



**Düzce Üniversitesi**  
**Mühendislik Fakültesi**  
**Bilgisayar Mühendisliği Bölümü**

**Yazılım Mühendisliği**

**BM 310 –Vize Raporu**

**Teslim Eden:**

**İsim-Soyisim** : **Mert Aktaş**

**Numara** : **171001037**

## İÇİNDEKİLER

SCRUM .....	4
TARİHSEL TEMELLERİ .....	4
SCRUM ÇERÇEVESİNİN BİZE SUNDUKLARI .....	5
SCRUM YAPISI .....	6
Kavramlar .....	6
Scrum Board .....	7
ROLLER .....	8
TOPLANTILAR.....	8
SCRUM'IN SINIRLARI .....	9
Başarı Garantisinin Olmaması .....	9
Öğrenilen Dersler ve Değerlendirme .....	9
Takım Bileşimini Engelleyen Faktörler .....	9
Test Maliyeti .....	10
Hukuki Hususlar .....	10
SCRUM KULLANAN FİRMALAR.....	10
RATIONAL UNIFIED PROCESS (RUP) .....	11
RUP'UN TARİHİ.....	11
RUP NEDİR? .....	11
RUP MİMARİSİ.....	12
RUP SÜRECİ .....	12
BAŞLAMAK.....	13
OLGUNLAŞMAK.....	13
YAPIM SÜRECİ.....	13
GEÇİŞ SÜRECİ.....	13
RUP'TA EN İYİ ALTI UYGULAMA YÖNTEMİ .....	13
Yinelemeli Olarak Geliştirmek .....	13
Gereksinimleri Yönetmek.....	14
Bileşenleri Kullanmak .....	14
Görsel Olarak Modellemek.....	14
Kaliteyi Doğrulamak .....	14
Kontrol Değişiklikleri Yapılmalı .....	14
RUP KULLANAN FİRMALAR.....	14
EXTREME PROGRAMMING .....	14

XP DEĞERLERİ .....	15
XP PRENSİPLERİ .....	16
Hızlı Geridönüm .....	16
Basitliği Tercih Etme.....	16
Artırımlı Değişiklik .....	16
Değişimi İstemek .....	16
Kaliteli İş .....	17
Öğrenmeyi Öğretmek .....	17
Az Başlangıç Yatırımı .....	17
Kazanmak İçin Oyna .....	17
Somut Denemeler.....	17
Sorumluluk Üstlenmek.....	17
Doğru Ölçüm .....	18
XP TEKNİKLERİ .....	18
Ayakta Toplantı.....	18
Kısa Aralıklarla Yeni Sürüm .....	18
Geriye Bakış .....	18
Mecaz .....	19
Ortak Sorumluluk.....	19
Sürekli Entegrasyon .....	19
Kod Standartları .....	19
Kalıcı Tempo .....	19
Test Etmek .....	20
Sade Tasarım.....	20
Yeniden Yapılandırma .....	20
XP ROLLERİ .....	20
Müşteri.....	20
Yazılımcı .....	21
Proje Menajeri.....	22
Koç .....	22
Testçi .....	22
Haklar ve Sorumluluklar .....	22
Müşteri Hakları .....	22
Yazılımcı Hakları .....	22
SÜRECİN İŞLEYİŞİ .....	23

XP PROJE SAFHALARI .....	23
EXTREME PROGRAMMING KULLANAN FİRMALAR.....	24
AGİLE .....	24
TARİHİ .....	24
AGİLENİN TANIMI .....	25
AGİLE YARARLARI .....	26
AGİLENİN FAYDALARI.....	27
Paydaş Katılımı .....	27
Kullanıcı Odaklı.....	27
Erken ve Öngörülebilir Teslimat.....	27
Öngörülebilir Maliyetler ve Takvim .....	27
Değişime İzin Verir .....	27
Şeffaflık .....	27
Kalite Artışı .....	28
İş Değerine Odaklanır .....	28
AGİLE KULLANANLAR.....	28
KARŞILAŞTIRMALAR.....	28
SCRUM VS RUP .....	28
RUP VS AGİLE .....	29
SCRUM VS AGİLE .....	30
SCRUM VS XP.....	30
XP VS RUP.....	32
XP VS AGİLE .....	33
KAYNAKÇA.....	35

## SCRUM

Agilenin alt dalıdır yazılımları geliştirmek için kullanılan bazı yöntemlerden biridir. Projelerin oldukça kısa olan sürelerde ve en kalitesinin yüksek olmasını sağlar. Değişen koşullara kolay uyum sağlar. Bu yöntemin en harika olan özelliği ise gözlemlere ve tekrarlara bağlı olmasıdır. Ayrıca da değişimlere ve gelişimlere açık olması da önemli özellikleridir.

## TARİHSEL TEMELLERİ

Ken Schwaber ve Jeff Sutherland isimli yazılımcıların 1990'ların başında geliştirdiği yöntemdir. Schwaber mantığını ortaya koymuştur. Scrum projesinin rolleri:

- Product Owner ( Ürün Sahibi )
- Team ( Takım )
- Scrum Master ( Scrum Ustası )

Proje yönetici rolü geleneksel bir şeydir ve bulunmaz. Scrum da projenin ilerleyişi sprint denilen en fazla bir ay süren sürelerde düzenler. Çevik manifesto burda vücut bulmaktadır.

Agile Yazılım Geliştirme Manifestosu'na göre;

Süreler ve araçlara göre, bireyler ve etkileşimler,

Kapsamlı dokümantasyona göre, çalışan yazılım,

Sozlesme pazarlıklarına göre, müşteri ile işbirliği,

Bir plana bağlı olmaktansa, değişime cevap vermek,

daha değerlidir.[7]

Yaygın kullanım önce proje geliştirmede görülmüştür. Zamanla her alanda uygulanabilir olduğu görülmüştür. Günümüzde hayatın her alanında yazılımın kullanıldığını görmekteyiz. [1]

Scrum'ın İngilizce kelime anlamıysa “ıtısıp kakısma”dır. Rugby’de oluşturulmuş olan küçük ekiplerin adı olarak da kullanılır. Takım toplanır,plan yapılır, görev dağılımı yapılır ve herkes ortak bir amaç uğruna çalışmaya başlar. [1]

Bu metod projelerin oldukça karmaşık olduğunu savuran bir metoddur. Olası bir hatada tüm projenin baştan planlanmasının zor olduğunu savunur ve oluşan karmaşıklığı engellemek ve en aza indirmek için 3 tane aşama geliştirilmiştir.

- Seffaflık: projenin ilerleyişi ve çıkan problemler günlük olarak kayıt edilmeli ve herkes bunu görebilmelidir.
- Denetleme: proje parça olarak veya functionları düzenli şekilde teslimi sağlanır ve değerlendirilme aşamasına geçilir.
- Uyarılma: projenin ihtiyaçları belirlenir bu projenin başında yapılmaz proje müşteriye her sunulduğunda tekrar değerlendirme aşamasına girer bunun sonucuna görede yeni düzenlemeler yapılır.

Amac her zaman en basta hayal edilmiş olan ve tasarlanmış olana uygun olan projenin en maliyetsiz ve en kaliteli olarak üretilmesi sağlanmalıdır. Tasarlanmış olan projenin hayata geçirilmesi için müşterinin açık bir şekilde ihtiyaçlarını belirten bir listeyi hazırlamış olması gerekmektedir. Talep listesini takıma verir. Takım buna göre projeyi aşamalı olarak olamaz. İki veya dört hafta süren sprint süreleri ile bu istekler müşteriye aşamalı şekilde teslimi gerçekleştirilir. müşterinin istekleri kullanıcı hikayesi olarak bilinir. Her sprint döneminde projenin bir kısmı bitmiş olur. Scrum yöntemler arasında en başarılı olanlarındandır. [2]

## SCRUM CERCEVESİNİN BİZE SUNDUKLARI

- İlk assama sisteme mustırının de dahhıl edilmesi ve ıstklerin daha kolay anlasılması saglanır.
- İki veyahut dort haaftalık aralıkkarla geliştirilmiş olan urunlerin teslimatı saglanır, musternin gerı bildirımı alınır ve mustırının isteklrine zamanında kaarsılık verilir.
- Takkım ıcındaki yazılmmcıların, analısttlerin ve mımaarların bir arrada calıssması ve sureekli sekilde iletısım haalinde olması, ayrıca yasaanan bırır proplemde kısılerin birbirlerine en kıssa zamanda ulaşabilmesi ve yassanan problemlerin cozulmesi saglanır.

- Kendi kendini organize edebilen ekiplerin oluşturulması sağlanmalıdır. görev bilincinin artması, projeyi benimsemeleri ve ekiplerin üzerindeki olusan baskının azalması sağlanır.
- Günlük olarak yapılabilecek kısa ve verimli toplantılar; ekibin yüz yüze bir iletişim halinde bulunmasını, yapılan ve ileride yapılacak işlerin en doğru ve en hızlı şekilde aktarılması ve ekibe olan sorumlulukta artmasını sağlar.
- Düzenli aralıklarla yapılan ekibin toplantıları ve proje değerlendirme toplantıları; sorunların belirlenmesi ve gerekli iyileştirmelerinin yapılmasına olanak sağlar.
- Scrum Board kullanımı; yapılacak olan, yapılmış olan ve bitmiş olan işlerin gösterilmesini, işlerin herkes tarafından takibinin yapılmasını ve işlerin kendi arasında da önceliklendirilmesini sağlar.[1]

## SCRUM YAPISI

### Kavramlar

1)Product Backlog; Projenin gerçekleştirilmesi için lazım olan gereksinimlerin bir listesi olarak bilinir. Projenin sonunda sorulara cevap aranır. Product owner müşteri den isteklerini alır önceliğine göre sıralar. Product owner, scrum takımındaki hangi işlerin yapılacağını belirleyen kişi olarak bilinir. Müşterinin ve projenin değişen ihtiyaçlarına göre listeye ekleme çıkarma işlemi yapabilir. Bu şekilde projenin her yeri değişime uğramış olur. Bu kolay bir şekilde gerçekleşir.

