

**T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇEVİK YÖNTEMLERLE MOBİL UYGULAMA TASARIMI VE  
GERÇEKLEŞTİRİLMESİ**

**İSMAİL DURU**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI  
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN  
PROF. DR. OYA KALIPSIZ**

**İSTANBUL, 2013**

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇEVİK YÖNTEMLERLE MOBİL UYGULAMA TASARIMI VE**  
**GERÇEKLEŞTİRİLMESİ**

İsmail DURU tarafından hazırlanan tez çalışması 10.01.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Prof. Dr. Oya KALIPSIZ  
Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Prof. Dr. Oya KALIPSIZ  
Yıldız Teknik Üniversitesi

Prof. Dr. Selim AKYOKUŞ  
Doğuş Üniversitesi

Yrd. Doç. Dr. Yunus Emre SELÇUK  
Yıldız Teknik Üniversitesi

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## ÖNSÖZ

---

Mobil uygulamalarda çevik yazılım geliřtirmeyi inceleyen ve mobil uygulama geliřtirmede çevik yazılım geliřtirme metodunun uygulanması sürecini iyileřtirmeyi amaçlayan bu tez çalıřması Yıldız Teknik Üniversitesi, Bilgisayar Mühendislięi Bölümü Yüksek Lisans programında yürütülmüřtür.

Bu tezin hazırlanması süresi boyunca, bilgisi, deneyimi ve önerileriyle arařtırma ve geliřtirmelerimi yönlendiren tez danıřmanım sayın Prof. Dr. Oya KALIPSIZ'a sonsuz teřekkürü borç bilirim.

Yıldız Teknik Üniversitesi Yazılım Anabilim Dalı öğretim üyeleri bařta olmak üzere Bilgisayar Mühendislięi Bölümündeki bütün deęerli hocalarıma ve arkadaşlarıma katkılarından ve desteklerinden dolayı teřekkür ederim.

Bu tezi, maddi ve manevi hiçbir fedakârlıktan kaçınmayıp desteklerini hiçbir zaman esirgemeyen sevgili aileme ithaf ediyorum.

Aralık, 2013

İsmail DURU

## İÇİNDEKİLER

---

	Sayfa
KISALTMA LİSTESİ.....	vii
ŞEKİL LİSTESİ.....	viii
ÇİZELGE LİSTESİ .....	x
ÖZET .....	xi
ABSTRACT.....	xiii
<b>BÖLÜM 1</b>	
GİRİŞ.....	1
1.1 Literatür Özeti .....	2
1.2 Tezin Amacı .....	3
1.3 Hipotez.....	4
<b>BÖLÜM 2</b>	
ÇEVİK SÜREÇLER.....	5
2.1 Tarihçe .....	6
2.2 Çevik Yazılım Geliştirme Özellikleri .....	6
2.3 Çevik Yazılım Geliştirme Mimarisi.....	7
2.4 Çevik Yazılım Geliştirme Metotları.....	8
2.4.1 Extreme Programlama.....	8
2.4.2 Scrum .....	9
2.4.3 Metodların Kristal Ailesi .....	9
2.4.4 Özellik Tabanlı Geliştirme .....	9
2.4.5 Rup Modeli İle Geliştirme .....	9
2.4.6 Dinamik Sistem Geliştirme Metodu .....	9
2.4.7 Adaptif Yazılım Geliştirme .....	9
2.4.8 Açık Kaynak Kodlu Yazılım Geliştirme.....	10
<b>BÖLÜM 3</b>	
MOBİL UYGULAMALAR .....	11

3.1	Gelişim Süreci.....	12
3.2	Mobil Uygulamaların Özellikleri.....	12
3.3	Mobil Uygulama Çeşitleri.....	13
3.3.1	Kişisel Yardımcılar .....	13
3.3.2	Coğrafi Konum Destekli Rehberler .....	13
3.3.3	Sosyal Platformlar.....	13
3.3.4	Çokluortam Araçları.....	14
3.3.5	Özel Çözümler.....	14
3.3.6	Marka Çözümleri .....	14
3.3.7	Anlık Mesajlaşma Araçları .....	14
3.3.8	Mobil Marketler.....	14
3.3.9	Zaman Geçiriciler .....	15
3.3.10	Mobil Eğitimciler.....	15
3.3.11	Eğlence Ve Yaşam Tarzı .....	15
3.4	Mobil İşletim Sistemleri .....	15
3.4.1	Apple/iPhone .....	15
3.4.2	Google/Android .....	16
3.4.3	Nokia/Ovi.....	16
3.4.4	Blackberry.....	16
3.4.5	Windows Mobile.....	16

## BÖLÜM 4

MOBİL UYGULAMALARDA ÇEVİK YAZILIM GELİŞTİRME .....	18	
4.1	İlgili Çalışmalar .....	19
4.2	Kullanılan Çevik Yazılım Geliştirme Metodolojileri.....	19
4.2.1	Mobil-D.....	20
4.2.2	Hibrid Metot Mühendisliği .....	21
4.2.3	Scrum.....	22
4.2.4	Masam .....	23
4.3	Mobil Uygulamalarda Çevik Yazılım Geliştirmenin Farklılıkları .....	23
4.4	Yazılım Geliştirme Süreci Yönetim Uygulamaları.....	24
4.4.1	Uygulamaların Özellikleri.....	25
4.4.1.1	Planlama Ve Zaman Yönetimi .....	26
4.4.1.2	Raporlama.....	26
4.4.1.3	İzin Ve Çalışma Günleri .....	26
4.4.1.4	İzleme Ve Analiz .....	26
4.4.1.5	Fikirler Ve Geri Besleme.....	26
4.4.1.6	Çevik Yazılım Geliştirme(Scrum) Özellikleri .....	27
4.4.1.7	Ekip Üyeleri Yönetimi .....	27
4.4.2	Uygulamaların İncelenmesi .....	27
4.4.2.1	Telerik Team Pulse .....	27
4.4.2.2	Trello .....	30
4.4.2.3	Atlassian Jira .....	32
4.4.2.4	Microsoft Team Foundation Server.....	33
4.4.2.5	Eclipse EPF .....	35

## BÖLÜM 5

MOBİL UYGULAMALARDA ÇEVİK YAZILIM SÜRECİ YÖNETİM ARACI ANALİZİ .....	37
5.1 Gereksinim Analizi .....	39
5.1.1 Gereksinimlerin Alınması.....	39
5.1.1.1 Anket.....	39
5.1.2 Gereksinimlerin Değerlendirilmesi.....	40
5.1.2.1 Şirket-Organizasyon Bilgileri .....	40
5.1.2.2 Kullanılan Yazılım Geliştirme Metodolojileri Bilgileri .....	42
5.1.2.3 Kullanılan Proje Yönetim Aracı Bilgileri.....	48
5.1.2.4 Mobil Uygulama Geliştirme Bilgileri .....	54

## BÖLÜM 6

MOBİL UYGULAMALARDA ÇEVİK YAZILIM SÜRECİ YÖNETİM ARACI TASARIMI VE .....	
GERÇEKLEMESİ.....	59
6.1 Tasarım.....	59
6.1.1 İşlevsel Modelleme .....	60
6.1.2 Veri Modeli .....	60
6.1.3 Veritabanı Tasarımı .....	60
6.1.3.1 Tablo Bilgileri .....	60
6.2 Gerçekleme .....	63
6.2.1 Ekran Örnekleri.....	63
6.3 Aracın Değerlendirilmesi.....	67

## BÖLÜM 7

SONUÇ VE ÖNERİLER .....	71
KAYNAKLAR.....	75
EK-A	
ANKET SORULARI .....	78
EK-B	
AYRINTILI KULLANICI DİYAGRAMI.....	84
EK-C	
AYRINTILI VARLIK İLİŞKİ MODELİ.....	85
ÖZGEÇMİŞ.....	86

## KISALTMA LİSTESİ

---

API	Application Programming Interface
EDGE	Enhanced Data Rates For GSM Evolution
EPF	Eclipse Process Framework
GSM	Global System For Mobile Communications
GPS	Global Positioning System
HSPA	High Speed Packet Access
INC	Incorporated
Java ME	Java Micro Edition
MAC iOS	Macintosh iPhone Operating System
MASAM	Mobile Application SW Based On Agile Methodology
Ms Project	Microsoft Office Project
PDA	Personal Digital Assistant
RPG	Role Playing Game
RIM	Research In Motion
RUP	Rational Unified Process
UML	Unified Modelling Language
UMTS	Universal Mobile Telecommunications System
WCDMA	Wideband Code Division Multiple Access
WIMAX	Worldwide Interoperability For Microwave Access
Wi-BRO	Wireless Broadband
Wi-Fi	Wireless Fidelity
2G	Second Generation
3G	Third Generation

## ŞEKİL LİSTESİ

	Sayfa
Şekil 4. 1	Mobile-D yaklaşımındaki aşamalar ..... 20
Şekil 4. 2	Mobile-D Geliştirme Günleri ..... 20
Şekil 4. 3	Hibrid Metod Mühendisliği Genel Yazılım Geliştirme Hayat Döngüsü ..... 21
Şekil 4. 4	Hibrid Metod Mühendisliği İle Geliştirilen Final Yöntem..... 22
Şekil 4. 5	Telerik TeamPulse Kullanıcı Görevleri Durumu ..... 29
Şekil 4. 6	Telerik TeamPulse 2. Uygulama Geliştirme Evresi Görev Tahtası ..... 30
Şekil 4. 7	İlk Sprint Sonucunda Elde Edilen Yazılımın Ekran Görüntüsü ..... 30
Şekil 4. 8	Trello Aracı İle Çevik Yazılım Proje Yönetimi..... 31
Şekil 5. 1	Çalışma Yeri ..... 40
Şekil 5. 2	Çalışılan Şirketin Sektörü..... 41
Şekil 5. 3	Çalışılan Organizasyonun Büyüklüğü ..... 41
Şekil 5. 4	Mobil Uygulama Geliştirilirken Kullanılan Ürün Gereksinim Özellikleri ..... 42
Şekil 5. 5	Scrum'dan Önce Kullanılan Yazılım Geliştirme Metodolojileri ..... 43
Şekil 5. 6	Scrum Yöntemini Seçmenin Ana Etkieni ..... 44
Şekil 5. 7	Scrum Yazılım Geliştirme Metodolojisi Kullanımında Önemli Özellikler ..... 45
Şekil 5. 8	Scrum Metodolojisinde Mobil Uygulama Geliştirirken Önemli Özellikler .. 46
Şekil 5. 9	Çevik Yöntemin Mobil Uygulama Geliştirmeye En Fazla Katkı Alanı ..... 47
Şekil 5. 10	Mobil Uygulama Proje Yönetim Metodolojilerinden Hangisi Uygun ..... 48
Şekil 5. 11	Mobil Uygulama Geliştirirken Kullanılan Proje Yönetim Aracı Özellikleri ... 49
Şekil 5. 12	Mobil Uygulamalarda Scrum Süreci Yönetiminde Araç Gerekli mi? ..... 50
Şekil 5. 13	Mobil Uygulamalarda Scrum Süreci Yönetiminde Araç Kullanıyor musun? 51
Şekil 5. 14	Mobil Uygulamalarda Yazılım Geliştirme Sürecinde Kullandığınız Araç? .... 52
Şekil 5. 15	Mobil Uygulama Geliştirme Sürecinde Kullanılan Aracın Eksik Özellikleri .. 53
Şekil 5. 16	Mobil Uygulama Proje Yönetim Aracına Hangi Özellikleri Eklerdiniz? ..... 54
Şekil 5. 17	Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi ..... 55
Şekil 5. 18	Yazılım Geliştiricinin Mobil Uygulama Geliştirme Tecrübesi ..... 56
Şekil 5. 19	Şirketteki Mobil Uygulama Geliştirici Sayısı..... 57
Şekil 5. 20	Mobil Uygulama Geliştirme Yapılan Platformların İsimleri ..... 58
Şekil 6. 1	Scrum Master Ve Ürün Sahibi Tabloları ..... 60
Şekil 6. 2	Takım Üyesi Ve Misafir Kullanıcı Tabloları ..... 61
Şekil 6. 3	Sistem Yöneticisi Tablosu ..... 61
Şekil 6. 4	Proje Ve Toplantı Tabloları..... 62
Şekil 6. 5	Kaynak Tablosu ..... 62



Şekil 6. 6	Kullanıcı Giriş Ekranı .....	63
Şekil 6. 7	Kullanıcı Kaydı Ekleme Ekranı .....	63
Şekil 6. 8	Sistem Yöneticisi Ekranı .....	64
Şekil 6. 9	Ürün Sahibi Ekranı.....	64
Şekil 6.10	Scrum Master Ekranı.....	65
Şekil 6.11	Takım Üyesi Ekranı.....	65
Şekil 6.12	Misafir Kullanıcı Ekranı.....	66
Şekil 6.13	Proje Tanımlama Ekranı .....	66
Şekil 6.14	Toplantı Tanımlama Ekranı .....	66
Şekil 6.15	Raporlama Ekranı-1 .....	67
Şekil 6.16	Raporlama Ekranı-2 .....	67

## ÇİZELGE LİSTESİ

	Sayfa
Çizelge 5. 1 Çalışma Yeri.....	40
Çizelge 5. 2 Çalışılan Şirketin Sektörü .....	40
Çizelge 5. 3 Çalışılan Organizasyonun Büyüklüğü .....	41
Çizelge 5. 4 Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi .....	42
Çizelge 5. 5 Yazılım Geliştirici Mobil Uygulama Geliştirme Tecrübesi .....	43
Çizelge 5. 6 Şirketteki Mobil Uygulama Geliştirici Sayısı.....	44
Çizelge 5. 7 Mobil Uygulama Geliştirme Yapılan Platformların İsimleri .....	45
Çizelge 5. 8 Mobil Uygulama Geliştirilirken Kullanılan Ürün Gereksinim Özellikleri ....	45
Çizelge 5. 9 Mobil Uygulama Geliştirmede Kullanılan Proje Yönetim Aracı Özellikleri	46
Çizelge 5. 10 Scrum'dan Önce Kullanılan Yazılım Geliştirme Metodolojileri.....	47
Çizelge 5. 11 Scrum Yöntemini Seçmenin Ana Etkeni .....	48
Çizelge 5. 12 Scrum Yazılım Geliştirme Metodolojisi Kullanımında Önemli Özellikler ...	49
Çizelge 5. 13 Mobil Uygulama Geliştirmede Scrum Metodolojisinin Önemli Özellikleri	50
Çizelge 5. 14 Mobil Uygulamada Scrum Süreci Yönetiminde Araç Gerekli mi? .....	51
Çizelge 5. 15 Mobil Uygulamada Scrum Süreci Yönetiminde Araç Kullanıyor musun?..	52
Çizelge 5. 16 Mobil Uygulamalarda Yazılım Geliştirme Sürecinde Kullandığınız Araç? ...	53
Çizelge 5. 17 Mobil Uygulama Geliştirme Sürecinde Kullanılan Aracın Eksik Özellikleri	54
Çizelge 5. 18 Çevik Yöntemin Mobil Uygulama Geliştirmeye En Fazla Katkı Alanı? .....	55
Çizelge 5. 19 Mobil Uygulama Proje Yönetimi Metodolojilerinden Hangisi Uygun .....	56
Çizelge 5. 20 Mobil Uygulama Proje Yönetimi Aracına Hangi Özellikleri Eklediniz? .....	57
Çizelge 6. 1 Proje Yönetim Aracı Özelliklerinde Meydana Getirilen Özelleştirmeler ...	68

## ÇEVİK YÖNTEMLERLE MOBİL UYGULAMA TASARIMI VE GERÇEKLEŞTİRİLMESİ

İsmail DURU

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Oya KALIPSIZ

Akıllı telefonların kullanımının yaygınlaşması ile mobil uygulama geliştirme alanında yürütülen proje sayısı çok artmıştır. Bununla birlikte kullanıcıların uygulamalardan beklentileri ve proje yönetimindeki gereksinimler de artmıştır. Yazılım geliştirme metotlarında, mobil uygulama geliştirmede karşılaşılan özel durumlar ve kısıtlar ek olarak incelenmemekte ve mobil uygulama platformlarda uygulama geliştirme hakkında yeterli bilgi sunulmamaktadır.

Bu çalışmada, çevik yazılım geliştirme prensipleri dikkate alınarak mobil uygulama geliştirme platformlarında proje yönetimi hakkında bilgilendirmede bulunulacaktır. Farklı mobil uygulama tiplerinin özelliklerinin yazılım proje yönetimindeki etkisi incelenerek, proje yönetim sürecinin iyileştirilmesi için yapılması gerekenler ortaya konulmaya çalışılacaktır. Mobil uygulama geliştirmede karşılaşılan kısıtlar ve bunların yazılım geliştirme süreci aşamalarındaki etkileri göz önünde bulundurularak elde edilen sonuçların analizi yapılacaktır.

Mobil uygulama geliştirmede değişen roller dikkate alınacak ve mobil uygulamalarda proje yönetimini kolaylaştıran araçlardan faydalanılacaktır. Performansı arttırmaya yönelik öneriler yapılacak ve önerilerin kullanım yer ve amaçları ortaya konulacaktır. Mobil uygulama geliştirmede endüstri ve pazarın talep ve gereklilikleri nedeni ile sahip olunması gereken kısıtlar ortaya konacaktır.

**Anahtar Kelimeler:** Mobil Uygulama Geliřtirme, Proje Yönetimi, Çevik Metot, Mobil Uygulama Geliřtirme Kısıtları, Çevik Metot Yönetim Uygulamaları, Mobil İşletim Sistemleri

**DESIGN AND IMPLEMENTATION OF MOBILE APPLICATIONS WITH AGILE  
METHODOLOGIES**

İsmail DURU

Department of Computer Engineering

MSc. Thesis

Adviser: Prof. Dr. Oya KALIPSIZ

With the widespread use of smart phones, the number of mobile application development has increased. However, requirements for project management and expectations of users from applications have also increased. In software development methodologies, special situations and constraints encountered in mobile application development are not investigated. In addition, in mobile application development platforms, sufficient information for application development is not available.

In this study, taking into account the principles of Agile software development principles, project management in mobile application development platforms will be informed. As investigating effect of the different types of mobile applications in software project management process to have better project management process, changes needed to be done will be presented. Constraints which encountered in mobile application development stages and their effects and importance in software development process will be studied and obtained results will be presented.

The changing roles in mobile application development and tools those makes project management process easier will be informed. Suggestions to improve quality and their purposes and usage platforms will be displayed. Because of the industry and market-requests and necessities, Constraints in mobile application development will be exposed.

**Key Words:** Mobile Application Development, Project Management, Agile Methodology, Mobile Application Development Constraints, Agile Methodology Management Applications, Mobile Operating Systems

### GİRİŞ

Mobil bilgisayarlardaki gelişmeler ve cep telefonlarından internet kullanımının yaygınlaşması ile mobil uygulama alanında yürütülen proje sayısı çok artmıştır. Var olan yazılım geliştirme metotları, mobil platformlar için bu yöntemlerin nasıl uygulanabileceğine özel olarak değinmemektedir.

Tez kapsamında, mobil uygulama geliştirme platformlarının kendine özgü koşulları dikkate alınarak, mobil uygulama geliştirmede izlenilmesi gereken yol, çevik yazılım geliştirme prensiplerinden faydalanılarak açıklanmaktadır.

Günümüze kadar, mobil uygulamalarda çevik yazılım geliştirme için önerilen metotlar vardır. Bu metotların kullanımı ve birbirleri ile karşılaştırmaları tezin hazırlanması sürecinde yapılan yayınlarda ortaya konulmuştur. Metotlardan mobil uygulamalar için Scrum'ın kullanımı üzerinde daha fazla durulmuş ve geliştirilen örnek bir uygulama üzerinde Scrum metodunun kullanımı incelenmiştir.

Kullanılan çevik metotlarla ilgili çalışmaların sonrasında mobil uygulama sistemlerinde çevik yazılım geliştirme yönteminin uygulanması ile elde edilen sonuçlar hakkında bilgilendirmede bulunulmuştur.

Tez çalışması kapsamında çevik yazılım geliştirme metotlarının uygulanmasını kolaylaştırmak için geliştirilen araçlardan faydalanılmıştır. Araçların özellikleri ve kullanımları, geliştirilen örnek uygulamalar yardımı ile incelenerek mobil uygulama geliştirmede çevik yazılım geliştirme metodunun araçlarla birlikte uygulanması sonucu görülen kazanımlar ortaya konulmuştur.

Çevik yazılım geliştirme sürecinde kullanılan araçlara ilgili çalışmalar, mobil uygulamalar için yeni bir işletim ve yönetim uygulaması geliştirilmesinin gerekli olduğunu ortaya koymuştur. Yeni işletim ve yönetim aracı geliştirilmesi sürecinde oluşturulan tasarım ile mobil uygulama geliştirmede proje yönetiminin etkinliğini arttırmak ve aynı zamanda var olan işletim ve yönetim uygulamalarının kullanımında görülen sorunları ortadan kaldırarak yüksek performans, kolay yönetim ve farklı türlerde mobil uygulamalar için adaptasyon sağlanması hedeflenmiştir. Tezin hedeflerinden biri de, örnek bir mobil uygulama üzerinden geliştirilen aracın var olan araçlara göre sahip olduğu farklılık ve avantajları ortaya koymak olmuştur.

### **1.1 Literatür Özeti**

Günümüze kadar, mobil uygulamalar ve çevik yazılımlarla ilgili gerçekleştirilmiş birçok çalışma mevcuttur. Çevik yazılımlarla ilgili gerçekleştirilen çalışmalarda, çevik yazılım geliştirme metodunun ilkeleri, özellikleri, uygulanması ve etkileri ortaya konulmuştur. Mobil uygulamalarla ilgili çalışmalarda, uygulamaların özellikleri ve çeşitleri, mobil uygulama geliştirme platformlarının çeşitleri ve özellikleri ile ilgili bilgiler sunulmuştur.

Mobil uygulamalarda çevik yazılım geliştirme, mobil uygulamalar ve çevik yazılımlarla ilgili çalışmaların kesişim noktasında bulunmaktadır. Mobil uygulamalarda çevik yazılım geliştirme metodunun daha iyi uygulanabilmesini sağlamak için yeni çevik metotlar önerilmiş ve metotların adımları ve özellikleri ortaya konulmuştur. Aynı zamanda, mobil uygulamalarda çevik yazılım geliştirmenin farklılıkları üzerinde çalışmalar da gerçekleştirilmiştir.

Çevik yazılım geliştirme sürecinin uygulanmasını kolaylaştırmaya katkı sağlamak için kullanılan araçlar vardır. Bu araçların özellikleri ve alt bölümleri ayrıntılı, olarak incelenebilmektedir. Araçlarla ilgili gerekli analizlerin yapılması ile hitap ettikleri kullanıcı kitleleri, sundukları farklılık ve avantajların belirlenmesi mümkündür.

Çevik yazılım geliştirme sürecinde kullanılacak birçok araç olmasına rağmen, mobil uygulamalarda çevik yazılım geliştirme sürecinde kullanılmak üzere, mobil uygulama geliştiriciler için özelleştirilmiş bir araç mevcut değildir. Mobil uygulama geliştirme süreci için özelleştirilmiş yeni bir aracın geliştirilmesi için ise, çevik yazılım gelişme



metotları, mobil uygulamalar ve çevik yazılım geliştirme süreci yönetiminde kullanılan araçlar hakkında yeterli bilgi sahibi olmak ve mobil uygulama geliştiricilerin beklentilerini göz önünde bulundurmak gerekmektedir.

