu hata toleransı olmayan sistemler için çevik yöntemlerin test ve kalite kontrol yöntemleri yeterli olmayabilir.

- Çevik yöntemlerde müşteri faydası ön planda tutulur fakat bu genel olarak müşterinin istediği fonksiyonların karşılanmasıdır. Yalın geliştirmenin bir kalıtımı olarak basitlik felsefesi uygulandığı için kullanıcı ara yüzü ve kullanıcı deneyimi gibi kullanıcı tatmini ile ilgili konular göz ardı edilebilir.

Prototip Geliştirme (Evrimsel Geliştirme):

Prototip geliştirme, bir tasarımın çeşitli yönlerinin test edilmesi, fikirler veya özelliklerin gösterilmesi ve erken kullanıcı geri bildiriminin toplanması için çalışan bir modeli (prototip) hızla bir araya getirme sürecidir. Prototipleme, tavsiye ve eleştiri için kullanıcıya nihai sistemin bir modelini

temsil etmek için programlanmış simülasyon tekniklerini kullanır. Vurgu, son kullanıcı ekranları ve raporları üzerindedir. Bu sadece bir model olduğu için iç kontroller öncelikli bir öğe değildir.

Prototipleme, bilgi sistemi kurma çalışmasına belirli bir soruna yönelik bir program ile başlamaktır. Temel olarak, prototip, sistem kullanıcılarına, sistemin tamamlanmış halinin ön görünümünü sağlamaktadır.

İşletme bazında düşünülecek olursa, belli bir bölüm seçilmekte ve bu bölüme özgü sorunu çözmek üzere bilgi sistemi oluşturulmaktadır. Bu sistem sadece o bölüme hizmet vermektedir.

İlerleyen süreçte ihtiyaç duyulduğunda diğer bölümlerde de benzeri programlar oluşturulmaya başlanıp, bu programın parçalarının bir araya getirilmesi veya ilk yazılan programın geliştirilmesi ile istenilen bilgi sistemine ulaşılabilmektedir.

Prototip geliştirmenin avantajları:

- Kullanıcıyı Alıştırma: Prototipleme yaklaşımının önde gelen amacı, kullanıcı ihtiyaçlarını büyük ölçüde karşılayan sistemler geliştirilmesidir.

- Hızlı Geliştirme Zamanı: Klasik yaklaşımda, anlamlı sonuçlara ulaşılması ve sistemi faaliyete geçirmek için yıllar gerekebilirken, bu yaklaşımda anlamlı sonuçlar elde edebilmek için kısa bir süre (birkaç hafta veya ay) yeterli olmaktadır.

- Daha Az Hata: Prototipleme yaklaşımı, hataların önceden belirlenmesini sağlamaktadır. Klasik yaklaşımla ise, hatalar ancak yıllar süren geliştirme sürecinin sonunda belirlenebilmektedir.

- Daha Çok Değişim Fırsatı: Prototipleme ile yöneticiler, geliştirilmekte olan her bir alt sistem veya komponentin çıktıları görebilmekte ve bu çıktılarla çalışabilmektedir. Bu sayede, yöneticiler geliştirme süreci içerisinde değişim önerilerinde bulunabilmektedir.

Prototip geliştirmenin dezavantajları:

- Sistem Geliştiriciler ile Yönetim Arasındaki Ortaklık İhtiyacı: Yönetimin aktif olarak projeye katılması, bazı yöneticiler tarafından ayrılmak istenmeyen bir zamana ihtiyaç duyulmasına neden olmaktadır.

- Toplam Geliştirme Maliyetinin Yüksek Olabilmesi: Yöneticilerin sürekli olarak sistemde değişiklik ihtiyacı mevcutsa bu bir dezavantaj olarak ortaya çıkmaktadır.

- Yeni veya Geliştirilmiş Sistemin Bilgisayar Kaynaklarını Etkin Olarak Kullanamaması: Prototipleme yaklaşımında, kullanıcı ihtiyaçlarına çok fazla önem verilirken, çeşitli bileşenlerin kullanımına daha az önem verilmektedir. Sistemlerin daha hızlı ve ucuz hale gelmesiyle, bu donanım limitleri daha önemli bir hal almaktadır.

Hızlı Uygulama Geliştirme (RAD):

Hızlı uygulama geliştirme, iyi tanımlanmış bir metodoloji dahilinde, bir dizi kanıtlanmış uygulama geliştirme tekniğini kullanarak geliştirme maliyetlerini düşürürken ve kaliteyi korurken, işletmelerin stratejik olarak önemli sistemleri daha hızlı geliştirmelerini sağlayan bir metodolojidir.

James Martin'in RAD'ye yaklaşımı süreci dört farklı aşamaya böler:

1- Gereksinim Planlama Aşaması: SDLC'nin sistem planlama ve sistem analiz aşamalarının unsurlarını birleştirir. Kullanıcılar, yöneticiler ve BT personeli, iş ihtiyaçları, proje kapsamı, kısıtlamalar ve sistem gereksinimleri üzerinde tartışır ve anlaşırlar. Ekip, kilit konular üzerinde anlaştığında ve devam etmek için yönetim yetkisini aldığında sona erer.

2- Kullanıcı Tasarımı Aşaması: Bu aşamada kullanıcılar sistem analistleriyle etkileşime girer ve tüm sistem süreçlerini, girdilerini ve çıktılarını temsil eden modeller ve prototipler geliştirir. RAD grupları veya alt grupları, kullanıcı ihtiyaçlarını çalışan modellere dönüştürmek için tipik olarak ortak uygulama geliştirme (JAD) teknikleri ve bilgisayar destekli yazılım mühendisliği (Computer Assisted Software Engineering, CASE) araçlarının bir kombinasyonunu kullanır. Kullanıcı tasarımı, kullanıcıların ihtiyaçlarını karşılayan sistemin çalışan bir modelini anlamalarını, değiştirmelerini ve nihayetinde onaylamalarını sağlayan sürekli etkileşimli bir süreçtir.

3- İnşaat Aşaması: SDLC'ye benzer program ve uygulama geliştirme görevine odaklanır. Ancak RAD'de kullanıcılar katılmaya devam eder ve gerçek ekranlar veya raporlar geliştirildikçe değişiklik veya iyileştirmeler önerebilir. Görevleri programlama ve uygulama geliştirme, kodlama, birim entegrasyonu ve sistem testidir.

4- Geçiş Aşaması: Veri dönüştürme, test etme, yeni sisteme geçiş ve kullanıcı eğitimi dahil olmak üzere SDLC uygulama aşamasındaki son görevlere benzer. Geleneksel yöntemlerle karşılaştırıldığında, tüm süreç sıkıştırılır. Sonuç olarak, yeni sistem çok daha kısa sürede kurulur, teslim edilir ve işletime alınır.

RAD'nin avantajları aşağıdakileri içerir:

- Daha İyi Kalite: Kullanıcıların gelişen prototiplerle etkileşime girmesini sağlayarak, bir RAD projesinden elde edilen iş işlevselliği, genellikle bir şelale modeliyle elde edilenden çok daha yüksek olabilir. Yazılım daha kullanışlı olabilir ve geliştiricileri ilgilendiren teknik sorunlardan ziyade son kullanıcılar için kritik olan iş sorunlarına odaklanma şansı daha yüksektir. Ancak bu, güvenlik ve taşınabilirlik dahil, genellikle işlevsel olmayan gereksinimler (AKA kısıtlamaları veya kalite özellikleri) olarak bilinen diğer kategorileri hariç tutar.

- Risk Kontrolü: RAD ile ilgili literatürün çoğu hız ve kullanıcı katılımına odaklansa da RAD'nin doğru bir şekilde yapılmasının diğer kritik bir özelliği de risk azaltmadır. Bir RAD yaklaşımı, temel risk faktörlerine erkenden odaklanabilir ve sürecin ilk bölümünde toplanan ampirik kanıtlara dayanarak bunlara uyum sağlayabilir. Örneğin, sistemin en karmaşık parçalarından bazıları prototiplemenin karmaşıklığıdır.

- Zamanında ve Bütçe Dahilinde Tamamlanan Daha Fazla Proje: Artımlı birimlerin geliştirilmesine odaklanarak, büyük şelale projelerini sürdüren yıkıcı başarısızlık şansı azalır. Şelale modelinde, tüm sistemin radikal bir şekilde yeniden düşünülmesini gerektiren 6 ay veya daha fazla analiz ve geliştirmeden sonra bir sonuca varmak yaygındı. RAD ile bu tür bilgiler keşfedilebilir ve süreç içinde daha erken harekete geçilebilir.

RAD'nin dezavantajları aşağıdakileri içerir:

- Çoğu taraf için RAD, deneyimli profesyonellerin çalışma biçimlerini yeniden düşünmelerini gerektiren yeni bir yaklaşımdır. İnsanlar hemen hemen her zaman değişime karşıdır ve yeni araçlar veya yöntemlerle üstlenilen herhangi bir projenin, yalnızca ekibin öğrenme gereksinimi nedeniyle ilk seferde başarısız olma olasılığı daha yüksektir.

- Normal çalışmada genellikle son kullanıcı tarafından görülmeyen, işlevsel olmayan gereksinimlere vurgu yapılabilir.

- Neredeyse tüm RAD yaklaşımlarının ortak noktası, kullanıcılar ve geliştiriciler arasında tüm yaşam döngüsü boyunca çok daha fazla etkileşim olmasıdır. Bu da fazla zaman kaybına neden olacaktır.

- Daha az kontrol mevcuttur. RAD'nin avantajlarından biri, esnek ve uyarlanabilir bir süreç sağlamasıdır. İdeal olan, hem sorunlara hem de fırsatlara hızla uyum sağlayabilmektir. Esneklik ve kontrol arasında kaçınılmaz bir değiş tokuş vardır. Birinin fazla olması diğerinin daha az olması anlamına gelir. Bir projede (örneğin hayati önem taşıyan bir yazılım) değerlerin çeviklikten daha fazla kontrolü varsa RAD uygun değildir.

- Tasarımı nispeten daha kötüdür. Prototiplere odaklanma, bazı durumlarda geliştiricilerin sürekli olarak bireysel bileşenlerde küçük değişiklikler yaptığı ve daha iyi bir genel tasarımla sonuçlanabilecek sistem mimarisi sorunlarını göz ardı ettiği "*hack ve test*" metodolojisiyle sonuçlanan çok ileri götürülebilir. Bu, özellikle James Martin'inki gibi sistemin kullanıcı arayüzüne çok fazla odaklanan metodolojiler için bir sorun olabilir.

- Ölçeklenebilirlik eksikliği mevcuttur. RAD tipik olarak küçük ve orta ölçekli proje ekiplerine odaklanır. Yukarıda belirtilen diğer sorunlar (daha az tasarım ve kontrol), çok büyük ölçekli sistemler için bir RAD yaklaşımı kullanırken özel zorluklar ortaya çıkarır.

Nesneye Yönelik Sistem Geliştirme (OOSD):

Nesneye yönelik sistem geliştirme (OOSD) analistlerin, geliştiricilerin ve programcıların bir sistemin daha büyük mantıksal parçalarını dikkate almalarına ve programlama sürecini netleştirmelerine izin veren, verileri ve prosedürleri nesneler halinde gruplandıran bir programlama tekniğidir. Klasik yaklaşımın aksine, veri merkezli, sınıf modelleri üzerinde geliştirilen bir yaklaşımdır. Bu yöntem kullanıldığında kodların ve modüllerin yeniden kullanılabilirliği kolaylaşmakta, geliştirme süresi kısalmakta, üretkenlik artmakta, yazılım kalitesi yükselmekte, anlaşılabilirdik kolaylaşmaktadır.

OOSD, sınırsız çeşitlilikteki verilerin yönetimine, karmaşık ilişkileri modelleme becerisine ve değişen bir ortamın taleplerini karşılama becerisine izin verir.

Bileşen Tabanlı Geliştirme:

Bileşen tabanlı geliştirme, hizmetlerini tanımlanmış ara birimler aracılığıyla kullanıma sunan, iş birliği yapan yürütülebilir yazılım paketlerinden uygulamaları bir araya getirir. Geliştirme süresini ve maliyetini azaltır, kaliteyi artırır, modülerliği destekler ve yeniden kullanımı kolaylaştırır.

Yazılım mühendisliği ve yazılım endüstrisi büyük sistemleri oluştururken daha önceden hazırlanmış bileşenler kullanarak üretkenlik ve kaliteyi artırma fikrini benimsemiştir. Bu fikir bilgisayar bilimlerinde çoğu alanda kullanılan böl ve birleştir yöntemi ile uyuşmaktadır. Bu kapsamda ortaya atılan bileşen tabanlı yazılım geliştirme (CBSD), 1990'ların başında ortaya çıkmış yeniden kullanımı artıran yöntemlerden biridir. Bileşenler, çalışma anında mevcut bir yazılım sisteminin çalışmasını etkilemeden yeni özelliklerin eklenmesini sağlayan bağımsız yazılım birimleridir. CBSD, bu bileşenleri kullanarak ekle ve çalıştır mantığıyla çalışır. Bu yapı yazılım sistemlerinin bakımı sürecinde büyük faydalar sağlamaktadır. Yeni gereksinimleri karşılamak için yeni bileşen eklemek veya var olan bileşenleri değiştirmek en uygun çözüm olarak görünmektedir. CBSD ile ilgili birkaç genel kural aşağıdaki gibi sıralanabilir:

- Ara yüz tabanlıdır. Bileşenler ara yüzü ve uygulamayı birbirinde ayırır ve uygulamanın detayını gizleyerek soyutlamayı sağlar.

- Mimari tabanlıdır. Daha önceden CBSD'ye göre tasarlanmış bir mimaride çalışabilirler, böylelikle yeni bileşen kolaylıkla diğer bileşenlerle birlikte çalışabilmektedir. Kullanılan mimariler genellikle MFC (Microsoft Foundation Class), CORBA, EJB gibi yapıların üzerinde kurulmaktadır. Bileşenler, ürüne özgü, alana özgü veya alan bağımsız olabilmektedir (CORBA Domain Objects gibi).

