

**T.C.  
BAHÇEŞEHİR ÜNİVERSİTESİ**

**SCRUM UYGULAMASI  
İÇİN BİR YÖNETİM ARACI  
GELİŞTİRİLMESİ**

**Yüksek Lisans Tezi**

**OLCAY SÖNMEZ**

**İSTANBUL, 2019**



**T.C.  
BAHÇEŞEHİR ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ  
MÜHENDİSLİK YÖNETİMİ**

**SCRUM UYGULAMASI  
İÇİN BİR YÖNETİM ARACI  
GELİŞTİRİLMESİ**

**Yüksek Lisans Tezi**

**OLCAY SÖNMEZ**

**Tez Danışmanı: DR. ÖĞR. ÜYESİ ADNAN ÇORUM**

**İSTANBUL, 2019**

T.C.  
BAHÇEŞEHİR ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ  
MÜHENDİSLİK YÖNETİMİ

Tezin Adı: Çevik proje yönetim metodolojilerinden Scrum çerçevesinin kullanılması ve yazılım çözümünün oluşturulması

Öğrencinin Adı Soyadı: Olcay Sönmez  
Tez Savunma Tarihi:

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğu Fen Bilimleri Enstitüsü tarafından onaylanmıştır.

Dr. Öğr. Üyesi Yüce Batu SALMAN  
Enstitü Müdürü  
İmza

Bu tezin Yüksek Lisans tezi olarak gerekli şartları yerine getirmiş olduğunu onaylarım.

Doç. Dr. Gül Tekin TEMUR  
Program Koordinatörü  
İmza

Bu Tez tarafımızca okunmuş, nitelik ve içerik açısından bir Yüksek Lisans tezi olarak yeterli görülmüş ve kabul edilmiştir.

\_\_\_\_\_ Jüri Üyeleri

\_\_\_\_\_ İmzalar

Tez Danışmanı  
Dr. Öğr. Üyesi Adnan ÇORUM

.....

Üye  
Dr. Öğr. Üyesi Betül ERDOĞDU ŞAKAR

.....

Üye  
Doç. Dr. Özlem ŞENVAR

.....

## TEŐEKKÜR

Bütün eđitim yıllarım boyunca beni hep destekleyen sevgili aileme teőekkürlerimi beyan ederim ve tez sürecinde desteđini esirgemeyen sevgili danıőman hocama Sayın Dr. Adnan Çorum'a teőekkürlerimi en içtenlikle sunuyorum.

İstanbul, 2019

Olçay SÖNMEZ



## ÖZET

### ÇEVİK PROJE YÖNETİM METODOLOJİLERİNDEN SCRUM ÇERÇEVE'UNUN KULLANILMASI VE YAZILIM ÇÖZÜMÜNÜN OLUŞTURULMASI

Olca SÖNMEZ

Mühendislik Yönetimi

Tez Danışmanı: Dr. Öğr. Üyesi Adnan ÇORUM

Ocak 2019, 41 sayfa

Globalde faaliyet gösteren firmalar iş süreçlerinde daha başarılı olmak adına birçok proje yönetim metodolojilerini kendi şirketlerinde uygulamaktadır. Bu proje yönetim metodolojilerinden en yaygın olanı şüphesiz çevik metodolojidir. Çevik olabilmek adına kullanılan araçlardan en başarılı olan ise Scrum çerçevesidir. Scrum çerçevesinin firmalar ve şahıslar tarafından daha iyi anlaşılması, hızla adapte olunabilmesi ve kullanılabilmesi bu tezin amaçlarından biridir. Tez kapsamı boyunca geliştirilmiş mobil uygulama ile daha interaktif, kullanıcı dostu ve anlaşılır bir süreç hedeflenmektedir. Mobil uygulama tercihi, günümüzde mobil dünyanın gelişen dinamikleri göz önünde bulundurularak belirlenmiştir.

**Anahtar Kelimeler:** Scrum Çerçeve, Çevik Metodoloji, Mobil Yazılım, İnteraktif, Süreç

## ABSTRACT

### THE WAY HOW TO USING SCRUM ÇERÇEVE AND CREATING A SOFTWARE PRODUCTION

Olcay SÖNMEZ

Engineering Management

Thesis Supervisor: Assist. Prof. Dr. Adnan ÇORUM

January 2019, 41 pages

Nowadays, all types of Company have their own project process and one of them is the Scrum Methodology. The purpose of this work is that make the Process more easy, understandable and more interactive so the benefit would save more time at work. The base of this project is to make all the Process clearer and giving fast answers. The Software Production for Scrum Framework is a Mobile Project. The usability of the mobile is getting increased and it will be more in the future so the reason why to making this project as a Mobile Production is fast interaction which saves the time at work.

**Keywords:** Scrum Framework, Agile Methodology, Mobile Software, Interaction, Process

## İÇİNDEKİLER

ŞEKİLLER .....	viii
1. GİRİŞ .....	1
2. LİTERATÜR TARAMASI .....	2
3. ÇEVİK PROJE YÖNETİM NEDİR? .....	4
3.1 ÇEVİK YAZILIM GELİŞTİRME TEMEL DEĞERLER .....	4
3.2 ÇEVİK PROJE YONETİM ÇERÇEVESİNDE ÖNEMLİ ADIMLARI ..	4
3.3 ÇEVİK YÖNETİMİN PROJELER ÜZERİNDEKİ ETKİLERİ.....	5
3.4 ÇEVİK PROJE YONETİMİN KULLANIM NEDENLERİ .....	6
3.5 ÇEVİK PROJE YÖNETİMİN METODOLOJİLERİ .....	7
3.6 SCRUM ÇERÇEVE NEDİR?.....	7
3.7 SCRUM ÇERÇEVE DEĞERLERİ.....	8
3.8 SCRUM KONTROL UYGULAMASI.....	9
3.8.1 Şeffaflık .....	9
3.8.2 Gözlem.....	10
3.8.3 Adaptasyon .....	10
3.9 SCRUM DEĞERLERİ .....	10
3.9.1 Cesaret.....	11
3.9.2 Odak .....	11
3.9.3 Taahhüt .....	11
3.9.4 Saygı .....	11
3.9.5 Açıklık .....	11
3.10 SCRUM TAKIMI.....	12
3.10.1 Ürün Sahibi (Product Owner).....	12
3.10.2 Geliştirme Takımı (Development Team) .....	13
3.10.3 Scrum Master .....	14
3.11 SCRUM TAKIMIN ÇALIŞMA AKIŞI .....	14
3.11.1 Sprint.....	15
3.11.2 Sprint Planlama.....	15
3.11.3 Günlük Scrum .....	16
3.11.4 Sprint Değerlendirme (Sprint Review) .....	16
3.11.5 Sprint Retrospektif (Sprint Retrospective).....	17
4 SCRUM ÇERÇEVE AIT GELİŞTİRME ORTAMI.....	18
4.1 MOBİL İŞLETİM SİSTEMLERİ .....	18

4.2	ANDROİD İŞLETİM SİSTEMİ.....	18
4.3	ANDROİD UYGULAMA GELİŞTİRMEK.....	18
4.3.1	Android Yazılım Geliştirme Kit (SDK).....	19
4.3.2	Android Debug Bridge.....	19
4.3.2	Android Derleme Yapısı .....	19
4.4	ANDROİD'DE ÖZEL DERLEME YAPILANDIRMASI .....	21
4.5	ANDROİD SAYFALARIN YAŞAM DÖNGÜSÜ.....	22
4.6	FIREBASE.....	25
4.7	FIREBASE GERÇEK ZAMANLI VERİTABANI.....	26
4.8	SCRUM TAKIMIN SÜREÇLERDEKİ ORTAK YORUM TANIMLANMASI.....	27
4.9	SCRUM TAKIMININ PROJE UYGULAMASINDAKİ YETKİNLİKLERİ.....	28
4.9.1	Proje Yöneticisi.....	28
4.9.2	Development Takımı.....	28
4.9.3	Scrum Master .....	29
5	SCRUM MOBİL UYGULAMANIN KULLANIM KILAVUZU .....	29
5.1	FIREBASE – REALTIME KURULUMU .....	30
5.2	FIREBASE ÜZERİNDEN KULLANICI GİRİŞ İÇİN YETKİNLİK VERME.....	31
5.3	UYGULAMA GİRİŞ SAYFASI .....	32
5.4	UYGULAMADAKİ PROJE LİSTESİ VE PROJE OLUŞTURMAK.....	33
5.5	SPRINT LİSTESİ.....	37
5.6	SPRINT DETAY SAYFASI.....	38
6	SONUÇ VE TARTIŞMA.....	40
	KAYNAKÇA .....	41

## ŞEKİLLER

Şekil 3.1: Çevik Yöntem Projelerdeki Kullanılabilirliği.....	5
Şekil 3.2: Scrum Çerçeve Değerleri Şematik Görünümü .....	7
Şekil 3.3: Scrum Çerçeve Kontrol Mekanizması.....	8
Şekil 3.4: Scrum Değerleri.....	9
Şekil 3.5: Scrum Takımı .....	11
Şekil 3.6: Scrum Yaşam Döngüsü .....	13
Şekil 4.1: Android Derleme Süreci .....	18
Şekil 4.2: Fragman Yaşam Döngüsü.....	22
Şekil 4.3: Etkinlik Sınıfın Yaşam Döngüsü .....	22
Şekil 4.4: Scrum Süreçlerinde Kullanılan Ortak Dil.....	26
Şekil 5.1: Android Studio Yer Alan Paket Adını Firebase Tanımlama .....	30
Şekil 5.2: Google Dosyasını Android Studio Eklenmesi.....	30
Şekil 5.3: Kullanıcı Firebase Gerçek Zamanlı Veri Tabanı.....	31
Şekil 5.4: Uygulamanın Giriş Sayfası.....	32
Şekil 5.5: Uygulama Proje Liste Sayfası.....	34
Şekil 5.6: Proje Ekleme Sayfası .....	35
Şekil 5.7: Projeye ait sprint eklemek.....	36
Şekil 5.8: Sprint liste ekran görüntüsü .....	38
Şekil 5.9: Sprint Detay Sayfası .....	39

## 1. GİRİŞ

Geçmişten bugüne kadar yapılan Proje Yönetim arařtırmaların sebepleri her zaman daha iyi sonuçlar elde etmeyi amaçlanmaktadır. Giderek artan karmařık iř yařamı daha fazla karmařık süreçlerle karşı karşıya olacađımız ve dolayısıyla daha çok yeni metodolojilerin ortaya çıkacađına anlayabiliriz. Bunlardan biri de günümüzde ön planda olan Çevik Proje yönetimidir. Çevik yönetimin oluşumu bazı dar bođazlı projelere çözüm üretebilmek amacıyla ortaya çıkmıřtır.

Çevik yazılım geliştirme, ihtiyaçların ve çözümlerin kendi kendini organize eden çapraz iřlevli ekipler arasındaki iř birliđiyle evrimleřtiđi, yinelemeli, geliřtirmeye dayalı bir çevik proje yönetimidir.

Günümüzde bazı firmalar her ne kadar Çevik Proje Yönetimini duymuř olsa da uygulamaya geldiđinde herhangi bir giriřimde bulunmamaktadır. Bunların başlıca sebepleri var olan süreçlere adapte olmasıdır dolayısıyla deđiřtirmek ile zam harcamamak istenmemesidir veya mali açıdan bazı çevik proje yönetim yazılım araçlarını karşılayamamasıdır.

Bu tez çalışmasında, herkesin kullanabileceđi, hızlı entegre olabileceđi ve anlayabileceđi bir mobil uygulama geliřtirildi. Çevik proje yönetimi ve bu yönetime ait olan Scrum Metodolojisi tanımı, özellikleri ve kullanılma şekillerinden ikinci bölümde bahsedilmiřtir. Bu proje yöntemin yazılım çözümü Mobil Uygulama olarak geliřtirilmiř olup ve gerçek-zamanlı veri tabanı kullanılarak hızlı etkileřimi son kullanıcıya kadar ulařtırabilmeyi hedeflerinden, kullanılan yazılım tekniklerinin sebeplerini tezin üçüncü bölümde bahsedilmiřtir. Yapılmıř bu Mobil uygulamanın kullanım şekillerinden bahsedilmiřtir ve son olarak sonuçlarından en son bölümde bahsedilmiřtir.

## 2. LİTERATÜR TARAMASI

Scrum Çerçevesi tez araştırması olarak alan birçok kurum konu edinmiştir. Fakat yapılan araştırmalara baktığımızda Scrum çerçevesi ve mobil uygulamayı ortak bir payda da baz alınmadığı gözlemlenmiştir. Yapılan literatür araştırmaları daha çok Scrum çerçevesinin proje geliştirme süreçlerindeki etkileri araştırılmış veya web, masaüstü uygulamalar geliştirilmiştir. Günümüzde birçok global firmalar Scrum çerçevesinin bir ürünü olarak yazılımsal ürünler geliştirmişlerdir.

Öncelikle bu tez çalışmasına yakın tezleri incelediğimizde Talin Üniversitesinin “Applying Agile Methodologies to Design and Programming” (Tatjana Pavlenko, 2012) tarafından hazırlanan bu çalışmada Sigma firmasına Scrum Çerçevesinin efektif kullanabilmeyi ve takıma katılacak yeni üyeye hızlıca adapte olabileceği bir ortam hazırlamayı hedeflemişlerdir. Scrum Yönetim aracı olarak web yazılım ürünü üretmişlerdir ve elde ettikleri sonuç ise takımlarda artan iletişim ve kendi kendilerini yönetebildiklerini etkin bir Scrum yaklaşım ara yüzünün olması gelişen bu durumu tetiklediğini gözlemlemişlerdir.