2)Product Backlog Item; Product backlogta gereksinimlere verilen addır..

3)Sprint(Kosma); Proje sprintlere ayrılır. Scrumun içindeki aktivitelerin hepsi sprintte gerçekleşmektedir. 1–2 haftalık süreçtir.

4)Sprint Backlog; projeyi Geliştiren takım product backlog maddelerini öncelik durumuna ve sıralarına göre sprintin içine alır. Bir sprint sürecinde yapılacak olan işlerin listesini oluşturmaktadır. İşlerin ayrıntılı şekilde zaman çizelgesi çıkartılır.

5)Scrum Board; Bir sprint dönemi içindeki yapılması gereken işlerin maddeleri bu bölümde yönetilir. Yapılacak olan işler “TO DO” alanına koyulur. Takım üyeleri bu işi yapmaya başladığı zaman “IN PROGRESS” bölümüne koyulur. Yapılan iş, test aşaması için hazır olursa “TO VERIFY” konumuna getirilir. İşin durumu kontrol edilir ve sonrada

“DONE” bolumune koyulur. Scrum toplantılarında bu maddelerin durumuna gore olan yerleri degıstırılebılır.[3]

6)Burndown Chart; Belli zamanda yapılması planlanan ıı ve zaman ılııkılarını takip ve kontrol etme amacıyla kullanılmakta olan grafiksel sunumdur. Dıkeydeki ekseninde puan degeri bulunur, yatayda ekseninde de zaman degeri bulunur. Takımlar belirlenmiř hızlarının ışıında olusturmus oldukları Sprint Backlog’ta bulunan ııleri gınlık řekilde bu grafik uzerinden takip etmektedir. Sprint Backlog’ta bulunan ıılerin puanından baslayarak, sprintlerin son ařamasında sıfıra ulasması beklenılır.

İdeal olan bır Burndown Chart grafığı yapmak , bır takımda oldukca zaman alan bır surec olabılmektedir. İdeal olanı yakalayabilmesi ıcınndeki takım ıcı iletısımının daıma mükkemlel řekilde olması gerekir. Takımında kendısını cok ıyı tanımıř ve takım hızının tahminin iyi olması gerekmektedir. Cok ıyı hazırlanan bır Product Backlog ve aktif bır Scrum Master’ın olması proje ačíından önemlidir. [4] Burndown Chart Scrum’un temel ilkelerinden olan seffaflığı saglar.

#### Scrum Board

TO DO (Yapılacaklar)	IN PROGRESS (Yapılanlar)	DONE (Tamamlananlar)
A B C E H	F J L M	G K Z

Tablo 1.1

Scrum Board'un kullanımı sırketler arasında oldukca yaygındır. Hem dijital ortamda verilerin tutulması saglar, hem takımının kolay olmasını saglar hem de gorsel acıdan kolaylık saglaması da tercih edilme nedenleri arasında bulunuyor. Yaygın olarak kullanılan Scrum Board'lra ornek olarak Asana ve Jıra'yı verılebılır.[1]



## ROLLER

Pig Roller; Scrum sureccine dahil olan kisiler yanni procede assıl ısı yapan kisilerdir.

Product Owner; Gelstirme takımıyla mustrı arasndakı iletisimini sagglar. Projenn tum ozellklerini tanımlamaktadır. Projenn onceliklerı ne ise ona gorre product backlogu olustururlar. Sprint'ı ıbtal edebilme yetksine sahiptir.

Scrum Master; ozellikle Scrum kurallarını ve teorlerini ve bütün pratiklerini ıyı bılır ve takımın bu kuralları uygulamasndan sormlu olan kisıdır. Takım yönetcısı degldır ve Takımmı rahadsız eden ve vermlı calısmalarrını engellyen olayları ortatan kaldırakla gorevlıdır.

Gelistirme Takımı; Bir Sprint'e alımmıs olan tum ıslerin tamamlayabilecck ozelliklerine sahip olan kisilerdir. Sprint Backlog'u olustururlar. Kendı kendilerini yönetebilirler. Genellikle ısın verilmesini beklemezler ve ısı kendilerine alıb gelistırlir. Kısılın tek bir gorrevlerı yoktur, karısık sekilde gorrev dagıglımı yaparrlar ve takımdakı heerkes herseyı yapıabilir durumda olmalıdır Takım 5-7 kısı arrasında degısır. Projenin gelstırılması ile ilgili tum sorunuluk gelstirme takımmına ait olur.

Chicken Roller; Scrum'ın ıslaysinde aktf olarak yeer almnyan kisilere denır. Mustrıler ve satccılar gıbı ornekler verilebilir.[3]

## TOPLANTILAR

Sprint Planning; Product backlog ile belırtılen gereksinimlerin, bu toplandı ile gelısdırme takımmı tarafından kuccuk goreevlere (task) ayrılmasıdır. Takımmıdaki her kısı kendi hızınna uygun olan gorevlerı kendıllerine alır. Toplantı Product Ownerın Gelistirme Takımını vede Scrum Master'ın katılımıyla gerceklestırılır. Sprintler; her sprının sonunda Product Owner'a sunulcakmak uzere yazmlım gelıstrmeyi hedfleyerek belrlenır.1-3 hftalık sprint olustrulur.

2) Daily Scrum; Her bir gun aynı yerde ve aynı saatlerde ayakustu yapılan 15'er dakika süüren ufak toplantılardır. Uyelerin hiçbirı davet edılmeyı beklemez kendılerı katılırlar. Bu toplantı onlerindeki 24 saatlerini planlamak ıcın yapılır. Takımdakı üyelerin hepsi dun ne yaptığını, bugün neleri yapacağını planlar ve ısını engelleyecek olan herhangi bir sorun var mı diye düşünür. Bu sayede eger herhangi bir sorunu var ise Scrum Master bu sorunu ortadan kaldırır ve uygun calısma ortamı saglar. Takım uyelerı arasında bu soruna

yardımcı olabilecek birisi var ise toplantının sonunda iletişime geçilebilir. Daily Scrum koşullar neler olursa olsunki yapılır. Takımdakilerden birinin işe geç kalması veya katılmaması toplantıyı etkilemez. Sadece takımdakilerin eksikliği ise toplantı yapılmaz.

3)Sprint Review; Her sprintin sonunda mutlaka yapılır. Yapılan tüm sprintler gözden geçirilir ve ortaya çıkanlar değerlendirilir. Amacı yazılımın müşteri isteklerine uygun şekilde geliştirilmiş olduğundan emin olmak ve projenin iyi olmasıdır. Eğer projede bir hata var ise farkedilerek ve hemen uygun bir şekilde düzeltilir.

4)Sprint Retrospective; Sprint dönemi boyunca yapılmış olan tüm işlerin kalitelerinin doğrularının ve yanlışlarının değerlendirilmiş olduğu toplantıdır. Bu toplantı takımın kendini olumlu şekilde geliştirebilmesi için güzel bir fırsattır. Bu toplantıda “Neler dahada iyi yapılabilir?”, “Nasıl dahada iyi yapılabilir?” sorularına cevap aranır. Bu asamalardan sonraki toplantı tekrar olarak yapılır ve eksikler giderilir.[3]

## SCRUM'IN SINIRLARI

### Basarı Garantisinin Olmaması

Scrum diğer surec modelleri gibi daha az başarı garantisi sunar ve uygulamasıyla seffaflık ve düzenli ürün teslimatı oluşturur. Görülebilen engeller ile ürün adım adım geliştirilir ve gerektiğinde değiştirilebilir olmasını sağlar.

### Öğrenilen Dersler ve Değerlendirme

Scrum kullanılırken, kişi kendine özgün tahminlerini kalıcı ve sürekli hale getirir. Scrum ilk günden itibaren ürün geliştirmede olması gereken şekilde sapmaları gösterir. Ürünün hızlı, iyi, uygun fiyatlı veya yüksek kalitede olması takımın kazandığı deneyimlere bağlıdır.

### Takım Bileşimini Engelleyen Faktörler

Scrum'ın uygulanmasını bazı faktörler engelleyebilir. Geliştirme takımı kendi kendini organize etme ilkesiyle şekillenir.

Scrum'da proje ekipleri sprint'deki tüm görevlerin üzerinde birlikte çalışırlar ve böylelikle testçi, yazılımcı ve tasarımcı uyumu da takıma doğrudan yansır. Ancak deneyimli bir takım ise dezavantajları telafi edebilir ve avantaja çevirebilir.

## Test Maliyeti

Sık gelen deęisiklik talepleri, iyileştirme ve bırım testleri gerektirir. Sık olan teslimatlarda yeterli regresyon testi kullanılır, test otomasyonu cogunlukla dogru sonuc vermez. Bu yuzden manuel testler her Sprint sonucunda tekrarlanmalıdır ve boylece Scrum'da etkin test calısması onlenir.

## Hukukı Hususlar

Scrum uygulaması birçok yasal sınırlamalar icerir. Yazılımın geliştirilmesi surecinde, özellikle proje sıparısı asamasında, meslekı ve proje formulasyonu vs. baglayıcı olan hedefler belirlenmez. Hukukı acıdan yapılacak ısın ve kabul kriterlerinin uzerinden yaklaşılr ve geleneksel prosedurleri gecerlıdır. İş sozlesmesi cercevesindeki anlaşılmazlıkların kaynagı, sozlesmeye uygun net ve dogru beyanın verilmemesi ya da sorumlulukların belirsizlığı neden olur.[5]

## SCRUM KULLANAN FİRMALAR

- |                    |                             |
|--------------------|-----------------------------|
| •Microsoft         | •Intuit                     |
| •Yahoo             | •Nielsen Media              |
| •Google            | •First American Real Estate |
| •Electronic Arts   | •BMC Software               |
| •High Moon Studios | •Ipswitch                   |
| •Lockheed Martin   | •John Deere                 |
| •Philips           | •Lexis Nexis                |
| •Siemens           | •Sabre                      |
| •Nokia.            | •Salesforce.com             |
| •Capital One       | •Time Warner                |
| •BBC               | •Turner Broadcasting        |
| •Intuit            | •Oc[6]                      |

## RATIONAL UNIFIED PROCESS (RUP)

Türkçe anlam olarak Rasyonel Birleştirilmiş İşlem olarak bilinir. RUP; 2003 yılından bu yana IBM'in bir bölümü tarafından oluşturulan bir tekrarlanabilen yazılım geliştirme süreci çerçevesidir.