- Bileşenler için sınıflandırma şu şekilde yapılmıştır: Grafiksel kullanıcı arayüzü (GUI) bileşenleri en çok bulunan bileşenlerdendir. Bu bileşenlerde kod karmaşıklığı az olduğu için geliştirme maliyeti düşüktür. Örnek olarak; butonlar, metin alanları, veri gösterme için özelleştirilmiş tablolar gibi kullanıcı ara yüzleri verilebilir. Servis bileşenleri; daha karmaşık ve maliyeti yüksek bileşenlerdir. Uygulamaların ihtiyaç duyduğu ortak servisleri, veri tabanı erişimini sağlarlar. Örnek olarak; veri tabanı işlem servisleri verilebilir. Alana özgü bileşenler; geliştirmesi ve tekrar kullanımı en zor olan bileşenlerdir. Bileşenin kullanılacağı alan iyi tanımlanmış olmalıdır. Üretkenlik artışı çok fazladır. Örnek olarak; bir sigorta şirketi uygulaması için fatura, poliçe gibi bileşenlerin kullanılması verilebilir.

- Geleneksel yazılım geliştirme yöntemlerinden farklı olarak CBSD'de yazılım geliştirme süreci bileşen geliştirme ve bileşen entegrasyonu olarak ikiye ayrılır. Bu ayrıma göre bileşenlerin uygulanabilirlik, genel bir bileşen olma, diğer bileşenlerle birlikte çalışabilirlik gibi özellikleri incelenir. Bir bileşen bir sisteme eklenmeden önce bazı uyarlamaların yapılması gerekebilir. Buna bileşenin optimizasyonu adı verilmektedir. Bu işlem yapılırken bileşene yapılan değişiklik sonucu bileşenin temel özellikleri etkilenmemelidir. Örnek olarak; bir menü 10 bileşenin mobil cihazlarda daha küçük boyutta ve çözünürlükte gösterilmesi menü bileşeninde bir düzenleme yapılarak sağlanabilir. Birbiriyle etkileşim içinde olan iki bileşen birleşerek yeni bir bileşen oluşturabilir. Buna bir bakıma uygulama çatısı da denilebilir. Heineman'a göre, bileşenlerin birbiriyle etkileşimi istemci/sunucu ve yayıncı/abone mimarileri ile sağlanabilmektedir. İlkinde bileşenler bir istemci gibi davranarak istediği bilgiyi dönen yöntem çağrımı yaparlar. İkincisinde ise bir bileşene kayıt olup ondan sürekli bildirimler alırlar. Bir bileşen modeli bu sayılan özellikleri gerçekleştirmiş olmalıdır. Bahsedilen seçme, kullanma ve entegrasyon zorluklarına rağmen bu yaklaşım sıkça kullanılmaktadır.

Web Tabanlı Uygulama Geliştirme:

Sunucu ve istemci arasında işlem gören, istemci tarafından dosya/uygulama yüklenmesi yapmadan işlev gösteren tarayıcı tabanlı çalışan uygulamalar web tabanlı uygulama olarak nitelendirilir.

Bu yaklaşım, işletmeler içinde ve arasında kod modüllerinin daha etkin entegrasyonunu sağlamak için XML dillerini (SOAP, WSDL, UDDI) kullanır.

Yazılım Yeniden Yapılandırma:

Yazılım yeniden yapılandırma, mevcut sistemlerden bileşenlerin çıkarılmasını ve bu bileşenlerin yeni sistemler geliştirmek veya mevcut sistemlerin verimliliğini artırmak için yeniden yapılandırılmasını içeren bir süreçtir.

Herhangi bir organizasyonda yapı, sistem, süreç ve uygulanan politikalarda hızlı ve radikal bir şekilde yeniden tasarım ve değişiklikler yapılarak organizasyonun daha yüksek performansa ulaşmasını amaçlayan yönetim tekniğidir. Böylece mevcut yazılım sistemleri, işlevselliklerini uzatmak için modernize edilebilir.

Bir işletmenin iş süreçlerinin yeniden yapılandırılması (Business Process Reengineering-BPR) çabalarını incelerken, bilgi sistemleri denetçisinin aşağıdakilerin olup olmadığını tespit etmesi gerekir:

- BPR ekibi, BPR/süreç değişikliği projesinin tamamlanmasından sonra öğrenilecek dersleri belgelemiştir. İşletmenin değişim çabaları, işletmenin genel kültürü ve stratejik planı ile tutarlıdır.

- Yeniden yapılanma ekibi, değişimin işletme personeli üzerinde yapabileceği olumsuz etkileri en aza indirmeye çalışmaktadır.

Tersine Mühendislik:

Tersine mühendislik, bilgisayar destekli yazılım mühendisliği (CASE) teknolojisi kullanılarak mevcut uygulama sistem kodunun yeniden tasarlanıp kodlanabildiği bir yazılım mühendisliği tekniğidir. Örneğin; yazılım bakım/iyileştirilmesi kapsamında ilgili kaynak kodun anlaşılması, yazılım geliştirme sürecinde alınacak bir karar için gerekli bilgilerin çıkarılması, bir yazılım hatasının veya güvenlik açığının tespit edilmesi/düzeltilmesine yardımcı olabilir.

Bilgi sistemleri denetçisi aşağıdaki risk maddelerinden haberdar olmalıdır:

- Geri derleyiciler belirli bilgisayarlara işletim sistemlerine ve programlama dillerine bağlı işlevlere sahip araçlardır. Bu bileşenlerden birindeki herhangi bir değişiklik, yeni bir geri derleyici geliştirmeyi veya satın almayı gerektirebilir.

- Yazılım lisans sözleşmeleri genellikle lisans sahibinin yazılımı tersine mühendislik yapmasını saklayan hükümler içerir. Böylece hiçbir ticari sır ve programlama tekniği tehlikeye girmez.

DevOPS:

DevOps, çatışmaların ve engellerin ortadan kaldırılabilmesi için geliştirme ve operasyon süreçlerinin entegrasyonunu ifade eden yaşam döngüsüdür. Her bir aşama bir diğeriyle bağlantılıdır ve aşamalar role özel değildir. Gerçek bir DevOps kültüründe her rolün, aşamaların her biriyle belirli bir ölçüde ilişkisi vardır. Bu entegrasyon büyük faydalar sağlayabilir ancak yeni riskler de yaratabilir. DevOps'u kullanma kararı bir kurumun ortamı, risk toleransı, kültürü ve geliştirme projesinin kapsamı gibi faktörlere dayanarak yapılmalıdır.

DevOps ortamı değiştirdiğinden ve genellikle bir işletmenin kontrol ortamını ve kabul edilen risk seviyesini etkilediğinden, bilgi sistemleri denetçisi görevlerin uygun şekilde ayrılmış (SoD) olduğundan emin olmalıdır.

İş Süreçlerinin Yeniden Yapılandırılması ve Süreç Değişimi:

İş süreçlerinin yeniden yapılandırılması (BPR), mevcut iş ortamında hayatta kalabilmek için mevcut süreçlerin yeniden yapılandırılmasına dayanmaktadır. Bu süreç genelde manuel sistem süreçlerinin otomatizasyonuyla ilişkindir. Manuel müdahale ve kontrollerin azaltılması hedeflenir.

BPR adımları aşağıdaki gibidir:

- Yeni sürecin uygulanması ve izlenmesi,
- Bir proje planının geliştirilmesi,
- Sürekli bir iyileştirme sürecinin oluşturulması,
- Gözden geçirilecek alanların tanımlanması,
- İncelenmekte olan süreci anlama,
- Sürecin yeniden tasarım ve düzenlenmesi.

BPR'de kilit kontroller, iş süreci dışında yeniden yapılandırılabilmektedir. Bu yapılandırma bilgi sistemleri denetçisi tarafından özellikle denetlenmelidir.

2.1.4. Sistem Geliştirme Araçları ve Kod Geliştirme Yardımcıları

Sistem geliştirme sürecinde muhtelif araç ve kod geliştirme yardımcıları kullanılmaktadır. Bu kısımda, söz konusu araç ve yardımcılara yer verilecektir.

Bilgisayar Destekli Yazılım Mühendisliği (CASE):

Bilgisayar destekli yazılım mühendisliği (CASE), yazılım geliştirme sürecinde yüksek kaliteli ve hatasız yazılım ürünleri sağlamak için bilgisayar destekli yazılım araç ve yöntemlerin yazılım geliştirilmesinde bilimsel uygulamasıdır. CASE, kontrol odaklı ve disiplinli bir yaklaşım sağlar. Tasarımcılar, geliştiriciler, testçiler, yöneticiler ve diğerlerinin geliştirme sırasında proje aşamalarını görmelerine yardımcı olur. Bilgi sistemleri denetçisi, CASE'in getirdiği gelişim surecindeki değişiklikleri tanıyabilmeli ve CASE'i bir denetim aracı olarak kullanabilmelidir.

Yazılım sistemleri, yazılım süreç aktivitelerine otomatik destek sağlamaya yöneliktir. CASE sistemleri metot desteği için kullanılır.

CASE, uygulamaları tasarlamak ve uygulamak için kullanılan yazılım araçları etki alanıdır. CASE araçları bilgisayar destekli tasarım, donanım ürünleri tasarlamak için kullanılan (CAD) araçlarına benzerler. CASE yazılımı, genellikle yazılım geliştirme sürecinde kullanılabilecek otomatikleştirilmiş araçlarla birlikte bilgi sistemlerinin geliştirilmesi için yöntemlerle ilişkilendirilir.

CASE yaklaşımının avantajları aşağıdakileri içerir:

- Testin yanı sıra yeniden tasarlamaya da önem verildiğinden, bir ürünün beklenen ömrü boyunca servis maliyeti önemli ölçüde azalır.
- Geliştirme sürecinde organize bir yaklaşım izlendiğinden ürünün genel kalitesi artar.
- Gerçek dünya gereksinimlerini karşılama şansı daha olası ve daha kolaydır.
- Dolaylı olarak, yüksek kaliteli ürünlerin geliştirilmesine yardımcı olarak işletmeye rekabet avantajı sağlar.

CASE araçları kaliteli, kusur içermeyen, kullanışlı ve sürdürülebilir yazılımları geliştirmek, yazmak, kodlamak ve tasarlamak için kullanılır. CASE araçları, kriterlerine ve türlerine göre aşağıda yer verilen gruplarda incelenebilir:

- Kod üreten yazılım araçları grubu,
- Üründe veya yazılımda yapılan tasarımlar için kullanılan tasarım araçları grubu,
- Veri modellemede kullanılan araçlar grubu,
- Versiyonları kontrol eden, yazılım ayarlama yapan araçlar grubu,
- Model dönüşüm yazılımları grubu,
- Kod yeniden üretim yazılımları grubu.

Kod Üreteçleri:

Yazılım dünyasındaki nesne tabanlı dillerin gelişmesiyle programcılık, yazılım algoritmaları, veri tabanı kavramları ve yazılım süreçleri değişimlere uğramıştır. Bir bilgi sistemleri denetçisi, sistem analisti tarafından tanımlanan parametrelere veya bir CASE ürününün tasarım modülü tarafından geliştirilen veri/varlık akış şemalarına dayanarak program kodu üreten araçlar ve bu tür araçlar tarafından üretilen kaynak kodunun farkında olmalıdır.

Kod üreteçleri, temel olarak kullanıcının ne yapmak istediğini programa öğretmesiyle başlar. Genelde kullanıcının istediği giriş forumları, kayıt girme, düzeltme, silme gibi bir temel işlemlerdir. Eğer mantıksal işlemlere yapacağı zaman ise yine kod yazma araçları bazı işlemleri gerçekleştirmektedir.

Bu değişimlerin en önemli süreci makro programlama olarak nitelendirilen "*yapacağın işlemleri bana söyle ben yapayım*" programlama türüdür. Başka bir tanımı ise paket programların bazılarının içerisinde bulunan, kullanıcıya kolaylık olması açısından, sürekli tekrar edilen (rutin) işlemlerin otomatik hale getirilmesi için kullanılan bir komut/komutlar dizisidir. Çünkü bilinen bir şey var kullanıcı bir nesneyi seçer ve ona değişik hareketler kazandırır. Yazma, silme, yazdırma, seçme, hareket etme gibi birçok özellik kazandırabilir. Bunu yaparken hiçbir programlama da bilmez. Program kendi kodunu kendisi yazar.

Dördüncü Nesil Diller (4GLs):

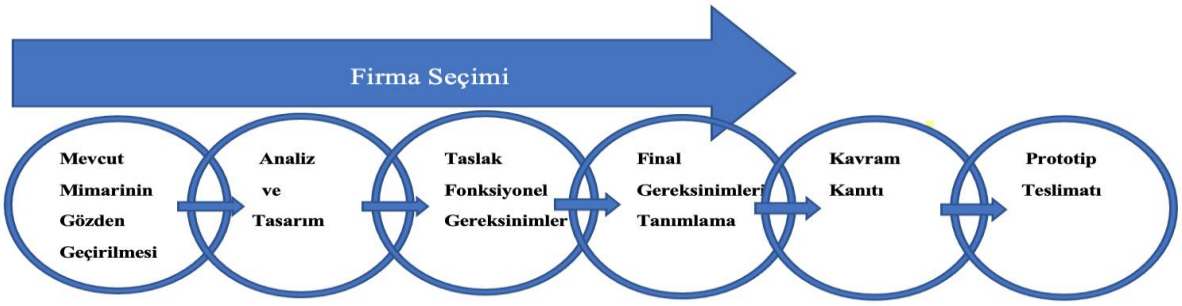
Dördüncü nesil (programlama) dil (4GL), 3GL'lerden insan diline, düşünme biçimine ve kavramsallaştırmaya yaklaşmaya çalışan bir grup programlama dilidir. 4. nesil dil, etki alanına özgü dil veya yüksek üretkenlik dili olarak da bilinir. Bu tür bir dilin amacı, profesyonel olmayan geliştiricilerin (örneğin, bilgisayar mühendisleri olmayanlar) kendi uygulamalarını geliştirmelerine izin vermektir. 4GL'ler, yazılım geliştirmenin toplam süresi, çabası ve maliyetinin azaltmak için tasarlanmıştır.

4GL'lerin temel alanları ve ailelerini; veritabanı sorguları, rapor üreteçleri, veri işleme, analiz ve raporlama, ekran ressamları ve üreteçleri, GUI yaratıcıları, matematiksel optimizasyon, web geliştirme ve genel amaçlı dilleri oluşturmaktadır.

4GL'ler ortamsal açıdan bağımsız ve basit bir dil altkümesine ve bir tezgâh yaklaşımına sahip olan, prosedürel olmayan dillerdir. Bu tür diller, programcıların çok az kod içeren programlar yazmalarına izin verir ve bu da üretkenliklerini artırır. Bu dillerde yazılan programlar daha yavaş çalışır, daha fazla kaynak gerektirir ve programcılar bazı durumlarda 3 E veya 2 E nesli bir dil kullanmayı tercih ederler.