## 1.2 Tezin Amacı

Tezimizin amacı, çevik yazılım geliştirme metodunun mobil uygulama geliştirme sürecinde uygulanmasında karşılaşılan ve aşağıda değinilen zorluklar, kısıtlar ve gereksinimler göz önünde bulundurularak mobil uygulamalarda çevik yazılım geliştirme metodunun kullanımı ile elde edilen faydaları arttırmak ve metodun daha yüksek performans ile etkin olarak uygulanabilmesini sağlamaktır. [1]

Mobil uygulama sistemlerinde Java ME, Symbian, iOS ve Android gibi işletim sistemleri ile kapalı sistemden açık platforma geçiş ile birlikte ortaya çıkan güncel durumun ve gelecek öngörülerinin incelenmesinin ardından çevik yazılımlar, mobil uygulama geliştirmede kullanılmaya başlamıştır. Mobil uygulamalarda çevik yöntemler uygulanırken karşılaşılan bazı zorluklar vardır ki bu zorluklar aşağıdaki gibi listelenebilir[1]:

- Platforma özgü geliştirme yeteneği gereksinimi
- Mobil cihazlardaki çeşitliliğin getirdiği taşınabilirlik sorunu
- Mobil uygulamalar için test tabanlı geliştirme araçlarının olmaması
- Sık yayın çevrimine alışma
- Çevik metotların değiştirilmeden kullanılmasının yarattığı sorunlar

Mobil uygulama geliştirmede çevik yazılım geliştirme metodunu kullanmayı gerekli kılan bazı kısıtlar ve gereksinimler vardır. Bunlar aşağıdaki gibi listelenebilir:

- Mobil uygulama geliştirmede isteklerin ve platformların çok sık değişebilmesi [2]
- Mobil uygulamalarda geliştirilen ürünün pazara hızlı çıkma zorunluluğunun olması [2]
- Mobil yazılım geliştirmede genelde küçük uygulama geliştirme ekiplerinin yer alması [3]

- Mobil uygulamaların boyutları genelde 10.000 satırdan daha az olması [3]

### **1.3 Hipotez**

Günümüzde çevik metotların kullanım alanları her geçen gün genişlemektedir. Çevik metotların mobil uygulamalarda kullanımı henüz büyük bir geçmişe sahip olmamakla birlikte, her geçen gün artmaktadır.

Tezimiz, mobil uygulamaların ve çevik yazılımlar üzerinde gerçekleştirilecek çalışmalarla mobil uygulamalarda çevik yazılım geliştirmenin daha iyi hale getirilebileceği kabulü üzerine oluşturulmuştur. Bu kabul ile birlikte, mobil uygulamalar, mobil platformları ile yazılım geliştirme süreçlerinin özellikleri ve sahip oldukları kısıtlar dikkate alınarak sürecin iyileştirilebilmesi için gerekli ihtiyaçlar belirlenmiştir.

Mobil uygulamalar, mobil platformlar ve yazılım geliştirme süreçleri ile ilgili çalışmalardan sonra mobil uygulamalar için özelleştirilmiş bir araç ile mobil uygulama geliştirmenin daha etkin hale getirilebileceğine karar verilmiştir.

## BÖLÜM 2

---

### ÇEVİK SÜREÇLER

Çevik yazılım geliştirme, çevik yazılım geliştirme manifestosunu kendisine rehber edinen yazılım geliştirme metodudur. Çevik yazılım geliştirme manifestosunun değer verip kanaat getirdiği temel ilkeler aşağıdaki gibi sıralanabilir [4];

- Süreçler ve araçlardan ziyade bireyler ve etkileşimlere
- Kapsamlı dokümantasyondan ziyade çalışan yazılıma
- Sözleşme pazarlıklarından ziyade müşteri ile işbirliğine
- Bir plana bağlı kalmaktan ziyade değişime karşılık verme

Çevik yazılım geliştirme, yazılım geliştirme süreçlerinden kaynaklanan sorunlara karşı çevik yazılım geliştirme metodunun ilkeleri ile çözüm getirmek için ortaya atılmıştır. Çevik yazılım geliştirme metodunun çözüm getirdiği sorunlar:

- Teknolojinin çok hızlı gelişmesi ve yeniliklerin projeye uygulanamaması
- Müşterilerin proje başlangıcında gereksinimlerini tam olarak ortaya koyamamaları
- Müşterilerin gereksinimlerinin çok sık değişmesi nedeni ile ortaya çıkan yazılımın müşterilerin güncel ihtiyaçlarına cevap verememesi
- Her geçen gün projelerin yönetiminin gittikçe daha zor ve karmaşık hale gelmesi

## 2.1 Tarihçe

Son birkaç yıldır yazılım geliştirme metotlarındaki en önemli yeniliklerden biri de çevik yazılım geliştirme ilkelerinin tanımlanması olmuştur. 2001 yılında Çevik Yazılım Geliştirme Manifestosu oluşturulduktan sonra 2001 yılı dâhil olmak üzere geçen yıllar içerisinde uygulanmaya başlayan birkaç çevik yazılım geliştirme metodu vardır. Bu metotların bazı örnekleri aşağıdaki gibi listelenebilir [5]:

- DSDM
- Uç Programlama (Extreme Programlama)
- Özellik Tabanlı Geliştirme
- Scrum

Bu metotların güçlü noktaları, metotlar uygulandığında yazılım geliştirme sürecinin değişen çevreye karşı daha duyarlı hale getirilebilmesidir. Bu metotlarda, çalışan yazılım, ayrıntılı dokümantasyonun daha üzerinde tutulur; kişiler ve etkileşimler, araçlar ve süreçlerden daha önemli kabul edilir ve müşteri iş birliği, müzakere edilen sözleşmeden daha değerlidir. [5]

Son yıllarda, çevik yöntemlerin başarısı çok sayıda vaka üzerinde kanıtlanmıştır. Günümüzde küçük şirketlerden uluslararası şirketlere kadar birçok şirket Scrum yöntemini uygulamaya başlamış durumdadır. Araştırmalar, şirket içerisinde Scrum yönteminin kullanımının önemli başarıları getirebileceğini ve Scrum yönteminin kullanım alanlarının yerel projelerle sınırlı olmadığını göstermiştir. [5]

## 2.2 Çevik Yazılım Geliştirme Özellikleri

Çevik yazılım geliştirme, dinamik, bağlama özgü, sertçe değişmeye zorlayan ve büyüme odaklı bir yazılım geliştirme metodudur. Bu durum metodun kullanımındaki başarıma ve kazanma isteği ile ilgilidir. Günümüzde, ortaya çıkan rekabetçi alanlarda başarılı olma, kar kazanmada pazar payı ve müşteriler birçok şirketin korku kaynağı olan rekabetçi dalgaların merkezinde yer almaktadır [6]

İnsanları etkin kullanmak, manevra kabiliyeti, hız ve maliyet tasarrufu kazandırır. Yüz yüze konuşma, dökümanları okuma ve yazmaya göre daha hızlı düşünce alışverişinde

bulunabilmeyi sağlar. Birlikte çalışan birkaç tasarımcı, her birinin tek başına üreteceği tasarımdan daha iyi bir tasarım ürünü ortaya koyabilir. [6]

Geliştiriciler, müşterilerle ve sponsorlarla konuştuklarında zorlukları giderebilir ve öncelikleri ayarlayabilirler. Geliştiricilerin birlikte çalışmaları mümkün olmadığında, ilerlemek için alternatif yollar bulabilirler. Süreçler, araçlar, belgeler, sözleşmeler ve planlar yararlıdır ama ilerlemeyi gösterme zamanı geldiğinde karşı tarafa bir şeylerin verilmesi gerekir. Nelerin kalacağı ve nelerin verileceği noktasında açık olmak gereklidir. [6]

Bireyler arası etkileşimlere bağlı olarak hızlı bilgi paylaşımı, gerektiğinde süreci değiştirmeyi kolaylaştırır. Çalışan yazılımı kullanmak bizim ne kadar hızlı sonuç üretebileceğimizi ve geri bildirim alabileceğimizi ölçmemize izin verir. Bireyler arasındaki sık etkileşim, belgelemeyi en aza indirme durumunun etkilerini telafi eder.[6]

### **2.3 Çevik Yazılım Geliştirme Mimarisi**

Yazılım mimarisi, yazılım sisteminin yapısını ve karakteristiğini tanımlayan kısıtlar kümesidir. [7]

Eğer bir çevik yazılım geliştirme süreci yazılım geliştirme için kullanılıyorsa, geliştirilen yazılım sistemi mutlaka çevik olmalıdır. [7]

Yalnızca değiştirilmesi kolay olan bir yazılım sistemi çevik olacaktır. Bir yazılım sisteminin eksikliği yalnızca düzende ve yapıda ise yazılımın değiştirilmesi kolaydır. Yazılım sisteminde düzen ve yapıyı veren yazılım mimarisidir. Mimarisi olmadan geliştirilen yazılım, kanun ve düzeni olmayan bir toplum gibidir. [7]

Bir uygulamanın sürdürülebilirliğini(ve dolayısıyla çevikliğini) arttıran ve uygulamaya düzen getiren iki şey vardır. Bunlar [7];

- Yazılım Mimarisi (Makro Seviyede)
- Kodlama Standartları (Mikro Seviyede)

Yazılım mimarisi, genellikle uygulama seviyesinde tanımlanır ve tek başına çevik olmalıdır. Ancak kodlama standartları kodu daha okunabilir yapar ve bu yüzden çeviklik

satıcı/topluluk düzeyinde veya organizasyon seviyesinde tanımlanmalı ve yazılım hayat döngüsü içerisinde hiçbir zaman değiştirilememelidir. [7]

Çevik yazılım geliştirme manifestosunun temel prensiplerinden birisi, en iyi mimari ve tasarımın ortaya çıkması için kendi kendine organize olan ekiplerin olması gerekliliğidir. Bu nedenle, çevik proje mimarisi bir kişi tarafından veya bir anda tanımlanmış olmamalıdır. Bu mimari, çapraz fonksiyonel bir takım tarafından tanımlanmalı ve zamanla en iyi mimari haline gelmelidir. [7]

İlk uygulama mimarisi, bir mermi izi gibi dikkatlice seçilmiş kullanıcı hikâyelerinin tamamlanmış bir ürün kalite uygulaması olmalıdır. Bu mimari, bütün takım tarafından topluca geliştirilmelidir ve takım üyelerinin kendi yetenekleri üzerinde erken geribildirim üretilebilmelidir.

Bu nedenle, çevik yazılım geliştirmede “Yazılım Mimarisi” terimi aşağıdaki gibi yeniden tanımlanabilir. [7]

Yazılım mimarisi, uygulamanın çevikliğini sürdürmek ve var olan uygulamanın fonksiyonel ve fonksiyonel olmayan gereklilikleri gerçekleştirebilecek bir yapıya sahip olduğundan emin olmak için kendi geliştirme aktivitesini kontrol etmek üzere, takım tarafından kendine dayatılan kısıtların kümesidir. [7]

## **2.4 Çevik Yazılım Geliştirme Metotları**

Çevik süreçler, yazılım sektöründe kullanılan mevcut geleneksel yöntemlere alternatif olarak geliştirilmiş, modern ve bürokrasiye mesafeli yazılım yöntemlerini ihtiva ederler. Çevik yazılım, bir yandan bir değer sistemini, diğer yandan da somut yazılım metotlarını içerir. Çevik yazılıma, yazılım sektöründe yeni bir filozofik akımı ya da yeni bir yazılım meta modeli olarak bakabiliriz. [8]

### **2.4.1 Extreme Programlama**

Extreme programlama, geleneksel geliştirme modellerindeki uzun geliştirme döngülerinin neden olduğu problemlere karşı çözüm olarak geliştirilmiştir. Extreme programlama, yazılım geliştirme sürecinde daha önceki yıllarda etkili bulunan uygulamalarla birlikte işleri gerçekleştirmek için bir fırsat olarak başlamıştır. [9]

#### **2.4.2 Scrum**

Scrum yaklaşımı, sistem geliştirme sürecini yönetmek için geliştirilmiştir. Scrum, esneklik, uyum ve verimlilik düşüncelerini yeniden tanıtan bir yaklaşımla sonuçlanan sistem geliştirme süreci için endüstriyel süreç kontrol teorisi düşüncelerini uygulayan ampirik bir yaklaşımdır. [9]

#### **2.4.3 Metodolojilerin Kristal Ailesi**

Metodolojilerin kristal ailesi, her bir proje için en uygun metotları seçmek için birkaç farklı alternatif içerir. Metodolojiler, aynı zamanda kristal yaklaşımı farklı projelerin değişen koşullarına uygun hale getiren ilkelere sahiptir. [9]

#### **2.4.4 Özellik Tabanlı Geliştirme**

Özellik tabanlı geliştirme, sistemleri geliştirmek için çevik ve uyarlanabilir bir yaklaşımdır. Özellik tabanlı geliştirme tüm yazılım geliştirme sürecini kapsamaz ve daha ziyade tasarım ve yapı aşamaları üzerinde yoğunlaşır. [9]

#### **2.4.5 RUP Modeli İle Geliştirme**

RUP, nesne yönelimli sistemler için artırılmış bir yaklaşımdır ve gerekliliklerin modellenmesinde ve bir sistem için temel yapılandırılmasında güçlü bir şekilde kullanım durumlarını kapsar. Önerilen geliştirme yöntemi, UML'in özellikle nesne yönelimli geliştirme için uygun olmasına rağmen, diğer metotları ortadan kaldırmaz. [9]

#### **2.4.6 Dinamik Sistem Geliştirme**

Dinamik Sistem Geliştirme Metodu'nun arkasındaki temel düşünce, üründeki işlevsellik miktarını sabitlemek ve ardından sabitlenen işlevselliğe ulaşmak için zaman ve kaynakları ayarlamak yerine, zaman ve kaynakları sabitleyip daha sonra işlevsellik miktarının bu sabitlere göre ayarlanmasıdır. [9]

#### **2.4.7 Adaptif Yazılım Geliştirme**

Adaptif yazılım geliştirme, karmaşık ve geniş sistemlerin geliştirilmesindeki problemler üzerine yoğunlaşmaktadır. Metot, sabit ilk örnek ile artan ve yinelemeli geliştirmeyi güçlü olarak teşvik etmektedir. Temel olarak, adaptif yazılım geliştirme "Kaos kenarında dengeleme" ifadesi ile ilgilidir. [9]

#### **2.4.8 Açık Kaynak Kodlu Yazılım Geliştirme**

Açık kaynak kodlu yazılım geliştirme, kendine has özellikleri olmasına rağmen diğer çevik yazılım geliştirme metotları ile birçok benzerliğe sahiptir. Açık kaynak kodlu yazılım paradigması herhangi bir masraf olmadan kaynak kodun değişiklik ve yeniden dağıtım için serbestçe kullanılabilir olmasını önermektedir. [9]



### MOBİL UYGULAMALAR

Günümüzde, mobil kullanıcıların, mobil operatörlerin, mobil geliştiricilerin ve cep telefonu üreticilerinin mobil uygulama geliştirme süreci için talepleri aşağıdaki gibi listelenebilir[10];

- Mobil kullanıcılar, kendi telefonlarını özelleştirmek için daha fazla seçenek, daha fazla fırsat ve daha fazla işlevsellik talep etmektedirler
- Mobil operatörler, takipçilerine yönetilebilir ve kazançlı bir şekilde katma değerli içerik saptamak istemektedirler.
- Mobil geliştiriciler, kullanıcıları kısıtlamayan güçlü mobil uygulamalar geliştirme özgürlüğü talep etmektedirler.
- Cep telefonu üreticileri, cihazlarını güçlendirmek için istikrarlı, güvenli ve uygun fiyatlı platform istemektedirler.

#### 3.1 Gelişim Süreci

Mobil uygulamalarla ilgili gelişim sürecinin başlangıcı, yirminci yüzyılın sonuna kadar uzanmaktadır. İlk geliştirilen uygulamalar, küçük oyunlar, zil sesi editörleri, hesap makineleri, takvimler ve benzerleri idi. Yeni bin yılın başlangıcı ile mobil içerikler ve mobil uygulamalar hızlı bir pazar evrimi gördü. Üreticiler, daha fazla uygulama üreterek kendi ürünlerini müşteriler için daha çekici hale getirmeye çalıştılar. Akıllı telefonların, işletim sistemleri (Windows Mobile, Symbian, RIM, Android, MAC iOS) ile standart

telefonların işletim sistemlerinde yer almayan 3. Parti uygulama eklemeye izin vermesi, mobil uygulamalarda hızlı bir gelişim başladı. [10]

Apple şirketinin 2007 yılında ilk iPhone telefonunu üretmesi ile mobil uygulama geliştirme ivme kazandı. [11]

Günümüzde, yaklaşık bir milyon uygulamanın Apple ve Android mağazalarında mevcut olduğunu göz önünde bulundurularak yazılım sektöründe mobil uygulamaların etkisinin önemli olduğu söylenilebilir. [11]

### **3.2 Mobil Uygulamaların Özellikleri**

Günümüzde kullanılan mobil uygulamaların temel özellikleri [11]:

- Mobil uygulama terimi, akıllı telefonlar ve diğer taşınabilir cihazlar üzerinde çalışan yazılımları tanımlamak için kullanılmaktadır.
- Mobil uygulamalar, akıllı telefonlar ve kişisel dijital yardımcılar (PDA) için eklenti yazılımlardır.
- Mobil uygulamaların temaları oyunlarda, müzik oynatıcılara ve verimlilik araçlarında değişmektedir.
- Mobil uygulamaların arasında en popüler olanları oyun, sosyal medya, harita, haber, iş, hava ve seyahat bilgisi uygulamalarıdır.
- Tüm mobil uygulamalar cihaza ait iletişim arabirimleri (Wi-fi, WiBro/mobil WiMAX, GSM/EDGE, W-CDMA/UMTS/HSPA ve Bluetooth), ses ve video işlemcileri, kamera, sensörler ve GPS modülü gibi teknik özelliklerden en az birini kullanmaktadır.
- Mobil uygulama kullanıcıları, cihazlarında ortalama olarak 21 yüklü uygulamaya sahiptirler.
- Mobil uygulamaların %25'i ücretlidir ve fiyatları 1 Dolar'dan başlamaktadır.
- Birçok mobil uygulama, ileride üretilecek ücretli uygulamalar için müşteri kazanabilmek adına ücretsiz olarak sunulmaktadır. Diğerleri ise var olan internet sitesi veya hizmete ait müşteriler içindir.

- Mobil uygulamalar, çoğu durumda üçüncü parti geliştiriciler tarafından programlanır. Ama bu uygulamalar ürünün dağıtım, ödeme ve sınırlı pazarlaması ile ilgilenen uygulama marketi sahiplerinin (örneği Apple gibi telefon üreticisinin kendisi) onayına kalırlar. Ve her uygulamanın satış fiyatının %30'unu uygulama marketi sahibine öderler.
- Açık ve birlikte çalışılabilir API'ler için standartların üzerinde odak artışı, mobil uygulamaların başarısına katkı sağlayacaktır.

### **3.3 Mobil Uygulama Çeşitleri**

Mobil uygulamalar, birçok kişi tarafından sosyal medya uygulamaları, oyun uygulamaları, yardımcı uygulamalar, bankacılık uygulamaları gibi çeşitlere ayrılmaktadır. Bu ayrımın yanında, mobil uygulamaları birbirinden ayrılan özelliklerine göre yerli, mobil web tabanlı ve hibrid uygulamalar olarak daha genel ayırarak gruplandıranlar da vardır. [12]

Mobil uygulamaları sınıflandırmanın en basit ve en hızlı yolu olarak, mobil uygulama marketlerinin genel kategorileri dikkate alınabilir. Bu şekilde sınıflandırma, ihtiyaç duyulan uygulamanın bulunması için idealdir. Aşağıda gördüğünüz gruplandırma şekli ise mobil uygulama geliştirme ve mobil girişimler için ilginç olacak şekilde en yaygın türlerin ayrı gruplar halinde toplandığı gruplandırmadır. [13]

#### **3.3.1 Kişisel Yardımcılar**

Kişisel yardımcıları, genelde saat, takvim, notlar, hatırlatıcılar, yapılacakların uygulamaları gibi günlük görevleri yerine getirmek için bir organizatör olarak mobil cihazların temelini oluşturan uygulamalardır. Mobil cihazlar, ek yardımcı uygulamalarla donatıldığında sadece bir cihaz olmanın ötesinde, kişisel bir yardımcı haline gelmektedir.[13]

#### **3.3.2 Coğrafi Konum Destekli Rehberler**

Coğrafi konum, mobil cihazların en çok kullanılan özelliklerinden biridir ve haritalar, navigasyon, hava ve seyahat uygulamaları gibi birçok kullanım alanına sahiptir. Bu uygulamalar, internet bağlantısını kullanarak cihazın coğrafi konum koordinatlarını bulma yöntemi ile çalışırlar. [13]

### **3.3.3 Sosyal Platformlar**

Sosyal ađları kullanmadan hayatı devam ettirmenin zorlařtıđı bu gnlerde, çeřitli sosyal platformlarda toplanma, irtibatta olma ve bilgi paylařımı iin uygundur. En popler sosyal ađlar dahi, mobil uygulamalarla daha derin btnleřme sađlamaktadır [13]

### **3.3.4 Çoklu ortam Araları**

oklu ortam araları, mzik, ses ve resim gibi ođeleri barındırmaktadır. Bu aralar, mzik severler, yaratıcı insanlar, magazin takip edenler ve bunların dıřında hemen herkese hitap etmektedir.

oklu ortam araları kapsamında, grnt yakalama, dzenleme ve izleme, ses ve resim dosyaları ile oynama bunları kaydetme ve sosyal ađlarda paylařmak iin geliřtirilmiř birok uygulama tr vardır. [13]

### **3.3.5 zel zmler**

Uygulama marketleri, ierisinde dađıtıma aılmayan zel uygulama tiplerini iermektedirler. Bunlar, zel uygulamalardır ve i kullanım iindirler. Bu uygulamalar, kurumsal cihazların yetkili kullanıcıları tarafından kullanılabilirler. [13]

### **3.3.6 Marka zmleri**

Markalı uygulamalar, genelde mobil dnyadaki ziyareti kartlarıdır. Bu uygulamalar, ođu zaman markanın logosuna, ismini yakın renk paletine ve belki de slogana sahiptirler. Uygulamalar, alıřırken mobil kullanıcılara markanın tm hizmetlerine eriřme izni verirler. [13]

### **3.3.7 Anlık Mesajlařma Araları**

Telefon grřmelerinin, kısa mesajların ve anlık postanın bařlangıcının ardından iletiřimde birok deđiřiklik meydana gelmiřtir. Mesajlařma seenekleri, genelde diđer uygulama tiplerine eklenmiřtir. Anlık mesajlařma araları, metin, resim, ses ve video iletiřimi ve ses metin giriři gibi zellikleri iermektedirler. [13]

### **3.3.8 Mobil Marketler**

Mobil marketler, internet bađlantısı gerektiren istemci uygulamalardır. Kural olarak, sunucu tarafı burada veriyi depolar, iřler ve daha sonra istemci tarafı ile deđiřtirir. [13]

### **3.3.9 Zaman Geçiriciler**

Oyunlar, en fazla indirme işleminin yapıldığı kategorilerdir. Öyle ki, herkesin kendi beğeni ve tercihlerine göre, zaman geçirmesini sağlayabilecek şekilde alt kategorilerine ayrılmışlardır. Masa oyunları ve kart oyunları, bulmacalar ve eğitsel oyunlar aksiyonlar ve sporlar, stratejiler ve RPG'ler olmak üzere, herkes için her bir alt kategoride birçok alternatif bulunmaktadır. [13]

### **3.3.10 Mobil Eğitimciler**

Çocuklar, mobil cihazlarda oyun oynamayı severler ve mobil cihazlar da özel eğitim uygulamalarını da içerebilirler. Mobil eğitimci uygulamalar, genellikle oyun kategorisi ile ilişkilidir ama burada zaman kaybına yol açacak hiçbir şey yoktur. [13]

### **3.3.11 Eğlence Ve Yaşam Tarzı**

Mobil yüklemeleri, eğlence ve yaşam tarzı bayrağı altında toplayabiliriz. Bunlar, özel kullanıcıların ihtiyaçları için olan spor uygulamaları, rehber uygulamalar, yemek kitabı uygulamaları ve web siteleri, bilimle ilgili uygulamalar, çeşitli kılavuzlar ve referans kitapları, doğaya adanan uygulamalar, tarih, din, moda, sağlık, bahçe kısacası dünya üzerindeki her şeyle ilgili olan uygulamalardır. [13]

## **3.4 Mobil İşletim Sistemleri**

Bir mobil işletim sistemi, akıllı telefon, tablet, PDA veya diğer mobil cihazlarda faaliyet gösteren işletim sistemidir. Modern mobil işletim sistemleri, dokunmatik ekran, cep telefonu, Bluetooth, WiFi, mobil GPS navigasyonu, kamera, video kamera, konuşma tanıma, ses kaydedici, müzik çalar, yakın alanda iletişim ve kızıl ötesi dâhil olmak üzere kişisel bilgisayar işletim sistemlerinin özellikleri ile diğer özellikleri bir araya getirmektedirler. [14]