## RUP'UN TARİHİ

Rational Software başlangıçta bir yazılım süreci ürünü olarak geliştirmiştir. Ürün, birçok birbirinden farklı türde örnek olacak şekilde içerir. Ayrıca ayrıntılı olarak açıklanabilir içeren bir köprü bağlantılı bilgi tabanı içerir. RUP, sürecin kişiselleştirilmesini sağlayan IBM Rational Method Composer (RMC) ürününe dahil olan bir süreçtir.

Deneyimli bir Rasyonel teknik temsilcisi olan Philippe Kruchten, orijinal RUP ekibine liderlik etmekle görevlendirilmiştir. Bu süreç 1996'da Rasyonel'in Ivor Jacobson ve şirketi tarafından yazılmış olan Objectory Süreci'ni satın aldığı Rational Objectory Process'ın (ROP) oluşturulmasıyla başladı . Bu, sonraki sürümlerde, adı Birleşik Modelleme Dili ile nispeten benzetmek için Rational Unified Process (RUP) olarak tekrardan adlandırıldı.[8]

## RUP NEDİR?

Rational firması tarafından geliştirilen RUP, organizasyon içerisindeki sorumluluklarını detaylı olarak belirler ve disiplinli bir yazılım geliştirme süreci sunar. Yazılım geliştirme sürecinin her bir aşamasında kullanılabilecek araçları, hazır belge şablonu sayesinde proje ekibinin üretkenliğini oldukça arttırmaktadır. RUP içindeki aktivite gruplarına ise "disiplin" adı verilmektedir. Gereksinimler, analizler, tasarımlar, gerçekleştirmeler, test temel disiplinlerine örnek olarak verilir. RUP içindeki yazılım süreçleri 4 aşamadan oluşur;

- Başlangıç (Inception): Yaklaşık olarak tahminler yapılarak en temel gereksinimler ortaya koyulmaya çalışılır.
- Düzenleme (Elaboration): Daha gerçekçi tahminler yapılarak mimarı belirlenip ve yüksek riskler için çözüm sunulur.
- Oluşturma (construction): Ürün hazır bir hale getirilerek içindeki hatalar belirlenir.

- Gecis (transition): Musteriye urunun teslim edildiği ve müşteri den gelen geri bildiri me gore iyilestirmelerinin yapıldığı asama dır.

RUP ı c erisindeki her bir âsâmâdâ yâzılım muhendisliginin tum âktivite leri fârk lı orândâ uygulanır. Klâsik bir yâzılım gelistirme sureci olân câglâyân modelindeyse her bir âktivite bir onceki âktivitenin bitmesinden son râ gerceklestirilmektedir. Câglâyân modelinde kodlâmâ âktivitesi tâsârımdân son râ yapilirken; RUP ı c inde her bir âsâmâdâ kodlâmâ yapılabılır. RUP surecinde olân âsâmâların her birinin âmâclârı ogrenildiginden oturu her bir âsâmânın sonucundâ olusturulâbilmesi gereken cıktılâr sıralânâbilmektedir. Bâslângıc âsâmâsındân son râ proje plânı olusturulms olmalıdır. Duzenlme âsâmâsının bitmesiyle birlikte kullânımının senâryolârının ve âktorlerin yer âldığı model, tâsârım modellerı, yâzılım mımârıların tânımı veri modellerı, yeniden duzenlenmis olân risklerin listesi, kullânıcı ârâ yuz prototıpleri ve yâzılım gelistirme plânının hâzırlânms olmasi gerekir. Olusturulmâ âsâmâsı sonundâ kâynâk kodun, birim testlerin, test senâryolârının, test kodlârının, son kullânıcıâ verilecek olân urunun, kullânıcı belgesi olusturmâsı beklenir. Gecis âsâmâlarının bitiminde müşteri memnuniyeti sâptânmalıdır ve isteklerin ne derecede kârsılânılmıs olduđu, plânlâmâlarının bâsârlı olup olmâdğı her zâmân ânâlız edilmelidir. Âsâmâların sonundâ hâzırlânılmâsı gereken tum belgelerin ve sureclerin ı c indeki gelistiricilerin rollerindeki fâzlâlık RUP'un sâdeceâ buyuk olccklı olân firmâlarâ ozgu olduđu du dundurmektedir. Fâkât RUPun degisikligine ızın veren bir mımârısı olmasi sâysındede orgânizâsyon ı c in gerekli rol, âsâmâları tânımlânârâk kendi birlestirilen sureci olusturmâk mumkun olâbılır.[9]

## RUP MİMARİSİ

Bu surec normal olan bir yazılım urunu gibi tasarlanmıştır. Temelinde UML ile belirtilen nesne modellemesi ile iki boyuta sahiptir;

Yatay surec zamanını erteler. Projenin ilerleme oranı zaman açısından değerlendirilir.

Dikey surec ise mantıksal olarak yazılım muhendisligi aktivitelerini gruplamıs baslıca dışıplınleri acıklar.

## RUP SURECİ

Her surec tanımlanma ve tekrarlanan işlemleri ı c ermektedir. Tekrarlamıs işlemler belli bir bitis suresi ı c erir. Surecler ı c in birer amacın gerceklestirilmesi hedeflenir.

1. Baslamak
2. Olgunlaşmak
3. Yapım süreci
4. Gecis süreci

#### BASLAMAK

Bu surecte yâpılâcâk olân ısın cozumunun ne oldugunun, neyî nâsıl yâpmâk ıstedıgının, beklenen yârârlârın, gercekleştirme fiyatı, ısı yerının o ân olân durumu ile olmâyı ıstedıgı yer hâkkındâ kı ânâlız, beklenilen rıskler, yâzılım cozumlerinden beklenilen fâydâlâr ve kârlâr gıbı âktıvıtların gercekleşmesi sâglânır.

#### OLGUNLASMAK

Bu surecte proje yavas yavas olusmaya baslar ve saha analizıyle temel yazılım mimarısı ortaya çıkmaya baslar.

#### YAPIM SURECİ

Bu surecte, kodu geliştirme yapılarak çalışan bır sistemi ortaya koyma amacı ıcine girilir.

#### GEÇİS SURECİ

Bu surecte geliştirilmiş olân uygulâmâ musretiye ulâstırılır. Musteriye sistem hâkkındâ gerekli bilgiyle ve eğitim verilir. Sistem test edilmeye bâslânılır. Urun dâhâ bâslângıctâ verilen surece ve belirlenen kâlite kriterine gore degerlendirilir.

Câlisân urun ile olmâsı beklenen urun ârâsındâ fârklılık kâlmâyıncâyâ kâdâr tum bu surecler bâstân sonâ kâdâr tekrârlânır.[10]

#### RUP'TA EN İYİ ALTI UYGULAMA YONTEMİ

Hataları olabıldığınce en aza indirmek ıcın ve verimliliği artırmak ıcın olan yazılım projeleri ıcın yazılım muhendisligi uygulamaları tanımlanmıştır. Bunlar;

##### Yınelemeli Olarak Geliştirmek

Tum gereksinimlerini onceden bilmek en ıyı yontemlerdendir; âncâk genelde durum boyle degildir. Geliştirme âsâmâlârı âcısındân mâliyetı olâbıldığınce en âzâ indirmek ıcın cozumler sunân bırkâc yâzılım geliştirme süreci bulunur.

## Gereksinimleri Yönetmek

Her zaman kullanıcılar tarafından belirlenmiş olan gereksinimler unutulmamalıdır.

## Bileşenleri Kullanmak

Gelişmiş olan bir projenin yıkılması sadece önerilme ile kalmaz aynı zamanda kaçınılmaz olur. Bu tek tek bileşenlerin daha büyük olan bir sisteme entegre edilmeden önce test etme yeteneğinin artmasını sağlar. Ayrıca da, kodun tekrardan kullanımı çok büyük bir artıdır. Nesne yönelimli olan programlama kullanılarak bu daha kolay gerçekleştirilebilir.

## Görsel Olarak Modellemek

Tüm ana bileşenleri, kullanıcıları ve etkileşimleri temsil etmek için diyagramların kullanılması önemlidir. Birleşik Modelleme Dili kısaltması olan "UML", bu görevi daha iyi uygulanabilir şekilde getirmek için kullanılabilecek uygun bir araçtır.

## Kaliteyi Doğrulamak

Testi her zaman projenin önemli bir parçası haline getirmek gerekir. Proje ilerledikçe testler zorlaşır, fakat herhangi bir yazılım ürünü oluştururken bu sabit bir faktör asla olmalıdır.

## Kontrol Değişiklikleri Yapılmalı

Birçok proje ekipleri tarafından, bazen farklı yerlerde, farklı platformlarda, vb. kullanılmaktadır. Sonuç olarak da, bir sistemde yapılan değişikliklerin her zaman sürekli olarak senkronize edilmesi ve doğrulanmasını sağlamak önemli bir adımdır.[11]

## RUP KULLANAN FİRMALAR

- |                           |                         |
|---------------------------|-------------------------|
| • Ülke Capında Sığorta    | • Brightstar Medya Org. |
| • Massachusetts Topluluğu | • Mars                  |
| • Amerika Bankası         | • Deemsys Inc.[12]      |

## EXTREME PROGRAMMING

En popüler olan çevik süreçler arasında XP olarak bilinmekte süreç Ekstrem Programlamadır. Kent Beck ve arkadaşları tarafından 1996 yılında Chrysler firması bünyesi altında yapılan bir projedir.