2.1.5. Altyapı Geliştirme/Edinim Süreçleri

Altyapı geliştirme ve edinim süreçleri aşağıdaki tabloda da yer verildiği üzere bünyesinde birçok süreç barındırır. Bu süreçler aşağıda ele alınacaktır.



Altyapı Geliştirme/Edinim Süreçleri Şeması

Fiziksel Mimari Analizinin Proje Aşamaları:

Fiziksel mimari analizi kapsamında gereksinimler, bir kavram kanıtı kullanılarak doğrulanır. Kavram kanıtı, temel kurulum ve işlevselliği gösteren çalışan bir prototip sağlamalıdır. Fiziksel mimari analizine yönelik proje aşamaları ve tedarikçi seçim süreci fiziksel mimari analizinin bir parçasıdır.

a. Mevcut Mimarinin Gözden Geçirilmesi

Mevcut mimarinin gözden geçirilmesi aşamasında, en güncel dokümanların gözden geçirilmesi yapılır. Bir toplantı düzenlenerek, fiziksel mimarideki doğrudan etkilenen tüm alanlardaki uzmanların (genel BT altyapısı, sunucu, depolama, güvenlik vb.) katılımı sağlanmalıdır. Bu aşamada fiziksel mimariyi etkileyen tüm operasyonel kısımların karakterize edilmesine özel dikkat gösterilmelidir. Fiziksel mimarinin aşağıdaki unsurlarına özel dikkat gösterilir:

- Zemin sorunları,
- Boyut sınırları,
- Ağırlık sınırları,
- Akım güç kaynağı,
- Çevresel çalışma sınırları (minimum ve maksimum sıcaklık ve nem),
- Fiziksel güvenlik sorunları.

Burada ilk toplantının çıktısı mevcut altyapı bileşenlerinin ve hedef fiziksel mimariyi tanımlayan kısıtlamaların bir listesidir.

b. Analiz ve Tasarım

Mevcut mimariyi inceledikten sonra, iyi uygulamalara bağlı kalarak ve iş gereksinimlerini karşılayarak, gerçek fiziksel mimarinin analizi ve tasarımı yapılmalıdır.

c. Taslak Fonksiyonel Gereksinimler

Mevcut mimariyi inceledikten sonra iyi uygulamalara bağlı kalarak iş ve gereksinimlerini karşılayarak, gerçek fiziksel mimarinin analizi ve tasarımı yapılmalıdır.

Diğer taraftan, taslak fonksiyonel gereksinimler yazılırken, tedarikçi seçim süreci paralel olarak ilerler.

d. Fonksiyonel Gereksinimlerin Yazılması

Taslak fonksiyonel gereksinimleri bitirdikten ve bu projenin ikinci kısmını besledikten sonra fonksiyonel gereksinimler belgesi yazılır ve eklenen tüm tarafların personeli ile ikinci mimari toplantısı düzenlenir. Toplantıda bu belge değerlendirilir. Bunun sonucunda tartışılacak ve rafine edilmesi veya eklenmesi gereken gereksinimlerin bir listesi oluşturulacaktır.

Kavram kanıtının (Proof of Concept, POC) planlaması ikinci toplantıdan sonra tamamlanmış fonksiyonel gereksinimlerle başlar.

e. Kavram Kanıtlama

Seçilen donanım, yazılım ve güvenlik gereksinimleri dahil tüm beklentileri karşılayabildiğini kanıtlayabilmek için bir POC oluşturulması özellikle tavsiye edilir. POC'nin çıktıları, testleri ve sonuçlarını tanımlayan ilişkili belge ve test protokolleri dahil olmak üzere çalışan bir prototip olmalıdır.

Başlangıçta, POC, aşağıda açıklanan satın alma aşamasının sonuçlarına dayanmalıdır. Bu amaçla, hedef donanımın temsili bir alt kümesi kullanılır. POC'yi çalıştırmak için kullanılan yazılım, test sürümleri veya önceden sağlanan yazılım olabilir. Bu nedenle, ek maliyetlerin minimum olması beklenir. Çerçevenin çoğu öğesi, maliyetleri düşük tutmak için basitleştirilmiş bir biçimde uygulanır. Daha sonraki aşamalarda nihai biçimlerine genişletilecektir.

Hazırlanan prototip aşağıdaki özellikleri göstermelidir:

- Temel güvenlik altyapısının esas kurulumu,

- Kontrol bileşenlerinin doğru işlevselliği,

- Tanımlanan güvenlik önlemlerinin temel ancak işlevsel uygulaması,

- Güvenli işlemler,

- Kurulum kısıtlamaları ve sınırları (sunucu boyutu, sunucu akım tüketimi, sunucu) açısından karakterizasyon ağırlık, sunucu odası fiziksel güvenliği),

- Performans,

- Operasyonel durumda temel yük devretmeyi dahil etmek için güvenilir esneklik,

- Finansman ve maliyetlendirme modeli,

- Veri ve algoritma.

Dağıtım için hazırlanan ilgili uygulama projeleri de POC'nin bir parçası olmalıdır. Bunun nedeni bunların üretimin fiziksel mimarisinde kullanılanlar ile aynı nitelikte kullanılacak olmalarıdır. Bu aşamanın sonunda, üretimin boyutlandırılması ve düzeni için POC sonuçlarını içerecek şekilde uyarlandığı son bir toplantı yapılır.

İşletme, uygulamaların dağıtımı ve işletimi için bir dış kaynak kullanımı/sınır ötesi modelle uğraşıyorsa, ek hususlar geçerli olabilir. Bununla beraber, BT ortamını çalıştırma platformu da (sahip olunan, bulut tabanlı, sanallaştırma) ek hususlar getirebilir. Örneğin, işletme yüksek düzeyde düzenlemeye tabi bir sektörde veya yüksek kullanılabilirlik gerektiren bir sektörde faaliyet gösteriyorsa, POC test edilirken veri gizliliğini sağlamak için yeterli fazlalık ve güvencenin dikkate alınması gerekebilir.

Altyapı Uygulamaya Konmasının Planlanması:

Altyapı uygulamasının planlanmasında, sonuçların kalitesini sağlamak aşamalı bir yaklaşım kullanılması gerekir. Diğer projelerle iletişim süreçleri oluşturmak da önemlidir. Bu farklı aşamalar sayesinde, bileşenler bir araya getirilir ve tedarik aşamasında ve sonrasında seçim süreci kullanılarak mevcut ve iletişim kurulabilecek tedarikçilerin net bir şekilde anlaşılması sağlanır. Ayrıca, teslimat, kurulum ve test planlarının geliştirilmesini içeren sonraki adımları hazırlamak için temel iş ve teknik gereksinimlerin kapsamını seçmek gerekir. Diğer taraftan, geleceğe yönelik bir çözüm sağlamak için doğru becerilere sahip doğru ortakları seçmek çok önemlidir. İhtiyaç analizi bu sürecin bir parçası değildir. Ancak sonuçları sürekli olarak sürece besler. Bu aşamalarla bir Gantt şeması üretilirse, bazı aşamalar muhtemelen çakışacaktır. Bu nedenle, farklı aşamalar yinelemeli bir süreç olarak düşünülmelidir.

Aşağıdaki dört farklı aşamada sonraki projelere hazırlanmak için tüm bileşenlerin birbirine uyması gerekir (örn. veri geçişi).

1. Tedarik Aşaması

Tedarik aşamasında, seçilen çözüme genel bir bakış sağlamak ve çıktıların nicel yapısını belirlemek için işletme ve analiz projesi arasında iletişim kurulur. Gereksinim ifadeleri de üretilir. Ayrıca tedarik süreci, hizmet düzeyi yönetimi sürecini başlatır. Bu faaliyetler sırasında, tercih edilen ortaklar müzakere sürecine davet edilir ve teslimatlar, sözleşmeler ve SLA'lar[1] imzalanır. Özetle bu aşamanın içeriğini;

- İletişim surecini oluşturma ve teslimatları, sözleşmeleri ve SLA'leri belirleme ve

- Gereksinim beyanını üretme

oluşturur.

[1] *SLA, müşterinin Plan(lar)ının sürekliliğinin korunması için müşterinin, VeriTeknik'e yüklemiş olduğu sınırlı sorumlulukların tümüne verilen kısa isimdir.*

2. Teslimat Süresi

Teslimat süresi aşamasında teslimat planı geliştirilir. Bu aşama, bazı kısımlarda satın alma aşamasıyla çakışır. Teslimat planı, öncelikler, hedefler ve gayri resmi hedefler, temel gerçekler, ilkeler, iletişim stratejileri, kilit göstergeler ve kilit görev ve sorumluluklardaki ilerleme gibi konuları içermelidir. Bu aşamanın içeriğini teslimat planını geliştirme kapsamında öncelikler, amaçlar, önemli gerçekler, ilkeler, iletişim stratejileri, anahtar göstergeler, önemli işlerdeki ilerlemeler ve sorumluluklar oluşturur.

3. Kurulum Planı

Kurulum planlama aşamasında, kurulum planı etkilenen tüm taraflarla iş birliği içinde geliştirilir. Ek bir adım, planı ilgili taraflarla ve tabii ki entegrasyon projelerinden sorumlu olanlarla gözden geçirmektir. Bu yinelemeli bir süreçtir. Bu aşama ise özetle ilgili taraflarla planı oluşturma ve gözden geçirme şeklinde tanımlanabilir.

4. Kurulum Test Planı

Kurulum planının bilinen bağımlılıklarına dayalı olarak test planı geliştirilir. Test planı, mümkün olduğunda uygulamalar ve altyapı için test senaryolarını, temel gereksinimlerin özelliklerini, süreçlerin tanımını ve ölçüm bilgilerini içerir. Projenin ilk kısmı (fiziksel mimarinin analizi) tamamlanmalı ve gerekli altyapı kararları alınmalıdır. Son aşama ise test senaryolarını, temel gereksinim spesifikasyonları, süreçlerin ve ölçütlerin tanımlanmasını içeren test planı geliştirme şeklinde verilebilir.

2.1.6. Donanım/Yazılımı Tedarik Süreci

Donanım ve yazılım tedarik sürecinde aşağıda belirtilen hususlar dikkate alınır.

Donanım Edinimi:

a. Sistem Edinim Etkenleri

Bir sistemi geliştirmek mi yoksa satın almak mı kararı için aşağıdaki etkenler değerlendirilmelidir:

- Sistemin işlevsel olması gerektiği tarih,

- Sistemi satın almak yerine sistemi geliştirme maliyeti,

- Gereken kaynaklar, personel ve donanım,

- Bir tedarikçi sisteminde, lisanslama özellikleri (örneğin, yıllık yenileme, sürekli) ve bakım maliyetleri,

- Yeni sistemle arayüz oluşturmaya ihtiyaç duyacak diğer sistemler,

- Stratejik iş planları, risk iştahı, yasal uyumluluk gereklilikleri ve işletmenin BT altyapısı ile uyumluluk,

- İşlevsellik değişiklikleri için gelecekteki olası gereksinimler.

b. Sistem Spesifikasyonları

Yeni bir sistem edinirken, teknik özellikler aşağıdakileri içermelidir:

- Organizasyonel tanım (merkezi/merkezi olmayan, dağıtılmış, dış kaynaklı, insanlı veya insansız),

- Güvenlik sağlamlığı için donanım ve yazılım değerlendirme güvencesi seviyeleri,

- Bilgi işleme gereksinimleri,

- Donanım gereksinimleri,

- Sistem yazılım uygulamaları,

- Destek gereksinimleri,

- Uyarlanabilirlik ve dönüşüm gereksinimleri,

- Sistem kısıtlamaları.

Bu alanda bir denetim yaparken, bilgi sistemleri denetçisi aşağıdakileri de dikkate almalıdır:

- Satın alma işleminin bir iş gereksinimi ile başlayıp başlamadığını ve bu ihtiyaç için donanım gereksinimlerinin spesifikasyonlarda dikkate alınıp alınmadığını belirlemek.

- Birkaç işletmenin değerlendirilip değerlendirilmediğini ve aralarındaki karşılaştırmanın belirtilen kriterlere göre yapılıp yapılmadığını belirlemek.

Yazılım Edinimi:

a. Gereksinimlerin Tanımı

Gereksinimlerin tanımı, bir sistemin ne yapması gerektiği, kullanıcıların bir sistemle nasıl etkileşime gireceği, sistemin hangi koşullarda isleyeceği ve sistemin yerine getirmesi gereken bilgi kriterleri hakkında açıklamalar içermelidir. Gereksinimler tanımlanırken ihtiyaç analizi çıktıları, performans gereksinimleri, tasarım gereksinimleri ve müşteri gereksinimleri de göz önünde bulundurulmalıdır.

Gereksinim tanımlarını başarıyla tamamlamak için proje ekibi aşağıdaki gibi görevleri yerine getirmelidir:

- Mevcut kısıtlamaların belirlenmesi,

- Paydaş/tarafların belirlenmesi,

- Uyuşmazlık/uygunsuzlukların belirlenmesi ve düzeltilmesi,

- Gereksinimlerin yapısal şekilde belirlenmesi ve tüm paydaşların görüşünün alınması,

- Uyuşmazlık ve aksaklıkların çözülmesi,

- Gereksinimlerin eksiksiz, tutarlı, acık, doğrulanabilir, değiştirilebilir, test edilebilir ve izlenebilir olduğunu doğrulamak.

b. Teklif Talebi (RFP)

Teklif talebi (RFP), satıcılardan müşterinin sorunlarına veya iş gereksinimlerine çözüm önermelerini isteyen bir belgedir. Bunlardan bazıları iş hakkında, yasal ve güvenlikle ilgili gereksinimlerin yanı sıra geliştirme ile ilgili gereksinimlerdir. Talep, tanımlanmış bir ürün veya hizmet için farklı tedarikçilerden ayrıntılı ve karşılaştırılabilir teklifler almak için resmi bir yöntemdir. Satın alma kararına ilişkin gerekli tüm bilgileri içeren kapsamlı ve resmi belgedir.

RFP süreci boyunca tedarikçilere karşı şeffaf bir tutumla ihtiyacın net bir şekilde dile getirilmiş olması kritik öneme sahiptir. Gizlilik kaygısı duyulan bir proje ise, masaya oturması muhtemel tüm tedarikçilerle gizlilik sözleşmesi imzalanabilir.

Gereksinimlerin net olarak tanımlanması için bir teklif talebi hazırlanır. Teklif talebi hazırlanırken aşağıdaki unsurların değerlendirilmesi zorunludur.