### **3.4.1 Apple/iPhone**

2007 yılında adının değişmesi ve adından bilgisayar kelimesinin kaldırılması, Apple şirketinin ağırlıklı olarak bilgisayar sektöründe olmaktan, daha geniş medya yakınsama

endüstrisine geçişini sağlayan stratejik değişikliğe gittiğini göstermektedir. iPhone, Apple için medya, internet ve telekom sektörlerindeki yakınlaşmayı sağlamıştır. [15]

Geniş ekranı, iyi 2G/3G ve kablosuz bağlantı gibi yazılım ve donanım yetenekleri, ağ bağlantıları, GPS, modifiyeli Safari tarayıcısı, kamera, hareket sensörü, dokunmatik ekran, internet, medya ve telefon servislerinde kullanıcı dostu ara yüz özellikleri ile iPhone, akıllı telefon pazarında büyük başarı sahibi ve eğilimin merkezi olmayı başarmıştır. [14]

### **3.4.2 Google/Android**

Google, Temmuz 2005'te küçük mobil yazılım geliştirme kampanyası Android Inc.'i satın aldı. Bu satın alma, Google için internet üzerindeki başarılı işlerini genişletmek ve mobil marketi de içermek için yapılan açık bir hareketti. Bu harekette amaç, herhangi bir mobil cihaz üzerinden çalıştırılabilen, mobil telefonlar için uygulama geliştirilebilen, donanımdan bağımsız açık kaynaklı bir işletim sistem geliştirmek olmuştur. Daha önce bahsedildiği gibi, Android ve iPhone platformları arasındaki önemli fark, Android'in özel bir donanım sistemine ait olmak için herhangi bir girişimi olmaması ve mobil telefon üzerinde ortak donanım tiplerinin birçoğu için sürücüler sunmasıdır. [14]

### **3.4.3 Nokia / OVI**

Nokia da, Apple ve Google gibi kendi 'Uygulama Marketi'/'Pazar Yeri'ni oluşturarak OVI ismini vermiştir. OVI mağazası, Nokia'nın Symbian cihazları tarafından erişilebilmektedir. Mağaza, Apple ve Google mağazalarına nispeten yenidir. Bu durum, OVI marketten ulaşılabilen uygulama miktarı, iPhone ve Android marketleri ile karşılaştırıldığında sınırlı miktarda kalmasından anlaşılabilir. [14]

### **3.4.4 Blackberry**

Blackberry de, 'App World' isimli bir uygulama mağazası oluşturmuştur. App World, biraz uygulamaya sahip olmakla birlikte iPhone'un içerdiği miktarda uygulama içermekten uzaktır. 'App World' içerisinde yer alan uygulamalar, hava durumunu öğrenme, var olan web servislerine erişme gibi grafik kısmı üzerinde daha az yoğunlaşan geleneksel uygulamalardır. [14]

### **3.4.5 Windows Mobile**

Windows, 2009 yılı Ekim ayında Windows Mobile (sürüm 6.5) yeni güncelleştirme tanıtımının yanı sıra, Windows Marketplace isimli yeni bir uygulama mağazasına erişimi de başlatmıştır. Markette mevcut olan uygulamaların miktarı, sınırlıdır ve uygulamaların sahip olduğu işlevsellikler iPhone ve Android uygulamaları gibi yaratıcı ve fantastik değildir. Windows Marketplace dışında, Windows Mobile uygulamalarını içeren uygulama mağazaları da vardır.[14]

### MOBİL UYGULAMALARDA ÇEVİK YAZILIM GELİŞTİRME

Mobil uygulama karakteristikleri ile çevik yazılım geliştirme metotlarının karakteristikleri karşılaştırılmaya çalışıldığında, çevik yazılım geliştirme metotlarının sınırlarının iyice belirlenmemiş olmasından kaynaklanan zorluklar oluşmaktadır. Yazarların bu konudaki çalışmaları dört bölüme ayrılmaktadırlar. Bu çalışmalar [14];

- Bilgilendirme ve adaptasyon
- İnsan ve toplum faktörü
- Çevik yazılım geliştirme metotlarının bakış açısı
- Karşılaştırmalı çalışmalar

Bulgular, çevik yöntemlerin tanıtımının, özellikle yazılımda çevik uygulamaların geleneksel uygulamalarla tamamen yer değiştirmedeği ve geleneksel metotlarla birlikte çalıştığı durumlarda verimliliğe faydası olduğunu göstermektedir [14].

Mobil uygulama geliştirmede geleneksel plan tabanlı yazılım geliştirme yöntemleri yerine çevik yazılım geliştirme yöntemleri uygulandığında görünen olumlu etkiler, aşağıdaki gibi listelenebilir:

1. Süreçteki görünürlüğün artması
2. Teknik sorunlara erken tanı
3. Sorumluluk paylaşımı
4. Verimli bilgi paylaşımı



5. Düşük hata yoğunluğu
6. Sürekli geliştirme temposu

#### **4.1 İlgili Çalışmalar**

Rational Unified Process yöntemi, pratik bakış açısı görüşü ile 2003 yılında ortaya konmuştur. 2004 yılında, yeni bir yazılım geliştirme metodu olarak mobil uygulamalar için özelleştirilmiş olarak Mobile-D, geliştirilmiştir. Mobile-D yöntemi, Extreme programlama, Kristal metot ve diğerlerine göre daha ağır bir metot olan Rational Unified Process gibi iyi kurulmuş olan çevik metotlara ait elemanlar içerir ve çevik uygulamalara dayanmaktadır. [15]

Mobile-D ile ilişkili uygulamalar, test odaklı geliştirme, çiftli programlama, sürekli birleştirme, yeniden düzenleme gibi yazılım süreci iyileştirme görevleridir. Mobile-D metodu, diğer metotlar için bir temel olarak hizmet vermiştir. [15]

#### **4.2 Kullanılan Çevik Yazılım Geliştirme Metodolojileri**

Mobil bilgisayarlardaki gelişmeler, cep telefonlarından internet kullanımının yaygınlaşması ve Java ME, Symbian, iOS ve Android gibi işletim sistemleri ile mobil uygulama geliştirme alanında yürütülen proje sayısında önemli artış meydana gelmiştir. Var olan çevik yazılım geliştirme metotları, mobil platformlar için bu yöntemlerin nasıl uygulanabileceğine özel olarak değinmediği için mobil uygulama geliştirme platformlarının kendine özgü koşulları dikkate alınarak çevik yazılım geliştirme metotları geliştirilmiştir. [15]

Çevik yazılım geliştirme metotlarının, mobil uygulama geliştirmedeki katkıları aşağıdaki gibidir [16];

- Uygulamaların sürekli kullanımını ve dayanıklılığını artırma
- Çok hızlı değişen mobil uygulamaların yapısına adapte olmayı hızlandırma
- Risk yönetimi için daha gelişmiş öngörüye sahip olmayı mümkün kılma
- Mobil uygulama üretimindeki verimliliği artırma
- Mobil uygulamalarda platform kısıtlarından kaynaklanan sorunları azaltma

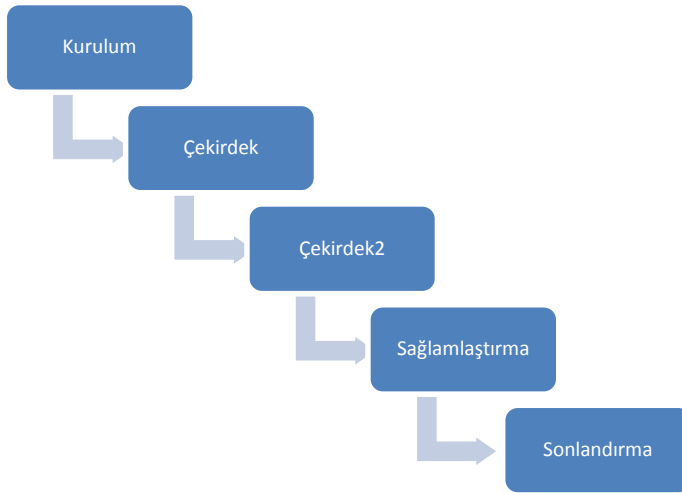
Mobil uygulama geliřtirmede kullanılabilircek evik yntemler, ařađıdaki gibi sıralanabilir [14]:

- Mobile-D [18]
- Hibrid Metod Mhendisliđi Yaklařımı [1]
- Scrum [3]
- MASAM [2]

#### 4.2.1 Mobile-D

Mobile-D, diđer modellerden nce geliřtirilen bir model olmakla birlikte diđer modeller iin referans olmuřtur. Mobile-D'den sonra geliřtirilen modeller, Mobil-D'nin ayrıntılı analizi yapılıp belli zellikleri geliřtirilerek retilmiřtir. retilen modeller iin test ve eriřim ortamlarının oluřturulmuř olduđu gzlemlenmiřtir. [18]

Mobile-D yaklařımını takip eden bir geliřtirme projesi beř ařamaya blnmektedir [18]



řekil 4.1 Mobile-D yaklařımındaki ařamalar

Bu yntemdeki her ařama  tr geliřtirme gnne sahiptir. [18]



řekil 4.2 Mobile-D Geliřtirme Gnleri

Mobile-D'de aynı rnn farklı paraları zerinde aynı anda birden fazla takım alıřıyorsa ayrıca bir btnleřtirme gnne ihtiya duyulmaktadır. Bu ařamaların

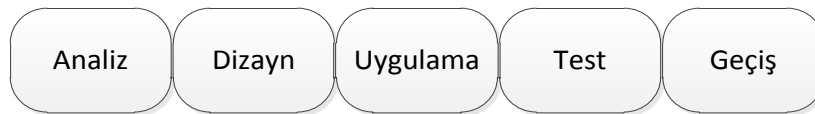
uygulamalarında dokuz temel bileşen yer almaktadır. Bu bileşenler aşağıdaki gibi sıralanabilir [18]:

1. Aşamalama ve yürütme
2. Mimari Çizgi
3. Mobil Test Temelli Geliştirme
4. Devamlı Birleştirme
5. Eşli Programlama
6. Ölçütler
7. Çevik Yazılım Geliştirme Süreci
8. Şirket Dışı Müşteri
9. Kullanıcı Merkezli Odak

#### 4.2.2 Hibrid Metod Mühendisliği

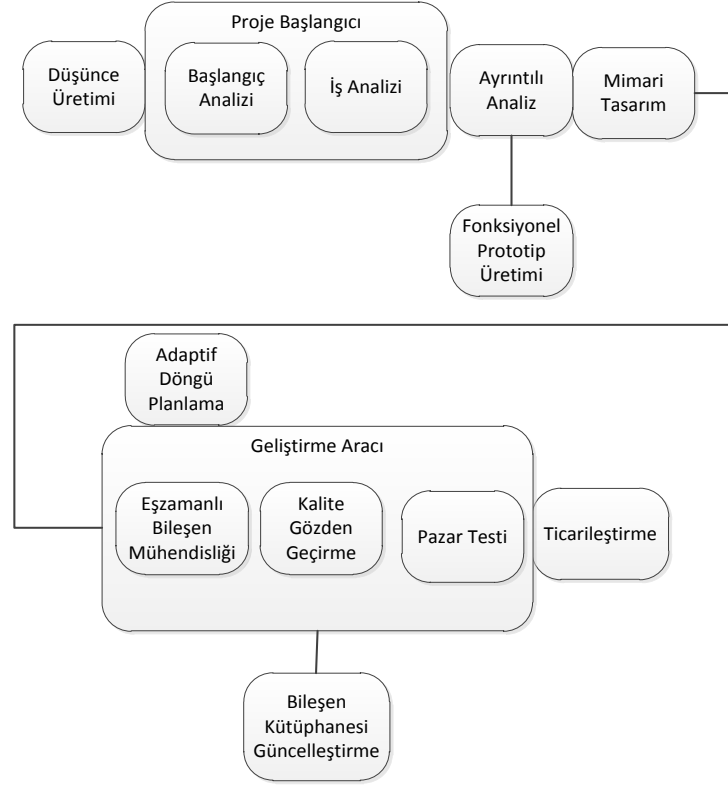
Hibrid Metod Mühendisliği yöntemi geliştirilirken, mobil uygulama geliştirme metotları incelenip, mobil uygulama karakteristiği gözden geçirilmiş ve bu işlemlerden sonra mobil uygulama geliştirmedeki zorluklar tespit edilmiştir. [1]

Hibrid Metod Mühendisliği yöntemi, genel yazılım geliştirme hayat döngüsüne ait modeli alıp, 3 farklı iterasyondan geçirmektedir. [1]



Şekil 4.3 Hibrid Metod Mühendisliği Genel Yazılım Geliştirme Hayat Döngüsü

Bu iterasyonlar sonucunda mobil uygulama geliştirme için yöntemin oluşturduğu final model aşağıdaki şekildeki gibi ortaya konmuştur: [1]



Şekil 4.4 Hibrid Metod Mühendisliği İle Geliştirilen Final Yöntem

#### 4.2.3 Scrum

Mobil uygulamalar için Scrum, mobil uygulama geliştirme projelerinde anında öğrenip geliştirmeyi sağlayacak şekilde Scrum metodunun mobil uygulama geliştirmede kullanımı incelenmektedir. [3]

Scrum yöntemi ile geliştirilen uygulamanın testi, gerçek bir proje sahibi ve gerçek bir Scrum uzmanı ile gerçekleştirilmiştir. [3]

Bu yöntemde, Scrum metodu kullanılarak kısa sürede başarılı bir ürün gerçekleştirilmesi hedeflenmiştir ve bu yöntem üzerinde yapılan çalışmaların sonucunda bu hedefin gerçekleştirilebileceğine yönelik hipotez doğrulanmıştır. [3]

Mobil uygulamalarda Scrum yönteminin başarısındaki anahtar etkenler aşağıdaki gibi verilmiştir [3]:

- Güçlü bağlılık ifadesi (göreve bağlılık)
- Planlanmış görevlere odaklanma

Scrum yöntemi ile gerçekleştirilen çalışmaların bir sonraki aşamasında farklı ülkelere dağılmış geliştiriciler kullanılarak nasıl bir sonuç elde edileceğini gözlemlemek hedeflenmektedir. [3]

Mobil uygulamalarda Scrum yazılım geliştirme metodunun uygulanmasını kolaylaştırmak için araçlardan faydalanılmış ve bu araçların özellikleri geliştirilen örnek uygulamalar yardımı ile ortaya konulmuştur. Aracın özelliklerine ve geliştirilen örnek uygulamaya ilerleyen bölümlerde değinilecektir.

#### **4.2.4 Masam**

MASAM, Kore'deki bir mobil uygulama geliştirme şirketinde uygulanan bir yöntemdir. MASAM, Mobile-D'den sonra geliştirilmiştir ve MASAM metodu[2] Eclipse Süreç Çerçevesi [19] olarak tanımlanmış ve dağıtımına açılmıştır.

MASAM, Mobile-D ile benzer olarak klasik yaşam çevrimi veya şelale metodu yerine çevik yaklaşımları önermektedir. [2]

MASAM standart süreci, 4 alt aşamadan oluşmaktadır. Bu adımlar aşağıdaki gibi listelenebilir: [2]

- Geliştirme hazırlık aşaması
- Somutlaştırma aşaması
- Ürün geliştirme aşaması
- Ticarileştirme aşaması

#### **4.3 Mobil Uygulamalarda Çevik Yazılım Geliştirmenin Farklılıkları**

Çevik yazılım geliştirme metodu karakteristikleri ile mobil uygulama özellikleri arasındaki doğrudan karşılaştırma, çevresel değişkenlik, üretilen belgelerin miktarı, ilgili planların miktarı, geliştirme takımının büyüklüğü, uygulamanın geliştirilmekte olan miktarı, müşteri kimliği ve nesne yönelimi özellikleri üzerine yoğunlaşarak Abrahamsson tarafından gerçekleştirilmiştir. Karşılaştırma sonucunda, müşteri tanımının dışında kalan tüm diğer çevik karakteristikler mobil uygulama geliştirme için uygun çıkmıştır.[15]

Mobil uygulamalarda çevik yazılım geliştirme yöntemi uygulandığında ortaya çıkan durumları ve elde edilen sonuçları inceleyebilmek için çalışmalar gerçekleştirilmiş ve bu çalışmaların sonuçları geleneksel plan tabanlı yazılım geliştirme yöntemleri ile kıyaslanmıştır. [17]

Mobil uygulamalarda çevik yazılım geliştirme yöntemleri uygulanırken karşılaşılan ve gelecek araştırmalara yön verecek zorluklar [17]:

- Platforma özgü geliştirme yeteneği gereksinimi
- Mobil cihazlardaki çeşitliliğin getirdiği taşınabilirlik sorunu
- Mobil uygulamalar için test tabanlı geliştirme araçlarının olmaması
- Sık yayın çevrimine alışma
- Çevik metotların değiştirilmeden kullanılmasının yarattığı sorunlar

Mobil uygulama geliştirmede diğer uygulamalardan farklı olarak çevik proje geliştirme metodunu kullanmayı gerekli kılan bazı kısıtlar ve gereksinimler vardır. Bunlar:

- Mobil uygulama geliştirmede istekler ve platformlar çok sık değişebilmesi [3]
- Mobil uygulamalarda geliştirilen ürünün pazara hızlı çıkma zorunluluğunun olması [3]
- Mobil yazılım geliştirmede genelde küçük uygulama geliştirme ekiplerinin yer alması [1]
- Mobil uygulamaların boyutlarının genelde 10.000 satırdan daha az olması [1]

#### **4.4 Yazılım Geliştirme Süreci Yönetim Uygulamaları**

Günümüzde, çevik yazılım geliştirme metodunun uygulanmasını kolaylaştırmak için süreç yönetim ve kontrol aracı kullanmanın gerekliliği artmaktadır. Süreç yönetim aracı kullanımı ile, kolay yönetim, yüksek performans ve daha profesyonel bir yapı sağlanmaktadır.

Süreç yönetim araçları ile ilgili gerçekleştirdiğimiz ilk yayın çalışmalarında, Trello [21] ve Telerik TeamPulse [22] araçlarının kullanımı, özellikleri ve faydaları geliştirilen örnek uygulamalar yardımı ile ortaya konulmuştur. İlerleyen çalışmalarda ise mobil uygulama

geliştiriciler tarafından sıkça kullanılan Jira [23], Microsoft Team Foundation Server [24] ve Eclipse EPF [19] araçlarının kullanımı ve özellikleri incelenmiştir. [20]

Süreç yönetim araçları, geliştirilen tüm projelerin, tüm görevlerin, tüm takımın iyi organize edilebildiği ve her üyenin ayrı hesap ve yetkilere sahip olduğu kolay kullanıma sahip olduğu araçlardır. Süreç yönetim araçları, mobil uygulama geliştirmede çevik yöntemin kullanılmasında da önemli faydalar sağlamaktadır. [20]

Mobil uygulama geliştiricilere göre var olan yazılımlarda en fazla giderilmesi gereken eksiklikler, fikir ve geri besleme, raporlama, izleme ve analiz bölümlerindeki eksikliklerdir. [20]

Mobil uygulama geliştiricilerin yeni geliştirilecek araçla ilgili görüşlerini almak için hazırlanan anket sorularına verilen cevaplara göre kullanıcı hikâyelerinden oluşma ve iş değeri içermesi en önemli iki araç özelliği olarak öne çıkarken, önceliklendirme ve ürün sahibi için katkı yetkisi de diğer iki önemli özellik olarak öne çıkmaktadır. [20]

Mobil uygulamalarda çevik yazılım geliştirme sürecinin yönetimini kolaylaştırıcı bir araç kullanmanın faydaları aşağıdaki gibidir [16];

- Daha etkili kararlar vermeye yardımcı olma
- Düşüncelerin, kullanıcı hikâyelerinin ve farklı kaynaklardan geribildirimlerin analizini ve görüntülenmesini kolaylaştırma
- Devam eden geliştirme süreci içerisinde planlamanın ve sonuç ölçümünün yapabilme
- Uygulamanın iş akışını yönetmeyi kolaylaştırma ve her yerden proje üyeleri arasındaki iletişimi mümkün kılma
- Projenin durum analizini kolaylaştırma

#### **4.4.1 Uygulamaların Özellikleri**

Yazılım geliştirme süreci yönetim uygulamaları ile ilgili incelemelere geçilmeden önce kullanılan uygulamalarda yaygın olan özellikler hakkında bilgi sahibi olunması, uygulamaların anlaşılmasını kolaylaştıracaktır. Uygulamaların yaygın özellikleri, aşağıda alt başlıklar halinde sunulmuştur.

#### **4.4.1.1 Proje Planlama Ve Zamanlama**

Proje planlaması, projenin bütün yönlerini kapsayan ve projenin uygulanmasının temeli olan bölümdür. [25] Proje yönetim planı ise, projenin nasıl yürütüleceğini, izleneceğini, kontrol edileceğini ve kapatılacağını belirler. [26] Planlamanın mükemmelliği ve yetersizliği, projenin başarısını ve başarısızlığını doğrudan etkiler. [25]

Zaman yönetimi, projenin zamanında bitirilmesini sağlayacak süreçlerin tanımlandığı bölümdür. [25] Proje zaman yönetimi ise, projenin zamanında tanımlanabilmesi için gerekli süreçlerden oluşur. [26] Zaman yönetimi, işlerin tanımı; süre tahminleri, Gannt şeması vb. zaman çizelgelerinin geliştirilmesi ve denetiminden oluşur. [25]

#### **4.4.1.2 Raporlama**

Raporlama, proje sürecinin her aşamasını kayda geçirme ve belgeleme işlemidir. Raporlarda, öncelikle planlanmış olana karşılık gerçekleştirilmiş olan ilerleme (performans) değerlendirilmeli ve bilgiler az ve öz bir şekilde sunulmalıdır. [25]

#### **4.4.1.3 İzin Ve Çalışma Günleri**

Başarılı bir projenin en temel özelliklerinden biri, zaman kaybının olmamasıdır. Kullanılan araçlar, iş akışını desteklemeye ve kolayca zaman raporlamalarına izin vermektedir. Zaman çizelgesi ekranı, kullanıcının yer aldığı tüm projeler dâhilindeki iş kalemleri için zaman girebilmelerine izin vermektedir. [22]

#### **4.4.1.4 İzleme Ve Analiz**

İzleme ve analiz bölümleri, takım performansı, projenin zamanına uygun olarak devam edip etmediği, projenin içeriğinde değişiklik olup olmadığı, işlerin gecikmesi durumu, proje üyelerinin geride bırakılan zaman dilimi içerisinde üzerinde çalıştığı işlerin gözlemlenmesi, bir durum içerisinde fazla iş yığılımı olup olmadığının gözlemlenmesi ve sonuçların analiz edilmesi gibi durumlar için kullanılmaktadır. [22] Projenin sürekli olarak izlenmesi, proje ekibinin projenin ne kadar sağlıklı yürüdüğünü ve özellikle dikkat edilmesi gereken alanları saptayabilmesini sağlamaktadır. [26]



#### **4.4.1.5 Fikirler Ve Geri Besleme**

Fikir ve geri besleme bölümleri, kullanıcıların proje içerisinde fikirlerini bildirmelerine ve verilen iş veya atamalarla ilgili cevaplarını sunmalarına imkân verirler. İçerdikleri oylama, yorumlama ve özel mesajlaşma gibi özelliklerle proje içerisinde etkin etkileşimin sağlanmasında önemli katkı sağlarlar. [22]

#### **4.4.1.6 Çevik Yazılım Geliştirme (Scrum) Özellikleri**

Günümüzde birçok yazılım geliştirme süreci yönetim uygulaması, çevik yazılım geliştirme sürecinin kişilerine ait rollere (Proje Sahibi, Scrum Master, Scrum Takımı) ve çevik metodun Sprint iş yığını, proje iş yığını, kullanıcı senaryoları durumu özelliklerine ve işlerin ilerleme süreçlerine ait özellikleri (tamamlanmış, devam eden ve başlanacak) de içermektedirler.