Extreme Programmingle olusturrulan bu cevık surecte musterıın gereksınmlerı merkezıı rol oynar. Yazılım surecında XP ile belirgn olmamakla ve cok cabuk degısıklıge ugrayabilen musterı ısteklerine ayak uydurması kolaylasır. Bu konvensıyonel olan yazılım metotlarında pek mumkun degıldr cunku projenin oncesinde musterıının ihtiyacları son detaylarna kadar kagıda dokulur. Olusan bırır dokumantasyon baz alınmakta, yazılım surecı gercekleştirılmıs olur. Proje surecı ilerledıkce musterıler tarafından yapılması istnen degısıklıklrın malıyetlerı cok yuuksek olabilir cunkuu bulunan mevcut yapı istenlen degısklerin dogru bır sekılde yapılmasını engelleyebilir. XP, kullanılan projelerde formalitenin ve burokrasinin mumgun olan az seviyeye gelmesine onm verr Cevk olablmeık ıcın cok az ykle yolaa cıkılması gerekir. Bu yzden projenin oncesinde gens cabta tasarm ve dokumantasyon olusturulmasına ızın verilmez. dokumantasyon ve tasarm olusturlmadan calısıldığı amlamna da gelmez. [13]

## XP DEGERLERİ

temelde drt deır uzerne kuruludur;

XPnin özzü bu dortt deır ıcındedir. Bu deırler gercekleştirildiğı zaman XP ogenımı kullanımı oldukca kolaylasır. Bu deırlern gecerlılığı olmayan ortamlrda XPnen uygulandğı pek fazla gorulemez XP ilee vermlı olan bır cevık surc olusturulabilmesi amacınla bu deırlern kabl gormessı ve uyulanması gerekir. XP genelde basit yontemlerle sonuca ulasmayı ıstr cunkk sadece yontemlerle hızlı ve de düşük malyetlıprojeler yapılabilir. Bununun yanı sıra bazı bast cozumler ile olustrlan programn bakımı ve gelstirilmesi daha kolay olur. Bazı cozmeler daha kolay anlatılabilir adapte edilir. Bu biraz zamanla kazandırması anlamna gelir.

Yazılım sektorunde en onmlı konulardan bırısı de kalitte kontrolu yapılmasıdır. Projelerde kalitte kontrolu geridonuslm ile saglanır. Yazılımcılar yazdıkları testlerden geridonuslm alarak kaliteli olmasını saglar ve Kısa surelerde yeni surumler olusturularak musterılerden geridonum ile programın ihtiyacları tatm edp, etmedğı kontrol edilir. Yazılım esnasında sreklı bır araya getirilerek programın ven zon kısı durumu hakkında geridonulm saglaanr. metodolojisinin uygulanabilmesi ıcın deıgik katmlarda geridonuml mekanizmalarının olusturulmus olması gerekmektedir. İnsan ıcın suwater neyse, XP ıcın geridonuml o olur.

Tumm projede çalışan kısılerin sreklı olaak aralarında iletisim halinde gerekir. Bireylerin arasında yuzyuze gorusmeler onemlidir. Bu sayeede saglıklı bııgı aktarımı gercekleşir.



Boylce yalmıs anlasımlar ortdan kaldırılmıs olur. Egr takmm ıcerisinde iletım gcclu ise, dkumantasyon olusturmak kullanmak gereksiz hale de gelebilir. olusan zamman kaybını onlemektedir. Dkumantasyon kullanımı beska sebeplerden dolayı gerekli hale gelebilir ammavelakin projenin basrısı ıcın dkumantasyon şart değildir. [13]

## XP PRENSİPLERİ

### Hızlı Geridonum

Sıklısma olarak hızlı olarak geridonum almak projenin gıdsatını olumlu etkilemektedir. Geridonum almak sayysinde yalmıs annamalar hataların ortdan kaldırılması sağlanır.

### Basıtlığı Tercih Etme

Bast cozzumler kollay bunyeye kabul edılır oldukca kssa zammnda olusturulabilir. Bu geridonumlm oldukca hızlı sekilde gerceklesmesini saglamaktadır. Bsıt olan czumlerin kvvranması ve anlatılması da cok dhaa klaydır. yazılımcılardan annkı ıstekleri tadmin etmek ıcın bastt olan cozzumu beklemektedir. Yazılımcı gelecekte olabilecek eklmeleri ve degısklıkları dusunmemelidir. O an ıcın elinden gelenin en iyısını yapmalıdır.

### Artırımlı Degısklık

Bast cozzomler uygulansa bllle yazılım sistemleri zamman gectıkce krmasık brr sekile donusur. Ypılan een uffak bir degısklık olsa brrle, sstemın dusunulmeyen bollumu uzerinde hata ollmasına sebep olabilir ve Olusan buu hataların kontrol altında tutulabilmesi ıcın degısklıkların uffak caplarda olması gerekir. Byuk degısklıklar berrabberinde sorunlar getrebilir. Bu sebepten oturu degısklıkların uffak caplarda yapılması gerekmektedir.

### Degısıımı İstemek

İlerleye bilmek ıcın kendimize brr yoon belirlememiz , yenlıklere de acık olmaız gerekir. Yenıklere de acık olmaak buyuk cezaret gerektirmektedir. Bılınmeyenlerle ugrsmak, rahatsız edccı bir sey ollabilir fakat bssarıyı elde eddeblmek ıcın degısıımı ıstmmek gerekmektedir.

## Kaliteli İş

Ekstrem Programlama projelerinde kaliteli bir iş ortaya konulabilecek bir ortamın oluşması gerekir. Hiç bir yazılımcı hatalı program yazmayı istemez. Çalışma ortamının etkisiyle de yüksek kalitede proje yapmak heenn yazılımcının özgüvenini artırmaya hem de müşteriyi tatmin edecek ürünlerin ortaya koyulmasını sağlar.

## Öğrenmeyi Öğretmek

yazılımcı takımlarında kıdem farkı yoktur. Tecrübeli olan yazılımcılar bildiklerini daha az tecrübeli yazılımcılarla paylaşırlar heem bilginin çoğaltılmasını sağlarlar ve hem de takım arkadaşlarıyla teknik açıdan aynı seviyeye gelirler. Yazılımcılara komut vererek iş yaptırmak yerine kendiliğinden bazı şeyleri öğrenerek doğru bir şekilde görevlerini yapmalarını sağlanmalıdır.

## Az Başlangıç Yatırımı

çok modern ve fazla araç gereç kullanılmasını beklemez. Başlangıç giderleri ne kadar az olursa, projenin iptal olursa kayıp aynı oranda daha az olur. Başlangıçta tüm takımın dar bir finanstman korsesini giymesi sağlanır ve süreç için daha önemli olan görevlere odaklanmaları sağlanır.

## Kazanmak İçin Oyna

XP takımları daima kazanmak için oynamaktadır. Amaç her zaman galibiyet almaktır. Her zaman gözlerinin önünde her zaman kazanılan sonuç vardır yani programı tamamlamak ve müşteriye teslim etmek en büyük hedefleridir. yazılımcı takıma sürekli sonundaki işi görmek için gerekli tüm imkanlarını sunmaktadır.

## Somut Denemeler

Verilen kararların sonuçlarını kontrol etmek için denemeler yapılır, çünkü alınan tüm kararlar her zaman doğru olmayabilir ve doğru sonucu vermeyebilir. Bir kontrol mekanizmasına ihtiyaç vardır. Bu da somut denemeler sayesinde nerede kaldığını tespit etmektedir. Bu somut denemeler yazılım sistemleri için de geçerli olan bir uygulamadır.

## Sorumluluk Üstlenmek

Sorumluluk birilerine görev olarak verilmemelidir aksine bireyler kendileri sorumluluklarını kendileri üstlenmelidirler. Eğer ki bir bireye ya da bir takıma yapılması

cok zor olan bir sürecin tüm sorumluluğu yüklenilirse, bu birey yada takım için tüm motivasyonunun düşmesinin ve kaybedme korkusunun pekmesmesini oldukça hızlandırır. Eğer bireyler yada takımlar kendilerine ait sorumluluklarını kendileri secebilselerse, hemide yaptıkları işte kendilerini iyi hisederler hemide yüksek motivasyonlarıyla üstlendikleri işi başarıyla tamamlamış olurlar.

#### Dogru Olcum

Proje gıdısının kontrol edilebilmesi için değık türde olan ölçümlerin yapılması gerekir. Bu ölçümler doğru yapılırsa projenin ilerleyişı de doğru şekilde olmuş olur. Yazılımcılar tarafından doğru yapılmayan ölçümler sürecin gıdısını olumsuz etkiler. [13]

#### XP TEKNİKLERİ

XP teknikleri yazılımcıların değık prensiplerini uygulama asamasında yardımcı olur.

Bunlar;

##### Ayakta Toplantı

Projede çalışanlar her gün 15 dakikayı aşmayan ve ayakta yapılan bu toplantılarda bir araya gelip proje hakkında birbirlerine bilgi verirler.

##### Kısa Aralıklarla Yeni Surum

XP süreçlerinde programa yeni dahil edilen ve değıkliğe uğrayan planlar yeni surumlar oluşturularak müşterinin beğenisine sunulur. Bu sayede müşteriler yeni surumu inceleyerek, ihtiyaçlarına uyup uymadığını kontrol edebilirler. Eğer yeni surum müşteriye uygun değilse değıkliğe uğrayabilir. Bu değıklikler sayesinde müşteri isteklerine yüksek oranda uyan bir programın oluşması sağlanır.

##### Geriye Bakış

Proje çalışanları düzenli olarak geriye bakarak, meydana gelen sorunları gözden geçirmelidirler. Budaki amaç gelecekte tekrarlanabilme ihtimalini en aza indirmektir. Geriye bakış 1- 6 aylık zaman dilimi için tüm projede çalışan yazılımcılar ya da seçilen bireyler tarafından yapılmalıdır. Geriye bakış toplantıları en az bir ay ile üç ay arasında yapılabilmektedir. [13]

## Mecaz

projelerde hazırlanan programın nasıl bir işlevsellik olacağını ekiptekilerin gözlerinde canlandırmasını sağlayabilecek mecazı bir isim, örneğin yada bazı resimler kullanılabilir. Bunlar proje çalışanlarının ortak bir sonuçta da buluşarak ne yapılması gerektiği hakkında bir fikir edinmelerini kolaylaştıran bir adımdır.

