

**T.C.
SAKARYA ÜNİVERSİTESİ
İŞLETME ENSTİTÜSÜ**

**COVID-19 PANDEMİSİNİN PROJE YÖNETİM
METOTLARININ İŞLEYİŞİNE OLAN ETKİSİNİN
ARAŞTIRILMASI**

YÜKSEK LİSANS

**Uğur AZAKLI
ORCID: 0000-0002-7398-3138**

**Enstitü Anabilim Dalı : Yönetim Bilişim Sistemleri
Enstitü Bilim Dalı : Yönetim Bilişim Sistemleri**

**Tez Danışmanı: Dr.Öğr.Üyesi Naciye Güliz UĞUR
ORCID: 0000-0003-2364-5445**

AĞUSTOS – 2024

Uğur AZAKLI tarafından hazırlanan “Covid-19 Pandemisinin Proje Yönetim Metotlarının İşleyişine Olan Etkisinin Araştırılması” başlıklı bu tez, 22/07/2024 tarihinde Sakarya Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddeleri uyarınca yapılan Tez Savunma Sınavı sonucunda başarılı bulunarak, jürimiz tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman: Dr.Öğr.Üyesi Naciye Güliz UĞUR

Sakarya Üniversitesi

Jüri Üyeleri: Prof.Dr. Aykut Hamit TURAN

Sakarya Üniversitesi

Doç. Dr. Fatma Şebnem AKAL İLKHAN

Marmara Üniversitesi

 SAKARYA ÜNİVERSİTESİ	T.C. SAKARYA ÜNİVERSİTESİ İŞLETME ENSTİTÜSÜ TEZ SAVUNULABİLİRLİK VE ORJİNALLİK BEYAN FORMU	Sayfa: 1/1
Öğrencinin		
Adı Soyadı	: Uğur Azaklı	
Öğrenci Numarası	: Y219054005	
Enstitü Anabilim Dalı	: Yönetim Bilişim Sistemleri Anabilim Dalı	
Enstitü Bilim Dalı	: Yönetim Bilişim Sistemleri	
Programı	: Yönetim Bilişim Sistemleri Tezli Yüksek Lisans	
Tezin Başlığı	: Covid-19 Pandemisinin Proje Yönetim Metotlarının İzleyişine Olan Etkisinin Araştırılması	
Benzerlik Oranı	: %16	
Sakarya Üniversitesi İşletme Enstitüsü Lisansüstü Tez Çalışması Benzerlik Raporu Uygulama Esaslarını inceledim. Enstitünüz tarafından Uygulama Esasları çerçevesinde alınan Benzerlik Raporuna göre yukarıda bilgileri verilen tez çalışmasının benzerlik oranının herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim. 01 / 07 / 2024 Uğur Azaklı		
Sakarya Üniversitesi İşletme Enstitüsü Lisansüstü Tez Çalışması Benzerlik Raporu Uygulama Esaslarını inceledim. Enstitünüz tarafından Uygulama Esasları çerçevesinde alınan Benzerlik Raporuna göre yukarıda bilgileri verilen öğrenciye ait tez çalışması ile ilgili gerekli düzenleme tarafımda yapılmış olup, yeniden değerlendirilmek üzere gsbtz@sakarya.edu.tr adresine yüklenmiştir. Bilgilerinize arz ederim.		
Uygundur Danışman Unvanı / Adı-Soyadı: Dr.Öğr.Üyesi Naciye Güliz UĞUR Tarih: 01 / 07/ 2024 İmza:		
☐ Kabul Edilmiştir ☐ Reddedilmiştir EYK Tarih ve No: / / 20.... /	Enstitü Birim Sorumlusu Onayı	

İÇİNDEKİLER

KISALTMALAR	iii
TABLOLAR	iv
ŞEKİLLER	v
ÖZET.....	vi
ABSTRACT	vii
GİRİŞ	1
BÖLÜM 1. COVID-19 SALGINI VE UZAKTAN ÇALIŞMA.....	3
1.1. Covid-19	3
1.2. Uzaktan Çalışma.....	4
1.3. COVID-19 Pandemisi Sırasında Uzaktan Çalışma	4
1.4. COVID-19'un Çevik Proje Yönetimine Etkisi	5
BÖLÜM 2. YAZILIM GELİŞTİRME PROJE YÖNETİMİ.....	7
2.1. Proje Konsepti	7
2.2. Proje Yönetimi Konsepti	8
2.3. Proje Yönetiminin Kısa Tarihi	10
2.4. Proje Yönetiminin Faydaları	12
2.5. Proje Yönetiminin Dezavantajları	13
2.6. Proje Yönetim Süreçleri	13
2.7. Proje Yönetimi Bilgi Alanları	15
BÖLÜM 3. YAZILIM GELİŞTİRME PROJE YÖNETİMİ.....	18
3.1. Yazılım Geliştirme	18
3.2. Yazılım Geliştirme Proje Yönetimi.....	19
3.2.1. Yazılım Proje Yönetiminin Zorlu Faktörleri	19
3.2.2. Yazılım Proje Yönetimi Süreçleri	20
3.2.2.1. Proje Başlatma ve Kapsam Tanımı	21
3.2.2.2. Planlama	21
3.2.2.3. Yazılım Projesi Yürütme.....	22
3.2.2.4. Gözden Geçirme ve Değerlendirme	24
3.2.2.5. Yazılım Projesinin Kapanışı	24
BÖLÜM 4. YAZILIM GELİŞTİRME METODOLOJİLERİ	26
4.1. Yazılım Geliştirme Yaşam Döngüsü.....	26
4.2. Waterfall Modeli	27

4.2.1. Waterfall Modeli Adımları	28
4.2.2. Waterfall Modelinde Roller ve Sorumluluklar	30
4.2.3. Waterfall Modelinin İlkeleri ve Özellikleri	30
4.2.4. Waterfall Modelinin Faydaları	32
4.2.5. Waterfall Modelinin Kısıtlamaları.....	33
4.3. Agile Yaklaşımı.....	34
4.3.1. Agile Proje Yönetiminin Kısa Tarihi ve Prensipleri	34
4.3.2. Agile Yaklaşımının Özellikleri.....	36
4.3.3. Agile Yaklaşımının Faydaları.....	38
4.3.4. Agile Yaklaşımının Kısıtlamaları.....	38
4.3.5. Agile Yaklaşım Alt türleri	39
4.3.5.1. Kanban	40
4.3.5.2. Scrum	42
BÖLÜM 5. AGİLE VE WATERFALL ARASINDAKİ FARKLAR.....	49
5.1. Agile	49
5.2. Waterfall.....	49
5.3. Agile ve Waterfall Arasındaki Farklar	49
BÖLÜM 6. YÖNTEM VE BULGULAR	54
6.1. Örneklem İlişkin Genel Veriler	54
6.2. Araştırmada Kullanılan Faktörlere Ait Analiz	54
6.3. Ölçme Aracının Geçerliliği ve Güvenilirliği.....	55
6.4. Faktör Analizi.....	56
6.5. Hipotez Testleri	57
6.5.1. Bağımsız Örneklem T-Testi	57
6.5.2. Tek Yönlü Varyans Analizi (ANOVA).....	62
SONUÇ VE ÖNERİLER.....	70
KAYNAKÇA	73
EKLER.....	77

KISALTMALAR

AD	: Algılanan Değer
AVE	: Ortalama Açıklanan Varyans
CCPM	: Kritik Zincir Proje Yönetimi (Critical Chain Project Management)
CPM	: Kritik Yol Yöntemi (Critical Path Method)
IT	: Bilgi Teknolojisi (Information Technology)
PMI	: Proje Yönetim Enstitüsü (Project Management Institute)
PMBOK	: Proje Yönetimi Bilgi Birikimi Kılavuzu (Project Management Body of Knowledge)
PRINCE	: Projelerin Kontrollü Ortamda Yönetimi (Projects IN Controlled Environments)
PRINCE2	: PRINCE Yönteminin İkinci Versiyonu
PERT	: Program Değerlendirme ve Gözden Geçirme Tekniği (Program Evaluation Review Technique)
SPSS	: Sosyal Bilimler İçin İstatistik Programı (Statistical Package for the Social Sciences)
SWEBOK	: Yazılım Mühendisliği Bilgi Birikimi (Software Engineering Body of Knowledge)
TOC	: Kısıtlar Teorisi (Theory of Constraints)

TABLÖLAR

Tablo 1: Projeler ve Operasyonların Karşılaştırılması	7
Tablo 2: Proje Yönetimi Süreç Grupları	14
Tablo 3: Katılımcıların Cinsiyet ve Proje Yönetim Metoduna Göre Dağılımları	54
Tablo 4: Faktör Analizi	54
Tablo 5: KMO ve Bartlett Küresellik Testi	57
Tablo 6: Proje Yönetim Modeline Göre T-Testi	58
Tablo 7: Proje Yönetim Modeline Göre Faktörler T-Testi	61
Tablo 8: Proje Yönetim Modellerinin Karşılaştırılması: ANOVA Analizi	62
Tablo 9: Proje Yönetim Metodlarının Post Hoc Karşılaştırma Sonuçları	65
Tablo 10: ANOVA Tablosu: Proje Yönetim Modelleri ve Deneyim Faktörlerinin Etkisi	68
Tablo 11: Proje Yönetim Modelleri ve Deneyim Düzeylerinin Karşılaştırılması: Scheffe ve Tamhane Test Sonuçları	69

ŞEKİLLER

Şekil 1: Şelale Modelinde Uygulanan Aşamalar	28
Şekil 2: Agile Metod Adımları	37
Şekil 3: Kanban Tahtası.....	41
Şekil 4: Scrum Süreci	46



ÖZET

Azaklı, U. (2024). *Covid-19 pandemisinin proje pönetim metotlarının işleyişine olan etkisinin araştırılması* (Yayımlanmamış yüksek lisans). Sakarya Üniversitesi.

Bu tez çalışmasında, COVID-19 pandemisinin Agile ve Waterfall proje yönetim metodlarının işleyişine olan etkileri incelenmiştir. Çalışma, IT sektöründeki çalışanlar arasında yapılan anketlerle elde edilen verilerin SPSS programı kullanılarak analiz edilmesi üzerine odaklanmıştır. Araştırmanın amacı, Agile ve Waterfall metodolojilerini kullandığını iddia eden bireylerin, bu metodolojilere ne ölçüde sadık kaldıklarını belirlemek ve uygulamalarındaki tutarsızlıkları ortaya koymaktır.

Tezin teorik kısmında, Agile ve Waterfall metodolojilerinin temel prensipleri, avantajları ve dezavantajları ayrıntılı olarak ele alınmıştır. Ayrıca, bu iki metodolojinin proje yönetimi süreçlerindeki uygulamaları ve etkinlikleri karşılaştırılmıştır. Araştırmanın ampirik kısmında ise, IT sektöründe çalışan profesyonellere yönelik bir anket uygulanmış ve elde edilen veriler istatistiksel olarak analiz edilmiştir.

Analiz sonuçları, Agile metodolojisinin esneklik, hızlı adaptasyon ve müşteri odaklılık açısından daha avantajlı olduğunu, ancak Waterfall metodolojisinin daha net bir yapı ve daha belirli süreçler sunduğunu göstermektedir. Ayrıca, araştırma bulguları, firmaların genellikle hibrit modeller kullanarak her iki metodolojinin avantajlarından faydalandığını ortaya koymuştur.

Sonuç olarak, bu tez çalışması, Agile ve Waterfall metodolojilerinin kurumsal firmalardaki uygulamalarına dair önemli bulgular sunmakta ve proje yönetiminde en uygun metodolojinin seçilmesi konusunda değerli bilgiler sağlamaktadır. Bu çalışmanın, proje yönetimi alanında çalışan profesyoneller ve akademisyenler için faydalı bir kaynak olacağı düşünülmektedir.

Anahtar Kelimeler: Yazılım, Proje Yönetimi , Agile, Waterfall.

ABSTRACT

Azaklı, U. (2024). *Investigation of the impact of the Covid-19 pandemic on the functioning of project management methods* (Unpublished master's thesis). Sakarya University.

In this thesis, the impact of the COVID-19 pandemic on the functioning of Agile and Waterfall project management methodologies has been examined. The study focuses on analyzing survey data collected from IT sector employees using the SPSS software. The primary goal is to determine the extent to which individuals who claim to use Agile or Waterfall methodologies adhere to these practices and to identify any inconsistencies in their application.

The theoretical part of the thesis elaborates on the fundamental principles, advantages, and disadvantages of Agile and Waterfall methodologies. It also compares the application and effectiveness of these two methodologies in project management processes. The empirical part involves conducting a survey among IT professionals and statistically analyzing the collected data.

The analysis results indicate that the Agile methodology is more advantageous in terms of flexibility, quick adaptation, and customer orientation, while the Waterfall methodology provides a clearer structure and more defined processes. Furthermore, the research findings reveal that companies often utilize hybrid models to leverage the benefits of both methodologies.

In conclusion, this thesis presents significant findings regarding the application of Agile and Waterfall methodologies in corporate firms and offers valuable insights for selecting the most appropriate methodology for project management. This study is expected to be a useful resource for professionals and academics in the field of project management.

Keywords: Software, Project Management, Agile, Waterfall

GİRİŞ

COVID-19 pandemisi, modern çalışma düzenini kökten değiştiren ve uzaktan çalışmayı hızla yaygınlaştıran küresel bir olay olarak tarihe geçti. Pandeminin başlangıcından itibaren, sosyal mesafe kuralları ve karantina önlemleri, birçok kuruluşu geleneksel ofis düzeninden uzaklaşarak uzaktan çalışma yöntemlerini benimsemeye zorladı. Bu dönüşüm, iş yapış şekillerinde köklü değişikliklere yol açarak, uzaktan çalışma kavramının hem avantajlarını hem de zorluklarını ortaya çıkardı (Dünya Sağlık Örgütü, 2020).

Uzaktan çalışma, teknolojinin ve dijital iletişim araçlarının yoğun kullanımını gerektiren bir çalışma düzenidir. Bu bağlamda, internet, e-posta ve çeşitli telekomünikasyon araçları, çalışanların evlerinden veya işverenin belirlediği diğer lokasyonlardan işlerini sürdürmelerine olanak tanır (Sullivan, 2003). Pandemi sürecinde birçok şirket, uzaktan çalışma modeline hızla adapte olarak iş sürekliliğini sağlamış ve çalışanlarının sağlığını koruma amacını gütmüştür. Ancak bu süreç, aynı zamanda uzaktan çalışmanın verimlilik, motivasyon ve iş-yaşam dengesi üzerindeki etkilerini de gündeme getirmiştir.

Bu tez çalışmasında, COVID-19 pandemisinin uzaktan çalışma üzerindeki etkileri ve bu yeni çalışma düzeninin yazılım geliştirme projelerine yansımaları incelenmiştir. Özellikle, çevik (Agile) ve geleneksel (Waterfall) yazılım geliştirme metodolojilerinin uzaktan çalışma koşullarında nasıl uygulandığı ve bu metodolojilerin etkinliği araştırılmıştır. Araştırma kapsamında, yazılım sektöründe çalışan profesyonellerle yapılan anketler aracılığıyla, uzaktan çalışma koşullarında bu iki metodolojinin avantajları, dezavantajları ve karşılaşılan zorluklar analiz edilmiştir.

Tezin teorik kısmında, Agile ve Waterfall metodolojilerinin temel prensipleri, avantajları ve dezavantajları ayrıntılı olarak ele alınmıştır. Agile metodolojisi, esneklik ve hızlı adaptasyon yeteneği ile öne çıkarken, Waterfall metodolojisi daha yapılandırılmış ve sıralı bir yaklaşım sunar. Her iki metodolojinin de güçlü ve zayıf yönleri değerlendirilmiş ve uzaktan çalışma ortamında hangi durumlarda daha etkin oldukları tartışılmıştır.

Ampirik kısımda ise, yazılım geliştirme projelerinde çalışan profesyonellere yönelik bir anket çalışması gerçekleştirilmiş ve elde edilen veriler SPSS programı kullanılarak

analiz edilmiştir. Bu analizler, uzaktan çalışma düzeninde Agile ve Waterfall metodolojilerinin kullanım farklılıklarını ve proje yönetim süreçlerindeki etkinliklerini ortaya koymuştur. Ayrıca, şirketlerin bu metodolojilerin avantajlarından hibrit modeller kullanarak nasıl faydalandığı da incelenmiştir.

Sonuç olarak, bu tez çalışması, COVID-19 pandemisinin uzaktan çalışma üzerindeki etkilerini ve Agile ile Waterfall metodolojilerinin bu yeni çalışma düzeninde nasıl uygulandığını kapsamlı bir şekilde ele alarak, proje yönetimi alanında değerli bilgiler sunmayı amaçlamaktadır. Bu çalışma, uzaktan çalışma koşullarında proje yönetimi süreçlerini iyileştirmek isteyen profesyoneller ve akademisyenler için faydalı bir kaynak olmayı hedeflemektedir.



BÖLÜM 1. COVID-19 SALGINI VE UZAKTAN ÇALIŞMA

1.1. Covid-19

2020 yılı, küresel ekonomi ve çalışma dünyasında benzeri görülmemiş değişiklikler getirdi. 11 Mart'ta Dünya Sağlık Örgütü, yeni koronavirüs salgınına "pandemi" olarak tanımladı ve dünya genelindeki hükümetlere bu sorunu ciddiye almaları ve çeşitli radikal önlemlerle bu halk sağlığı acil durumunun ilk dalgasına hazırlanmaları çağrısında bulundu (Dünya Sağlık Örgütü, 2020). COVID-19 pandemisi ilk olarak Eylül 2019'da Çin'in Wuhan kentinde tespit edildi ve tüm dünyaya yayıldı. 6 Mayıs 2022 itibariyle, pandeminin ilk salgınından bu yana DSÖ'ye bildirilen COVID-19 doğrulanmış vaka sayısı 513,955,910, ölüm sayısı ise 6,249,700 olarak kaydedilmiştir (Dünya Sağlık Örgütü, 2022). COVID-19 virüsünün neden olduğu hastalık, insanların fiziksel olarak birbirine yakın olduğu ortamlarda öksürme, hapşırmaya ve konuşma ile yayılan virüslü küçük damlacıklar yoluyla bulaşmaktadır. Bu son derece bulaşıcı hastalığın, yayılmasını önlemek için sosyal mesafe, maske kullanımı, el-vücut temizliği gibi kişisel önlemler uygulanmış ve tüm dünyada çeşitli önlemler alınmıştır. Virüsle mücadele etmek için dünya genelinde ciddi önlemler alınmış, uluslararası seyahat yollarının kısıtlanması ve ülke genelinde karantinalar kurulması gibi tedbirler uygulanmıştır. Bazı ülkelerde tüm seviyelerdeki fiziksel eğitim faaliyetleri askıya alınmış ve uzaktan eğitime başlanmıştır. Türkiye de bu süreçten etkilenmiştir. 16 Mart 2020'de ilköğretim, ortaöğretim ve yükseköğretim kurumları uzaktan eğitime başlamıştır (T.C. Sağlık Bakanlığı, 2020). Pandemi sürecinde bulaş riskini azaltmak amacıyla, işlerin uzaktan yürütülebileceği durumlarda uzaktan çalışma tercih edilmiştir.

COVID-19 pandemisi sırasında, virüse yakalananların büyük çoğunluğunun sağlık koşullarının kötüleşmesi ve yoğun bakıma ihtiyaç duyulması, tüm hastanelerde kapasitenin aşılmasına neden olmuştur. Bu yeni virüsün uzun süreli tedavisi, her hastada çeşitli semptomların kendini göstermesi ve sürecin başında aşı çalışmalarının tamamlanmamış olması, yaşamın her alanında olumsuzluklara neden olmuştur. İnsanların maske, mesafe ve hijyen gibi alışılmadık davranış alışkanlıklarını benimsemek zorunda kalması, özel hayatın yanı sıra çalışma hayatını da derinden etkilemiştir.

1.2. Uzaktan Çalışma

Uzaktan çalışma (evden çalışma), çalışanın işini internet, e-posta ve telefon gibi telekomünikasyon ve teknoloji araçlarını kullanarak evinden veya işveren/işyeri tarafından belirlenen başka bir merkezden yaptığı çalışma türünü ifade eder. Uzaktan çalışma ve evden çalışma terimleri birbirinin yerine kullanılabilir. Uzaktan çalışma, telekomünikasyon ve teknoloji araçlarını kullanarak ofis lokasyonunda farklı bir alanda çalışmayı tarif etmektedir. (Sullivan, 2003).

1.3. COVID-19 Pandemisi Sırasında Uzaktan Çalışma

COVID-19 pandemisi nedeniyle birçok organizasyon ve işveren uzaktan çalışmayı yaygınlaştırdı. Sokağa çıkma yasakları yürürlüğe girdiğinde, işverenler tarafından birçok çalışan evden çalışmaya yönlendirildi ve bu, tarihteki en kapsamlı toplu uzaktan çalışma deneyimine yol açtı (Bilginoglu, 2021).

Randstad'ın raporuna göre, çalışanların %83'ü COVID-19 sebebiyle yeni çalışma ortamlarına uyum sağlamak durumunda kaldı. Türkiye'de yapılan bir ankette, çalışanlar uzaktan çalışmanın dezavantajları olarak aile üyeleriyle geçirilen sürenin artması, sosyal ortamlarda bulunamamanın yarattığı psikolojik sorunlar ve iş verimliliğinin düşmesini belirtti (Baycik vd., 2021).

Statistics Canada'ya göre, yeni uzaktan çalışanların %90'ından fazlası evde en az önceki iş yerlerindeki kadar üretken olduklarını bildirdi. Evden çalışanların büyük çoğunluğu, evden çalışmanın verimliliklerini olumsuz etkilemediğini ifade etmektedir (Statistics Canada, 2021). Global Workplace Analytics'in çalışmasına göre, ABD'deki mevcut iş gücünün %56'sı kısmen veya tamamen evden yapılabilecek işler üstlenmektedir ve uzaktan çalışanlar eski çalışma düzenine geri dönmek istememektedir (Gosselin, 2021).

Uzaktan çalışmanın olumlu yönleri arasında dikkat dağınıklığının azalması, daha sessiz bir ortamda çalışma ve daha esnek çalışma saatleri bulunur. Melian ve Zebib'in çalışmasına göre, katılımcıların %70'inden fazlası evden çalışırken daha üretken olduklarını belirtmiştir. Uzaktan çalışmanın %13'lük bir performans artışı sağladığı, karmaşık işler yapan ve paydaşlarla az etkileşim gerektiren kişilerin daha verimli olduğu belirtilmiştir (Golden ve Gajendran, 2019).

Uzaktan çalışmada üretkenliği ölçen çalışmalar, ortalama olarak uzaktan çalışanların ofiste çalışanlara göre ayda 1.4 gün veya yılda 16.8 gün daha fazla çalıştığını

göstermiştir (Airtasker, 2020). Ancak, evde çalışma ortamının olmaması, artan iş yükü ve çalışma ortamının bozulması, uzaktan çalışmanın motivasyon üzerindeki olumsuz etkileridir (Tuna, 2020). İşletmelerin dinamik bir çalışma ortamı sağlamak için dijital cihazlar, sanal iş birliği yazılımları ve platformları gibi teknoloji eksikliğini gidermeleri önemlidir (Gascoigne, 2020).

1.4. COVID-19'un Çevik Proje Yönetimine Etkisi

COVID-19 pandemisi, ofis iletişiminin sona ermesiyle birlikte iç iletişim üzerinde önemli bir etki yaratmıştır. Ancak, bir araştırmaya göre bu durumun takımın genel iş birliği üzerinde büyük bir etkisi olmadığı ancak verimliliği azalttığı belirtilmiştir. Ayrıca, takım üyeleri, çevrimiçi bağlantı ve etkileşimleri artırmanın önemini vurgulamaktadır (Badiale, 2020). Örneğin, sorunlarla başa çıkmak için çevrimiçi sanal kahve molaları düzenlenebilir.

Pandemi, işletmelerin hızlı bir şekilde çevik hale gelmesini gerekli kılmıştır. Bu durum, tüm ilgili süreçlerde uzaktan çalışmayı desteklemek ve iş esnekliğini artırmak amacıyla çalışma yapılmasına yol açmıştır (Nandwani, 2020). Geleneksel proje yönetimi süreçleri, aşamalı yapı yaklaşımı gibi, proje istikrarı açısından birçok fayda sağlamış olsa da COVID-19 pandemisi gibi beklenmedik engellerle başa çıkmak için gereken esneklik ve çeviklikten yoksundur (Sonjit vd., 2021).

Çevik proje yönetimi yaklaşımlarını kullanan şirketlerde çalışanlar, yüksek derecede özerklik, eşitlik ve iteratif teslimat avantajlarına sahiptirler (Koch ve Schermuly, 2021). Çevik proje ekipleri, uzun vadeli proje planlarına bağlı kalmadıkları için, iş ortamındaki değişikliklere hızlı bir şekilde uyum sağlayabilirler ve COVID-19 ihtiyaçlarına göre süreçleri hızla değiştirebilirler (Janssen ve Voort, 2020). Ayrıca, çevik yöntemler uzaktan çalışmayı daha kolay hale getirebilir.

Sonuçlara göre, çevik işletmeler COVID-19'a uyum sağlama konusunda diğerlerine göre daha başarılı olmuştur. İşlerini hızlandırmış ve önceliklerini gözden geçirerek ve hedefler ve anahtar sonuçlar gibi stratejiler kullanarak değişen sektör ortamına uyum sağlamışlardır (Jadoul vd., 2020). Çevik takımlar, hızlı değişen kurumsal hedeflere, yıkıcı teknolojilere ve dijitalleşmeye uyum sağlama esnekliği ile COVID-19'un getirdiği duruma uyum sağlamak için önemli bir avantaja sahiptirler (Comella vd., 2020).

Alexander ve diğerkleri (2020) alıřmasına gre, uzaktan alıřma sırasında yeni alıřanların uyum srecinde eřitli zorluk yařanabileceğini ortaya koymaktadır. Oryantasyon, eđitim, ekip yesi olma gibi konularda uzaktan alıřma ortamı gerekli imkânı sađlayamamaktadır. Liderlik ve ynetim stilleri, kk ekiplerle uyumlu olup olmadıkları ve bu stillerin nasıl tamamlandığı nemlidir.

Sonuç olarak, evik yntemlerin uzaktan alıřmaya adaptasyonunu kolaylařtırdığı, ancak bu srecin belirli zorluklar ierdiği ve ynetim stillerinin ekip deneyimlerini byk lde etkilediđi belirtilmektedir. Organizasyonların, diđital iř birliđi aralarını ve yeniliki uygulamaları benimseyerek bu yeni alıřma ortamına adapte olmaları gerekmektedir.



BÖLÜM 2. YAZILIM GELİŞTİRME PROJE YÖNETİMİ

2.1. Proje Konsepti

Proje Yönetim Enstitüsü'ne (PMBOK, 2017) göre, "proje, benzersiz bir ürün, hizmet veya sonuç oluşturmak için üstlenilen geçici bir girişimdir." Bu tanımda "geçici" terimi, projenin belirli bir başlangıcı ve kaçınılmaz bir sonu olduğu anlamına gelir. Buna karşılık, "benzersiz" terimi, ürün veya hizmetin diğer ürün veya hizmetlerden farklı olması gerektiğini ifade eder. (PMBOK, 2000, s. 4)

Her proje, çıktısı somut veya soyut olabilen benzersiz bir ürün, hizmet veya sonuç yaratır. Hiçbir görev aynı biçimde ve koşullarda tekrar etmez. Bazı proje çıktılarında ve faaliyetlerinde tekrarlayan unsurlar bulunabilse de, bu durum proje çalışmasının temel, benzersiz özelliklerini değiştirmez. Horine'e (2009) göre projeler şu şekilde örneklendirilebilir;

- Yeni bir ev inşa etmek
- Yeni bir yazılım uygulaması geliştirmek
- Mevcut üretim süreçlerinin değerlendirilmesi
- Yeni bir radyo reklamı oluşturmak

Projelerin aksine, organizasyona ait operasyonlar ve operasyonel işler, organizasyonu canlı tutan sürekli, tekrarlayan faaliyetlerdir. Sürekli işlemlere örnek olarak;

- Alacak ve borç hesapları faaliyetlerinin gerçekleştirilmesi
- Günlük üretim süreçlerinin yürütülmesi
- Müşteri sipariş süreçlerinin yerine getirilmesi (Horine, 2009, s. 8).