#### **4.4.1.7 Ekip Üyeleri Yönetimi**

Yazılım mühendisliği literatüründe en sık rastlanan cümlelerden biri, “Yazılım geliştirme bir takım sporudur” cümlesidir. Yetkinlik, beceri ve geliştirme takımı üyelerinin sorumlulukları dikkate alınarak ekip üyelerine görev ataması yapılmalıdır. Rollerin atanması ile yazılım geliştiricilerin kaynak koddan sorumlu olması, sistem analistinin kullanım durumu diyagramlarından sorumlu olması gibi çalışma ürünlerinin özel tipleri için sorumluluk paylaşımı yapılabilir. [23]

#### **4.4.2 Uygulamaların İncelenmesi**

Çevik yazılım proje geliştirme metotlarını uygulamayı kolaylaştırmak için çeşitli araçlardan faydalanılmaktadır. Bu araçlardan bazıları aşağıdaki gibi listelenebilir:

- Trello [21]
- Telerik TeamPulse [22]
- Atlassian Jira [23]
- Microsoft Team Foundation Server [24]
- Eclipse EPF [19]

Tez çalışması kapsamında yukarıda listelenen araçlar incelenmiş ve incelenen araçlardan bazılarının basit ara yüz ve az bileşenle işlemlerini gerçekleştirirken bazı

araçların ise daha karmaşık bir ara yüz ile kapsamlı işlevler gerçekleştirdiği gözlemlenmiştir.

#### **4.4.2.1 Telerik TeamPulse**

Telerik TeamPulse, çevik yazılım geliştirme sürecine ait tüm özellikleri bünyesinde toplamaya çalışan bir proje yönetim aracıdır. Sürekli olarak ekibi birbiri ile bağlı tutarak, gereksinimleri ve problemleri yönetme özelliği, planların gerçekleştirilmesini ve ilerlemeyi izlemeyi sağlar. Scrum, Kanban veya farklı bir çevik yazılım geliştirme metodunu kullanan tüm ekiplerin ihtiyaçlarını karşılamayı hedefleyen Telerik TeamPulse ve Microsoft Team Foundation Server aracı ile entegrasyon özelliğine sahiptir. [22]

Telerik TeamPulse aracının öne çıkan özellikleri, aşağıdaki gibi listelenebilir;

- Planlama
- Analiz
- Gereksinim Yönetimi
- Çoklu Proje Yönetimi
- Hata Yönetimi
- Zaman Takibi
- Çevik Sürece Adaptasyonu Hızlandırma
- Takım İşbirliği
- Aracı Özelleştirme
- Müşteri Geribildirimleri için Portal
- Otomatik test çözümü

Tez çalışması kapsamında, Telerik Team Pulse aracı ile mobil uygulama geliştirmede Scrum çevik yazılım geliştirme metodunun uygulanmasını gözlemlemek ve gözlemlerin sonucunda elde edilen sonuçların incelenmesi için “Çevik Etkinlik” isimli örnek bir mobil uygulama projesi geliştirilmiştir. Geliştirilen projenin kullanım senaryoları ve görev listesi Telerik TeamPulse uygulaması ile yönetilmiştir. Bu araç sayesinde, çevik

yazılım geliştirme yönteminin gerektirdiği süreçlerin adımları tanımlanmış ve adımların gerçekleştirilme durumu takip edilebilmiştir. [16]

Mobil uygulamanın geliştirilmesi sürecinde, Android işletim sistemi üzerinde 2 uygulama geliştiricisi ve 1 proje yöneticisi görev almıştır. Geliştirme sırasında, uygulama 2 ana geliştirme evresine(Sprint) ayrılmıştır ve uygulamanın 1. geliştirme evresi(Sprint) tamamlanarak ve 2. geliştirme evresine(Sprint) başlanıldığı anda uygulamaya ve Telerik TeamPulse aracına ait ekran görüntüleri aşağıda paylaşılmıştır. [16]

Uygulamada görev alan kullanıcılara ait görevlerin hangi aşamada olduğu, Telerik TeamPulse’da her bir çevrim sürecinin ilerleme durumu ile görülecek durumda Şekil 4.5’te yer almaktadır. [16]

Name	Tasks	Priority	Status	Assigned To	Priority Class.
Temel randevu veritabanı	5	1	Resolved	Ismail Duru	1 - Must Have
Randevu listeleme	4	2	Resolved	Erdem Köse	1 - Must Have
Randevu silme	2	3	Resolved	Erdem Köse	1 - Must Have
Randevuya kişi ekleme	3	4	Resolved	Ismail Duru	2 - Should Have
Randevu hatırlatma	3	5	Resolved	Erdem Köse	2 - Should Have
Randevu düzenleme	3	6	Active	Erdem Köse	1 - Must Have
Randevu yerinin belirlenmesi	5	7	Active	Ismail Duru	3 - Could Have
Randevu ek zamanlama ayarları	4	8	New	Ismail Duru	2 - Should Have
Randevu not ve resim bilgileri	6	9	New	Erdem Köse	3 - Could Have

Şekil 4.5 Telerik TeamPulse Kullanıcı Görevleri Durumu

Uygulamanın 2. geliştirme evresinde görev alan 1 uygulama geliştiricinin 2 farklı kullanıcı hikâyesine ait görevlerinin ilerleme aşamasının Telerik TeamPulse programının görev tahtası ekranındaki durumu, Şekil 4.6’da görülmektedir. [16]

Parent	New	Active	Closed
<p>▼ Erdem Köse (2 Items)</p> <p>▲ Ismail Duru (2 Items)</p> <p><b>Ismail Duru</b></p> <p>Randevu yerinin belirlenmesi</p> <p>Estimate: none</p> <p>Status: Active</p> <p>Add a New Task</p> <p><b>Ismail Duru</b></p> <p>Randevu ek zamanlama ayarları</p> <p>Estimate: none</p> <p>Status: New</p> <p>Add a New Task</p>	<p>▲ Add a New Story ▼</p> <p>▲ Add a New Story ▼</p> <p><b>Ismail Duru</b> PERT</p> <p>Randevu sınıfının yer bilgisi tutacak şekilde güncellenmesi</p> <p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyonu test edilmesi</p> <p><b>Ismail Duru</b> PERT</p> <p>Tekrar bilgisinin nasıl yönetileceğinin araştırılması</p> <p><b>Ismail Duru</b> PERT</p> <p>Randevu veritabanının bitiş tarihi ve tekrar bilgisini tutacak şekilde güncellenmesi</p> <p><b>Ismail Duru</b> PERT</p> <p>Randevu görüntüleme, ekleme ve düzenleme ekranının güncellenmesi</p> <p><b>Ismail Duru</b> PERT</p> <p>Randevu ek zamanlama ayarlarının test edilmesi</p>	<p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyonu kodlanması</p> <p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyon imkanlarının araştırılması</p> <p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyon analizi</p>	<p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyon imkanlarının araştırılması</p> <p><b>Ismail Duru</b> PERT</p> <p>Google Maps entegrasyon analizi</p>

Şekil 4.6 Telerik TeamPulse 2. Uygulama Geliştirme Evresi Görev Tahtası

Uygulamanın randevu ekleme, randevu listeleme ve randevuya kişi ekleme bölümlerinin Android işletim sistemi üzerindeki ekran görüntüleri Şekil 4.7’de yer almaktadır. [16]



Şekil 4.7 İlk Sprint Sonucunda Elde Edilen Yazılımın Ekran Görüntüsü

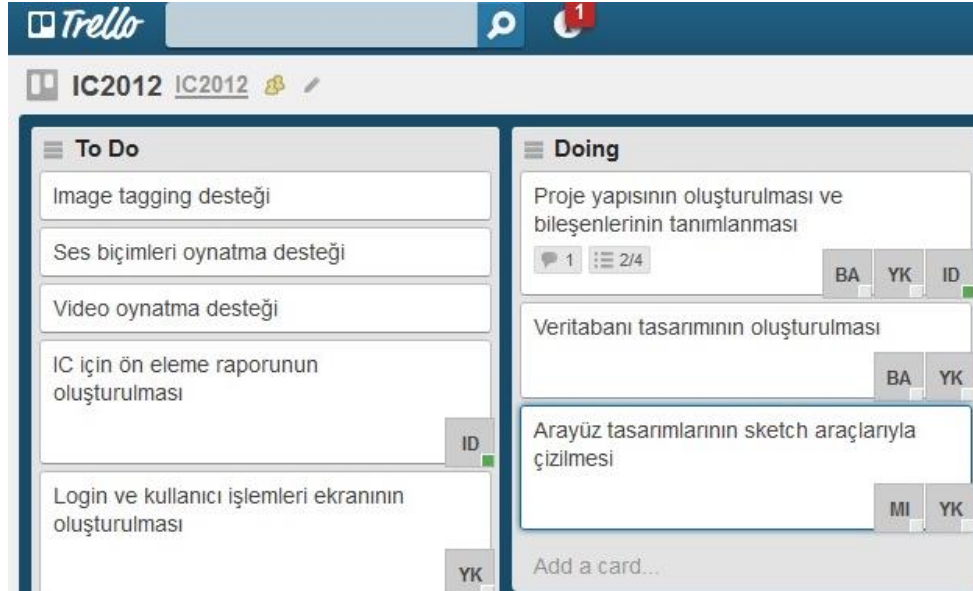
Uygulama geliştirme süreci sonucunda, çevik metodun, mobil uygulama geliştirme sürecinin kısıt ve gereksinimler nedeni ile karşılaşılan problemlere çözüm getirdiği ve

Telerik TeamPulse aracının mobil süreçleri yönetmede faydalı bir araç olduğu sonucuna varılmıştır. [16]

#### 4.4.2.2 Trello

Trello, çevik yazılım geliştirme metodu ile geliştirilen tüm projelerin, tüm görevlerin, tüm takımın iyi organize edilebildiği ve her üyenin ayrı hesap ve yetkilere sahip olduğu kolay kullanıma sahip bir araçtır. Aynı zamanda, yeni kullanıcıların kolayca anlayabileceği liste ve kart özelliklerine ve kullanıcıların dakikalar içinde program üzerinde çalışmaya başlayabilecekleri basit ara yüze sahiptir. [21]

Tez sürecinde, Trello aracının kullanıldığı örnek bir uygulamaya ait ekran görüntüsü Şekil 4.82’de yer almaktadır. [17]



Şekil 4.8 Trello Aracı İle Çevik Yazılım Proje Yönetimi

Trello aracının alt bölümleri ve bölümlerin kullanım amaçları aşağıdaki gibi listelenebilir [21]:

- **Sahneler:** Sahne, listelerin tamamını içeren bölümdür. Genellikle her bir proje için ayrı sahneler kullanılmaktadır. Sahnelerin kullanımı çok kolaydır ve sahneler kullanılarak proje durumu kolayca yönetilebilir.
- **Ortak Sahneler:** Ortak sahneler, müşterilere yol haritası ve o anda sahip olunan özellikler hakkında bilgi vermek amacı ile kullanılır.

- **Listeler:** Listeler, yalnızca basit listelerden oluşabileceği gibi süreçteki bir aşamayı gösterecek şekilde daha güçlü halde de kullanılabilirler. Listeler basitçe sürüklenme yöntemi ile iş akışına yeni bir görünüm kazandırılabilceği gibi liste üzerindeki bir kart başka bir listeye taşınarak süreç içindeki ilerleme de ortaya konulabilir.
- **Kartlar:** Kartlar, görevlere karşılık gelmektedir. Takımın yapmaya ihtiyaç duyduğu şeyleri izlemek için kart oluşturabileceğiniz gibi kartlar, dosya ekleme, video gömme, kullanıcı atama, tarih ekleme, kontrol listesi ekleme veya yalnızca yapılacak işleri gösterecek şekilde de kullanılabilir.
- **Üyeler ve Organizasyonlar:** Kullanıcılar, sahneler üzerinde yetkileri kapsamında kontrole sahiptir. Tüm şirket ve takım görebileceği şekilde sahneler oluşturulabileceği gibi sahneler yalnızca bazı kişilerin görebileceği şekilde de özelleştirilebilir. Sahneleri tüm şirket kullanıcılarının tek davetle uygulamaya dâhil edilebilmesine imkân vermektedir.
- **Bildirimler:** Trello, kullanıcılara yorum aldıklarında, atandıkları kartlar üzerinde herhangi bir değişiklik olduğunda veya başka bir sahne için davet gönderdiklerinde, bilgi güncelleme sütunu ile anlık olarak neler olduğunu bildirimlerle haber verir.
- **Tüm Cihazlar Üzerinde Çalışma:** Trello, akıllı telefonlar, tabletler ve dizüstü bilgisayarlar gibi farklı araçlar üzerinde çalıştırılabilir.

#### 4.4.2.3 Atlassian Jira

Jira, merkezi bir sunucu üzerinde çalışan ve herhangi bir bilgisayar üzerindeki web tarayıcıdan erişilebilen bir uygulamadır. [27]

Jira uygulamasının gereksinimleri ve bu gereksinimlere ihtiyaç duyulan yerler, aşağıdaki gibi listelenebilir: [27]

- Web Tarayıcıları (istemci tarafı)
- Java platformları (JDK/JRE) (sunucu tarafı)
- İşletim sistemi (sunucu tarafı)

- Uygulama sunucuları (sunucu tarafı -JIRA WAR-EAR kullanılıyorsa)
- Veri tabanları (sunucu tarafı)

Atlassian firması tarafından geliştirilen Jira uygulaması, web tabanlı çalışan iş, hata takip ve proje yönetim aracıdır. Hata takibinde başarılı bir konuma ulaşan uygulama, Java platformu üzerinde çalışan açık kaynak kodlu bir yazılımdır. [27]

Hızlı proje yönetimi ve kolay iş takibi yapılabilmesi, JIRA aracının en önemli özelliklerindedir. Değiştirilebilir özel filtreler, ayarlanabilir iş takip grafikleri ve otomatik kullanıcı bilgilendirme e-postaları da Jira aracının sahip olduğu önemli özellikleri arasındadır. Profesyonel hata takibi özelliğine sahip olan Jira aracı, yüzlerce karışık sorunu ve çözümlerini düzenli bir şekilde kullanıcılara sunmaya imkân vermektedir. Talep ve alan oluşturulabilmesi gibi geniş kullanımlı özellikleri, kurumsal ihtiyaçlara göre şekillenebilirken raporlama ve analiz özellikleri de kullanıcıların pratikliğini arttırmaktadır. [27]

Jira, geniş eklenti özelliklerine de sahiptir. Ücretli ve ücretsiz eklentileri ile sahip olduğu genişletilebilir kullanım özelliği, Jira aracının en güçlü yanlarından birisidir. Eklentiler içerisinde yer alan Türkçe dil paketi kullanıcılara ücretsiz indirme ve kurma imkânı sağlamaktadır. [27]

JIRA uygulamasını anlayabilmek için önemli olan temel modülleri ve bu modüllerin özellikleri aşağıdaki gibi listelenebilir: [27]

- **Talepler (Issues):** Bir Jira projesi, talepler topluluğudur ve talepler, JIRA aracının temel yapıtaşını oluşturmaktadır. Talep oluşturulurken standartlar kullanılabileceği gibi kişinin kendi talep tipini de oluşturması da mümkündür. Jira, farklı organizasyonlarda değişik şekillerde kullanılmaktadır. Firmaların istekleri doğrultusunda bir talep, herhangi bir yazılım hatası, proje görevi, tatil başvurusu vb. olabilmektedir. Talepler, eş zamansız olarak cevaplanabilmekte veya diğer kullanıcılara iletilebilmektedir. Bir veri tabanının kullanımı da taleplerin gözden kaçırılmamasını-kaybolmamasını sağlamaktadır.
- **İş Akışları (Workflows):** İş akışları, işlerin ilerleyişini takip etme olanağını görsele dökmeye yarayan bir özelliktir. Jira aracında iş akışlarının istenildiği gibi

tasarlanması imkânını da sunulmaktadır. Jira'da özel iş akışları oluşturulabilmektedir. Jira aracının "İş Akışı Tasarlayıcısı" eklentisi, hızlı bir şekilde yeni bir görünüm oluşturulabilmesini ve oluşturulan iş akışlarının görüntülenebilmesini sağlamaktadır.

- **Alanlar (Fields):** Jira'da var olan standart alanlar kullanılabileceği gibi, kendi alanlarını oluşturup, o alanları kullanma imkânı da vardır.
- **Ekranlar (Screens):** Kendi alanlarınızla kendi ekran tasarımlarınızı yapmanızın mümkün olduğu Jira'da, süreç içerisinde karşılaşılan ekranlar esnek bir şekilde oluşturulabilmektedir.

#### 4.4.2.4 Microsoft Team Foundation Server

Microsoft Team Foundation Server, Microsoft'un Birlikte Çalışma Sunucusu(Team Collaboration Server) uygulamasıdır. Temel olarak, yazılım geliştirme üzerine çalışan ekiplerin birlikte çalışmalarını sağlamak-kolaylaştırmak ve bu çalışmalarını takip etmekle işlerin bir kısmını otomatik hale getirmek gibi amaçlarla kullanılan Microsoft Team Foundation Server aracının çok farklı faaliyetlerde kullanılabilme potansiyeli de vardır. [28]

Microsoft Team Foundation Server, kurumsal projeler için tasarlanmıştır, takım içerisindeki iletişimi ve buna bağlı olarak verimi arttırarak, iş durumunu izleme, takım rollerini yönetme, iş süreçlerini çıkartma ve sunduğu araçlarla proje yönetimini kolaylaştırarak ürünü ortaya çıkartmada etkin rol oynamaktadır. [29]

Microsoft'un kendi ihtiyaçlarını karşılamak için ürettiği Team Foundation Server, yazılım geliştirme sırasında ihtiyaç duyulacak birçok alt sistemi bünyesinde barındırmaktadır. Bu alt sistemlerin birbirleri ile bütünleşik çalışması sayesinde izlenebilirlik imkânı çok artmakta ve parça parça küçük araçların bir araya getirilmesi ile oluşan bir sisteme göre çok daha fazla katma değer üretilmektedir. [28]

Microsoft Team Foundation Server aracının sahip olduğu 7 temel işlem aşağıdaki gibi listelenebilir: [28]

- Kaynak Denetimi
- İş Nesnesi Takibi



- Belge Yönetimi
- Raporlama
- Yapılandırma
- Laboratuvar Yönetimi
- Versiyon Kontrolü

Microsoft Team Foundation Server, “çalışma ögesi izleme” özelliği ile işleri takip etmekte, gruplandırmakta ve işlere ait rapor üretimini yapmaktadır. Bunların yanında zamanlama, kritik yol hesabı, zaman/efor takibi ve maliyet hesabı gibi özellikleri de bünyesinde barındırmamaktadır. [28]

Microsoft, Team Foundation Server aracı, yanında Ms Project gibi bir proje yönetim aracına da sahiptir. Ve bu iki araç, birbirleri ile ilişkili olarak işlevlerini sürdürmektedir. [28]

Microsoft Team Foundation Server, iş öğelerinin yanına herhangi bir tipte dosya eklemeye izin vermekle birlikte bir doküman yönetim aracı değildir. Doküman yönetim işlevleri için Sharepoint aracından faydalanmaktadır. [28]

Microsoft Team Foundation Server uygulamasının istemci katmanı, uygulama katmanı ve veri katmanı olarak üç katmanlı mimari yapısı aşağıdaki gibi listelenebilir: [29]

- **İstemci Katmanı:** İstemci katmanı, uygulama katmanının sunduğu web servislerini kullanarak sahip olduğu ara yüz ile kullanıcıdan gelen isteklere cevap veren katmandır.
- **Uygulama Katmanı:** Uygulama katmanı, istemci katmanının erişimi için ASP.NET web servislerini sunan katmandır.
- **Veri Katmanı:** Microsoft Team Foundation Server, veri katmanına direk olarak iletişime izin vermemektedir. İstemci tarafından gelen tüm istekler, web servis tarafından uygulama katmanına iletilmektedir. İstemci katmanı, SQL Server 2005 üzerine inşa edilmiştir.

#### 4.4.2.5 Eclipse EPF

Eclipse Süreç Çerçevesi (EPF), Eclipse Vakfı tarafından yönetilen ve Eclipse Teknoloji Projesi'nin altında yer alan açık kaynak kodlu bir projedir. EPF Composer'de, ön tanımlı şema kullanımı ile özel bir yolla yapılandırarak kendi yazılım geliştirme sürecini oluşturmak mümkündür. [19]

EPF Composer, organizasyonla ilgili ve bireysel projelerde süreci yürütmeden ve uygulamadan sorumlu süreç mühendisleri, proje liderleri, proje ve program yöneticileri için bir platform sunan araçtır. Sürecin uygulanmasında genelde 2 anahtar problemin ele alınması gerekmektedir. İlk olarak, yazılım geliştiricilerin yazılım geliştirme metotlarını ve anahtar uygulamalarını anlaması gerekmektedir. İkinci olarak, geliştirme takımlarının, proje yaşam döngüsü boyunca kendi geliştirme metotlarını ve anahtar uygulamalarını nasıl uygulayacaklarını da tanımlamaları gerekir. [30]

EPF Composer'de, yukarıdaki ihtiyaçlarla ilgili 2 ana amaç aşağıdaki gibi listelenebilir: [30]

1. Yazılım geliştirme uygulayıcıları için içerik araştırmalarını, yönetmelerini ve dağıtmalarını sağlayacak bilgi tabanlı entelektüel sermaye üretmek
2. Süreç mühendislerini ve proje yöneticilerini kendi kritik geliştirme projelerinde seçme, düzenleme ve hızlı birleştirme işlemlerini destekleyerek süreç mühendisliği yetenekleri oluşturmak

Eclipse EPF OpenUP Rollerini aşağıdaki gibi listelenebilir: [19]

- Analist rolü
- Mimar (Yazılım Geliştirme) rolü
- Geliştirici rolü
- Proje Yöneticisi rolü
- Testçi rolü
- İlgili Grubu üyeliği rolü
- Herhangi biri rolü

### MOBİL UYGULAMALARDA ÇEVİK YAZILIM GELİŞTİRME SÜRECİ YÖNETİM ARACI ANALİZİ

Tez çalışmaları kapsamında, yeni bir çevik yazılım geliştirme süreci yönetim uygulaması üreterek, çevik yazılım geliştirme metodu ile var olan yönetim araçlarının içerdikleri özellikleri, mobil uygulama geliştiricilerin görüşleri doğrultusunda özelleşmek ve böylece mobil uygulama geliştirme sürecinde iyileştirme meydana getirmek hedeflenmiştir. Bu hedef doğrultusunda, mobil uygulamaların ve Scrum metodu yönteminin özellikleri dikkate alınarak daha etkin uygulama geliştirilebilmesi için programlarda meydana getirilmesi gereken değişiklikler 2 temel başlıkta aşağıdaki gibi listelenebilir;

- Uygulamalarda var olan özelliklerin özelleştirilmesi
- Geliştirilen uygulamada ihtiyaç duyulan yeni özelliklerin eklenmesi

Proje geliştirme sürecinin alt adımları için süreç yönetim araçlarına eklenmesine karar verilen özellikler ve özelliklerin alt maddeleri aşağıdaki gibi listelenebilir: [20]

- **Proje tanımlama:** Mobil platformun ismi, mobil uygulama türü, mobil uygulamanın markete çıkma süresi, rakip mobil uygulamalarının bilgisi, mobil uygulamanın yenilik bilgisi, mobil uygulamanın hedef indirme sayısı, mobil uygulamanın ücret bilgisi, mobil uygulamanın oylama özelliği içerme bilgisi, mobil uygulamanın hedef kullanıcı kitlesi bilgisi, desteklenen mobil platformların bilgisi, çoklu platform desteği içerme durumu, mobil uygulama kullanım kolaylığı seviyesi (1, 2, 3),

mobil uygulama geliştiricinin mobil uygulama geliştirme tecrübesi, ekibin mobil uygulama geliştirme bilgisi

- **Proje ekibi yönetimi:** Üyenin mobil uygulama geliştirme tecrübesi, üyenin mobil uygulama geliştirme rolü(yazılımcı, tasarımcı ve testçi gibi), üyenin Scrum yöntemindeki görevi, üyenin Scrum yöntemindeki tecrübesi, üyenin mobil uygulama geliştirdiği platformların bilgisi, üyelerin mobil uygulama geliştirmede alanlarındaki görevlerinin bilgileri.
- **Kullanıcı Yönetimi:** Kullanıcının mobil uygulama projeleri, kullanıcının mobil uygulama geliştirmedeki rolü, kullanıcının mobil uygulama geliştirmedeki tecrübesi, kullanıcının Scrum metodu rolü, kullanıcının Scrum metodu tecrübesi
- **Proje Takibi:** Projelerin platform türlerine göre ayrılması, proje takibini yapacak kişilerin mobil uygulama geliştirmedeki rolleri ve projede görev alan kişilerin Scrum metodu rol ve tecrübe bilgileri, projelerin uygulama türü bilgisi
- **Raporlama:** Süreç ilerlemesi raporu, işlem durumu raporu, uygulama platformuna göre raporlama, markete çıkma süresine göre raporlama, çalışan özelliklerine göre raporlama
- **Proje Detay Bilgisi:** Platformun markete çıkma süresi, mobil uygulama türü, rakip uygulamaların bilgisi, mobil uygulamanın hedef indirme sayısı
- **Projede Görev Üstlenme:** Projede görev üstlenirken, var olan projelerin özellikleri, ihtiyaç duyulan ekip üyesi özellikleri, ihtiyaç duyulan süre bilgilerinin sağlanması
- **Kaynak Yönetimi Yapma:** Projedeki mobil platformlar ve çevik yazılım geliştirme kaynaklı kısıtların yönetilmesi

Mobil uygulamalarda çevik yazılım geliştirme sürecini yönetme uygulamasının geliştirilmesi sürecine, ayrıntılı analiz çalışmaları ile başlanmış ve, önceki bölümlerdeki çalışmalar kapsamında edinilen bilgi birikiminden özellikle de mobil uygulamalarda kullanılan çevik yazılım geliştirme metodlarından Scrum metodu ve yazılım geliştirme süreci yönetim uygulamaları ile ilgili çalışmalardan faydalanılmıştır.