Globalde Scrum Çerçevesi için yapılan uygulamalara incelemek istediğimizde aslında en önemlileri arasında jira gelmektedir. Jira, Atlassian tarafından üretilmiş bir proje yönetim aracıdır. Günümüzde jira kullanım oranına bakıldığında yaklaşık 75.000 müşterisi mevcut. Web, mobile yazılım ürünleri bulunmaktadır.

Bu araştırmalara bakıldığında tez çalışması olarak direkt olarak Scrum çerçevesi Mobil bir uygulamada incelenmemiştir. Türkiye’ de yapılan Scrum Çerçevesi için tez araştırmalara bakıldığında proje çalışmalardaki Scrum çerçevesinin kullanılarak etiklerinden bahsedilmiştir. Bu araştırmaya örnek olarak Bahçeşehir üniversitesinde yapılan “Türkiye’deki bankacılık uygulamalarında çevik (Scrum) yazılım geliştirme metodolojisi kullanışlı mı?” incelediğimizde (Murat Şahin, 2016) Bankacılık sektöründe Scrum çerçevesinin kullanılması uygunluğu araştırılmıştır.

Genel olarak literatür taramasına bakıldığında tespit edilen açıklık Türkiye de başlatılmış Scrum Çerçeveyi temelini baz alarak bir gerçek zamanlı mobil uygulama geliştirilmemiştir. Bu tez çalışmada gerçek zamanlı bir mobil uygulama üretilerek Scrum çerçeveyi efektif kullanımı artıracığını hedeflemektedir.



### 3. ÇEVİK PROJE YÖNETİM NEDİR?

Çevik Proje Yönetimi, bir metodoloji değildir bir yaklaşım olup bu yaklaşımdan yararlanılarak bazı metotlar oluşturulur. Temelinde yatan bu yaklaşımda projeyi küçük parçalar halinde oluşturup bu parçaları teker teker incelemeyi sağlamaktadır. Bu şekilde bileşenlere ayırmak aslında proje daha kolay ve tekrarlı bir şekilde incelemeyi, test edebilmeyi ve geri bildirimlerle projenin esnek süreçlere sahip olduğu anlamına gelir.

Genel bir tanım yapmak gerekirse, projeyi aşamalara bölerek geliştirmesinin her aşamasını öngörmek ve süreçlerde esnek olmayı ve kullanıcılarla iletişimde olmasını sağlayan bir yönetim biçimidir.

#### 3.1 ÇEVİK YAZILIM GELİŞTİRME TEMEL DEĞERLER

Çevik yönetimin tarihsel gelişimine bakacak olursak bir grup Yazılım Geliştirme uzmanlar tarafından oluşmuş ve sonucunda çevik proje yönetimine temelinde yatan 4 önemli değer meydana gelmiştir. Bunlar;

- I. Süreçlerde ve kullanılan araçlar üzerinde bireylerin etkileşimi,
- II. Son kullanıcılarla sürekli etkileşim de olmak,
- III. Dokümantasyondan ziyada çalışan bir sonuç elde etmek,
- IV. Değişime açık olmak

Sayıdığımız bu özellikler aslında çevik yönetimi temelini anlamamız açısından çok önemlidir. Yukarıda saydığımız bu 4 temel değerler ilişkili uygulamalı seriler dönüştürülebilir ve böylelikle sürekli kullanıcılarla karşılaştırmaya başladığımızda projenin riski ve yinelemeli çalışmalarını azaltmış oluruz peki bu uygulamalar nelerdir? Bir sonraki başlığımızda “çevik proje yönetim çerçevesinde önemli uygulamaları nelerdir?” başlığında kapsamlı incelenmiştir.

#### 3.2 ÇEVİK PROJE YONETİM ÇERÇEVESİNDE ÖNEMLİ ADIMLARI

Çevik proje yönetim dünyasında fazla değişkenlikler ve detaylar bulunmaktadır fakat çoğunlukla görebileceğiniz en önemli prensipleri ve uygulaması aşağıda sıralanmıştır;

- I. Yinelemeli ve Artan Gelişime: Bu özellik birçok şekle dönüşebilir (Sprinter, Rapid Prototip). En önemli faktör burada son kullanıcılara erken teslim edebilmek. Bu bize önceden inceleme ve geri bildirimlere olanak tanır.

Geliştiriciler bir sonraki aşamaya geçmeden önce yapılacak olan yorumlarda amaca açık olması gerekmektedir.

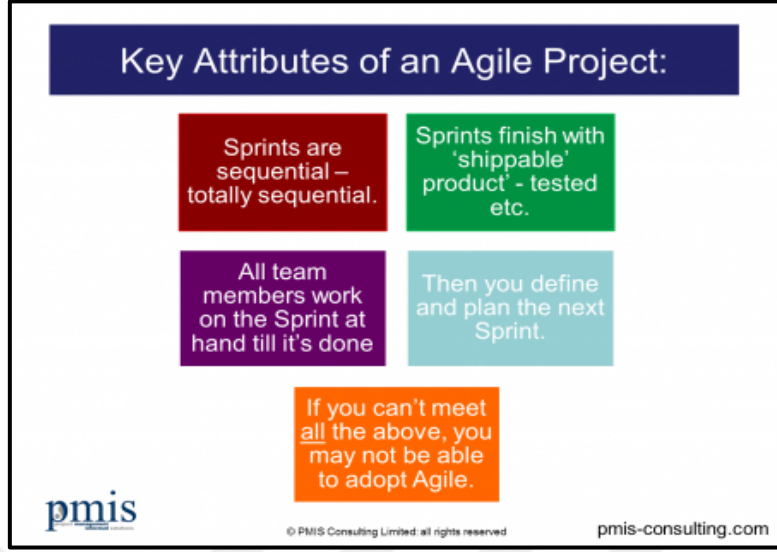
- II. Efektif Takım kurmak: Gerçekte çevik projenin çekirdeğinde etkili bir takım yatar. “Onlar ve bizler” gibi organizasyonel yapının etkisini azaltıp geliştiriciler, kullanıcılar ve iş yöneticilerle etkileşimi kolaylaştırmak gerekir. Bunun önemli yaraları çok açıktır ve projenin yoldan çok uzaklaşmamasını sağlar.
- III. Etkili iletişim: Çevik yönetim her zaman yüz yüze iletişimi destekler. Bu prensip her ne kadar zorlayıcı olsa da prensibi pek çok şekilde genişletebilir ve geliştirilebilir.

### **3.3 ÇEVİK YÖNETİMİN PROJELER ÜZERİNDEKİ ETKİLERİ**

Çevik proje yönetimi aslında yazılım projelerde hayat bulmuştur ve dolayısıyla çevik yönetim yazılım projelerinde en uygun yaklaşımdır. Çevik yönetim özellikle beklentilerin veya gereksinimlerin belirsiz olduğu projelerde sıkça kullanılabilen bir yaklaşımdır. Bazı kullanıcılar veya işletmeciler gereksinimleri tanımlamada çok zayıflardır dolayısıyla çevik yönetimin bu konuda çözüm üretmesi zorlandığı tek konudur.

Ancak, gereksinimler belirsiz olması şart anlamına gelmemektedir aksine belirli olması da bir avantajdır ve bunun beraberinde geliştiriciler ve kullanıcıların etkileşim halinde çalışmasını artırmaktadır.

**Şekil 3.1: Çevik Yöntemin Projelerdeki Kullanılabilirliği**



*Kaynak: pmis-consulting.com*

### **3.4 ÇEVİK PROJE YONETİMİN KULLANIM NEDENLERİ**

Çevik proje yönetimin günümüzde en yaygın kullanılan bir proje yönetim biçimidir. Sürekli ve hızla gelişen pazar ve teknoloji ihtiyaçları karşılamak için çevik proje yönetimin kullanımdan geçer. Çevik Proje yönetimin başlıca kullanım nedenleri;

- I. Çevik yönetim global bir yaklaşımdır ve takımlara öğrenme fırsatı verir. Bu fırsat çevik yönetimin tekrarlı olma prensibi yatar.
- II. Ekibin bir taslak oluşturmasını sağlar ve her bir sprintin geliştirmesini sağlar.
- III. Süreç de zaman kazanımını sağlar.
- IV. Çevik yönetimin takım çalışması prensibi proje vizyonunu geliştirir.
- V. İşteki öncelikleri belirlemede yardımcı olur.
- VI. Projedeki değişiklikleri ön tahminlerde bulunabilir.
- VII. Gelen geri bildirimlerle projedeki düzenlemeleri hızlandırır.
- VIII. Ekipler bir sonraki çözüme veya sürece bir taslak oluşturabilir.
- IX. Her döngü parçası ve yenileme sonrası hızlı geri bildirimler alınır.
- X. Ekibin dinamiğini artırır.

### 3.5 ÇEVİK PROJE YÖNETİMİN METODOLOJİLERİ

Çevik proje yönetimi en sık duyulan ve bilinen metodolojiler;

- I. XP (Extreme Programlama)
- II. Yalın Prensipleri ve araçları
- III. Kanban Prensipleri ve araçları
- IV. FDD (Feature- driven Development)
- V. DSDM (Dynamic Systems Development Method)
- VI. MDD (Model-driven Development Method)
- VII. DAD (Disciplined Agile Delivery)
- VIII. TDD (Test-driven Development)
- IX. BDD (Behaviour-driven Development)

Yukarıda saymış metotlar çevik proje yönetime aittir. Bir de sürekli yanlış bilinen bir Scrum Çerçeve mevcuttur. Scrum Çerçeve yukarıdaki maddeler arasında yer almaz çünkü bir metodoloji değil bir yöntemdir. Bunun nedeni raporlama, kaynak kodlamanın nasıl olacağını izah etmek yerine daha çok yinelemeli modeller, ön görüş ve risk analizi yapmaya yöneliktir.

### 3.6 SCRUM ÇERÇEVE NEDİR?

Scrum 1993 yılında Jeff Sutherland tarafından geliştirildi ve hedefi geliştirme ve yönetim metotları çevik proje yönetim prensipleri doğrultusunda oluşturmaktır. Scrum kompleks yapıdaki proje içerisinden üretken ve yaratıcı projelerini sunabilecekleri bir Çerçeve' tur ve bu Scrum Çerçevenin anlaşılması kolay fakat uzmanlaşmayı zorlaştırır.

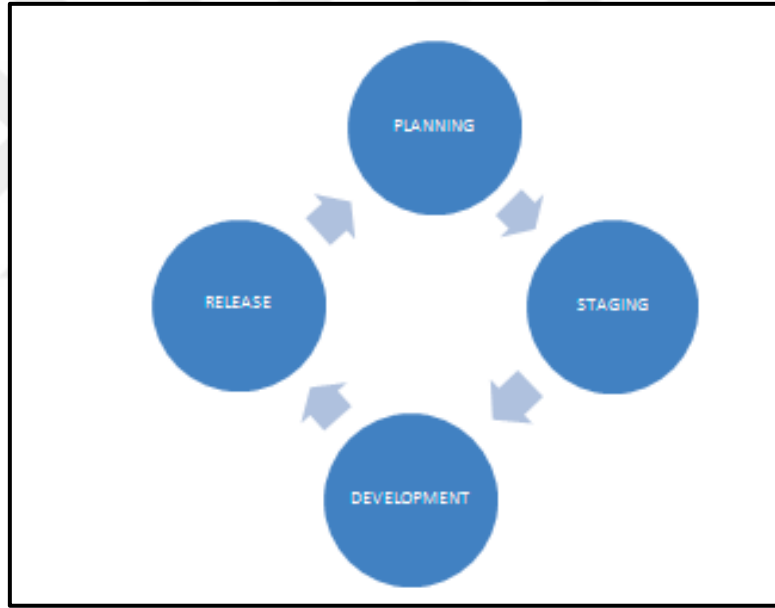
Scrum, süreçleri kompleks olan projeleri yönetmek için oluşturulmuş bir Çerçeve' tur. Kesinlikle süreç, teknik veya bir metot değildir aksine birçok süreç geliştirilebilen ve teknikler oluşturulabilen bir Çerçevedir. Scrum, iş tekniklerini proje yönetimini anlaşılır ve temiz bir sürece dönüştüren ve böylelikle her bir ürünlerini sürekli olarak geliştirir ve aynı zamanda iş – takım atmosferini de geliştirir.

Scrum Çerçeve Scrum takımlarından oluşur ve belirli roller atanır. Çerçeve içerisinde her bir bileşen belirli bir amaca hizmet etmeli ki Scrum süreci başarılı bir şekilde ulaştırabilmelidir. Scrum bünyesinde oluşan roller birbiriyle bağımlıdır.