TEKLİF TALEBİ (RFP)

Teklif Talebi Unsurları Şeması

Yazılım edinmeye ilişkin olarak bilgi sistemleri denetçisi aşağıdakileri yapmalıdır:

- Sözleşme imzalanmadan önce hukuk danışmanı tarafından kontrol edildiğine emin olmak,

- Listelenen ögelerin kapsamasını ve seçilen firmanın, RFP belgeleri tarafından desteklenip desteklenmediğinin doğrulanması için RFP'yi incelemek,

- Bir çözüm edinim kararının uygun olup olmadığını belirlemek için fizibilite çalışması ile ilgili dokümantasyonunu incelemek,

- Sistemin, firmanın RFP'ye verdiği yanıtlarla eşleşmesini sağlamak için gündem temelli sunumlara ve konferans salonu pilot uygulamalarına katılmak,

- İmzalama işlemi başlamadan önce firma sözleşmesini gözden geçirmek,

- Sözleşmenin imzalanmadan önce hukuk danışmanı tarafından gözden geçirildiğinden emin olmak,

- Güvenlik müdahale ve sorumluluklarının firma tarafından dahil edilmesini sağlamak için RFP'yi incelemek,

- Bir çözüm elde etme kararının uygun olup olmadığını belirlemek için fizibilite çalışmasından gelen dokümantasyonu analiz etmek (ortak kriter değerlendirmelerinin dikkate alınması dahil),

- Seçilen firmanın RFP belgeleriyle desteklenip desteklenmediğini belirlemek,

- Listelenen ürünleri içerdiğinden emin olmak için imzalanmadan önce firma sözleşmesini gözden geçirmek,

- Bu bölümde listelenen ögelerin kapsamasını sağladığından emin olmak için RFP'yi incelemek.

2.1.7. Test Süreçleri

Yazılım ve donanım edimini sırasında, söz konusu edinimlerin testlerinde muhtelif yöntemler kullanılır. Bu yöntemlere aşağıda yer verilmektedir.

Test Kategorileri:

a. Birim Test

Birim testi, bir başka yazılımın işlevlerini (metot, sınıf vs.) çalıştıran yazılım yöntemidir. Yazılım, birim testi ile test edilen kodun beklenen duruma (durum testi) gelip gelmediğini veya olayların oluş sırasının (davranış testi) çalışıp çalışmadığı kontrol edilir.

Belli bir program veya modülün program mantığını test eder. Programın iç operasyonunun spesifikasyonları göre işlemesini sağlar. Prosedürel tasarımın kontrol yapısına odaklı bir dizi test senaryosu kullanır.

b. Arayüz ve Entegrasyon Testi

Arayüz testi, bir alandan diğerine bilgi ileten iki veya daha fazla bileşenin bağlantısını değerlendiren bir donanım veya yazılım testidir. Entegrasyon testi veya genel test olarak da bilinen bu test yöntemi birden fazla modül veya bileşeni olan sistemin tümünü kontrol eden donanım veya yazılım testidir. Genelde birim testleri (unit test) biten yazılımlar için uygulanır.

Entegrasyon testindeki asıl amaç, oluşturulan yazılım modüllerinin bir araya getirilerek doğruluğunun sağlanmasıdır. Yazılım ürünü için oluşturulan tüm modüller bir araya getirilir ve bu şekilde test edilir. Burada ki amaç, metotlar birim başına testten geçerken, modüller halinde bir araya geldiğinde bazı hatalara sebep oluyor olabilirler. Entegrasyon testleri, bu tarz yazılım ürünü problemlerinin henüz canlı (prod) ortama çıkmadan tespit edilmesini veya geliştirilen yeni modülün de sorunsuz çalışabileceğinden hızlı bir şekilde emin olunmasını sağlar.

c. Sistem Testi

Toplu olarak yeni veya değiştirilmiş bir sistem oluşturan değiştirilmiş programların, nesnelerin, veritabanı şemasının vb. belirtilen sisteme uygunluğunun değerlendirilmesi için tasarlanmış eksiksiz bir entegre sistem üzerinde yapılan bir dizi testtir.

d. Son Kullanıcı Testi

Uygulama aşamasında gerçeklesen, işletmenin kalite güvence metodolojisini uygulayan, son kullanıcı/müşteri isteklerine dayanan sistem testidir. Uygulamanın işlevsel yönüne odaklanır. Sistemin uygulamaya hazır olmasını ve dokümante edilmiş tüm gereksinimlerini karşılamasını sağlar. Canlı ortama benzer koşullarda güvenli bir test veya hazırlık ortamında gerçekleştirilir. Test adımları kullanıcı bakış açısı ile yazılarak, BT bölümü ve son kullanıcı ile gerçekleştirilir.

Bu testtin amacı; yazılımı, iş gereksinimlerine göre doğrulamaktır. Yazılımın kullanıcı ve müşteri tarafından kabul edilip edilmeyeceğini belirlemek için test edilmesine olanak sağlar. Bu doğrulamalar iş gereksinimlerine aşina olan son kullanıcılar tarafından gerçekleştirilir. Fonksiyonel, sistem ve regresyon testleri gerçekleştirildikten sonra kullanıcı kabul testleri gerçekleştirilir. Yazılım yayınlanmadan gerçekleştirilen son testtir.

Diğer Testler:

a. Alfa ve Beta Testi

Alfa testi olarak adlandırılan ilk aşama, uygulama sisteminin ilk sürümlerinde yalnızca yazılımı geliştiren işletmedeki kullanıcılar tarafından gerçekleştirilir (sistem testleri). Geliştirmeden sonraki yazılım testinin ilk aşamasıdır. Yazılımın beta testine geçmesi için alfa testini geçmesi gerekir. Alfa testinin iki aşaması vardır. İlk aşama, geliştiriciler tarafından test edilmesidir. Temel amaç, hataları hızlı bir şekilde tespit etmektir. Testin ikinci aşaması, sistemin kullanıcı tarafında olduğu gibi gerçek ortamda mükemmel bir şekilde çalışmasını sağlayan kalite güvence ekibi tarafından yapılmaktadır.

Beta testi (beta testing), gerçek kullanıcılar tarafından gerçek ortamda gerçekleştirilmektedir. Uygulamanın kullanıcılar tarafından test edildiği bir test çeşididir. Kullanıcılar için beta versiyonu yayınlanmaktadır. Ardından kullanıcılardan gelecek geri dönüşler doğrultusunda düzeltmeler yapılmaktadır. Beta testi, ürün arıza risklerini azaltır ve müşteri doğrulaması yoluyla ürünün kalitesinin artmasını sağlar.

b. Pilot Testi

Sistemin bir bileşenini veya tüm sistemi gerçek zamanlı bir çalışma koşulu altında doğrulayan kavram kanıtı gibi bir sistemin spesifik ve önceden belirlenmiş yönlerine odaklanan bir yazılım testidir. Amacı bir araştırma projesinin fizibilitesini, zamanını, maliyetini, riskini ve performansını değerlendirmektir. Bu test tam olarak UAT ve üretim arasında yapılır.

Pilot testinde, seçilen bir grup son kullanıcı test edilen sistemi dener ve sistemin tam olarak konuşlandırılmasından önce geri bildirim sağlar. Diğer bir deyişle, aşağıdaki kullanılabilirlik testi için kostümlü prova yapmak anlamına gelir.

Pilot testi, sistemdeki hataların erken tespitine yardımcı olur.

Pilot testi, sürekli ve düzenli kullanıma karşı test etmek için müşteri sahasına (veya kullanıcı simülasyonlu bir ortama) bir sistem kurmakla ilgilidir. En yaygın test yöntemi, zayıf alanlarını bulmak için sistemi sürekli olarak test etmektir. Bu zayıflıklar daha sonra geliştirme ekibine hata raporları olarak geri gönderilir ve bu hatalar sistemin bir sonraki yapısında giderilir. Bu süreçte bazen kabul testleri de uyumluluk testinin parçası olarak dahil edilir. Bu, eskisinin yerini alacak bir sistem geliştirildiğinde ortaya çıkar.

Yazılım mühendisliğinde pilot testi, ürün veya hizmetin potansiyel bir pazarı olup olmadığı gibi soruya cevap verecektir.

Bu test aşağıda belirtilen nedenlerden dolayı önemlidir.

- Test için kullanılan yazılım ve prosedürde hata ayıklama, tam ölçekli uygulama için ürün hazırlığını kontrol etme, zaman ve kaynak tahsisinde daha iyi karar verme, hedef kitlesinin programa tepkisini ölçme fırsatı verir; başarı ölçümü ekibe kullanılabilirlik testi için kullanacakları etkinlikleri uygulama şansı verir.

- Gerçek zamanlı çalışma koşulları altında sistemin bir bileşenini veya tüm sistemi doğrulamaktır.

- Tam olarak UAT ve üretim arasında yapılır.

- Ürünün tam ölçekli uygulamaya hazır olup olmadığını kontrol etmeye yardımcı olur.

c. Beyaz Kutu Testi

Beyaz kutu testi, bir programın/modülün beklenen davranışını doğrulamak için bu program/modülün uygulanmasının ve kod aralıkları bilgisinin kullanıldığı bir test yaklaşımıdır. Bu test tasarım tekniği veri akışlarına, kontrol akışlarına, ifade kapsama, dal kapsama gibi konulara odaklanır. Beyaz kutu testleri aynı zamanda saydam (cam) kutu testi olarak da bilinir. Yazılımın dış kabuğunu, kutunun iç işleyişini görmeyi sembolize eder. Aynı şekilde, kara kutu testinde yer alan kara kutu, kutunun, yazılım içeriğinin görülmediğini ve kapalı olduğunu temsil eder. Son kullanıcı sistemin içini görmez. İşlevselliği ile ilgilenir.

d. Kara Kutu Testi

Kara kutu testi, uygulamanın veya ürünün işlevselliğine odaklanan ve kod aralıkları bilgisi gerektirmeyen bir test yaklaşımıdır. Bu testlerdeki amaç, gereksinimleri karşılayan çıktıların alınıp alınmadığının ölçümlenmesidir. Sistemden alınması beklenen çıktılar kadar beklenmeyen çıktılar da test edilmelidir. Bu test tasarım tekniği, test uzmanlarının sıklıkla kullandığı bir yöntemdir. Kara kutu testi; birim, entegrasyon, sistem ve kullanıcı kabul testi gibi testlerin her seviyesinde kullanılabilir.

Yazılım test tasarım tekniklerinden olan beyaz kutu testleri geliştirilen yazılımın içyapısı ve iş akışıyla ilgilenirken kara kutu testleri sistemin işlevselliği ile ilgilenir. Kara kutu testleri son kullanıcı perspektifinden test etmeyi içerir. Dolayısıyla kod içeriği ve işleyişin bilinmesine gerek yoktur.

e. Fonksiyonel/Doğrulama Testi

Fonksiyonel/doğrulama testi, oluşturulan yazılımın müşteri gereksinimlerine göre izlenebilmesini sağlamak için sistemin işlevselliğini ayrıntılı gereksinimlere karşı test eder.

Yazılımın tüm gereksinimlere uygun olduğundan emin olmak için test edildiği, yazılım geliştirme içinde kullanılan bir yazılım test sürecidir. İşlevsel test, yazılımı, işlevsel gereksinimlerinde belirtilen tüm gerekli özelliklere sahip olduğundan emin olmak için kontrol etmenin bir yoludur.

Aşağıdakileri kontrol etmek için genellikle işlevsel test kullanılır:

- Temel İşlevsellik

- Temel Kullanılabilirlik

- Ulaşılabilirlik

- Hata Koşulları

f. Regresyon Testi

Regresyon testi, canlıda çalışan uygulama/kod vb. üzerindeki değişikliklerin veya düzeltmelerin yeni hatalar getirmediğinden emin olmak için bir test senaryosunun veya test planının bir kısmını yeniden koşma işlemidir. Bu değişiklik/düzeltmeler yeni bir fonksiyon, hata çözümü ya da performans geliştirmesi olabilir. Regresyon testleri genellikle değişiklikler son aşamaya geldiğinde ve yazılımın yeni sürümü yayınlanmadan önce gerçekleştirilir. Regresyon testlerinin öncelikli amacı, uygulamanın kritik alanlarının hala beklendiği gibi çalıştığını kontrol etmektir. Regresyon testleri:

- Yazılımın değişiklik sonrasında, son kalitesinin kontrol edilmesini,

- Daha önce çıkan hataların düzeldiğinin kontrolünü,

- Yazılım ekibinin ürün hakkında güveninin artmasını,

sağlar.

g. Paralel Test

Paralel test, test verisini iki sisteme, değiştirilmiş sisteme ve alternatif bir sisteme (muhtemelen orijinal sistem), besleme ve uygulamanın iki sürümü arasındaki sonuçları karşılaştırma sürecidir. Bu test, test yürütme süresini azaltmak için bir uygulamanın birden çok sürümünün veya alt bileşeninin aynı anda farklı sistemlerde aynı girişle test edildiği bir yazılım test türüdür. Bu testin amacı, eski sürüm ile yeni sürümün aynı mı yoksa farklı mı davrandığını bulmak ve yeni sürümün daha verimli olup olmadığını sağlamaktır. Paralel testin yapılma nedenleri:

- Uygulamanın yeni sürümünün doğru çalıştığından emin olmak,

- Yeni ve eski sürüm arasındaki tutarlılıkların aynı olduğundan emin olmak,

- İki sürüm arasındaki veri formatının değişip değişmediğini kontrol etmek,

- Yeni uygulamanın bütünlüğünü kontrol etmek.

h. Uyumluluk Testi

Uyumluluk testi, yeni veya değiştirilmiş sistemin, var olan sistemleri olumsuz yönde etkilemeden hedef ortamında çalışabileceğini onaylamak için yapılan testtir. Yazılan kodun uyumluluğunu kontrol eder. Donanım/yazılım/işletim sistemi/ağ ortamında ne kadar iyi performansa sahip olduğunu tespit eder ve doğrular. Bu testin hedefleri aşağıdaki gibidir:

- Geliştirme ve bakım sürecinin öngörülen metodolojiyi karşıladığının belirlenmesi,

- Geliştirmenin her aşamasında oluşan çıktıların standartlara, prosedürler ve yönergelere uygun olup olmadığını garanti etme,

- Eksiksiz ve makul olup olmadığını kontrol etmek için projenin belgeleri değerlendirme.