Geliştirilen süreç yönetim aracının özelliklerine karar verilirken, mobil uygulama geliştiricilerin alışkanlıklarını öğrenmek ve yeni geliştirilecek araçla ilgili beklentilerini almak için tezin sonundaki EK 1 de sorularına yer verilen anket çalışması gerçekleştirilerek aracın daha faydalı olabilecek özelliklerle üretilmesine çalışılmıştır.

Analiz aşamasında, daha önceden bahsedilen çalışmalara ek olarak Volkan Baytam tarafından gerçekleştirilen “Scrum yazılım geliştirme metodolojisi için yönetim sistemi tasarımı ve gerçekleşmesi” [31] isimli yüksek lisans tezi kapsamında geliştirilen “Scrum Mapp” uygulamasından da faydalanılmıştır.

Analiz çalışmalarında gereksinim analizi üzerinde yoğunlaşmıştır. Gerçekleştirilen analiz çalışmalarının sonuçlarından, uygulamanın sahip olması gereken farklılıklar ve uygulamanın kullanıcılar için sağlayacağı avantajlar belirlenirken faydalanılmıştır.

## **5.1 Gereksinim Analizi**

Gereksinim analizi çalışmaları, önceki bilgi birikimlerinden ve mobil uygulama geliştiricilerle tez çalışması kapsamında gerçekleştirilen ve EK-A’da sorularına yer verilen anket sonuçlarının değerlendirilmesinden oluşmaktadır.

### **5.1.1 Gereksinimlerin Alınması**

Mobil uygulamalar için özelleştirilmiş yönetim aracını geliştirme sürecinin analizinin ilk adımı olarak, çevik yazılım geliştirme sürecinde kullanılan süreç yönetim uygulamalarından faydalanılmıştır. İkinci adımda, “Mobil Uygulamalar İçin Yazılım Geliştirme Ekibi Ve Süreç Değerlendirmesi” isimli anketle mobil uygulama geliştiricilerin yazılımla ilgili beklentiler alınmıştır. [20]

#### **5.1.1.1 Anket**

Geliştirilen araç hakkında, mobil uygulama geliştiricilerin ihtiyaçlarını belirlemek amacıyla gerçekleştirilen “Mobil Uygulamalar İçin Yazılım Geliştirme Ekibi Ve Süreç Değerlendirme Anketi” isimli anket, yazılım geliştiricilerin kendilerinin ve çalıştıkları firmaların mobil uygulama geliştirme tecrübelerini, mobil uygulama geliştirme yaptıkları platformların isimlerini, mobil uygulama geliştirmede kullanılacak aracın içermesini düşündükleri özellikleri, mobil uygulama geliştiricilerin Scrum bilgisini ve mobil uygulama geliştirirken kullandıkları araçları, kalite ve üretim hızını arttırmak için

eklenmesi gereken özellikler hakkındaki görüşlerini ve mobil uygulama geliştiricilerin çevik yazılıma yatkınlık durumlarını öğrenmeyi amaçlamaktadır. Anket soruları ve şıkların tamamı, tezin sonunda yer alan EK-A'da mevcuttur.

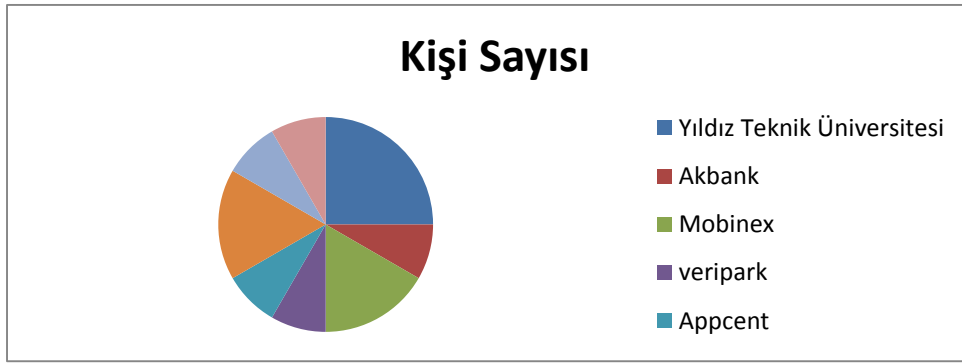
### 5.1.2 Gereksinimlerin Değerlendirilmesi

Bu bölümde mobil uygulamalarda çevik yazılım geliştirme süreci yönetim aracının özelliklerine karar vermek üzere gerçekleştirilen anket sorularına verilen cevaplar ve cevaplarla ilgili değerlendirmelerimize yer verilmiştir.

#### 5.1.2.1 Şirket-Organizasyon Bilgileri

Çizelge 5.1 Çalışma Yeri

Çalışma Yeri	Kişi Sayısı
Yıldız Teknik Üniversitesi	3
Akbank	1
Mobinex	2
Veripark	1
Appcent	1
Turkcell Teknoloji	2
İntertech	1
Bahçeşehir Üniversitesi	1



Şekil 5.1 Çalışma Yeri

Çizelge 5.2 Çalışılan Şirketin Sektörü

Çalışma Şirketin Sektörü	Kişi Sayısı
<b>Mobil uygulama geliştirme</b>	7
Yazılım Geliştirme (Genel)	3
Eğitim (Yazılım Geliştirme vb.)	3
Bilgi Teknolojileri	1
Yazılım Tasarımı	0
Diğer	0



Şekil 5.2 Çalışılan Şirketin Sektörü

Çalışılan sektör sorusuna verilen cevaplar, anketi dolduranların yarısının çalıştığı şirketin yalnızca mobil uygulama geliştiren şirket çalışanları, kalanların da genel yazılım geliştirme işi yapanlar, yazılım geliştirme eğitimi veren kurumların (üniversiteler) üyeleri ve az sayıda da bilgi teknolojileri alanında çalışanlardan oluştuğunu göstermektedir.

Çizelge 5.3 Çalışılan Organizasyonun Büyüklüğü

Organizasyon Büyüklüğü	Kişi Sayısı
Küçük<50	7
Orta<X<250	0
Büyük>250	7



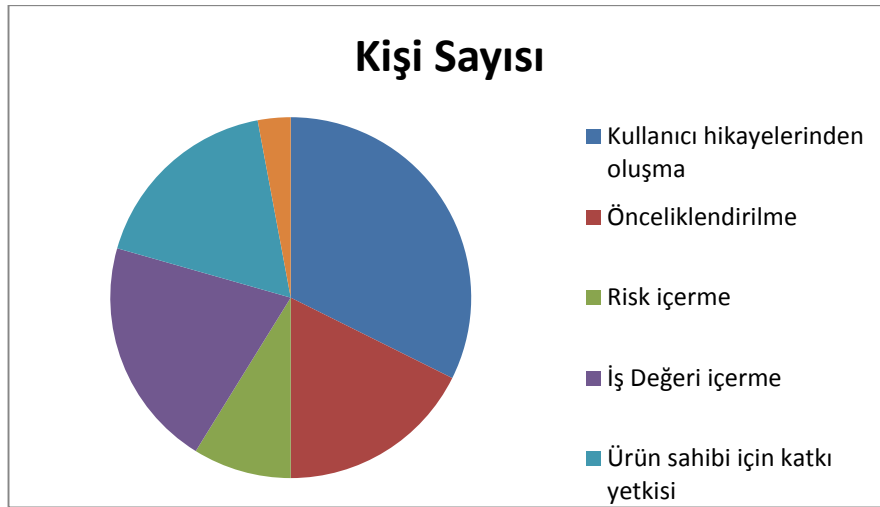
Şekil 5.3 Çalışılan Organizasyonun Büyüklüğü

Çalışılan organizasyonun büyüklüğü sorusuna verilen cevaplara göre, çalışanların yarısı (%50'lik kısım) küçük organizasyonlarda çalışırken diğer yarısı ise büyük organizasyonlarda çalışmaktadır. Bu durum, mobil uygulama geliştiricilerin genelde büyük veya küçük organizasyonlarda çalıştığını ve orta büyüklükte organizasyonlarda mobil uygulama geliştirmenin yeterli düzeyde olmadığını ortaya koymaktadır.

### 5.1.2.2 Kullanılan Yazılım Geliştirme Metodolojileri Bilgileri

Çizelge 5.4 Mobil Uygulama Geliştirilirken Kullanılan Ürün Gereksinim Özellikleri

Mobil Uygulama Geliştirirken Kullanılan Ürün Gereksinim Özellikleri	Kişi Sayısı
Kullanıcı hikâyelerinden oluşma	11
Önceliklendirilme	6
Risk içermeye	3
İş Değeri içermeye	7
Ürün sahibi için katkı yetkisi	6
Diğer	1



Şekil 5.4 Mobil Uygulama Geliştirilirken Kullanılan Ürün Gereksinim Özellikleri

Mobil uygulama geliştirilirken kullanılan ürün gereksinim özellikleri ile ilgili soru ile mobil uygulama geliştiricilerin çalıştıkları şirketlerde uygulama geliştirirken göz önünde bulundukları gereksinim özellikleri ortaya konulmak istenmiştir. Elde edilen sonuçlara göre, şirketlerin gereksinim özellikleri içerisinde en fazla kullanıcı



hikâyelerinden oluşma özelliği öne çıkarken, önceliklendirme, ürün sahibi için katkı yetkisi ve iş değeri içermeye özellikleri de diğer önemli özellikler olarak öne çıkmaktadır.

Çizelge 5.5 Scrum'dan Önce Kullanılan Yazılım Geliştirme Metodolojileri

Scrumdan Önce Kullanılan Yazılım Süreçleri İsimleri	Kişi Sayısı
Spiral	0
Waterfall	8
Unified	0
Process	0
Prototip	4
Diğer	4

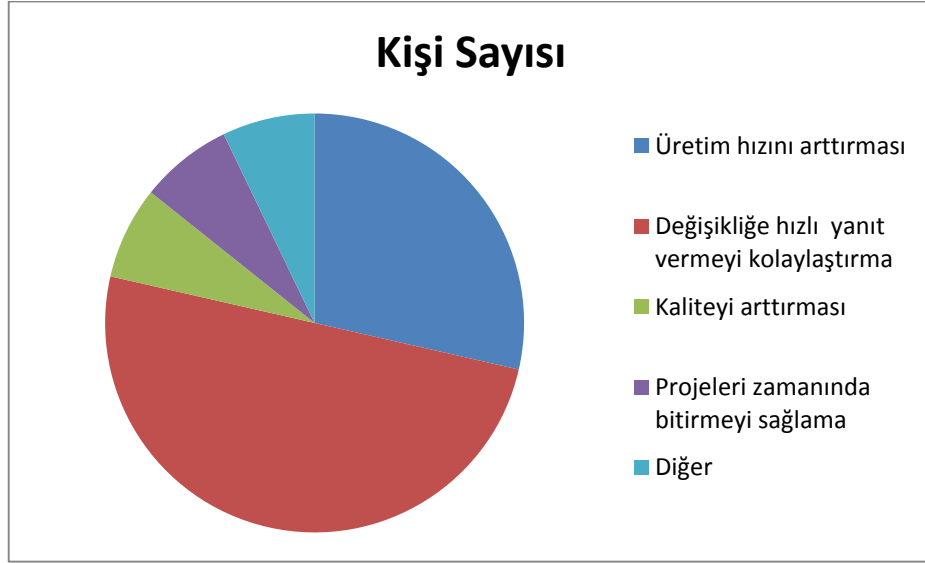


Şekil 5.5 Scrum'dan Önce Kullanılan Yazılım Geliştirme Metodolojileri

Scrum'dan önce kullanılan yazılım geliştirme metotları sorusuna verilen cevaplar, mobil uygulama geliştiricilerin yarısının Scrum'dan önce Waterfall modelini kullandığını göstermiştir. Anket sonuçlarına göre Prototip yazılım geliştirme modeli de, mobil uygulama geliştiriciler arasında yaygın olarak kullanılan bir metot olurken, mobil uygulama geliştiricilerin bir kısmının önerilen beş yazılım geliştirme modelinin dışında modeller kullanmaya başladığı da görülmüştür. Bu durum, mobil uygulama geliştiricilerin yeni bir süreç modeli arayışı içerisinde olduğunu da ortaya koymuştur.

Çizelge 5.6 Scrum Yöntemini Seçmenin Ana Etkeni

Scrum Yöntemini Seçmenin Ana Etkeni	Kişi Sayısı
Üretim hızını arttırması	4
Değişikliğe hızlı yanıt vermeyi kolaylaştırma	7
Kaliteyi arttırması	1
Projeleri zamanında bitirmeyi sağlama	1
Diğer	1



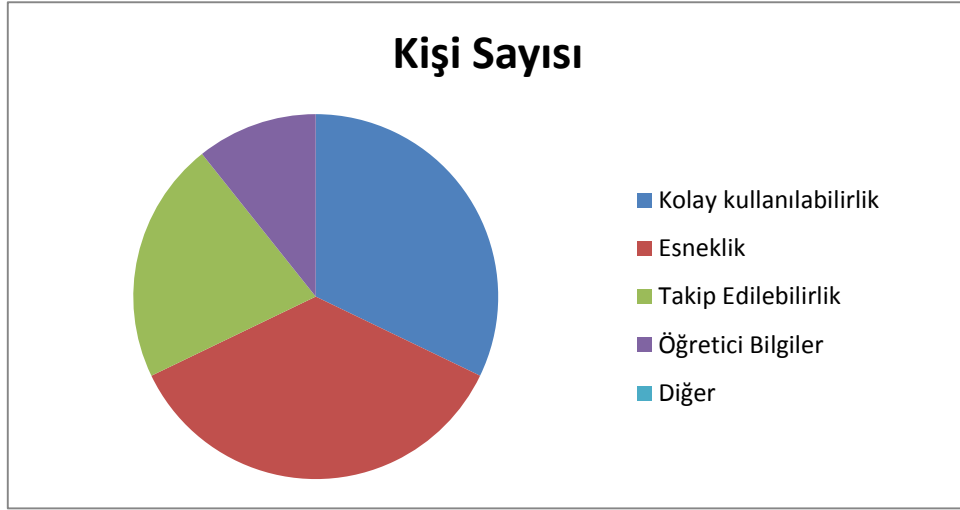
Şekil 5.6 Scrum Yöntemini Seçmenin Ana Etkeni

Scrum yöntemini seçmenin ana etkeni sorusu ile mobil uygulama geliştiricilerin çevik yazılım geliştirme (Scrum) metodunu kullanmaya başlama nedenlerini öğrenmek hedeflenmiştir.

Anket sonuçları, değişikliğe hızlı yanıt vermeyi kolaylaştırma ve üretim hızını arttırma özelliklerinin, Scrum yazılım geliştirme metodunun seçilmesinin iki temel nedeni olduğunu ortaya koymuştur. Sorulara verilen cevaplar, mobil uygulama geliştirmede hızlı ve değişimlere çabuk adapte olabilecek yazılım geliştirme metoduna geçmenin önemini de ortaya koymaktadır.

Çizelge 5.7 Scrum Yazılım Geliştirme Metodolojisi Kullanımında Önemli Özellikler

<b>Scrum Yazılım Geliştirme Metodolojisinin Uygulanmasında Önemli Olan Özellikler</b>	<b>Kişi Sayısı</b>
Kolay kullanılabilirlik	9
Esneklik	10
Takip Edilebilirlik	6
Öğretici Bilgiler	3
Diğer	0



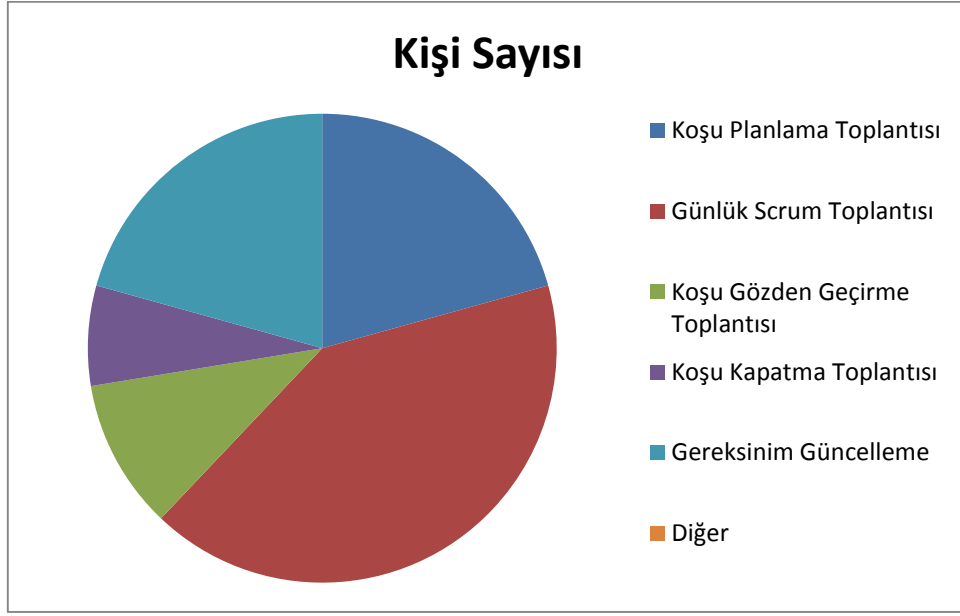
Şekil 5.7 Scrum Yazılım Geliştirme Metodolojisi Kullanımında Önemli Özellikler

“Scrum Yazılım Geliştirme Metodu Kullanımında Önemli Özellikler” sorusu ile, geliştirilen aracın mobil uygulama geliştiricilerine sağlamayı hedeflediği kolaylık ortaya konulmak istenmiştir.

Anket sonuçları, Scrum yazılım geliştirme metodunda en önemli üç özelliğin kolay kullanılabilirlik, esneklik ve takip edilebilirlik olduğunu ortaya koymuştur.

Çizelge 5.8 Scrum Metodolojisinin Mobil Uygulama Geliştirmedeki Önemli Özellikleri

<b>Mobil Uygulama Geliştirmede Scrum Yazılım Geliştirme Metodunun Önemli Olan Özellikleri</b>	<b>Kişi Sayısı</b>
Koşu Planlama Toplantısı	6
Günlük Scrum Toplantısı	12
Koşu Gözden Geçirme Toplantısı	3
Koşu Kapatma Toplantısı	2
Gereksinim Güncelleme	6
Diğer	0



Şekil 5.8 Scrum Metodolojisinin Mobil Uygulama Geliştirmedeki Önemli Özellikleri

“Scrum Metodolojisinin Mobil Uygulama Geliştirmedeki Önemli Özellikleri” sorusu ile, mobil uygulama geliştiriciler için Scrum metodunda en önemli olan özellikleri öğrenmek hedeflenmiştir.

Anket sonuçları, mobil uygulama geliştiriciler için en önemli özelliğin günlük Scrum toplantıları olduğunu, koşu planlama toplantısı ve gereksinim güncelleme işlemlerinin de diğer önemli özellikler olduğunu ortaya koymuştur.

Çizelge 5.9 Çevik Yöntemin Mobil Uygulama Geliştirmeye En Fazla Katkı Alanı?

Çevik Yazılım Geliştirme Mobil Uygulama Geliştirme Sürecine En Fazla Hangi Alanda Katkı Sağladığı Alan	Kişi Sayısı
Yeniliklerin projeye uygulanması	4
Müşterilerin başlangıçta gereksinimlerini belirlemesi	1
Müşterilerin gereksinimlerinin sık değişmesi	8
Projelerin yönetiminin karmaşık hale gelmesi	0



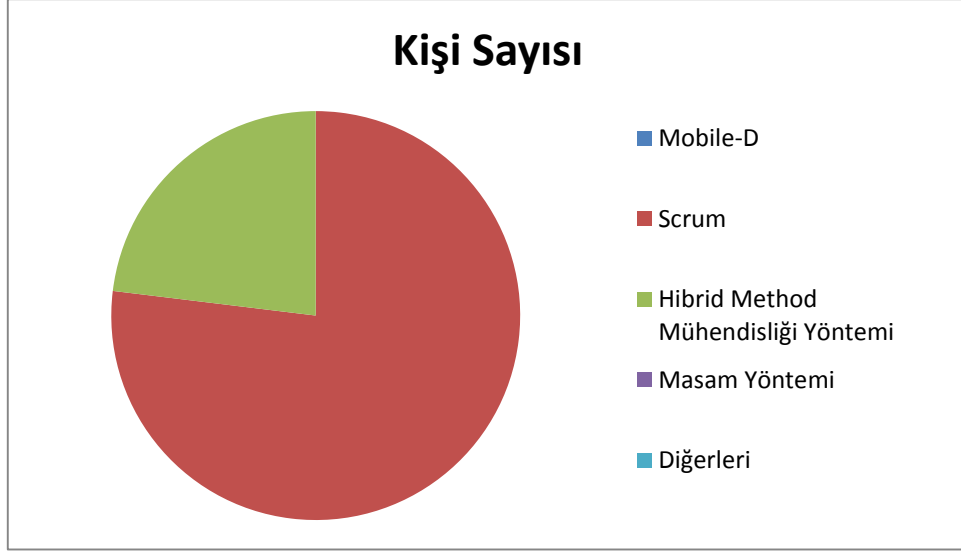
Şekil 5.9 Çevik Yöntemin Mobil Uygulama Geliştirmeye En Fazla Katkı Alanı?

“Çevik Yöntemin Mobil Uygulama Geliştirmeye En Fazla Katkı Alanı” sorusu ile, mobil uygulama geliştiricilerin daha önce kullandıkları yazılım geliştirme metotları göz önünde bulundurarak çevik metodun en fazla katkı sağlayacağı alan beklentisini ortaya koymak hedeflenmiştir.

Anket sonuçlarına göre, müşterilerin gereksinimlerinin sık değişmesi ve yeniliklerin projeye uygulanması, önem sırasına göre en önemli iki özellik olarak öne çıkmıştır.

Çizelge 5.10 Mobil Uygulama Proje Yönetimi Metodolojilerinden Hangisi Uygun

Mobil uygulamalarda Proje Yönetiminde Kullanılan Metodolojilerden Hangisi Ekibiniz İçin uygundur?	Kişi Sayısı
Mobile-D	0
Scrum	10
Hibrid Method Mühendisliği Yöntemi	3
Masam Yöntemi	0
Diğerleri	0



Şekil 5.10 Mobil Uygulama Proje Yönetimi Metodolojilerinden Hangisi Uygun

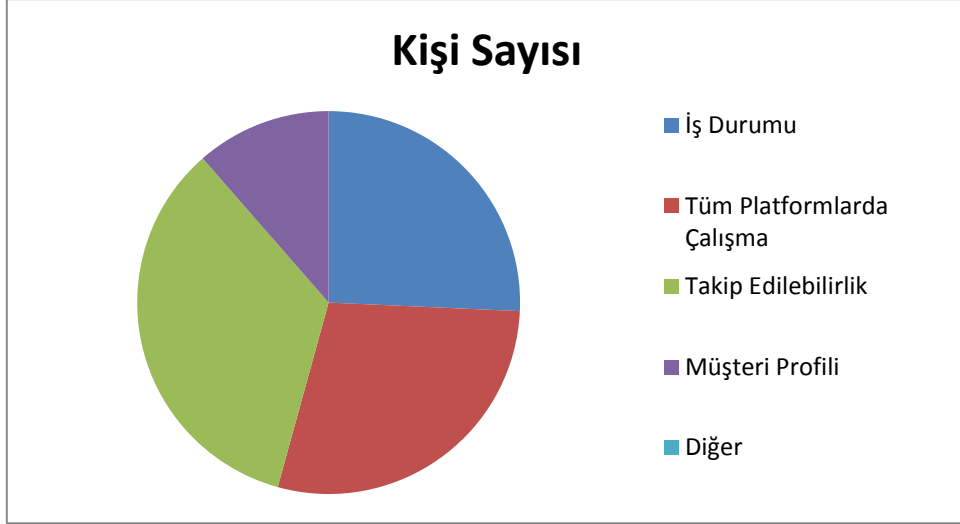
“Mobil Uygulama Proje Yönetimi Metodolojilerinden Hangisi Uygun” sorusu, mobil uygulamalar için geliştirilmiş olan yazılım geliştirme metotlarından mobil uygulama geliştirme ekipleri için en uygun olanı ortaya koymak hedeflemiştir.