### 3.7 SCRUM ÇERÇEVE DEĞERLERİ

Scrum geliştirme sürecinde takım çalışmasının eforları önemli bir unsurdur. Takım eforlarında geliştirme sürecin sürdürülebilir ve hızlı bir şekilde son kullanıcıya ulaştırılması önem arz etmektedir. Bir kere projenin gereksinimleri belli olduktan sonra, Scrum geliştirme süreçleri yinelemeli numaralara bölünür. Scrum geliştirme sürecin yaşam döngüsü bu 4 evreden oluşmaktadır:

**Şekil 3.2: Scrum Çerçeve Değerleri Şematik Görünümü**



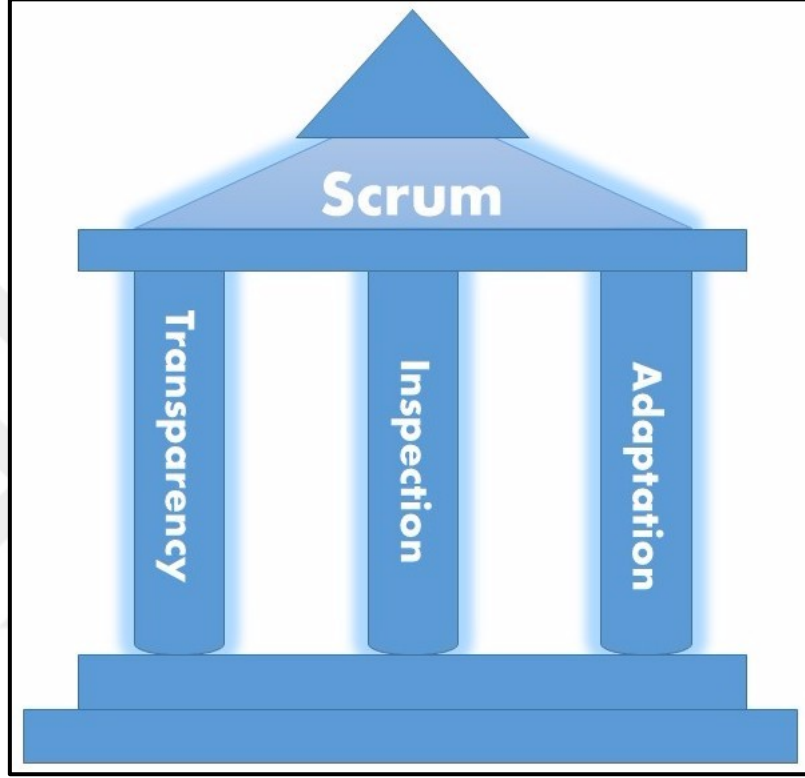
*Kaynak: The International Journal Of Engineering And Science (Ijes)*

- I. **Planlama:** Projenin planlama sürecidir. Bu evrede projenin gereksinimleri ve maliyeti belirlenir.
- II. **Kademelendirme (Staging):** Projenin gereksinimleri belirlendikten sonra Staging evresinde süreçleri önem dereceleri belirlenir.
- III. **Geliştirme (Development):** Projenin geliştirme evresidir. Önem dereceleri belirlenmiş olan süreçlere göre geliştirme yapılmaya başlanır.
- IV. **Yayınlama (Release):** Projenin veya bir sürecin yayına geçtiği evre

### 3.8 SCRUM KONTROL UYGULAMASI

Scrum Çerçevesinin temelinde deneycilik söz konusudur. Scrum de yer aşan yinelemeli ve artırımı yaklaşım risk kontrol denetimini yapmaktır ve 3 prensipten oluşmaktadır.

**Şekil 3.3: Scrum Çerçeve Kontrol Mekanizması**



*Kaynak: masterofproject.com/*

#### 3.8.1 Şeffaflık

Proje süreçlerinde sorumlu olacak kişiler tarafından belirlenmiş ortak tanımlamalar getirmelidirler. Örneğin; proje süreçlerinin başlangıç ve bitiş noktalarına bir tanım getirilecek olunursa proje başlangıcı için “In Progress” proje bittiğinde ise “Done” gibi kavramlar tanımlandığında nelerin takibi yapılıp yapılmayacağı sorumlu kişilerde tarafından belirlenir ve bir sonraki sürecin hızlandırmasında etkisi olacaktır. Projenin gereksinimler açık ve anlaşılır olması projenin teslimat süresini hızlandıracaktır.

### 3.8.2 Gözlem

Scrum uygulayan sorumlu kişiler tarafından Sprint amacı doğrultusunda sıkça gözlemlenmelidir ki Sprint esnasında oluşabilecek hataları düşürebilesinler fakat iş sürecini aksatacak kadar çok sık olmamalıdır.

### 3.8.3 Adaptasyon

Süreç esnasında hedef dışına çıkacak olacak herhangi bir durum bir gözlemci tarafından saptanırsa süreci düzenleyecek oturumlar düzenlenmelidir ve düzeltmeyi en kısa sürede yapılmalıdır ki süreçte daha fazla saptama gerçekleşmesin.

Yapılacak olan bu oturumlar daha sonra açıklamalarda anlatılacağı üzere aşağıdaki gibidir;

- I. Sprint Tanımlama
- II. Günlük Scrum (Daily Scrum)
- III. Sprint Değerlendirme (Sprint Review)
- IV. Sprint Retrospektif (Sprint Retrospective)

## 3.9 SCRUM DEĞERLERİ

Şekil 3.4: Scrum Değerleri



Kaynak: [scrumguides.org](http://scrumguides.org)

Scrum Çerçeve de yer alan 5 temel değer vardır. Bunlar;

### **3.9.1 Cesaret**

Scrum takımlarının kendi kendini örgütleyen (self-organizing), risk almaktan çekinmeyen, çözüm odaklı takımlar olması beklenmektedir. Bunların olabilmesi için scrum takımının, organizasyonun cesur olması ve takımların doğru işler yapabilmesi adına organizasyonun da kendilerine bu alanı yaratması ve gerekli güveni vermeleri gerekir.

### **3.9.2 Odak**

Sprintleri belirlemek için düzenlenen toplantılarda oluşturulan her bir sprintin hedeflerine odaklanması ve başka işlerin önemini arttırıp sprintin odağı değiştirilmemelidir. Eğer sprinti bırakıp başka bir işle başlanırsa hiçbir başarılı sonuca ulaşılamaz. Bu odağı korumak adına Scrum Master rolü büyük bir pay düşmektedir, takımı dış etkenlerden koruması ve sprint hedefine ulaştırılması sağlamalıdır.

### **3.9.3 Taahhüt**

Sprint belirleme toplantısında scrum takımı, sprint sonunda teslim edilecek olan özellikleri üstlenirler. Burada scrum takımının tam olarak anladığı, soru işaretleri barındırmayan işleri üstlenmesi ve sprint boyunca söz verilen işlerin tamamlanması için çalışması beklenmektedir.

### **3.9.4 Saygı**

Bir takımda yer alan üyeler aynı yeteneklere veya deneyime sahip değildirler. Bu tarz farklılıklar olacaktır. Burada önemli olan daha yetenekli olan üyemizin yardımcı olması beklenmektedir ve daha deneyimsiz olan üyelere saygılı ve teşvik edici yaklaşmalıdır.

### **3.9.5 Açıklık**

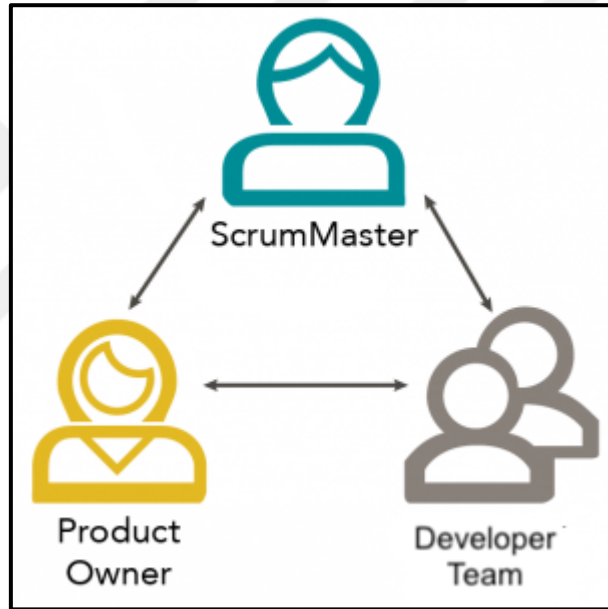
Açıklık projenin gelişiminde yani zamanında teslim edilmesi ve doğru bir şekilde tamamlanması için önemli bir etkiye sahiptir. Ürün konusunda her şey şeffaf olmalı, ürünün gideceği yol belirlenirken herkesin bir yorumu olmalıdır. Yazılım geliştirme diğer işlere göre takip edilmesi, hangi noktada nasıl zorlukların oluşabileceğinin tahmin edilmesi çok daha zor bir süreçtir. Bu nedenle scrum takımının geliştirme ile ilgili duyduğu tüm kaygıları, öngörülerini açık şekilde sprintin ilgili kişisine toplantılarda paylaşması gerekir. Oluşan bir geliştirme sorununun açık bir şekilde ortaya konulması, bunu nasıl çözeriz şeklinde yaklaşımlarla takım çalışması desteklenmelidir.

Aynı şekilde diğer taraftan Product Owner kendi taraflarındaki ürün ile ilgili gelişmeleri scrum takımı ile paylaşmalı, tüm takımın ürünün vizyonu hakkında haberdar olmasını sağlamalıdır.

### 3.10 SCRUM TAKIMI

Scrum takımlar kendi kendini yönetir ve çapraz fonksiyonlu olmalı. Kendi kendine yönetebilen takımlar kendileri karar verme özelliğine sahip olmalıdır. Çapraz fonksiyonlu takımlar takım dışında kalan başka üyeye bağımlı olmadan işi tamamlanması beklenmektedir. Scrum Çerçeveye ait bu takım modeli esneklik, yaratıcılık ve üretkenliği daha yetkin kullanabilmek için yaratılmıştır.

Şekil 3.5: Scrum Takımı



Kaynak: [www.solutionsiq.com](http://www.solutionsiq.com)

Scrum takımı Scrum Master, Ürün Sahibi (Product Owner) ve Developer Team den oluşmaktadır.

#### 3.10.1 Ürün Sahibi (Product Owner)

Ürün sahibi, ürünün başarıya götüreceği temellerini atar. Bu kişi iş akışını belirleyecek olan kişidir ve görevlerin önemliliğine göre derecelendirecek olan kişidir. Ürün sahibinin iş akışını belirlerken bazı noktalara dikkat etmelidir. Bunlar;

- I. Akışlar açık olmalıdır.
- II. Hedefler ve görevler en iyi şekilde gerçekleşmesi sağlanacak şekilde sıralamalıdır.
- III. Development takımının yaptığı işi değerliliğini arttırmalı
- IV. Development takımının alacağı olan işin anlaşılır olması şeffaf olmalıdır.

Ürün sahibi her zaman tek kişidir bir gruptan oluşmaz. Dolayısıyla ürün sahibi başka bir kuruluştan gelen önerileri iş akışına yansıtabilir fakat bir kurul iş akışın önem derecesini ürün sahibinin izni olmadan sıralamayı değiştiremez.

Ürün sahibinin başarı elde edebilmesi için başkaları tarafından saygılı olmak zorundadır. Sorumlu olacak kişi her zaman ürün sahibidir.

### **3.10.2 Geliştirme Takımı (Development Team)**

Scrum geliştirme (Development) bir metodoloji olup her bir

Geliştirme takımı, her bir aşamanın başarılı bir şekilde ulaştıran bir grup profesyonel üyelerden oluşmaktadır.

Bu takıma ait olan özellikler aşağıdaki gibidir.

- I. Kendilerini yönetebilen bir grup üyelerden oluşurlar. Geliştirme takımına bir başka departmanın üyeleri işin nasıl yapılacağını söyleyemez.
- II. Geliştirme takımı çapraz fonksiyonlu olmaları gerekmektedir yani tüm becerileri sahip olmaları gerekmektedir.
- III. Geliştirme takımı iş birliklerini farkındalıkla, iyi bir aktarımla atmosferi oluşturmaları gerekmektedir.
- IV. Geliştirme takımı zamanı iyi kullanmaları gerekmektedir. Zamanı boşa harcamamak adına günlük kısa toplantılar ve önemli projeler vakitlerini kullanırlar. Böylelikle geliştirme verimliliğini artırır ve hızlı geliştirmeyi oluşturur.
- V. Geliştirme takımının büyüklüğü işi yavaşlatacak kadar az olmamalıdır, fakat kendilerini yönetemeyecek kadar çok olmamalıdır. Genellikle 5 ile 7 arasında üyelerden oluşmaktadır.

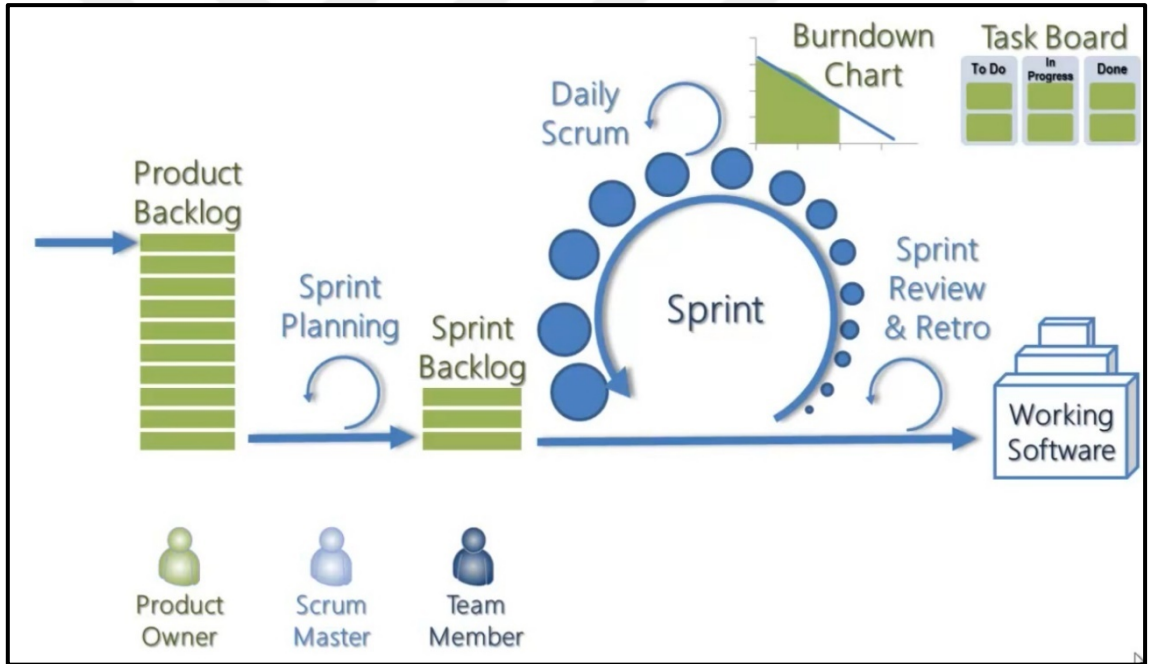
### 3.10.3 Scrum Master

Scrum Master çevik proje ynetime ait metotların Scrum takımında srdrlebilirliđinden emin olur. Scrum takımların çevik metotların prensiplerini ve pratik zmlerin kullanıldıđından emin olmak iin denetleme grevini de stlenir. Herhangi bir takım bazı sorunlarla karřılařır ve sorunları giderme grevini yine Scrum Master tarafından gerekleřtirilir.