Veri Bütünlüğü Testi:

Veri bütünlüğü testi**,** veritabanlarında veya veri ambarlarında tutulan verilerin doğruluğunu, bütünlüğünü, tutarlılığını, işlevselliğini, kalitesini ve yetkilendirmesini inceleyen manuel/otomatik bir dizi temel testtir. İki yaygın türü vardır:

İlişkisel bütünlük testleri - Veri öğesi ve kayıt tabanlı seviyelerde yapılan veri doğrulama rutinleridir.

Bilgi tutarlılığı testleri - VTYS (DBMS) tarafından korunması gereken bir veritabanının farklı tablolarındaki varlıklar arasındaki varlık ilişkilerinin tanımlanmasıdır.

Yazılım Testi:

Yazılım testleri, kullanıcı gereksinimlerinin doğrulanmış olduğunu, sistemin beklendiği gibi çalıştığını ve dahili kontrollerin amaçlandığı şekilde çalıştığını belirler. Bu sayede yazılımdaki hatalar bulunup düzeltilebilir ve gereksinimlere uygun hale getirilebilir. Yazılım testi ise bir yazılımın sonsuz sayıdaki çalışma alanından, sınırlı sayıda ve uygun şekilde seçilmiş testler ile beklenen davranışlarını karşılamaya yönelik, dinamik olarak yapılan doğrulama faaliyetlerini kapsamaktadır.

Yazılım testi, bir anlamda maliyet, zaman, kalite kıstaslarının optimize edilmesidir.

Bu teste iki ana yaklaşım bulunur:

1. Aşağıdan-yukarıya- Her bir birimin testi ile başlar. Birim testleri başarıyla tamamlanan alt düzey modüller birbirleriyle entegre edilir. Bu entegrasyondan sonra alt modüller ile ilgili entegrasyon testleri yapılır. Bu testlerin başarılı şekilde sonlandırılmasından sonra bir üst katman modülleri sistemi entegre edilir ve 44 gerekli entegrasyon testleri yapılır. Bu entegrasyon tüm sistem inşa edilinceye ve entegrasyon testleri başarıyla sonuçlanıncaya kadar devam eder. Bu yaklaşımda temel şart, entegre edilecek modüllerin birim testlerinin başarıyla tamamlanmış olmasıdır. Bu yaklaşım ile hedeflenen sistemin yüzdelik olarak ne kadarının tamamlandığı ve test edildiği kolay bir şekilde belirlenebilir.

2. Yukarıdan-aşağıya- Öncelikle sistem için en üst modülünün (en dış modül, genellikle kullanıcı grafik arayüz modülü) test edilmesiyle başlar. Bu modülün ihtiyaç duyduğu alt modüllerinin koçanları kullanılır. Bu yaklaşımda birçok koçan yazılması gerekir. En üst modül başarılı şekilde test edildikten sonra bir alt düzey modüller sistemle bütünleştirilir (sce.uhcl.edu). Bu entegrasyondan sonra gerekli alt düzey tümleştirme testleri yapılır. Bu durum en alt düzey modüller entegre edilinceye ve entegrasyon testleri başarıyla sonuçlanıncaya kadar devam eder. Bu yaklaşımda entegrasyon testleri kara kutu testleri olarak başlar ve ilerledikçe saydam kutu testlerine döner.

Bilgi sistemleri denetçisinin bu testlerde dikkat etmesi gereken hususları:

- Hata raporlarını kayıt ve takip için kullanılan prosedürleri gözden geçirmek,

- Sistem güvenliğinin tasarlandığı gibi çalıştığını doğrulamak,

- Doğruluğu için döngüsel işleme ve kritik raporları doğrulamak,

- Paralel test sonuçlarını ve kullanıcı kabul testini gözden geçirmek,

- Sistemin son kullanıcıları ile yeni yöntemleri, prosedürleri ve kullanım talimatlarını anladıklarını belirlemek için görüşmek,

- Test planını, hata raporlarını, son kullanıcı belgelerini ve eksiksizlik ve doğruluk için kullanılan prosedürleri gözden geçirmek,

- Dahili kontroller için testlerin planlanıp gerçekleştirildiğini belirlemek için birim ve sistem test planlarını gözden geçirmek,

- Kullanıcı kabul testini incelemek ve kabul edilen yazılımın uygulama ekibine teslim edildiğinden emin olmak (Satıcı bu sürümü değiştirememelidir.),

- Kontrol toplamlarını ve dönüştürülmüş veriyi toplamak.

2.1.8. Üretim Ortamına Aktarım

Karmaşık BT sistemlerinin etkin ve verimli şekilde geliştirilmesi ve bakımı, bir işletme içinde titiz yapılandırması, değişiklik ve sürüm yönetimi süreçlerinin uygulanması ve uyum sağlanmasını gerektirir. Bu süreçler, sistemi oluşturan BT bileşenlerinin nitelikleri (donanım, yazılım, bellenim ve fiziksel bağlantı ortamı kablosu, fiber, radyo frekansı [RF] dahil olmak üzere ağ bağlantısı) üzerinde sistematik, tutarlı ve açık bir kontrol sağlar. Bilgi işlem ortamlarının yapılandırma durumu bilgisi, bu sistemlerin zamanında bakımının sağlanmasının yanı sıra sistem güvenilirliği, kullanılabilirliği ve güvenliği için kritik öneme sahiptir. BT sistemlerindeki değişiklikler, iş süreçlerinde istenmeyen sonuçları en aza indirmek için dikkatlice değerlendirilmeli, planlanmalı, test edilmeli, onaylanmalı, belgelenmeli ve iletilmelidir.

Bilgi sistemleri denetçisi, yapılandırma, değişiklik ve sürüm yönetimini yönetmek için mevcut araçlardan ve geliştirme personeli ile üretim ortamı arasında SoD'yi sağlamak için mevcut kontrollerden haberdar olmalıdır.

Kontrol etme süreci; donanım, ağ ve sistem mimarlarının hem donanım varlığı hem de envanter izleme sistemlerinde yapılan değişiklikleri veya güncellemeleri gözden geçirmesi ve onaylamasıyla paralel olarak kod düzenlemelerini önler veya yönetir.

Check-in (giriş), bir öğeyi kontrollü ortama taşıma işlemidir. Bir değişiklik gerektiğinde (ve bir değişiklik kontrol formu tarafından desteklendiğinde), konfigürasyon yöneticisi öğeyi kontrol eder Değişiklik yapıldıktan sonra farklı bir sürüm numarası aracılığıyla kontrol edilebilir. Teslim alma işlemi aynı zamanda eş zamanlı kod düzenlemelerini de önler veya yönetir. Donanım ile ağ ve sistem mimarları hem donanım varlığı hem de envanter takip sistemlerinde yapılan değişiklikleri veya güncellemeleri inceler ve onaylar.

Yapılandırma yönetiminin çalışması için yönetim desteği çok önemlidir. Konfigürasyon yönetimi süreci, bir konfigürasyon yönetim planı ve işletim prosedürleri geliştirip takip ederek uygulanır. Bu plan sadece geliştirilen yazılımla sınırlı kalmamalı, tüm sistem döokümantasyonunu, test planlarını ve prosedürlerini de içermelidir.

Özetle iş surecilerinde istenmeyen sonuçların en aza indirilmesi için BT sistemlerindeki değişiklikler dikkatlice değerlendirilmeli, planlanmalı, test edilmeli, onaylanmalı, belgelenmeli ve anlatılmalıdır. Bilgi sistemleri denetçisinin, geliştirme personeli ile üretim ortamı arasında görevler ayrılığını (SoD) sağlamak için konfigürasyon, değişim ve sürüm yönetimi ve mevcut kontrollerin yönetimi için mevcut araçların farkında olması gerekir.

Veri Taşıma:

Veri dönüştürmenin amacı, verilerin anlam ve bütünlüğü korunurken, sistemdeki mevcut verilerin yeni formata, kodlamaya ve yapıya dönüştürülmesidir. Veri dönüştürme işlemi, dönüştürülen verilerin doğruluğunun ve eksiksizliğinin doğrulanmasına izin veren denetim izleri ve günlükler gibi bazı araçlar sağlamalıdır. Bu doğruluk ve eksiksizliğin doğrulanması, manuel süreçlerin, sistem yardımcı programların, satış araçlarının ve tek seferlik özel uygulamaların kombinasyonuyla gerçekleştirilebilir.

Veri taşıma projesi dikkatli bir şekilde planlanmalı ve aşağıdaki riskleri en aza indirmek için uygun metodolojiler ve araçlar kullanmalıdır:

- Eski ve taşınmış işlemler arasında yaşanan çatışmalar ve anlaşmazlıklar,
- Verilerin tasıma sürecinde yaşanan veri tutarsızlığı ve bütünlüğü kaybı,
- Rutin işlemlerin kesintisi,
- Verinin güvenlik ve gizlilik ihlali.

Uygulamaya Almanın Planlanması:

Üretim ortamını kurma adımlarının her biri, kimin sorumlu olacağı, adımın nasıl doğrulanacağı ve geri dönüş prosedürü dahil olmak üzere belgelendirilmelidir. Gerçekleştirme tarihinden çok önce bir gerçekleştirme planı hazırlanmalıdır. Başarılı testten sonra, sistem işletmenin değişim kontrol prosedürlerine göre gerçekleştirilir. Bir geri dönüş senaryosu gerçekleştirilmelidir.

Geçiş (Canlıya Geçiş veya Sistem Geçişi) Teknikleri:

a. Paralel Geçiş

Paralel geçiş tekniği, eski sistemin çalıştırılması, ardından hem eski hem de yeni sistemlerin eş zamanlı çalıştırılmasıdır. Bu yaklaşım 2 sistemin karşılaştırılmasına olanak tanır. Yeni sistemin çalıştığına dair güven kazanıldıktan sonra tamamen yeni sisteme geçilir İki sistemin aynı anda çalışmasının maliyeti yüksektir ancak risk daha düşüktür.

b. Fazlı (Aşamalı) Geçiş

Fazlı (aşamalı) geçiş tekniği, alt sistemlerin sırayla yeni sisteme geçişidir. Eski sistemden yeni sisteme sırayla modül bazında geçiş sağlanır. Son modüle ulaşana kadar aşamalı olarak kaldırılır.

c. Ani Geçiş

Ani geçiş tekniğinde, yeni sistem, bir son tarih ve saatte eski sistemden değiştirilir ve yeni sisteme geçiş gerçekleştiğinde eski sistem durdurulur. Genel olarak eski sistem görevini yerine getiremediğinde tercih edilir. Ani geçiş yaklaşımının riski yüksektir. Eğer sistem istenilen verimi elde edemezse geri dönüş maliyeti çok yüksek olacaktır.

Bilgi sistemleri denetçisi, uygulamaya alınmadan önce uygun onayların alındığını doğrulamalı ve aşağıdakileri yapmalıdır:

- Sistemi üretime almadan önce doğru ve tam olduklarından emin olmak için tüm veri dönüşümünü doğrulamak,

- Tamlığını sağlamak ve test aşamasındaki tüm güncellemelerin dahil edildiğinden emin olmak için tüm sistem belgelerini incelemek,

- Üretim zamanlamasının yürütülmesinde kullanılan sistem parametrelerinin yanı sıra, sistemi zamanlamak ve çalıştırmak için kullanılan programlanmış prosedürleri gözden geçirmek.

Devreye Alma Planı:

Sürümün devreye alınması için aşağıdaki aşamalar gerçekleştirilir:

- Rollerin belirlenmesi,

- Gerekli yetenek/becerilerin kazanılması ve gerekli eğitimlerin verilmesi,

- İş yükünün rol ve sorumluluklara göre dağıtılması,

- Her ihtimal göz önünde bulundurularak fazlı bir geçiş planı oluşturulması.

Eğitim Süreci:

Projenin sisteme getirdiği yenilikler kapsamında eğitim ihtiyaçları doğabilmektedir. Bunun yanı sıra rol ve sorumlulukların da bu kapsamda gözden geçirilmesi gerekmektedir. Takibinde bir personelin eğitim planı hazırlanırken aşağıda yer alan tüm detayların bulunması beklenmektedir:

- İçerik,

- Zaman planı bilgisi,

- Süre,

- Teslim şekli,

- Eğiticiyi eğitme kavramı.

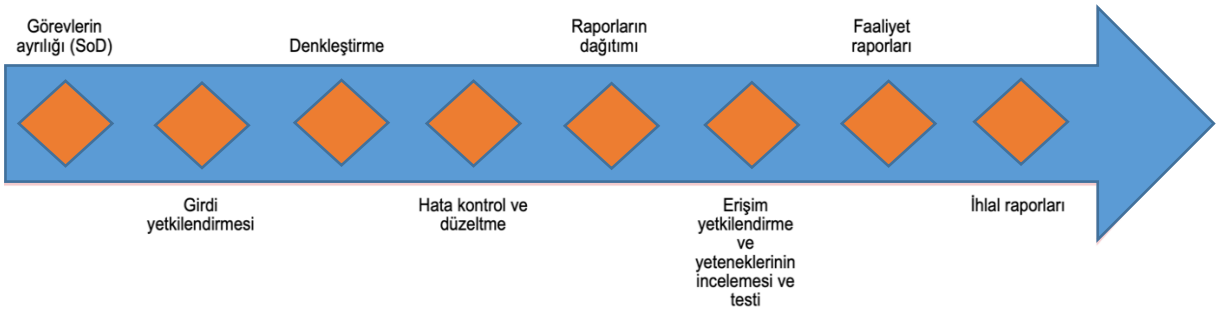
Bu plan yeni yapı için rol tanımlarını beceri profillerini ve boşluk analizi sonuçlarını dikkate almalıdır. Plan hazırlanırken eğitilmesi gereken personelin mevcut sistemi çalıştırması gerektiği de dikkate alınarak iş gücü kaybına neden olmayacak şekilde bir koordinasyon yürütülmelidir.

Geliştirme sürecinde yer alan tüm birimler, her zaman son kullanıcıdan birkaç adım önde olmalıdır. Bir sonraki adımda; uygulamanın çoklu ortamlara, internet ortamına veya başka etkileşimli ortamlara taşınması istenebilir. Böylece, belki başlangıçta düşük maliyetlerde hesaplanan uygulama, değerini de arttıracak teknik özelliklere sahip olabilir.

Son Kullanıcı Eğitimi

Bu eğitimin amacı son kullanıcının yetkinliği artırılarak sistemi daha etkin ve verimli kullanabilmesini sağlamaktır. Bu sürecin başarılı olabilmesi için geliştirme sürecinden başlanarak bir eğitim planı hazırlanmalıdır. Bu şekilde geliştirilecek bir eğitim stratejisinde zamanlama, kapsam ve dağıtım mekanizmaları önem arz etmektedir.