Anket sorularına verilen cevaplar, yazılım geliştiricilerinin büyük çoğunluğu için Scrum’ın en uygun metot olduğunu ve Scrum’dan sonra Hibrid Method Mühendisliği metodunun geldiğini göstermektedir.

### 5.1.2.3 Kullanılan Proje Yönetim Aracı Bilgileri

Çizelge 5.11 Mobil Uygulama Geliştirmede Kullanılan Proje Yönetim Aracı Özellikleri

Mobil Uygulama Geliştirmede Kullanılan Proje Yönetim Aracı Özellikleri	Kişi Sayısı
İş Durumu	9
Tüm Platformlarda Çalışma	10
Takip Edilebilirlik	12
Müşteri Profili	4
Diğer	0



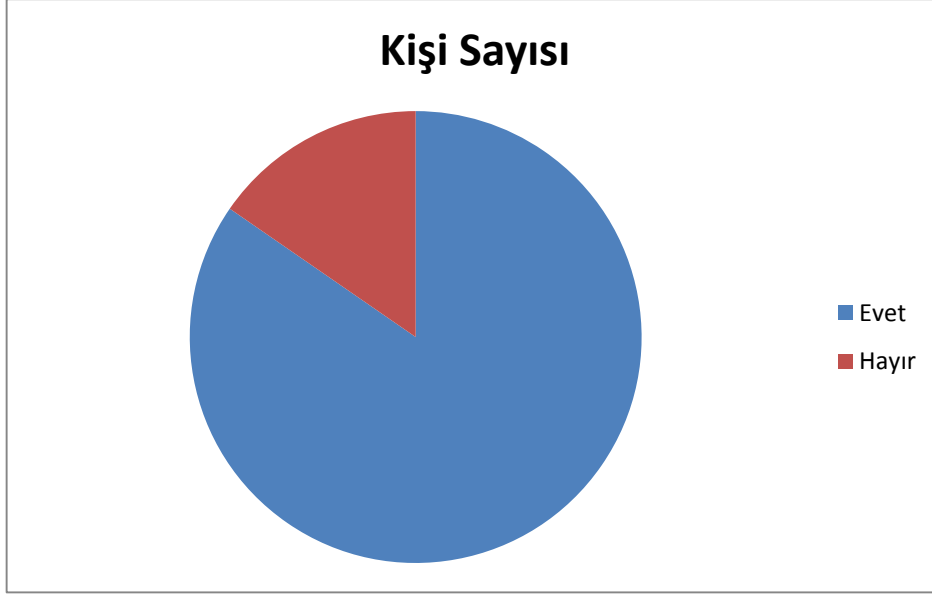
Şekil 5.11 Mobil Uygulama Geliştirmede Kullanılan Proje Yönetim Aracı Özellikleri

Mobil Uygulama Geliştirmede Kullanılan Proje Yönetim Aracı Özellikleri sorusu ile mobil uygulama geliştiricilerin, yeni geliştirilen proje yönetim aracından beklentilerini ortaya koymak hedeflenmiştir.

Anket sonuçları, iş durumunu gösterme, tüm platformlarda çalışma ve takip edilebilirlik özelliklerinin mobil uygulama geliştiricileri için geliştirilecek araçta bulunması gereken en önemli üç özellik olduğunu ortaya koymuştur.

Çizelge 5.12 Mobil Uygulamalarda Scrum Süreci Yönetiminde Araç Gerekli mi?

Mobil Uygulama Geliştirme Süreci Yönetiminde Özel Bir Araç süreç yönetimini, ürün kalitesini ve üretim hızını Arttırır mı?	Kişi Sayısı
Evet	11
Hayır	2



Şekil 5.12 Mobil Uygulamalarda Scrum Süreci Yönetiminde Araç Gerekli mi?

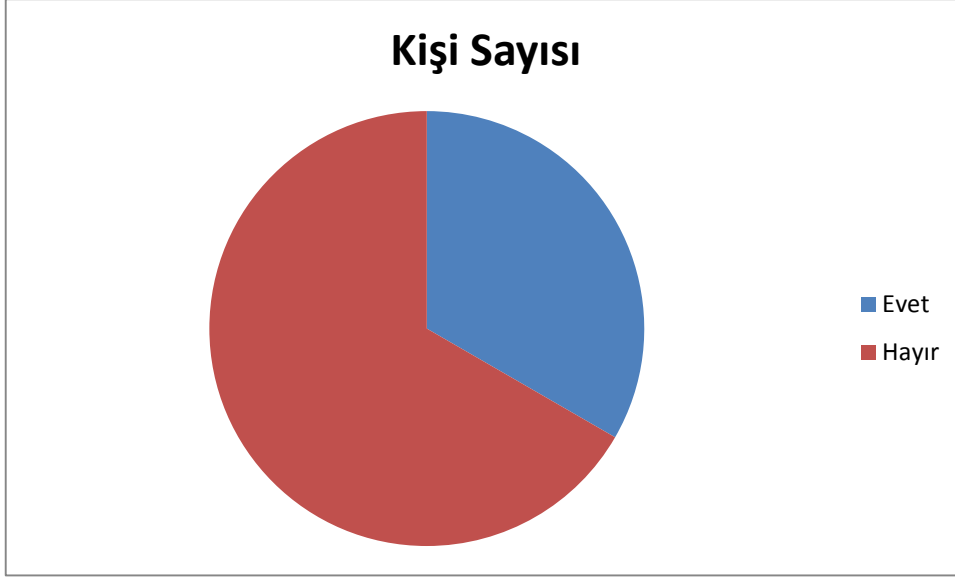
“Mobil Uygulamalarda Scrum Süreci Yönetiminde Araç Gerekli mi?” sorusu ile, yeni araç geliştirilmesi süreci öncesi mobil uygulama geliştiricilerin aracın gerekliliği konusunda ne düşündüklerini öğrenmek hedeflenmiştir.

Anket sonuçları, mobil uygulama geliştiricilerin Scrum süreci yönetiminde özel bir aracın süreç yönetimini, ürün kalitesini ve üretim hızını arttıracığını düşündüğünü ortaya koymuştur.

Çizelge 5.13 Mobil Uygulamada Scrum Süreci Yönetiminde Araç Kullanıyor musun?

<b>Mobil Uygulama Geliştirmede Scrum Sürecini Yönetimi İçin Özel Bir Araç Kullanıyor musun?</b>	<b>Kişi Sayısı</b>
Evet	4
Hayır	8





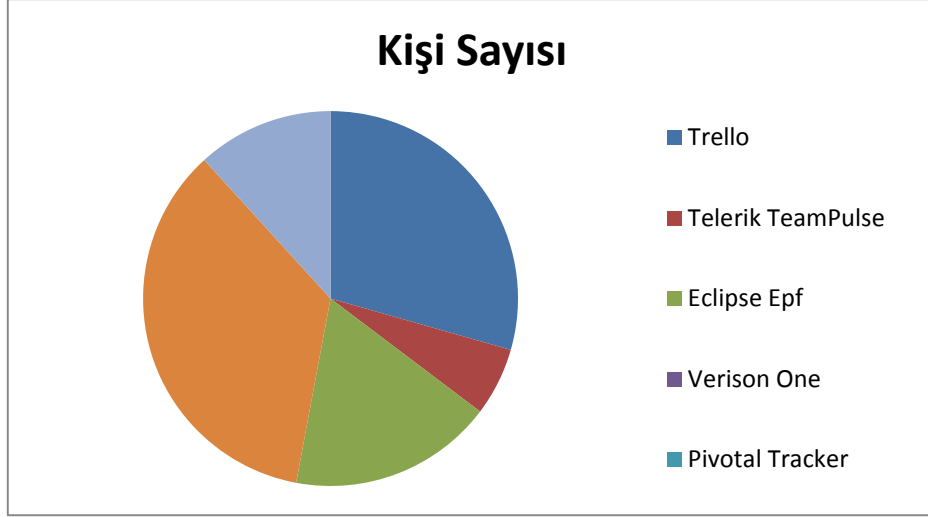
Şekil 5.13 Mobil Uygulamada Scrum Süreci Yönetiminde Araç Kullanıyor musun?

“Mobil Uygulamada Scrum Süreci Yönetiminde Araç Kullanıyor musun?” sorusu ile, mobil uygulama geliştiricilerin araçlar hakkında bilgisi ve var olan faydalanma durumunu öğrenmek hedeflenmiştir.

Anket sonuçları, mobil uygulama geliştiricilerin birçoğunun mobil uygulama geliştirirken kullandıkları özel bir aracın olmadığını ortaya koymuştur. Bu durum, mobil uygulama geliştiricilerinin ihtiyaçlarını dikkate alan özel bir aracın geliştirilmesi gerekliliğini de ortaya koymaktadır.

Çizelge 5.14 Mobil Uygulamalarda Yazılım Geliştirme Sürecinde Kullandığınız Araç?

Mobil Uygulama Geliştirmede Yazılım Geliştirme Sürecini Takip Etmek İçin Kullandığınız Aracın İsmi	Kişi Sayısı
Trello	5
Telerik TeamPulse	1
Eclipse Epf	3
Verison One	0
Pivotal Tracker	0
Jira	6
Microsoft Team Foundation Server	2
Diğer	0



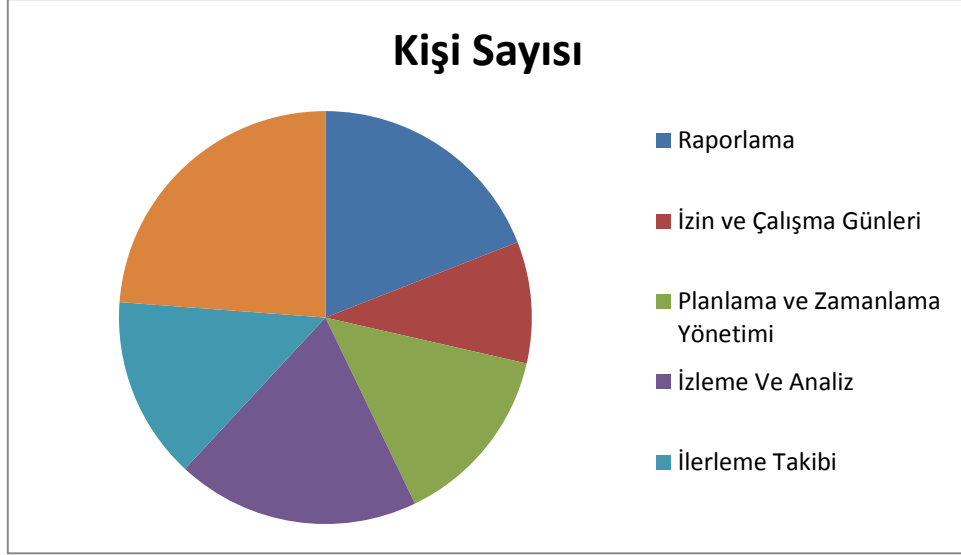
Şekil 5.14 Mobil Uygulamalarda Yazılım Geliştirme Sürecinde Kullandığınız Araç?

Mobil uygulama geliştiricilerin yazılım geliştirme sürecini takip etmek için kullandıkları aracın ismini soran soru ile yeni araç geliştirirken göz önünde bulundurmak üzere mobil uygulama geliştiricilerin eski alışkanlıklarını ve popüler araçları öğrenmek hedeflenmiştir.

Anket sonuçları, mobil uygulama geliştiriciler tarafından en fazla tercih edilen 3 aracın sıra ile Jira, Trello ve Telerik TeamPulse araçları olduğunu ortaya koymuştur.

Çizelge 5.15 Mobil Uygulama Geliştirme Sürecinde Kullanılan Aracın Eksik Özellikleri

Mobil Uygulamalarda Yazılım Geliştirme İçin Kullanılan Araçların Eksik Özellikleri	Kişi Sayısı
Raporlama	4
İzin ve Çalışma Günleri	2
Planlama ve Zamanlama Yönetimi	3
İzleme Ve Analiz	4
İlerleme Takibi	3
Fikirler ve Geri Besleme	5
Diğer	0



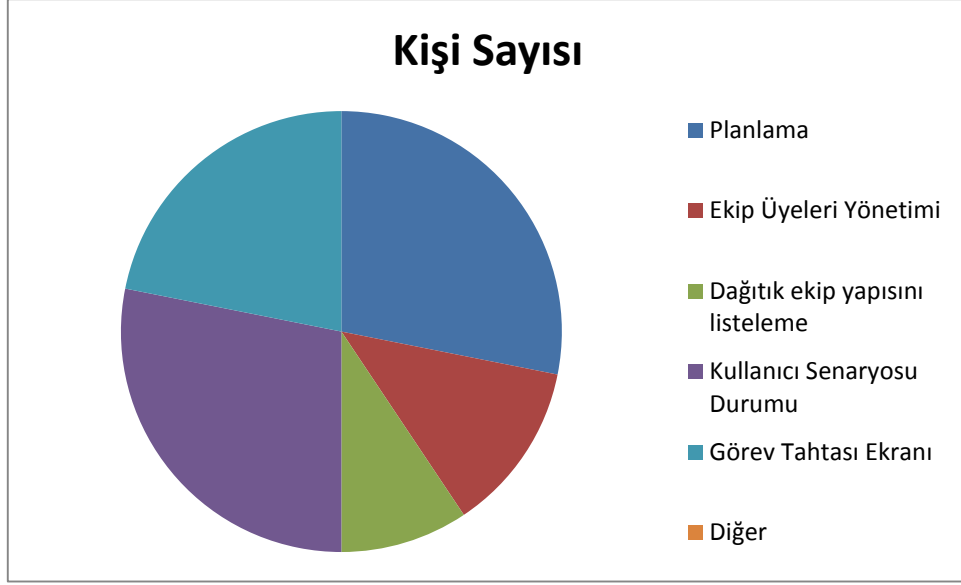
Şekil 5.15 Mobil Uygulama Geliştirme Sürecinde Kullanılan Aracın Eksik Özellikleri

“Mobil Uygulama Geliştirme Sürecinde Kullanılan Aracın Eksik Özellikleri” sorusunun hedefi, mobil uygulama geliştiricilerinin var olan yazılımlarda eksikliğini hissettikleri özellikleri ortaya koyarak geliştirilecek yazılım ile bu eksikliklerin giderilmesini sağlamaktır.

Anket sonuçları, fikir ve geri besleme, raporlama, izleme ve analiz bölümlerinin en fazla eksikliğı hissedilen bölümler olduğunu ortaya koymuştur.

Çizelge 5.16 Mobil Uygulama Proje Yönetimi Aracına Hangi Özellikleri Eklerdiniz?

Mobil Uygulamalar İçin Proje Yönetim Aracı Geliştirse İdiniz Hangi Özellikleri Eklerdiniz?	Kişi Sayısı
Planlama	9
Ekip Üyeleri Yönetimi	4
Dağıtık ekip yapısını listeleme	3
Kullanıcı Senaryosu Durumu	9
Görev Tahtası Ekranı	7
Diğer	0



Şekil 5.16 Mobil Uygulama Proje Yönetimi Aracına Hangi Özellikleri Eklediniz?

“Mobil Uygulama Proje Yönetimi Aracına Hangi Özellikleri Eklediniz?” sorusu ile, yeni geliştirilen aracın içermesi gereken alt bölümleri ve bu bölümlerin önem derecesini ortaya koymak hedeflenmiştir.

Anket sonuçları, Planlama, Kullanıcı Senaryosu Durumunu ve Görev Tahtası ekranının en önemli 3 özellik olduğunu ortaya koymuştur. Ekip üyeleri yönetimi ve dağıtık ekip yapısı listeleme özellikleri de, sıra ile diğer 2 önemli özellik olurken, cevaplar bu özellikleri içermenin araçtan beklentileri karşılamada yeterli olduğunu da göstermektedir.

#### 5.1.2.4 Mobil Uygulama Geliştirme Bilgileri

Çizelge 5.17 Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi

Organizasyonun Mobil Uygulama Geliştirme Tecrübesi	Kişi Sayısı
1-5 Yıl	12
5-10 Yıl	1
10-20 Yıl	1
20-50 Yıl	0
50+ Yıl	0



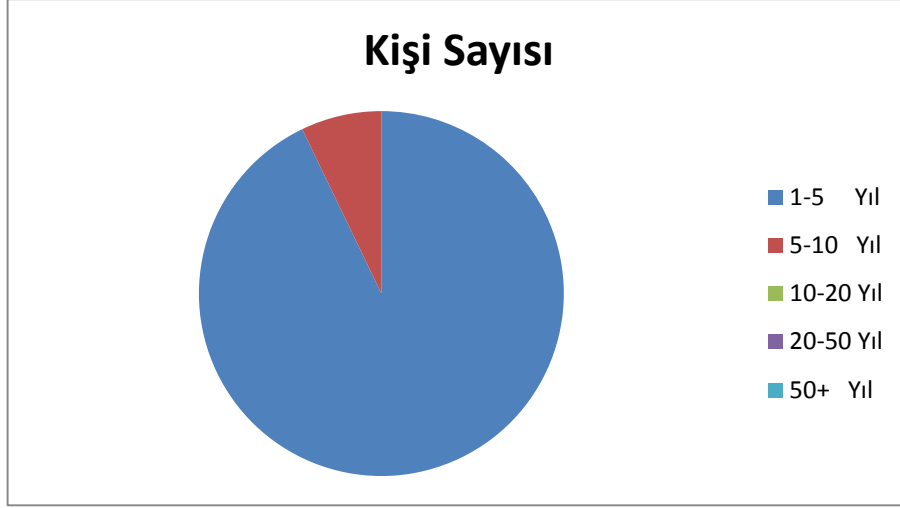
Şekil 5.17 Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi

“Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi” sorusu ile, anketi dolduranların çalıştıkları şirketlerin ne kadar zamandır mobil uygulama geliştirdiklerini öğrenmek hedeflenmiştir.

Anket sonuçları, organizasyonların genellikle son 5 yıl içinde mobil uygulama geliştirmeye başladığını göstermektedir. Bu durum, çalışanların mobil uygulama geliştirmedeki tecrübe eksikliğini, ülkemizde mobil uygulama geliştirmenin yeni olduğunu ve son 5 yıl içinde yaygınlaştığını göstermektedir.

Çizelge 5.18 Yazılım Geliştiricinin Mobil Uygulama Geliştirme Tecrübesi

Yazılım Geliştiricinin Mobil Uygulama Geliştirme Tecrübesi	Kişi Sayısı
1-5 Yıl	13
5-10 Yıl	1
10-20 Yıl	0
20-50 Yıl	0
50+ Yıl	0



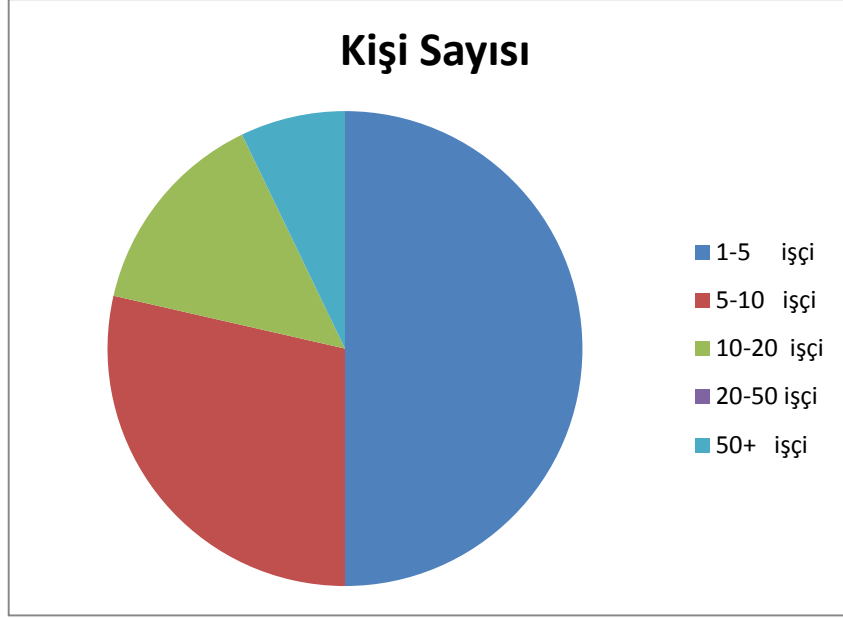
Şekil 5.18 Yazılım Geliştiricinin Mobil Uygulama Geliştirme Tecrübesi

“Çalışılan Organizasyonun Mobil Uygulama Geliştirme Tecrübesi” sorusu ile, anketi dolduranların ne kadar zamandır mobil uygulama geliştirdiklerini öğrenmek hedeflenmiştir.

Anket sonuçları, mobil uygulama geliştiricilerin büyük oranda son 1-5 yıl arası zaman periyodunda bu alanda çalışmaya başladığını göstermiştir. Bu durum, çalışanların tecrübe azlığını ve mobil uygulama geliştirmenin ülkemizde son 5 yılda ön plana çıktığını göstermektedir.

Çizelge 5.19 Şirketteki Mobil Uygulama Geliştirici Sayısı

Şirketteki Mobil Uygulama Geliştiren Takım Üyesi Sayısı	Kişi Sayısı
1-5 işçi	7
5-10 işçi	4
10-20 işçi	2
20-50 işçi	0
50+ işçi	1



Şekil 5.19 Şirketteki Mobil Uygulama Geliştirici Sayısı

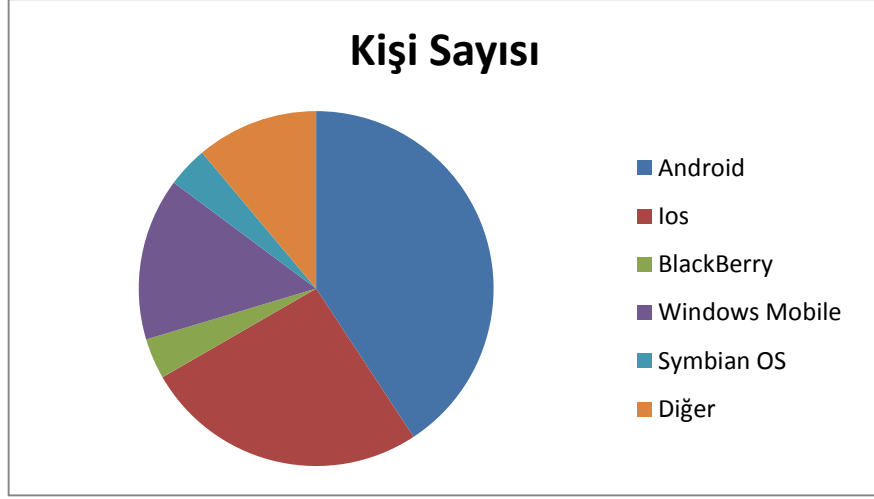
“Şirketteki Mobil Uygulama Geliştirici Sayısı” sorusu ile şirketlerin bünyesinde ne kadar mobil uygulama geliştirici barındırdığı öğrenilerek çevik yazılım geliştirme sürecinde yer alacak mobil uygulama geliştirici sayısını belirlemek hedeflenmiştir.

Anket sonuçları, hem büyük hem de küçük şirketlerde mobil uygulama geliştirici sayılarının düşük olduğunu göstermektedir. Düşük ekip üyesi sayısı klasik Scrum yazılım geliştirme metodunun yapısına uygundur ve uygulanması daha kolaydır.

Scrum uygulanırken süreçte yer alan üye sayısının ortaya koyması, geliştirilecek aracın sahip olacağı tasarımı belirlemek açısından önemlidir.

