

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**E-TİCARET’TE YENİ NESİL
KARMA SİSTEM YAKLAŞIMI
VE BİR UYGULAMA MİMARİSİ**
Yüksek Lisans Tezi

Tezi Hazırlayan:
YILDIRAY YILDIRIM

İSTANBUL, 2018

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**E-TİCARET’TE YENİ NESİL
KARMA SİSTEM YAKLAŞIMI
VE BİR UYGULAMA MİMARİSİ**
Yüksek Lisans Tezi

Tezi Hazırlayan:
YILDIRAY YILDIRIM

Öğrenci No:
130820012

Danışman:
Dr. Öğr. Üyesi EGE KİPMAN

İSTANBUL, 2018

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “E-Ticaret’te Yeni Nesil Karma Sistem Yaklaşımı ve Bir Uygulama Mimarisi” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde tarafımdan yazıldığını, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmamın içinde kullanıldıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım.(24/05/2018)

YILDIRAY YILDIRIM



T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi 130820012 no'lu
.....GILDIRIM GILDIRIM.....'in 11/6/18 tarihinde yapılan tez savunma sınavı¹
sonucunda 45 dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle,
....KABUL.... kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

Tez Başlığı³ :E-TİCARET'TE YENİ NESİL KARMA SİSTEM
YAKLAŞIMI VE BİR UYGULAMA MİMARISI.....

Tez Sınav Jürisi

Öğretim Üyesi

İmza

Danışman

: Dr. Ege KIRMAN

Üye

: Prof. Dr. Bülent D. İbrahim

Üye

: Dr. Ediz SAKAL

¹Jüri üyeleri, söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tezsınava alır. Tez savunma sınav süresi en az 45, en çok 90 dakikadır. Jüri üyeleri, sınav öncesi yapılacak toplantıda, kendilerinin danışmanından bir üyeyi başkan seçer. Tezsınava, tez çalışmasının sunulması ve buna ilişkin soru-cevap bölümünden oluşur. Tezsınava, öğretim elemanları, lisansüstü öğrenciler ve alanın uzmanlarından oluşan danışmanların katılımına açık ortamlarda gerçekleştirilir. Belirli en geç toplantı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda, jüri en geç on beş gün içinde toplanarak adayın tezsavınmasını onaylar. (05 Ağustos 2017 tarihli 30145 sayılı Resmi Gazete'de Yayınlanan Değişiklik-Madde 29-3)

²Tezsınanın tamamlanmasından sonra jüri, tez hakkında salt çoğunlukla "kabul", "düzeltme" veya "ret" karar verir. Jüri başkanı, jüri üyelerince imzalanmış karar tutanağını, tezsınasını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve birinci fıkradaki süreleri jüri önünde savunur. Süresi içerisinde "düzeltme" savunmasını yapmayan öğrencinin enstitü ile ilişkisi kesilir. (Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği-Madde 29-4)

³İleride doğabilecek aksaklıkların engellenmesi için tezin başlığını yazılması gerekmektedir.

Adı Soyadı : YILDIRAY YILDIRIM
Danışmanı : Dr. Öğr. Üyesi EGE KİPMAN
Türü ve Tarihi : Yüksek Lisans, 2018
Alanı: : Mühendislik
Anahtar Kelimeler : B2B, B4B, SAAS, Çevik Yazılım Yaklaşımı, B4B Karma Sistemi SAAS Uygulaması

ÖZ

E-TİCARET’TE YENİ NESİL KARMA SİSTEM YAKLAŞIMI VE BİR UYGULAMA MİMARİSİ

Bu çalışmada B2B ve B4B yapıları incelenmiş özellikleri, kullanım amaçları ve farkları ortaya konmuştur. Türkiyede kullanılan B2B çeşitlerinin yapısı incelenmiş, B2B’den B4B’ye geçişin nedenleri tüm detaylarıyla irdelenmiştir. Bunun yanı sıra bulut bilişim ve bir bulut bilişim yapısı olan SAAS servis yapısı incelenmiş, klasik yapıdan maliyet, erişebilirlik, stabilizelik ve güvenlik açısından daha iyi olduğu tespit edilerek teze konu olan B4B Karma sistemi bu anlayış ile bir SAAS servisi olarak inşa edilmiştir. Uygulama geliştirilmeden evvel ilkin çevik yazılım metodolojisi incelenmiş, uygulama geliştirilirken çevik yazılım metodolojisinin bir alt basamağı olan Scrum yöntemi kullanılmıştır. Scrum ile birlikte yazılım projesinin zaman, insan kaynağı, maliyet bakımından klasik yöntemlere göre daha fazla fayda sağladığı ve daha esnek olduğu tespit edilmiştir. Nihayetinde bir B4B karma sistemi uygulamasının prototipi ortaya konmuştur.