Scrum Master, Scrum takımı rehberlik eder yani her bir sprint boyunca iřleri ayarlar, toplantıları kolaylařtırır ve takımlar arasında iyi bir iletiřim olduđundan emin olur.

## 3.11 SCRUM TAKIMIN ALIřMA AKIřI

řekil 3.6: Scrum Yařam Dngs



Kaynak: [neurontesting.blogspot](http://neurontesting.blogspot)

Scrum metotların kullanılabilmesi iin bitirilmesi beklenen proje tanımlanmaya alıřılır. rn sahibi iř srelerini belirler veya rn iř listesini (Product Backlog) planlar. rn iř listesi takımlar tarafından bitirilmesi istenen listedir. Scrum da Sprint denilen bir terim vardır. Sprint bir sonraki Scrum sprintin ulařma iin bitirilmesi beklenen bir iřtir. Her bir sprint bir Sprint Planlama ile bařlar ve bir sonraki sprintlerde neler yapılacađı belirlemek iin yapılır. Her bir takım toplanır, neler tamamlandıklarını ve neler yapacaklarını

görüşüklerini günlük Scrum toplantılar düzenlenir. Bu toplantılar Scrum Master ‘ın rehberliđi dođrultusunda yapılır. Sprintin somunda bir Demo hazırlıđı ile 4 saatlik bir deđerlendirme toplantısı yapılır ve bu toplantın sonucuna göre proje yayına alınır.

Scrum’daki bilgi yönetimi bir grup takımların kendileri arasında iletişim kurmak ve müşterilerle bilgi akışını gerçekleştirip projenin gelişimini sağlamaktır. Scrum da yer alan etkinlikler:

### **3.11.1 Sprint**

Sprint, her bir yinelemeli sürece sprint ifade edilir ve her bir sprintin 2 ile 4 hafta sınırları olacak şekilde zaman kısıtlamaları verilir. Bir önceki sprint biter birmez bir sonraki sprinte geçilir.

Sprintler durumun “Bitti” ye çekene kadar ürünün hedefi dođrultusunda geliştirme gerçekleştirilir. Sprint gelişim boyunca herhangi bir deđişiklik sokulamaz. Proje ait bilgiler arttıkça tekrar bir sprint düzenlemesi gerçekleştirilebilir.

Kesinlikle 1 aydan daha fazla süre verilmez verildiđi takdirde başka bir ürüne dönüş ihtimali vardır ve giderek ana Projenin riski artar.

### **3.11.2 Sprint Planlama**

Sprint Planlama, Sprint gerçekleşecek olan işler belirlenir ve bütün takımların birlikte çalışmalarıyla planlar belirlenir. En fazla 1 aylık Sprintler için planlama süresi 8 saat olmalıdır daha kısa zaman verilmiş sprintlerin planlaması haliyle daha kısa olmaktadır.

Scrum takımların sprint planlamadaki görevleri deđişkenlik göstermektedir. Geliştirme takımı, Sprintin geliştirmesi boyunca özelliklerin, işlevselliđini tahminlerini yapar. Product Owner’ lar, sprintin hedeflerini ve sprintin amacına yönelik ürün içeriklerini (Product Backlog) belirler. Sadece geliştirme takımı sprintlerde nelerin tamamlanacağını belirler.

Geliştirme takımı, sprint hedefler dođrultusunda işin “Bitti” durumuna geçmesi için bu toplantıda öngörülerini sunmaktadır. Önce gerekli alt yapıyı kurar ve gerekli eforu verir ve toplantı sonunda oluşturulan Sprint iş listesi (Sprint Backlog) takımların kendi kendine organize ile hangi sprintin ne kadar sürede biteceđini belirler ve işe başlar.

### 3.11.3 Günlük Scrum

Geliştirme Takımı kendileri arasında yaptıkları çalışmalarındaki faaliyetlerin bilgi akışlarını gerçekleştirdiği ve gün içerisinde yapacaklarını anlattığı bir aktivitedir. Bu aktivite 15 dakika ile sınırlı olacak şekilde gerçekleştirilmesi gerekmektedir.

Belli başlı kuralları vardır:

- I. Bütün geliştirme takımı toplantıya bütün güncellemelerle toplantıya katılır.
- II. Toplantı her gün aynı gün ve aynı saate olmalı takımda biri eksik olsa bile
- III. Toplantı 15 dakika ile sınırlı olmalı.

Günlük Scrum gerçekleştirildiğinde takımdaki üyeler bu sorulara cevap verilecek şekilde hazırlanmış olmalı;

- I. Düünden itibari ile neler tamamlandı?
- II. Bugün yapılacak işler nelerdir?
- III. Herhangi bir problem veya sorun mevcut mu?

Günlük Scrum Toplantının yararı

- I. Takımı birlik halinde tutar
- II. Bilgi paylaşımı
- III. Takımı odak halinde tutar
- IV. Takımı oluşturur.

### 3.11.4 Sprint Değerlendirme (Sprint Review)

Bu toplantıya ait prensipler;

- I. Son kullanıcı, Ürün sahibi ve Scrum Master olmak üzere katılım sağlanır.
- II. Bu toplantının amacı sprintin tamamlanıp tamamlanmadığına karar vermek
- III. Geri bildirim almak için düzenlenir.
- IV. 4 saat ile sınırlı bir toplantıdır.
- V. Tamamlanmayan herhangi bir madde sprint backlog(Sprint iş listesi) tekrar eklenir.
- VI. Eklenen yeni madde önceliği daha yüksektir bir sonraki sprinte göre
- VII. Eğer müşteriler yani son kullanıcılar tarafından bunu test edebilmek somut bir ürüne ihtiyaç duyar iseler Alpha ve Beta olmak iki farklı test ortamları oluşturulur.

### 3.11.5 Sprint Retrospektif (Sprint Retrospective)

Scrum takımı bu aşamada öz eleştiriler yapıldığı ve bu öz eleştirilerle bir sonraki sprint için bazı iyileştirmeler yapılır. Bu toplantı en fazla 3 saat ile sınırlıdır.

Sprint Retrospektifin amacı:

- I. Bir önceki sprintin nasıl geçtiğini, takım içerisindeki iletişim ve ilişkiler öngörülme çalışılır
- II. Sprint le olumlu olan noktaları belirlemek ve onları sıralamak
- III. Scrum takımının çalışma tarzının da bazı iyileştirmeler yapmaya çalışmak.
- IV. Scrum Uzmanının (Master) bu aşamada yapması gerekenler:
  - V. Scrum takımını cesaretlendirmek
  - VI. Sprint süreçleri daha keyifli olması için bazı iyileştirmeler yapmaya çalışır.
  - VII. Scrum takımını sprint bitiş durumuna adaptasyon sağlayarak ürünlerdeki kaliteyi arttırmaya çalışır.

## 4 SCRUM ÇERÇEVE AİT GELİŞTİRME ORTAMI

Bu bölümde Scrum Çerçeve' a ait olan yöntemlerin Mobil Platformda geliştirilmiştir. Mobil Platformda hazırlanması amacı günümüzde hızla gelişen teknolojiler nedeniyle ve hızlı erişebilen bir araç olacağından proje bu ortamda hazırlanmıştır.

### 4.1 MOBİL İŞLETİM SİSTEMLERİ

Akıllı bir cihazda birçok işletim sistemi bulunmaktadır. Bu işletim sistemlerin en önemlisi şüphesiz ki Apple IOS ve Google Android. Android yüzde 80 oranında Android uygulama store olan Google Play Store de paylaşılmışken, Apple yüzde 14 oranında Apple ait App Store da uygulamalar paylaşılmıştır. Apple IOS da uygulama geliştirebilmek için Swift veya Objective kullanılırken, Android ortamında uygulama geliştirebilmek için Java veya Kotlin kullanılır.

### 4.2 ANDROID İŞLETİM SİSTEMİ

Bölüm 2 de anlatmaya çalıştığımız Scrum Çerçevenin çalışma prensibi Android uygulaması hazırlanmıştır. Android platform açık kaynaklı olması, sistemsel olarak ulaşılabilirliği daha rahat bir Platform olması nedeniyle önceliği Android platforma verilmiştir.

Android Google tarafından hazırlanmış açık kaynaklı bir platformdur. Android işletim sistemin de uygulama geliştirebilmek için Android Studio ihtiyaç vardır. Uygulama da java dili kullanıldığı için Google kendi java sanal makine (JVM) geliştirmiştir ve adına Dalvik Virtual Machine (VM) vermiştir. Bütün java dosyaları Dalvik VM göre düzenlenmiş olması gerek. Java kodları Android işletim sistemine derlenmeden önce Dalvik VM görevi byte çevirmektir.

### 4.3 ANDROID UYGULAMA GELİŞTİRMEK

Android yazılım geliştirme süreci Android işletim sistemi için uygulamayı oluşturma sürecidir. Uygulamalar çoğunluk java diliyle geliştirildiği için java dilini kullanan Android yazılım geliştirme kit (SDK) denilen bir geliştirme ortamı vardır ve başka geliştirme ortamlarda mevcuttur.

### 4.3.1 Android Yazılım Geliştirme Kit (SDK)

Android yazılım geliştirme seti (SDK) kapsamlı bir geliştirme araçları seti içerir. Bunlar arasında bir hata ayıklayıcı, kitaplıklar, QEMU'ya dayalı bir el cihazı öykünücüsü, belgeler, örnek kodlar ve eğiticiler bulunur. Şu anda desteklenen geliştirme platformları Linux (herhangi bir modern masaüstü Linux dağıtımı), Mac OS X 10.5.8 veya üstü ve Windows XP veya daha sonraki sürümlerini çalıştıran bilgisayarları içermektedir. Ayrıca, Android yazılımı özel Android uygulamaları kullanılarak Android'de geliştirilebilir. Android uygulamaları .apk formatında paketlenmiştir ve Android işletim sisteminde data/app klasörü altında saklanır (klasör güvenlik nedenleriyle yalnızca root kullanıcısına erişilebilir). APK paketi dex dosyalarını (Dalvik yürütücüsü olarak adlandırılan bayt kod dosyaları derlenmiştir), kaynak dosyalarını vb. İçerir.

### 4.3.2 Android Debug Bridge

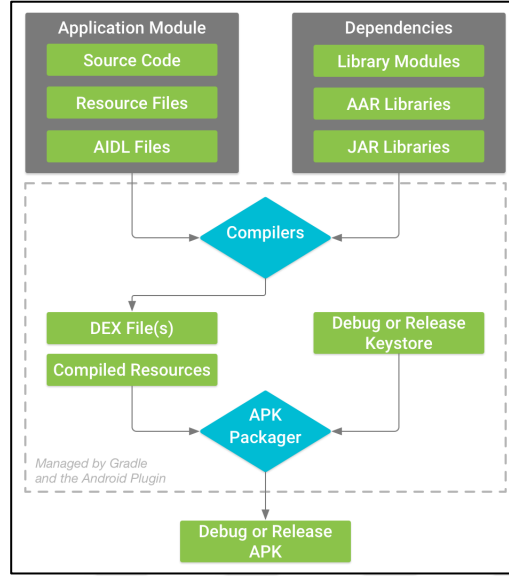
Android Hata Ayıklama Köprüsü (ADB) Android SDK paketinde bulunan bir araçtır. Birbiriyle iletişim kuran istemci ve sunucu tarafı programlarından oluşur. ADB'yi kontrol etmek için çok sayıda grafiksel kullanıcı arabirimi olmasına rağmen ADB'ye komut satırı arabirimi üzerinden erişilir.

### 4.3.2 Android Derleme Yapısı

Android derleme sistemi uygulama kaynaklarını ve kaynak kodunu derler ve bunları test edebileceğiniz, dağıtabileceğiniz, imzalayabileceğiniz ve dağıtabileceğiniz APK oluşturur. Android Studio gradle kullanır. Gradle, özel bir derleme aracıdır ve geliştirici için özel esnek yapılandırmaları tanımlayarak, yönetebilecek ve otomasyon hazırlanabilecek özel bir yapıdır. Bir uygulamaya ait olan tüm versiyonların ortak olan yapının belirli kısımları yeniden kullanarak her derlemeye ait kendi kod ve kaynaklarını oluşturabilir.

Android uygulama paketini (APK) oluşturabilmek için derleme süreci birçok bir araç ve süreçler içerir. Derleme süreci çok esnek olduğundan süreçlerde geçen olaylar Şekil 4.1 de gösterilmiştir.