Çizelge 5.20 Mobil Uygulama Geliştirme Yapılan Platformların İsimleri

Mobil Uygulama Geliştirme Yapılan Platformların İsimleri	Kişi Sayısı
Android	11
Ios	7
BlackBerry	1
Windows Mobile	4
Symbian OS	1
Diğer	3



Şekil 5.20 Mobil Uygulama Geliştirme Yapılan Platformların İsimleri

“Mobil Uygulama Geliştirme Yapılan Platformların İsimleri” sorusu ile, mobil uygulama geliştiricilerin uygulama geliştirdikleri platformlar belirlenerek uygulama geliştirilirken platformlara bağlı önemli olan özellikler belirlenmek istenmiştir.

Anket sonuçları, mobil uygulama geliştiricilerin çoğunun Android işletim sistemi üzerine uygulama geliştirdiğini aynı zamanda Ios ve Windows mobile işletim sistemleri kullanımının da yaygın olduğunu göstermiştir.



### **MOBİL UYGULAMALARDA ÇEVİK YAZILIM GELİŞTİRME SÜRECİ YÖNETİM ARACI TASARIMI VE GERÇEKLEMESİ**

Geliştirilen araçla ilgili analiz çalışmaları tamamlandıktan sonra, uygulamanın tasarımına geçilmiş ve tasarım bölümü işlevsel modelleme, veri modeli, veri tabanı tasarımı olarak 3 adımda gerçekleştirildikten sonra uygulamanın gerçekleştirme (kodlama) aşamasına geçilmiştir.

Bu bölüm içerisinde, tasarım ve gerçekleştirme işlemlerinin içerdiği ve aracın geliştirilmesi sürecinde gerçekleştirilen tüm alt adımlarla ilgili paylaşımlarda bulunulacaktır.

#### **6.1-Tasarım**

Mobil uygulamalarda çevik yazılım geliştirme süreci yönetim aracının sunulan tasarımı ile mobil uygulama geliştirmede proje yönetiminin etkinliğini arttırmak, süreç yönetimini kolaylaştırmak ve mobil uygulama geliştiriciler için var olan işletim ve yönetim uygulamalarının kullanımındaki eksiklikleri ortadan kaldırmak hedeflemiştir.

[20]

Uygulamanın tasarımının gerçekleştirilmesinde dikkate alınan özellikler, aşağıdaki gibi listelenebilir; [20]

- Tüm platformlarda çalışma gerekliliği
- Mobil uygulamalarda öne çıkan özelliklerdeki değişiklik
- Markete hızlı çıkma zorunluluğu
- Mobil uygulama geliştirme sürecinde teknoloji farklılıkları

### 6.1.1 İşlevsel Modelleme

Bu bölümde, uygulamanın içerdiği 5 kullanıcı tipi olan Ürün Sahibi, Scrum Master, Takım üyesi, Sistem Yöneticisi ve Misafir Kullanıcının gerçekleştirdikleri işlemler, aktörleri ile birlikte EK B 'de kullanıcı diyagramı olarak sunulmuştur.

### 6.1.2 Veri Modeli

Uygulamanın kullanıcıları ve içerdiği temel fonksiyonlarla ilgili varlıkların özellikleri ve varlıkların diğer varlıklarla olan ilişkileri, EK-C'de ayrıntılı varlık ilişki modeli olarak sunulmuştur.

### 6.1.3 Veri tabanı Tasarımı

Bu bölümde, uygulamanın veri tabanı tabloları ve bu tabloların birbirleri ile ilişkilerini ortaya koyan bilgiler paylaşılacaktır.

#### 6.1.3.1 Tablo Bilgileri

ScrumMaster	UrunSahibi
KullaniciId	KullaniciId
KullaniciAdi	KullaniciAdi
KullaniciSifre	KullaniciSifre
TakimId	TakimId
ProjeId	ProjeId
SprintId	BacklogId
BacklogId	GozdenGecirmeId
ToplantıId	ScrumRolu
ScrumRolu	ScrumTecrubesi
ScrumTecrubesi	MobilUygulamaGelistirmePlatformu
MobilUygulamaGelistirmePlatformu	MobilUygulamaGelistirmeRolu
MobilUygulamaGelistirmeRolu	MobilUygulamaGelistirmeTecrubesi
MobilUygulamaGelistirmeTecrubesi	

Şekil 6.1 Scrum Master ve Ürün Sahibi Tabloları

<b>TakimUyesi</b>	
KullaniciId	
KullaniciAdi	
KullaniciSifre	
TakimId	
ProjeId	
SprintId	
BacklogId	
ScrumRolu	
ScrumTecrubesi	
MobilUygulamaGelistirmePlatformu	
MobilUygulamaGelistirmeRolu	
MobilUygulamaGelistirmeTecrubesi	

<b>MisafirKullanici</b>	
KullaniciId	
KullaniciAdi	
KullaniciSifre	
ProjeId	
SprintId	
ToplantId	

Şekil 6.2 Takım Üyesi ve Misafir Kullanıcı Tabloları

<b>SistemYoneticisi</b>	
KullaniciId	
KullaniciAdi	
KullaniciSifre	

Şekil 6.3 Sistem Yöneticisi Tablosu

Proje		Toplantı	
<input type="checkbox"/>	Projeİd	<input type="checkbox"/>	Toplantıİd
<input type="checkbox"/>	ProjeAdi	<input type="checkbox"/>	ToplantıAdi
<input type="checkbox"/>	ProjeKisileri	<input type="checkbox"/>	ToplantıKonusu
<input type="checkbox"/>	ProjeBaslangicTarihi	<input type="checkbox"/>	ToplantıAmaci
<input type="checkbox"/>	ProjeBitisTarihi	<input type="checkbox"/>	SprintAdi
<input type="checkbox"/>	SprintSayisi	<input type="checkbox"/>	ScrumMasterAdi
<input type="checkbox"/>	MobilUygulamaTuru	<input type="checkbox"/>	UrunSahibiAdi
<input type="checkbox"/>	MobilPlatformİsmi	<input type="checkbox"/>	TakimUyeleriAdlari
<input type="checkbox"/>	MarketeCikmaSuresi	<input type="checkbox"/>	ToplantıSuresi
<input type="checkbox"/>	RakipMobilUygulamalar	<input type="checkbox"/>	ToplantıTarihi
<input type="checkbox"/>	KullanımKolayligi		
<input type="checkbox"/>	Yenilikİcerme		
<input type="checkbox"/>	UcretliOlma		
<input type="checkbox"/>	HedefİndirmeSayiAraligi		
<input type="checkbox"/>	HedefKullaniciKitlesi		
<input type="checkbox"/>	OylamaOzelligiİcerme		
<input type="checkbox"/>	CokluPlatformDestegiİcerme		
<input type="checkbox"/>	DesteklenenDigerMobilPlatfor...		

Şekil 6.4 Proje Ve Toplantı Tabloları

Kaynak	
<input type="checkbox"/>	Projeİd
<input type="checkbox"/>	ProjeAdi
<input type="checkbox"/>	Kisiİd
<input type="checkbox"/>	CihazAdi
<input type="checkbox"/>	UygulamaBoyutu
<input type="checkbox"/>	UygulamaİsletimSistemi
<input type="checkbox"/>	UygulamaGirdileri
<input type="checkbox"/>	UygulamaCiktilari

Şekil 6.5 Kaynak Tablosu

## 6.2 Gerçekleme

Bu bölümde, uygulamanın içerdiği ekranlar hakkında temel tanıtıcı bilgiler ile ekranlara ait görüntüler paylaşılacaktır.

### 6.2.1.1 Ekran Örnekleri



[Anasayfa](#)  
[Projeler](#)  
[Toplantılar](#)  
[Raporlar](#)  
[Kullanıcı Yönetimi](#)

Kullanıcı Adı :

Kullanıcı Tipi : Urun Sahibi

Şifre :

Tüm hakları saklıdır. İzin alınmadan kullanılamaz

Şekil 6.6 Kullanıcı Girişi Ekranı



[Anasayfa](#)  
[Projeler](#)  
[Toplantılar](#)  
[Raporlar](#)  
[Kullanıcı Yönetimi](#)

Kullanıcı Adı :

Kullanıcı Şifre:

Kullanıcı Şifre(Tekrar) :

Scrum Metodu Rolü : Urun Sahibi

Scrum Tecrübesi : 1 Yıl

Mobil Uygulama Geliştirme Platformu : Android

Mobil Uygulama Geliştirme Rolü : Yazılım Geliştirme

Mobil Uygulama Geliştirme Tecrübesi : 1 Yıl

Tüm hakları saklıdır. İzin alınmadan kullanılamaz

Şekil 6.7 Kullanıcı Kaydı Ekleme Ekranı



Şekil 6.8 Sistem Yöneticisi Ekranı



Şekil 6.9 Ürün Sahibi Ekranı



Şekil 6.10 Scrum Master Ekranı



Şekil 6.11 Takım Üyesi Ekranı





<a href="#">Anasayfa</a> <a href="#">Projeler</a> <a href="#">Toplantılar</a> <a href="#">Raporlar</a> <a href="#">Kullanıcı Yönetimi</a>	<b>KAYNAK VE GÖREV GÖRÜNTÜLEME</b>		<b>KİŞİSEL BİLGİ YÖNETİMİ</b>	
	<input type="button" value="Kaynak Görüntüleme"/>		<input type="button" value="Kişisel Bilgi Yönetimi"/>	
	<input type="button" value="Görev Bilgileri Görüntüleme"/>			
	<b>PROJELER</b>		<b>TOPLANTILAR VE SPRINTLER</b>	
	<input type="button" value="İşlem Durumu Raporlama"/>		<input type="button" value="Toplantı Katılım Raporu"/>	
	<input type="button" value="Çalışan Özelliklerine Göre Raporlama"/>		<input type="button" value="Toplantı İçerik Raporu"/>	
	<input type="button" value="Uygulama Geliştirme Platformlarına Göre Raporlama"/>		<input type="button" value="Sprint Bilgileri Görüntüleme"/>	
	Tüm hakları saklıdır. İzin alınmadan kullanılamaz			

Şekil 6.12 Misafir Kullanıcı Ekranı



<a href="#">Anasayfa</a> <a href="#">Projeler</a> <a href="#">Toplantılar</a> <a href="#">Raporlar</a> <a href="#">Kullanıcı Yönetimi</a>	Proje Adı :	<input type="text"/>
	Görev Alacak Kişiler :	<input type="text"/>
	Başlangıç Tarihi :	<input type="text"/>
	Bitiş Tarihi :	<input type="text"/>
	Sprint Sayısı :	<input type="text" value="1"/>
	Mobil Uygulama Türü	<input type="text" value="Kişisel Yardımcılar"/>
	Mobil Platform İsmi	<input type="text" value="Android"/>
	Markete Çıkma Süresi	<input type="text" value="0-1 Ay"/>
	Rakip Mobil Uygulamalar (V/Y)	<input type="text" value="Var"/>
	Mobil Uygulama Kullanım Kolaylığı (1,2,3)	<input type="text" value="1"/>
	Mobil Uygulama Yenilik Getirmesi (E->1 /H->0)	<input type="text" value="1"/>
	Mobil Uygulama Ücretli Olması (E->1 /H->0)	<input type="text" value="1"/>
	Mobil Uygulama Hedef İndirme Sayısı	<input type="text" value="0-1000"/>
	Mobil Uygulama Hedef Kullanıcı Kütlesi (Yaş Aralığı)	<input type="text" value="0 -10"/>
	Mobil Uygulama Oylama Özelliği (E->1 /H->0)	<input type="text" value="1"/>
	Çoklu Platform Desteği (E->1 /H->0)	<input type="text" value="1"/>
Desteklenen Diğer Mobil Platformlar	<input type="text"/>	
	<input type="button" value="Temizle"/>	<input type="button" value="Proje Tanımla"/>

Şekil 6.13 Proje Tanımlama Ekranı



<a href="#">Anasayfa</a> <a href="#">Projeler</a> <a href="#">Toplantılar</a> <a href="#">Raporlar</a> <a href="#">Kullanıcı Yönetimi</a>	Toplantı Adı :	<input type="text"/>
	Toplantı Konusu :	<input type="text"/>
	Toplantının Amacı :	<input type="text"/>
	Toplantının Yer aldığı Sprint :	<input type="text"/>
	Toplantı Scrum Master'ı	<input type="text"/>
	Toplantı Ürün Sahibi :	<input type="text"/>
	Toplantı Takım Üyeleri	<input type="text"/>
	Toplantı Süresi	<input type="text"/>
	Toplantı Tarihi	<input type="text"/>
		<input type="button" value="Temizle"/>

Tüm hakları saklıdır. İzin alınmadan kullanılamaz

Şekil 6.14 Toplantı Tanımlama Ekranı





[Anasayfa](#)  
[Projeler](#)  
[Toplantılar](#)  
[Raporlar](#)  
[Kullanıcı Yönetimi](#)

**PROJELER**

[İşlem Durumu Raporlama](#)

[Çalışan Özelliklerine Göre Raporlama](#)

[Uygulama Geliştirme Platformlarına Göre Raporlama](#)

[Uygulama Türüne Göre Raporlama](#)

**TOPLANTILAR**

[Toplantı Katılım Raporu](#)

[Toplantı İçerik Raporu](#)

**GÖREVLİ KULLANICILAR**

[Ürün Sahibine Göre Rapor](#)

[Scrum Master'a Göre Rapor](#)

[Takım Üyesine Göre Rapor](#)

ProjeAdı	ProjeKisileri	ProjeBaslangicTarihi	ProjeBitisTarihi	SprintSayisi	MarketeCikmaSuresi	HedefIndirmeSayi	Araligi
Proje1	ismail	22.12.2013	27.12.2013	1	0-1 Ay	0-1000	
proje 2	ismail, ahmet	22.12.2013	27.12.2013	3	6-12 Ay	1000-5000	

Tüm hakları saklıdır. İzin alınmadan kullanılamaz

Şekil 6.15 Raporlama Ekranı-1

[Anasayfa](#)  
[Projeler](#)  
[Toplantılar](#)  
[Raporlar](#)  
[Kullanıcı Yönetimi](#)

**PROJELER**

[İşlem Durumu Raporlama](#)

[Çalışan Özelliklerine Göre Raporlama](#)

[Uygulama Geliştirme Platformlarına Göre Raporlama](#)

[Uygulama Türüne Göre Raporlama](#)

**TOPLANTILAR**

[Toplantı Katılım Raporu](#)

[Toplantı İçerik Raporu](#)

**GÖREVLİ KULLANICILAR**

[Ürün Sahibine Göre Rapor](#)

[Scrum Master'a Göre Rapor](#)

[Takım Üyesine Göre Rapor](#)

KullaniciAdi	ScrumRolu	ScrumTecrubesi	MobilUygulamaGelistirmePlatformu	MobilUygulamaGelistirmeRolu	MobilUygulamaGelistirmeTecrubesi
ismail	Scrum Master	1 Yıl	Android	Yazılım Geliştirme	1 Yıl
ismaildr	Scrum Master	3 Yıl	Blackberry	Sistem Analizi	2 Yıl

Tüm hakları saklıdır. İzin alınmadan kullanılamaz

Şekil 6.16 Raporlama Ekranı-2

Yazılım geliştirme süreci tamamlandıktan sonra, mobil uygulama geliştiricilerin yazılımla ilgili görüşleri alınmış ve örnek bir uygulama üzerinden geliştirilen aracın var olan yönetim araçlarına göre durumu ve sunduğu avantajlar ortaya konarak geliştirilen aracın, mobil uygulamaya geliştirmede süreç yönetimini kolaylaştırma ve performansı artırma hedefini ne kadar sağladığı belirlenmeye çalışılmıştır. [20]

### 6.3 Aracın Değerlendirilmesi

Tez çalışmasının çıktılarında biri olarak, uygulama geliştirilirken mobil uygulama geliştiriciler için kolay kullanım ve yüksek performans sağlama hedefini gerçekleştirmek için klasik proje yönetim araçlarının özelliklerine eklemeler yapılarak özelleştirmelere gidilmiştir. Proje yönetim araçlarının ekranlarındaki özelleştirmeler Çizelge 6.1’de yer almaktadır.

Çizelge 6.1 Proje Yönetim Aracı Özelliklerinde Meydana Getirilen Özelleştirmeler

EKRAN	GERÇEKLEŞTİRİLEN ÖZELLEŞTİRME
Sisteme giriş	Sisteme girişin yetkiye göre özelleştirilmesi (Ürün Sahibi, Scrum Master, Takım Üyesi, Misafir Kullanıcı ve Sistem Yöneticisi olarak)
Kişisel bilgileri görüntüleme	Mobil uygulama geliştirme ve Scrum yazılım metodu özelliklerinin görüntülenmesi
Kişisel bilgi yönetimi	Mobil uygulama geliştirme ve Scrum yazılım metodu özelliklerinin yönetimi
Proje tanımlama	Mobil uygulamalara ve Scrum yazılım geliştirme metoduna ait özelliklere tanımlamada yer verilmesi
Proje ekibi yönetimi	Proje ekibi üyelerinin Scrum rollerine ek olarak mobil uygulama geliştirme rol bilgilerine göre yönetilmesini sağlama
Kullanıcı kaydı ekleme	Kullanıcıların sisteme Scrum yazılım geliştirme metodu rolleri ve mobil uygulama geliştirme özelliklerine göre eklenmesi
Kullanıcı kaydı silme	Kullanıcılar sistemden Scrum yazılım geliştirme metodu rolleri ve mobil uygulama geliştirme özelliklerine göre silinmesi
Proje takibi yapma	Projelerin çalışan özellikleri (Scrum yazılım geliştirme metodu rolleri), uygulama geliştirme platformları ve mobil uygulama türlerine göre takip edilebilmesini sağlama

Çizelge 6.1 Proje Yönetim Aracı Özelliklerinde Meydana Getirilen Özelleştirmeler

(Devamı)

Raporlama	Raporlama ekranına mobil uygulama ve Scrum metodunun çalışan özellikleri, uygulama geliştirme platformu ve mobil uygulama türlerine göre raporlama, Sprint toplantılarına ait katılım ve içerik raporlarının eklenmesi
Proje detayları görüntüleme	Proje üyelerinin mobil uygulamalara ait (platformun markete çıkma süresi, uygulama türü, rakip uygulamalar, hedef indirme sayısı) özellikleri ile Scrum yazılım geliştirme metoduna ait özelliklerini görüntülenmesini sağlama
Sistem kullanıcılarını yönetme	Sistem kullanıcılarının Scrum yazılım geliştirme metodu rollerine göre yönetilmesini sağlama
Kaynak yönetme	Mobil platformlardan kaynaklı kısıtların yönetiminin sağlama
Toplantı tanımlama	Toplantı tanımlanırken, Scrum metoduna ait Sprint bilgileri ile katılımcıların Scrum metodu rolü bilgilerinin de alınması
Projede görev üstlenme	Projede Scrum metodu ve mobil uygulama geliştirme rolüne göre görev üstlenme

Geliştirilen araçta en fazla özelleştirme gerçekleştirilen ekranlara ait daha kapsamlı bilgiler, aşağıdaki gibi listelenebilir: [20]

- **Sisteme Giriş:** Scrum yazılım geliştirme metodunun kullanıcı tipleri temel alınarak, kullanıcı tipine göre özelleştirilmiş bir giriş ekranı oluşturulmuştur. Sistemin giriş ekranında yer verilen kullanıcı türleri sırası ile Ürün Sahibi, Scrum Master, Takım Üyesi, Misafir Kullanıcı ve Sistem Yöneticisi'dir.
- **Kişisel sayfalar:** Sistemde yer alan beş tür kullanıcı için, kullanım diyagramı (işlevler ve aktörleri olarak sistem kullanıcıları) ve ER diyagramı (varlıkların özellikleri ve birbirleri ile olan ilişkileri) dikkate alınarak tüm sisteme uygun olacak ve basit kullanım sağlayacak şekilde kişisel sayfalar oluşturulmuştur.

- **Proje tanımlama:** Proje tanımlama ekranı, gerek mobil uygulama geliştirme özellikleri gerekse Scrum yazılım geliştirme metodunun özellikleri dikkate alınarak ve yeni bir projeye ait tüm bilgiler (proje adı, projede görev alacak kişiler, projenin başlangıç ve bitiş tarihleri, Sprint sayısı, geliştirilen mobil uygulamaya ait tür, hedef platform ismi, markete çıkma süresi, rakip mobil uygulama bilgisi, kullanım kolaylığı, yenilik getirme, ücretli olma, hedef indirme sayısı, hedef kullanıcı kitlesi, oylama özelliği bilgileri, çoklu platform desteği, desteklediği ek platformların isimleri) girilebilecek şekilde basit ara yüz ile gerçekleştirilmiştir.
- **Toplantı tanımlama:** Toplantı tanımlama ekranı, Scrum yazılım geliştirme metodunun özellikleri dikkate alınarak toplantı tanımlama için gerekli tüm bilgilere (toplantıya ait ad, konu, amaç, yer aldığı Sprint, Scrum Master, Ürün sahibi, takım üyeleri, süre ve gerçekleştirilme tarihi) yer verilerek gerçekleştirilmiştir.
- **Sistem kullanıcıları yönetme:** Sistem kullanıcıları yönetimi ekranı, mobil uygulama ve Scrum yazılım geliştirme metodu rolleri dikkate alınarak kullanıcı ekleme ve kullanıcı silme işlemleri ile kullanıcılarının rolleri dikkate alınarak bilgilerinin yönetiminin yapılabilmesini sağlayacak şekilde basit ara yüzle gerçekleştirilmiştir.
- **Proje takibi yapma:** Proje takibi ekranı oluşturulurken, mobil uygulama geliştirme ve Scrum yazılım geliştirme metodu göz önünde bulundurularak projelerin işlem durumuna, çalışan özelliklerine, uygulama geliştirme platformlarına ve mobil uygulama türüne göre takibini yapabilmelerini sağlamasına dikkat edilmiştir.
- **Raporlama:** Raporlama ekranı, basit ara yüz üzerinden projelerin işlem durumu, çalışan özellikleri, uygulama geliştirme platformu ve mobil uygulama türüne göre raporlanması, toplantılara ait katılım ve içerik raporlarının oluşturulması ve her bir kullanıcı türüne göre ayrı ayrı raporlama özelliğine sahip olan ekrandır.

### SONUÇ VE ÖNERİLER

Bu bölüme kadar, tez çalışması kapsamında ilk olarak çevik yöntemler, mobil uygulamalar, mobil uygulamalarda kullanılan çevik yazılım geliştirme metotları ve süreç yönetim araçları ile ilgili çalışmalar tamamlanmıştır. Daha sonra, çalışmada elde edilen bilgi birikiminden de faydalanarak mobil uygulama geliştiricilerle gerçekleştirilmiş ve anket sonuçları değerlendirilmiştir. Ardından da mobil uygulama geliştiriciler için özelleştirilmiş çevik yazılım geliştirme süreci yönetim aracının analiz, tasarımı işlem adımları ve kodlanması gerçekleştirilmiştir.

Tezin hedeflenen faydaları sağlaması için, çalışmalara çevik yazılımlar ve mobil uygulamalarla ilgili gerekli ön bilgilerin sağlanması ile başlanması önemli olmuştur. Çalışmaların ilerleyen aşamaları olan mobil uygulamalarda kullanılan çevik yazılım geliştirme metotları ve çevik yazılım geliştirme sürecinde kullanılan araçlar incelenmesinde örnek uygulamalardan faydalanılmıştır. Araçların kullanımı ve özellikleri ile ilgili edinilen kapsamlı bilgiler, ilgili akademik konferanslarda yayınlanan bildiriler yoluyla paylaşılmıştır.

Tezin kapsamında gerçekleştirilen anket ile hedef kullanıcı kitlesi olan mobil uygulama geliştiricilerin tüm çalışmada elde edilen bilgi birikimi ve problemlere ilişkin alternatif çözüm yolları ile ilgili görüşleri ve beklentilerini almak hedeflenmiştir. Gerçekleştirilen anket sonuçlarına ve sonuçların kapsamlı değerlendirmelerine, tez çalışmasının içeriğinde yer verilmiştir. Bu değerlendirmeler, mobil uygulama geliştiricilerin bakış açısından tüm çalışmada elde edilen bilgi birikiminin

kullanılabilirliğinin değerlendirilmesini ve alternatif çözüm yollarının durumunu ortaya koyması açısından önemli olmuştur.