Eğitim için öncelikle bir pilot uygulama yapılabilir daha sonra pilot gruptan gelen geri bildirimler ışığında eğitim üzerinde düzenlemeler yapılıp ve eğitim planında da ayarlamalara gidilebilir. Eğitim verme de zamanlama çok önemlidir. Eğitim erken verilirse kullanıcılar sistem üretime geçene kadar çoğunu unuturlar. Eğitim geç verilirse pilot grubun geri bildirimleri ile eğitimi olgunlaştıracak zaman kalmaz ve eğitimde istenilen başarı elde edilemez. Ayrıca eğitimler kullanıcıların organizasyon içindeki rollerini, beceri düzeylerini ve ihtiyaçlarını karşılayacak şekilde düzenlenmelidir.



Kullanıcı Prosedürleri ve Dokümantasyon Şeması

Sistemin son kullanıcı ve sistem yöneticileri için hazırlanan kılavuzları kapsamaktadır. Geliştirme hakkında bilgi verir, gereksinimleri, teknik özellikleri, iş gerekçelerini ve kılavuzları kullanıcı belgelerini, sorun giderme kılavuzlarını, kurulum ve başvuru kılavuzlarını içerir.

Bilgi sistemleri denetçisi tarafından gözlemlenmesi ve test edilmesi gereken kullanıcı prosedürleri ve dokümantasyonlar aşağıda verilmektedir.

• **SoD** - Hiç kimsenin aşağıda belirtilen süreçlerden birden fazlasını gerçekleştirmeye yetkili olmamasını sağlar:

- Başlatma,

- Yetkilendirme,

- Doğrulama/dağıtım.

İş tanımlarının gözden geçirilmesi, yetki seviyeleri ve prosedürlerin gözden geçirilmesi, SoD'nin varlığı ve uygulanması hakkında bilgi sağlayabilir.

• **Giriş yetkisi** - Girilen belgelerde tanımlı yetkilendirme veya benzersiz parolalar kullanılması gerekir. Uygun yetkilendirme girdi belgelerinin bir örneğine bakılarak veya bilgisayar erişim kuralları gözden geçirilebilir. Veri doğrulamaları takip edilmeli ve geçersiz işlemler incelenerek otomatik günlüğe kaydetmesi sağlanmalıdır. Bu geçersiz işlem faaliyet raporu, yönetimsel incelemenin kanıtı için test edilmelidir. Aşırı geçersiz işlemler, verimliliği artırmak için doğrulama ve düzenleme rutinlerinde değişiklik yapılması gerektiğini gösterebilir.

• **Denkleştirme** – Çalıştırılan kontrol toplamları ile ve diğer uygulama toplamlarının zamanında uygunluğu doğrulanmalıdır. Bu, bağımsız denkleştirmelerle veya geçmiş uzlaşmaları gözden geçirilerek test edilebilir.

• **Hata denetimi ve düzeltme** - Düzeltme ve yeniden gönderme şeklinde uygun inceleme kanıtlarını içeren rapor üretilmelidir. Giriş hataları ve "*red*"ler yeniden gönderilmeden önce gözden geçirilmelidir. Yapılan düzeltmelerin yönetim tarafından incelenmesi ve yetkilendirilmesine ilişkin kanıtlar dokümante edilmelidir. Bu eforun testi, geçmiş hata düzeltmelerini yeniden düzenleyerek veya gözden geçirerek gerçekleştirilebilir.

• **Raporların dağıtımı** - Kritik çıktı raporları güvenli bir alanda üretilerek saklanır ve yetkili bir şekilde dağıtılır. Dağıtım süreci, dağıtım çıktı günlükleri gözlemlenerek ve gözden geçirilerek test edilebilir. Çevrimiçi çıktı raporlarına erişim kısıtlanmalıdır. Çevrimiçi erişim kuralları incelenerek veya kullanıcı çıktısı izlenerek test edilebilir.

• **Erişim yetkileri ve yeteneklerinin incelenmesi ve testi** - Erişim seviyeleri hakkında bilgi sağlar (kontrol tabloları). Erişim, iş tanımlarına dayalı olmalı ve SoD'yi sağlamalıdır. Test erişiminin yönetimin istediği gibi verildiğinden emin olmak için kuralların gözden geçirilmesi yolu ile gerçekleştirilebilir.

• **Faaliyet raporları** - Kullanıcı bazında etkinlik hacmi ve çalışma saatleri hakkında ayrıntılar sağlamaktadır. Faaliyetlerin yalnızca yetkilendirilmiş saatlerde yapıldığına güvence vermek için düzenli olarak gözden geçirilmelidir.

• **İhlal raporları** - Başarısız ve yetkisiz erişim girişimlerini terminalin konumu erişim girişim tarihi ve saati gibi detayları içerir şekilde gösterir. Bu raporlar yönetimin gözden geçirmesine yönelik kanıt dokümanı olarak değerlendirilebilir. Tekrarlanan yetkisiz erişim ihlalleri, erişim kontrollerini atlama girişimlerini gösterebilir.

2.1.9. Kritik Başarı Faktörleri

Kritik başarı faktörleri, iş verimliliği ekonomik değer, müşteri hizmetleri ve kaliteyi içermektedir. Gereksinimler için, yöntemlerin aktarılabileceği bir şablon oluşturması SDLC'nin sağladığı temel avantajıdır. SDLC'nin kritik başarı faktörleri aşağıdakileri içermelidir:

- Proje başlangıcında iş için gerekli dokümanların temin edilmesi,

- Tüm kararların alınabileceği karar adımlarında, karar vericiler yer almalı,

- Yetkili personelin komitelere katılması ve proje yaşam döngüsü boyunca dahil olması sağlanmalı, böylece gecikmelerin önlenmesi sağlanabilir.

Kritik Başarı Faktörleri Ölçümleri:

SDLC kapsamında kritik başarı faktörleri için aşağıdaki hususlarda yapılacak ölçümler takip edilebilecektir.

a. Üretkenlik

- Her kullanıcı için harcanan para,

- Kullanıcı bazında yapılan işlemler için hareket sayısı,

- Aylık toplam işlem sayısı.

b. Kalite

- Anlaşma sağlanamayan konuların sayısı,

- Kötüye kullanım ya da sahtecilik kullanım sayısı,

- Tutarsızlık sayısı.

c. Ekonomik Değer

- Yönetim maliyetleri.

d. Müşteri Hizmetleri

- Müşteri problemleri ile ilgili müşteriye geri dönüş süresi,

- Kullanıcılar ile iletişim periyodu.

2.1.10. Sistem Geliştirme Süreçlerindeki Riskler

Potansiyel Riskler:

Sistem geliştirme süreçleri için süreci tasarlarken ve geliştirirken birçok potansiyel riskle karşılaşılabilir. Bu riskler aşağıdaki gibi özetlenebilir:

a. İş Riski: Değişiklik sonrasında oluşturulan yeni sistemin kullanıcı iş ihtiyaçlarını, gereksinimlerini ve beklentilerini karşılayamama olasılığıdır.

b. Proje Riski: Proje faaliyetlerinde belirlenen tasarım ve geliştirmelerin dışına çıkmasıdır.

c. Tedarikçi Riski: Gereksinim ve beklentilerin net olarak iletilmemesi, tedarikçilerin geç teslimat yapmasına neden olur.

d. Paydaşlar Riski: Gerekli girdilerin sağlanamamasına neden olur.

e. Teknoloji Riski: Teknoloji altyapısına uyumsuz ve verimsiz olunmasıdır.

Bilgi sistemleri denetçisi tarafından takip edilmesi gereken konular aşağıdaki gibidir:

- Geliştirme ve tasarım adımlarının incelenmesi, gözden geçirilmesi,

- Proje aşamalarında risk analizlerinin yapılması ve periyodik kontrollerinin yapıldığına ilişkin kayıtların kontrol edilmesi,

- Proje hedef ve amaçlarının incelenmesi.

2.2. Sistem Bakım ve Destek Süreçleri

2.2.1. Sistem ve Uygulama Bakımı

Uygulamalardan sonra sistemler için sürekli geliştirme ya da bakım süreçleri başlatılır. Üretim, uygulama ve çalıştırılabilir kodun bütünlüğünü korurken, uygulama sistemlerinde değişiklik yönetimi süreci uygulanır. Proje tasarım aşamasında, değişikliklerin kayıt altına alınması ve gerçekleştirilmesi için standart bir değişiklik yönetimi süreci olmalıdır. Bu kapsamda bilgi sistemleri denetçisinin yapması gereken işlemler aşağıdaki gibidir:

- Sistemde mevcut olan kontrolleri gözden geçirerek, tasarıma göre çalıştığına teyit alınması,

- Yönetime doğru raporların ulaşıp ulaşmadığını teyit etmek için, maliyet faydalarının analizlerinin kontrol edilmesi,

- Giriş ve çıkış kontrol raporlarını gözden geçirerek, sistemin doğru çalıştığına emin olunması,

- Hedeflenene ve gereksinim duyulan işlemlerin gerçekleşip gerçekleşmediğinin kontrol edilmesi,

- Gerçekleştirilen değişikliklerin gözden geçirilerek, gereksinim duyulan değişiklikler ile uyumunun teyit edilmesi,

- Operatörlerin hata günlüklerini inceleyerek, sistemde kaynak ya da işletme sorununun olup olmadığının tespit edilmesi.

Talep/Hata Yönetimi:

Gereksinimlerden elde edilen isterlere göre test senaryoları oluşturulur. Test senaryolarının sonuçları beklenen ile gerçekleşen arasında uyumsuzluk varsa hata olarak değerlendirilebilir. Karşılaşılan bu hatalar ile ilgili bir araç yardımıyla (Jira), sözlü ya da başka bir yöntem ile ilgililerine iletilmeli ve raporlanmalıdır. İletilen hatalar incelenir ve düzeltmeleri sağlandıktan sonra teste tabi tutulur. Hataya özel "*onay testi*" ve sistemde diğer alanları etkileyip etkilemediğini kontrol etmek için "*regresyon testi*" gerçekleştirilir. Hata raporları oluşturulurken, test araçları ya da dokümanlardan yararlanılabilir. Hataların tamamı giderilerek bitiş kriterlerine uygun hale gelmesi sonrasında test sonuç raporları hazırlanır.

a. Hataların Önem Derecesi (Severity)

Testlerde esnasında tespit edilen yazılım hatasının önem derecesi, sistemin çalışmasındaki hata etkisinin derecesini belirler. Tespit edilen hatalar *önem derecesine* göre ayrılır.

Düşük (Low) –> Testlerde ilerlemeye etkisi olmayan minör olarak değerlendirilen hatalardır. Test edilen öğe bu hatalar ile teslim edilebilir.

Orta (Medium) --> Testler devam eder. Bu hatalar ile teslim edilen ürün, daha sonra düzeltilebilecek hatalar içerebilir. Sistemin çalışmasını bozmayan hatalardır.

Yüksek (High) –> Sistemin çalışmasına etki eden fakat testlere devam etmeye engel olmayan hatalardır. Düzeltilmesi öncelikli olarak değerlendirilecek hatalardır.

Kritik (Critical) –> Testleri engelleyen ve çözülmesi mutlaka gerekli olan hatalardır.

b. Hataların Öncelik Seviyesi (Priority)

Testlerde tespit edilen hatalar analiz edilerek bir önceliklendirme yapılır ve karşılaşılan hatalar için öncelik seviyeleri belirlenir. Genel olarak testlerin devam etmesini engelleyen kritik seviyedeki hatalar için öncelik verilir. Hatalar için öncelik seviyesi aşağıdaki gibi olabilir.

Düşük (Low) –> Sistemin çalışmasına etki yaratmayan hatalardır.

Orta (Medium) –> Sistemin çalışmasını engellemeyen ancak düzeltilmesi gerekli ve teslimden sonra düzeltilebilecek hatalardır.

Yüksek (High) –> Sistemin işleyişini durdurmayan ama çalışmasını etkileyen hatalardır.

Kritik (Critical) –> Sistemin işleyişini ve çalışmasını engelleyen hatalardır. Düzeltilmesi acil ve önemli hatalardır.

Gereksinimlere göre hazırlanan test senaryolarında beklenen ile gerçekleşen durumların uyumsuz olduğunda oluşan hataları yukarıda Önem ve öncelik seviyelerine göre hata detayları oluşturulabilir. Proje ya da talep kapsamında hata açılacak ise belirtilmelidir. Hata özeti açık ve anlaşılır olacak şekilde eklenir.

c. Hata Kaydı Açılması

Hata kaydı açılması aşamasında, hatalar için öncelik seviyesi belirlenir ve hata açıklaması detaylı ve açık bir şekilde girilir. Hata açıklamasında test senaryosunda kullanılan data, ortam bilgileri de girilmelidir. Test senaryosu girilir. Beklenen ve gerçekleşen durum belirtilir. Ekran görüntüsü ve alınabilirse video ile desteklenir. Hata oluşturulur ve düzeltecek kişi ataması yapılır. Hatalar takip edileceği, JIRA (Atlassian tarafından geliştirilen ve hata takibi ve çevik proje yönetimi sağlayan hata izleme aracı), TestRail, HP ALM gibi araçlardan faydalanılabilir ya da excelde tablo oluşturularak mail yoluyla geliştiricilere bildirilir.

Testler sırasında karşılaşılan hataların açılması ile düzeltilmesine kadar süreç aşağıdaki gibi izlenebilir. Testleri yapan çalışanlar tarafından tespit edilen hataların öncelikle önem derecesi belirlenir. Kullanılıyorsa bir araç yardımıyla Hata kaydı oluşturulur. Hatayı çözecek kişiye atanır. Testleri engelleyen durum olması halinde çözüm beklenir. Engelleyen bir durum yoksa diğer test senaryolarına devam edilir. Hatanın atandığı kişi çözüm için çalışma başlatır. Analiz yapılır. Hata olduğu kabul edilmeyen durumlarda yazılımcı tarafından hata reddedilir. Hatanın kabul edilerek düzeltildiği durumda tekrar test yapılması için testi yapan kişiye atanır. Hata düzeltilmiş ve testi başarılı olan hata kaydı kapatılır. Test başarısız sonuçlanırsa hata kaydı yeniden açılır ve düzeltilmesi için süreç yeniden yönetilir.