Proje ve operasyon terimlerinin karşılaştırması aşağıdaki Tablo 1'de gösterilmiştir. (Horine, 2009, s. 9)

Tablo 1

Projeler ve Operasyonların Karşılaştırılması

ÖZELLİK	PROJELER	OPERASYONLAR
Temel Benzerlikler	Planlanmış, yürütülmüş ve kontrol edilmiştir. İnsanlar tarafından gerçekleştirilir. Kaynak kısıtlamaları vardır.	Planlanmış, yürütülmüş ve kontrol edilmiştir. İnsanlar tarafından gerçekleştirilir. Kaynak kısıtlamaları vardır.
Amaç	Hedeflere ulaşmak ve sona erdirmek.	Organizasyonu sürdürmek.
Zaman	Geçici. Belirli başlangıç bitiş noktaları.	Sürekli.
Sonuç	Benzersiz ürün, hizmet veya sonuç.	Benzersiz olmayan ürün, hizmet veya sonuç.

Tablolar Devamı		
İnsanlar	Proje ihtiyaçlarını karşılamak için oluşturulmuş dinamik, geçici ekipler. Genellikle organizasyon yapısına uymaz.	Fonksiyonel ekipler genellikle organizasyon yapısına uyumludur.
Yöneticinin Yetkisi	Organizasyon yapısına göre değişir. Genellikle varsa, doğrudan hat yetkisi minimaldir.	Genellikle resmi, doğrudan hat yetkisi.

Projeler sürekli değildir, belirli bir dönem içinde tamamlanan faaliyetler bütünüdür. Ancak, projenin sonunda elde edilen ürün veya hizmet geçici olarak uygulanmaz; projelerde üretilen ürün veya hizmetler sürekli niteliktedir. Projelerdeki geçicilik, bir projenin belirli bir başlangıcı ve bitişi olduğunu ifade eder. Geçicilik ifadesi, bir projenin kısa vadeli olduğu anlamına gelmez.

Her projenin ulaşılması gereken belirli ve iyi tanımlanmış bir hedefi vardır. Projelerde amaç, projenin var olma sebebi olan odak noktasıdır. Bir plan olmadan proje olmaz. Her iş belirli kısıtlamalar altında ve belirli kaynaklar kullanılarak yapılır. Her iş tek bir amaç için yapılır. Tüm projeler belirsizliklerle başa çıkmak zorundadır. Çünkü uygulama öncesinde planlanır ve bu nedenle risk unsuru taşır. Yüksek gerilimli projeleri tanımlamak ve gerçekçi hedefler belirlemek muhtemelen zordur. Bir projenin tüm aşamaları ardışık olarak değişime tabi olduğunda, planlama süreci zorlu hale gelir. Kaynaklar sağlanabilir, süreler üzerinde anlaşılabilir, ancak bazı faaliyetler gecikirse yapılan tüm planların yeniden çizilmesi gerekecektir. Değişikliklerin sonuçlarıyla başa çıkmak için tüm proje planlaması yeterince esnek olmalıdır. (Slack vd., 2007, s. 497)

Tüm projeler belirsizliklerle başa çıkmak zorundadır. Çünkü uygulama öncesinde planlanır ve bu nedenle risk unsuru taşır. Yüksek gerilimli projeleri tanımlamak ve gerçekçi hedefler belirlemek muhtemelen zordur. Bir projenin tüm aşamaları ardışık olarak değişime tabi olduğunda, planlama süreci zorlu hale gelir. Kaynaklar sağlanabilir, süreler üzerinde anlaşılabilir, ancak bazı faaliyetler gecikirse, yapılan tüm planların yeniden çizilmesi gerekecektir. Değişikliklerin sonuçlarıyla başa çıkmak için tüm proje planlaması yeterince esnek olmalıdır. (Slack vd., 2007, s. 497)

Bu tanımlardan çıkarılacak sonuç şudur ki; proje, bir ihtiyaca yanıt verir, belirli bir amaç için yapılır, sadece bir kez gerçekleştirilir, başlangıç ve bitiş tarihine sahiptir, bir organizasyon içinde gerçekleştirilir ve esnek bir yapıya sahiptir.

2.2. Proje Yönetimi Konsepti

Proje yönetimi, Kerzner (2017) tarafından, kaynakların kullanımı, maliyetlerin ve risklerin azaltılması, değerin maksimize edilmesi amacıyla bir iş hedefi doğrultusunda

gerçekleştirilen planlama, yönetim, örgütleme ve denetim faaliyetlerinin bütünü olarak tanımlanmıştır. Butler'a göre, proje yönetimi karmaşık girişimlerle güçlendirilmiştir. Konsantre ve entegre bir yönetim sağlamak amacıyla tasarlanmıştır ve şu unsurları içerir (Butler, 1973, s. 85):

- Örgütsel kaynakların önemli bir bölümünün belirli bir konuya odaklanması
- Yüksek düzeyde birbirine bağlı uzmanlaşmış faaliyetler
- Maliyet ve ürün performansının sonundaki göreceli güç kısıtlamaları

Proje kısıtlamaları, proje yönetiminin, belirli bir faaliyette şirket kaynaklarını zaman içinde, maliyet ve performans/teknoloji dahilinde yönetmek veya kontrol etmek için tasarlandığını göstermektedir. Zaman, maliyet ve performans projede kısıtlamalar olarak kabul edilir. Bir projenin zaman, maliyet ve performans dahilinde yönetilmesinin mümkün olduğu kabul edilmelidir. Bunlar bir projenin birincil kısıtlamaları olarak kabul edilir ve genellikle başarının ölçüldüğü kriterler olarak değerlendirilir (Kerzner, 2017, s. 6).

Zaman kısıtlamaları dahilinde projenin ne kadar sürede tamamlanacağı belirlenmeye çalışılır. Ürün veya hizmetin tamamlanması için gerekli olan süre planlanır. Başlangıçta üzerinde anlaşılan zaman çizelgesine uyulmaması, müşteri veya sponsor için tazminat gerektirebilir. Bu da maliyetlerin artmasına neden olur.

Maliyet kısıtlaması kapsamında, projenin tamamlanma maliyeti hesaplanmaya çalışılır. Böylece mevcut kaynaklarla elde edilecek çalışan bir sistemin maliyeti belirlenebilir. Maliyet kapsamı, projede çalışan kişilerin ücretlerini, ekipman alımlarını, yazılım ve donanım maliyetlerini, kiralama, bilgisayar ağları, yönetim ve proje yürütülmesi için gerekli kırtasiye masraflarını içerir. Projenin maliyet yönetimi denildiğinde, sadece gerçekleşen harcamaların izlenmesi ve kaydedilmesi anlaşılmamalıdır. Proje maliyet yönetimi, zaman-maliyet ilişkisini dikkate alarak, iyi düşünülmüş, iyi tasarlanmış ve projeyi üzerinde anlaşılan tarihte bitirmeyi amaçlayan tüm çalışmaları kapsar.

Performans/teknoloji kısıtlaması kapsam olarak değerlendirildiğinde, "Hangi hedefe ulaşmak isteniyor?" sorusunun yanıtı aranır. Projenin tamamlandığında sponsorun veya müşterinin hangi ürün veya hizmeti elde etmek istediğine karar verilir. Proje yönetimi kısıtlamaları birbiriyle ilişkilidir ve bir kriterde yapılan değişiklikler diğerlerini etkiler. Proje yönetimi, proje gereksinimlerini karşılamak için kullanılan bilgi, teknoloji ve araçlardan oluşan bir yazılımdır. Proje yönetimi, uygun versiyonu ve

uygulamayı, ayrıca proje için gerekli olan proje yönetim süreçlerini bir araya getirir. Proje yönetimi sayesinde, organizasyonların projeleri etkili ve verimli bir şekilde yürütmeleri sağlanır (Kerzner, 2017, s. 9).

2.3. Proje Yönetiminin Kısa Tarihi

Modern proje yönetiminin tarihi, "proje yönetimi" kavramının ortaya çıkmasıyla başlamıştır. Proje yönetiminde yaşanan gelişmeler beş ana başlık altında incelenecektir.

1900 ve Öncesi:

Antik projelerden kalan mimari ve endüstriyel kültürel mirasın, günümüz teknolojisi kullanılmadan hızlı ve ucuz bir şekilde elde edilmesi zordur. Birkaç işveren dışında, işçilerin refahı ve güvenliği için gerekli önlemler alınmamıştır. İşçiler, ucuz bir kaynak olarak görülmüştür. Sanayi devriminin bir sonucu olarak, işçi sınıfı yeni yeni ortaya çıkıyordu, bu nedenle resmi hükümet organizasyon yapıları, işçileri dışlayarak askeri ve sivil yönetimden oluşuyordu. Bu durum, birçok işçinin fiziksel yorgunluktan yaralanması, hastalanması ve ölmesiyle sonuçlandı (Lock, 2007, s. 2).

Günümüz teknolojisi kullanılmadan, antik projelerden kalan mimari ve endüstriyel kültürel mirası hızlı ve ucuz bir şekilde elde etmek bizim için zordur. Birkaç işveren dışında, işçilerin refahı ve güvenliği için gerekli önlemler alınmamıştır. İşçiler ucuz bir kaynak olarak görölüyordu. Sanayi devriminin bir sonucu olarak, işçi sınıfı yeni yeni ortaya çıkıyordu, bu nedenle resmi hükümet organizasyon yapıları, işçileri dışlayarak askeri ve sivil yönetimden oluşuyordu. Bu durum, birçok işçinin fiziksel yorgunluktan yaralanması, hastalanması ve ölmesiyle sonuçlanmıştır (Lock, 2007, s. 2).

1900 - 1949 Dönemi:

Bu dönemde, hızlı sanayileşme ve Birinci Dünya Savaşı sırasında mühimmat ihtiyacını karşılamak için fabrika çalışanlarını ve üretkenliklerini incelemek amacıyla metodologlar ve endüstri mühendisleri ortaya çıktı. Henry Gantt bu isimler arasında öne çıkıyordu. Taylor için çalıştılar ve Model T arabası ile proje yöneticileri için önemli oldular. Henry Ford, bugün hâlâ popüler ve evrensel olarak kullanılan çizelgeleri geliştirerek üretim hattı üretimini ünlü hale getirdi. Bilgisayarlar olmadan, bu çizelgeler değişiklikler konusunda esnek değildi ve çalışma programlarına çevrilmesi sıkıcı olduğundan uygulanması ve kullanılması zordu. Bu nedenle, Gantt çubuk grafikler genellikle seçildi ve genellikle çıkarılabilir manyetik veya takılabilir ekler veya kartlar

kullanarak yeniden programlamaya izin veren özel grafiklere dayalıydı. İş atamalarından personel ve makinelerle ilgili her detay, ofiste sergilenen programlarla kontrol edilirdi (Lock, 2007, s. 3).

1950 - 1969 Dönemi:

Teknolojinin gelişmesiyle birlikte dijital bilgisayarların hayatımıza girmesi, karmaşık yol ağlarının işlenmesini ve güncellenmesini daha kısa sürede ve pratik bir şekilde yapmamızı sağladı. 1950'lerde gelişen teknolojiyle ortaya çıkan bu güçlü zamanlama ve planlama aracından faydalanan kuruluşlardan biri Amerikan savunma sanayii ve Du Pont oldu. Ayrıca, imalat ve inşaat sektörleri de bu yeni teknikleri hızla benimsedi. O dönemin bilgisayarları hem çok pahalı hem de çok büyüktü, büyük, klimalı ve temiz odalar gerektiriyordu. Bilgisayarlar pahalıydı ve çalıştırılması için gereken bütçe çok büyüktü; bu, en büyük organizasyonların bile karşılayabileceğinden çok daha fazlaydı. Bu nedenle, küçük organizasyonlardaki birçok planlamacı, bu bilgisayarların topluca işletildiği ofislerden faydalandı. Bu ofisler, hem bilgisayar üreticileri hem de bilgisayarı olan büyük şirketler tarafından kurulmuştu. Proje yönetimi saygın bir meslek olarak görülmesine de meslek tanımına dahil edildi. Irk, cinsiyet ve yaş ayrımcılığı devam etmesine rağmen, organizasyonlar işyerindeki insanların refahına önem verdi (Lock, 2007, s. 3).

1970 - 1989 Dönemi:

Bu dönemde bilgi teknolojilerinde önemli gelişmeler yaşandı. Endüstriyel proje yönetimi önceki gibi devam etti, ancak daha fazla proje yönetimi yazılımının ve bilişim teknolojisinin yayılması, farklı bir proje yöneticisi türünün ortaya çıkmasını sağladı. Bu yeni proje yöneticileri, IT projelerini geliştiren ekipleri yönlendirmek için gerekli teknik ve zihinsel becerilere sahip olan IT proje yöneticileriydi. Proje planlama veya programlama bilgisine sahip olmayan ve bu yöntemleri öğrenmek istemeyen kişilerin aksine, bu IT proje yöneticileri genellikle kıdemli sistem analistleriydi. Hizmetlerine olan yüksek talep, onların sık sık iş değiştirmelerine ve hızla daha yüksek maaş skalalarına yükselmelerine olanak tanıyordu.

Proje yönetimi yazılımları yaygınlaştı, ancak işlemler büyük, pahalı ve en temel bilgisayarlar üzerinde topluca yürütülmeye devam etti. Grafikler, modern ekipmanlara kıyasla ilkel ve yetersizdi. Bazen veri girişi, ağ diyagramlarındaki verilerin veya kartların delikli olması nedeniyle her ağ etkinliği için iki kart gerektiriyordu. Bu delikli

kartlar, genellikle proje yöneticilerinin girmesinin yasak olduğu, temiz ve klimalı odalarda çalışan eğitimli bilgisayar operatörlerine getirilmek zorundaydı. İlk işlem sonuçları, birkaç yeni kart delme ve düzeltme gibi hataları listeleyen devasa bir çıktı yığını gibi görünüyordu. Yine de, bu süreç heyecan verici ve eğlenceliydi (Lock, 2007, s. 4).

1980 - 1989 Dönemi:

Bu on yıl boyunca, proje yöneticilerinin IT profesyonellerine olan bağımlılığı azalmıştır. Artık çoğu proje yönetimi yazılımını çalıştırabilen masaüstü bilgisayarlar ve grafikler için küçük yazıcılar mevcuttu. Ancak, yöneticilerin teknolojiyle uğraşmaktan yönetecekleri şeye geçememeleri nedeniyle, verimlilik bu teknoloji büyümesiyle paralel gitmedi. İnsanlar sık sık birbirlerinin masalarında toplanır, deneyimlerini paylaşır ve IT profesyonellerine çok daha az bağımlı olmayı öğrenirdi. İşlem süreleri o kadar kısaldı ki, IT uzmanı programları hızlı bir şekilde klavyelerinden hazırlayıp çalıştırabiliyordu (Lock, 2007, s. 4).

1990 - 2000+ Dönemi:

Bu dönemde teknoloji, değişimin itici gücü olmaya devam etmiş ve proje yöneticilerinin eylemlerini büyük ölçüde etkilemiştir. 1996 yılında PRINCE yöntemi PRINCE2'ye yükseltildi; kısa bir süre sonra, 1997 yılında Kritik Zincir Proje Yönetimi (CCPM) adı verilen alternatif bir yöntem tanıtıldı. CCPM, Eliyahu M. Goldratt tarafından geliştirilen "projeleri planlama ve yönetme" yöntemidir. CPM ve PERT'ten farklı olarak, TOC'den türetilen bu yöntem, projeyi tamamlamak için gerekli olan temel özelliklerden ziyade temel kaynaklara vurgu yapar. Bu dönemde proje yönetimi, artık iki ayrı dal olarak görülmemektedir: endüstri ve IT. Projelerin daha hızlı ve daha iyi sonuçlar getirmesi için şirketler daha geniş ve tatmin edici bir yaklaşımı tercih etmektedir. 2001 yılında Agile Manifesto yazıldı. Agile Manifesto, yazılım geliştirme ekiplerinin bir ekip olarak iyi performans göstermelerini hedefleyen temel değerlere dayanmaktadır (Lock, 2007, s. 4).

2.4. Proje Yönetiminin Faydaları

PMBOK'a (2017) göre, proje yönetimi, organizasyonların projeleri etkili ve verimli bir şekilde yürütmelerini sağlar. Ayrıca, etkili proje yönetimi bireylerin, grupların ve kamu ve özel sektör kuruluşlarının şu konularda yardımcı olmasına katkıda bulunur:

- İş hedeflerine ulaşmak;
- Paydaş beklentilerini karşılamak;
- Daha öngörülebilir olmak;
- Başarı şansını artırmak;
- Doğru ürünleri doğru zamanda teslim etmek;
- Sorunları ve problemleri çözmek;
- Risklere anında yanıt vermek;
- Kurumsal kaynakların kullanımını optimize etmek;
- Başarısız projeleri tanımlamak, toparlamak veya sonlandırmak;
- Kısıtlamaları kapsam, kalite, zamanlama, maliyetler, kaynaklar açısından yönetmek;
- Kısıtlamaların proje üzerindeki etkilerini dengelemek;
- Değişimi daha iyi yönetmek.

2.5. Proje Yönetiminin Dezavantajları

Proje yönetiminin birçok avantajının yanı sıra bazı dezavantajları da vardır (Butler, 1973, s. 88):

- Karmaşık organizasyonlar
- Yapının bozulma eğilimi
- Yönetim gücü
- Personeli kullanmada zorluk

2.6. Proje Yönetim Süreçleri

PMBOK'a (2017) göre, “Bir Proje Yönetimi Süreç Grubu, belirli proje hedeflerine ulaşmak için proje yönetimi süreçlerinin mantıksal olarak gruplanmasıdır. Süreç Grupları, proje aşamalarından bağımsızdır.” Projeler genellikle başlangıçtan tamamlanmaya kadar aynı aşamalardan geçer. Projenin uygulanmasıyla başlayan yaşam döngüsünün ilk aşamasında, projeye bir yönetici atanır ve projeye katılacak kilit oyuncular, yani ekip üyeleri belirlenir. Daha sonra, proje misyonunda ana hatlarıyla belirtilen projenin ana hedefleri tanımlanır. Bu aşamada, ekip liderinin, yöneticilerin ve ekipteki diğer personelin görevleri belirlenir. Proje yönetimi süreçleri, beş proje yönetimi süreç grubuna ayrılmıştır: başlatma, planlama, yürütme, kontrol ve kapatma. (Cleland vd., 2006, s. 1-17).

Proje yönetimi süreç gruplarının tanımı aşağıdaki Tablo 2'de gösterilmiştir (Horine, 2009, s. 10).

Tablo 2

Proje Yönetimi Süreç Grupları

Süreç Grubu	Tanım	Yaygın Terimler
Başlatma	Proje veya aşamayı yetkilendirme.	"ön planlama" "başlangıç" "tanımlama"
Planlama	Projenin hedeflerini tanımlamak ve gözden geçirmek, bu hedeflere ulaşmak için en iyi plan geliştirme yolunu seçmek.	"plan geliştirme" "aşamayı belirleme" "gerçekleştirme"
Yürütme	İnsanları ve kaynakları koordine ederek planı uygulamak ve "gerçekleştirmek."	"tamamlama" "koordinasyon"
Kontrol	Proje hedeflerinin "ilerleme takibi" yapılarak, düzenli olarak izlenip ölçülmesi, plandan sapmaları belirleyerek gerekli düzeltici önlemlerin alınmasını sağlamak.	"ilerleme takibi" "rotada kalma"
Kapatma	Proje veya aşamanın kabulünü resmileştirmek ("müşteri kabulü") ve düzenli bir şekilde sonlandırmak.	"müşteri kabulü" "geçiş" "kapanış"

Kaynak: Horine (2009)

Başlatma Süreç Grubu: Yeni bir projeyi veya mevcut bir projenin yeni bir aşamasını tanımlamak için gerçekleştirilen süreçler, proje veya aşama başlangıç yetkisini alarak başlatılır. Hazırlık aşamasında, proje belirlenir, amaçları doğrultusunda değerlendirilir ve ilk onay verilir. Proje yaşam döngüsü boyunca geçerli olacak çalışma ve çevresel koşulları belirleyen, temel kuralları ve kontrol listelerini oluşturan, projenin resmi başlangıç belgesi olan proje spesifikasyonunu oluşturmak için yapılan ilk resmi çalışmalar içerir. Projeyi etkileyebilecek veya proje sonuçlarından etkilenebilecek proje paydaşlarının ilk kayıtları bu süreçte tamamlanır (PMBOK, 2017, s. 53).

Planlama Süreç Grubu: Projenin kapsamını belirlemek, hedefleri netleştirmek ve projenin gerçekleştirilmesi için gerekli olan eylem planını tanımlamak için planlama süreçleri gereklidir. Planlama süreçleri, projenin kapsamını oluşturmak, hedefleri netleştirmek ve projenin gerçekleştirilmesi için gereken eylem planını belirlemek için gereklidir (PMBOK, 2017, s. 53). Planlama, nihai, belirlenmiş bir hedefe ulaşmak için net ve kesin hedeflerin (ve bu hedefleri gerçekleştirmek için yapılması gereken iş faaliyetlerinin) belirlenmesini içerir. Örneğin, hedef bir sorunun çözülmesi veya mevcut durumdan farklı bir duruma ulaşılması olabilir. Proje sponsorları veya müşteri

tarafından planlama onaylandıktan sonra, uygulama süreci başlar (Weiss vd., 1992, s. 4-5).

Yürütme Süreç Grubu: Proje yönetim planında tanımlanan işleri tamamlamak ve proje gereksinimlerini karşılamak için gerçekleştirilen organize süreçler. Uygulama aşamasında, projeye dahil olan herkes aktif bir rol oynar. Projenin tüm planlanmış aşamaları bu süreçte gerçekleştirilir. İnsanları organize etmenin yanı sıra, proje yönetimi, planda tanımlanan işleri gerçekleştirmek için gerekli olan insan kaynakları, malzemeler ve para gibi kaynakları bir araya getirmeyi içerir. Ayrıca, planı uygulamak için gerekli olan yapının kurulduğundan emin olur (Weiss vd., 1992, s. 4-5).

İzleme ve Kontrol Süreç Grubu: Bu süreçler, projenin ilerlemesini ve performansını izlemek, gözden geçirmek ve düzenlemek için gereklidir. Planda değiştirilmesi gereken alanları belirlemek ve ilgili değişiklikleri başlatmak bu süreç grubunun konularıdır. Kaynaklar bir bütünleşik yapıya dönüştürüldüğünde, bu yapının proje ilerledikçe izlenmesi ve korunması gerekecektir. Proje yaşam döngüsü boyunca belirli noktalarda bir raporlama yapısının tanımlanması ve oluşturulması, kontrol ve izleme süreçleri ile gerçekleştirilir. Ortaya çıkan raporlar, tarihsel kayıtlar oluşturur ve performans kriterlerinin dışına çıkan durumlar ve olaylar hakkında erken uyarılar içerir (Weiss vd., 1992, s. 4-5).

Kapatma Süreç Grubu: Proje, aşama veya sözleşmeyi resmen tamamlamak veya kapatmak için gerçekleştirilen süreçlerdir. Bu rehberde süreç akış diagramları kullanılmıştır. Proje yönetim süreçleri, belirli girişler ve çıkışlarla birbirine bağlıdır; bir sürecin sonucu veya çıktısı, aynı Süreç Grubu içinde olmasa bile başka bir sürecin girdisi olabilir (PMBOK, 2017, s. 53). Üretilen ürün veya hizmetin gerçek bir ortamda test edilmeye hazır olması ile proje aşamasının sonu başlar. Proje, proje sponsorlarının ve müşterinin onayı ile teslim edilir (Horine, 2009, s. 11).

2.7. Proje Yönetimi Bilgi Alanları

Proje Yönetimi Bilgi Alanları, proje yönetiminin farklı yönlerini kapsayan ve her biri belirli bir alanı hedefleyen süreç ve uygulamalardan oluşur. PMBOK'a (2017) göre, bu bilgi alanları, proje yönetiminde etkinlik ve verimlilik sağlamak amacıyla belirlenmiştir. Aşağıda, proje yönetimi bilgi alanlarının kısa açıklamaları verilmiştir (PMBOK, 2017):

Entegrasyon Yönetimi: Projenin tüm unsurlarının koordinasyonunu içerir. Bu, projenin farklı yönleri arasındaki çatışmaları yönetmeyi, rekabet eden talepler arasında

denge kurmayı, kaynakları değerlendirmeyi ve görevleri, kaynakları, paydaşları ve diğer proje unsurlarını koordine etmeyi kapsar.

Kapsam Yönetimi: Proje kapsamını planlama, tanımlama, iyileştirme ve yönetmeyi içerir. Kapsam yönetimi görevleri; kapsam yönetimi planlaması, gereksinimlerin toplanması, kapsamın tanımlanması, iş kırılım yapısının oluşturulması, kapsamın doğrulanması ve kapsamın kontrol edilmesini kapsar.

Zaman Yönetimi: Proje zamanı ve zaman çizelgesi hakkında bilgi üretmek ve geri almak, proje zaman çizelgesinin durumunu belirlemek, planı güncellemek ve projedeki değişikliklerin zaman değişikliklerine neden olması durumunda temeli yeniden hazırlamak için kullanılır.

Maliyet Yönetimi: Maliyetlerin ve bütçenin planlanması, yönetilmesi, kontrol edilmesi ve yeniden hesaplanmasını içeren bilgi alanıdır. Bu, maliyet yönetimi planlaması, maliyetlerin tahmin edilmesi, bütçenin belirlenmesi ve maliyetlerin kontrol edilmesini kapsar.

Kalite Yönetimi: Projede kaliteyi planlamak, yönetmek ve kontrol etmek için kullanılır. Bu, kalite yönetimini planlama, kaliteyi yönetme ve kaliteyi kontrol etmeyi içerir.

Kaynak Yönetimi: Proje, proje ekibi ve kaynakların temel üretim parametreleriyle nasıl başa çıkacağını gösteren bilgi alanıdır. Bu bilgi alanı, kaynak yönetimi planlaması, operasyonel kaynakların tahmin edilmesi, kaynakların temin edilmesi, ekip geliştirilmesi, ekibin yönetilmesi ve kaynakların kontrol edilmesini içerir.

İletişim Yönetimi: İletişim yönetimini planlama, iletişimleri yönetme ve iletişim görevlerini izleme ile ilgili bilgi alanıdır ve projenin başlangıcından sonuna kadar iletişimin nasıl organize edileceğini kapsar.

Risk Yönetimi: Risklerin nasıl ele alınacağını belirleyen bilgi alanıdır. Proje risk yönetimi, risklerin tanımlanması, risk yönetimi planlaması, nitel ve nicel risk analizi yapılması, risk yanıtlarının planlanması, risk yanıtlarının uygulanması ve risklerin izlenmesini içerir.

Tedarik Yönetimi: Proje tedariklerini yönetmek, tedarik yönetimini planlamak, tedarikleri yürütmek ve tedarikleri kontrol etmek için kullanılan bilgi alanıdır.

Paydaş Yönetimi: Projenin paydaşlarını yönetmek için bilgi alanıdır. Bu, paydaşların tanımlanması, paydaş katılımının planlanması, paydaş katılımının yönetilmesi ve paydaş katılımının izlenmesini içerir.



BÖLÜM 3. YAZILIM GELİŞTİRME PROJE YÖNETİMİ

3.1. Yazılım Geliştirme

ISO tanımlarına göre yazılım, bir veri işleme sisteminin çalışmasına bağlı olan programlar, prosedürler ve belgelerden oluşur. Dolayısıyla, yazılım soyut bir üründür. Yazılım, program modüllerindeki çoklu mantıksal yolların ve arayüz detaylarının tüm kombinasyonlarının birleşimine dayanır. Yazılım oluşturma, yenilikçi problem çözmeyi ve benzersiz çözümler üretmeyi gerektirir. Yazılım, belgelerin paylaşılmasını gerektiren bilişsel ve insan temelli bir geliştirme sürecidir (Ruhe vd., 2014, s. 11).