**Şekil 4.1: Android Derleme Süreci**



Kaynak: [developer.android.com](http://developer.android.com)

Şekil 4.1 de gösterildiği gibi bir Android uygulamanın oluşması geçen adımları gösterilmiştir. Bu adımları inceliyor olursak:

- I. Derleyiciler, kaynak kodunu ve derlenmiş kaynak kodları DEX (Dalvik Executable) dosyalarına dönüştürür. Bu dosyalar Android cihazlarda çalışan bayt kodları ve derleme dosyaları içerir.
- II. APK Paketi, DEX dosyalarını ve derlenmiş kaynakları tek bir Android uygulama paketinde birleştirir. Uygulamanız bir Android cihazına yüklenmeden ve yayına alınmadan önce, Android uygulama paketinin imzalanması gerekir.
- III. Android uygulama paketi 2 anahtar kullanarak imzalar:
  - a. Eğer geliştirici bir onarma(debug) sürümü oluşturuyorsa test etmek ve profil oluşturmak bu sürüm oluşturulur. Paketleyici uygulamanızı hata ayıklama anahtar imzalar. Android Studio, yeni projeleri bir hata ayıklama anahtarını otomatik olarak yapılandırır.
  - b. Eğer bir yayınlama süreci oluşturuluyorsa geliştirici uygulamayı yayına almayı planlamaktadır. Bunun için paketleyici manuel olarak bütün bilgileri girerek bir anahtar şifresi oluşturur. Anahtar şifre oluşturabilmek için öncelikle bir Android Studio ya ihtiyacınız olacaktır. Daha sonra geliştirilen uygulamaya bir yayınlama

sürümü oluşturulur ve daha sonra Android Studio da bulunan imzala tabına tıklayarak gerekli bilgileri girip imzalanmış bir Android uygulama paketi oluşturulur.

- IV. Yayınlama sürümü oluşturulma esnasında belleğinizi daha az kullanabilmek zipalign adında bir araç kullanılarak optimize edilir.

Bu süreçler sonucunda hata ayıklama sürümü ve yayınlama sürümü oluşturulmuş olup test edilebilir ve yayına alınabilir iki adet Android uygulama paketi elde edilir.

#### **4.4 ANDROID'DE ÖZEL DERLEME YAPILANDIRMASI**

Gradle ve Android eklentileri yardımı ile derleme yapının özelliklerini düzenlemesinde yardımcı olur. Bunlar;

- I. Derleme Türleri (Build Types), gradle uygulamasının uygulamanızı oluştururken ve paketlerken kullandığı belirli özellikleri tanımlar ve genellikle geliştirme yaşam döngüsünün farklı aşamaları için yapılandırılır. Örneğin, hata ayıklama sürümü türü test edebilme seçeneklerini etkinleştirir ve Android uygulama paketinin hata ayıklama anahtarıyla işaretlerken, yayına alabilme sürümü ise uygulamanın büyüklüğünü küçültme, gizlenebilir ve yayınlama için bir yayınlama anahtarıyla Android uygulama paketini imzalar. Uygulamayı oluşturmak için en az bir derleme türü tanımlamanız gerekir; Android Studio, varsayılan olarak hata ayıklama ve yayın oluşturma türlerini oluşturur.
- II. Ürün niteliği (Product Flavor), uygulamanın farklı versiyonları temsil eder. Yani Uygulamanın belirli kısmını ücretlendirmek veya bir kısmını ücretsiz olması istenirse gradle 'ın ürün niteliği özelliğini tanımlayarak çeşitli versiyonlar üretilebilir.
- III. Derleme varyasyonları, ürün niteliğinin ve derleme tipinin birleşiminden oluşan çapraz bir üründür. Gradle bu ürünü uygulamayı derlemek için kullanır. Derleme türünü kullanarak geliştirme boyunca test sürümü veya yayına alabilmek için yayın sürümü derlenebilir. Direkt olarak derleme varyasyonlarını yapılandırılmazsa da derleme varyasyonlarını oluşturan derleme türü ve ürün niteliğini yapılandırarak derlenebilir. Ek ürün niteliği ve derleme türü oluşturularak ek derleme varyasyonlar oluşturulmuş olunur.
- IV. Manifesto girdileri, manifesto dosyasının bazı özellikleri için derleme varyasyonlarında değerler belirtilebilir. Bu değerler var olan versiyonu manifesto dosyasının üzerine yazar. Böyle bir özelliği kullanmanın faydası birden fazla

android uygulama paketi oluşturmak ve oluşturulan bu android uygulama paketleri farklı paket adları, farklı sürümler oluşturmada büyük bir katkıda bulunacaktır.

- V. Bağımlılıklar (Dependencies), derleme sistemi projenin bağımlılıklarını yerel dosya sisteminden ve uzak depo yönetir.
- VI. İmzalama (Signing), derleme sistemi derleme yapısındaki belirlenmiş imzalama özelliğini etkinleştirir ve android uygulama paketini otomatik derleme süreci esnasında imzalar. Derleme sistemi test sürümü oluştururken varsayılan bir anahtar ve sertifika kullanılarak şifre sorgusunu önlemek için test android uygulama paketi oluşturur fakat derlemse sisteme yayınlanması niyetinde oluşturulan android uygulama paketini imzalamaz. Burada geliştiricinin bir anahtar oluşturması gerekmektedir.
- VII. ProGuard, her bir derleme varyasyonu için belirlenmiş proguard kuralları ile derleme sistemi tarafından etkinleştirilir. Derleme sistemi proguard vasıtasıyla oluşturulan kurallarla, geliştirme esnasında oluşturulmuş sınıfları küçültür ve gizler.
- VIII. Çoklu android uygulama paketleri, derleme sistemi çoklu android uygulama paketi oluşturmayı etkinleştirir ve her bir uygulama paketi kodlar ve gerekli kaynakları içerir.

#### **4.5 ANDROID SAYFALARIN YAŞAM DÖNGÜSÜ**

Kullanıcı ve uygulama arasında gerçekleşen etkileşimler için sınıflara tanımlanan 2 sınıf türleri mevcuttur. Bunlar etkinlik(activity) ve fragment olarak adlandırılmaktadır.

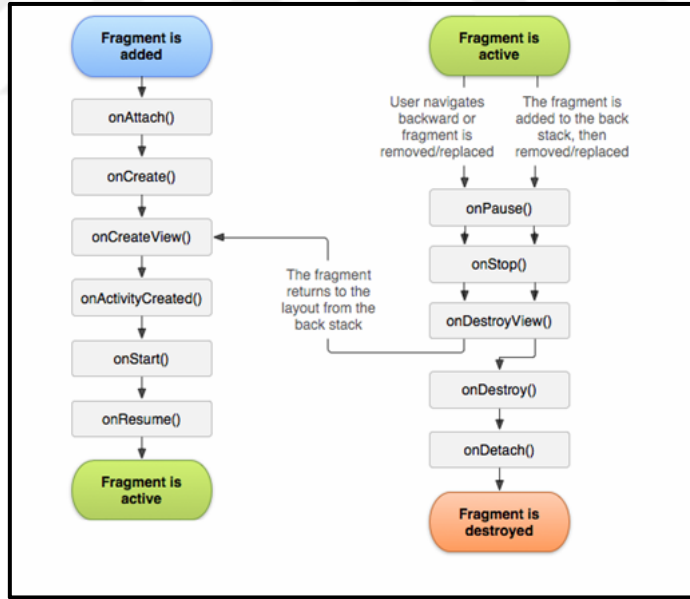
Mobil uygulamalarda diğer platformlarda olduğu deterministik olmayan bir yönü vardır. Yani kullanıcı mail olan bir uygulamayı açtığında maillerin listesini görmektedir fakat aynı zaman başka bir uygulama vasıtasıyla da mail uygulamasını açabilir. Bu durumu kolaylaştırmak için etkinlik sınıfı oluşturuldu. Bir uygulamadan başka bir uygulama çağırıldığında etkinlik sınıfı başlatılır. Böylece etkinlik sınıfı kullanıcının etkileşime geçen bir android bileşenidir. Etkinlik, uygulamanın tasarımını çizdiği pencereyi sağlar

Fragment (Fragman), uygulamada kullanıcı arabirimin davranışı ve etkinlik sınıfın bir kısmını temsil eder. Birçok fragman tek bir etkinlik sınıfında kullanılabilir ve bir etkinlik sınıflarında tekrar tekrar kullanılabilir. Bir fragman, kendi yaşam döngüsüne sahip olan,

kendi giriş olaylarını alabilen ve etkinlik sınıfı çalışırken ekleyebileceğiniz veya kaldırayabileceğiniz bir etkinlik sınıfının modüler bölümü olarak düşünebilir. Bir fragmanın çalışabilmesi için her zaman bir etkinlik sınıfına ihtiyaç duyar ve fragmanın yaşam döngüsü etkinlik sınıfının faaliyetine ait yaşam döngüsünden doğrudan etkilenir. Örneğin etkinlik sınıfı yaşam döngüsü duraksama yaşandığında fragman da aynı zamanda duraksar. Bununla birlikte, bir etkinlik devam ederken (yaşam döngüsünde devamlılık durumu), her fragman, ekleme veya kaldırma gibi bağımsız olarak değiştirilebilir. Böyle bir fragman işlemi gerçekleştirildiğinde, etkinlik sınıfı tarafından yönetilen bir arka yığına da ekleyebilirsiniz – etkinlik sınıfındaki her bir arka yığın girdisi, gerçekleşen parça işleminin bir kayıdır. Arka yığın, kullanıcının Geri düğmesine basarak bir parça işlemini tersine çevirmesine (geriye doğru gitmesine) izin verir.

Her ne kadar fragman etkinlik sınıfının bir kısmını temsil etse her sayfa türünün kendisine ait yaşam döngüleri mevcuttur.

**Şekil 4.2: Fragman Yaşam Döngüsü**

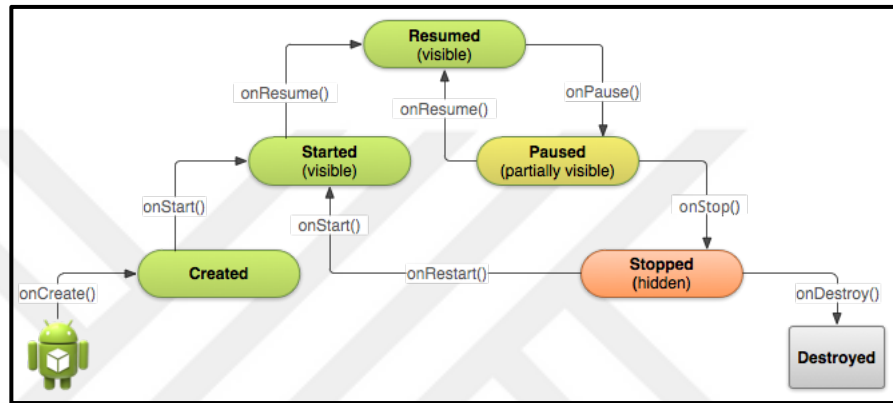


*Kaynak: medium*

Şekil 4.2 görüldüğü üzere bir fragmana ait yaşam adı gösterilmektedir. Fragman sayfası oluşturulduğunda en azından aşağıda listelenen metotları çağırılması gereklidir.

- onCreate metodu, sistem bu metodu fragman oluşturulmaya başlandığı an itibaren tetiklenir. Bu metot içerisinde temel bileşenleri oluşturulmalı ki uygulama durdurulup tekrar sayfa yenilendiğinde tekrar kullanılabilir olsun.
- onCreateView metodu, sistem bu metodu uygulamanın tasarımını çizmeye başlandığında tetiklenir.
- onPause metodu, sistem bu metodu kullanıcı sayfadan ayrıldığı an itibariyle tetiklenir.

**Şekil 4.3: Etkinlik Sınıfın Yaşam Döngüsü**



*Kaynak: android developer page*

Şekil 4.3 de gösterildiği üzere bir etkinlik sınıfının yaşam döngüsü gösterilmiştir. Etkinlik sınıfına ait olan bu yaşam döngüsünün geri çağırma yöntemleri içinde, kullanıcı etkinlik sınıfından ayrılması ve yeniden girdiğinde etkinliğinizin nasıl davranacağını tanımlanabilir. Örneğin bir video uygulaması düşünüldüğünde video oynatma esnasında kullanıcı sayfadan ayrılma esnasında video ve ağ bağlantısını durdurup sayfadan ayrılmasına izin verilir fakat tekrar uygulamaya girdiğinde kullanıcı kaldığı yerden devam etmeli. Diğer bir deyişle, her bir geri çağırma yöntemi belirli işleri yapmasını sağlar. Doğru işi doğru zamanda yapmak ve geçişleri düzgün bir şekilde gerçekleştirmek, uygulamanızı daha sağlam ve performanslı hale getirir.

Kullanıcı aktiviteden ayrılmaya başladıkça, sistem etkinlik sınıfını ortadan kaldırmak için yöntemleri çağırır. Bazı durumlarda, bu ortadan kaldırma sadece kısmidir; aktivite hala bellekte (kullanıcının başka bir uygulamaya geçtiğinde olduğu gibi) bulunur ve yine de ön plana geri dönebilir. Kullanıcı bu etkinliğe geri dönerse, etkinlik kullanıcının kaldığı yerden devam eder. Sistemin belirli bir süreci (söz konusu faaliyetlerle birlikte) öldürme

olasılığı, o zamanki etkinliğin durumuna bağlıdır. Aktivite durumu ve bellekten kaldırma, durum ile ortadan kaldırmaya arasındaki ilişki hakkında daha fazla bilgi sağlar.

Etkinliğinizin karmaşıklığına bağlı olarak, muhtemelen tüm yaşam döngüsü yöntemlerini uygulamanız gerekmez. Ancak, her birini anlamanız ve uygulamanızın kullanıcıların beklediği gibi davranmasını sağlayanları uygulamanız önemlidir. Uygulamanın karmaşıklığına bağlı olarak yaşam döngüsünü metotları kullanılır.

#### **4.6 FIREBASE**

Firestore, web ve mobil uygulamalarını sunucu tarafı programlaması olmadan geliştirmemizi sağlayan bir teknolojidir ve böylelikle geliştirme süreci daha kolay ve hızlı olur. Bir uygulama geliştirmek için her zaman servisler kurmak zorunda değiliz. Firestore'e gerekli servisleri sağlayabilir: kullanıcıları doğrulama, veri saklama ve erişim kurallarını uygulama gibi servisler hazırda kullanılabilir olması uygulama geliştiriciler için zaman kazanımı olacaktır.