Name and Surmane : YILDIRAY YILDIRIM
Supervisor : Dr. Öğr. Üyesi EGE KİPMAN
Degree and Date : Master, 2018
Major: : Engineering
Key Words : B2B, B4B, SAAS, Agile Software Development
Methodology, B4B Hybrid System SAAS Application

ABSTRACT

A NEW GENERATION OF MIXED SYSTEM APPROACH AND AN APPLICATION ARCHITECTURE IN E-COMMERCE

In this study, B2B and B4B structures have been analyzed, and their intended use and differences have been presented. The structures of B2B types used in Turkey have been examined, and the reasons of switching from B2B to B4B have been studied in details. In addition to this, the Cloud Computing and SAAS service structure, a cloud computing structure, have been analyzed, and ascertaining that these are better than conventional structure in terms of cost, accessibility, stabilization and security, a B4B hybrid system, which is the subject of this thesis, has been constructed. Prior to developing the software, firstly the agile software methodology has been examined and, while developing the software, the SCRUM method, a lower step to agile software methodology, has been used. Together with SCRUM, it has been ascertained that, compared to conventional method, this software project is more flexible and useful in terms of time consumed, workforce and cost. Eventually a prototype for B4B hybrid system application has been introduced.

İÇİNDEKİLER

Sayfa No.

ÖZ

ABSTRACT

TABLolar LİSTESİ İV

ŞEKİLLER LİSTESİ V

KISALTMALAR Vii

BİRİNCİ BÖLÜM

GİRİŞ VE AMAÇ

GİRİŞ VE AMAÇ 1

İKİNCİ BÖLÜM

B2B, B4B, SAAS YAPISI VE ÇEVİK METOT KAVRAMLARI

B2B, B4B, SAAS YAPISI VE ÇEVİK METOT KAVRAMLARI 4

2.1 B2B 4

2.1.1 B2B Sisteminin Hizmet Yapısı ve Özellikleri 5

2.1.1.1 B2B Hizmet Yapısı 5

2.1.1.2 Uçtan Uca B2B Çözümünün Temel Özellikleri 6

2.1.2 B2B Neden Önemli 7

2.2 B4B Yapısı 9

2.3. SAAS ve Bulut Bilişim 10

2.3.1 Bulut Bilişim Nedir? 10

2.3.2 Bulut Bilişim Servis Modelleri? 11

2.3.2.1 Servis Olarak Yazılım (Software as a Services, SaaS) 11

2.3.2.2 Servis Olarak Platform (Platform as a Service, PaaS) 12

2.3.2.3 Servis Olarak Altyapı (Infrastructure as a Service, IaaS) 12

2.3.3 Bulut Bilişim Üzerine Çözüm Türleri 12

2.3.3.1 Genel Bulut Çözümleri 12

2.3.3.2 Özel Bulut Çözümleri 13

2.3.3.3 Hibrit Bulut Çözümleri 13

2.3.4 Bulut Bilişim Avantajları 13

2.3.5 Bulut Bilişim Dezavantajları 14

2.3.6 SAAS Yapısal Özellikleri 15

2.3.7 SAAS Uygulamalarının Avantajları 16

2.3.8 SAAS Uygulamalarının Dezavantajları 18

2.4 Yazılım Geliştirme Metodolojileri ve Çevik Yaklaşım	20
2.4.1 Geleneksel Model (Şelale Modeli)	20
2.4.2 V Model	21
2.4.3 Prototipleme Modeli	22
2.4.4 Spiral Model.....	23
2.4.5 Artımlı Geliştirme Modeli	24
2.4.6 Evrimsel Geliştirme Modeli.....	24
2.4.7 Evrimsel Prototipleme Modeli	24
2.4.8 Çevik Yaklaşım.....	25
2.4.8.1 Çevik Yaklaşım Manifestosu	26
2.4.8.2 Çevik Yaklaşım Prensipleri.....	30
2.4.8.3 Çevik Yaklaşım Türleri	37
2.4.8.3.1 Uç Programlama - Ekstrem Programlama - EP (Extreme Programming)	37
2.4.8.3.2 Dinamik Sistem Geliştirme Metodu – DSGM (Dynamic System Development Model).....	37
2.4.8.3.3 Uyarlanabilir Yazılım Geliştirme - UYG (Adaptive Software Development)	38
2.4.8.4 Kristal (Crystal)	38
2.4.8.5 Scrum.....	39
2.4.8.5.1 Scrum Temel Kavramlar.....	40
2.4.8.5.2 Roller	41
2.4.8.5.3 Scrum Etkinlikleri / Zaman Planlaması / Toplantılar (Scrum Events / Meetings)	43
2.4.8.5.4 Scrum Kavramlar / Araçlar (Scrum Artifacts).....	45
2.4.8.5.5 Çevik Yaklaşım Avantajları ve Dezavantajları	47
2.4.8.5.6 Scruma Başlamak	48