Program Değişikliklerinin Denetimi ve Acil Değişiklikler:

Bilgi sistemleri denetçisi tarafından, uygulamanın tamamlanması sonrasında aşağıdaki hususları dikkate almalıdır:

- Acil durum değişiklikleri ile ilgili prosedürlerin yeterliliği ve uygulama yönetimi,

- Kullanıcı taleplerine yönelik gerçekleştirilen geri dönüşler (zaman ve maliyet) için memnuniyet ölçüm kayıtları,

- Güvenlik erişim kısıtlamalarının, üretim kaynak ve çalışabilirlik modülleri üzerindeki etkinliği ve yeterliliği,

- Acil durum değişiklik yönetimine ait prosedürleri,

- Acil durum oturum açma kimlikleri ile ilgili güvenlik erişim kısıtlamalarının yeterliliği,

- Değişiklik kontrollerinin, kullanıcı ve geliştirici grupları için yeterli bir prosedürün olup olmadığının kontrol edilmesi,

- Kullanıcı değişiklik taleplerinde yetkilendirme, önceliklendirme ve izlemek için önceden belirlenmiş metodolojilerin kullanımı ve yeterliliğinin değerlendirilmesi,

- Değişiklik kontrol günlüğünde gösterilen tüm değişikliklerin çözülüp çözülmediğinin kontrol edilmesi.

Bilgi sistemleri denetçisi, onaylama, müdahale süresi, yanıt etkinliği ve süreçten kullanıcının memnuniyeti konusundaki olası iyileştirmeler için tüm değişim yönetimi surecini gözden geçirmelidir.

Geri Dönüş Prosedürleri:

Başarılı tamamlanan testler sonrasında değişim kontrol prosedürlerine göre sistemin kuruluşu gerçekleştirilmelidir. Sistem kuruluşu yapılmadan önce kuruluş sürecinin nasıl yapılacağına ilişkin bir gerçekleştirme planı hazırlanmalıdır. Kuruluş süreç adımları tek tek belirlenmeli ve sorumluları eklenmelidir. Adımların nasıl doğrulanacağı ve geri dönüş planlarına ilişkin prosedürler oluşturulmalıdır. Burada geri dönüş senaryoların gerçekleştirilmesi çok önemlidir.

Uygulanan değişikliğin başarısız olması durumunda, geri döneceği sürümler (bilinen sonuçlar) referans alınmalıdır. Geri dönüş prosedürlerinin başarılı olması için, bu sürecin başarılı işletilebilmesi gerekir.

Bilgi sistemleri denetçisi tarafından kullanıcının değişiklik taleplerinin geri dönüşü (zaman ve maliyet) için memnuniyeti değerlendirilmesi gerekmektedir.

Uygulama Sonrası Gözden Geçirme ve Öğrenilmiş Dersler:

Proje işletme yararlarını ve maliyetlerini gerçekleştirmek, projenin genel başarısını ve iş birimleri üzerindeki etkisi görüldükten sonra gözden geçirme yapılabilir. Bu aşamada ölçütler aşağıdakileri içerir:

- Toplam sahip olma maliyeti (TCO),

- Yatırım getirisi (ROI).

Bilgi sistemleri denetçisi tarafından uygulama sonrası gözden geçirme sırasında aşağıdaki hususları dikkate almalıdır:

- Kontroller gözden geçirilmeli ve tasarıma göre çalıştığından emin olunmalı,

- Yönetime doğru raporların ulaşıp ulaşmadığını teyit etmek için, maliyet faydalarının analizlerinin kontrol edilmeli,

- Operatörlerin oluşturduğu hata günlükleri incelenerek, kaynak veya işletme sorunu yaşanıp yaşanmadığı tespit edilmeli,

- Sistemin hedeflerine ve gereksinimlerine ulaşılıp ulaşılmadığı kontrol edilmeli,

- Giriş ve çıkış kontrol raporlarını gözden geçirerek, sistemin doğru çalıştığına emin olunmalı,

- Gerçekleştirilen değişikliklerin gözden geçirilerek, gereksinim duyulan değişiklikler ile uyumunun teyit edilmeli.

2.3. Uygulama Kontrolleri

Uygulama kontrolleri ilgili iş süreçlerinin bilgi sistemlerine, girişinden çıkışına kadar, organizasyonel hedeflere katkı sağlayarak tüm süreç boyunca tam ve doğru bir şekilde işlevselliği yerine getirmesi amacıyla tasarlanmış kontrollerdir.

Verilerin sistemlere girilmesiyle başlayan uygulama kontrolleri; yetkilendirmelerin, gizlilik, bütünlük ve tutarlılıklarının kontrol edilmesi, oluşan hataların tespit edilmesi ve raporlanması, veri işleme, hesaplama, karşılıklı anlaşmaların sağlanması ve oluşan çıktıların kontrol edilmesi gibi kritik kontroller sağlar. Uygulama kontrolleri ile aynı zamanda hatasız ve planlanan sonuca ulaşmak sağlanır. İstisnai durumlar ve gerçekleşen hataların kök nedenlerinin araştırılarak otomatik kontroller ile prosedürlerin birleştirilmesi ile aksiyonlar alınabilir.

Uygulama kontrolleriyle;

- Bilgisayar sistemlerine sadece doğru, geçerli ve tam olacak şekilde verilerin girilmesi,
- Belirlenmiş olan doğru görevleri yerine getirmesi,
- Kayıt altına alınan verilerin korunması,
- Gerçekleşen sonuçların beklentileri karşılaması,

sağlanır.

Uygulama kontrolleri 3 gruba ayrılır. Bunlar:

- Girdi kontrolleri,
- İşlem kontrolleri,
- Çıktı kontrolleri.

Uygulama kontrolleri kapsamında bilgi sistemleri denetçisi aşağıdaki hususlara dikkat etmelidir:

- Verimlilik ve etkinliğin sağlanabilmesi için uygulamalardaki operasyonel taraflara dikkat etmek,
- Kritik uygulama bileşenlerini ve iş adımlarını belirlemek,
- Test stratejisi belirlemek,
- Uygulama kontrol etkinliğini ve oluşan kontrol zayıflıklarının etkisini incelemek,
- Etkinliğin sağlanabilmesi için kontrolleri test etmek,
- Planlanan kontrol hedeflerine ulaşılma başarısını tespit edebilmek için test sonuçlarını ve diğer denetim kanıtlarını analiz etmek.

2.3.1. Giriş/Kaynak Kontrolleri

Girdi kontrolleri, yapılan işlemlerin yönetim tarafından onaylandığını ve yetkilendirmelerin yönetim tarafından yapıldığını doğrular. Sadece geçerli sayılan ve yetki verilen verilerin girilmesini ve bu işlemleri sadece bir kez yapılmasına olanak sağlar.

a. Yetkilendirmeler

Yetkilendirme süreci ile aşağıdakiler sağlanır:

- İlgili belgelerde yer alan imzaların yetkili kişiler tarafından atılıp atılmadığının kanıtı,
- Basılı/dijital belgelerdeki kaynakların referans kontrolleri,
- Erişim yetkilerinde kısıtlamaları, birbirinden farklı parola yönetimini ve olası yetkisiz erişim durumlarında raporlamanın yapılması,
- Sadece yetki verilen kişiler tarafından, erişim izni verilen verilere erişim ve işlemleri yapabilmelerine imkân sunması,
- Her iş alanlarında yetkilerin ve iş adımlarının çalışan rol tanımlarına göre sınırlandırılması.

b. Toplu Kontroller ve Mutabakat

Toplu kontrollerin sağlanması için girdilerde işlem kontrolleri toplu yapılır ve kontroller sonucunda mutabakatlar sağlanır. Bu kapsamda kontrol edilen veriler:

- Toplam para miktarı,
- Dokümanların toplamı,
- Toplama ait özetler,
- Hesaplarda kontroller,

- İş kayıtlarının toplamı,

- Anlaşmalar,

- Oluşturulan toplam kalemler.

c. Hata Raporlanması ve Aksiyon Takibi

Girdi hatası işleme, sisteme sadece doğru verinin kabul edilmesini doğrular. Bu kapsamda aşağıdaki işlemler yapılır:

- Hatalı işlemler ret edilir, paketin (batch) geri kalanında işleme sürecine devam edilir.

- Hata olsa bile işleme sürecine devam edilir. Hata olan girdilerde sonradan hata düzeltme yapılması için işaretleme sağlanır.

- Hata içeren tüm paket (batch) düzeltilmesi için reddedilir.

- Hata içeren parti, düzeltilene kadar askıya alınır ve şüpheli işlem olarak bekletilir. Ret edilme işlemi uygulanmaz.

2.3.2. İşleme Prosedürleri ve Kontrolleri

Uygulama işlemlerinin güvenirliğini sağlamak için işleme prosedür ve kontrolleri takip edilir. İşlem kontrolleri, girilen veriler ile istenen çıktıları üretecek şekilde işlenmesini sağlar. Tüm işlemlerin, güvenlik ilkeleri çerçevesinde gerçekleştirilmesine dikkat edilmelidir.

a. Veri Doğrulama ve Düzenleme Prosedürleri

Sisteme girilecek verilerin hatasız olması için, veri doğrulama ve düzeltme prosedürleri kullanılır. Bu prosedürler ile eksik, hatalı ya da ilgisiz verilerin tespit edilmesi sağlanır. Takip edilen prosedürlere yönetici tarafından manuel müdahaleler varsa, bu müdahalelere ilişkin kayıtların tutulması ve periyodik olarak gözden geçirilmesi sağlanmalıdır. Veri doğrulama ve düzenleme prosedürlerine aşağıda yer verilmektedir.

- Sıra kontrolü: İşlemlere ait kontroller bir sırayı takip etmelidir. Tekrar eden numara ya da beklenmeyen kontrol numaraları ile karşılaşılması durumunda ret edilir ya da istisna olarak kayıt altına alınır. Örneğin oluşan faturalar sıra numarasına göre kayıt altına alınır. İlgili günün ilk faturası belirli bir sayı ile başlar ve yine aynı şekilde belirli bir numara ile bitmesi beklenir. Eğer arada farklı numaralar ile karşılaşılırsa, faturaya ait hatalı girdi olarak kabul edilir.

- Limit kontrolü: Veriler belirlenmiş bir miktara kadar girilebilecektir. Örneğin maaş verileri X TL'yi geçmeyecektir. Bu rakamın üzerinde bir maaş verisi girildiğinde sistem tarafından reddedilecektir.

- Aralık kontrolü: Veriler belirli bir değer aralığında olmalıdır. Maaş örneğinden gidilirse, en düşük maaş ile en yüksek maaş arası olmalıdır şeklinde bir tanımlama yapılabilir. Bu durumda belirlenen değerlerin dışı geçersiz kabul edilecektir.

- Geçerlilik kontrolü: Daha önceden belirlenen kriterlere uygun formatta veri girişi beklenmektedir. Örneğin maaş bilgisi alanında medeni durum ile ilgili veri girişi yapılacak alana sadece E veya B harfleriyle giriş imkânı tanınır. Başka bir sembol, rakam harf girişi yapılamaz.

- Makullük kontrolü: Veriler önceden belirlenen, makul değerler sınırı içerisinde kalmalıdır. Bunun dışında veri girişi yapılması durumunda hata mesajı gelecektir. Maaş örneğinden gidecek olursak, veri girilecek alana milyon gibi bir rakam girildiğinde sistem hata verecektir veya ilgili alana, beklenenden çok farklı bir rakam girilmesi ile ilgili hata mesajının oluşmasına neden olacaktır.

- Tablo bakma listeleri: Önceden tanımlanan kriterlere ait liste ile girdi olarak sağlanan verilerin uyumlu olması gerekmektedir. Listede yer almayan değerlere ait veri girişi yapılması engellenir.

- Mevcudiyet kontrolü: Önceden belirlenen kriterlere göre veri girişleri sağlanır. Örnek olarak; doğum tarihi, telefon numarası vb. verilebilir.

- Anahtar doğrulaması: Tuş vuruşları ile doğru kişi tarafından verinin giriş yapıp yapmadığı doğrulanır. En farklı kontrol yöntemlerinden birisidir.

- Kontrol basamağı: Matematiksel olarak bir değer hesaplanır ve aktarılacak verilere eklenir. Aktarım sırasında verinin bütünlüğünün bozulmasına yönelik bir hata olup olmadığı anlaşılır.

- Bütünlük kontrolü: Veri giriş alanlarında her zaman 0 ya da boşluktan farklı girişlerin yapılması beklenir. Örneğin, işe yeni giren bir personelin T.C. kimlik no alanının dolu olması gerekmektedir. Boş olması durumunda giriş ret edilir.

- Mükerrerlik kontrolü: Geçmişte yapılan girişler ile yeni yapılan girişler karşılaştırılarak mükerrer giriş olması engellenir. Örneğin, tedarikçiye ait fatura numarası ile önceden ödemesi tamamlanan faturalar kontrol edilir ve iki kez ödeme yapılmasının önüne geçilir.

- Mantıksal ilişki kontrolü: Bir ya da daha fazla koşulu sağlayan veriler doğru kabul edilir. Örneğin, tam zamanlı çalışma kapsamında çalışan profilinde 25 yaş sınırı varsa, işe alınma tarihinin doğum tarihinden en az 25 yıl geçmiş olması beklenir.

b. İşleme Kontrolleri

Biriken verilerin doğruluğunu ve eksizliğini doğrulamak için kullanılır. Yetki kapsamında gerçekleşen işlemler dışında veritabanında bulunan verilerin doğru ve eksiksiz kalması sağlanır. İşleme kontrollerine ana başlıkları itibarıyla aşağıda yer verilmektedir.

- Manuel yeniden hesaplama: Beklenen görevi yerine getirdiğinin doğrulanması adımıdır.

- Düzenleme: Veri girişlerinde, verilerin doğruluğu, eksizliği kapsamında hata olması durumunda kabul edilmeyerek bir otomasyon aracı tarafından veriyi giren kişiye yönlendirilmesidir.

- Çalıştırmadan çalıştırmaya kontrol toplamları: İşleme süreci boyunca veri değerlerinin verifikasyonudur. Okunan veriler süreç içerisinde güncellenir.

- Programlanmış kontroller: Veri ve işlemdeki hataları tespit eden ve düzeltici faaliyetleri otomatik olarak başlatan programlardır. Örneğin, hatalı veriler işlenmek üzere girilmişse, uygulama program uygun verilerin kullanılması gerektiğini bildiren iletileri görüntüleyebilir.