Ayrıca iOS, web, OS X ve Android istemcileri destekler. Firestore ile geliştirilen uygulamalar, verileri nasıl depolayacağını düşünülmesine gerek kalmadan verileri kontrol edebilir ve kullanabilir.

Firestore kullanım avantajları olduğu tabii birtakım dezavantajlarda mevcuttur. Öncelikle avantajlardan bahsedecek olursak;

- I. Firestore verileri bulut saklayan bir veri tabanıdır ve sürekli olarak dataları senkronize eder. Dolayısıyla yeni bir servise ihtiyaç yoktur.
- II. Cihazın donanımı düşünmeden (hafıza gibi) gerek kalmadan verileri iyi bir şekilde saklanabileceği bir depoya sahiptir.
- III. Birçok veriye kolaylıkla gerçek zamanda işleyebilir
- IV. Kullanıcı doğrulamayı destekler
- V. Kurulumu çok kolaydır.
- VI. Firestore, tüm bilgi alışverişi için 2048 parça SSL şifrelemesi gerektirdiğinden veriler korunur.

Dezavantajları bakıldığında ise aşağıda sıralanan durumlar bir dezavantaj olarak gözüktüğü de geliştirme sürecinde zaman kazanmanızda geliştirici büyük bir artı olacaktır.

- I. Firebase ücretli olması en büyük dezavantajlarından ve eğer kullanım oranı artıkça maliyeti artabilir.
- II. Günümüzde kullanılan veri tabanlarıyla karşılaştığımızda Firebase verileri JSON olarak saklamaktadır ve bu özelliğinin olması verileri başka ortamlara taşımak bir hayli zor olabilir.

#### **4.7 FIREBASE GERÇEK ZAMANLI VERİTABANI**

Real-time veri tabanı, dataların bulut (cloud) olarak depolandığı ve anlık süreçlerin gerçekleştiği bir veri tabanıdır. Bağlanan her bir alıcı ile senkronize halindedir. Bu uygulamamda Google ait olan Firebase aracı kullanılmıştır.

Firestore, Google tarafından mobil ve web platformlar da kullanılmak için geliştirilmiştir. Firestore'in birtakım özellikleri vardır ve bu özelliklerin biri olan gerçek zamanlı veri tabanı projede kullanılmıştır.

Projede gerçek zamanlı veri tabanı kullanılma sebebi her bir cihazla anlık işlem yapabiliyor olması. Örneğin tamamlanması beklenen bir Sprinti geliştirici tarafından "bitti" durumuna çektikten sonra anlık olarak ürün sahibi veya Scrum Master haberdar edilebiliyor olacaktır.

Firestore Realtime Veritabanı, doğrudan istemci tarafı koddan veri tabanına güvenli erişim sağlayarak zengin, işbirlikçi uygulamalar oluşturmaya olanak tanır. Veriler yerel olarak kalıcıdır ve çevrimdışıyken bile, gerçek zamanlı olaylar son kullanıcıya duyarlı bir deneyim yaşatmaya devam eder. Cihaz yeniden bağlantı kurduğunda, Gerçek Zamanlı Veri Tabanı, yerel veri değişikliklerini istemci çevrimdışıyken gerçekleşen uzaktan güncellemelerle senkronize eder ve herhangi bir çakışmayı otomatik olarak birleştirir.

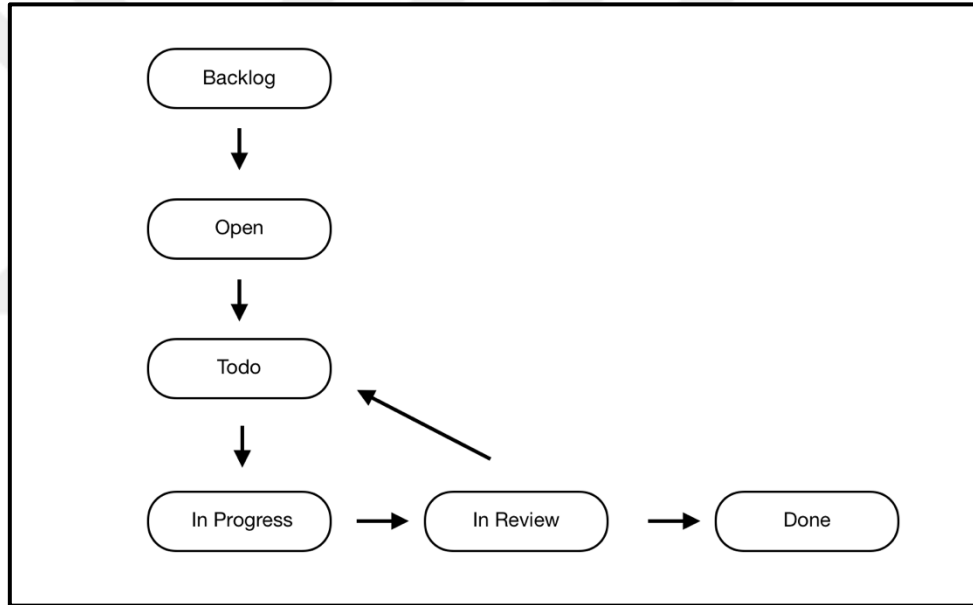
Gerçek Zamanlı Veritabanı, verilerinizin nasıl yapılandırılacağını ve verilerin nasıl okunabileceği veya yazılabileceğini tanımlamak için Firestore gerçek zamanlı veri tabanı güvenliği kuralları adlı esnek, ifadeye dayalı bir kural dili sağlar. Firestore doğrulama ile entegre edildiğinde geliştiriciler, kimin hangi verilere erişebileceğini ve bunlara nasıl erişebileceklerini tanımlayabilir.

Gerçek Zamanlı Veri Tabanı bir NoSQL veri tabanıdır ve ilişkisel bir veri tabanına kıyasla farklı optimizasyon ve işlevselliklere sahiptir. Gerçek zamanlı veri tabanı API'si, yalnızca hızlı bir şekilde çalıştırılabilen işlemlere izin verecek şekilde tasarlanmıştır. Bu, milyonlarca kullanıcıya yanıt vermeden ödün vermeden mükemmel bir gerçek zamanlı deneyim oluşturmanızı sağlar. Bu nedenle, kullanıcıların verilerinize nasıl erişmesi gerektiğini düşünmek ve daha sonra buna göre yapılandırmak önemlidir.

#### 4.8 SCRUM TAKIMIN SÜREÇLERDEKİ ORTAK YORUM TANIMLANMASI

Scrum çerçevesinin özelliklerinden bir transparan olma hedefini bu tez çalışmasında baz alındı ve yazılım aşamasında sprintlerin ve projenin durumunu belirlenen süreçler aşağıdaki gibidir

**Şekil 4.4: Scrum Süreçlerinde Kullanılan Ortak Dil**



- I. Backlog, 2.bölümde de bahsedildiği gibi ürün iş listesini belirlenir. Proje yöneticileri veya ürün sahipleri projenin amacına göre hedefleri belirler. Ürün iş listesini (backlog) belirlemeden bir sonraki adıma geçilmez. Önce tanımlar belirlendikten sonra bir sonraki adım “Open” a geçilir.
- II. Open durumu, sprint iş listesi belirlendikten sonra proje “Open” duruma geçilir. Bu durumda geliştiriciler “Sprint Planlaması” için toplantı düzenler. Bu toplantıda geliştiriciler sprinti bitimi ile alakalı veya sistemsel gereksinimlerin tahminlerini yapar.

- III. To-do durumunda geliştirici takımından biri sprinti kendi üzerine alır süreyi başlatmak için bir sonraki adım olan “In Progress” e geçer.
- IV. In Progress durumunda geliştirici takımından bir üye sprinte başladığını ifade eder.
- V. In Review geliştirici takımından bir üye sprinti tamamladıktan sonra in review durumuna dönüştürür. Bu durumda ürün sahibi projenin testini yapar ve test esnada çıkan herhangi bir durum karşısında geri bildirimini yapar ve to-do durumuna getirilir.
- VI. Done, bu durumda sprint proje amacına uygun olduğundan teyit edilmiştir. Gerekli test aşaması tamamlandıktan sonra bu parça ürün projenin entegre edilir.

## **4.9 SCRUM TAKIMININ PROJE UYGULAMASINDAKİ YETKİNLİKLERİ**

Scrum Çerçeve konusunda da anlatıldığı üzere Scrum takımın belli başları görev dağılımları vardır. Scrum Çerçeve genişletebilir ve geliştirebilir yöntemleri olduğundan bu uygulamada takımların belli başlı görevleri vardır. Bunlar;

### **4.9.1 Proje Yöneticisi**

Proje Yöneticileri, projenin girişini ve ürün iş listesini oluşturmakla görevlidir ve ürün iş listesi oluşturulmadan sprintin durumu “backlog” durumunda olmalıdır. Ancak ortaya çıkan iş listesi açık ve anlaşılır olduğundan sprint durumunu “Open” a çekmelidir ki geliştirme takımıyla Sprint Planlama toplantısı gerçekleştirebilmek için.

### **4.9.2 Development Takımı**

Development takımı Sprint planlamasından sonra sprintin durumunu “In progress” e çekmelidir ki sprinti oluşturan kişinin görevin üzerinde çalıştığını haberdarı olması için. Herhangi sistemsel eksiklik dahilinde sprinti kuran kişiye uygulama üzerinden haberdar olmalıdır ki bir sonraki toplantıda veya Scrum Master tarafından bu durum çözülebilesin.

Ancak geliştirme takımı çalıştığı Sprint üzerinden bittiği takdirde sprintin durum “In Review” e çekilmelidir.” In Review” durumuna çekilmiş sprint Proje Yöneticileri tarafından ve test etmesi gereken kişiler tarafından bir demo sunum ile taktim edilir. Proje Yöneticileri tarafından test edilen ürün tamam alındığında Sprintin durumu “done” çekilir ve bir sonraki sprinte geçilir.

### 4.9.3 Scrum Master

Scrum Master tüm süreçlerde olmak zorundadır. Scrum Çerçeve' e ait metodolojileri kullanıldığından emin olmalıdır ve bir diğer görevi sprint işlem görme esnasında takımlar arasında herhangi bir iletişim bozukluğu olup olmadığını tahmin etmek veya proje aktarımların anlaşılır olduğundan emin olmak için uygulamada her aşamasında yetkisi vardır.

## 5 . SCRUM MOBİL UYGULAMANIN KULLANIM KILAVUZU

Bu tezin çalışmasını gerçekleştirmiş olduğum mobil uygulamanın adı “Scrum Management Tool” ‘dır. Bu uygulama amacı Scrum Çerçeve süreçlerini yönetebileceğimiz bir uygulama yaratmaktır. Mobil uygulama ile süreçlerden daha hızlı haberdar olabileceğimiz ve hızlı geri bildirimlerle projenin sürelerinde hızlandırma ve verimliliğini arttıracaktır.

Proje bir Android Uygulaması olduğu sadece Android cihazlarda çalışmaktadır. Firebase ait olan gerçek zaman özelliği kullanıldığı bazı kurulumlar gerçekleştirilmelidir. Bu kurulumlar 4.1 daha detaylı anlatılmıştır.

Projedeki kullanıcı girişlerini sağlayabilmek için Firebase üzerinden kullanıcıya yetkinlik verilmesi gerekmektedir. Bu konu 4.2 de daha detaylı anlatılmıştır. Daha sonra uygulama Android Uygulama Marketi Google Play de bulunmadığından uygulamanın APK sını kurulması gerekmektedir.

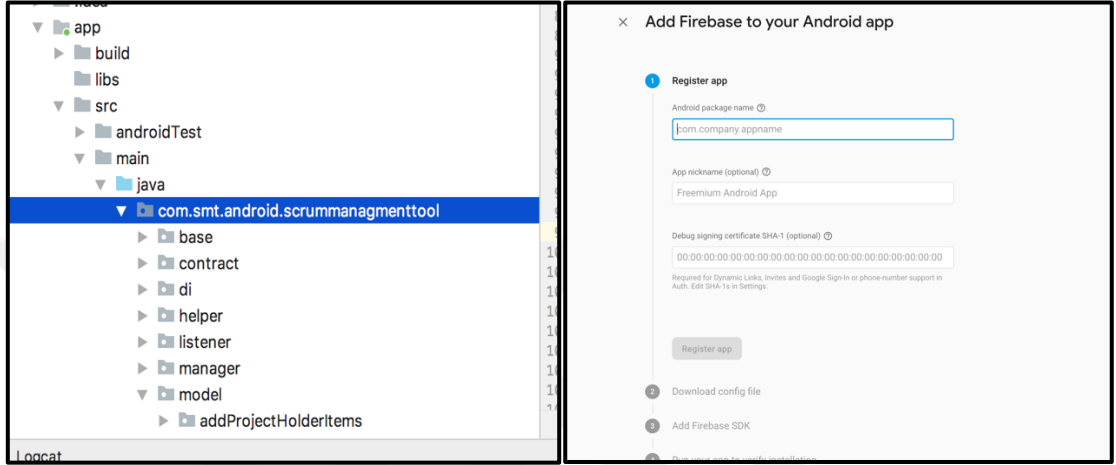
APK açılımı Android package dir yani sadece Android işletim sistemli çalışan bir uzantıdır. Sadece yapılması gereken APK (Android Package) 'yı telefona yükleyip kurulumu gerçekleştirmektir.

Bir APK yı cihaza kurabilmek cihazınızın Geliştirme Seçeneğini açmanız gerekmektedir. Bunun için ayarlar menüsünden “Sistem” bölümüne tıklanmalı ardından “telefon hakkında” bölümünden versiyon numarasına 7 kere tıklayıp cihazın geliştirme özelliğine açılmış olur.