ÜÇÜNCÜ BÖLÜM

UYGULAMAYI GELİŞTİRİRKEN KULLANILAN TEKNOLOJİLER	
UYGULAMAYI GELİŞTİRİRKEN KULLANILAN TEKNOLOJİLER	51

DÖRDÜNCÜ BÖLÜM

BULGULAR VE B4B KARMA SİSTEMİ

BULGULAR VE B4B KARMA SİSTEMİ	54
--	-----------

4.1 Çevik Metodolojilerin Projede Uygulanması.....	54
---	-----------

4.1.1 Neden Çevik Yaklaşım	55
----------------------------------	----

4.1.1.1 Bireyler ve Etkileşimler, Süreç ve Araçların Üstündedir	55
---	----

4.1.1.2 Çalışan Program, Kapsamlı Dokümantasyonun Üstündedir	55
4.1.1.3 Müşteri İle İşbirliği, Sözleşmeler ve Anlaşmaların Üstündedir	56
4.1.1.4 Değişikliklere Yanıt Vermek, Bir Planı Takip Etmekten Daha Üstündür	56
4.1.2 Çevik Yaklaşımın Projede Uygulanması	57
4.1.2.1 Scrum Metodolojisinin Projeye Uyarlanması	57
4.1.2.2 Scrum Kavramları	60
4.1.2.3 Ürün Sahibi.....	62
4.1.2.4 Scrum Master	62
4.1.2.5 Geliştirme Takımı.....	62
4.1.2.6 Ürün Listesinin Oluşturulması (Product Backlog).....	65
4.1.2.7 Sprint Listesi (Product Backlog)	67
4.1.2.8 Sprint Planlama ve Sprint.....	68
4.1.2.9 Günlük Scrum.....	76
4.1.2.10 Sprint Review	76
4.1.2.11 Sprint Retrospektifi	77
4.1.2.12 BurnDown Chat.....	77
4.2 B4B Karma Sistemi SAAS Uygulaması	78
4.2.1 B4B Karma Sistemi Mimari Yapısı	79
4.2.2 Veritabanı Mimarisi	81
4.2.3 Uygulama Servisi	85
4.2.4 UML Diagrams ve Usecase	90
4.2.5 B4B Karma Sistemine Genel Bakış	91
4.2.6 Kullanıcı Profilleri	91

BEŞİNCİ BÖLÜM

SONUÇ VE ÖNERİLER

SONUÇ VE ÖNERİLER	99
KAYNAKÇA.....	103
EK KAYNAKLAR	107
ÖZGEÇMİŞ	108

TABLÖLAR LİSTESİ

Sayfa No.

Tablo 2.1. Çevik Manifesto (orginal).	27
Tablo 2.2. Çevik Manifesto (Türkçe).....	28
Tablo 2.3. Çevik Prensipler (İngilizce).....	31
Tablo 2.4. Çevik Prensipler (Türkçe).....	32
Tablo 4.1. Ürün İş Listesi (Product Backlog).....	66
Tablo 4.2. Sprintlere Ayrılmış Ürün İş Listesi.....	68
Tablo 4.3. Sprint 4 İçin Ayrılmış Ürün İş Listesi.....	69
Tablo 4.4. Sprint İş Listesi “ <i>yapılması gereken görevler belirtilmiştir</i> ”.....	71
Tablo 4.5. Sprint Dökümanı.....	72
Tablo 4.6. Örnek Bir Kullanıcı Hikayesi.....	73

ŞEKİLLER LİSTESİ

Sayfa No.