Öte yandan, yazılım geliştirme, seçilen teknolojinin gereken rolü yerine getirip getiremeyeceğini ve nasıl yerine getireceğini keşfetme sürecidir. Yazılım geliştirme, sadece yazılım oluşturma süreci değildir; aynı zamanda amacına en uygun yazılımı nasıl oluşturacağını öğrenme sürecidir. Yazılım geliştirme, insan tarafından gerçekleştirilen en karmaşık faaliyetlerden biri olarak kabul edilir. Yazılım geliştirme sıklıkla, “bilginin edinildiği ve proje sırasında bilginin üretildiği bir öğrenme süreci” olarak tanımlanır. Yazılımı ve yazılım geliştirmeyi anlamak için yapılan tanımlar genel olarak şu şekilde ifade edilebilir (Stepanek, 2005, s. 22):

- Yazılım, en önemli sorununun karmaşıklık olması nedeniyle benzersiz ve soyut bir üründür.
- Yazılım geliştirmeye başlamadan önce gereksinimleri tanımlamak kolay değildir.
- Yazılım geliştirme teknolojileri hızla değişmektedir.
- Yazılım geliştirme, birçok teknolojiye sahiptir ve bu teknolojiler, tek bir bireyin uzmanlık kazanabileceğinden daha karmaşıktır.
- Diğer ürünlerin aksine, yazılım inşa edilmez; bunun yerine tasarlanarak oluşturulur.
- Yazılım hızla değiştirilebilir ve bu değişikliklerin doğru bir şekilde uygulanması gereklidir.
- Yazılım, başlangıçta tasarlandığı kadar mükemmel değildir; amacına uygun hale getirilmesi için her zaman değişiklikler gerektirecektir.

3.2. Yazılım Geliştirme Proje Yönetimi

Yazılım projeleri, uzmanlaşmış projelerdir ve proje yönetiminin temel özelliklerine sahip olmalarına rağmen, içerikleri nedeniyle özel tekniklere ihtiyaç duyarlar. Yazılım projelerini diğer proje türlerinden ayıran önemli farklılıklar vardır. Diğer proje yönetimi alanlarında nihai ürünün fiziksel varlığı oluşturulurken, yazılım projelerinde oluşan ürün zihinsel süreçlere dayalı olduğundan soyuttur. Bu nedenle, yazılım projelerinde tüketilecek maliyeti ve iş gücünü hesaplamak çok daha zordur ve belirsizlikler genel proje yönetimine göre daha yüksek olduğundan başarısızlık olasılığı daha yüksektir. Diğer proje yönetimi alanlarında yürütülen projeler, zamana ve mekâna göre çok fazla değişiklik göstermez. Bu projelerde proje yönetimi deneyimi kazanılır ve bu deneyimler yeni projelerde kullanılır. Yazılım projelerinde, bilgi teknolojilerinin gelişimi ve müşteri gereksinimlerindeki değişiklikler, mevcut projede bile değişikliklere neden olabilir ve önceki projelerde kazanılan deneyimi geçersiz kılabilir. Bundan dolayı, yazılım projelerindeki başarı oranları düşüktür. Projelerin başarısını ölçen kriterler, proje yönetimi üçgeni olarak da bilinen kapsam, zaman, kaynak unsurlarıdır (Sommerville, 2011, s. 94).

3.2.1. Yazılım Proje Yönetiminin Zorlu Faktörleri

Yazılım, soyut ve şekillendirilebilir bir üründür. Yazılım projelerini karmaşık hale getiren temel özellikler şunlardır (PMI, 2013):

- Proje ve onun karmaşık özellikleri
- Kaynakların gerçekçi olmayan ölçeklendirilmesi
- Proje ve ürün değeri
- Proje ve ürün kapsamındaki başlangıç belirsizliği
- Projenin gelişimiyle elde edilen bilgiler

Yazılım projesinin ortaya çıkışı, projenin yazılım gereksinimleri ve kapsamı hakkında bilgi edinildikçe değişebilir. Yazılım personelinin entelektüel sermayesi farklılık gösterebilir. Yazılım ekipleri ve proje paydaşları ile iletişim ve koordinasyon genellikle netlikten yoksundur. Yazılım projeleri yenilik gerektirdiğinden soyutlanmışlardır ve paydaşlar yazılım ürünü ile iletişim kurarken risk ve belirsizlik içermesi nedeniyle ifade etmekte veya anlamakta zorlanabilirler.

Yazılım projeleri için birincil planlama ve tahmin, bu faaliyetlerin belirsizliği veya eksik veya pratik olmayan ihtiyaçlara bağlı olması nedeniyle karmaşıktır. Doğru tahminler hazırlamak da zordur çünkü yazılım geliştiricilerin verimliliği ve etkinliği büyük ölçüde değişir. Ürün karmaşıklığı, program modülleri içindeki birden çok mantıksal yolun veri ve arayüz kombinasyonları ile birleştirilmesini zorlaştırır, bu da yazılımın geliştirilmesini ve değiştirilmesini zorlaştırır.

Tüm mantıksal yolları ve arayüzleri tüm giriş verileri ve diğer giriş uyarıları kombinasyonları altında test etmek için gereken zaman nedeniyle kapsamlı yazılım testi pratik değildir. Bu, diğer yazılım ürünleriyle entegrasyon ve performans sorunlarına neden olabilir. Çoğu yazılım birbirine bağlı olduğundan bilgi güvenliği teknikleri gereklidir ve bu nedenle yazılım güvenliği büyük ve büyüyen bir sorundur. Yazılımın soyut doğası nedeniyle yazılım kalitesinin nesnel olarak ölçülmesi ve değerlendirilmesi zordur. Yazılım geliştiriciler sürekli gelişen ve sık sık güncellenen süreçler, yöntemler ve araçlar kullanabilirler. Bir yazılım ürünü, çeşitli donanım platformlarında ve altyapı yazılımlarında çalışmak zorunda olabilir. Çalışan yazılım bağımsız bir ürün değildir; bunun yerine bilişim donanımında çalıştırılır ve genellikle çeşitli donanımların, diğer yazılımların ve manuel prosedürlerin bir parçasıdır. Platform teknolojileri, altyapı yazılımları ve tedarikçi tarafından sağlanan yazılımlar sık sık değiştirilir veya güncellenir, bu da geliştirilmekte olan yazılımda değişiklik yapılmasını gerektirebilir. (PMI, 2013, s. 6)

3.2.2. Yazılım Proje Yönetimi Süreçleri

Farklı ölçeklerdeki projelerde birçok hareketli parça vardır. Başarılı bir projeyi yürütmek için birçok şeyi koordine etmek ve izlemek gerekmektedir. Bu nedenle, projeler daha küçük ve daha sindirilebilir parçalara, yani proje aşamalarına bölünür. Proje aşamaları, büyük ölçekli projelerin daha verimli ilerleme kaydetmesini sağlamak için alınmasını ve organize edilmesini sağlar. Yazılım proje yönetim süreçleri, üst düzey süreçlerden etkilenen ve yazılım geliştirme ile ilgili alt düzey süreçleri yöneten süreçlerdir. Proje yönetim süreçleri, yönetime projede neler olduğunu görmelerine yardımcı olan ve aynı zamanda projeyi kontrol etmelerine olanak tanıyan süreçlerdir. Dolayısıyla, bunlar yönetim süreçleridir. Bu süreçler ayrıca proje risk yönetimi, çaba ve maliyet tahmini gibi süreçleri de içerir. (Ahmed, 2012, s. 10)

3.2.2.1. Proje Başlatma ve Kapsam Tanımı

Projenin başlatılması ve kapsamının tanımlanması ile yazılım projesine başlama kararını kapsayan süreçlerdir. Bu süreçlerin odaklandığı nokta, çeşitli yöntemler kullanarak gereksinimleri etkili bir şekilde elde etmek ve projenin fizibilite çalışmasını analiz etmektir. Yürütülen faaliyetler arasında gereksinimlerin müzakere edilmesi ve belirlenmesi, fizibilite analizi yapılması ve gereksinimlerin gözden geçirilip gerekirse düzeltilmesi yer alır (SWEBOK, 2014).

- Gereksinimlerin tartışılması ve belirlenmesi: Gereksinimlerin tartışılıp karara bağlanmasını, yerine getirilecek işlerin belirlenmesini ve sınırların çizilmesini sağlar. Gereksinimler, çeşitli teknikler kullanılarak müşteriden elde edilir, analiz edilir, tanımlar yapılır ve sonunda doğrulanır. Bu şekilde, kısıtlamalar ve hedefler belirlenir ve projenin kapsamı netleştirilir.
- Fizibilite analizi: Fizibilite analizinin hedefi, projenin amacını net bir şekilde tanımlamak ve alternatif yaklaşımları değerlendirerek uygulanan çözümün mevcut kaynaklar, teknoloji ve sosyo-kültürel faktörler çerçevesinde en uygun çözüm olup olmadığını belirlemektir.
- Gereksinimlerin gözden geçirilmesi ve yenilenmesi: Proje sürecinde gerçekleşen gelişmeler çerçevesinde gereksinimlerin yeniden ele alınması ve gerekirse yenilenmesi gerekmektedir. Değişikliklerin kabul edilmesi durumunda, değişikliğin etkilerini belirlemek için izlenebilirlik ve risk analizi yapılmalıdır. (SWEBOK, 2014, s. 7-5)

3.2.2.2. Planlama

Proje planlama sürecinde yapılacak ilk ve önemli faaliyet, projeye uygun metodolojinin seçilmesi ve gerekirse metodolojinin proje kapsamı, yazılım isterleri, risk analizi ve değerlendirmesi, yazılımın kullanım alanı, işlevsel ve teknik karmaşıklık düzeyi ve yazılım kalite isterleri çerçevesinde kurgulanması olmalıdır. Tüm paydaşların katılımıyla bir risk değerlendirmesi yapılmalı ve tüm paydaşların üzerinde anlaşmaya varıldığı bir risk profili oluşturulmalıdır. Yazılım kalite yönetimi süreçleri, planlama sürecinin bir parçası olarak belirlenmelidir. Planlamanın adımları kısaca aşağıdaki gibi sıralanabilir (SWEBOK, 2014);

- Sürecin Planlanması: Uygulanan metodolojilere göre çeşitlilik gösterir. Şelale modeli gibi öngörücü (predictive) yöntemler, plan tabanlı yöntemler

olduğundan, projenin başında yoğun planlama gerektirirler. Bu yöntemlerde, yazılım proje faaliyetleri sıra ile uygulanır. Agile yöntemler gibi uyarlayıcı (adaptive) yöntemlerde ise planlamaya fazla zaman harcanmaz ve faaliyetler tekrarlı bir sırayla (iterative sequence) uygulanır.

- **Kaynak Tahsisi:** Ekipman, personel ve tesisler belirli görevler için tahsis edilmeli ve hem tüm proje hem de belirli bölümler için sorumluluklar atanmalıdır. Özellikle kişilerin yeteneklerine göre kullanımı ve kaynakların optimize edilmesi açısından önemli bir konudur.
- **Risk Yönetimi:** Risk ve belirsizlik birbiriyle ilişkili ancak ayrı kavramlardır. Belirsizlik, bilginin eksik olmasından kaynaklanır. Risk ise projeye olumsuz etki yapacak bir olayın gerçekleşme olasılığıdır. Risk yönetimi, risk faktörlerinin belirlenmesi, her bir faktörün gerçekleşme ihtimali ve potansiyel etkilerinin değerlendirilmesi, risklerin öncelik sırasına göre sıralanması ve olasılıkları ile olumsuz etkilerini en aza indirmek amacıyla risk azaltma stratejilerinin oluşturulmasını kapsar.
- **Kalite Yönetimi:** Bir yazılım projesinde, yazılım kalite gereksinimleri hem niteliksel hem de niceliksel açıdan belirlenmelidir. Ayrıca, kalite güvence ve iyileştirme süreçleri ile ürünlerin doğrulama ve geçerlilik süreçleri, kalite planlama aşamasında ayrıntılı bir şekilde tanımlanmalıdır. (SWEBOK, 2014, s. 7-6)
- **Plan Yönetimi:** Değişim olasılığının bulunduğu yazılım projelerinde, planların etkin bir şekilde yönetilmesi gerekmektedir. Yazılım geliştirme sürecinde seçilen planlar ve yöntemler düzenli olarak izlenmeli, gözden geçirilmeli, raporlanmalı ve ihtiyaç duyulduğunda güncellenmelidir.

3.2.2.3. Yazılım Projesi Yürütme

Yazılım projesinin yürütme aşamasında gerçekleştirilen faaliyetler aşağıdaki başlıklar altında toplanabilir (SWEBOK, 2014).

- **Planların Uygulanması:** Proje planlarında belirlenen adımlar ve süreçler, proje ekibi tarafından hayata geçirilir. Yazılım geliştirme faaliyetleri, belirlenen zaman çizelgesi ve kaynak planlaması doğrultusunda başlatılır ve yürütülür. Her bir görev ve sorumluluk net bir şekilde tanımlanarak, proje ekibi üyelerine

dağıtılır. Planların uygulanması, projenin hedeflenen çıktılara ulaşması için kritik bir adımdır.

- **Yazılım Tedariki:** Proje için gerekli olan yazılım ve araçların tedarik edilmesi sağlanır. Bu, yazılım lisanslarının alınmasını, gerekli yazılım araçlarının temin edilmesini ve uygun donanım altyapısının oluşturulmasını içerir. Yazılım tedariki, projenin teknik gereksinimlerini karşılamak için hayati öneme sahiptir ve proje başlangıcında bu süreçlerin tamamlanmış olması gereklidir.
- **Tedarikçi Protokol Yönetimi:** Projede yer alan tedarikçilerle yapılan anlaşmalar ve protokoller dikkatle yönetilir. Tedarikçilerin performansı izlenir ve değerlendirilir, gerektiğinde müdahaleler yapılır. Tedarikçilerin zamanında ve belirlenen kalitede teslimat yapması, proje başarısı için önemlidir. Proje yöneticisi, tedarikçi ilişkilerini yönetmek ve ortaya çıkabilecek sorunları çözmekle sorumludur.
- **Ölçüm Süreçlerinin Uygulanması:** Projenin ilerlemesini ve performansını değerlendirmek için belirlenen ölçütler kullanılır. Bu ölçütler, yazılımın kalite, performans ve işlevselliğini değerlendirir. Ölçüm süreçleri, projenin belirlenen hedeflere ne derece ulaştığını görmek için gereklidir ve düzenli olarak gerçekleştirilir. Ölçüm sonuçları, gerektiğinde düzeltici eylemler için kullanılır.
- **İzleme Süreci:** Proje ilerlemesi sürekli olarak izlenir ve planlarla karşılaştırılır. Bu süreçte, projenin kilometre taşlarına ve teslimatlarına ne kadar uyduğu değerlendirilir. İzleme süreci, projenin hedeflenen zaman çizelgesine ve bütçeye uygun ilerlemesini sağlamak için kritik bir adımdır. Proje yöneticisi, izleme sürecinde elde edilen verileri analiz eder ve gerektiğinde düzeltici önlemler alır.
- **Kontrol Süreci:** Projede ortaya çıkan sapmalar ve sorunlar tespit edilir ve analiz edilir. Düzeltici önlemler alınarak projedeki sapmalar minimize edilir. Kontrol süreci, projenin planlanan hedeflere ulaşmasını sağlamak için sürekli bir geri bildirim döngüsü sağlar. Bu süreç, proje risklerini yönetmek ve olası problemleri önceden belirleyip çözmek için önemlidir.
- **Raporlama:** Proje ilerlemesi ve performansı hakkında düzenli raporlar hazırlanır. Bu raporlar, proje ekibi ve paydaşlarla paylaşılır ve projenin durumu hakkında bilgi sağlar. Raporlama, proje yönetiminin projenin mevcut durumu hakkında

bilgi sahibi olmasını ve gerektiğinde stratejik kararlar almasını sağlar. Düzenli raporlama, proje iletişimini ve şeffaflığı artırır.

Bu başlıklar altında yürütülen faaliyetler, yazılım projesinin başarılı bir şekilde tamamlanmasını ve projenin hedeflerine ulaşmasını sağlar.

3.2.2.4. Gözden Geçirme ve Değerlendirme

Belirlenen zamanlarda ve gerektiğinde, proje hedeflerine ulaşma düzeyi ve paydaş ihtiyaçlarının durumu gözden geçirilmelidir. Bu süreçte gerçekleştirilen faaliyetler, ihtiyaçların karşılanma düzeyini belirlemek ve performansı gözden geçirip değerlendirmeyi içerir (SWEBOK, 2014).

- **İhtiyaçların Karşılanma Düzeyinin Belirlenmesi:** Proje boyunca belirli aralıklarla, paydaşların ihtiyaçlarının ne ölçüde karşılandığı değerlendirilir. Bu değerlendirme, paydaşlardan alınan geri bildirimler, yapılan anketler ve düzenli toplantılar aracılığıyla gerçekleştirilir. Bu süreçte, paydaşların beklentilerinin ve gereksinimlerinin ne ölçüde karşılandığına dair veriler toplanır ve analiz edilir. İhtiyaçların karşılanma düzeyinin belirlenmesi, projenin paydaş memnuniyetini sağlamak için önemlidir.
- **Performansın Gözden Geçirilmesi ve Değerlendirilmesi:** Proje performansı, belirlenen hedefler ve kilometre taşlarına göre düzenli olarak gözden geçirilir ve değerlendirilir. Bu süreçte, projenin zaman çizelgesine, bütçesine ve kalite gereksinimlerine uyumu kontrol edilir. Performans değerlendirmesi, projenin genel sağlığını ve başarı olasılığını belirlemek için kritik bir adımdır. Performansın gözden geçirilmesi, proje yöneticisinin ve ekibin, gerektiğinde düzeltici önlemler almasını sağlar.

3.2.2.5. Yazılım Projesinin Kapanışı

Bir proje, belirli bir aşama veya iterasyon, tüm planlar ve süreçler tamamlandığında kapanış aşamasına ulaşır. Bu noktada, başarı kriterlerinin değerlendirilmesi önemlidir. Projenin kapanışı sırasında, arşivleme işlemleri yapılmalı, geriye dönük incelemeler gerçekleştirilmeli ve süreç geliştirme faaliyetleri hayata geçirilmelidir (SWEBOK, 2014);

- **Başarı Kriterlerinin Değerlendirilmesi:** Kapanış aşamasında, projenin belirlenen hedeflere ve başarı kriterlerine ulaşp ulaşmadığı değerlendirilir. Bu

değerlendirme, proje hedeflerinin, zaman çizelgesinin, bütçenin ve kalite gereksinimlerinin ne ölçüde karşılandığını belirlemek için yapılır. Başarı kriterlerinin değerlendirilmesi, projenin genel performansını ve sonuçlarını ölçmek için kritik bir adımdır.

- Arşivleme: Proje kapanışında, tüm proje belgeleri ve verileri düzenli bir şekilde arşivlenir. Arşivleme süreci, proje sürecinde üretilen tüm dokümantasyonun, raporların, tasarım dosyalarının, kodların ve diğer önemli belgelerin güvenli bir şekilde saklanmasını sağlar. Bu belgeler, gelecekteki projeler için referans olarak kullanılabilir ve proje bilgilerini korur.
- Geriye Dönük İnceleme (Retrospective Review): Projenin kapanış aşamasında, proje ekibi ve paydaşlar ile birlikte geriye dönük bir inceleme yapılır. Bu inceleme sırasında, projenin güçlü ve zayıf yönleri, karşılaşılan zorluklar ve elde edilen başarılar değerlendirilir. Geriye dönük inceleme, projeden çıkarılan derslerin belirlenmesi ve gelecekteki projelerde iyileştirme fırsatlarının tanımlanması için önemlidir.
- Süreç Geliştirme: Proje süresince elde edilen deneyimler ve geriye dönük inceleme bulguları kullanılarak, süreç geliştirme faaliyetleri gerçekleştirilir. Bu faaliyetler, proje yönetimi ve yazılım geliştirme süreçlerinin sürekli iyileştirilmesini hedefler. Süreç geliştirme, gelecekteki projelerde daha verimli ve etkili çalışma yöntemlerinin uygulanmasını sağlar.
- Kapanış Raporu: Projenin kapanış aşamasında, tüm süreçleri ve sonuçları özetleyen bir kapanış raporu hazırlanır. Bu rapor, projenin genel performansını, başarılarını ve öğrenilen dersleri içerir. Kapanış raporu, projenin resmi olarak tamamlandığını belgelemek için önemli bir dokümandır ve proje ekibi ile paydaşlara sunulur.

Bu adımlar, projenin kapanış aşamasında gerçekleştirilmesi gereken kritik faaliyetleri kapsar. Başarı kriterlerinin değerlendirilmesi, arşivleme, geriye dönük inceleme ve süreç geliştirme faaliyetleri, projenin etkin bir şekilde tamamlanmasını ve gelecekteki projeler için değerli bilgi ve deneyimlerin korunmasını sağlar (SWEBOK, 2014).

BÖLÜM 4. YAZILIM GELİŞTİRME METODOLOJİLERİ

Yazılım geliştirme faaliyetlerinin tamamını gerçekleştirmek için zamanla çeşitli geliştirme yöntemleri geliştirilmiştir. Her bir model, önceki modellerde karşılaşılan bir eksikliği giderebilirken, projelerin büyüklüğüne, üretilecek ürünlere, müşteri yapısına ve iş alanına göre ayrıışan metodolojiler hala kullanılmaktadır. Tek bir model, standart olarak kabul edilmemektedir.

4.1. Yazılım Geliştirme Yaşam Döngüsü

Yazılım ürününün hem üretim sırasında hem de müşteri kullanımı sırasında geçtiği tüm aşamalara yazılım geliştirme yaşam döngüsü denir. Yazılım geliştirme süreci, zamanlamaya dayalı ve içeriğe göre bölünmüş aşamalardan oluşur. Yazılım geliştirme yaşam döngüsü (SDLC), yazılım gereksinimlerini yazılım ürünlerine dönüştüren yazılım süreçlerini içine alır. Yazılım ürün yaşam döngüsü (SPLC), yazılım geliştirme yaşam döngüsüne ek olarak yazılım yapılandırma yönetimi ve yazılım kalite güvencesi süreçlerini ve yazılımın devreye alınması, bakımı, desteklenmesi, değerlendirilmesi ve sonlandırılması gibi "uçtan uca" süreçleri içerir. Yazılım geliştirme prosesleri arasında zamansal bir öncelik yoktur. Yazılım yaşam döngüsü modelleri (SDLC veya SPLC), bu süreçler arasındaki sıralamaları oluşturur. Yaşam döngüsü modelleri, felsefelerine göre kritik süreçleri tanımlar ve bu süreçler arasındaki sıralamaları ve etkileşimleri belirler. (SWEBOK, 2014, s. 8-4).

Yazılım yaşam döngüsünde belirli temel adımlar bulunmaktadır. Bu adımlar Planlama, Analiz, Tasarım, Üretim ve Bakım olarak sıralanabilir. Planlama aşaması, proje planının oluşturulduğu, fizibilite çalışmalarının yapıldığı ve insan kaynakları ile ekipman gibi gereksinimlerin belirlendiği süreci kapsar. Analiz aşamasında, mevcut sistemler detaylı bir şekilde incelenir ve sistem ile yazılım gereksinimleri, varsa temel operasyonel sorunlar tespit edilir. Tasarım aşaması, analizden elde edilen gereksinimlerin karşılanması amacıyla yazılımın temel yapısının oluşturulduğu süreçtir. Bu aşamada, yazılımın tüm bileşenleri tanımlanır ve ayrıntılı bir şekilde tasarlanır. Üretim aşaması, tasarlanan yazılım bileşenlerinin geliştirilmesi, test edilmesi ve kurulması süreçlerini içerir. Bakım ise, yazılımın teslim edilmesinden sonra, karşılaşılan sorunların giderilmesi, yeni özelliklerin eklenmesi ve performans iyileştirmelerinin yapılmasını kapsayan faaliyetlerdir.

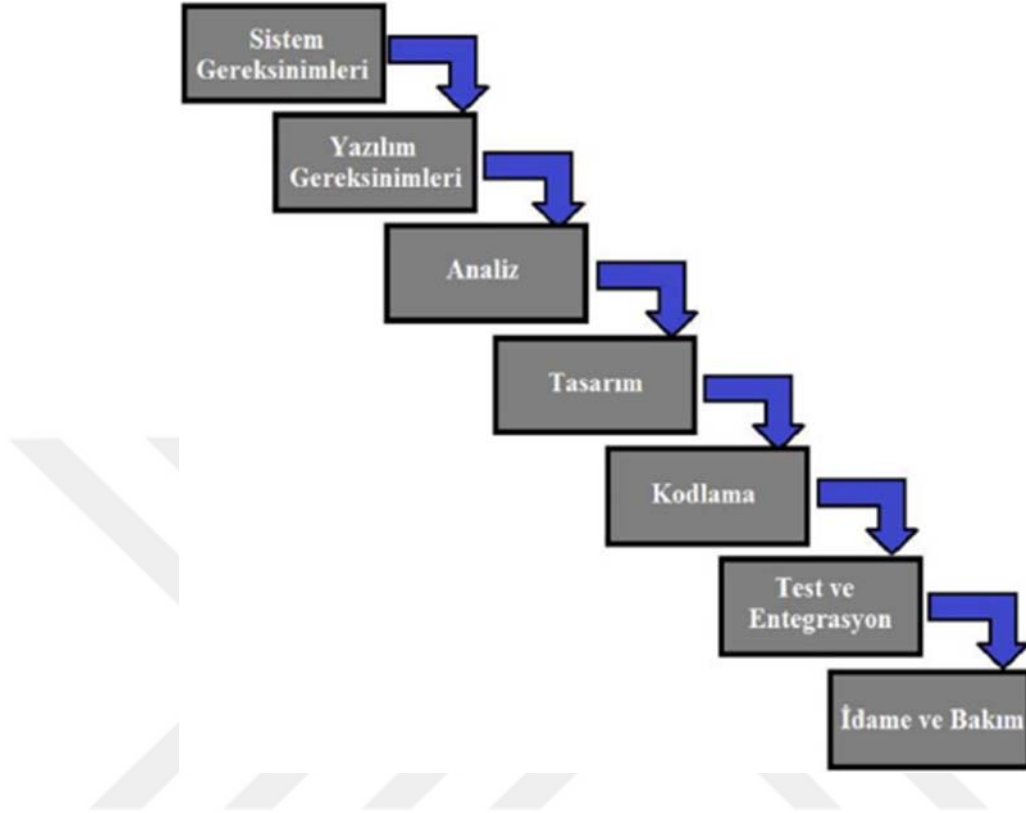
4.2. Waterfall Modeli

Şelale modeli, en eski ve temel modellerden biridir. Yazılım projelerinde uygulanan faaliyetler ardışık aşamalarda gerçekleştirilir. Bu yaklaşım, parçaların sırayla tamamlanarak bir bütün meydana getirmesinden oluşur. Bir adımı tamamlamadan bir sonrakine geçilmemesi gerektiği savunulur (Munassar vd.,2010).

Müşteri, projenin başında tüm gereksinimleri sağlar. Tüm gereksinimlerin detaylı bir şekilde belirlendiği projeler için uygun olan bu yöntem, proje geliştirme sürecinde gereksinimlerin değişme olasılığı olan projeler için uygun değildir. (Pressman vd., 2019, s. 25). Şelale yaklaşımında, yazılım geliştirme sürecinin tamamı ayrı aşamalara bölünmüştür. Şelale modelinde, tipik olarak bir aşamanın sonucu, bir sonraki aşama için girdi sağlar. Bu model, gereksinim analizi, tasarım, kodlama ve birim testleri, entegrasyon testleri ve bakım-onarım aşamalarından oluşur. (Sommerville, 2007, s. 94). Aşağıdaki şelale modelinde, bir sonraki adıma geçmeden önce her adım tamamlanır (Stanley vd., 2020):

Şekil 1

Şelale Modelinde Uygulanan Aşamalar



Kaynak: Gencer, C. ve Kayacan, A. (2017)

4.2.1. Waterfall Modeli Adımları

Gereksinimlerin Analizi: Bu aşamada, tüm olası sistem gereksinimleri toplanır, geliştirilir ve bir gereksinim tanımlama dokümanında belgelenir. İlk aşamada, müşteriden gelen talepler analiz edilir. Oluşturulacak programın doğasını anlamak için analistin yazılımın bilgi alanını, gereken fonksiyonları, davranışları, performansı ve arayüzü anlaması gerekmektedir. Hem sistem hem de yazılım gereksinimleri belgelenir ve müşteriyle birlikte gözden geçirilir. Yazılım projesinin başlangıcında, müşteri ihtiyaçları analiz edilir, belgelenir ve onaylanır. Müşterinin taleplerinin üründe uygulanabilirliği ve fizibilitesi önceliklendirilir. Daha sonra, isterlerin uygulanması için gerekli olan zaman, maliyet ve benzeri unsurlar belirlenir. Bu aşamada, isterlerin doğru anlaşılması ve ihtiyacı karşılayacak bir çözüm üretilmesi gerekmektedir; hatanın geri alınması ve düzeltilmesi, Şelale modelinde önemli mali kayıplara ve proje teslim süresinde gecikmelere neden olacaktır (Pressman vd., 2019).