## Ortak Sorumluluk

projelerde yazılımcılar birçok ortak sorumluluk taşır. Buradaki her bir kodun parçasının herhangi bir yazılımcı tarafından gerçekleştirilen durumlarla değiştirilebileceği anlaşılmaktadır. Böylece yapılması gereken işler hiç aksamaz. Her yazılımcıya özel bir kod bölümü olmaz. Aynı her yazılımcı programın her bir bölümü üzerinde çalışma hakkına sahip olur. Bir yazılımcının işi gelememesi durumunda ise, başka bir yazılımcı kolaylıkla onun görevlerini üstlenebilir ve böylece de işler hiç aksamaz.

## Sürekli Entegrasyon

Projede yapılan yeni ilerlemeler sürekli olarak sisteme entegre edilmelidir. Sürekli entegrasyon işleminin yapılması sayesinde yapılan tüm değişikliklere her programcının sistem üzerinde kolaylıkla ulaşması sağlanır. Ayrıca sistem entegrasyonu için de gerekli olan zaman azalır çünkü oluşabilecek olan hatalar erkenden tespit edilir ve hemen ortadan kaldırılır.

## Kod Standartları

Yazılımcılar tarafından aynı kalitede olan kod yazılması yapabilmek için, kodu yazarken kullanılacak olan kuralların oluşturulması gerekir. Kod standartları projenin ilerlemesinde ve doğru bir şekilde tamamlanmasında oldukça önemlidir.

## Kalıcı Tempo

procelerinde yazılımcılar haftalık belli olan mesai saatlerini asamazlar. Gereğinden fazla çalıştırılmış olan çok yorulan bir yazılımcının iş hayatında verimliliği olması beklenmez. Yazılımcıların motivasyonu ve çalışma enerjilerinin olabildiğince yüksek olması için gününde sekiz saat fazla çalışmalarına izin verilmelidir. Çünkü bu enerjinin düşmesine sebep olur. Enerji düşüklüğü ise verimin düşmesine sebebiyet verir. Eğer fazladan mesai ihtiyacı varsa bu projenin doğru ve sorunsuz bir şekilde olmadığını gösterir.

## Test Etmek

Olusturulan projelerin kalitelerinin kontrolden gecmesi gerktir. Bu, yazılım sırasında olusturulan bazı testlerle yapılır. Yazılımcılar projeler için test hazırlarlar. Yapılan testlerde projelerin işlevleri kontrol edilir. Müşteri ihtiyaçlarını test etmek için akseptans denilen testler hazırlanır. Projenin entegrasyonunu test etmek için entegrasyon testleri hazırlanır ve yapılır.

## Sade Tasarım

Programcılar üstlenmiş oldukları görevleri en basit haliyle projeye dahil etmelidir. Bu işlem sayesinde programın basit bir yapıda kalmasını sağlar ve ileride değiştirilebilir ve genişletilebilir halde olmasını sağlamaktadır. Sade olan bir tasarım yazılım sisteminin de kompleks olan bir yapıda olmasının önüne geçer. Bunun yanı sıra basit olan tasarımlar daha kolayca daha hızlı projeye dahil edilebilir. Basit olanı anlamak ve anlamak kolaydır.

## Yeniden Yapılandırma

Tasarımda yapılan hatalar yazılım sisteminin daha erte tamir edilmeyecek halle dönüşmesine sebebiyet verir. Bu yüzden de bu hatalar hemen giderilmelidir. Bu yüzden yapılan yapılandırma işlemine de “refactoring” adı verilir. Hazırlanmış olan unit testleri ile yapılan bu değişikliklerin sonuçları kontrol edilir. Buradan bakıldığında unit testi olmayan bir sistem üzerinde yeniden yapılandırma işlemi pek fazla mümkün olmayabilir çünkü unit testleri projedeki yani etkilerin bulunmasına yardımcı olur ve unit testi yapılmadığı zaman bu yan etkileri görmek çok zor hale gelir. [13]

## XP ROLLERİ

Bir projede ekipte çalışanların sorumluluk alanlarının tanımlanması için bazı roller vardır. Her bir rolün de kendine ait sorumlulukları ve hakları vardır. Bu roller kalıcı değildir. Ekibe içinde farklı kısıllara farklı roller verilebilir ve daha sonra rollerin değiştirilmesi yapılabilir. Proje ihtiyaçları doğrultusunda yeni rollerin oluşması mümkün olabilir.

## Müşteri

Proje müşteri sayesinde var olur. Müşteri ihtiyaçlarına ,isteklerine cevap verebilen bir yazılım için ücret öder yani yatırımını yapar.

Projede neyin programlanacağına müşterinin ihtiyaçları karar verir. Müşteri isteklerini kullanıcı hikayeleri oluşturarak takıma iletir. Yazılımcılar müşteriye bu süreçte yardımcı olup isteklerine karşılık verirler. Kullanıcı hikayesinin oluşturulma sorumluluğu müşteriye atılır. Her bir kullanıcı hikayesi yazılım sistemindeki özellikleri açıklar. Yazılımcıların müşterinin isteklerini projeye dahil edebilmesi için kullanıcı hikayesini anlamaları gerekmektedir. İsteklerin projeye dahil edilme sürecinde bu çok önemlidir. Buna ek olarak da müşteri tarafından oluşturulmuş olan hikayelerin test edilmesi gerekir.

Müşteriler projenin çalışma alanı hakkında gerekli bilgiye sahip olur. Yazılımcılar müşteriye karşısına çıkan sorunları çözme amacıyla sorular sorarlar. Bu soruların cevaplarını müşteriye bulurlar.

Müşteri hangi kullanıcı hikayesinin projeye uygulanacağını seçmelidir. Bu konuda müşteriye kısıtlama uygulanmaz. Müşteri kendi istediğini seçer ve yazılımcılar projeye uygular.

Kullanıcı hikayelerinin projeye dahil edilip edilmediğini kontrol etmek amacı ile müşterinin hazırlamış olduğu akseptans testleri yazılımcı ya da testçi tarafından projeye uygulanır. Bu test kullanıcı hikayesinin doğru uygulanıp uygulanmadığını kontrol eder.[13]

#### Yazılımcı

Sistem analizleri, tasarımlar, testler ve proje uygulamaları yazılımcılar tarafından yapılıyor.

Müşterinin hazırladığı kullanıcı hikayelerinin uygulanma süresini yazılımcılar tahmin eder. Bu olay yazılımcıların projenin planlamasına dahil edildiğini gösterir. Bazı geleneksel olan projelerde bu sürenin tahminini teknik bilgisi olmayan yöneticilerin yaptığı görülür fakat bu süre bakımından doğruyu yansıtmaz ve tahminler gerçekleşmez.

Her yazılımcı unit test güdülü olur ve bir adet takım arkadaşıyla beraber çalışır. “Pair programming” olarak bilinir. İki yazılımcının bir arada kod yazması bilgi paylaşımı sayesinde verimliliği artırır. Ayrıca bu iletişimin ve birlikte çalışmanın yanı sıra uyumun artmasını sağlar. Test odaklı çalışmak projenin verimini ve projenin geliştirilmesini sağlar. İşlerin daha kolay ilerlediği görülür. XP kullanılan projelerde yazılımcılar kodların test odaklı yazarlar ve gerekli test sınıflarını oluştururlar böylece projenin geliştirilmesi daha kolay olur [13]

## Proje Menajeri

Musteriyle yazılımcıları bir araya getiren kısıdır. Birlikte çalışabilecekleri verimli ortamı oluşturur. Tek başına proje planlamasından sorumlu olmaz. Yazılımcılara görevlerini vermez yazılımcılar kendi aralarında görev dağılımı yaparlar. Toplantıları ayarlar ve karşılaşılan sorunların giderilmesini sağlar.

## Koc

Proje sürecini tanıyan ve nasıl bir yol izleneceğini bilen bir experdir. Koc projenin başında takımı oluşturmakla ve onlara rehberlik yapmakla görevli kısıdır. Projede sorun çıktığında müdahale eder. Yazılımcılar XP yöntemlerinin dışına çıkarsa onlara müdahale eder. Bazen projenin gerçekleştirilme sürecinde etkin rol oynar. Yazılımcılara testler konusunda da yardım eder.

## Testci

Musterinin oluşturduğu akseptans testlerini projeye uygulayan yazılımcıdır. Aynı zamanda unit testlerini ve entegrasyon testlerinin uygulanmasında takım arkadaşlarına yardımcı olur.

## Haklar ve Sorumluluklar

Projede çalışan yazılımcılar haklarını ve sorumluluklarını doğru bir şekilde bilmezlerse projede aksaklık ve yazılımcılarda verimlilik kaybı yaşanabilir. Aynı zamanda müşterinin de haklarını doğru bir şekilde bilmesi için doğru bir şekilde tamamlanmasında oldukça önemlidir. [13]

## Musteri Hakları

Musteri bütçe planlamasını doğru yapabilmek için toplam sürecin ne kadar süreceğini, neyin ne zaman ve nasıl olacağını, neyin olmayacağını, ve bunların sebeplerini bilme hakkına sahiptir.

Musteri projesi hakkındaki fikirleri değiştirme hakkına sahiptir. Eğer isterse projeye yeni fikirler dahil edebilir.

Musteri yazılımcılardan en verimli ve değerli projeyi isteme hakkına sahiptir.

Musteri projenin ilerleyişini görme ve test etme hakkına sahiptir.

## Yazılımcı Hakları

Yazılımcının takım arkadaşlarına soru sorma ve cevap alma hakkı vardır.