- Hesaplanan tutarların makul olarak doğrulanması: Miktarlarda hesaplama sonucunda uygunluğu için doğrulama yapar. Uygunluk önceden belirlenen kriterlere göre yapılmaktadır. Uygun bulunmayan işlemler incelenmek üzere ret edilir.

- Miktarlar üzerindeki limit kontrolleri: Daha önceden belirlenen limitler ile tutarların doğruluğu kontrol edilir. Limitler dışında olan işlemler daha sonra incelenmek üzere ret edilir.

- Dosya toplamları mutabakatı: Mutabakat, geçmişe ait manuel tutulan hesaplara ilişkin kontrol kayıtları ya da bağımsız kontrol dosyaları kullanılarak yapılır.

- İstisna raporları: İstisna raporları önceden belirlenen kriterler dışında yer alan yanlış işlemler veya verilerle ilgili bilgi sistemi tarafından oluşturulur.

c. Veri Dosyası Kontrol Prosedürleri

Veri dosyası kontrol prosedürleri, depolanan veriler üzerinde sadece yetkili kişiler tarafından işlem yapılabilmesine olanak sağlar. Bu kontrol prosedürlerine aşağıda yer verilmektedir.

- Düzeltme önce ve sonrası imaj raporlaması: Dosyada yer alan veriler için ilgili işlem öncesi ve sonrası imajları alınır. Böylece gerçekleşen işlemlerin veriler üzerindeki etkisi anlaşılmaya çalışılır.

- Bakım hatası raporlama ve işleme: Tüm hata raporlarına ilişkin gerekli aksiyonların ve düzeltmelerin zamanlı olarak yapıldığının teyit edilmesi için kullanılır. Düzeltme işlemi yapacak kişi ile düzeltme işleminin doğruluğunu teyit eden kişilerin farklı olması, görevler ayrılığı ilkesi için kritiktir.

- Kaynak belge saklama: Kaynak dokümanlar, işleme tabi tutulan verilerin geri getirilmesi, düzeltilmesi, teyit edilmesi için belirli bir süre tutulmalıdır. Bu saklanan verilere sadece yetkili kişiler iş ihtiyaçları çerçevesine erişebilmelidir. Bu belgelerin imha edilme süreci de güvenli ve kontrollü bir şekilde gerçekleştirilmelidir.

- İç ve dış etiketleme: Çıkarılabilir depolama araçları/medyaları için iç ve dış etiketleme yapılmalıdır. Bu yöntem ile uygun verinin işlem için kullanıldığı teyit edilir. Dış etiketleme ile doğru medya aracının kullanıldığı, iç etiketleme ile de bu medyadaki ilgili veri dosyalarının kullanıldığı teyit edilmiş olur.

- Versiyon kullanımı: Verilerin en güncel versiyonlarının veya istenilen versiyonlarının kullanılmasını sağlar. Örneğin, işlemler verilerin en güncel versiyonu ile yapılır ancak geri döndürme adımları verilerin eski versiyonu üzerinden gerçekleştirilir.

- Veri dosyası güvenliği: Yetkisiz kişiler tarafından yapılacak erişimler engellenir, ancak yetkili kişiler tarafından yapılan hatalı işlemlerin engellenmesi sağlanamaz.

- Bire-bir kontrol: İşlenmesi beklenen doküman listesi ile bilgi sistemleri tarafından işlenen doküman listeleri karşılaştırılır. Dosyaların istenilen şekilde işlendiğinin teyit edilmesi sağlanır.

- Önceden kaydedilmiş giriş: Veri girişlerinde hataların önlenebilmesi için kullanıcı girmeden önce bazı verilerin girilmiş olarak gelmesi sağlanabilir. Örneğin, kullanıcı adı ve şifresinin hazır gelmesi gibi.

- İşlem hareket günlükleri (loglar): Büyük işlemler bilgi sistemleri tarafından kayıt altına alınır. Denetim izi sağlaması için, giriş alanı, işlemin tarihi, kullanıcı adı ve terminal alanı gibi bilgiler tutulur. Böylece, ilgili kişiler tarafından hangi işlemlere ait kayıtların tutulacağının veya raporlanacağının belirlenmesine ve sistemsel hataların tespit edilmesini sağlar.

- Dosya güncelleme ve erişim yetkileri: Önleyici kontrol olarak depolanan verinin uygun şekilde korunmasını ve güncelliğini sağlar. Genel giriş yetkilerin dışında kalan bazı uygulama girişler için de sınırlamalar getirilebilir. Sadece yetkili kişilerin ilgili alanlara girişi sağlanmalıdır. Böylece log kayıtlarının alınması yanında güçlü kontrol ortamı da sağlar.

Parite kontrolü (eşlik sınaması): Aktarılan veride meydana gelen tek sayıda hataları sezmek için kullanılır. Böylece, verideki birlerin sayısını tek ya da çift olacak şekilde düzenlemektir. Veriye bir eşlik biti eklenir. Eşlik biti bir ya da sıfır yapılarak tüm veri grubu içindeki birlerin sayısının çift ya da tek olması sağlanır. Birlerin sayısının çift olmasına çift eşlik, tek olmasına da tek eşlik durumu denir.

2.3.3. Çıktı Kontrolleri

Çıktı kontrolleri, kullanıcılar için oluşturulan verilerin tutarlı ve güvenli sunulabilmesini ve iletilmesini sağlar. Bu kontrollere aşağıda yer verilmektedir.

- Hassas ve kritik verilerin güvenli bir yere kaydedilmesi ve depolanması: Kritik verileri içeren formların olası problemler nedeniyle zarar görmesi ya da çalınması durumunda sürekliliği sağlamak adına güvenli bir alanda depolanması adımıdır. Depolanan veriler ile mevcut envanterler ile düzenli periyotlarda karşılaştırılmalıdır. Ortaya çıkan farklar ile ilgili araştırma yapılmalıdır.

- Bilgisayar üretimli devredilebilir araçlar, formlar ve imzalar: Üretilen tüm formlar ile eldeki listeler karşılaştırılmalıdır. Ortaya çıkan hatalar ve eksiklikler araştırılmalıdır.

- Rapor doğruluğu, tamlığı ve zamanlı olması: Güvenli olarak bilinen sistemlerden bile hatalı rapor çıktıları oluşabilmektedir. Bunu engellemek için rapor çıktılarını hazırlamaya yönelik hazır formatlar, şablonlar, değişiklik yönetimi talep rapor tasarımı ve oluşturma özellikleri, şablonlar hazırlanabilir.

- Sistemden üretilen raporlar: Yönetim tarafından iş kararları için kullanacağı veya iş süreçleriyle ilgili sonuçlar hakkında bilgiler içeren raporları tanımlamaktadır. Bu raporlarda yer alan verilerin doğru ve eksiksizliği bilgi sistemleri tarafından yapılması beklenen fonksiyondur. Sistemden elde edilen raporların doğruluğunun teyit edilmesi için aşağıdaki yöntemler kullanılabilir:

a. Rapor dağıtımı: Uygun ve onaylanmış dağıtım parametrelerine uyumlu çıktı raporları dağıtılmalıdır. İşlem adımları birimlere ait raporların tam ve zamanında iletmiş olduğunu teyit etmelidir. Rapor dağıtımları loglanmalı, dağıtım kanallarına erişim kısıtlanmalıdır. Hassas veri içeren raporlarda gizlilik ve erişebilirlik kontrolleri için ilave kontroller uygulanmalıdır.

b. Denkleştirme ve uzlaşma: Rapor çıktısı oluşmasını sağlayan uygulamaların oluşturduğu çıktılarda rakamsal değerler ve içeriğindeki bilgiler belirli periyotlarda kontrol edilmelidir. Denetim izleri üzerinden işlem adımları takip edilmelidir.

c. Çıktı hatası işleme: Gerçekleşen hatalara ilişkin rapor ve kontrol etme prosedürlerinin oluşturulması gerekmektedir. Zaman damgalı hata raporları olmalı, hataların düzeltilmesi ve incelemelerin sağlanması için işlem sahibine iletilmelidir.

d. Çıktı raporu saklama: Yasal düzenlemelerde yer alan kayıt saklama ile ilgili yönergeler politikalara dahil edilmelidir. Bu kapsamda hukuk birimlerinden görüş ve yönlendirmeleri kritik önem taşır.

e. Alındı raporları doğrulaması: Kritik verilere ait oluşturulan raporların ilgili kişi ya da kişiler tarafından alındığına ilişkin log kayıtlarının, çıktıların dağıtım prosedürüne uygun olarak yapıldığına dair kanıt olarak saklanmalıdır.

Örnek Sorular

Soru 1: Aşağıdakilerden hangisi "Sistem Geliştirme Yaşam Döngüsü" temel fazlarından biri değildir?

A) Değerlendirme

B) Geliştirme

C) Konfigürasyon

D) Tasarım

E) Fizibilite Çalışması

Cevap: A

Soru 2: Kritik başarı faktörleri aşağıdakilerden hangisini içermez?

A) İş verimi

B) Kalite

C) Ekonomik değer

D) Müşteri hizmeti

E) Tutarsızlık

Cevap: E

Soru 3: Uygulama kontrolleri ile ne sağlanması beklenmez?

A) İşlem sonuçlarının beklenti ile uyumunun kontrol edilmesi

B) Fizibilite çalışması

C) İşlemin doğru görevi tamamlayıp tamamlamadığının kontrol edilmesi

D) Verinin korunması

E) Doğru verileri kullanılması ve güncellemelerin yapılmış olmasının kontrol edilmesi

Cevap: B

Soru 4: Aşağıdakilerden hangisi üretim ortamına aktarım sırasında izlenen adımlardan olan "uygulamaya almanın planlanması" aşamasında yapılır?

A) Eğitim verilmesi

B) Fazlı geçiş planının oluşturulması

C) Canlıya geçişin sağlanması

D) Geri dönüş senaryosu gerçekleştirilmesi

E) Kullanıcı kabul testlerinin yapılması

Cevap: D

I Sistem güvenliğinin tasarım ile uyumluluğunun doğrulanması

II Kullanıcı Kabul testlerini kontrol ederek, Kabul edilen uygulamanın teslim edildiğinin doğrulanması

III Hata raporlarının kontrol edilmesi ve takibi için kullanılan prosedürlerin uyumluluğunun doğrulanması

IV Dahili kontroller için testlerin planlanıp gerçekleştiğinin doğrulanması

V son kullanıcılar ile görüşme yaparak, prosedürlerin, kullanım talimatlarının ve yeni yöntemlerin anlaşılırlığının doğrulanması

Soru 5: Yukarıda yer verilenden hangileri yazılım testlerinin kontrolü esnasında bilgi sistemleri denetçisinin dikkat etmesi gereken hususlardandır?

A) I, II, III

B) III, IV

C) III, V

D) II, III, V

E) Hepsi

Cevap: E

KAYNAKÇA

Kurumsal Çerçeve, Kütüphane, Standart, Yönetmelik ve Tebliğler:

COBIT 2019, Governance Management Objectives, ISACA, 2018.

ITIL kütüphanesi v3F

ISO Standartları

SPK Bilgi Sistemleri Yönetimi Tebliğ Madde 4 Tanımlar

SPK Bilgi Sistemleri Yönetimi Tebliği Madde 10

Kitap:

ISACA, "CISA Gözden Geçirme Klavuzu 27. Baskı", ISBN 978-1-60420-855-9

Information Reference Model: Quality Criteria for Information, COBIT 2019, Introduction and Methodology, 2018.

APO14.06 Ensure a data quality assessment approach, COBIT 2019, Governance and Management Objectives, 2018.

EBA Guidelines on ICT and security risk Management, 28 Kasım 2019, European Banking Authority.

The NIST Definition of Cloud Computing (NIST Special Publication 800-145).

Cloud Security Alliance, "Best Practices for Mitigating Risks in Virtualized Environments.", 2015.

Web:

ISACA, https://www.isaca.org

PMI, http://projebilgialani.pmi.org.tr/

IU, https://ww4.ticaret.edu.tr/sbe/wp-content/uploads/sites/129/2020/03/projeyonetimiau207-%C4%B0stanbul%C3%9Cniversitesi.pdf

IU, http://auzefkitap.istanbul.edu.tr/kitap/endustrimuhlt_ue/projeyonetimi.pdf, Proje Yönetimi

IU, http://auzefkitap.istanbul.edu.tr/kitap/kok/projeyonetimiau207.pdf, Proje Yönetimi

Cumhuriyet Üniversitesi, http://pdo.cumhuriyet.edu.tr/wp-content/uploads/PROJE-Y%C3%96NET%C4%B0M%C4%B0..pdf

DergiPark, https://dergipark.org.tr/tr/download/article-file/560415

DergiPark, https://dergipark.org.tr/tr/download/article-file/277396

DergiPark, 328845 (dergipark.org.tr)

IMedium, https://medium.com/architectural-patterns/yazılım-geliştirme-modelleri-62915545c51e, YGM Modelleri

CSA, https://cloudsecurityalliance.org/blog/2021/04/06/what-an-auditor-should-know-about-cloud-computing-part-1/

CSA, https://cloudsecurityalliance.org/blog/2021/04/27/what-an-auditor-should-know-about-cloud-computing-part-3/

CSA, https://cloudsecurityalliance.org/research/working-groups/internet-of-things/

CSA, https://cloudsecurityalliance.org/research/working-groups/blockchain/

Bilişim. Konferans Bildirisi, https://ab.org.tr/ab07/kitap/salahli_yasar_AB07.pdf

EMO, Microsoft Word - 16.doc (emo.org.tr)

Tez, https://webdosya.csb.gov.tr/db/cbs/icerikler/salihsoylu_tez_v10-20180925134450.pdf

Tez, T09306.pdf (sakarya.edu.tr)

Wiki, Waterfall model - Vikipedi (wikipedia.org)

YTU, http://dspace.yildiz.edu.tr/xmlui/bitstream/handle/1/6206/0152763.pdf?sequence=1&isAllowed=y,

YTU, ÖNSÖZ (yildiz.edu.tr)

YBS Ansiklopedisi, https://ybsansiklopedi.com/wp-content/uploads/2015/08/Yazılım-Geliştirme-Modelleri-Yazılım-Yaşam-DöngüsüSDLCYBS.pdf

https://medium.com/databulls/bilgi-sistemi-uygulama-kontrolleri-e646627b22f9

Hızlı uygulama geliştirme (stringfixer.com)

https://acikbilim.yok.gov.tr/bitstream/handle/20.500.12812/89079/yokAcikBilim_10062556.pdf?sequence=-1&isAllowed=y