Tasarım: İlk aşamadaki ister özellikleri bu aşamada incelenir ve sistem tasarımı hazırlanır. Sistem Tasarımı, donanım ve sistemi tanımlamaya yardımcı olur ve genel sistem mimarisini belirler. Bu aşamada, teknik detaylardan ziyade, müşterinin ihtiyaç duyduğu yazılım ürününün süreci çıkarılır. (Pressman vd., s. 26). Yazılım tasarımı, esasen bir programın veri yapısı, yazılım mimarisi, arayüz temsilcileri ve algoritmik detaylar gibi farklı özelliklerine odaklanan çok aşamalı bir süreçtir. Tasarım süreci, gereksinimleri, kodlama başlamadan önce kalite açısından değerlendirilebilecek bir yazılım temsilcisine dönüştürür. Gereksinimler gibi, tasarım da belgelenir ve yazılım yapılandırmasının bir parçası haline gelir (Sommerville, 2007).

Kod Üretimi: Sistem tasarımından elde edilen girdilerle, sistem bir sonraki aşamaya entegre edilir. Tasarımın makine tarafından okunabilir bir forma dönüştürülmesi gereklidir. Bu görev, kod üretimi adımıyla gerçekleştirilir. Tasarım detaylı bir şekilde yapılmışsa, kod üretimi daha verimli bir şekilde gerçekleştirilebilir. Bu aşamada, yazılım geliştirme ekibi kodlama sürecine geçecektir. Kodlama sürecinde dikkat edilmesi gereken en önemli unsurlardan biri, kodun ekip tarafından belirlenen standart kurallara uygun olarak yazılmasıdır. Bu sayede, kodlama süreci düzenli ve kontrollü bir şekilde ilerleyecektir (Pressman vd., 2019).

Birim ve Entegrasyon Testlerinin Yapılması: Kod oluşturulduktan sonra program testi başlar. Test süreci, yazılımın mantıksal iç yapısına odaklanır ve tüm ifadelerin test edilmesini sağlar. Hataları ortaya çıkarmak ve tanımlanan girdilerin doğru sonuçlar üretilip üretilmediğini kontrol etmek amacıyla testler gerçekleştirilir. Uygulama aşamasında, geliştirilen tüm bileşenler her bir birimin test edilmesinin ardından bir araya getirilir ve sisteme entegre edilir. Entegrasyonun ardından, tüm sistem olası hatalar veya arızalar açısından kapsamlı bir şekilde test edilir. Geliştirme ekibinin çalışmalarını tamamlamasının ardından, test ekibi devreye girer ve geliştirilmiş talebe ek olarak tüm sistemi test eder. Eğer bir sorun tespit edilirse, test ekibi bu durumu yazılım geliştirme ekibiyle paylaşır. Sistemde hiçbir sorun kalmayana kadar test süreci devam eder. Son olarak, süreç müşteri testine geçer ve bu test süreci, müşterinin kabulüyle tamamlanır. (Özdemir, 2017)

Bakım-Destek: Yazılım müşteriye teslim edildikten sonra, geliştirme ve değişiklik talepleri olabilir. Değişiklikler, hatalarla karşılaşıldığı için meydana gelebilir. Yazılım, dış ortamda meydana gelen değişikliklere uyum sağlamak için adapte edilmelidir veya müşteri, işlevsel veya performans iyileştirmeleri isteyebilir. Ayrıca, ürünü geliştirmek

için daha iyi sürümler yayınlanabilir. Müşteri ortamında bu değişiklikleri yapmak için bakım yapılır. Yazılım desteği/bakımı, yeni bir program yerine mevcut bir programa her bir önceki aşamayı yeniden uygulamayı içerir. (Sommerville, 2007, s. 94)

4.2.2. *Waterfall Modelinde Roller ve Sorumluluklar*

Şelale modeli, genellikle kamu kurumları ve büyük şirketler tarafından her türlü büyük ölçekli projenin yönetim standardı olarak kabul edilmekte ve uygulanmaktadır. Bu nedenle, ekipteki kişi sayısı çoğunlukla en az on beş kişidir. Her üyenin belirli görevleri vardır ve ekipteki diğer kişilerin görevlerini üstlenmemelidirler. Proje Ekibi Üyeleri ve Görevleri (Sergeev, 2016);

Proje Yöneticisi: Ekibin en temel sorumlu kişisidir. Projenin genel seyri, planlaması ve olası sorunlar için proje yöneticisi ile iletişime geçilmelidir.

Analist: Müşteriden gelen istekleri analiz eden, doğru yönlendirme ve istenilen iş için tasarım yapan kişidir. İş analisti olarak da bilinir.

Yazılım Mimarı: Tasarım sürecindeki bir diğer önemli rol. Teknik bağlantıları kuran kişidir.

Yazılım Geliştirme Sorumlusu: Sistemde tasarımı gerçekleştiren kişidir. Yazılım için gerekli kodlamadan sorumludur.

Test Sorumlusu: Kontroller yapıldıktan sonra, tüm sistemi sorunsuz hale gelene kadar test eden kişidir. Olası sorunları çözmek için yazılım geliştirme ekibi ile iletişime geçer ve sorun çözülene kadar süreci takip eder.

4.2.3. *Waterfall Modelinin İlkeleri ve Özellikleri*

Her aşamanın sonunda, dokümantasyondan oluşan bir çıktı elde edilir ve her çıktı bir sonraki aşamanın girdisini oluşturur. Çıktı-girdi ilişkisi nedeniyle olası değişikliklerin maliyeti proje için yüksek olacağından, aşamalar arasında atlama veya büyük çaplı geri dönüşlere izin verilmez. Şelale modelinin uygulandığı projelerde, aşama sonları açıkça tanımlanır ve proje, her aşama değişikliğinde yoğun denetim, kapsamlı dokümantasyon ve yönetim onayı ile yürütülür. Bu sebeple, uygulanması sıkı disiplin gerektirmektedir. Bu model, "büyük ön tasarım" felsefesini benimser ve kodlama aşamasından önce kapsamlı bir tasarım oluşturulur (Farell, 2007).

Sistem ve yazılım ihtiyaçları, gerekli her şeyin başlangıçtan itibaren bilindiği ve değişiklik olmayacağı varsayımlarıyla ilk başta belirlenir. Bu nedenle, tamamen anlaşılan ve değişme olasılığı düşük olan projeler için daha uygundur. Çünkü bu varsayımlar, şelale modelinin en büyük zayıflığını oluşturur. Müşteri, proje başında ihtiyaçlar belirlendiğinde projeye dahil olur, ancak tasarım ve kodlama aşamalarında aktif bir rol almaz; bu süreçlerde yer almaz. Müşteri, test ve entegrasyon aşamalarında yeniden devreye girer. Şelale modelinde, planlama, zaman çizelgeleri, hedef tarihler, bütçe ve tüm sistemin tek seferde kullanıma alınması en çok önem verilen unsurlardır. Bu modelin temel amacı, proje süresince değişikliklere izin vermeyerek kapsam, zaman ve kaynaklar gibi unsurları başlangıçta belirlemek ve değişiklikleri, büyük bir risk unsuru olarak kabul edip en aza indirmektir. (Vohra vd., 2013). Şelale modeli ortaya çıktığında ve koşullar değerlendirildiğinde, iş süreçleri günümüzdeki kadar bilgi sistemlerine bağımlı değildi ve projelerin acilen tamamlanması gerekmiyordu. Teknolojik gelişme ve iş hayatındaki değişim, günümüzdeki kadar hızlı değildi, bu nedenle projeler günümüzdeki kadar hızlı ilerlemiyordu. Teknoloji ve kaynak kısıtlamaları nedeniyle, bilgi sistemleri donanım merkezliydi ve yazılımın temel amacı, depolama kapasitesi ve işlem gücü gibi sınırlı donanım kaynaklarını en optimum şekilde kullanmaktı (Munassar vd., 2010). Bu nedenlerden dolayı, şelale modeli ortaya çıktığında daha çok bir risk yönetim yöntemi olarak görülmüş ve bu riskin yönetilmesi için gerekli görülen konular ve şelale modeli ile geleneksel proje yönetim dünyasının temeli olan konular şunlardır:

- Planlama ve bütçeleme için doğru yapılabilmesi için tüm gereksinimlerin erken tanımlanması gerekiyordu.
- Gereksinimlerin erken belirlenmesi ve değişmeden sabitlenmesi, aşırı kapsama engel olurdu.
- Tüm tasarım ve karar alma süreçleri, gelecekte başvurulmak üzere belgelenmeliydi.
- Aşamalar arasındaki yoğun incelemeler, tüm paydaşların süreçten haberdar olmasını ve potansiyel sorunların erken tespit edilmesini sağlardı.
- Mimari uygunluğun sağlanabilmesi için tasarım faaliyetlerine başlamadan önce tüm gereksinimlerin belgelenmesi gerekiyordu.
- Tüm kodlama süreçleri, tüm sistemin yeteneklerinin değerlendirilebilmesi için testten önce tamamlanmalıydı.

- Tüm gereksinimlerin, sistemin uygulanmasından önce karşılanması gerekiyordu (Palmquist vd., 2013).

4.2.4. *Waterfall Modelinin Faydaları*

Waterfall modelinin faydaları aşağıdaki gibi sıralanabilir (Munassar vd., 2010; Balaji vd., 2012; Farell, 2007).

- Projenin başında isterlerin açık, anlaşılır ve net bir şekilde belirlenmesi, daha uygun zaman ve maliyet tahmini yaparak projedeki belirsizlikleri azaltır.
- Yaygın olarak kullanılan bir yöntem olduğundan, teorik olarak genellikle iyi bilinmektedir.
- Doğrusal bir süreç olduğu için uygulaması kolaydır.
- Kim neyi ne zaman yapmalı şeklinde sıkı bir şekilde tanımlandığından, dalgalanan performansa sahip proje personeli, proje yöneticileri ve proje ekipleri için uygundur.
- Zaman, maliyet ve çaba konusunda makul bir tahmin yapılırsa, tahsis edilen kaynakların aşılma olasılığı minimumdur.
- Bir aşamaya başlamadan önce önceki aşamanın tamamlanması, proje ilerlemesinin takibini kolaylaştırır. Ayrıca, aynı anda ilerleyen bir aşamanın olmaması kaynak takibini kolaylaştırır.
- Dokümantasyonun sürekli bir faaliyet olması, projenin sonunda dokümantasyon amaçlı geçmişe dönme çabasını önler ve projenin herhangi bir anında dokümantasyonun eksiksiz olmasını sağlar. Bu da projenin mevcut adımının bu dokümantasyon üzerinden değerlendirilmesine olanak tanır.
- Şelale modeli disiplinli ve stabildir, proje üyelerine projenin iyi yönetildiği ve her şeyin kontrol altında olduğu hissini verir.
- Müşterinin projeye ne zaman ve ne kadar katkıda bulunacağı bellidir ve projenin tamamı boyunca zaman ayırması beklenmez.
- Birden fazla projede çalışan proje personelinin işleri çakışmasını önler. Böylece, analiz tamamlandıktan sonra başka bir projede çalışmaya devam eden bir çalışan, geri dönmek zorunda kalmaz.
- Hazırlanan detaylı ve kapsamlı dokümantasyon, iletişim aracı olarak kullanılabilir ve özellikle büyük projelerde ve tüm unsurların aynı ortamda olmadığı projelerde yüz yüze iletişim ihtiyacını azaltır.

4.2.5. *Waterfall Modelinin Kısıtlamaları*

Waterfall modelinin kısıtlamaları ise şöyledir (Munassar vd., 2010; Farell, 2007; Awad, 2005):

- Değişiklik, iyileştirme veya düzeltme için geri dönüşler şelalede yukarı yüzmeye benzetilir ve bu zor ve maliyetlidir. Bu nedenle, değişikliklere yanıt veremez ve süreç içinde gelen talepler genellikle kabul edilmez.
- Herhangi bir aşamada tespit edilemeyen olumsuzluklar, sonraki tüm aşamaları etkiler.
- Yapısal özelliği ve sıkı kontrolleri nedeniyle esnek olmayan, yavaş ve ağır bir doğaya sahiptir.
- Projenin başında gereksinimlerin tamamen ve net bir şekilde tanımlanmasını beklemek çok gerçekçi olmayabilir ve müşterilerin ihtiyaçlarının tam ve doğru bir şekilde ifade edilememesi, projenin sonunda hatalı veya yanlış bir ürünle sonuçlanabilir.
- Geliştiricilerin, müşteriler veya kullanıcılarla yüz yüze görüşerek gereksinimleri anlamak yerine analiz ve tasarım aşamalarında hazırlanan belgelerden okumaları, gereksinimlerin anlaşılmasını zorlaştırır ve geliştiricilerin katkıda bulunmasına olanak tanımaz.
- Gereksinim tutarsızlıkları, eksik sistem bileşenleri ve beklenmedik geliştirme ihtiyaçları sadece tasarım veya kodlama aşamasında tespit edilebilir.
- Tüm kodlama tamamlanana kadar sistem performansı ölçülemez ve tüm sistem testi yapılana kadar hatalar tespit edilemez.
- Yoğun ve güncel dokümantasyon, zaman alıcı ve iş gücü yoğun bir faaliyettir ve genellikle geliştiriciler tarafından gereksiz bir yük olarak görülür.
- Yazılı sistem tanımlamaları, kullanıcılar veya müşteriler tarafından okunması zor ve tamamen anlaşılması ve uygulanması zordur.
- Bir aşama tamamlanmadan bir sonraki aşamanın başlamaması nedeniyle, tüm proje personeli ilgilendirmeyen gecikmeler bile, tüm proje personeli için zaman kaybına neden olabilir.
- Küçük projeler ve küçük ekipler için yüksek yönetim giderleri gereksiz maliyetlerdir.
- Yazılım geliştirme, yaratıcılık gerektirir ve şelale modelinin oldukça disiplinli yaklaşımı, yaratıcılığı olumsuz etkiler.

4.3. Agile Yaklaşımı

Stanley ve diğerleri (2020), "agile, işleri kısa çalışma aşamalarına bölmeye odaklanan, sık sık proje değerlendirmeleri yapılan ve gerektiğinde planlamanın yürütme sırasında adapte edildiği bir proje yönetim yöntemi" olduğunu belirtmişlerdir. Agile yazılım geliştirme, paydaşların değişen ihtiyaçlarına yönelik maliyet etkin, yinelemeli ve artımlı bir yaklaşımdır. Kendi kendini organize eden ve yüksek kaliteli çözümler üreten ekipler tarafından yeterli formalizasyonu içeren pratik bir yönetim çerçevesi içinde son derece işbirlikçi bir yaklaşımla buluşur. Sonuç olarak, agile yöntemler, geleneksel yöntemlerin yetersiz kaldığı konular için alternatif çözümler olarak ortaya çıkmıştır.

4.3.1. Agile Proje Yönetiminin Kısa Tarihi ve Prensipleri

Agile Proje Yönetimi fikri ilk olarak 1986 yılında Takeuchi ve Nonaka tarafından Harvard Business Review'de yayımlanan bir makalede tanıtıldı. Ancak, Cervone'a (2011) göre, Agile Proje Yönetimi konsepti, yazılım geliştirmede agile yöntemlerin kullanımının ilk kez 1995'te tartışılmasına kadar pek dikkat çekmedi. 2001 yılında, agile yöntemlerin yaratıcıları ve önde gelen uygulayıcıları olan on yedi yazılım mühendisi, dokümantasyona dayalı geleneksel yazılım geliştirme süreçlerine alternatif bulmak ve metodolojiler temelinde ortak bir zemin oluşturmak amacıyla bir araya geldiler. Agile Software Development Alliance olarak adlandırılan bu uzmanlar, yapılan çalışmalar sonucunda agile yöntemlerin temel prensiplerini ve değerlerini belirlediler ve Agile Manifesto'yu yayımladılar. Agile Manifesto, Agile Software Development Alliance hareketi adı altında oluşturuldu. Agile Software Development Alliance (2001), dört değeri içeren bir manifesto ilan etti: “Yazılım geliştirmede daha iyi yollar keşfediyoruz, bunu yaparak ve başkalarına yardımcı olarak. Bu çalışma aracılığıyla, değer verdiğimiz şeyler şunlardır:”

- “Süreçler ve araçlardan ziyade bireyler ve etkileşimler”
- “Kapsamlı dokümantasyondan ziyade çalışan yazılım”
- “Sözleşme pazarlığından ziyade müşteri ile iş birliği”
- “Bir planı takip etmekten ziyade değişime yanıt vermek.”

Agile manifestosuyla birlikte, aşağıda listelenen 12 prensip de oluşturulmuştur:

- “En yüksek önceliğimiz, değerli yazılımın erken ve sürekli teslimi yoluyla müşteriyi memnun etmektir.” Projenin ilk aşamalarından itibaren sürekli kod

gönderimleriyle gerçekleştirilir. Müşteri, yazılımı bir bütün olarak teslimattan önce başlatarak değer kazanmaktadır. Agile yöntemlerin günümüzde yaygın olarak kullanılmasının başlıca nedenlerinden biri, yapılan yatırımların hızlı geri dönüş sağlamasıdır.

- “Gelişimin geç aşamalarında bile değişen gereksinimlere hoş geldiniz. Agile süreçler, müşterinin rekabet avantajı için değişimi kullanır.” Amaç, müşterinin ihtiyaçlarını karşılayan, onlara fayda sağlayan ve gerçek değer katan yazılım geliştirmektir. İhtiyaçlardaki değişiklikler, projenin ilerleyen aşamalarında bile yazılıma yansıtılmalıdır. Test odaklı geliştirme, kapsamlı otomatik testler, sürekli entegrasyon, basit tasarım ve kod iyileştirme gibi uygulamalar sayesinde değişikliklerin maliyetleri minimize edilir ve süreç değişikliklere daha uygun hale getirilir.
- “Çalışan yazılımı sıklıkla, birkaç haftadan birkaç aya kadar teslim edin, tercihen daha kısa zaman aralıklarında.” Yazılım teslim aralığı tipik olarak 1-4 haftadır. Bu şekilde, sürekli geri bildirim sağlanır ve yazılım, müşterinin isteklerine göre evrilir ve gelişir.
- “İş insanları ve geliştiriciler, proje boyunca her gün birlikte çalışmalıdır.” Farklı roller arasında sürekli aktif iletişim sağlanır. Rol bazlı takımlar yerine, yazılım özelliklerine göre takımlar oluşturulur. Testçiler, analistler ve yazılım geliştiriciler aynı takımda birlikte çalışır ve sürekli iletişim halinde olurlar.
- “Projeleri motive olmuş bireyler etrafında oluşturun. Onlara ihtiyaç duydukları ortamı ve desteği verin ve işi başarmaları için onlara güvenin.” Motive etmek, "harekete geçirmek, teşvik etmek" anlamına gelir. Ürünün sahibi veya yöneticisi olarak, motivasyon, ortam ve destek sağlanacak ve motive olmuş bir ekip güven ortamında çalışacaktır. Böylece projeye şeffaflık ve dürüstlük kazandırılacaktır.
- “Bilgiyi bir geliştirme ekibine iletmenin ve ekip içinde paylaşmanın en verimli ve etkili yöntemi yüz yüze konuşmadır.” Ürün sahibi veya yöneticisi ile ve takım üyeleri arasında yüz yüze toplantılar yapılması sinerji yaratacak ve motivasyonu artıracaktır.
- “Çalışan yazılım, ilerlemenin birincil ölçüsüdür.” Çalışan yazılım, tamamlanmış yazılımla aynı şey değildir. Çalışan yazılımın son kullanıcıya değer katması için tamamen bitmiş olması gerekmez. Çalışan yazılımın kademeli olarak tanıtılması, son kullanıcılara daha düzenli geri bildirim sağlama fırsatı sunar. Böylece,

geliştirme ekibi gereksiz özellikler ve işlevler yaratmadan sadece ihtiyacı karşılar.

- “Agile süreçler sürdürülebilir gelişimi teşvik eder. Sponsorlar, geliştiriciler ve kullanıcılar sürekli bir tempoyu sürdürebilmelidir.” Planlamanın sağlıklı olabilmesi için ekibin iş teslim hızının güvenilir olması gerekir. Örneğin, fazla mesai gibi yöntemlerle ekibin hızını geçici olarak artırmak tercih edilen yöntemlerden biri değildir. Bunun yerine, işler ekibin kapasitesi göz önünde bulundurularak parçalara ayrılmalı ve beklentiler bu şekilde şekillendirilmelidir.
- “Teknik mükemmeliyet ve iyi tasarıma sürekli dikkat, çevikliği artırır.” Teknik olarak kusursuz ve basit çözümler oluşturmaya özen gösterilir. En iyi tasarım, hızla gelişen tasarımıdır. Müşteri tarafından önceliklendirilen her özellik, teknik mükemmeliyet ve sağlam bir tasarımla oluşturulmalıdır. Karmaşıklık uğruna mutlak ihtiyaçtan sapılmamalıdır. Bu, projenin ilerleyen aşamalarında teknik borca yol açabilir.
- “Basitlik--yapılmayan iş miktarını en üst düzeye çıkarma sanatı--esastır.” Basitlik kavramı, akla gelen ilk kaçamak çözümü uygulamak değildir. Anlaşılması ve sonradan değiştirilmesi kolay, en düşük maliyetli ve mevcut ihtiyaçları karşılayan bir çözüm üretmektir.
- “En iyi mimariler, gereksinimler ve tasarımlar kendi kendini organize eden ekiplerden çıkar.” Buradaki kritik nokta, kendi kendini organize edebilme yeteneğidir. Üst yönetim tarafından desteklenen ekipler, gerekli olduğunda risk alarak üzerlerinde herhangi bir baskı hissetmeden kendi organizasyonlarını sağlayacaklardır. Böylece, teknik borç oluşma riski de azalır.
- “Düzenli aralıklarla, ekip daha etkili olmanın yollarını düşünür, ardından davranışlarını buna göre ayarlar ve düzenler.” Daha etkili ve verimli çalışmak için geriye dönük toplantılarla yapılan çalışmalar gözden geçirilir. (PMI, 2017)

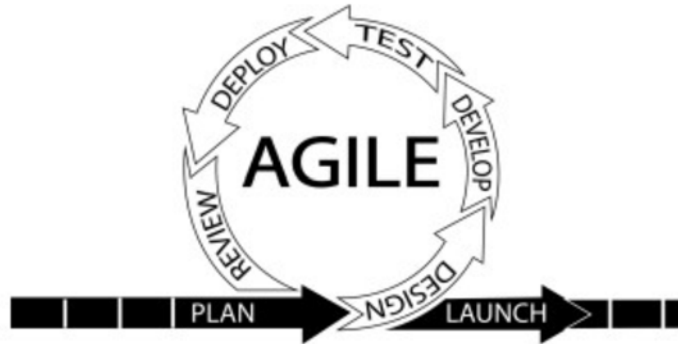
4.3.2. Agile Yaklaşımının Özellikleri

1980'lerin sonlarında ilk kez tanıtılan agile proje yönetimi metodolojisi, doğrusal olmayan bir yaklaşım ile karakterize edilir. Gereksinimler ve çözümler, "Planlama" adımını tamamlamaktan ziyade, kendi kendini organize eden, çapraz fonksiyonel ekipler arasındaki iş birliği yoluyla süreç boyunca geliştirilir.

Proje çalışmaları, proje dönemi boyunca sürekli olarak yürütülür. Agile yöntemler, uyarlanabilir planlama, sürekli iyileştirme ve erken teslimatı içerir. Değişime çevik, hızlı ve esnek tepkiler verilmesini teşvik eder. Agile yöntemlerde uygulanan iterasyonlar, önceki iterasyonlarda edinilen deneyimler ve tespit edilen kusurlar ile şekillendirilir. Tamamlanması gereken görevler, iş değerlendirmelerine göre önceliklendirilir. Proje ekibi, mevcut kaynaklar ve kısıtlamalar doğrultusunda işleri en iyi nasıl yapacaklarına karar verir. Proje ekibi, belirlenmiş iterasyon süresi içinde belirlenen görevleri tamamlamak zorundadır. İterasyonun sonunda teslim edilecek ürünün oluşturulmasından, güçlü ve zayıf yönleriyle birlikte ekip sorumludur. Bu nedenle, ekip içi iş birliği çok önemlidir. Bu bağlamda, agile yöntemlerin dayandığı temel prensipler deneyimsellik, önceliklendirme, kendi kendini organize etme, zaman çerçevesi ve iş birliği olarak sıralanabilir. Bu yöntem giderek daha popüler hale gelmekte ve sıklıkla yazılım oluşturma sürecini yönetmek için kullanılmaktadır. Takım çalışmasına odaklanır ve bir yazılım özelliğini tamamen bitirmeden bir sonrakine geçmez. Aşağıdaki şekilde agile iteratif akış görülmektedir. (Stanley vd., 2020).

Şekil 2

Agile Metod Adımları



Kaynak: Demirag, Öztürk ve Ünal (2023)

Agile yöntemlerde, yazılım geliştirme faaliyetleri iteratif aşamalar halinde uygulanır. Bu iterasyonlar sonucunda kullanılabilir ürünler oluşturulur ve geliştirme süreçleri, müşteri veya kullanıcıların geri bildirimleri doğrultusunda ve varsa değişen gereksinimlere göre tekrar edilir. Bu iteratif süreçler, müşterinin istediği tam ürün üretilene kadar devam eder. Özetle, Agile, görevleri kısa çalışma aşamalarına bölmeye, projeyi sık sık gözden geçirmeye ve yürütme sırasında gerektiğinde planlamayı

uyarlamaya odaklanan bir proje yönetim yöntemidir. Agile yaklaşımı, diğer proje yönetim yöntemlerine göre daha esnek ve daha az katıdır. (Stanley vd., 2020)

4.3.3. Agile Yaklaşımının Faydaları

Agile yaklaşımın faydaları aşağıdaki gibi sıralanabilir (Floara vd., 2014);

- Agile yöntemlerin bakış açısı nedeniyle, proje boyunca değişiklikler memnuniyetle karşılanır ve hatta teşvik edilir. Ayrıca, iteratif süreç tasarımı sayesinde uyarlanabilirlik yüksektir ve değişen koşullar kolayca yönetilir.
- Kısa toplantılar ve yüz yüze iletişime odaklanan yöntemler sayesinde, proje ekibinin güveni ve motivasyonu artar, bu da verimlilik seviyesini yükseltir.
- Projenin ilerleyişi, kısa süreli tekrarların sonucu olan çalışan ürün sayesinde daha sağlıklı bir şekilde izlenir. Proje riski azalır, potansiyel hatalar daha erken tespit edilir, teslimat daha öngörülebilir hale gelir ve ürün pazara daha hızlı ulaşır.
- Müşteriler veya kullanıcılar, proje süreci boyunca proje personeliyle yüz yüze iletişim kurabildiklerinden, iş ve yazılım geliştirme süreçlerinin entegrasyon seviyesi yükselir. Sonuç olarak, yazılımın işlevsel kalitesi artar ve istenen özelliklerin tam olarak ifade edilmesiyle müşteri memnuniyeti sağlanır.
- Döngünün sonunda iletilen işlevsel üründen gelen geri bildirimlerle belirlenen sorunlar, sonraki aşamalarda ele alınır ve ürünün destek ve bakım süreçleri daha basit hale getirilir, bakım ve destek maliyetleri azaltılır.
- Agile yöntemler, uygulanan aşamalar arasında geçişler ve geri dönüşler için uygun olduğundan, etkileşim sırasında belirlenen geriye dönük problemler kolayca düzeltilebilir.
- Genel olarak, üretilen yazılımın sadece ürün özellikleri dokümantasyona dönüştürülür ve diğer süreçler hariç tutulur. Bu nedenle, dokümantasyon algısı oluşmamalıdır. Hafif dokümantasyon ile dokümantasyon iş yükünü azaltır.
- Tüm iterasyonlarda entegrasyon ve test faaliyetlerinin gerçekleştirilmesi, bu alanda çalışan personelin iş yükünün birikmesini önler ve daha istikrarlı bir çaba sağlar.