Yazılımcı kullanıcı hikayelerinin projeye dahil edilme süresinin tahminini yapma hakkına sahip olur. Ayrıca bu tahminleri değiştirme hakkı da vardır.

Yazılımcıya görev verilmez onun kendi sorumluluklarını alma hakkı vardır.

Yazılımcının her zaman en iyi kalitede olan ısı ortaya çıkama hakkı vardır.

Yazılımcı projedeki öncelikleri bilme ve uygulamaya geçirme hakkına sahiptir. [13]

## SURECİN İSLEYİŞİ

Öncelikle müşteri ihtiyacına uygun olan projenin kullanıcı hikayesini oluşturur. Hikayedeki işlerin önceliklerini belirler. Yazılımcılar da kullanıcı hikayesinin projeye aktarılma süresini tahmin ederler.

Müşteri yazılımcılarla birlikte projenin tekrarlama ve surumun planını hazır hale getirir. Her bir surum için birden çok tekrar gerekir ve müşteri bu surumu kullanabilir.

Müşteri ilk defa olan tekrarlama süreci için kullanıcı hikayesini uygun şekilde seçer.

Yazılımcılar tekrarlama işlemi için seçilen hikayeleri projeye dahil ederler ve bu yapılırken müşteriyle fikir alışverişinde bulunulur.

Tekrarların sonunda yazılımcılar müşterilere çalışan bir sistem sunar. Müşteriler bu sistemi değerlendirir ve geridönümde bulunur.

Proje süreci boyunca edinilen yeni tecrübeler sayesinde tekrarlamalar planlanır. Müşteriyeni ihtiyaçların projeye dahil edilmesini isterse yeni hikayeler oluşturur ve yazılımcılar da bunu projeye dahil ederler.

Yapılan ilk surumu müşteri kullanmaya başlar ve yeni surum planlandığında sonraki tekrarlama buna devam edilir. [13]

## XP PROJE SAFHALARI

- Kesif Safhası: Proje başlangıcındaki aktiviteler bulunur. Bu bölümde müşteri tarafından kullanıcı hikayeleri oluşturulur. Yazılımcılar araştırmalarını yapar.
- Planlama Safhası : Kesif safhasından sonra gelir. Bu bölümde müşteriler yazılımcıların da yardımı ile tekrar ve surum planını oluşturup kullanıcı hikayelerinin yazılımcılar tarafından projeye dahil edilme sürecini tahmin etmelerine yardımcı olur.



- Tekrarlama ve Surum Safhası: Kullanıcı hikayesinin projeye dahil edilmesi aşaması, tekrarlama ve surumde gerçekleşir. Tekrarlamada gerçekleşen projeye dahil etme aşamaları müşteri tarafından seçilir. Projenin durumunu kontrol etmek için müşteri testler uygular. Bu testleri yazılımcılar ya da testçiler uygular. Her bir tekrarlama aşamasından sonra müşteri çalışan bir proje sunulur. Müşterinin proje hakkındaki fikirleri alınır. Tekrar sona erdikten sonra tüm süreç gözden geçirilir. Önceden olan hatalar gözden geçirilir ve tekrarlanmaması sağlanır
- Bakım Safhası: Projenin genel bakımı yapılır. Proje bu safhada geliştirilir. Kullanıcılar için eğitimler hazırlanır ve onlara bilgi verilir. Sistem hataları bu safhada giderilir. Müşterinin isteklerine göre büyük surumun hazırlıklarına başlanır. Böyle bir durumda da safhalarda en başa dönülür ve projeye başlanır. [13]

## EXTREME PROGRAMMING KULLANAN FİRMALAR

- Menlo Innovations
- 8th Light
- thoughtworks.com
- Pivotal Labs[14]

## AGİLE

Agile yöntemi, yazılımı geliştirmede kullanılan bir proje yönetimi yaklaşımı olarak bilinir. Agile yöntemi proje ekiplerinin yazılım geliştirme sürecine yardımcı olur. Sprint kullanılır.

## TARİHİ

Agile ile ilgili fikirler 1970 yıllarında ortaya çıkmaya başlamıştır. Geleneksel olan proje yaklaşımlarının olduğu bu dönemde buna karşı tepki olması amacıyla ortaya çıkmıştır. Üzerine incelemeler ve çalışmalar yapılmıştır. Dr. William Royce'un yayınladığı makalede bundan bahsedilmiştir.

Dr. Royce projenin gelişiminin tamamlandıktan sonra bir sonraki aşamaya geçilebileceğini savunmuştur. Yazılımcıların projenin tüm ihtiyaçlarını karşılamaları gerekli olduğunu savunmuştur. Sonraki adımın projenin tasarımının yapılması olduğunu savunmuştur. Projenin aşamalarını tamamlayan gruplar arasındaki temasın az olduğunu

vurulamıştır. Agile'nin kurucuları proje geliştiricileri tarafından incelenmesinin yazılımın gelişmesinde yararlı etkisinin olduğuna inanıyorlardı.

## AGİLENİN TANIMI

Bir projenin belli bir aşaması için ayrılmış olan süreye Sprint denir. Süresi dolan sprintler projenin bittiğine işaret eder. Sprint, bir projenin belirli bir aşaması için ayrılan süredir. Sprintlerin süresi dolduğunda proje tamamlanmış sayılır. Ekiptekiler arasında projenin gelişimiyle ilgili bazı anlaşmazlıklar olsa da projenin belirli bir yerinden sonra daha fazla aşama katedilmez. Projede kalan bölümler ise zamanla geliştirilmeye devam edilir.

## AGİLE METODUNUN GENEL PRENSİPLERİ

- Müşteriyi olabildiğince mutlu edebilmek için yazılımı güncel tutmak önemlidir.
- Müşterinin piyasadaki diğer rakipleriyle rekabetinin iyi olması için projenin gereksinimlerine dikkat edilmelidir.
- Yapılan proje sık sık sunulmaya özen gösterilmelidir. Projenin teslimatı olabildiğince erken yapılmaya özen gösterilmelidir.
- Projenin her aşamasında ekipler, yöneticiler ve diğer kişiler birlikte çalışmalıdır.
- Projede çalışan kişilerin motivasyonu her zaman üst seviyede olmalıdır. Yazılımcılara ihtiyaçları olan uygun ortamları ve sağlanmalıdır. Her zaman arkalarında destekleri olduğunu bilmeleri sağlanmalıdır.
- Ekip üyeleri ile iletişim her zaman yüz yüze olmalıdır çünkü bu iletişim en doğru iletişim türüdür.
- Projenin çalışması ilerlemenin en öncelikli ölçümü olur.
- Agile süreci sürekli bir kalkınmayı hedefler ve yazılımcılar buna göre teşvik edilir.
- Projenin teknik açıdan kusursuz olması ve tasarımının iyi olması çevikliği artırır.
- Agile'de sadelik oldukça önemlidir.
- Ekipler kendilerini organize edebilir ve tasarım konusunda iyi sonuçlar çıkarırlar.
- Takımın seviyesi düzenli olarak kontrol edilmeli ve sürekli bir gelişim için yapılabilecekler gözden geçirilmelidir. [15]

## AGİLE YARARLARI

Agile yazılım uzmanlarının projelerinde yaşadıkları deneyimlerden ortaya çıkmıştır. Agile ile sınırlandırılmalar ortadan kalkmış olmuştur. Zorluk derecesi azalmıştır. Agile proje geliştirme konusunda oldukça iyi bir yöntemdir. Yazılımcılar agile'i yoğun olarak kullanmaktadırlar. Agile projenin hızlı bir şekilde teslim edilmesini hedeflemektedir. Yazılımcılar projeye odaklanır ve riskler azalır.

Agilede yinelenmeli planlama ile geri bildirim kullanılır. Bu projede müşterinin gereksinimlerini karşılamak için ürünün sürekli düzenlenmesi gerekir. Bir projede durum olmak ve değerlendirmek; süreç içerisinde değişen veya değişmesi gerekli olan ihtiyaçlara kolayca uyulanabilir. Projenin gelişmesinde olmak ve değerlendirmek büyük rol oynar. [15]

Agile şirketler için doğru olan projeleri oluşturmaya yardımcı olur. Agile projenin gelişiminde yazılımcılara ürünün güncel bir halde kalmasına yardımcı olur. Bu da projenin rakip firmalar ile ortak pazarda güncel olarak dikkat çekmesini ve olumlu donutler almasını sağlar. Pazarın seviyesi konunur ve bir çalışmaların geride kalıp güncelliğini yititmesini engeller. Agile metodu geliştiriciler için oldukça önemlidir. Agile bir çok acıdan eleştirilir fakat müşterinin mutlu bir süreç ve sonuç yaşayabilmesi adına doğru yöntem olarak görülür.

Proje tam ve eksiksiz olarak müşterinin istediği şekilde karar verilen sürede teslim edilmesi gerekmektedir. Proje sürecinde müşterinin istekleri karşılanmazsa bile gereken sürede teslimi gerçekleşir. Müşteri gereksinimleri sürekli olarak güncellenir ve projeye uygulanır. Bu sayede müşteri projenin gıdasatından memnun olur. Bu sürekli değişim müşteriye ve yazılımcılara ihtiyaç duyduklarından fazlasını verir. Agile metodu yazılım geliştirme açısından kazanç sağlayan bir çözümdür. [15]

Agilenin önceliğini müşteriye memnun etmek ve yazılımcıların işini kolay hale getirmektir. Olabildiğince erken teslim edilen projeler her zaman müşteriye ve yazılımcıyı büyük bir yükten kurtarır. Müşterinin talep ettiği işlerin parçalara bölünüp aşama aşama yapılması ve her birinin sonunda teslim edilmesiyle erken geri dönüş sağlanır ve bu da projenin gelişmesine katkıda bulunur. [16]

## AGİLENİN FAYDALARI

### Paydas Katılımı

Agile, müşteriyle yazılımcı ekibinin bir araya gelmesini ve anlaşmasını sağlar.