Anket çalışmalarının sonucunda mobil uygulamalarda çevik yazılım geliştirme sürecinde proje yönetiminin daha etkin hale getirilmesinde (alternatif çevik metotlar içerisinde ülkemizde en fazla tercih edilen yazılım geliştirme metodu olan) mobil uygulamalar için özelleştirilmiş Scrum'ın kullanılmasına karar verilmiştir. Aynı zamanda mobil uygulama geliştirme özellikleri dikkate alınarak geliştirilecek yeni bir aracın çok büyük katkıları olabileceği görülmüştür.

Tezin final çalışmalarını oluşturan mobil uygulamalar için yeni bir işletim ve yönetim uygulaması gerçekleştirilmesine kapsamlı analiz ve tasarım adımları ile başlanmıştır. Böylece, uygulamanın kodlanmasına başlanmadan önce gerek kullanıcı türleri gerekse uygulama nesnelere ait tüm özellikler ve bu özelliklerin birbirleri ile ilişkileri belirlenmiştir. Analiz, tasarım ve uygulamanın kodlanması tamamlandıktan sonra, uygulamaya ait ekran görüntüleri ile ilgili önemli bilgiler de tezin beş ve altıncı bölümlerinde paylaşılmıştır. Bu bölümlerde sağlanan bilgiler, hem mobil uygulama geliştirme ve Scrum yazılım geliştirme metodu özelliklerini hem de geliştirilen aracın yapısını ve özelliklerine ait bilgileri ortaya koyması açısından önemlidir.

Tez çalışmasının en önemli hedefi, çevik yazılım geliştirme metodunun mobil uygulama geliştirme sürecinde uygulanmasındaki zorluklar, kısıtlar ve ek gereksinimlerin bilgisi ışığında çevik yazılım geliştirme metodunun mobil uygulama geliştiricileri tarafından kullanımında rehber olmak olmuştur. Çevik yazılım geliştirme metodunun mobil uygulama geliştirmede kullanımında, önerilen metodun yüksek performans ve etkin kullanımı ile yüksek fayda sağlaması da tezin en önemli hedeflerinden biridir.

Çevik yazılım geliştirme metotları uygulandığında, geleneksel plan tabanlı yazılım geliştirme yöntemlerine göre görünen olumlu etkiler aşağıdaki gibi listelenebilir:

- Süreçteki görünürlüğün artması
- Teknik sorunlara erken tanı
- Sorumluluk paylaşımı
- Verimli bilgi paylaşımı

- Düşük hata yoğunluğu
- Sürekli geliştirme temposu

Mobil uygulama geliştirme platformlarında kapalı platformlardan açık platformlara geçiş ile birlikte mobil uygulama kullanımı ve üretimi artmıştır. Mobil uygulama geliştirme sürecinde kullanılan çevik yazılım geliştirme metotları, 4 farklı model halinde incelenmektedir. İncelemeler sonucunda modellerin içeriği ve alt adımları ortaya konulmuştur. Tez çalışmaları kapsamında modellere bağlı olarak, çevik yazılım geliştirme yöntemini uygulamayı kolaylaştıracak programlar test edilmiş ve bu programların yardımı ile mobil uygulama geliştirme sürecinde daha verimli sonuçlar elde edilebildiği görülmüştür.

Mobil uygulama geliştirme sürecinde kullanılan modellerden Mobile-D, mobil uygulama geliştirmede kullanılan diğer çevik yazılım geliştirme modelleri olan Hibrid Method Mühendisliği, Scrum ve Masam'dan önce geliştirilmiş ve bu modeller için referans olmuştur. Mobile-D'den sonra geliştirilen 3 model, Mobil-D'nin ayrıntılı analizi yapıp belli özelliklerinin geliştirilmesi ile ortaya çıkmıştır. Üretim sırasında, modeller için test ve erişim ortamları oluşturulmuştur.

Çevik yazılım geliştirme süreçlerinin mobil uygulama geliştirmedeki katkıları aşağıdaki gibidir;

- Uygulamaların sürekli kullanımını ve dayanıklılığını arttırma
- Çok hızlı değişen mobil uygulamaların yapısına adapte olmayı hızlandırma
- Risk yönetimi için daha gelişmiş öngörüye sahip olmayı mümkün kılma
- Mobil uygulama üretimindeki verimliliği arttırma
- Mobil uygulamalarda platform kısıtlarından kaynaklanan sorunları azaltma

Tezin final çalışması olarak geliştirilen araç, mobil uygulama geliştirme ve Scrum yazılım geliştirme metoduna ait var olan özelliklerinin geliştirilmesi ve kullanıcıların aracı kullanımı sonucu ihtiyaç duyulan özellik eklemelerinin yapılması halinde, yaygın kullanıma geçerek mobil uygulama geliştirme sürecinde hedeflediği katkıyı sağlayabilecek yapıdadır.

Yeni aracın geliştirilmesinin ardından, mobil uygulama geliştiricilerle yapılan görüşmelerde, aracın, mobil uygulama geliştiricilerin görüş ve beklentileri alındıktan sonra hedef kullanıcı kitlesi (mobil uygulama geliştiriciler) için özelleştirilerek geliştirilmiş olmasının ve kullanım kolaylığının getirdiği önemli avantajlara sahip olduğu görülmüştür.

Mobil uygulamalarda çevik yazılım geliştirme sürecinin yönetimini kolaylaştırıcı bir araç kullanmanın faydaları aşağıdaki gibidir;

- Daha etkili kararlar vermeye yardımcı olma
- Düşüncelerin, kullanıcı hikâyelerinin ve farklı kaynaklardan geribildirimlerin analizini ve görüntülenmesini kolaylaştırma
- Devam eden geliştirme süreci içerisinde planlamanın ve sonuç ölçümünün yapılabilme
- Uygulamanın iş akışını yönetmeyi kolaylaştırma ve her yerden proje üyeleri arasındaki iletişimi mümkün kılma
- Proje durumunun analizini kolaylaştırma



## KAYNAKLAR

- [1] Rahimian, V. ve Ramsin, R., (2007). "Designing An Agile Methodology for Mobile Software Development: A Hybrid Method Engineering Approach", development. In Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications , 337-341
- [2] Jeong, Y.-J., Lee, J. H. ve Shin, G.-S., (2008). "Development Process of Mobile Application SW Based on Agile Methodology", Advanced Communication Technology, ICACT 2008. 10th International Conference on, vol.1, no., pp.362-366, 17-20
- [3] Scharff, C. ve Verma, R. (2010). "Scrum to support mobile application development projects in a just-in-time learning context," In Proceedings of the 2010 ICSE Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '10), 25-31.
- [4] Çevik Yazılım Geliştirme Manifestosu, <http://agilemanifesto.org/iso/tr/>, 20 Aralık 2013
- [5] Vlaanderen, K., Jansen, S., Brinkkemper, S.ve Jaspers, E., (2010). "The Agile Requirements: Applying SCRUM Principles To Software Product Management" Refinery," Information And Software Technology ,53(2011) 58-70
- [6] Highsmith, J ve Cockburn, A., (2001). "Agile Software Development: The Business Of Innovation", IEEE Computer Society, 120-122
- [7] Role Of Software Architecture in Agile Software Development", <http://ruwandotnet.wordpress.com>, 22 Nisan 2013
- [8] Çevik Süreç, <http://www.ozcanacar.com>, 01 Aralık 2013
- [9] Abrahamsson, P., Salo, O., Ronkainen, J. ve Warsta, J., (2002). "Agile Software Development Methods", Espoo 2002, VTT Publications 478. 107 p.
- [10] C., Jhon F., "History Of Mobile Applications", Theory And Practice Of Mobile Applications, MAS 490-001 (2012)
- [11] Minelli, R., (2009). "Mobile Applications", ITU-T TechWatch Alert
- [12] Mobile App Types, <http://sajidmanzoor.wordpress.com/2013/04/30/mobile-apps-types/>, 30 Nisan 2013

- [13] Types Of Mobile Applications, [http://mobidev.biz/blog/types\\_of\\_mobile\\_applications.html](http://mobidev.biz/blog/types_of_mobile_applications.html), 10 Haziran 2013
- [14] H., Allan, S., Antonio ve T., Reza, (2009) "Mobile Platforms", CMI international conference on social networking and communities
- [15] Spataru, A. C., (2010). Agile Development Methods For Mobile Applications, Master Thesis, School Of Informatics University Of Edinburgh, Edinburgh.
- [16] Duru, I., Köse, E. ve Kalıpsız, O., (2012). "Mobil Uygulama Geliştirmede Çevik Yöntemin Uygulanması," in National Software Architecture Conference, UYMK'12, Paper 12
- [17] Duru, I., Köse, E. ve Kalıpsız, O., (2012). "Mobil Uygulama Sistemlerinde Çevik Yazılım Geliştirme Metodunun İncelenmesi," in Electric-Electronic And Computer Engineering Symposium, ELECO'12, Paper 199
- [18] Abrahamsson, P., Hanhineva, A., Hulkko, H., Ihme T., Jaalinoja, J., Korkala, M., Koskela, J., Kyllönen, P. ve Salo, O., (2004). "Mobile-D An Agile Approach for Mobile Application Development", OOPSLA'04, 174-175
- [19] Eclipse Process Framework Project (EPF), <http://www.eclipse.org/epf/>, 10 Aralık 2012
- [20] Duru, I. ve Kalıpsız, O., (2013). "Mobil Uygulamalarda Çevik Yazılım Geliştirme," in Agile Software Development Approach Workshop, ÇEYA'13, Paper 004
- [21] Trello, <https://trello.com/>, 29 Haziran 2013
- [22] Telerik TeamPulse, <http://www.telerik.com/agile-project-management-tools>, 29 Kasım 2013
- [23] Atlassian Jira, <https://www.atlassian.com/software/jira>, 10 Aralık 2013
- [24] Microsoft Team Foundation Server, <http://www.microsoft.com/visualstudio/tur/products/visual-studio-team-foundation-server-2012>, 14 Kasım 2013
- [25] Proje Hazırlama Ve Yönetme Rehberi, [http://www.isplaniproje.com/wp-content/uploads/2012/05/Proje\\_Haz\\_Rehberi.pdf](http://www.isplaniproje.com/wp-content/uploads/2012/05/Proje_Haz_Rehberi.pdf), Ekim 2008
- [26] Kolektif, (2013), Proje Yönetimi Bilgi Birikimi Kılavuzu, Beşinci Baskı, PMI TR, Ankara
- [27] Jira, <http://www.wikiturk.net/Madde/56873/JIRA>, 12 Haziran 2011
- [28] Team Foundation Server-Giriş, <http://www.csharpnedir.com/articles/read/?id=814>, 11 Aralık 2007
- [29] TFS Nedir?, <http://emretoptanci.blogspot.com/2010/09/tfs-nedir-team-foundation-server.html>, 2 Eylül 2010
- [30] Haumer, P., Eclipse Process Framework Composer, <http://www.eclipse.org/epf/general/EPFComposerOverviewPart2.pdf>, Nisan 2007

- [31] Baytam, V. ve Kalipsiz, O., (2011). "Scrum Yazılım Geliştirme Modeli Yönetim Aracı: ScrumMApp," in 5. National Software Engineering Symposium – UYMS'11

---

**ANKET SORULARI**

- 1) Ad-Soyad Bilgileriniz :
- 2) Çalıştığınız şirketin ismi nedir?
- 3) Çalıştığınız şirketin sektörü nedir?
  - a) Mobil uygulama geliştirme
  - b) Yazılım Geliştirme (Genel)
  - c) Eğitim (Yazılım Geliştirme vb.)
  - d) Bilgi Teknolojileri
  - e) Yazılım Tasarımı
  - f) Diğer
- 4) Yazılım geliştirici olarak kaç yıldır mobil uygulama geliştiriyorsunuz?
  - a) 1-5
  - b) 5-10
  - c) 10-20
  - d) 20-50
  - e) 50+
- 5) Organizasyonunuzun büyüklüğü nedir?
  - a) Küçük<50

b) Orta<X<250

c) Büyük>250

6) Hangi platformlar üzerinde mobil uygulama geliştirme yapıyorsunuz?

a) Android

b) Ios

c) BlackBerry

d) Windows Mobile

e) Symbian OS

f) Diğer

7) Organizasyonunuzdaki yazılım geliştirme ekibiniz kaç yıldır mobil uygulama geliştiriyor?

a) 1-5

b) 5-10

c) 10-20

d) 20-50

e) 50+

8) Şirketinizdeki mobil uygulama geliştiren takım üyesi sayısı?

a) 1-5

b) 5-10

c) 10-20

d) 20-50

e) 50+

9) Mobil uygulama geliştirirken kullandığınız ürün gereksinim listesinin özellikleri nelerdir?

a) Kullanıcı hikâyelerinden oluşma

- b) Önceliklendirilme
- c) Risk içirme
- d) İş Değeri içirme
- e) Ürün sahibi için katkı yetkisi
- f) Diğer

10) Mobil uygulama geliştirmede kullanılacak bir proje yönetim aracı hangi özellikleri içermelidir?

- a) İş Durumu
- b) Tüm Platformlarda Çalışma
- c) Takip Edilebilirlik
- d) Müşteri Profili
- e) Diğer

11) Scrum'dan önce kullandığınız yazılım süreçleri hangileridir?

- a) Spiral
- b) Waterfall
- c) Unified
- d) Process
- e) Prototip
- f) Diğer

12) Scrum'ı seçmenizdeki ana etken nedir?

- a) Üretim hızını arttırması
- b) Değişikliğe hızlı yanıt vermeyi kolaylaştırma
- c) Kaliteyi arttırması
- d) Projeleri zamanında bitirmeyi sağlama
- e) Diğer

13) Scrum yazılım geliştirme metodolojisinin uygulanmasında sizin için hangi özellikler önemlidir?

- a) Kolay kullanılabilirlik
- b) Esneklik
- c) Takip Edilebilirlik
- d) Öğretici Bilgiler
- e) Diğer

14) Mobil uygulama geliştirme göz önünde bulundurulduğunda scrum özelliklerinden hangileri çok önemlidir?

- a) Koşu Planlama Toplantısı
- b) Günlük Scrum Toplantısı
- c) Koşu Gözden Geçirme Toplantısı
- d) Koşu Kapatma Toplantısı
- e) Gereksinim Güncelleme
- f) Diğer

15) Mobil uygulama geliştirme süreci yönetimi için özel bir aracın süreç yönetimini, ürün kalitesini ve üretim hızını arttıracığını düşünüyor musunuz?

- a) Evet
- b) Hayır

16) Mobil uygulama geliştirmede Scrum sürecinin yönetiminde kullandığınız özel bir araç var mı?

- a) Evet
- b) Hayır

17) Mobil uygulamalarda yazılım geliştirmede sürecini takip etmek için hangi araçları kullanıyorsunuz?

- a) Trello

- b) Telerik TeamPulse
- c) Eclipse Epf
- d) Verison One
- e) Pivotal Tracker
- f) Jira
- g) Microsoft Team Foundation Server
- h) Diğer

18) Mobil uygulamalarda yazılım geliştirme için kullanılan araçların eksik özellikleri nelerdir?

- a) Raporlama
- b) İzin ve Çalışma Günleri
- c) Planlama ve Zamanlama Yönetimi
- d) İzleme Ve Analiz
- e) İlerleme Takibi
- f) Fikirler ve Geri Besleme
- g) Diğer

19) Çevik yazılım geliştirmenin mobil uygulama geliştirme sürecine en fazla hangi alanda katkı sağlayacağını düşünüyorsunuz?

- a) Yeniliklerin projeye uygulanması
- b) Müşterilerin başlangıçta gereksinimlerini belirlemesi
- c) Müşterilerin gereksinimlerinin sık değişmesi
- d) Projelerin yönetiminin karmaşık hale gelmesi

20) Mobil uygulamalarda proje yönetiminde kullanılan metodolojilerden hangisinin sizin ekibiniz için uygun olduğunu düşünüyorsunuz?

- a) Mobile-D

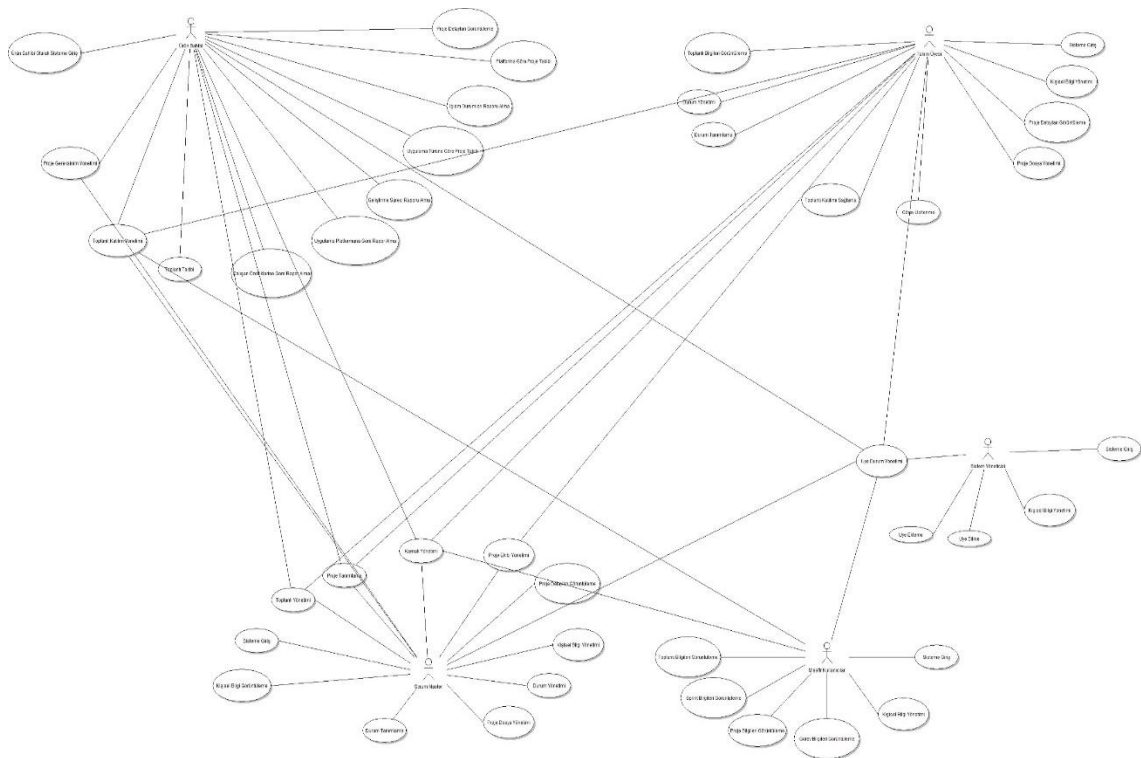


- b) Scrum
- c) Hibrid Method Mühendisliđi Yöntemi
- d) Masam Yöntemi
- e) Diđerleri

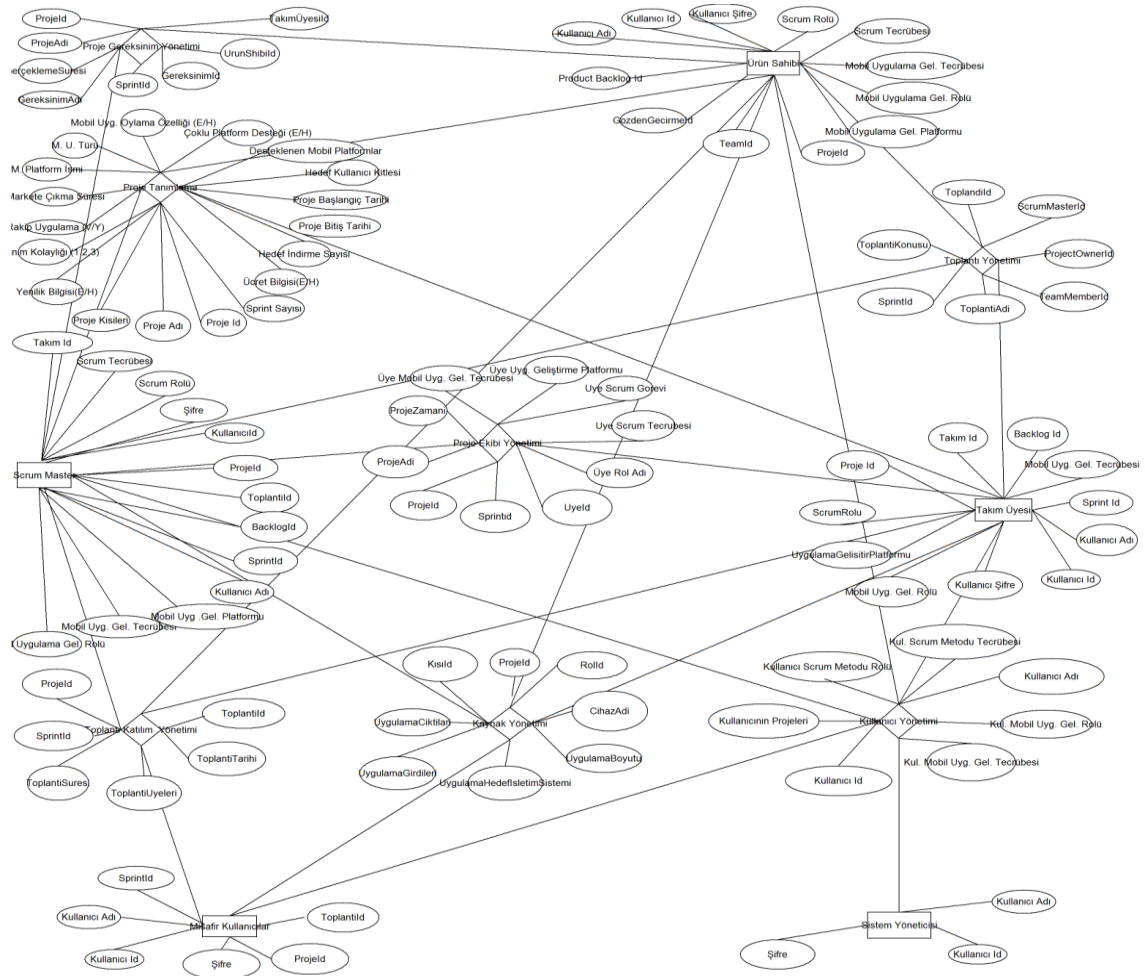
21) Mobil uygulamalar için proje yönetim aracı gerçekleştiriyor olsanız aşağıdaki özelliklerden hangilerini eklerdiniz?

- a) Planlama
- b) Ekip Üyeleri Yönetimi
- c) Dađıtık ekip yapısını listeleme
- d) Kullanıcı Senaryosu Durumu
- e) Görev Tahtası Ekranı
- f) Diđer

**AYRINTILI KULLANICI DİYAGRAMI**



## AYRINTILI VARLIK İLİŞKİ MODELİ



## ÖZGEÇMİŞ

---

### KİŞİSEL BİLGİLER

**Adı Soyadı** : İsmail DURU  
**Doğum Tarihi ve Yeri** : 01.04.1989, Şişli  
**Yabancı Dili** : İngilizce  
**E-posta** : [iduru@yildiz.edu.tr](mailto:iduru@yildiz.edu.tr)

### ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Müh.	İstanbul Üniversitesi	2011
Lise	Fen Bilimleri	Ümraniye Atakent Lisesi	2006

### İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2012	Y.T.Ü.	Araştırma Görevlisi
2010	Bits Bilgisayar	Software Developer (Intern)
2010	Turkcell Teknoloji	Software Developer (Intern)

2010	Teksan Generator	Software Developer (Intern)
2011	Bimeks	Hardware Supporter (Intern)

#### **YAYINLARI**

##### **Ulusal Bildiri**

- 1.(2012) **Duru I.**, Kose E., Kalipsiz O., Mobil Uygulama Sistemlerinde Çevik Yazılım Geliştirme Metotunun İncelenmesi, ELECO, 2012
- 2.(2012) **Duru I.**, Kose E., Kalipsiz O., Mobil Uygulama Geliştirmede Çevik Yöntemin Uygulanması, UYMK, 2012
- 2.(2012) **Duru I.**, Gelişen İnternet Ortamında Etkin Zaman Kullanımı, Değişimlere Ve Sosyal Ağlara Adapte Olma, ŞEHİRLERİN YAPILANDIRILMASI SEMPOZYUMU, 2012
- 2.(2013) **Duru I.**, Kalipsiz O., Mobil uygulamalarda Çevik Yazılım Geliştirme, ÇEYA, 2013

##### **Uluslararası Bildiri**

- 1.(2013) **Duru I.**, Akcay B., Furlang- Social Networking Infrastructure For Teaching Languages, ICQH, 2013