4.3.4. Agile Yaklaşımının Kısıtlamaları

Agile yaklaşımın kısıtlamaları şöyledir (Floara vd., 2014);

- Ürünler, belirli sorunlar veya işlemlere göre tanımlanan gereksinimleri karşılamak için yeterli olan çözümlerle agile yöntemlerle geliştirildiği için, diğer projelerde yeniden kullanılamaz.
- Kalabalık projelerde iletişim ve etkileşim azaldıkça, agile yöntemlerin etkinliği düşer.
- Fonksiyonel bütünlük gerektiren büyük ve karmaşık sistemler, tekrarlayan parçalara bölünemeyebilir.
- Müşteri veya son kullanıcının katılımı düşük olduğunda, gereksinimler yeterince tanımlanamayabilir.
- Değişime açık olması nedeniyle, projenin kaynak ve kapsam planlaması zordur.
- “Gerekli olduğu kadar dokümantasyon” anlayışıyla hazırlanan dokümantasyon yetersiz kalırsa, projeye sonradan katılan geliştiriciler, yasal düzenleme ve sertifikasyon süreçleri ile bakım aşamaları olumsuz etkilenebilir.
- Hata toleransı olmayan sistemler için, agile yöntemlerin test ve kalite kontrol prosedürleri yeterli olmayabilir; çünkü bu durumlarda yazılım hataları insan hayatını tehlikeye atabilir veya önemli mali kayıplara yol açabilir.
- Taşeronun teklif sunabilmesi ve sözleşme yapılabilmesi için gereksinimlerin doğru ve değişmez olması gerekir. Agile yöntemlerin belirsiz ve değişken ortamı, taşeron uygulamalarını zorlaştırır.
- Takımlar kendi kendini organize ettiğinden, bireylerin becerikli, deneyimli ve agile süreçlerine hâkim olması gerekir. Yetersiz ve deneyimsiz personel, takım performansını düşürebilir ve bu da proje geliştirme sürecini olumsuz etkileyebilir.
- Önceliklerini tam olarak tanımlayamayan ve iş listesinde sık sık öncelik değiştirmek isteyen müşteriler, planlara zarar verebilir.
- Agile yöntemlerde müşteri faydası önceliklidir, ancak müşteri faydası genellikle müşteri işlevlerini karşılar. Basitlik felsefesi yalın gelişimin bir mirası olarak uygulandığından, kullanıcı arayüzü ve kullanıcı deneyimi gibi kullanıcı memnuniyeti konuları göz ardı edilebilir.

4.3.5. Agile Yaklaşım Alt türleri

Farklı prensipler uygulanarak geliştirilen çeşitli Agile prensip çerçeveleri olmasına rağmen, birkaç yöntem yaygın olarak kullanılmaktadır. Tüm bu farklı Agile yöntemlerin

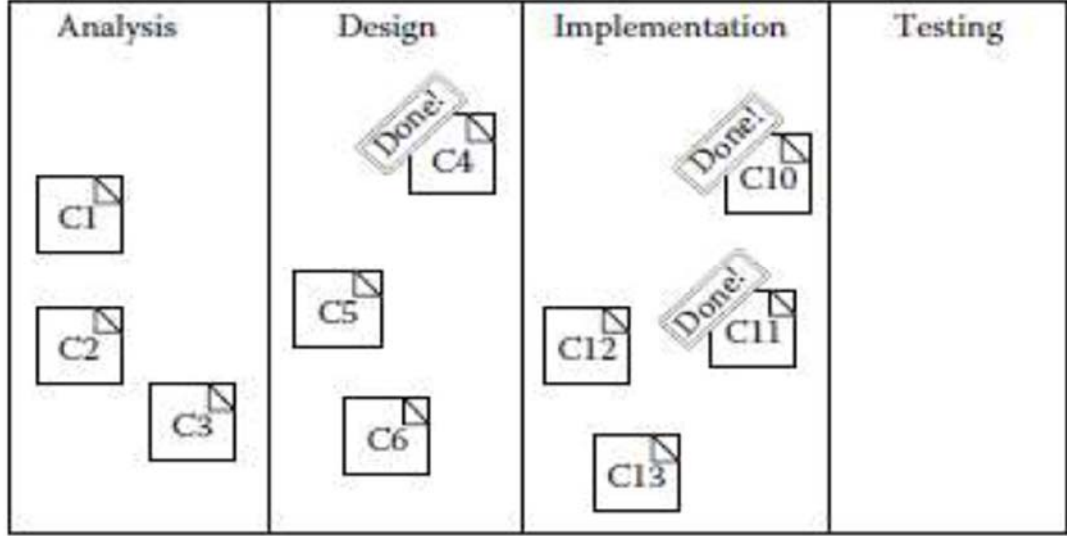
özü; çalışan bir ürünü sık aralıklarla teslim etmeye ve iş ve problem çözümlerine hızlı bir şekilde odaklanmaya yöneliktir. (Rajagopalan vd., 2016). Aşağıda en popüler iki yöntem yakından incelenmektedir.

4.3.5.1. Kanban

“Kanban, israfı ortadan kaldıran, tam zamanında, çekme tabanlı bir üretim sistemi kullanarak yönetim pratiğidir.” Toyota otomobil üreticisinde çalışan Taiichi Ohno, araba üretiminde verimliliği artırmak için bir fikir geliştirdi. Kanban, Japonca'da "tabela" anlamına gelir. Bu, çekme sürecini izlemek için bir tabelanın kullanılması gerektiği gerçeğinden gelir. Tabelalar, üretim zincirinin farklı aşamalarına karşılık gelen farklı alanlarda düzenlenir. Farklı iş ögelerini temsil etmek için kartlar kullanılır ve her kart üretim hattında monte edilen farklı bir ögeyi temsil eder. İş ögeleri sıralandıkça kartlar ilan tahtasında hareket eder ve ilerlemeyi ve ihtiyaçları takip etmenin basit bir yolunu sağlar. (Villafiorita, 2014). Kanban kullanımı 1940'larda başlamışken, bu yöntemin bilişim endüstrisinde kullanımı 2004 yılına dayanmaktadır. David J. Anderson, Microsoft'ta küçük ve sorunlu bir ekibe yardım ediyordu. Kanban kullanımı, mevcut ve gelecekteki çalışmaların görselleştirilmesiyle başladı. (Ahmad vd., 2013). Bu tablo, farklı süreçleri gösteren farklı sütunlar içerir. Genellikle bu sütunlar belirli durumlarla adlandırılır. Örneğin; açık, üzerinde çalışılıyor, gözden geçiriliyor ve tamamlandı gibi durumlar çoğunlukla kullanılır. Geliştirme süreçlerinin bulunduğu bilgi sistemleri departmanları, tasarım ve geliştirme gibi yukarıda belirtilen durumların yanı sıra ek olarak kullanılır. Böylece, kartın, yani görevin mevcut durumu bu tablo üzerinde hızlı bir şekilde takip edilebilir. Aşağıdaki şekil, bir Kanban tahtasını göstererek süreci illüstre eder (Villafiorita, 2014).

Şekil 3

Kanban Tahtası



Kanban prensipleri üç ana sütun üzerine kuruludur.

1. Takım içinde şeffaflığı artırmak ve üzerinde çalışılan proje veya ürünün durumunu göstermek için, fiziksel ortamda kullanılabilecek tablo şeklinde grafiklerin veya internet üzerinden takip edilebilecek grafiklerin kullanımını sağlayan araçlar kullanılır. Takım, birlikte karar vererek tabelada gösterilecek sütunları adlandırabilir. Görselleştirme, ana prensiplerden biridir.
2. Üzerinde çalışılan işlerin sayısını sınırlamak, yani iş ilerleme sınırı belirlemektir. Sınır, takım üyelerinin maksimum çalışma kapasitesini gösterir. Amaç, takım üyelerinin kapasitesinin ötesinde iş almasını önlemektir. İş ilerleme sınırı dolmuş olan bir takım üyesi, mevcut işini tamamlamadan yeni bir işe başlamamalıdır. Sınırın belirlenmesi ile daha kaliteli işler ortaya çıkacak ve takım üyelerinin çoklu görev yapması engellenecektir.
3. İş akışını artırmak ve teslim süresini kısaltmaktır. Örneğin, iş tamamlandığında ve iş ilerleme sınırı maksimuma ulaşmadığında, bekleyen iş listesinden en önemli ve öncelikli iş takım üyesi tarafından alınır ve üzerinde çalışılmaya başlanır. Kanban prensipleri üç ana prensip üzerine kuruludur; işin görünür hale getirilmesi, takım üyelerinin sınırları dahilinde çalışmalarına izin verilmesi ile çalışan, belirlenen zamanda en değerli çıktıyı üretir (Villaforita, 2014).

4.3.5.2. Scrum

Schwaber ve Sutherland (2017) 'e göre, "İnsanların karmaşık uyarlanabilir sorunları ele alabileceği, bu sırada en yüksek değere sahip ürünleri verimli ve yaratıcı bir şekilde teslim edebileceği bir çerçeve." Ken Schwaber ve Jeff Sutherland tarafından geliştirilen Scrum yöntemi, geniş ve karmaşık yapıları geliştirmede uygulanır. Scrum bir kısaltma değildir, Rugby sporundan alınan bir terimdir. Hata durumunda veya top oyun dışına çıktığında oyunu yeniden başlatma yöntemini ifade eder (Stanley vd., 2020).

Scrum, türler, süreçler ve teknikler kullanabileceğimiz bir çerçevedir. Bunu yaparken, aşağıdaki destekleyici kavramlar kullanılır (Schwaber vd., 2020):

- Şeffaflık: Yapılan iş tüm takım ile paylaşılmalı ve takımın tüm üyeleri için bitişini kabul etmelidir.
- Gözlem: İşin doğru bir şekilde tamamlanması için yapılan iş sık sık takip edilmelidir.
- Adaptasyon: Gözlemden sorumlu kişi, işin tamamlanmasının karmaşık ve kabul edilemez olduğunu öngörürse, üründe hızlı bir değişiklik yaparak geliştirme sürecinin sapmalarını ve zorluklarını önleyecektir.

Scrum metodolojisinin temel prensipleri (Gechman, 2019):

- Kendi Kendini Organize Eden Takımlar: Agile'ın güçlü yönlerinden biri, takım üyelerinin belirli kısa vadeli hedefler üzerinde çalışırken sürekli etkileşimde bulunduğu, çok işlevli kendi kendini organize eden takımlardır. Scrum takımı, çapraz fonksiyonel ve kendi kendini organize eden bir takımdır.
- Kendi Kendini Yöneten Takımlar: Scrum, düzenli geri bildirim ve kısa vadeli planlama ile hedefe ulaşmayı sağlar. Takımdaki herkes, hedefin ne olduğunu anlamalı, işin zorlu olduğunu bilerek doğru yönde ilerlemeye ve odaklanmaya devam etmelidir.
- Bilgi Paylaşımı: Bilgi, tüm takım üyeleri tarafından paylaşılır. İletişim, takım üyelerinin aynı yerde bulunmasıyla kolaylaştırılır ve bazen kalan işi gösteren yanma tabloları gibi bilgilendirici grafikler içeren "takım odaları" ile güçlendirilir.
- Minimum Süreçler ve Prensipler: Agile yaklaşımı, işleri mevcut iş süreçlerini yönetmek için gereken minimum süreçlere indirerek işler.

- Sık Sürüm Çıkma: Çoğu agile yaklaşımı, geleneksel yöntemlerden daha sık yazılım sürümleri çıkarır. Agile takımları, gereksinimleri anladıklarını doğrulamak için sık sık kontrol noktaları elde ederek kodu erken ve sık teslim ederler.
- Sürekli Test: Agile takımları, her iterasyonun sonunda test yaparak tüm planlanan özelliklerin yerine getirildiğinden emin olurlar. Bu sık test etme, riski azaltmaya yardımcı olur. Bu şekilde, proje yaşam döngüsünün erken aşamalarında daha fazla sorun tespit edilir.
- Destekleyici Altyapı: Yeterince stabil yazılım üretmek ve belirlenen kalite seviyesinde çalışmak için, özellikle büyük uygulamalar için destekleyici bir altyapıya ihtiyaç vardır. Örneğin, uygulamalar tarafından doğrulanan sürekli derlemeler, otomatikleştirilmiş birim seviyesinde test çerçevesi, otomatikleştirilmiş sistem seviyesinde regresyon testleri ve otomatikleştirilmiş yazılım sürüm süreci.
- Müşteri İş birliği: Yazılım geliştirme sürecinde ihtiyaç ve önem sırasına göre çalışarak, büyük taşların önceden yerleştirilmesini mümkün kılar. Günümüzde, büyük taşlar zamanla müşteri ihtiyaçlarına göre şekillendiği için, müşteri geri bildirimlerine göre yapılandırmayı mümkün kılar. Bu, Scrum metodolojisinin ayırt edici özelliğidir ve yüz yüze iletişim, etkinliğini en üst düzeye çıkarır. Ancak, sistemin karmaşıklığı ve büyüklüğü arttıkça, tam zamanlı çalışan, iyi bir iletişimci olan ve tüm sistemi anlayan tek bir kişi bulmak zor olabilir.

4.3.5.2.1. Scrum Takımı ve Roller. Scrum'da takım çalışması çok önemlidir. Takımdaki her bireyin kendi sorumlulukları vardır. Çalışma dönemi boyunca, takım üyeleri sonuç odaklı bir şekilde çalışmaları tamamlamalıdır. Gündeme getirilen bir işin nasıl yapılacağı ve yeniliklerin uygulanabilme yeteneği Scrum takımına aittir. Dört tür ekip yapısı görülebilir. Yönetici liderliğindeki takım yapısında, takım biraz daha pasif olup yönlendirmeye ihtiyaç duyar. Kendi kendini organize eden takım yapısında ise, Scrum takım üyelerinin görevleri ve rolleri önceden belirlenmiştir. Bu nedenle, işin nasıl yapılacağı Scrum takımına aittir. Kendi takımını kuran takım, gelen işleri, iş takibini ve süreç yönetimini çalışarak daha bağımsız hareket eder. Scrum rolleri, tüm takımın katılımıyla belirlenir.

Kendi kendini yöneten bir takım, daha üst düzey bir yönetim yapısını ifade eder. Yapılacak işin anlamını ve amacını vurgular. Genel durumdan sorumludur ve daha geniş bir perspektiften bakar (Gechman, 2019).

Scrum Master: Agile literatüründe, Takım Lideri veya Koç olarak da adlandırılan Scrum Master, Agile Takımı ile dış dünya arasında arayüz olarak hareket etmek de dahil olmak üzere, Yazılım Proje Yöneticisine benzer bazı işlevleri yerine getirir. Scrum Master'ın bazı görevleri şunlardır:

- Agile sürecinin koruyucusudur ve sürecin sorunsuz işlenmesini sağlar.
- Verimliliği etkileyen engelleri ortadan kaldırır ve kritik toplantıları kolaylaştırır.
- Scrum'ı yeterince iyi anlayarak diğer rolleri eğitmek ve yönlendirmekten ve diğer paydaşlara eğitim ve yardım sağlamaktan sorumludur.
- Projenin durumu hakkında sürekli farkındalık sağlar. Ancak, takımı yönetmez ve takım üyelerine görev atayamaz; görev atamaları takımın iç sorumluluğundadır.
- Sprint sırasında yeni gereksinim değişikliklerinin yapılmasını engeller. Takımın dikkatini dağıtmadan görevlerini yerine getirmesine ve ortaya çıkan dengeleyici ortamdan faydalanmasına olanak tanır (Heldman, 2018; Gechman, 2019).

Product Owner: Ürün Sahibi, müşteri gereksinimlerinin koruyucusu ve tüm gereksinimler ile planlanan uygulama sırası konusunda tek iletişim noktasıdır. Ürün Sahibi, organizasyon, müşteri, ürünle ilgili ihtiyaçlar ve takım arasında arayüz görevi görür. Projenin "patronu" olmakla birlikte, üyelerin kendi hedeflerini belirlemelerine ve kendilerini yönetmelerine olanak tanır. Her projenin sadece bir Ürün Sahibi vardır. Ürün Sahibinin bazı görevleri şunlardır:

- Backlog'un görevlerini net bir şekilde ifade eder.
- Takımı, birden fazla kaynaktan gelen özellik ve hata düzeltme taleplerinden kurtarır.
- Herkese önceliklerin ne olduğunu ve sıradaki işlerin neler olduğunu bildirir.
- İş partilerini herkes için görünür ve anlaşılır hale getirir. (Stanley vd., 2020)

Scrum Ekibi: Bu, ürünü geliştirme ve test etme işini pratikte yapan, kendi kendini organize eden, çapraz fonksiyonel bir grup insandır. Takım ürünün üretiminden sorumlu olduğundan, işin nasıl yapılacağına karar verme yetkisine de sahiptir. Takım üyeleri, işin önceliğine ve yeteneklerine göre görevlere kaydolurlar. İş için tahminlerde bulunurlar ve iterasyon süresini dolduracak kadar görev üstlenirler. Agile takımları kendi kendini yöneten, kendi kendini organize eden ve kendi kendini yöneten takımlardır. Takım büyüklüğü genellikle 5-9 kişi arasında tutulur. Bir Scrum Takımı, scrum master, product owner ve diğer takım üyelerinden oluşmaktadır. (Gechman, 2019; Heldman, 2018). Aşağıdaki şekil, tüm bu rollerin bir projede nasıl birlikte çalıştığını göstermektedir (Stanley vd., 2020).

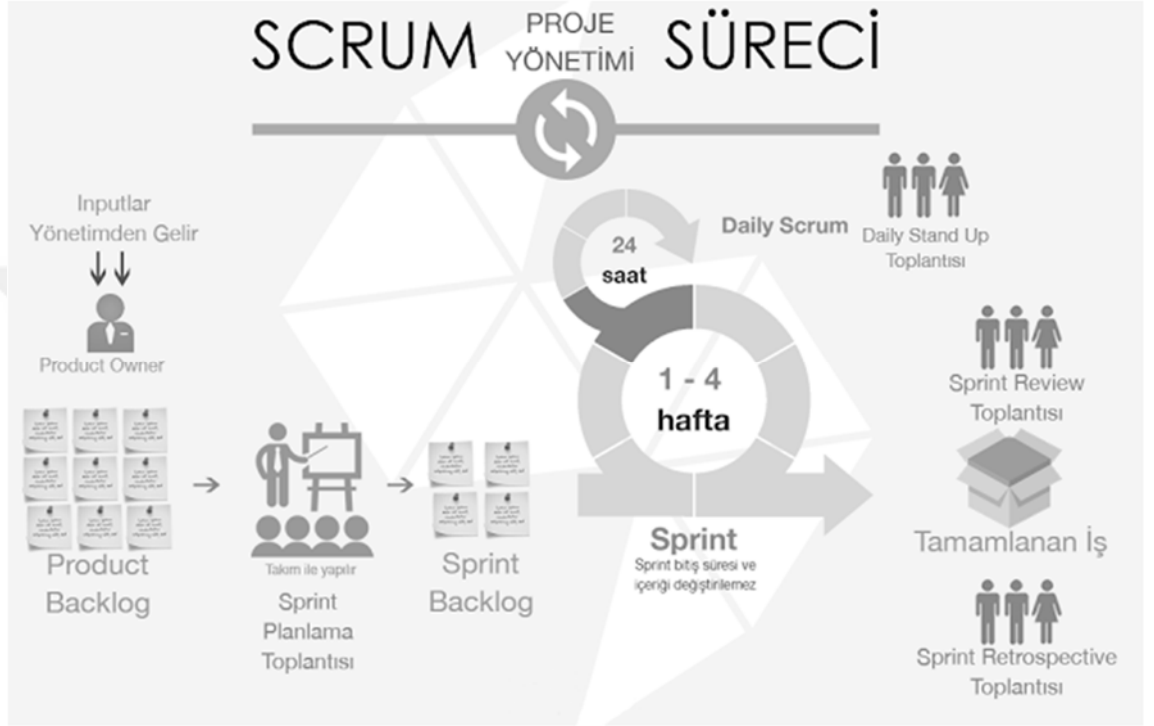
4.3.5.2.2. Scrum Aktiviteleri. Sprint Planlama: Sprint, iş listesinden seçilen işlerin sınırlı bir süre içinde tamamlandığı dönemdir. Tüm Scrum takımı, sprint planlamasına katılır ve Scrum Master toplantıyı yönetir. Sprint süresi genellikle bir hafta, iki hafta veya 30 gündür. Sprint süresi belirlendikten sonra, süre ile ilgili herhangi bir değişiklik yapılmamalıdır. Her Sprint sonu, diğer Sprint için bir başlangıçtır. Örneğin, bir aylık bir Sprint için sprint planlaması yaklaşık sekiz saattir. Sprint süresi kısaldıkça, bu süre de kısılır. Schwaber ve Sutherland (2020), Sprint Planlaması için şu soruları yanıtlamıştır: "Bu Sprint çalışan ürünü olarak ne teslim edilebilir? Ürünün çalışan parçası nasıl teslim edilecek?" şeklinde açıklamışlardır.

Ürünün bir parçası olarak ne teslim edileceği konusunda, ürün sahibi tüm Scrum takımına iş listesindeki öncelikli işleri anlatır, Scrum takımını neden gerekli olduğu ve iş tamamlandığında ürüne nasıl değer katacağı konusunda bilgilendirir. Öte yandan, geliştirme ekibi mevcut kapasitesini takım ile paylaşır ve kapasite sınırları dahilinde işleri seçer. Böylece, tüm Scrum takımı Sprint hedefini birlikte belirler. Mevcut Sprint'in tamamlanması ise, tüm planlanan işlerin verilen süre içinde "Tamamlandı" tanımına uygun olarak gerçekleştirilmesi ile olur. Takım, iş listesini değerlendirip nasıl gerçekleşeceğini belirledikten sonra Sprint planını ve Sprint iş listesini oluşturur.

Planlama yapıldıktan sonra, geliştirme ekibi Sprint backlog'undaki işleri daha küçük parçalara bölebilir. Böylece daha verimli çalışmayı amaçlar. Geliştirme süreci tamamen geliştirme ekibinin sorumluluğundadır (Stanley vd., 2020, s. 47).

Şekil 4

Scrum Süreci



Kaynak: medium.com/@zeliyacabuk

Sprint, başarılı olmadığı ve hedefi tehlikeye atan bir durum ortaya çıktığı takdirde iptal edilebilir. İptal yalnızca ürün sahibi tarafından yapılır. İptal nadir görülür. İptal edilen Sprint içindeki tamamlanmamış işlerin geliştirme çabaları yeniden değerlendirilir ve bu işler geri backlog'a gönderilir. Tamamlanan işler ürün sahibi tarafından kabul edilir (Schwaber vd., 2020, s. 8).

Daily (Toplantı): Daily Scrum, geliştirme ekibiyle yapılan ve sadece 15 dakika süren bir toplantıdır. Böylece, takım içindeki Sprint'teki işlerin durumu takım tarafından şeffaf bir şekilde bilinir. Her gün, o gün ne yapılacağı, varsa işleri engelleyen durumlar gibi konular kısaca ifade edilir. Bu şekilde, Sprint hedefinde oluşabilecek engeller önceden belirlenir. Günlük toplantıların içeriğinin yanlış anlaşılması oldukça yaygındır. İlk olarak, takım üyeleri bunun bir raporlama toplantısı olmadığını, aksine takım içinde şeffaflığı sağlamak ve potansiyel engelleri ortadan kaldırmak için yapıldığını anlamalıdır. (Heldman, 2018, s. 44) Bu toplantı iki şekilde yönetilebilir. Birincisi, takım

üyelerinin daha fazla konuşarak durumu güncellemesidir. İkincisi ise, geliştirme ekibi üyelerinin aşağıdaki soruları cevaplamasıyla yapılır ve bu sorular Schwaber ve Sutherland (2017) tarafından şu şekilde belirtilmiştir:

- Dün, ekibime sprint hedefine ulaşmasına nasıl yardımcı oldum?
- Bugün, ekibime sprint hedefine ulaşmasına nasıl yardımcı olacağım?
- Ekibin veya benim sprint hedefine ulaşmamı engelleyen herhangi bir engel görüyor muyum?

Sprint Rewiew (Sprint Gözden Geçirme): Sprint değerlendirme toplantısı, her Sprint'in sonunda yapılır. Bu toplantıda, Sprint boyunca geliştirilen ürünle yapılanlar gösterilir. Böylece, yapılan işin kattığı değer ve sonrasında neler yapılabileceği bu toplantıda değerlendirilir ve katılımcılarla paylaşılır. Bu toplantıyı ürün sahibi düzenler. Ürün sahibi, ilgili paydaşları ve Scrum takımını davet eder. Ürün sahibi, mevcut Sprint hedefi içindeki hangi işlerin yapıldığını ve hangilerinin yapılmadığını ve nedenini açıklar. Öte yandan, geliştirme ekibi, üzerinde çalıştıkları her işi detaylı bir şekilde katılımcılara iletir ve sorularını yanıtlar. Toplantının sonunda, katılımcılardan gelen sorular ve geri bildirimler çerçevesinde işbirliği sağlanır. Toplantıda müşteriler, ürün sahibi, geliştirme ekibi, Scrum Master, yöneticiler, CEO ve diğer katılımcılar bulunur. (Heldman, 2018, s. 45)

Sprint Retrospective: Sprint Retrospective, Scrum takımının bir sonraki adımları daha dikkatli atması için aynaya baktığı bir toplantıdır. Bu toplantı sırasında, retrospektif toplantıyla sona eren Sprint boyunca iletişim, açıklık, planlama ve takım çalışması gibi kriterler göz önünde bulundurularak bir değerlendirme yapılır. Bu toplantıdaki girdiler, bir sonraki Sprint için iyileştirme maddeleri haline gelir ve daha olumlu bir atmosfer yaratır. Toplantı, Sprint değerlendirme toplantısından sonra yapılır. Bir aylık bir Sprint için yaklaşık 3 saat süren bir retrospektif toplantı yeterli olacaktır. Retrospektif toplantı iki temel soru etrafında ilerler:

a. Neler iyi gitti?

b. Neler daha iyi yapılabilirdi?

Bu iki ana soruya dayanarak, takım üyelerinin görüşleriyle ortak bir payda oluşturmaya çalışılır ve bu noktalar bir sonraki Sprint'te dikkate alınır. Böylece daha iyi takım çalışması ve iş planlaması amaçlanır. Toplantının başında işbirlikçi bir ortam yaratılmalı, takım değerleri ve çalışma anlaşması göz ardı edilmemelidir.

Retrospektiften sonra, takım kararı doğrultusunda öncelikli olarak alınacak aksiyonlar belirlenir. Alınan kararlar için detaylı planlama yapılır. Son olarak, ölçülebilir sonuçlarla yeni hedefler belirlenir. (Villaforita, 2014, s. 223-225).



BÖLÜM 5. AGİLE VE WATERFALL ARASINDAKİ FARKLAR

5.1. Agile

Agile, esnek ve çevik bir projenin yönetiminde, işlevsel özelliklerin hızlı bir şekilde ve düzenli aralıklarla sağlandığı bir geliştirme yöntemidir. Agile yöntemi, hızlı ve sürekli iyileştirmeyi, sürekli müşteri geri bildirimini ve işlevselliği artırmayı teşvik eder.

Agile, projenin karmaşıklığı ve belirsizliği yüksek olduğunda ve gereksinimler sürekli olarak değişiyorsa kullanılmalıdır. Müşteri ile sürekli etkileşim gerektiren ve belirsiz veya değişken gereksinimlere sahip projeler için idealdir. Bu yaklaşım, proje ekibinin hızlı bir şekilde değişen koşullara uyum sağlamasını ve müşteri ihtiyaçlarına hızlı bir şekilde yanıt vermesini sağlar (Handscorn, 2018).

5.2. Waterfall

Waterfall, bir projenin aşamalarının, bir sonraki aşamaya geçmeden önce önceki aşamanın tamamlanmasını gerektiren sıralı bir modeldir. Bu, gereksinimlerin belirlendiği ve değişikliklere açık olmadığı daha düzenli, yapılandırılmış ve belirgin bir yaklaşımdır.

Waterfall, projenin gereksinimlerinin iyi tanımlanmış ve değişmeyecek olduğu durumlarda kullanılır. Projenin ne olması gerektiği konusunda net bir vizyonun olduğu, risklerin ve belirsizliklerin düşük olduğu projeler için uygundur. Bu yaklaşım, proje sürecinin baştan sona planlanması ve adımların sıralı olarak tamamlanması gerektiği durumlarda etkilidir (Laanti, 2008).

5.3. Agile ve Waterfall Arasındaki Farklar

Yeşil'e (2019) göre Waterfall ve Agile arasında ki farklar;

1. **Esneklik:** Agile metodoloji, gereksinimlerin ve çözümlerin zaman içinde organik olarak evrildiği dinamik bir süreçtir. Bu, değişen koşullara ve belirsizliklere daha esnek bir şekilde uyum sağlar. Diğer yandan, Waterfall yaklaşımı daha lineer bir yapıya sahiptir ve baştan sona belirlenen adımları takip eder.