Projeye müşteriye de büyük ölçüde dahil etmek memnuniyeti yükseltirken isteklerin de doğru oranda karşılanması sağlanır. İşbirliği artmış yükler azalmış olur. Kafada oluşabilecek sorun işaretleri erken surede giderilir. Projenin teslimi parça parça ve sık sık yapılarak aradaki güven artar.

### Kullanıcı Odaklı

Agile genelde projenin özelliklerini tanımlarken kullanıcı hikayelerinden faydalanır. Projede her bir sprintin sonunda müşteriye bir adet beta sürümü teslim edilir. Müşteri bunu kullanır ve ekibe geri dönüş sağlar. Bu da eksiklerin tamamlanmasını ve müşterinin memnun olmasını sağlar.

### Erken ve Öngörülebilir Teslimat

Sprint sürelerini 1-4 hafta olarak belirleyerek hızlı gelişmesi ve ileriyi görme imkanını sağlar. Sık sık teslim etmek de buna yardımcı olur. Beta testleri verimlilik ve erken planlama sağlar. [16]

### Öngörülebilir Maliyetler ve Takvim

Her sprintin sabit bir süresi ve maliyeti vardır. Maliyet ekip tarafından ise göre öngöülür. Her Sprint sabit bir süreye sahip olduğu için, maliyet sabit zamanda işin miktarına göre ekip tarafından öngörülebilir. Müşteri açısından önceki sprintlere bakarak ihtiyaçlarının karşılanma oranına göre maliyet tahmin edilir.

### Değişime İzin Verir

Yazılımcılar projenin her aşamasını müşteriye sunar ve değişiklik için fikirlerini alır. Projeyi isteklere göre yeniler ve her bir tekrar toplantısında yeni planlamalar yapar. Değişiklikler için açık kapı bırakır.

### Şeffaflık

Agile, projede müşteri için yazılımı en güncel şekilde hazırlar. Tüm özelliklerini şeffaf bir şekilde sunar. Toplantılar sayesinde de gözden geçirilir ve daha iyiye getirme imkanları doğar. Müşteri projeyi şeffaf bir şekilde gördüğü için gıdışatını anlaması kolaylaşır.[16]

## Kalite Artısı

Yazılım ekiplerinin projeyi parçalara ayırıp yönetmesi kalitenin yükselmesini sağlar. Her bir ekip farklı bir alana odaklanınca projenin kalitesi büyük ölçüde artar. Projenin gelişmesi ve test evrelerinin gerçekleştirilmesi de kalitenin artmasını sağlar. Testler sayesinde kusurlar erkenden görülür ve bunlar giderilerek proje kusursuz hale getirilmeye çalışılır.

## İş Değerine Odaklanır

Yazılımcıların hepsi müşterinin fikirlerine ve isteklerine değer verir. Proje onlar için müşterinin gözünde olduğu kadar değerli olmalıdır. Bu sayede en güzel sonuca ulaşılabilir.

Sonuçta bakacak olunursa teknolojinin ve iş gücünün arttığı bu dönemlerde dijitalleşen dünyada rekabet de hızla artarken agile kaliteyi yükseltir ve kalite bu dönemlerde oldukça önemlidir. Agile müşteri ve yazılımcı ekibi arasındaki iletişim ve işbirliğinin artmasına imkan sağlar. Bu da projelerin en verimli halde üretilmesi demektir. Agile dünya üzerinde oldukça yaygın kullanılan bir yöntemdir. [16]

## AGİLE KULLANANLAR

- |             |                      |
|-------------|----------------------|
| • IBM       | • Oracle             |
| • Cisco     | • Lilly              |
| • Microsoft | • Nasdaq             |
| • AT&T      | • Adobe              |
| • Philips   | • Boeing             |
| • Siemens   | • Humana             |
| • Nokia     | • Alcatel-Lucent[17] |

## KARŞILAŞTIRMALAR

### SCRUM VS RUP

- RUP ve SCRUM metodolojileri arasındaki fark nedir?

RUP ve Scrum genel olarak Agilenin alt kolu olarak bilinir. Projenin faaliyetlerine tekrarlamalı olarak yaklaşıp. Bununla beraber RUP projenin belli tarihlerde yapılmasını bekler. Scrum ise tüm proje yerine parça halinde kullanır ve parçaların her birini tekrar sonrasında yaklaşık olarak 4 haftada bir sunulmasını ve yeniden tanımlanmasını sağlar. Buna ek yapacak olursak RUP projeyi 4 aşamaya ayırır. Tüm bu döngüsü boyunca eşzamanlı iş akışını takip eder ve bunun sebebi de projenin belli aşamalar sayesinde kalitesinin en üst seviyeye çıkacağına olan inançtır. Aksine Scrum ise tüm bu döngünün tek tekrarda olacağını savunur. [18]

Scrum, RUP'a göre daha fazla kullanılan bir yöntemdir. RUP uygulama açısından Scrum'a göre daha maliyetli ve zor bir yöntem. Ayrıca RUP'un kullandığı araçların çoğu Rational tarafından artık desteklenmiyor.

Scrum kullanan çok sayıda kişinin fikirlerine dayanarak, çoğu kişi Scrum'ın RUP'tan daha başarılı olduğunu söylenebilmektedir.[19]

## RUP VS AGİLE

RUP, bir tekrara dayalı bir yöntem olarak kabul edilebilir. Tekrarlar önerilen tüm değişikliklere ayak uydurma açısından iyi bir yöntemdir. Her tekrarın sonucunda proje ilerlemeler kateder. Her tekrar önceki tekrarlarla bağlantılı olur.

Agile ise standart olan adımları takip eden bir yöntemdir. Agilenin adımları tekrara dayalı ve hızlı olan adımlardır. Agilenin amacı müşteriye 2-4 hafta içinde çalışan bir proje sunmaktır. [20]

Agilenin yaklaşımlarına benzer şekilde RUP da tekrara dayalı bir yöntemdir. RUP agileye göre daha kurallı bir yöntem olarak bilinir.

RUP daha önceden tanımlanan belli aşamaları kullanır. Proje dört aşamaya ayrılmıştır. Her aşamada, proje ile ilgili faaliyetlere vurgu yapan tekrarlardan oluşur.

RUP agileye göre daha fazla rol tanımlar. Agile proje içinde daha az rol tanımlar. RUP'ta bir sürü tanımlanmış rol vardır. Proje ilerleyişi açısından roller önemlidir. RUP bu konuda daha iyi bir yol izlemektedir.

RUP'ta kullanıcı hikayesi alınır. Buna göre hareket edilir. Agile yaklaşımlarında çoğu ekip kullanıcı hikayeleri kullanıyor bile, zorunlu değildir.

RUP ust uste yapılan bir surectir. Buyuk projeler kucuk parcalar bolunmustur. Agile ise sprintlere boler. Her sprint icin 2-4 haftalik zaman araligina sahiptir. Sprint proje acısından onemlidir.

İki yontem arasında buyuk farklar vardır. RUP daha cok geleneksel proje yonetimini kullanırken agile guclu ekipler kullanır.[21]

## SCRUM VS AGİLE

Agile bir yazılım gelistirme yontemidir. Hızlı bir gelisme olanak saglar. Scrum, Agile'in bir alt kumesi olarak bilinir. Baska sekilde ifade edecek olursak hızlı ve dogru bir sekilde biten kucuk is bolumleri olusturarak agile yonteminin desteklenmesini saglayan bir yontemdir. Her gun hızlı yapılan toplantılar raporlanır ve ilerlemeler ve sorunlar gozden gecirilir. Karsılasılan sorunlar en kısa surede cozume kavusturulur. Projenin tamamlanması hızlanmış olur.

Agile projede asamalı bir sekilde ilerlemeyi savunan bir yontemdir. Projeyi asamalara boler ve bittikce teslim ederek sorunları ve ilerlemeleri gozden gecirir. Zamana dayalı bir surec olarak bilinir.

Scrum, agilenin ilkerlerini kullanarak projenin ihtiyaclarını karsılayan yazılımını gelistirmeye calisan ve genelde karmasıklığı azaltan bir yontemdir.

Agile bir cerceve ise scrum cercevenin bir uygulamasıdır.[22]

## SCRUM VS XP

Scrum agilenin bir alt kumesiyken XP cevık bir yaklasım olarak bilinir.

### Scrum

- Scrum, bir yonetim cercevesıdır. Karmasık bir yapısı bulunur. Urun gelistirmek icin projeler bulunur. Proje gelistirmek icin proje gelistirme surecidir
- Scrum projesinde, urun genellikle 2-4 haftalik sureclerde proje kademeli sekilde gerceklestilir
- Scrumda takımlar capraz halde kendi kendilerini organize edebilirler.

- Sprintlere karra verıldıktan sonra degıstırılmez. Degıstırmek ıcın toplantı beklenır.
- Scrum takımlarında degısıık roller bulunur.
- Ekıp, burndown veya burnup grafikler kullanır ve ılerleme boylece gosterılır.
- Scrum takımı müşterının beklentılarını karşılayacak bır sekılde projesını gelıstırır. Gorevlerini ona gore secer.

#### XP

- XP, en kalıtelı yonetım sureclerından bırır.
- XP takımları genelde 1-2 haftalık sureclerde calısır.
- Tekrarlar, projenin ıhtıyaclarına gore ve mevcutta bulunan duruma gore farklılastırılabilir.
- XP dört asaması vardır.
- XP yalnızca kodun gelısmıs olmasına dıkkat eder.
- Bu projede önemli olan projenin ısteklerini karşılamak olarak bılırır.[23]

Scrum ve Extreme Programming oldukça uyumlu ıkı metodolojidır. Bıbırlerini tamamlarlar.

Bu ıkı surec arasında farklar var;

1. Scrum takımları 2-4 haftalık surece dahil ıken XP takımları 1-2 haftalık tekrar suresinden olusmaktadır.
2. Scrum, planı yapıldıktan sonra sprintlerde degısııklık olmasına kesinlikle ızın vermezken XP sürekli yapılan bır degısııklıge acıktır. Yalnızca degısııklık calısmaya baslamadan önce yılır.
3. Scrum'da projenin bırıktırıp müşteriye sunulması savunulurkene sprintlerin düşük olan öncelikleri vardır. XP takımların yapılan projenin müşteri tarafından belirlenen önceliklere gore calısır.