## 5.1 FIREBASE – REALTIME KURULUMU

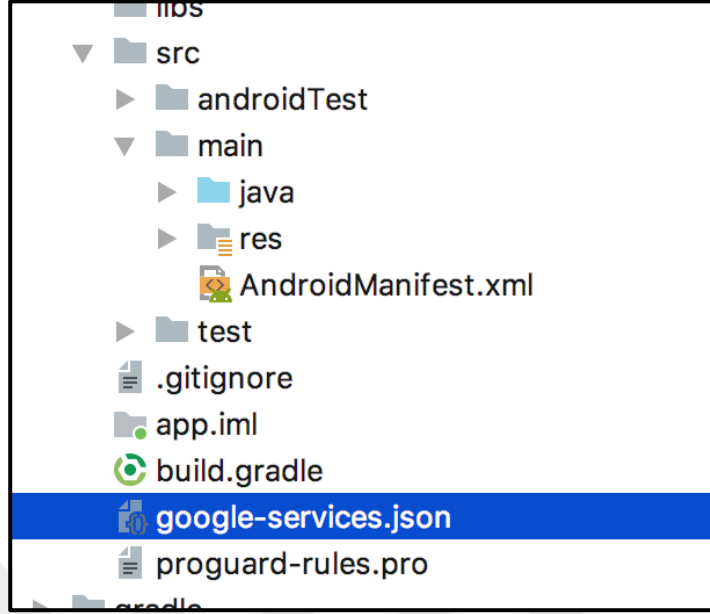
Öncelikle Firebase de kurulum yapmadan önce bir Google mail hesabı oluşturulması gerekmektedir. Ardından Android Studio üzerinden kurmuş olduğunuz projenin paket adını Firebase tanımlamanız gerekmektedir.

**Şekil 5.1: Android Studio Yer Alan Paket Adını Firebase Tanımlama**



Ardından Firebase paket adını tanımladıktan sonra bir sonraki aşama Firebase tarafından elde edilen googl-service.json dosyasını src dizinine eklenir Şekil 5.2 de olduğu gibi. Daha sonra uygulamayı derledikten sonra proje Firebase bağlanmış olur.

**Şekil 5.2: Google Dosyasını Android Studio Eklenmesi**



google-services.json, GoogleApiClient ile bağlantı kurarken doğrulanması gereken geliştirici kimlik bilgilerini ve yapılandırma ayarlarını içerir. Servisler, geliştirici hesabınızı algılayıp çalışabilecek durumda olsa da uygulamayı canlıya aldığınızda yani Google Store yayınlasanız da bu dosya olmadan metodlar çalışmayacaktır. Dolayısıyla burada dikkat edilmesi gereken Firebase konsol değiştirmeye kalktığınızda projelerde yeni Google servis dosyasını eklemek veya güncellemek gerekecektir.

## **5.2 FİREBASE ÜZERİNDEN KULLANICI GİRİŞ İÇİN YETKİNLİK VERME**

Herhangi bir kullanıcıya yetki verebilmek için 3 önemli rol tanımlanması gerekir. Bu roller sırasıyla

- I. Geliştirici (Developer)
- II. Scrum Uzman (Master)
- III. Project Yöneticisi

Rollerin dağılımı yukarıdaki gibi ifade edildiği gibi tanımlanmıştır. Firebase'in gerçek zamanlı özelliği veriler Json olarak bulut (cloud) da saklanır şekil 5.3 de görüldüğü gibi.

**Şekil 5.3: Kullanıcı Firebase Gerçek Zamanlı Veri Tabanı**



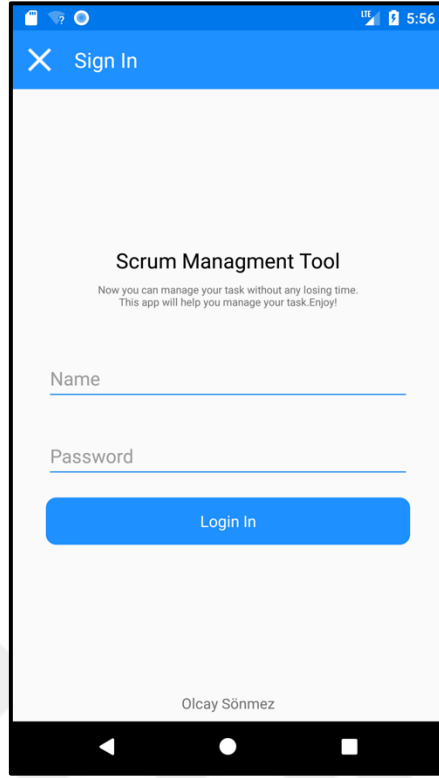
Bu durum da aynı yapıda “user” yazan bölümde “+” ya tıklayarak kullanıcı girişi sağlanır. Bu sayede kullanıcı uygulamaya giriş yaptığında. “occupation” değeri dikkate alınacaktır. Kullanıcı Project Manager ve Scrum Master olarak giriş yaptığında tüm yetkinliklere yani proje ekleme, sprint ekleme ve sprinti düzenleme hakkına sahip olacaktır fakat kullanıcı bir developer olarak giriş yaptığında Sadece sprint ekleme ve sprinti düzenleme hakkına sahip olacaktır.

### 5.3 UYGULAMA GİRİŞ SAYFASI

Kullanıcı giriş sayfası Şekil 5.4 görüldüğü gibidir. Kullanıcı, kullanıcı adı ve şifre bilgilerini girerek kullanıcı tipine bağlı olarak sprint ve ürün bilgilerini görebilmektedir. Bölüm 3.4 de anlatıldığı gibi kullanıcının tipine bağlı olarak uygulama da yetkinlikleri bulunmaktadır.

Kullanıcı bilgilerini girdikten sonra önce bir User sınıfı açılır ve oluşturulan nesne ile gerçek zamanlı veri tabanına bilgileri Firebase aracılığı ile istek yapılır ve veri tabanına yollanan veriler user json da bulunuyorsa kullanıcı girişi sağlanır ve uygulama boyunca User Manager sınıfında saklanır.

**Şekil 5.4: Uygulamamın Giriş Sayfası**



Kullanıcı uygulamayı açtığından alt bölümde oluşan “Profil” sekmesine tıklayıp “Login In” butonuna tıklaması gerekmektedir. Daha sonra Şekil 5.4 ‘deki ekran çıkacaktır ve giriş yapan kullanıcıya bağlı olarak uygulamada yetkinliklere göre sayfaları gözlemleyebilecektir.

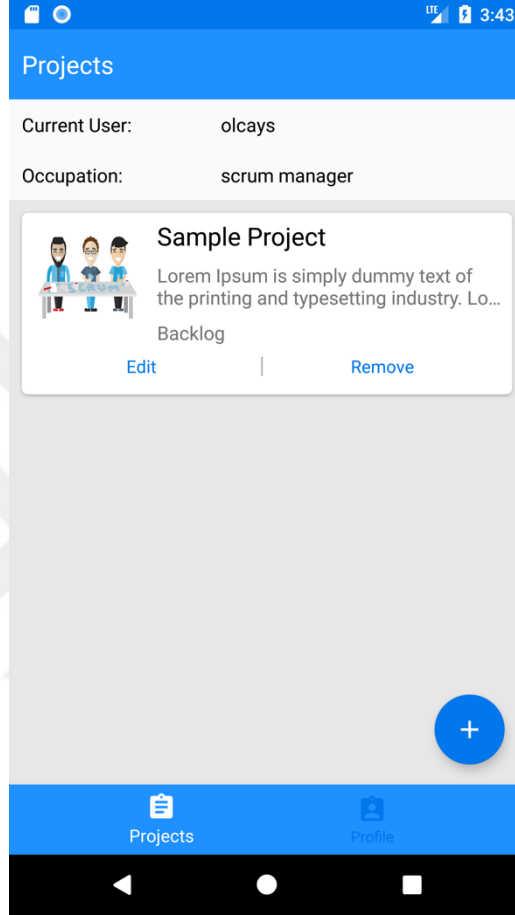
#### **5.4 UYGULAMADAKI PROJE LİSTESİ VE PROJE OLUŞTURMAK**

Uygulamamın proje liste görünümünün ekran görüntüsü şekil 5.5 verilmektedir. Giriş yapan proje yöneticileri veya Scrum Master ‘lar tarafından eklenen projelerin listeleri görüntülenmektedir. Buradaki listede aynı zamanda giriş yapan kullanıcın adı, kullanıcı tipi görüntülenmektedir.

Sayfa açılmadan önce verilerin yüklenebilmesi için firebase e bir sorgu atılmaktadır. Elde edilen sonuçlar daha sonra bu sayfada gösterilmektedir. Burada sonuçları listelemek için recycler view denilen android tasarım bileşeni kullanılmaktadır. Bu tasarım bileşenin en büyük özelliği dinamik verileri gösterebiliyor olması yani eklenen veriler listelendikten sonra, sonradan eklenen bir veriyi tekrar gösterebiliyor olmasıdır. Aşağıya veya yukarıya

kayırdığımız da ise cihazın belleğini fazla yormamasını adına listenin görülmeyen kısmını yok eder ve sadece görünen kısmı oluşturur daha sonra tekrar kayırdığımızda listeleri birer birer yeniden oluşturur.

**Şekil 5.5: Uygulama Proje Liste Sayfası**



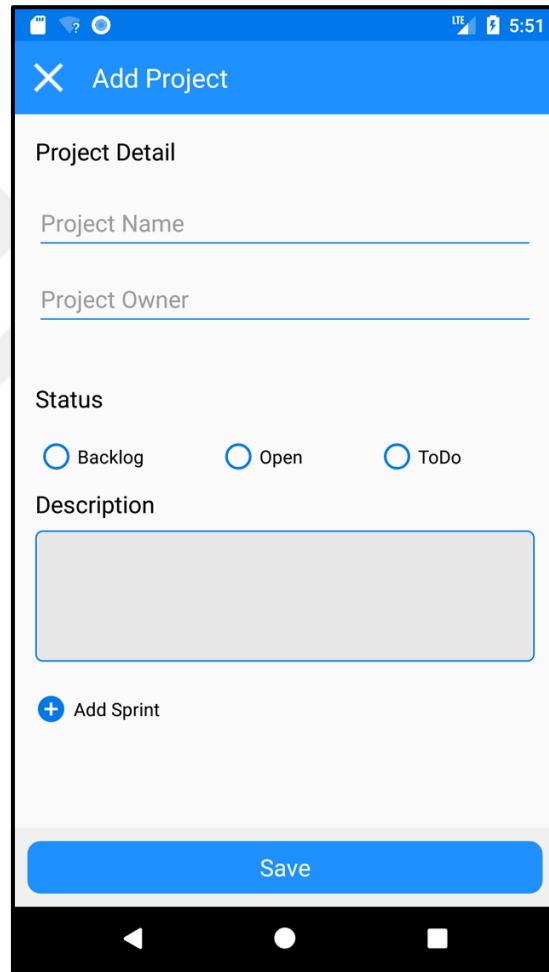
Scrum yönetici veya Proje yöneticileri tarafından girilen proje bilgileri yani sıra projelerin durumları görüntülenmektedir. Giren kullanıcı Proje projenin hangi durumda anlayabilmesi için aynı zamanda projelerin durumunu gösterilmektedir.

Giren proje yöneticisi veya Scrum Master projeleri girebilmesi sağ alt bölgede yer alan + butonuna tıklaması gerekmektedir ve tıkladıktan sonra şekil 5.6 olduğu gibi bir ekran sayfası açılmaktadır.

Burada yapılan firebase sorgusu sayfa kapanmadığı müddetçe sürekli veri tabanı dinler ve yeni bir proje ekledikten sonra firebase bir metodu ile tasarım bileşenine tetikler ve yeni bir güncelleme ile sayfayı yeni proje listesiyle günceller.

Bu sayfada kullanıcı dilerse projeye silebilir veya var olan projeyi güncelleyebilir. Her bir projenin oluşturulan görüntünün hemen altında iki adet butonu mevcuttur. Kullanıcı remove tıklarsa projeyi firebase aracılığı ile silebilir fakat kullanıcı edit butonuna tıklar ise proje detay sayfası şekil 5.6 da olduğu gibi bir sayfa açılmaktadır.

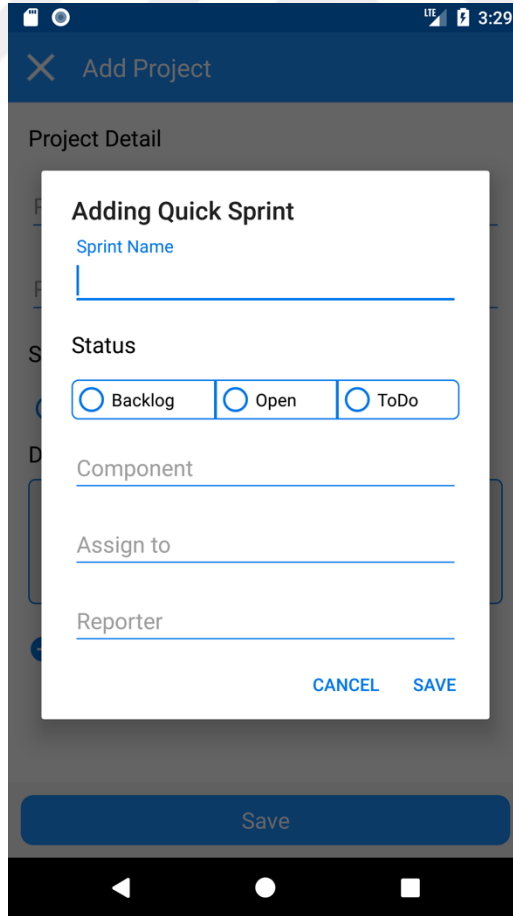
**Şekil 5.6: Proje Ekleme Sayfası**



Kullanıcı burada proje bilgilerini girerek proje'yi ve projeye ait bilgileri oluşturabilmektedir. Girebilecekleri bilgileri;

- I. **Project Name:** Projenin adı girilmesi gerekmektedir
- II. **Project Owner:** Proje Sahibinin adı girilmesi gerekmektedir
- III. **Status:** Proje hangi durumunda olduğunu proje liste sayfasında görüntüleyebilmek için Backlog, Open veya Todo durumunda olan seçenekleri tıklanması gerekmektedir.
- IV. **Description:** Bu bölümde proje bilgileri ve ne tür bir proje olacağından, gibi bilgileri girilmelidir.
- V. **Add Sprint:** Şekil 5.7 de gösterildiği gibi eğer oluşan herhangi bir sprint hızlı bir ekleme pop-up vasıtasıyla projeye ait sprintler eklenebilir.
- VI. **Kaydetme (Save) Butonu:** Burada kullanıcı tüm bilgileri girdikten sonra Firebase 'ye bilgiler kaydedilmektedir.