Agile'in Esnekliği:

Agile metodolojisi, isminin de ima ettiđi gibi, esnek ve eviktir. zellikle gereksinimlerin belirsiz olduđu veya zamanla deđiřebileceđi durumlarda kullanıřlıdır. rneđin, bir yazılım projesi bařladıđında tm gereksinimlerin net olarak bilinmediđi veya mřteri gereksinimlerinin zamanla deđiřebileceđi durumlarda Agile yntemler tercih edilir. Bu metodoloji, projeyi kısa sreli "sprintler" veya ařamalar halinde blerek iřler. Her sprintin sonunda, ekip rnn ilerlemesini deđerlendirir, yeni đrenilen bilgilere dayanarak gelecek planlarını gnceller ve gereksinimlerin nasıl deđiřtiđini ve projenin adaptasyonunu nasıl gerekleřtirebileceđini gzden geirir (Yeřil, 2019).

Waterfall'un Esnekliđi: Waterfall modeli, projenin her ařamasının bir ncekine sıkı sıkıya bađlı olduđu bir sretir. Bařlangıta belirlenen gereksinimlerin ve zmlerin projenin ilerlemesi sırasında genellikle deđiřtirilmediđi dřnlr. Bu yntem, gereksinimlerin bařlangıta net olarak anlařıldıđı ve projenin sresi boyunca pek fazla deđiřmeyeceđi durumlar iin uygundur, genellikle daha az belirsizlik ieren projeler iin tercih edilir. Ancak, gereksinimlerin deđiřebileceđi bir proje durumunda, Waterfall modeli esneklik sađlamakta zorlanabilir.

Esneklik aısından Agile ve Waterfall arasında seim yaparken, projenin gereksinimlerinin ne kadar iyi anlařıldıđını ve bu gereksinimlerin zamanla ne kadar deđiřebileceđini dikkate almak nemlidir. Agile yntemi, belirsizlik ve deđiřikliklere daha iyi uyum sađlayabilirken, Waterfall daha belirgin ve deđiřmeyen gereksinimlerle daha uyumludur (Yeřil, 2019).

2. **Ařamalar:** Waterfall modeli, bir ařamadan diđerine geerken nceki ařamanın tamamlanmasını bekler ve ardıřık bir ilerleme sađlar. Diđer taraftan, Agile modeli, projeyi paralara bler ve her bir parayı ayrı ayrı geliřtirerek ve test ederek esneklik sađlar. Bu řekilde, projenin esnekliđi ve adaptasyonu artar.

Agile'ın Ařamaları: Agile metodolojisi, projeyi kk paralara veya "sprintlere" bler. Her bir sprint genellikle birka hafta srer ve kendi gereksinimlerini, tasarımını, kodlamasını ve test etme srelerini iinde barındırır. Bu sebeple, her sprint, kendine zg bir proje olarak deđerlendirilebilir. Bir sprint tamamlandıđında, ekip geliřtirdiđi rnn zerinde bir inceleme ve deđerlendirme yapar ve ardından bir sonraki sprint iin planlama yapılır. Bu sre, projedeki tm gereksinimler karřılanana kadar devam eder. Bu yaklařım, ekibin taleplere hızla yanıt vermesine ve projenin nceliklerini veya gereksinimlerini deđiřtirebilmesine imkan sađlar. Bir sonraki sprint iin planlama yapılırken, nceki sprintlerden elde edilen geri bildirimler ve đrenilenler dikkate alınır.

Sonuç olarak, Agile ve Waterfall metodolojileri arasındaki farklar, projenin esneklik ihtiyaçlarına, belirsizlik seviyesine ve gereksinimlerin değişkenliğine bağlıdır. Agile, değişken ve belirsiz projelerde tercih edilir; projeyi sprintler halinde yönetir ve hızla adapte olabilir. Waterfall ise sabit ve önceden belirlenmiş gereksinimlere sahip projeler için uygundur; aşamalı bir ilerleme sağlar ancak değişikliklere adapte etmek zordur. (Flora vd., 2014).

Waterfall'un Aşamaları: Waterfall modeli, bir projenin gelişiminin belirli ve sıralı aşamalardan oluştuğu planlı bir yaklaşımdır. Bu aşamalar genellikle gereksinimlerin toplanması, tasarım, kodlama, test etme ve bakım gibi adımları içerir. Her aşama, bir sonraki aşamaya geçilmeden önce tamamlanmalıdır. Örneğin, bir yazılım geliştirme projesinde, ilk olarak tüm gereksinimler toplanır ve belgelendirilir. Ardından, belirlenen gereksinimler tasarım aşamasına geçilmeden önce ayrıntılı bir şekilde incelenir. Tasarım tamamlandıktan sonra, kodlama aşamasına geçilir ve yazılım üretilir. Daha sonra, yazılım test edilir ve hatalar düzeltilir. Her aşama tamamlandıktan sonra bir sonraki aşamaya geçilir ve genellikle bir önceki aşamaya dönüş yapılmaz (Flora vd., 2014).

3. Müşteri Katılımı: Agile yaklaşımı, müşteri geri bildirimini ve katılımını sürekli olarak teşvik eder; bu da ürünün müşteri gereksinimlerine daha yakından uyarlanmasını sağlar. Waterfall modelinde ise müşteri genellikle yalnızca proje başında ve sonunda yer alır ve geri bildirim süreci daha sınırlıdır.

Agile'da Müşteri Katılımı: Agile, müşteri katılımını ve geri bildirimleri sürekli teşvik eder. Her sprint sonrasında yapılan inceleme toplantıları, bu teşviki destekler. Bu toplantılarda, müşteriler ya da paydaşlar, ekibin yaptıklarını gözlemleyebilir, yeni özellikleri test edebilir ve geri bildirimlerini iletebilirler. Bu geri bildirimler, bir sonraki sprintin planlamasını etkiler ve ekibin müşteri ihtiyaçlarına ve beklentilerine daha hızlı yanıt vermesini sağlar. Agile yaklaşımının önemli bir özelliği, "Müşteri İş birliği" prensibidir. Agile Manifesto'ya göre, "Müşteri iş birliği, sözleşme müzakeresinden daha önemlidir." Bu, müşteri ve geliştirme ekibi arasındaki sürekli iletişim ve iş birliğini vurgular (Handscorn, 2018).

Waterfall'da Müşteri Katılımı: Waterfall modeli, genellikle projenin başında ve sonunda müşteri katılımını odaklar. Proje başlangıcında, müşteri gereksinimlerini ve beklentilerini belirler ve bu, projenin tüm aşamalarını yönlendirir. Ancak, projenin

ilerlemesi sırasında, müşteri doğrudan etkileşime girmez ve genellikle projenin sonunda bir revizyon ve onay süreciyle karşılaşılır. Bu yaklaşım, gereksinimlerin başlangıçta tam olarak belirlendiği ve projenin süresi boyunca değişmeyeceği durumlarda etkilidir. Ancak, gereksinimlerin veya beklentilerin değişebileceği durumlarda, Waterfall modeli, Agile metodolojisinin sunduğu sürekli müşteri geri bildirimini ve katılımını sağlamayabilir.

Sonuç olarak, Agile ve Waterfall metodolojileri arasında müşteri katılımı ve geri bildirim konusunda önemli farklar mevcuttur. Agile, müşteri katılımını ve sürekli geri bildirimini teşvik ederken, Waterfall genellikle projenin başında ve sonunda müşteri katılımını ve geri bildirimini içerir. Bu farklılıklar, projenin gereksinimlerinin ve beklentilerinin ne kadar değişebileceğine bağlıdır (Handscomb, 2018).

- 4. Riskler ve Belirsizlikler:** Agile metodoloji, belirsizlikler ve risklerle daha iyi başa çıkabilir çünkü bu tür durumlara daha hızlı ve esnek bir şekilde uyum sağlayabilir. Öte yandan, Waterfall modelinde belirsizlikler ve riskler genellikle projenin başında ele alınır ve sonraki aşamalarda değişiklik yapmak daha zor olabilir.

Agile ve Riskler/Belirsizlikler: Agile metodolojisi, belirsizlikler ve risklerle daha etkin bir şekilde başa çıkmak amacıyla özel olarak tasarlanmıştır. Sürekli geri bildirim döngüsü ve iteratif yapısı, belirsizlikleri ve riskleri yönetmek için daha fazla imkân sağlar. Her sprint sonrası, ekip projenin hangi yönlerinin iyi çalıştığını ve hangi yönlerin iyileştirilmesi gerektiği konusunda değerlendirmede bulunur. Bu süreç içerisinde herhangi bir risk yada belirsizlik tespit edilirse, ekip bunları ele alır ve bir sonraki sprinti buna göre ayarlar. Örneğin, bir özellik istenen şekilde çalışmıyorsa veya müşteri beklentilerini karşılamıyorsa, ekip bu durumu düzeltmek için konuyu bir sonraki sprintte ele alabilir. Bu, riskleri hızlı ve düzenli olarak ele alma ve bunları hızlı bir şekilde yönetme fırsatı sağlar (Laanti, 2008).

Waterfall ve Riskler/Belirsizlikler: Waterfall metodolojisi, projenin başlangıcında belirsizlikleri ve riskleri ele almayı gerektirir. Başlamadan önce tüm gereksinimlerin ve beklentilerin belirlenmesi ve detaylı bir plan yapılması önemlidir. Bu aşamada, ekip belirsizlikleri ve riskleri değerlendirir ve bunları yönetmek için çeşitli stratejiler geliştirir.

Ancak, Waterfall metodolojisiinde belirsizlikler ve riskler projenin en bařında ele alındığından dolayı, projenin sonraki aşamalarında ortaya çıkan belirsizliklere ve risklere yanıt vermek daha zordur. Eğer projenin sonraki aşamalarında beklenmedik bir belirsizlik yada risk ortaya çıkarsa, bu sorunu yönetmek için genellikle daha az esneklik vardır.

Sonuç olarak, her iki metodoloji de belirsizlik ve risk yönetiminde benzersiz bir yaklaşım sunar. Agile, hızlı deęişimlere ve belirsizliklere uyum sağlayabilmesiyle ön plana çıkar, Waterfall ise daha kapsamlı ve başlangıçta belirlenen bir plana dayanır. Projeye özgü ihtiyaçlar ve gereksinimler, hangi metodolojinin tercih edileceğine karar vermede önemli bir rol oynar.



BÖLÜM 6. YÖNTEM VE BULGULAR

Bu bölümde, IT sektörü çalışanlarının anket sorularına verdikleri cevapların istatistiksel analizinden elde edilen bulgular sunulmuştur.

6.1. Örneklem İlişkin Genel Veriler

Örneklemeye dair genel veriler, 582 katılımcıdan elde edilen veri seti üzerinden analiz edilmiştir.

Agile ve Waterfall metodolojileriyle çalışan toplam katılımcı sayısı 570'dir. Bu katılımcıların %46,9'u kadın, %53,1'i ise erkek bireylerden oluşmaktadır. Toplam katılımcıların %83,2'si Agile, %16,8'i Waterfall proje yönetim modeliyle çalışmaktadır.

Tablo 3

Katılımcıların Cinsiyet ve Proje Yönetim Metoduna Göre Dağılımları

Cinsiyet	Proje Yönetim Modeli	Kişi Adeti	% Cinsiyet	% Proje Yönetim Modeli	% Toplam
Erkek	Agile	243	81,8%	51,3%	42,6%
	Waterfall	54	18,2%	18,2%	9,5%
Kadın	Agile	231	84,6%	48,7%	40,5%
	Waterfall	42	15,4%	16,8%	7,4%
Toplam	Agile	474	83,2%	100%	83,2%
	Waterfall	96	16,8%	100%	16,8%
Bütün Toplam		570	100%	100%	100%

6.2. Araştırmada Kullanılan Faktörlere Ait Analiz

Ankette kullanılan sorular, ilgili olduğu metoda göre kodlanmıştır. "A" ile başlayan sorular Agile yöntemine, "W" ile başlayan sorular Waterfall yöntemine, "O" ile başlayan sorular ise her iki yönteme de yakın olan sorulara işaret etmektedir. Anket altı faktör (Ekip, Proje, Süreç, Organizasyonel, Teknik ve Müşteri İlişkileri) ile gruplandırılmıştır.

Tablo 4

Faktör Analizi

Sorular	Ekip	Proje	Süreç	Organizasyonel	Teknik	Müşteri İlişkileri
O7-[Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir.]	0.789					
A4-[Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir.]	0.74					
O6-[Proje ekibinin teknik yeterliliği ve uzmanlık seviyesi projeye uygundur.]	0.721					
A6-[Proje ekibi işleri kendi aralarında organize edebilir.]	0.668					
A5-[Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir.]	0.626					

Tablolar Devamı		
A14-[Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir.]	0.491	
O10-[Projede, teknik standartlara uyum konusunda problem yaşanmaz.]	0.475	
O5-[Çalışma ortamımız proje için uygundur.]	0.471	
W4-[Proje sürecinin en başında maliyet ve kaynak planlaması detaylı olarak yapılır.]	0.824	
W5-[Risk analizi, proje sürecinin başında yapılır.]	0.814	
W3-[Proje kapsamı, proje sürecinin en başında net olarak belirlenir.]	0.799	
O12-[Proje ekibine gereksinim duyulan konularda eğitim verilir ve teknik bilgi seviyesi artırılır.]	0.523	
W1-[Projenin kapsamı ve hedefleri net tanımlanmıştır.]	0.523	
A13-[Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan bir yapı tercih edilmiştir.]	0.48	
W2-[Projede birim ve entegrasyon testine önem verilmektedir.]	0.456	
A10-[Projede aksiyon alma konusunda hızlı ve esneğiz.]	0.783	
[Proje süreçleri hızlı ve esnek bir şekilde yönetilmektedir.]	0.76	
A9-[Proje ekibi, gereksinimlere uyum sağlama konusunda hızlı ve esnektir.]	0.754	
A8-[Proje sürecinde gelen talepler sürekli olarak değerlendirmeye alınabilir ve önceliklendirilebilir.]	0.561	
A18-[Proje süreci boyunca dinamik bir takvim ile çalışılır.]	0.493	
O8-[Güçlü iletişim proje ekibi için önceliklidir.]	0.492	
O4-[Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilirim.]	0.672	
A1-[Hiyerarşik yerine işbirlikçi bir kültür ile çalışıyorum.]	0.574	
O3-[Yüzyüze iletişim çalışma ortamında çok önemlidir.]	0.567	
A2-[Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir.]	0.562	
O1-[Yöneticim beni destekler.]	0.547	
A3-[Proje ekibi bir arada çalışır.]	0.461	
A17-[Projede öncelikle en önemli modüller canlıya alınır. (Amaç, mümkün olan en kısa sürede çalışan yazılımı canlıya almaktır.)]	0.752	
A16-[Proje parçalı olarak, planlanan belli zamanlarda canlıya alınır.]	0.629	
O13-[Üzerinde çalışılan proje, hayati öneme sahip olmayan bir projedir.]	0.52	
O11-[Proje için mevcutta kullanılan alt yapı ve teknoloji, tüm isterleri karşılayacak düzeydedir.]	0.495	
A15-[Proje sürecinde optimum seviyede dokümantasyon yapılır. (Dokümana aşırı zaman harcanmaz.)]	0.472	
A11-[Müşterinin (iş birimi, ürün sahibi vb) projeye bağlılığı ve projeye sürekli ilgilenmesi proje ekibi için kritiktir.]	0.594	
A7-[Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır.]	0.572	
A12-[Müşteri yetkinliği, (iş birimi, ürün sahibi vb) projenin her aşamasında çıkan sorunlarla ilgilenmesi ve karar alması için yeterlidir.]	0.551	

6.3. Ölçme Aracının Geçerliliği ve Güvenilirliği

Ölçme aracının test edilmesindeki ana amaç, yapısal modeldeki her bir bileşenin güvenilir ve geçerli bir ölçüm aracıyla değerlendirildiğini sağlamaktır. Ölçüm araçlarının güvenilirliğini ve geçerliliğini doğrulamak için birden fazla yöntem

bulunmaktadır. Özellikle geçerlilik testleri kapsamında, açıklayıcı faktör analizi önemli bir rol oynamaktadır (Nunnally, 1967).

Faktör analizi iki ana yaklaşıma sahiptir: açıklayıcı faktör analizi ve doğrulayıcı faktör analizi. Açıklayıcı faktör analizi, değişkenler arasındaki ilişkilerden yola çıkarak faktörlerin keşfedilmesine odaklanırken; doğrulayıcı faktör analizi, değişkenler arasındaki ilişkileri daha önce belirlenmiş bir hipotez veya teoriyi test ederek inceler. Açıklayıcı faktör analizi, ölçeklerin yapı geçerliliğini değerlendirmede sıklıkla kullanılan bir yöntemdir (Büyüköztürk, 2002). Araştırmacıların, ölçme aracının hangi faktörleri ölçtüğü konusunda bir bilgiye sahip olmadığı ve belirli bir hipotezi test etmek yerine ölçme aracıyla ilgili faktörlerin doğasını anlamaya çalıştığı durumlar açıklayıcı faktör analizi olarak adlandırılmaktadır (Tavşancıl, 2006). Açıklayıcı faktör analizi kullanılması bu çalışma için uygun görülmüştür.

Faktör analizi, dört temel aşamada gerçekleştirilmektedir. Bu aşamalar şunlardır: faktör analizi için veri seti uygunluğu değerlendirilmesi, faktörlerin elde edilmesi, faktörlerin rotasyonu ve faktörlerin isimlendirilmesi (Kalaycı, 2014). Devam eden bölümlerde bu aşamalar ayrıntılı olarak ele alınmıştır.

6.4. Faktör Analizi

Faktör analizinde örneklem sayısı kritik bir öneme sahiptir. Analiz sonucunda ortaya çıkacak faktörlerin güvenilirliği, belirli sayıda katılımcı tarafından yanıtlanmasına bağlıdır. Bu durum, çok değişkenli bir istatistiksel yöntem olan faktör analizinde hem çoklu normallik hem de açıklanan varyans miktarını etkiler. Faktör analizinde, diğer istatistiksel yöntemlerde olduğu gibi, büyük örneklemlerden elde edilen veriler, küçük örneklemelere göre daha az örneklem hatası içerir ve değişkenler arasındaki ilişkilerin daha anlamlı sonuçlar vermesi olasılığı artar (Kline, 1998).

Faktör analizi literatüründe, belirli standartlar olmamakla birlikte, örneklem sayısı ile gözlenen değişken arasında %10'luk bir oranın istenen sonuçlara ulaşmak açısından yeterli olduğu, ancak bu oranın %20'nin altına inmemesi gerektiği genel kabul görmektedir (Hair vd., 1998; Kline, 1998). Bu araştırmada nihai faktör analizine dahil edilen ifadeler göz önünde bulundurulduğunda, 1654 katılımcıdan oluşan örneklem 10:1'lik oranın oldukça üstündedir ve Hair ve diğerlerinin (1998) ile Kline'nin (1998) belirttiği ölçütlere uygundur.

Veri setinin faktör analizine uygunluğunu değerlendirmek için Örneklem Yeterlilik Testi (KMO - Kaiser-Meyer-Olkin) ve değişkenler arasındaki ilişkilerin varlığını belirlemek amacıyla Bartlett Küresellik Testi (Bartlett's Test of Sphericity) kullanılır. Örneklem büyüklüğünün yeterli kabul edilebilmesi için KMO değerinin 0.5'in üzerinde olması gereklidir. Bu değer 1'e yaklaşması, her bir değişkenin diğer değişkenler tarafından hatasız tahmin edilebildiğini gösterir (Hair vd., 2005). KMO değeri 1'e yaklaştıkça mükemmel kabul edilirken, 0.5'in altındaki değerler yetersiz olarak değerlendirilir (Kaiser, 1974; Pett vd., 2003; Kalaycı, 2014).

Tablo 5

KMO ve Bartlett Küresellik Testi

Test	Değer
KMO Örneklem Yeterlilik Değeri	0.896
Bartlett Küresellik Testi	
Yaklaşık Ki-Kare	10462.15
Serbestlik Derecesi (df)	595
Anlamlılık Düzeyi (p)	0

KMO değeri 0.896, örneklem yeterliliğinin oldukça iyi olduğunu gösterir. Genel kabul gören değerlere göre, KMO değeri 0.5'in üzerinde olduğunda yeterli, 0.7 ve üzerinde olduğunda iyi, 0.9'a yakın veya üzerinde olduğunda ise mükemmel olarak değerlendirilir. Bu durumda, 0.896 değeri mükemmel oldukça yakın bir değer olduğu için, veri setinin faktör analizi için çok uygun olduğunu belirtir.

6.5. Hipotez Testleri

Bu bölümde, verilerin çok değişkenli istatistiksel yöntemler kullanılarak değerlendirilmesine ilişkin ayrıntılar ve analiz sonuçları paylaşılmaktadır.

6.5.1. Bağımsız Örneklem T-Testi

T-testi, sosyal bilimlerde yaygın olarak kullanılan testlerden biridir. Araştırma kapsamındaki grupların karşılaştırılması, iki veya daha fazla grup arasında gerçekleştirilebilir. İki grubun karşılaştırılmasında T-testi veya Mann-Whitney U testi kullanılabilir. Bu testler, iki bağımsız grubun ortalamalarının karşılaştırılmasına dayanır (Erdoğan, 2003). Tek veya iki grup arasında fark olup olmadığını belirlemek amacıyla kullanılan bu testler, tek değişkenli hipotez testleridir ve temel varsayımları (Altunışık vd., 2005) aşağıdaki gibi özetlenebilir:

1. Parametrik test koşullarının sağlanması gerekmektedir.
2. Bağımlı değişkene yönelik ölçümlerden elde edilen puanlar, aralık veya oran ölçeğinde olup karşılaştırılacak grup ortalamaları aynı değişkene ait olmalıdır.
3. Bağımlı değişkenle ilgili ölçümlerin her iki grupta da normal dağılım göstermesi gerekir.
4. Karşılaştırılacak ortalama puanlar, ilişkisiz örneklemelerden elde edilmelidir (Büyüköztürk, 2002).

Ön koşulların sağlanmasıyla birlikte, anket cevapları puanlarının proje yönetim modellerine göre farklılık gösterip göstermediği Bağımsız Örneklem T-testi kullanılarak incelenmiştir.

Tablo 6

Proje Yönetim Modeline Göre T-Testi

Sorular	Proje Yönetim Modeli	N	Ortalama	Standart Sapma	Ortalama Hata
O1 - [Yöneticim beni destekler.]	Agile	474	4.37	0.715	0.033
	Waterfall	96	4.5	0.665	0.068
O2 - [Projeye bağlı bir sponsor veya yönetici var.]	Agile	474	4.18	0.824	0.038
	Waterfall	96	4.56	0.558	0.057
A1 - [Hiyerarşik yerine işbirlikçi bir kültür ile çalışıyorum.]	Agile	474	4.06	0.833	0.038
	Waterfall	96	3.88	0.932	0.095
O3 - [Yüzyüze iletişim çalışma ortamımda çok önemlidir.]	Agile	474	3.48	1.136	0.052
	Waterfall	96	3.53	1.066	0.109
A2 - [Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir.]	Agile	474	4.2	0.742	0.034
	Waterfall	96	3.53	1.066	0.109
O4 - [Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilirim.]	Agile	474	3.2	1.114	0.051
	Waterfall	96	3.13	1.059	0.108
A3 - [Proje ekibi bir arada çalışır.]	Agile	474	3.93	0.789	0.036
	Waterfall	96	3.75	1.005	0.103
O5 - [Çalışma ortamımız proje için uygundur.]	Agile	474	4.08	0.743	0.034
	Waterfall	96	4.13	0.653	0.067
O6 - [Proje ekibinin teknik yeterliliği ve uzmanlık seviyesi projeye uygundur.]	Agile	474	3.96	0.83	0.038
	Waterfall	96	3.88	0.824	0.084
A4 - [Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir.]	Agile	474	4.14	0.699	0.032
	Waterfall	96	3.84	0.67	0.068
O7 - [Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir.]	Agile	474	3.97	0.707	0.032
	Waterfall	96	3.94	0.662	0.068
A5 - [Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir.]	Agile	474	4.08	0.738	0.034
	Waterfall	96	3.88	0.743	0.076

Tablolar Devamı

A6 - [Proje ekibi işleri kendi aralarında organize edebilir.]	Agile	474	4.13	0.719	0.033
	Waterfall	96	3.88	0.861	0.088
A7 - [Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır.]	Agile	474	4.03	0.792	0.036
	Waterfall	96	3.81	0.812	0.083
W1 - [Projenin kapsamı ve hedefleri net tanımlanmıştır.]	Agile	474	3.71	0.958	0.044
	Waterfall	96	3.75	0.871	0.089
A8 - [Proje sürecinde gelen talepler sürekli olarak değerlendirilmeye alınabilir ve önceliklendirilebilir.]	Agile	474	4	0.887	0.041
	Waterfall	96	3.75	0.906	0.092
[Proje süreçleri hızlı ve esnek bir şekilde yönetilmektedir.]	Agile	474	3.95	0.871	0.04
	Waterfall	96	3.56	0.792	0.081
A9 - [Proje ekibi, gereksinimlere uyum sağlama konusunda hızlı ve esnektir.]	Agile	474	4.05	0.818	0.038
	Waterfall	96	3.63	0.785	0.08
A10 - [Projede aksiyon alma konusunda hızlı ve esneğiz.]	Agile	474	4.18	0.776	0.036
	Waterfall	96	3.81	0.886	0.09
O8 - [Güçlü iletişim proje ekibi için önceliklidir.]	Agile	474	4.3	0.776	0.036
	Waterfall	96	4.34	0.54	0.055
O9 - [Projede takvime uyma konusunda problem yaşanmaz.]	Agile	474	3.4	1.038	0.048
	Waterfall	96	3.38	0.861	0.088
A11 - [Müşterinin (iş birimi, ürün sahibi vb) projeye bağlılığı ve projeye sürekli ilgilenmesi proje ekibi için kritiktir.]	Agile	474	4.03	0.827	0.038
	Waterfall	96	3.84	0.838	0.085
A12 - [Müşteri yetkinliği, (iş birimi, ürün sahibi vb) projenin her aşamasında çıkan sorunlarla ilgilenmesi ve karar alması için yeterlidir.]	Agile	474	3.6	0.886	0.041
	Waterfall	96	3.59	0.789	0.081
O10 - [Projede, teknik standartlara uyum konusunda problem yaşanmaz.]	Agile	474	3.82	0.842	0.039
	Waterfall	96	3.81	0.73	0.075
A13 - [Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan bir yapı tercih edilmiştir.]	Agile	474	3.65	0.909	0.042
	Waterfall	96	3.59	0.789	0.081
A14 - [Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir.]	Agile	474	3.59	1.014	0.047
	Waterfall	96	3.44	0.792	0.081
A15 - [Proje sürecinde optimum seviyede dokümantasyon yapılır. (Dokümana aşırı zaman harcanmaz.)]	Agile	474	3.4	1.056	0.049
	Waterfall	96	3.06	0.868	0.089
W2 - [Projede birim ve entegrasyon testine önem verilmektedir.]	Agile	474	3.78	0.992	0.046
	Waterfall	96	4.06	0.751	0.077
A16 - [Proje parçalı olarak, planlanan belli zamanlarda canlıya alınır.]	Agile	474	3.97	0.868	0.04
	Waterfall	96	3.5	1.152	0.118
A17 - [Projede öncelikle en önemli modüller canlıya alınır. (Amaç, mümkün olan en kısa sürede çalışan yazılımı canlıya almaktır.)]	Agile	474	3.76	0.965	0.044
	Waterfall	96	3.5	1.124	0.115
O11 - [Proje için mevcutta kullanılan alt yapı ve teknoloji, tüm isterleri karşılayacak düzeydedir.]	Agile	474	3.95	0.871	0.04
	Waterfall	96	3.84	0.838	0.085
O12 - [Proje ekibine gereksinim duyulan konularda eğitim verilir ve teknik bilgi seviyesi artırılır.]	Agile	474	3.73	0.985	0.045
	Waterfall	96	3.75	0.906	0.092

Tablolar Devamı

O13-[Üzerinde çalışılan proje, hayati öneme sahip olmayan bir projedir.]	Agile	474	3.23	1.104	0.051
	Waterfall	96	3.03	1.08	0.11
W3-[Proje kapsamı, proje sürecinin en başında net olarak belirlenir.]	Agile	474	3.43	1.14	0.052
	Waterfall	96	4.03	0.774	0.079
W3-[Proje kapsamı, proje sürecinin en başında net olarak belirlenir.]	Agile	474	3.91	0.871	0.04
	Waterfall	96	3.38	1.088	0.111
W4- [Proje sürecinin en başında maliyet ve kaynak planlaması detaylı olarak yapılır.]	Agile	474	3.66	1.017	0.047
	Waterfall	96	4.09	0.727	0.074
W5- [Risk analizi, proje sürecinin başında yapılır.]	Agile	474	3.55	1.06	0.049
	Waterfall	96	3.81	0.812	0.083

Anlamlı cevaplar aşağıdaki gibi listelenmiştir.