#### Farklar

Scrumda projenin ilerlemesinde müşterinin ıstekleri önemsenır ve öncelikleri buna gore belirlemeye calısılır.



Scrum muhendislik uygulamaları önermez ama XP yapar. XP testlere çok önem verir.

## XP

1. XP, tekrarlamalar kullanır
2. XP 1-2 haftalık surec kullanır
3. XP rolleri Koc, Musteri, yazılımcılar ve Testci'dir.
4. XP zaman çizelgesinde değişiklik yapılmasına izin verir

## Scrum

1. Scrum sprint kullanır
2. Scrum 2-4 haftalık surec kullanır
3. Scrum zaman çizelgesinde değişiklik yapılmasına izin vermez
4. Scrum Rollerı Geliştirme ekibi, müşteri ve Scrum Master'dır.[24]

## Benzerlikler

Her ikisi de yinelemelidir.

Her ikisi de çok hızlı planlama yapar

Görünür görsel çalışma yöntemi varsır

Her ikisinin proje sonunda gönderilebilir kodu vardır

Her ikisinde capraz olan fonksiyonel takımları vardır

Her ikisinde sosyal çalışma ortamı vardır

Her ikisinde takım odaklı bir süreçtir [25]

## XP VS RUP

### Benzerlikler

- 1-) Projeyi baştan yapmak
- 2-) İki de nesneye dayalı prensibe sahiptir
- 3-) Değişim süreçlerin en önemli adımıdır
- 4-) Kullanım senaryoları ve oykuları benzerdir

- 5-) kritik olan riskler belirlenir
- 6-) müşterinin istekleri önceliklidir
- 7-) projeye testler uygulanır
- 8-) başlangıç fazı ve oyku oluşturma benzerdir
- 9-) düzenleme fazı ve oyun planlama aşamaları benzerdir
- 10-) ikisi de planlamada derine inmez

#### Farklar

- 1-) RUP büyük ölçekli projelerde kullanılır
- 2-) RUP maliyete önem verdiğinden dolayı değişikliklerin olmasını çok önemsemez fakat XP değişiklikleri destekler
- 3-) RUPun içinde ikili programlama yoktur
- 4-) RUP projelerinde daha fazla yazılımcı bulunur
- 5-) XP kendini daha fazla ve daha kısa sürede yeniler
- 6-) XP, RUP içindeki disiplinin hepsini içermez.
- 7-) RUP XPye göre daha fazla seçim yapılabilirliği yüksek bir süreç sunar
- 8-) XP'de yazılımcıların sorumlulukları daha çoktur
- 9-) XP'de herkes ortaklaşa çalışabilir ama RUPda herkes kendi çalışır
- 10-) RUP mimariye daha çok önem verir
- 11-) XP, dokümanların hep güncel olmasını beklemez ama RUP her daim güncel olmasını ister[26]

#### XP VS AGİLE

##### XP

- XP kabiliyetli ve kapsamlı kod sağlayabilen yöntemdir
- XP 1-2 haftalık süreçte çalışır
- Tekrarlar projenin drumuna göre değişebilir

- XP nin dort degeri vardır
- XP sadece yazılımcıların kodu geliştirmesine odaklanır görünüm onlar için onemlidir.
- Önce müşterinin istekleri yapılır

## AGİLE

- Proje Yönetiminin alternatıfıdır.[27]

Bana göre bu metodolojiler arasında yazılım sektörüne küçük, orta ve büyük çaplı firmalar olmak üzere uygulanabilirliği en mantıklı olan metodoloji Extreme Programming. Diğerlerine göre çok yönlü bir yapıya sahip olması ve kolay uygulanabilir olması ve uyum sağlayabilmesi seçimimin başlıca sebeplerindendir.

## KAYNAKÇA

- 1 Bin Yaprak, “Scrum Serisi 1: Scrum Nedir?”, erişim: 16 Nisan 2020, <https://binyaprak.com/yazilar/scrum-serisi-1-scrum-nedir>
- 2 Wikipedia, “Scrum”, erişim: 16 Nisan 2020, <https://tr.wikipedia.org/wiki/Scrum>
- 3 Medium, “SCRUM NEDİR?”, erişim: 16 Nisan 2020, <https://medium.com/@secilcor/scrum-nedi%CC%87r-6a4326951dd8>
- 4 GoktugTuran, “Sprint Burndown Chart Nedir?”, erişim: 16 Nisan 2020, <http://goktugturan.com/blog/sprint-burndown-chart-nedir/>
- 5 Wikipedia, ” Scrum'ın sınırları”, erişim: 16 Nisan 2020, [https://tr.wikipedia.org/wiki/Scrum#Scrum'%C4%B1n\\_s%C4%B1n%C4%B1rlar%C4%B1](https://tr.wikipedia.org/wiki/Scrum#Scrum'%C4%B1n_s%C4%B1n%C4%B1rlar%C4%B1)
- 6 MMO, ” Scrum kullananlar”, erişim: 16 Nisan 2020, [http://www1.mmo.org.tr/resimler/dosya\\_ekler/66b78a0d08fff91\\_ek.pdf](http://www1.mmo.org.tr/resimler/dosya_ekler/66b78a0d08fff91_ek.pdf)
- 7 Wikipedia, “Tarihsel Temelleri”, erişim: 16 Nisan 2020, [https://tr.wikipedia.org/wiki/Scrum#Tarihsel Temelleri](https://tr.wikipedia.org/wiki/Scrum#Tarihsel_Temelleri)
- 8 Wikipedia, “History”, erişim: 16 Nisan 2020, [https://en.wikipedia.org/wiki/Rational Unified Process#History](https://en.wikipedia.org/wiki/Rational_Unified_Process#History)
- 9 Selçuk Üniversitesi, “RUP(RATIONAL UNIFIED PROCESS)” erişim: 16 Nisan 2020, <http://acikerisimarsiv.selcuk.edu.tr:8080/xmlui/bitstream/handle/123456789/9103/199826.pdf?sequence=1&isAllowed=y>
- 10 Nedir, “RUP nedir?”, erişim: 16 Nisan 2020, <https://www.nedir.com/rup>
- 11 Wikipedia, “Six best practices”, erişim: 16 Nisan 2020, [https://en.wikipedia.org/wiki/Rational Unified Process#Six best practices](https://en.wikipedia.org/wiki/Rational_Unified_Process#Six_best_practices)
- 12 Hg Insights, “Companies Curently Using IBM Rational Unified Process”, erişim: 16 Nisan 2020, <https://discovery.hgdata.com/product/ibm-rational-unified-process>
- 13 KurumsalJava, “Extreme Programming Nedir”, erişim: 16 Nisan 2020, <http://www.kurumsaljava.com/2008/11/21/extreme-programming-nedir/>

- 14 Quora, “What companies curenly practice XP (extreme Programming)?”, erişim: 16 Nisan 2020, <https://www.quora.com/What-companies-curenly-practice-XP-eXtreme-Programming>
- 15 TopTalent, “Agile Nedir? Agile Metodu Nasıl Uygulanır?”, erişim: 16 Nisan 2020, <https://toptalent.co/agile-nedir-agile-metodu-nasil-uygulanir>
- 16 LinkedIn Serpil Bakal, “NİÇİN AGILE ?”, erişim: 17 Nisan 2020, <https://www.linkedin.com/pulse/ni%C3%A7in-agile-serpil-bakal-psm>
- 17 ArtPlusMarketing, “This Is Why Fortune 500 Companies Use An Agile Approach To Process Improvement”, erişim: 17 Nisan 2020, <https://artplusmarketing.com/this-is-why-fortune-500-companies-use-an-agile-approach-to-process-improvement-d4095d410a61>
- 18 Chiron, “What is the difference between RUP and SCRUM methodologies?”, erişim: 17 Nisan 2020, <http://www.chiron-solutions.com/chiron-professional-journal/2010/12/20/what-is-the-difference-between-rup-and-scrum-methodologies/>
- 19 Quora, “As an IT professional, which development methodology is more successful: RUP (Rational Unified Process) or Scrum?”, erişim: 17 Nisan 2020, <https://www.quora.com/As-an-IT-professional-which-development-methodology-is-more-successful-RUP-Rational-Unified-Process-or-Scrum>
- 20 BusinessAnalystTrainingOnline,” Agile Vs RUP”, erişim: 17 Nisan 2020, <http://www.businessanalysttrainingonline.com/index.php/scrummaster/item/255-agile-vs-rup>
- 21 Quora, “What are key differences between agile and rup methodologies?”, erişim: 17 Nisan 2020, <https://www.quora.com/What-are-key-differences-between-agile-and-rup-methodologies>
- 22 Quora, “What is the relation between Scrum and Agile?”, erişim: 17 Nisan 2020, <https://www.quora.com/What-is-the-relation-between-Scrum-and-Agile>
- 23 Quora, “What are the differences between SCRUM, XP and Agile in general?”, erişim: 17 Nisan 2020, <https://www.quora.com/What-are-the-differences-between-SCRUM-XP-and-Agile-in-general>

24 Quora, “What are the differences between Scrum and Extreme Programming?”, erişim: 18 Nisan 2020, <https://www.quora.com/What-are-the-differences-between-Scrum-and-Extreme-Programming>

25 Quora, “What are the similarities between XP and Scrum?”, erişim: 18 Nisan 2020, <https://www.quora.com/What-are-the-similarities-between-XP-and-Scrum>

26 CsharpNedir, “RUP ve XP Süreçlerinin Karşılaştırılması”, erişim: 18 Nisan 2020, <http://www.csharpnedir.com/articles/read/?id=458>

27 Quora, “What are the differences between SCRUM, XP and Agile in general?”, erişim: 18 Nisan 2020, <https://www.quora.com/What-are-the-differences-between-SCRUM-XP-and-Agile-in-general>