**Şekil 5.7: Projeye ait sprint eklemek**



The screenshot shows a mobile application interface. At the top, there is a dark blue header with a white 'X' icon and the text 'Add Project'. Below this, the main content area is titled 'Project Detail'. A white dialog box is overlaid on the screen, titled 'Adding Quick Sprint'. Inside the dialog, there is a 'Sprint Name' input field, a 'Status' section with three radio button options: 'Backlog', 'Open', and 'ToDo', and three more input fields: 'Component', 'Assign to', and 'Reporter'. At the bottom right of the dialog are two buttons: 'CANCEL' and 'SAVE'. Below the dialog, a dark blue button with the text 'Save' is visible. The bottom of the screen shows the Android navigation bar with back, home, and recent apps icons.

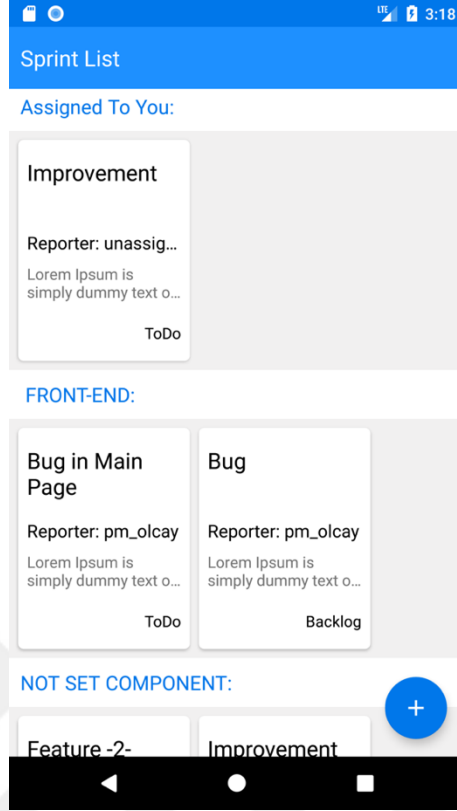
Şekil 5.7 'de görüntülenen pop-up proje ekleme esnasında süreci biraz daha hızlandırabilmek adına sprintler eklenebilir. Sprint eklerken doldurabileceğimiz bilgileri;

- I. **Sprint Name:** Sprint Adı girilmelidir
- II. **Status:** Sprint hangi durumda göstermek adına Backlog, Open veya to-do durumları seçilmelidir.
- III. **Component (Bileşen):** Kullanıcı sprinti tanımlandığında ne tür bir sprint olduğunu belirtecek bir alandır. Tıkladığında belirli liste açılacaktır. Listede kullanıcı firebase de tanımlanan kategoriler belirlenecektir. (front-end, back-end, mobil).
- IV. **Assign To:** Bu bölümde sprinti gerekli tüm fonksiyonlarıyla tamamlayacak kişiler seçilir. Bu alana tıkladığınızda kullanıcı listesi oluşacaktır ve sorumlu olan kişiler seçilmelidir.
- V. **Reporter:** Sprinti atanmış kişinin gerekli bilgileri sağlayabileceği bir kaynak olmalıdır. Bu alana da tıkladığınızda kullanıcı listesi açılacaktır ve sorumlu kişiyi seçip geliştirme yapacak kişiye kaynak olmalıdır.

## 5.5 SPRINT LİSTESİ

Bu bölümde şekil 5.8 de de görüldüğü üzere kullanıcı burada oluşturulan projenin sprint listesini gösterilmektedir. Burada görüntülenen sprintler sprint adlarıyla atanmış kişiye bağlı olarak sprintleri öncelikli olarak görecektir aynı zamanda diğer kullanıcılara atanmış sprintleri görebilecektir. Ayrıca kullanıcı sprintlerinin hangi durumda olduğunu görebilmesi için listede sprint durumları görüntülenmektedir. Sayfanın sağ alt bölümünde bulunan kullanıcı tipine bakılmaksızın sprint ekleme sayfası açılmaktadır bölüm 5.6 da detaylı olarak anlatılmıştır.

### Şekil 5.8: Sprint liste ekran görüntüsü



Kullanıcılar tarafından girilen sprintler kategorilere ayrılır. Sprintleri ekleme aşamasında kullanıcı bileşenleri girdiği takdirde girilen sprintler şekil 5.8 de olduğu gibi kategorilere ayrılır. Burada kullanıcı ara yüz, servisler, mobil veya hiç bileşen girmeden sprintler ekleyebilir.

Ayrıca uygulamaya giriş yapan kullanıcıya tanımlanan sprintler varsa sıralamanın en başında kendisine atanan sprintleri görebilecektir.

#### 5.6 SPRINT DETAY SAYFASI

Sprint detay sayfası sprint ekleme sayfası şekil 5.9 da olduğu gibidir. Bu sayfada kullanıcı sprint detayını güncelleyebilir veya şekil 5.8 de olduğu sağ alt bölümdeki + butonuna tıklayarak yeni bir sprint ekleyebilir. Bu duruma bağlı olarak şekil 5.8 de görüldüğü gibi sağ üst bölgede yazı değişmektedir. Yani kullanıcı eğer bir sprinti güncellemek isterse yazı “update” yazıyor olacak ve Firebase veri tabanında güncelleme gerçekleşmiş olacaktır. Eğer ki kullanıcı yeni bir sprint eklemek isterse yazı “ADD” yazıyor olacaktır ve Firebase bir Json olarak yeni bir alan eklemiş olacaktır.

**Şekil 5.9: Sprint Detay Sayfası**

Sprint Name  
Sample Subject

Assign to Reporter  
dv\_olcay Sample Reporter

Status

Backlog  Open  
 ToDo  In Progress  
 In Review

Description

Comments

Write a comment

Sprint detay sayfasında sprint eklemek şu alanlar doldurulmalıdır:

- I. **Sprint Name:** Sprint'in adı yazılmalıdır
- II. **Assign to:** Hangi geliştiriciye atandığına dair geliştirici adı yazılmalıdır.
- III. **Reporter:** Sprinti oluşturan kullanıcının adı veya sprintin sorumlu olan kullanıcı adı yazılmalıdır.
- IV. **Status:** Burada kullanıcı Sprintin durumunu Backlog, Open, todo, In Progress, In Review seçenekleriyle belirtmelidir. Dikkat edilmesi gereken geliştirme takımından olan kullanıcı sadece todo, In Progress ve In review durumunu seçebilmelidir. Tekrar Open veya todo ya çekmek için Proje yöneticiler tarafından yapılması gerekir.
- V. **Description:** Burada Sprint hedefleri yazılmalıdır. Yani Sprintten beklentileri belirterek geliştiriciye gerekli bilgiler verilmelidir.
- VI. **Comment:** Burada atanmış veya reporter arasında gerçekleşen konuşmalar tutulur veya yazılır. In Prohress esnasında çıkan herhangi bir durum esnasında durumu anlatan yorum yazılıp ilgili kişiye haber verilir.

## 6 SONUÇ VE TARTIŞMA

Bu proje Scrum Çerçeve metodolojileri doğrultusunda değişime ve sürece hızlı bir şekilde adapta olmak amacıyla geliştirilmiştir. Sprintlerdeki süreçlerin takibi ve geri bildirim hızı hızlı olacağını amaçlamaktadır.

Uygulamada yer alan durumu yani proje veya sprint durumları Scrum Takımı için ortak bir konuşma dili yaratmaktadır. Bu aslında Scrum Çerçeve deki transparan olma özelliğinden gelmektedir. Çünkü Scrum takımı sprintteki süreçleri hızlandırabilmek için süreçlerdeki aksiyonlar bir o kadar açık olmalıdır. Buradaki ortak yorumlama prensibi her Scrum Takımı için anlaşılır olması şarttır.

Sprint iş yapım süresi esnasında yorumlar yazılabiliyor olması Scrum Çerçeve' un geri bildirim özelliğindeki temeller yatmaktadır. Geliştirici takımın iş yapım esnasında karşılaştıkları herhangi bir durum esnasında anlık bildirimlerin kullanıcıya ulaşılır olması projedeki iş bitiş süresinden aksaklık yaşamamasını süresini kısıtlar. Herhangi sapma olduğu tespit edildiği esnada durum değerlendirme toplantısı düzenlenerek sorunun çözüme kavuşması süresini hızlandırır.

Projedeki test aşamasındayken karşılanan sorunların sprintleri hızlıca oluşturmak için bu projenin uygulaması yardımcı olabilecektir. Anında müdahale için ekleme menüsünden sprintin gerekli bilgileriyle geliştirme takımına ulaştırılır.

Sonuç olarak projenin kaliteli ve verimli bir ürün oluşturmak iyi bir proje yönetimden gelir. Dolayısıyla proje yönetimi ne kadar etkin ve hızlı olursa proje de bir o kadar hızlı çözümlere ulaşır. Scrum Çerçeve un kurucuları bu konuda geliştirme yaparken başarısız olan bir proje yönetimin sonuçlarından meydana gelmiştir. Bu uygulama temelinde aynı hedefler yer almaktadır. Başarılı bir proje yönetimi başarılı bir proje süreçlerinden geçer.

## KAYNAKÇA

### *Sürelî Yayınlar*

- Elbool M., Selçukcan Erol Ç., 2017, *Scrum Metodu kullanılarak Bir Mobil Uygulama Geliştirme Sürecinin gerçekleştirilmesi*, Bilişim Teknolojiler Dergisi, **10(2)**, pp 169-176
- Permana Adi Guna P., 2015, *Scrum Method Implentation in a Software Development Porject*, International Journal of Advanced Computer Science and Applications, **6(9)**, pp 198-204
- Beck, K., 1999. Embracing change with extreme programming. *IEEE Computer*, **32(10)**, ss. 70–77.
- Bissi, W., Serra Seca Neto, A. G. ve Emer, M. C. F. P., 2016. The effects of test driven development on internal quality, external quality and productivity: A systematic review. *Information and Software Technology*, **74**, ss. 45–54.
- Campanelli, A. S. & Parreiras, F. S., 2015. Agile methods tailoring – A systematic literature review. *Journal of Systems and Software*, **110**, ss. 85-100.
- Cohen, D., Lindvall, M. ve Costa, P., 2004. An Introduction to Agile Methods. *Advances in Comp.*, **62**, ss. 1–66.

## ***Diğer Yayınlar***

Scrum-Master.de, Agiles Projectmanagment, [https://scrum-master.de/Was\\_ist\\_Scrum/Scrum\\_auf\\_einer\\_Seite\\_erklaert](https://scrum-master.de/Was_ist_Scrum/Scrum_auf_einer_Seite_erklaert).

[erişim tarihi 01 Aralık 2018].

Akçicek R., 2012, *Agile Project Yönetimi nedir*,

<https://ridvanakcicek.wordpress.com/2012/12/10/agile-proje-yonetimi-nedir/>.

[erişim tarihi 15 Kasım 2018].

Benek O., 2016, *Agile dönüşüm aşamaları ve kazanımlar*.

<https://omurbenek.com/2016/03/06/agile-donusum-asamalari-ve-kazanimler/>

[erişim tarihi 02 Aralık 2018].

PEM, 13 Maddede neden agile proje yönetimi.

<http://www.projeegitimmerkezi.com/bdetail-d-275-13-maddede-neden-agile-proje-yonetimi>

[erişim tarihi 02 Aralık 2018].

PMIS, What is agile Project managment

<https://www.pmis-consulting.com/articles/agile-project-management/>

[erişim tarihi 20 Ekim 2018].

Kayer T., 2012, *Birisi çevik yönetim mi dedi?*

<https://turgaykayaer.wordpress.com/2012/12/19/birisi-cevik-yonetim-agile-management-mi-dedi/>

[erişim tarihi 20 Ekim 2018].

Kaya G., 2018, *Agile çevik proje yönetimi yaklaşımı, Agile Manifesto?*

<http://www.projesehri.com/agile-cevik-proje-yonetimi-yaklasimi-agile-manifesto-scrum/>

[erişim tarihi 21 Ekim 2018].

Effective, Scrum Events.

<http://www.effectivepmc.com/blog/scrum/scrum-events>

[erişim tarihi 20 Eylül 2018].

DevTeam.Space, How to build a Scrum Development team?,

<https://www.devteam.space/blog/how-to-build-a-scrum-development-team/>

[erişim tarihi 02 Aralık 2018].

Rajagopalan S. , Mathew S., *Choice of Agile Methodologies in Software Development: A Vendor Perspective*

<https://pdfs.semanticscholar.org/d161/50e945f5f07fd5352b035946e207361022a8.pdf>

[eriřim tarihi 10 Kasım 2018].

Neelima E. , Saile Durga Naga K., 2013, *Agile Methodology and its knowledge process*,

<http://www.theijes.com/papers/v2-i3/E023022027.pdf>

[eriřim tarihi 5 Kasım 2018].