A5 - Proje Ekibi, Proje Yönetim Süreçlerine Hızlı Bir Şekilde Uyum Sağlayabilir:

Agile yöntemi için ortalama değer 4.08, Waterfall yöntemi için 3.88'dir. Agile modelinde değişen koşullara uyum sağlama Waterfall modeline göre biraz daha fazladır.

A7 - Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır:

Agile yöntemi için ortalama değer 4.03, Waterfall yöntemi için 3.81'dir. Agile modelinde müşteri ile ilişkiler Waterfall modeline göre biraz daha fazladır.

A8 - Proje sürecinde gelen talepler sürekli olarak değerlendirilmeye alınabilir ve önceliklendirilebilir:

Agile yöntemi için ortalama değer 4, Waterfall yöntemi için 3.75'tir. Agile modelinde taleplerin değerlendirilmesi ve önceliklendirilmesi Waterfall modeline göre biraz daha fazladır.

A11 - Proje Ekibi, Proje Yönetim Süreçleri ile İlgili Sürekli Olarak Değerlendirme ve Geliştirme Yapabilir:

Agile yöntemi için ortalama değer 4.03, Waterfall yöntemi için 3.84'tür. Agile modelinde müşteri bağlılığı ve ilgisi Waterfall modeline göre biraz daha fazladır

A12 - Müşteri yetkinliği, (iş birimi, ürün sahibi vb) projenin her aşamasında çıkan sorunlarla ilgilenmesi ve karar alması için yeterlidir:

Agile yöntemi için ortalama değer 3.6, Waterfall yöntemi için 3.59'dur. Agile ve Waterfall modelleri arasında müşteri yetkinliği konusunda anlamlı bir fark yoktur.

A13 - Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan bir yapı tercih edilmiştir:

Agile yöntemi için ortalama değer 3.65, Waterfall

yöntemi için 3.59'dur. Agile modelinde tasarım basitliği ve anlaşılabilirliği Waterfall modeline göre biraz daha fazladır.

A14 - Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir: Agile yöntemi için ortalama değer 3.59, Waterfall yöntemi için 3.44'tür. Agile modelinde kodların gözden geçirilmesi ve iyileştirilmesi Waterfall modeline göre biraz daha fazladır.

A17 - Projede öncelikle en önemli modüller canlıya alınır. (Amaç, mümkün olan en kısa sürede çalışan yazılımı canlıya almaktır): Agile yöntemi için ortalama değer 3.76, Waterfall yöntemi için 3.5'dir. Agile modelinde en önemli modüllerin canlıya alınma oranı Waterfall modeline göre daha fazladır.

W5 - Risk analizi, proje sürecinin başında yapılır : Agile yöntemi için ortalama değer 3.55, Waterfall yöntemi için 3.81'dir. Waterfall modelinde risk analizine verilen önem Agile modeline göre daha fazladır.

Altı faktör altında toplanan anket sorularının cevapları, proje yönetim modellerine göre farklılık gösterip göstermediği Bağımsız Örneklem T-testi kullanılarak incelenmiştir.

Tablo 7

Proje Yönetim Modeline Göre Faktörler T-Testi

Faktör	Proje Yönetim Modeli	N	Ortalama	Standart Sapma	Ortalama Hata
Ekip	Agile	474	3.9699	0.57832	0.02656
Ekip	Waterfall	96	3.8477	0.54477	0.0556
Proje	Agile	474	3.6447	0.73903	0.03394
Proje	Waterfall	96	3.8705	0.52686	0.05377
Süreç	Agile	474	4.0633	0.61792	0.02838
Süreç	Waterfall	96	3.7448	0.64616	0.06595
Organizasyonel	Agile	474	3.8734	0.57874	0.0266
Organizasyonel	Waterfall	96	3.7187	0.62047	0.06335
Teknik	Agile	474	3.6479	0.68339	0.0314
Teknik	Waterfall	96	3.8794	0.69816	0.07126
Müşteri İlişkileri	Agile	474	3.8861	0.63797	0.02937
Müşteri İlişkileri	Waterfall	96	3.75	0.66491	0.06786

Agile, ekip, süreç, organizasyonel ve müşteri ilişkileri faktörlerinde yüksek puanlar olarak bu alanlarda olumlu algılanmaktadır. Waterfall ise proje ve teknik faktörlerde daha yüksek puanlar olarak bu alanlarda daha başarılı görünmektedir. Bu sonuçlar, her iki modelin de farklı alanlarda güçlü yönleri sahip olduğunu ve proje yönetim modelinin seçiminde bu faktörlerin dikkate alınması gerektiğini göstermektedir.

6.5.2. Tek Yönlü Varyans Analizi (ANOVA)

Varyans analizi, birden fazla ortalama arasında fark olup olmadığını test etmek için kullanılır. Çalışmanın önceki bölümünde de ifade edildiği üzere, iki ortalama arasındaki anlamlı farkları tespit etmek için T-testi kullanılabilir. Ancak, Bağımsız Örneklem T-testi yalnızca iki grup arasındaki farklılıkların incelenmesi için uygundur ve birden fazla grubun karşılaştırılması gerektiğinde yetersiz kalır. Bu tür durumlarda, daha uygun testler Tek Yönlü ANOVA (One Way Analysis of Variance) veya Kruskal-Wallis testidir (Altunışık vd., 2005).

Tek Yönlü ANOVA, en temel varyans analizi türüdür. Bu analizde iki değişken bulunur: biri kategorik özellik taşıyan bağımsız değişken, diğeri ise metrik özellik taşıyan bağımlı değişkendir. Bağımsız değişkenin içinde iki veya daha fazla grup olabilir. Tek Yönlü ANOVA, bu gruplar arasında bağımlı değişkenin ortalamaları arasında fark olup olmadığını belirlemeyi amaçlar (Kalaycı, 2014).

Tablo 8

Proje Yönetim Modellerinin Karşılaştırılması: ANOVA Analizi

Soru		Kareler Toplamı	sd	Ortalama Kare	F	Sig.
O1-[Yöneticim beni destekler.]	Between Groups	1,554	2	0,777	1,567	0,210
	Within Groups	287,127	579	0,496		
	Total	288,680	581			
O2-[Projeye bağlı bir sponsor veya yönetici var.]	Between Groups	14,882	2	7,441	12,205	0,000
	Within Groups	352,989	579	0,610		
	Total	367,871	581			
A1-[Hiyerarşik yerine işbirlikçi bir kültür ile çalışıyorum.]	Between Groups	2,842	2	1,421	1,975	0,140
	Within Groups	416,601	579	0,720		
	Total	419,443	581			
O3- [Yüzyüze iletişim çalışma ortamımda çok önemlidir.]	Between Groups	6,999	2	3,500	2,812	0,061
	Within Groups	720,485	579	1,244		
	Total	727,485	581			
A2- [Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir.]	Between Groups	35,382	2	17,691	27,785	0,000
	Within Groups	368,659	579	0,637		
	Total	404,041	581			
O4-[Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilirim.]	Between Groups	0,902	2	0,451	0,373	0,689
	Within Groups	699,057	579	1,207		
	Total	699,959	581			
A3-[Proje ekibi bir arada çalışır.]	Between Groups	2,715	2	1,357	1,981	0,139
	Within Groups	396,703	579	0,685		
	Total	399,418	581			

Tablolar Devamı						
O5-[Çalışma ortamımız proje için uygundur.]	Between Groups	0,275	2	0,138	0,259	0,772
	Within Groups	307,766	579	0,532		
	Total	308,041	581			
O6-[Proje ekibinin teknik yeterliliği ve uzmanlık seviyesi projeye uygundur.]	Between Groups	1,634	2	0,817	1,204	0,301
	Within Groups	392,820	579	0,678		
	Total	394,454	581			
A4-[Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir.]	Between Groups	7,065	2	3,532	7,318	0,001
	Within Groups	279,466	579	0,483		
	Total	286,531	581			
O7-[Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir.]	Between Groups	6,053	2	3,027	6,119	0,002
	Within Groups	286,400	579	0,495		
	Total	292,454	581			
A5-[Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir.]	Between Groups	4,469	2	2,235	4,062	0,018
	Within Groups	318,541	579	0,550		
	Total	323,010	581			
A6- [Proje ekibi işleri kendi aralarında organize edebilir.]	Between Groups	5,136	2	2,568	4,634	0,010
	Within Groups	320,905	579	0,554		
	Total	326,041	581			
A7- [Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır.]	Between Groups	4,600	2	2,300	3,685	0,026
	Within Groups	361,400	579	0,624		
	Total	366,000	581			
W1-[Projenin kapsamı ve hedefleri net tanımlanmıştır.]	Between Groups	3,489	2	1,745	1,988	0,138
	Within Groups	508,073	579	0,878		
	Total	511,562	581			
A8-[Proje sürecinde gelen talepler sürekli olarak değerlendirilmeye alınabilir.]	Between Groups	5,992	2	2,996	3,836	0,022
	Within Groups	452,250	579	0,781		
	Total	458,242	581			
[Proje süreçleri hızlı ve esnek bir şekilde yönetilmektedir.]	Between Groups	12,160	2	6,080	8,368	0,000
	Within Groups	420,660	579	0,727		
	Total	432,820	581			
A9- [Proje ekibi, gereksinimlere uyum sağlama konusunda hızlı ve esnektir.]	Between Groups	15,326	2	7,663	11,752	0,000
	Within Groups	377,535	579	0,652		
	Total	392,861	581			
A10-[Projede aksiyon alma konusunda hızlı ve esneğiz.]	Between Groups	10,831	2	5,415	8,662	0,000
	Within Groups	361,989	579	0,625		
	Total	372,820	581			
O8-[Güçlü iletişim proje ekibi için önceliklidir.]	Between Groups	0,617	2	0,308	0,565	0,568
	Within Groups	315,713	579	0,545		
	Total	316,330	581			
O9-[Projede takvime uyma konusunda problem yaşanmaz.]	Between Groups	4,930	2	2,465	2,451	0,087
	Within Groups					

Tablolar Devamı						
O9-[Projede takvime uyma konusunda problem yaşanmaz.]	Between Groups	4,930	2	2,465	2,451	0,087
	Within Groups	582,389	579	1,006		
	Total	587,320	581			
A11- [Müşterinin (iş birimi, ürün sahibi vb) projeye bağlılığı ve projeye ilgilenmesi]	Between Groups	3,398	2	1,699	2,468	0,086
	Within Groups	398,602	579	0,688		
	Total	402,000	581			
A12- [Müşteri yetkinliği, projenin her aşamasında çıkan sorunlarla ilgilenmesi yeterli.]	Between Groups	4,970	2	2,485	3,277	0,038
	Within Groups	439,045	579	0,758		
	Total	444,015	581			
O10- [Projede, teknik standartlara uyum konusunda problem yaşanmaz.]	Between Groups	0,052	2	0,026	0,039	0,962
	Within Groups	387,907	579	0,670		
	Total	387,959	581			
A13- [Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan yapılar.]	Between Groups	4,633	2	2,316	2,968	0,052
	Within Groups	451,862	579	0,780		
	Total	456,495	581			
A14-[Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir.]	Between Groups	2,034	2	1,017	1,061	0,347
	Within Groups	554,853	579	0,958		
	Total	556,887	581			
A15-[Proje sürecinde optimum seviyede dokümantasyon yapılır.]	Between Groups	9,125	2	4,562	4,348	0,013
	Within Groups	607,514	579	1,049		
	Total	616,639	581			
W2-[Projede birim ve entegrasyon testine önem verilmektedir.]	Between Groups	10,343	2	5,172	5,549	0,004
	Within Groups	539,616	579	0,932		
	Total	549,959	581			
A16-[Proje parçalı olarak, planlanan belli zamanlarda canlıya alınır.]	Between Groups	17,655	2	8,828	10,462	0,000
	Within Groups	488,525	579	0,844		
	Total	506,180	581			
A17- [Projede öncelikle en önemli modüller canlıya alınır.]	Between Groups	6,325	2	3,162	3,198	0,042
	Within Groups	572,582	579	0,989		
	Total	578,907	581			
O11-[Proje için mevcutta kullanılan alt yapı ve teknoloji, tüm isterleri karşılayacak.]	Between Groups	1,278	2	0,639	0,853	0,427
	Within Groups	433,691	579	0,749		
	Total	434,969	581			
O12-[Proje ekibine gereksinim duyulan konularda eğitim verilir.]	Between Groups	0,834	2	0,417	0,445	0,641
	Within Groups	542,506	579	0,937		
	Total	543,340	581			
O13-[Üzerinde çalışılan proje, hayati öneme sahip olmayan bir projedir.]	Between Groups	3,317	2	1,658	1,368	0,256
	Within Groups	702,163	579	1,213		
	Total	705,479	581			
W3-[Proje kapsamı, proje sürecinin en başında net olarak belirlenir.]	Between Groups	34,889	2	17,444	15,000	0,000
	Within Groups	673,359	579	1,163		
	Total					

Tablolar Devamı						
	Total	708,247	581			
A18-[Proje süreci boyunca dinamik bir takvim ile çalışılır.]	Between Groups	22,829	2	11,414	14,025	0,000
	Within Groups	471,228	579	0,814		
	Total	494,057	581			
W4- [Proje sürecinin en başında maliyet ve kaynak planlaması yapılır.]	Between Groups	17,800	2	8,900	9,402	0,000
	Within Groups	548,071	579	0,947		
	Total	565,871	581			
W5- [Risk analizi, proje sürecinin başında yapılır.]	Between Groups	5,758	2	2,879	2,741	0,065
	Within Groups	608,160	579	1,050		
	Total	613,918	581			

Bu ANOVA tablosu, belirli faktörlerde proje yönetim modelleri arasında anlamlı farklar olup olmadığını ortaya koymaktadır. Özellikle "projeyle bağlılık", "süreçlere hakimiyet" ve "değişen koşullara uyum sağlama" gibi faktörlerde anlamlı farklar bulunmuştur. Bu bulgular, proje yönetim modellerinin belirli açılardan farklı etkiler yaratabileceğini göstermektedir.

ANOVA tablosu, gruplar arasında istatistiksel olarak anlamlı bir fark olup olmadığını belirlemek için kullanılır, ancak bu farkın hangi gruplar arasında olduğunu göstermez. Bu nedenle post hoc analizlerine ihtiyaç duyulur. Post hoc testleri, hangi spesifik gruplar arasında fark olduğunu belirler ve çoklu karşılaştırmaların getirdiği hataları (Type I error) kontrol ederek daha güvenilir sonuçlar sağlar. Örneğin, Agile, Waterfall ve Hybrid yöntemleri arasındaki farkları belirlemek için kullanılır. Bu testler, araştırmacıların daha derinlemesine ve spesifik yorumlar yapmalarına olanak tanır, böylece sonuçların doğruluğu ve güvenilirliği artırılır. Bu nedenle, ANOVA'dan sonra post hoc analizleri yapmak, güvenilir ve ayrıntılı sonuçlar elde etmek için gereklidir.

Tablo 9

Proje Yönetim Metodlarının Post Hoc Karşılaştırma Sonuçları

Dependent Variable		(I) Proje Yönetim Modeli	(J) Proje Yönetim Modeli	(I-J)	Standart Sapma	Sig.	95% Confidence Interval	
							Lower Bound	Upper Bound
O1-[Yöneticim beni destekler.]	Scheffe	Agile	Waterfall	-0.13	0.079	0.242	-0.33	0.06
O2-[Projeye bağlı bir sponsor veya yönetici var.]	Scheffe	Agile	Waterfall	-0.39	0.087	0	-0.6	-0.17
A1-[Hiyerarşik yerine işbirlikçi bir kültür ile çalışıyorum.]	Scheffe	Agile	Waterfall	0.19	0.095	0.141	-0.04	0.42
O3- [Yüzyüze iletişim çalışma ortamımda çok önemlidir.]	Scheffe	Agile	Waterfall	-0.05	0.125	0.922	-0.36	0.26
A2- [Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir.]	Scheffe	Agile	Waterfall	0.66	0.089	0	0.45	0.88

Tablolar Devamı									
O4-[Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilir.]	Scheffe	Agile	Waterfall	0.08	0.123	0.82	-0.22	0.38	
A3-[Proje ekibi bir arada çalışır.]	Scheffe	Agile	Waterfall	0.18	0.093	0.151	-0.05	0.41	
O5-[Çalışma ortamımız proje için uygundur.]	Scheffe	Agile	Waterfall	-0.05	0.082	0.835	-0.25	0.15	
O6-[Proje ekibinin teknik yeterliliği ve uzmanlık seviyesi projeye uygundur.]	Scheffe	Agile	Waterfall	0.08	0.092	0.682	-0.15	0.31	
A4-[Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir.]	Scheffe	Agile	Waterfall	0.3	0.078	0.001	0.1	0.49	
O7-[Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir.]	Scheffe	Agile	Waterfall	0.03	0.079	0.926	-0.16	0.22	
A5-[Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir.]	Scheffe	Agile	Waterfall	0.21	0.083	0.045	0	0.41	
A6-[Proje ekibi işleri kendi aralarında organize edebilir.]	Scheffe	Agile	Waterfall	0.25	0.083	0.011	0.05	0.46	
A7-[Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır.]	Scheffe	Agile	Waterfall	0.22	0.088	0.047	0	0.44	
W1-[Projenin kapsamı ve hedefleri net tanımlanmıştır.]	Scheffe	Agile	Waterfall	-0.04	0.105	0.926	-0.3	0.22	
A8-[Proje sürecinde gelen talepler sürekli olarak değerlendirmeye alınabilir ve önceliklendirilebilir.]	Scheffe	Agile	Waterfall	0.25	0.099	0.042	0.01	0.49	
[Proje süreçleri hızlı ve esnek bir şekilde yönetilmektedir.]	Scheffe	Agile	Waterfall	0.39	0.095	0	0.15	0.62	
A9-[Proje ekibi, gereksinimlere uyum sağlama konusunda hızlı ve esnektir.]	Scheffe	Agile	Waterfall	0.43	0.09	0	0.2	0.65	
A10-[Projede aksiyon alma konusunda hızlı ve esneğiz.]	Scheffe	Agile	Waterfall	0.36	0.088	0	0.15	0.58	
O8-[Güçlü iletişim proje ekibi için önceliklidir.]	Scheffe	Agile	Waterfall	-0.05	0.083	0.855	-0.25	0.16	
O9-[Projede takvime uyma konusunda problem yaşanmaz.]	Scheffe	Agile	Waterfall	0.02	0.112	0.978	-0.25	0.3	
A11-[Müşterinin (iş birimi, ürün sahibi vb) projeye bağlılığı ve projeye sürekli ilgilenmesi proje ekibi için kritiktir.]	Scheffe	Agile	Waterfall	0.18	0.093	0.149	-0.05	0.41	
A12-[Müşteri yetkinliği, (iş birimi, ürün sahibi vb) projenin her aşamasında çıkan sorunlarla ilgilenmesi ve karar alması için yeterlidir.]	Scheffe	Agile	Waterfall	0.01	0.097	0.997	-0.23	0.25	
O10-[Projede, teknik standartlara uyum konusunda problem yaşanmaz.]	Scheffe	Agile	Waterfall	0	0.092	0.999	-0.22	0.23	
A13-[Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan bir yapı tercih edilmiştir.]	Scheffe	Agile	Waterfall	0.05	0.099	0.872	-0.19	0.29	
A14-[Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir.]	Scheffe	Agile	Waterfall	0.16	0.11	0.357	-0.11	0.43	
A15-[Proje sürecinde optimum seviyede dokümantasyon yapılır. (Dokümana aşırı zaman harcanmaz.)]	Scheffe	Agile	Waterfall	0.34	0.115	0.014	0.05	0.62	
W2-[Projede birim ve entegrasyon testine önem verilmektedir.]	Scheffe	Agile	Waterfall	-0.28	0.108	0.032	-0.55	-0.02	
A16-[Proje parçalı olarak, planlanan belli zamanlarda canlıya alınır.]	Scheffe	Agile	Waterfall	0.47	0.103	0	0.22	0.72	
A17-[Projede öncelikle en önemli modüller canlıya alınır. (Amaç, mümkün olan en kısa sürede çalışan yazılımı canlıya almaktır.)]	Scheffe	Agile	Waterfall	0.26	0.111	0.067	-0.01	0.53	
O11-[Proje için mevcutta kullanılan alt yapı ve teknoloji, tüm isterleri karşılayacak düzeydedir.]	Scheffe	Agile	Waterfall	0.11	0.097	0.552	-0.13	0.34	
O12-[Proje ekibine gereksinim duyulan konularda eğitim verilir ve teknik bilgi seviyesi artırılır.]	Scheffe	Agile	Waterfall	-0.02	0.108	0.989	-0.28	0.25	
O13-[Üzerinde çalışılan proje, hayati öneme sahip olmayan bir projedir.]	Scheffe	Agile	Waterfall	0.2	0.123	0.259	-0.1	0.51	
W3-[Proje kapsamı, proje sürecinin en başında net olarak belirlenir.]	Scheffe	Agile	Waterfall	-0.6	0.121	0	-0.9	-0.3	
A18-[Proje süreci boyunca dinamik bir takvim ile çalışılır.]	Scheffe	Agile	Waterfall	0.53	0.101	0	0.28	0.78	
W4-[Proje sürecinin en başında maliyet ve kaynak planlaması detaylı olarak yapılır.]	Scheffe	Agile	Waterfall	-0.43	0.109	0	-0.7	-0.16	

Tablolar Devamı									
W5- [Risk analizi, proje sürecinin başında yapılır.]	Scheffe	Agile	Waterfall	-0.26	0.115	0.075	-0.54	0.02	

Bu tablo, Agile ve Waterfall proje yönetim modelleri arasındaki çeşitli bağımlı değişkenler için post hoc test sonuçlarını göstermektedir. Scheffe ve Tamhane testlerine göre gruplar arasındaki ortalama farkları, standart hataları, anlamlılık değerlerini ve güven aralıklarını içermektedir. Tabloyu yorumlayarak önemli bulgulara dikkat çekelim:

1. O2 - Projeye bağlı bir sponsor veya yönetici var: Agile ve Waterfall arasında anlamlı bir fark gözlenmiştir (Scheffe testi, $p < .05$). Bu, proje sponsorluğunun Agile ve Waterfall yöntemlerinde farklı şekilde algılandığını göstermektedir.
2. A2 - Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir: Agile ve Waterfall arasında anlamlı bir fark bulunmaktadır (Scheffe testi, $p < .05$). Çevik yaklaşımın kabulü Agile yönteminde daha yüksek olabilir.
3. O4 - Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilirim: Bu değişkende anlamlı bir fark bulunmamaktadır (Scheffe testi, $p > .05$). Hedeflere ulaşmada ödüllendirme konusunda iki yöntem arasında fark yoktur.
4. A4 - Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir: Agile ve Waterfall arasında anlamlı bir fark vardır (Scheffe testi, $p < .05$). Bu, Agile yönteminin ekip bağlılığı ve iş birliğinde daha etkili olduğunu gösterebilir.
5. O7 - Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir: Agile ve Waterfall arasında anlamlı bir fark gözlenmiştir (Scheffe testi, $p < .05$). Proje ekipleri Agile yönteminde süreçlere daha hâkim olabilir.
6. A5 - Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir: Agile ve Waterfall arasında anlamlı bir fark bulunmaktadır (Scheffe testi, $p < .05$). Agile yönteminin esnekliği ve uyum sağlama kabiliyeti daha yüksektir.
7. A6 - Proje ekibi işleri kendi aralarında organize edebilir: Agile ve Waterfall arasında anlamlı bir fark vardır (Scheffe testi, $p < .05$). Agile yönteminde ekiplerin kendi işlerini organize etme yeteneği daha yüksektir.
8. A7 - Proje ekibinin, müşteri ile iyi bir ilişkisi vardır: Agile ve Waterfall arasında anlamlı bir fark gözlenmiştir (Scheffe testi, $p < .05$). Agile yönteminde müşteri ilişkileri daha iyi olabilir.

9. W1 - Projenin kapsamı ve hedefleri net tanımlanmıştır: Bu değişkende anlamlı bir fark bulunmamaktadır (Scheffe testi, $p>.05$). Kapsam ve hedeflerin netliği konusunda iki yöntem arasında fark yoktur.
10. A8 - Proje sürecinde gelen talepler sürekli olarak değerlendirmeye alınabilir ve önceliklendirilebilir: Agile ve Waterfall arasında anlamlı bir fark vardır (Scheffe testi, $p<.05$). Agile yönteminde taleplerin değerlendirilmesi ve önceliklendirilmesi daha etkili olabilir.

Tablo, Agile ve Waterfall yöntemleri arasında belirli değişkenlerde anlamlı farklar olduğunu göstermektedir. Agile yöntemi çeviklik, ekip işbirliği ve müşteri ilişkilerinde daha avantajlıdır. Bu bulgular, proje yönetim yöntemlerinin farklılaştığı ve üstünlük sağladığı alanları anlamak için kullanılabilir.

Proje yönetimi ve deneyim arasında değerlendirdiğimizde ANOVA tablosu aşağıdaki gibi oluşmuştur.

Tablo 10

ANOVA Tablosu: Proje Yönetim Modelleri ve Deneyim Faktörlerinin Etkisi

Bağımlı Değişken	Kaynak	Kareler Toplamı	sd	Ortalama Kare	F	Anlamlılık
Proje Yönetim Modeli	Gruplar Arası	1.691	2	0.846	6.14	0.002
	Gruplar İçi	78.14	567	0.138		
	Toplam	79.832	569			
Deneyim	Gruplar Arası	3.87	2	1.935	1.6	0.204
	Gruplar İçi	687.962	567	1.213		
	Toplam	691.832	569			

Tabloda, faktör puanlarının proje yönetim modeli ve deneyim düzeyine göre farklılık gösterip göstermediğinin sınanması neticesinde elde edilen sonuçlar yer almaktadır. Buna göre ANOVA anlam değerinin (p) kabul edilen .05’lik anlamlılık değerinden küçük olması durumunda ilgili faktör puanlarında gruplar arası farklılık olduğu, $p>.05$ olduğu durumda ise gruplar arası fark bulunmadığı anlaşılmaktadır. Buna göre, proje yönetim modeli faktör puanları arasında anlamlı farklar bulunmaktadır, ancak deneyim düzeyine göre anlamlı bir farklılık bulunmamaktadır. Başka bir ifade ile proje yönetim modeli ifadelerine katılım düzeyi bazında değişkenlik göstermektedir. Farklılığın hangi gruplardan kaynaklandığının anlaşılması için Post Hoc sonuçlarından faydalanılmıştır.

Anlamlılık değeri .05’ten büyük olan tek faktör olarak, deneyim düzeyine göre farklılık gözlenmemektedir. Post Hoc değerlerinin yorumlanabilmesi için grup varyanslarının

eşitliği test edilmelidir. Bu eşitliğin test edilmesi için Levene Varyansların Homojenliği Testinden faydalanılmıştır (Coakers, 2005; Landau ve Everitt, 2004). Levene testi sonucunda elde edilen p değerleri .05'ten küçük olduğu için %5 anlamlılık düzeyinde varyansların farklı olduğu anlaşılmaktadır. Varyansların eşit olmaması nedeniyle gruplar arasındaki ikili karşılaştırmalarda Tamhane testi değerleri yorumlanmıştır.

Tablo 11

Proje Yönetim Modelleri ve Deneyim Düzeylerinin Karşılaştırılması: Scheffe ve Tamhane Test Sonuçları

Bağımlı Değişken		(I) Çalışma Şekli	(J) Çalışma Şekli	(I-J)	Standart Sapma	Sig.	Alt Sınır	Üst Sınır
Proje Yönetim Modeli	Scheffe	Agile	Waterfall	-.11	0.032	0	-.19	-.03
		Waterfall	Agile	.101	0.061	.27	-.05	.25
	Tamhane	Agile	Waterfall	-.11	0.062	.06	-.23	.03
		Waterfall	Agile	.101	0.063	.11	-.02	.22
Deneyim	Scheffe	Agile	Waterfall	-.1	0.063	.3	-.22	.02
		Waterfall	Agile	.1	0.063	.3	-.02	.22
	Tamhane	Agile	Waterfall	-.1	0.063	.3	-.22	.02
		Waterfall	Agile	.1	0.063	.3	-.02	.22

Bu tablo, Agile ve Waterfall proje yönetim modelleri ile deneyim düzeyleri arasında yapılan ikili karşılaştırmaların sonuçlarını göstermektedir. Scheffe testi sonuçlarına göre, Agile ve Waterfall arasında anlamlı bir fark bulunmaktadır; Agile modeli, Waterfall modeline göre daha olumlu algılanmaktadır ($p < .05$). Ancak, Tamhane testi sonuçlarına göre bu fark anlamlı değildir. Deneyim düzeylerine göre yapılan karşılaştırmalarda ise, her iki testin sonuçları da Agile ve Waterfall arasında anlamlı bir fark olmadığını göstermektedir ($p > .05$). Bu bulgular, proje yönetim modellerinin algısının deneyim düzeyine göre değişmediğini, ancak Agile modelinin genel olarak daha olumlu değerlendirildiğini göstermektedir.

SONUÇ VE ÖNERİLER

Sonuç

Bu yüksek lisans tezi, Agile ve Waterfall proje yönetim yöntemlerinin IT sektöründeki uygulamalarını ve bu yöntemlerin gerçekte ne kadar etkin kullanıldığını incelemeyi amaçlamaktadır. COVID-19 pandemisi ile hayatımıza giren uzaktan çalışma şekli, firmaların çalışanlarını daha iyi takip etmek amacıyla Agile yöntemlere doğru geçiş yapmasına sebep olmuştur. Ancak, Agile yöntemlerin tüm ritüelleri tam olarak uygulanmamakta ve Agile çalışanlar aslında farkında olmadan Waterfall yöntemini uygulamaktadır. Bu çalışma, bu düşüncüyü destekleyen bulguları ortaya koymaktadır.

Çalışmamızdaki bulgular, proje yönetiminde kullanılan metodolojilerin etkinliğini iyi anlamak adına belirli faktörler üzerinden ele alınmıştır. Bu faktörler: ekip, proje, süreç, organizasyonel, teknik ve müşteri ilişkileri olarak sınıflandırılmıştır. Bu bağlamda, her bir faktörün Agile ve Waterfall yöntemleri üzerindeki etkileri incelenmiş ve sonuçlar aşağıda sunulmuştur.

Ekip:

Araştırmamızda Agile ve Waterfall yöntemlerinin ekip işbirliği ve bağlılık üzerindeki etkileri incelenmiştir. T-testi sonuçlarına göre, Agile yönteminde ekip işbirliği ve bağlılık düzeyi ortalama 3.96, Waterfall yönteminde ise 3.84 olarak ölçülmüştür. Agile yönteminde ekip işbirliği ve bağlılık düzeylerinin yüksek olması beklenirken, her iki yöntemi kullanan ekiplerin işbirliği ve bağlılık düzeylerinin benzer olması dikkat çekicidir. Bu durum, Agile yönteminin tam anlamıyla uygulanmadığını ve ekiplerin aslında Waterfall yöntemine benzer bir şekilde çalıştığını düşündürülebilir.

Proje:

Proje planlaması ve yönetimi açısından, Waterfall yönteminin proje planlaması ve kapsam belirleme konularında daha güçlü olduğu bilinmektedir. T-testi sonuçlarına göre, Waterfall yönteminde proje planlaması ve kapsam belirleme konularında ortalama puan 3.87, Agile yönteminde ise 3.64 olarak bulunmuştur. Waterfall yöntemi bu faktörde Agile'a göre biraz daha yüksek ortalama puanına sahip olarak bunu desteklemektedir.

Süreç:

Süreç yönetimi ve esneklik açısından Agile yöntemi, çeviklik ve esneklik üzerine kurulu olmasına rağmen, Waterfall yöntemi kullanan ekiplerin de bu alanlarda yüksek puanlar verdikleri gözlemlenmiştir. T-testi sonuçlarına göre, Agile yöntemi için çeviklik ve esneklik ortalama puanı 4.06 iken, Waterfall yöntemi için bu puan 3.74 olarak bulunmuştur. Bu, Waterfall metodolojisini kullandığını iddia edenlerin de çevik ve esnek olma eğiliminde olduklarını göstermektedir.

Organizasyonel:

Teknik yeterlilik ve uzmanlık düzeyleri açısından Agile ve Waterfall yöntemlerini kullanan ekipler arasında anlamlı bir fark bulunmamıştır. T-testi sonuçlarına göre, Agile yönteminde teknik yeterlilik ve uzmanlık düzeyi ortalama 3.87, Waterfall yönteminde ise 3.71 olarak ölçülmüştür. Her iki yöntemi kullanan ekiplerin de teknik bilgi seviyesi benzer çıkmıştır, bu da metodolojilerin uygulamada teknik yeterlilik açısından benzer sonuçlar verdiğini göstermektedir.

Teknik:

Araştırma, Agile ve Waterfall yöntemlerinin teknik süreçlere etkilerini de incelemiştir. Teknik yeterlilik açısından Agile teknik ortalama puanı 3.64 ve Waterfall teknik ortalama puanı 3.87 olarak bulunmuştur. Dolayısıyla Waterfall yöntemi bu faktörde Agile'a göre biraz daha yüksek ortalama puanına sahip olarak yöntemler arasında teknik yeterlik açısından ayrışma söz konusudur.

Müşteri İlişkileri:

Müşteri ilişkileri açısından Agile yönteminin müşteri ile daha sık ve yoğun iletişimde olması beklenirken, Waterfall yöntemi kullananların da benzer şekilde müşteri ilişkilerini önemsedikleri görülmüştür. T-testi sonuçlarına göre, Agile yönteminde müşteri ile ilişkiler ortalama 3.88, Waterfall yönteminde ise 3.75 olarak ölçülmüştür. Bu bulgu, her iki yöntemi kullanan ekiplerin de müşteri ilişkilerinde benzer uygulamalara sahip olduğunu göstermektedir.

Öneriler

Araştırmamızın sonuçları, Agile ve Waterfall metodolojilerinin tam anlamıyla uygulanmadığını ve pratikte birbiriyle iç içe geçtiğini göstermektedir. Bu bağlamda, Agile ve Waterfall yöntemlerinin avantajlarını bir araya getiren yeni bir proje yönetim

modelinin geliştirilmesi gerektiği düşünülmektedir. Bu model, uzaktan çalışma koşullarında etkinliğini kanıtlamış, esneklik ve planlama unsurlarını dengeleyen bir yaklaşım olmalıdır.

Yeni bir proje yönetim modeli, Agile'ın esnekliğini ve müşteri odaklı yaklaşımını, Waterfall'ın detaylı planlama ve kapsam belirleme özellikleri ile birleştirmelidir. Birleştirilen özelliklerin sınırları çizilmeli ve kuralları net olarak ortaya konmalıdır. Yeni model en az Agile kadar şeffaf ve takip edilebilir olmalıdır.

Bazı kaynaklarda Hibrit yaklaşım ifadesi geçmektedir ancak bu yaklaşımın belli bir sınırı ve kuralı bulunmamaktadır.

Geliştirilecek yeni modelde, proje süreçleri iteratif ve inkremental olarak planlanmalı, ancak her iterasyon sonunda belirgin planlama ve değerlendirme adımları bulunmalıdır. Böylece, proje ekipleri hem esnek çalışabilir hem de detaylı planlama yaparak projeyi daha kontrollü bir şekilde yürütebilir.

Yeni modelin başarısı için şu adımlar izlenmelidir:

- 1. Eğitim ve Farkındalık:** Ekiplerin yeni metodolojiyi tam anlamıyla benimsemeleri için düzenli eğitimler verilmelidir.
- 2. Araç ve Teknoloji Kullanımı:** Metodolojinin gerektirdiği araçlar etkin bir şekilde kullanılmalı ve ekiplerin bu araçları nasıl kullanacakları konusunda yeterli bilgiye sahip olmaları sağlanmalıdır.
- 3. Sürekli Değerlendirme:** Proje süreçleri düzenli olarak gözden geçirilmeli ve metodolojinin ne kadar etkin olduğu sürekli olarak değerlendirilmelidir.

Bu yeni model, proje yönetiminde daha etkin ve verimli sonuçlar elde edilmesine yardımcı olabilir. Böylece, uzaktan çalışma döneminde ortaya çıkan zorluklar daha etkili bir şekilde yönetilebilir ve projelerin başarısı artırılabilir.

KAYNAKÇA

- Ahmed, A. (2011). *Software project management: A process-driven approach*. CRC Press.
- The benefits of working from home*. (2024, 31 Ocak). Airtasker. <https://www.airtasker.com/blog/the-benefits-of-working-from-home/>
- Alexander, A., De Smet, A. ve Mysore, M. (2020). Reimagining the postpandemic workforce. *McKinsey Quarterly*, 3,1-9.
- Altunışık, R., Coşkun, R., Bayraktaroğlu, S. ve Yıldırım, E. (2007). *Sosyal bilimlerde araştırma yöntemleri: SPSS uygulamalı*. Sakarya yayıncılık.
- Awad, M. A. (2005). A comparison between agile and traditional software development methodologies. *University of Western Australia*, 30, 1-69.
- Badiale, M. E. (2020). *The dynamics of communication in global virtual software development teams: a case study in the agile context during the Covid-19 pandemic*. (Yayınlanmamış yüksek lisans tezi) . Uppsala Üniversitesi.
- Balaji, S., ve Murugaiyan, M. S. (2012). Waterfall vs. V-Model vs. Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), 26-30.
- Baycık, G., Yangın, D. D., Yay, O. ve Doğan, S. (2021). COVID 19 pandemisinde uzaktan çalışma: Tespit ve öneriler. *Çalışma ve Toplum*, 3(70), 1683-1728.
- Bilginoğlu, E. (2021). COVID-19 pandemisi sırasında uzaktan çalışmanın artan önemi: Bilinen yanlışlar ve doğruları. *Çalışma ve Toplum*, 2(69), 1099-1146.
- Butler Jr, A. G. (1973). Project management: a study in organizational conflict. *Academy of Management Journal*, 16(1), 84-101. <https://doi.org/10.5465/255045>
- Büyüköztürk, Ş. (2002). *Sosyal bilimler için veri analizi el kitabı*. Ankara: Pegem Yayıncılık.
- Cleland, D. I. ve Gareis, R. (2006), *Global project management handbook: Planning, organizing, and controlling international project* (12. Baskı). McGraw-Hill.
- Comella, S., Garg, L., Thareja, S. ve McCall, B. (2020). Revisiting agile teams after an abrupt shift to remote. *McKinsey & Company*.
- Çabuk, Z. (2023, 26 Ekim). *Yazılım geliştirme yaşam döngüsü (SDLC)*, Medium. <https://medium.com/@zelihacabuk> adresinden 01 Mart 2024 tarihinde alınmıştır.
- Erdoğan, İ. (2003). *Pozitivist metodoloji: Bilimsel araştırma tasarımı, istatistiksel yöntemler, analiz ve yorum*. Erk Yayınları.

- Demirag, A., Öztürk, E. N. D., & Ünal, C. (2023). Analysis and Comparison of Waterfall Model and Agile Approach in Software Projects. *AJIT-e: Academic Journal of Information Technology*, 14(54), 183-203.
- Farrell, G. ve Shafika, I. (2007). *Survey of ICT and education in Africa: A summary report, based on 53 country surveys*. World Bank. <https://www.sciepub.com/reference/198976> adresinden 01 Ocak 2024 tarihinde alınmıştır.
- Floara, H. K. ve Chande, S. V. (2014). A systematic study on agile software development methodologies and practices. *International Journal of Computer Science and Information Technologies*, 5(3), 3548-3552.
- Gascoigne, C. (2020, 12 Nisan). *Flexible working: Lessons from the pandemic*. CIPD https://www.cipd.co.uk/Images/8051-working-from-home-report_tcm18-84208.pdf adresinden 9 Kasım 2023 tarihinde alınmıştır.
- Gechman, M. (2019). *Project management of large software-intensive systems controlling the software development process*. CRC Press.
- Gencer, C. ve Kayacan, A. (2017). Yazılım proje yönetimi: Şelale modeli ve çevik yöntemlerin karşılaştırılması. *Bilişim Teknolojileri Dergisi*, 10(3), 335-352.
- Golden, T. D. ve Gajendran, R. S. (2019). Unpacking the role of a telecommuter's job in their performance: Examining job complexity, problem solving, interdependence, and social support. *Journal of Business and Psychology*, 34(1), 55-69.
- Gosselin, S., (2021, 10 Mayıs). *The post-covid future of it remote work*. CIO Insight <https://www.cioinsight.com/blogs/the-post-covid-future-of-it-remote-work/> adresinden 01 Şubat 2024 tarihinde alınmıştır.
- Hair, J. F., Anderson, R. E., Tatham, R. ve Black, W. C. (1998). *Multivariate data analysis with readings*. Macmillan.
- Heldman, K. (2018). *PMP project management professional exam study guide*. Sybex.
- Horine, G. (2009). *Absolute beginner's guide to project management*. Que Publishing.
- IEEE Computer Society. (2014). *SWEBOK: Guide to the software engineering body of knowledge*. IEEE.
- Jadoul, Q., Nascimento, A., Salo, O. ve Willi, R. (2020). Agility in the time of COVID-19: Changing your operating model in an age of turbulence. *McKinsey & Company*. <https://www.mckinsey.com/business-functions/people-and-organizational-performance/our-insights/agility-in-the-time-of-covid-19-changing-your-operating-model-in-an-age-of-turbulence> adresinden 01 Ocak 2024 tarihinde alınmıştır.
- Janssen, M. ve Van der Voort, H. (2020). Agile and adaptive governance in crisis response: Lessons from the COVID-19 pandemic. *International journal of information management*, 55, 102180.

- Kaiser, H. F. (1974). An index of factorial simplicity. *psychometrika*, 39(1), 31-36.
- Kalaycı, Ş. (2014). *SPSS uygulamalı çok değişkenli istatistik teknikleri* (6. Baskı). Asil Yayın Dağıtım.
- Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling* (12. Baskı). Wiley.
- Kettunen, P. ve Laanti, M. (2008). Combining agile software projects and large-scale organizational agility. *Software Process: Improvement and Practice*, 13(2), 183-193.
- Kline, R. B. (1998). *Principles and practice of structural equation modeling*. Guilford Press.
- Koch, J. ve Schermuly, C. C. (2021). Managing the crisis: How COVID-19 demands interact with agile project management in predicting employee exhaustion. *British Journal of Management*, 32, 1265-1283. <https://doi.org/10.1111/1467-8551.12536>
- Lock, D. (2007). *Project management* (9. Baskı). Gower Publishing.
- Munassar, N. M. A. ve Govardhan, A. (2010). A comparison between five models of software engineering. *International Journal of Computer Science Issues*, 7(5), 94-101.
- Nandwani, G. (2020). How to maintain team agility in COVID-19 era. *International Journal of Trend in Research and Development (IJTRD)*, ISSN: 2394-9333.
- Nunnally, J. C. (1967). *Psychometric theory*. McGraw-Hill.
- Özdemir, E. (2017, 20 Nisan). *Waterfall yazılım geliştirme*. ErenOzdemir <http://erenozdemir.net/waterfall-yazilim-gelistirme/> adresinden 01 Kasım 2023 tarihinde alınmıştır.
- Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T. ve Ozkaya, I. (2013). Parallel worlds: Agile and waterfall differences and similarities. *Software Engineering Institute*, 1(1), 1-3.
- Pett, M. A., Lackey, N. R. ve Sullivan, J. J. (2003). *Making sense of factor analysis*. Sage.
- Pressman, R. S. ve Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9. Baskı). McGraw-Hill.
- Project Management Institute. (2013). *A guide to the project management body of knowledge (PMBOK® guide)* (5. Baskı). PMI.
- Project Management Institute. (2017). *A guide to the project management body of knowledge (PMBOK® guide)* (6. Baskı). PMI.

- Rajagopalan, S., & Mathew, S. K. (2016). Choice of agile methodologies in software development: a vendor perspective. *Journal of International Technology and Information Management*, 25(1), 3.
- Ruhe, G. ve Wohlin, C. (Eds.). (2014). *Software project management in a changing world (21.Sayı)*. Springer.
- Schwaber, K. ve Sutherland, J. (2017, 02 Aralık). *The scrum guide: The definitive guide to scrum: The rules of the game*. Scrumguide <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf> adresinden 07 Ocak 2024 tarihinde alınmıştır.
- Sergeev, A. (2016). *Team roles in waterfall methodology*. Hygger <https://hygger.io/blog/team-roles-in-waterfall-methodology/> adresinden 07 Ocak 2024 tarihinde alınmıştır.
- Slack, N., Chambers, S., ve Johnston, R. (2007). *Operations management* (5. Baskı). Financial Times/Prentice Hall.
- Sommerville, I. (2011). *Software engineering* (9. Baskı). Pearson Education Inc.
- Sonjit, P., Dacre, N., ve Baxter, D. (2021). Disruption and agility dynamics in project management processes: An institutional theory approach. *Advanced Project Management*, 21(7). <https://dx.doi.org/10.2139/ssrn.3830762>
- Stanley, J. C., ve Gross, E. D. (2020). *Project management handbook: An introduction to Scrum, Agile and DevOps for absolute beginners*. Prosper Consulting Inc.
- Statistics Canada. (2021, 02 Ocak). *Study: Working from home: Productivity and preferences*. <https://www150.statcan.gc.ca/n1/daily-quotidien/210401/dq210401b-eng.htm> adresinden 03 Ocak 2024 tarihinde alınmıştır.
- Stepanek, G. (2005). *Software project secrets: Why software projects fail*. Apress.
- Sullivan, C. (2003). What's in a name? Definitions and conceptualisations of teleworking and homeworking. *New Technology, Work and Employment*, 18(3), 158-165.
- Tavşancıl, E. (2006). *Tutumların ölçülmesi ve SPSS ile veri analizi*. Nobel Yayınları.
- Villafiorita, A. (2014). *Introduction to software project management*. CRC Press.
- Vohra, P. ve Singh, A. (2013). A contrast and comparison of modern software process models. In *Proceedings of the International Conference on Advances in Management and Technology (ICAMT 2013)*, 23-27.
- Weiss, J. W. ve Wysocki, R. K. (1992). *5-phase project management: A practical planning & implementation guide*. Perseus Books Publishing.

EKLER

Ek1. Anket Formu

COVID-19 PANDEMİSİNİN PROJE YÖNETİM METOTLARININ İŞLEYİŞİNE OLAN ETKİSİNİN ARAŞTIRILMASI

Değerli Katılımcı,

Bu anket uzaktan çalışma modelinde Waterfall ve Agile yöntemlerinin nasıl etkilendiğini ve adaptasyonunun ne şekilde gerçekleştiğini kavramayı amaçlayan bir yüksek lisans tezinin veri toplama aracı olarak geliştirilmiştir. Yanıtlarınız anonim olarak kayıt altına alınmaktadır ve kişisel bilgilerinizin toplanması mümkün değildir. Elde edilen veriler sadece yüksek lisans tezinde ve bağlantılı bilimsel, akademik araştırmalarda kullanılacaktır.

☐ Onaylıyorum ☐ Onaylamıyorum

Uğur AZAKLI, Yüksek Lisans Öğrencisi, Yönetim Bilişim Sistemleri, Sakarya Üniversitesi

Naciye Güliz Uğur, Dr. Öğr. Üyesi, Yönetim Bilişim Sistemleri, Sakarya Üniversitesi

Aşağıdaki anket sorularına açık kaynaklardan erişilmiştir. Kaynaklar, KAYNAKÇA bölümünde belirtilmiştir.

Anket Soruları

- Cinsiyetiniz nedir?

☐ Kadın ☐ Erkek ☐ Belirtmek istemiyorum

- Kaç yıllık iş deneyimine sahipsiniz?

☐ 0-2 yıl ☐ 3-5 yıl ☐ 6-9 yıl ☐ 10 yıl ve üzeri

- Hangi proje yönetim modeli ile çalışıyorsunuz?

☐ Şelale (Waterfall) ☐ Çevik (Agile) ☐ Diğer.....

- Ekipteki rolünüz nedir?

☐ Proje yöneticisi ☐ İş analisti ☐ Yazılımcı ☐ Test mühendisi ☐ Test ortam altyapı sorumlusu ☐ Yazılım mimarı ☐ Diğer.....

- Çalışma şekliniz nedir?

☐ Uzaktan ☐ Hibrit ☐ İş yeri lokasyonunda

Sorular	Referans	Kesinlikle Katılıyor	Katılıyor	Ne Katılıyor Ne Katılmıyor	Katılmıyor	Kesinlikle Katılmıyor	Fikrim yok
Yöneticim beni destekler.	Tekkanat (2022)						
Projeye bağlı bir sponsor veya yönetici var.	Tekkanat (2022)						
Hiyerarşik yerine işbirlikçi bir kültür ile çalışıyorum.	Tekkanat (2022)						
Yüzyüze iletişim çalışma ortamında çok önemlidir.	Tekkanat (2022)						
Çalıştığım kurumda çevik yaklaşım genel olarak kabul görmektedir.	Tekkanat (2022)						
Belirlenen hedeflerden daha iyi sonuçlar elde edildiğinde ödüllendirilirim.	Tekkanat (2022)						
Proje ekibi bir arada çalışır.	Tekkanat (2022)						
Çalışma ortamımız proje için uygundur.	Tekkanat (2022)						
Proje ekibinin teknik yeterliliği ve uzmanlık seviyesi projeye uygundur.	Tekkanat (2022)						
Proje ekibi, projeye karşı yüksek bir bağlılık ve işbirliği içindedir.	Tekkanat (2022)						
Proje ekibi, mevcut proje yönetim modelinin süreçlerine hakimdir.	Tekkanat (2022)						
Proje ekibi değişen koşullara hızlı bir şekilde uyum sağlayabilir.	Tekkanat (2022)						
Proje ekibi işleri kendi aralarında organize edebilir.	Tekkanat (2022)						
Proje ekibinin, müşteri (iş birimi, ürün sahibi vb) ile iyi bir ilişkisi vardır.	Tekkanat (2022)						
Projenin kapsamı ve hedefleri net tanımlanmıştır.	Tekkanat (2022)						
Proje sürecinde gelen talepler sürekli olarak değerlendirmeye alınabilir ve önceliklendirilebilir.	Tekkanat (2022)						
Proje süreçleri hızlı ve esnek bir şekilde yönetilmektedir.	Chow and Cao (2008)						
Proje ekibi, gereksinimlere uyum sağlama konusunda hızlı ve esnektir.	Chow and Cao (2008)						
Projede aksiyon alma konusunda hızlı ve esneğiz.	Chow and Cao (2008)						
Güçlü iletişim proje ekibi için önceliklidir.	Chow and Cao (2008)						
Projede takvime uyma konusunda problem yaşanmaz.	Chow and Cao (2008)						
Müşterinin (iş birimi, ürün sahibi vb) projeye bağlılığı ve projeye sürekli ilgilenmesi proje ekibi için kritiktir.	Chow and Cao (2008)						
Müşteri yetkinliği, (iş birimi, ürün sahibi vb) projenin her aşamasında çıkan sorunlarla ilgilenmesi ve karar alması için yeterlidir.	Tekkanat (2022)						
Projede, teknik standartlara uyum konusunda problem yaşanmaz.	Tekkanat (2022)						
Tasarım aşamasında mümkün olduğunca basit, anlaşılır ve karmaşık olmayan bir yapı tercih edilmiştir.	Tekkanat (2022)						
Geliştirme sürecinde kodlar sık sık gözden geçirilir, iyileştirilir.	Tekkanat (2022)						
Proje sürecinde optimum seviyede dokümantasyon yapılır. (Dokümana aşırı zaman harcanmaz.)	Chow and Cao (2008)						
Projede birim ve entegrasyon testine önem verilmektedir.	Chow and Cao (2008)						
Proje parçalı olarak, planlanan belli zamanlarda canlıya alınır.	Chow and Cao (2008)						
Projede öncelikli en önemli modüller canlıya alınır. (Amaç, mümkün olan en kısa sürede çalışan yazılımı canlıya almaktır.)	Chow and Cao (2008)						
Proje için mevcutta kullanılan alt yapı ve teknoloji, tüm isterleri karşılayacak düzeydedir.	Chow and Cao (2008)						
Proje ekibine gereksinim duyulan konularda eğitim verilir ve teknik bilgi seviyesi artırılır.	Chow and Cao (2008)						
Üzerinde çalışılan proje, hayati öneme sahip olmayan bir projedir.	Chow and Cao (2008)						

Proje kapsamı, proje sürecinin en başında net olarak belirlenir.	Chow and Cao (2008)						
Proje süreci boyunca dinamik bir takvim ile çalışılır.	Chow and Cao (2008)						
Proje sürecinin en başında maliyet ve kaynak planlaması detaylı olarak yapılır.	Chow and Cao (2008)						
Risk analizi, proje sürecinin başında yapılır.	Chow and Cao (2008)						

KAYNAKÇA

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961–971. <https://doi.org/10.1016/j.jss.2007.08.020>

Tekkanat, E. (2022). IMPACT OF REMOTE WORK ON AGILE PROJECTS' CRITICAL SUCCESS FACTORS: BANK CASE STUDY IN THE CONTEXT OF COVID-19 PANDEMIC. BAHÇEŞEHİR UNIVERSITY



Ek 2. Etik Kurul Onay Belgesi

Evrak Tarih ve Sayısı: 17.05.2024-E.362008



T.C.
SAKARYA ÜNİVERSİTESİ REKTÖRLÜĞÜ
Etik Kurulu



Sayı : E-61923333-050.99-362008
Konu : 69/35 Uğur AZAKLI

17.05.2024

Sayın Uğur AZAKLI

İlgi : 09.05.2024 tarihli ve E-050.99-0 sayılı yazınız.

Üniversitemiz Sosyal ve Beşeri Bilimler Etik Kurulunun 15.05.2024 tarihli ve 69 sayılı toplantısında alınan "35" nolu karar ile Uğur AZAKLI'nın başvurusu **uygun** görülmüş ve karar örneği ekte sunulmuştur.

Bilgilerinizi rica ederim.

Prof. Dr. Selim İNANÇLI
Sosyal ve Beşeri Bilimler Etik Kurulu
Başkanı

Ek: Karar Yazısı (1 Sayfa)

Bu belge, güvenli elektronik imza ile imzalanmıştır.

Doğrulama Kodu: BS9LCSHSPK Pin Kodu: 04092

Belge Takip Adresi : <https://turkiye.gov.tr/ehd/teK=5783&eD=BS9LCSHSPK&eS=362008>

Adres: Esentepe Kampüsü 54187 Söğütözü SAKARYA / KEP Adresi:
sakaryaniversitesi@hs01.ksp.tr
Telefon No: 0264 295 50 00 Faks No: 0264 295 50 31
e-Posta: ozsalikalem@sakarya.edu.tr Elektronik Ağ: www.sakarya.edu.tr

Bilgi için: Hanife Babacan
Unvanı: Birim Evrak Sorumlusu



Ek 2.1. Karar Belgesi

KARAR

35. Uğur AZAKLI'nın " Covid-19 Pandemisinin Proje Yönetim Metotlarının İşleyişine Olan Etkisinin Araştırılması " başlıklı çalışması görüşmeye açıldı.

Yapılan görüşmeler sonunda Uğur AZAKLI'nın " Covid-19 Pandemisinin Proje Yönetim Metotlarının İşleyişine Olan Etkisinin Araştırılması " başlıklı çalışmasının Etik açıdan **uygun** olduğuna oy birliği ile karar verildi.

ÖZGEÇMİŞ

Adı Soyadı: Uğur AZAKLI

ÖĞRENİM DURUMU

Öğrenim Derecesi	Öğrenim Yeri	Öğrenim Yılı
Yüksek Lisans	Sakarya Üniversitesi / İşletme Enstitüsü / Yönetim Bilişim Sistemleri	2024
Lisans	Sakarya Üniversitesi / Mühendislik Fakültesi / Endüstri Mühendisi	2016
Lise	Yahya Kaptan Anadolu Lisesi	2009

İŞ DENEYİMİ

Yıl	Yer	Görev
2023	OBSS Teknoloji	Kıdemli İş Analisti
2022	PiA	İş Analisti
2020	Türk Telekom	Proje Test Lideri
2018	Kafein Teknoloji	Yazılım Sürüm Yöneticisi
2018	Garanti Teknoloji	Test Analisti
2016	Keytorc Teknoloji	Test Mühendisi

YABANCI DİL

İngilizce