Şekil 2.1. Many to One	4
Şekil 2.2. Many to Many	5
Şekil 2.3. Saas Uygulaması Yapı Haritası.	16
Şekil 2.4. Geleneksel Yöntem Modeli	21
Şekil 2.5. V Modeli.	22
Şekil 2.6. Protatip Modeli	22
Şekil 2.7. Spiral Model	23
Şekil 2.8. Scrum Temel Kavramlar	40
Şekil 2.9. Scrum Yaşam Döngüsü	46
Şekil 4.1. Scrum Temel Kavramları Listesi	57
Şekil 4.2. Scrum Metodunun Proje Uyarılama Şeması.	59
Şekil 4.3. ScrumDo Kullanıcı Kayıt Ekranı.	60
Şekil 4.4. ScrumDo Workspace Oluşturma	61
Şekil 4.5. ScrumDo Rollerin Oluşturulması	63
Şekil 4.6. ScrumDo Scrum Takımı..	63
Şekil 4.7. ScrumDo Scrum Takımı Detayları ve Rollerini	64
Şekil 4.8. Ekran Protipi Örneği (Giriş Sayfası).....	67
Şekil 4.9. Ekran Protipi Örneği (Ürün Listesi)	67
Şekil 4.10. ScrumDo Üzerinde Story Tanımlama.....	74
Şekil 4.11. ScrumDo Üzerinde Sprint Tanımlama.....	74
Şekil 4.12. ScrumDo Üzerinde Scrum Board Görüntüsü -1.....	75
Şekil 4.13. ScrumDo Üzerinde Scrum Board Görüntüsü -2.....	75
Şekil 4.14. ScrumDo Üzerinde Sprint Detayları.....	75
Şekil 4.15. B4B Karma Sistemi Seması Genel Görünüm.....	78
Şekil 4.16. B2B Sistemi Mimarisi.....	79

Şekil 4.17. B4B Karma Sistemi Mimarisi.....	81
Şekil 4.18. Veritabanı Tablo Listesi.....	82
Şekil 4.19. Veritabanı İlişkisel Veri Modeli.....	84
Şekil 4.20. Web Servis Mimarisi.....	85
Şekil 4.21. Models Altında Yer Alan Tablo Sınıfı Örneği.....	86
Şekil 4.22. Models ve Context Genel Görünüm.....	86
Şekil 4.23. B4Bcontext Sınıfı Tanımları.....	87
Şekil 4.24. Models ve Context Sınıf İlişki Yapısı Örneği.....	87
Şekil 4.25. Models ve Context Tablolarının Veritabanına İşlenmesi.....	88
Şekil 4.26. Repository Sınıfı Örneği.....	88
Şekil 4.27. Mapper Metotları Örneği ve View Sınıfları.....	89
Şekil 4.28. Inteface ve Service Sınıfları.....	89
Şekil 4.29. Ürün ve Stok Yönetimi İçin Örnek Bir Use Case.....	90
Şekil 4.30. Ürüne Ait Class Diagramı.....	91
Şekil 4.31. B2B Kullanıcı Profili İş Akışı.....	92
Şekil 4.32. B4B Kullanıcı Profili İş Akışı.....	93
Şekil 4.33. Satıcı Profili Abonelik Tanımı İş Akışı.....	97
Şekil 4.34. Alıcı Profili Kayıt ve Abonelik Talebi.....	98

KISALTMALAR

SAAS	: Software as a Service
PAAS	: Platform as a Service
B2B	: Business to Business
B4B	: Business for Business
SQL	: Structured Query Language
ERP	: Enterprise Resource Planning
CRM	: Customer Relationship Management
HTML	: Hypertext Markup Language
CSS	: Cascading Style Sheets
IaaS	: Infrastructure as a Service
WCF	: Windows Communication Foundation
REST	: Representational State Transfer
XML	: Extensible Markup Language
JSON	: JavaScript Object Notation
IIS	: Internet Information Service
WEB	: World Wide Web
API	: Application Programming Interface

BİRİNCİ BÖLÜM

GİRİŞ VE AMAÇ

Türkiye ve dünyada birçok marka müşterilerine daha iyi hizmet verebilmenin arayışları içerisindeyken bir yandan da yeni müşteriler edinme ve pazar payını arttırma hedeflerini ortaya koymaktadırlar. Bu bağlamda şimdiye dek yapılan bütün çalışmalar ve edinilen tecrübeler satış faaliyetlerinin olumlu yönde gelişimini etkileyen faktörler olmuştur.

Geçmişten günümüze süregelen klasik pazarlama faaliyetlerinin bireysel bir çabadan modern pazarlama ekosistemine doğru dönüşümde etkin rol oynayan bu tecrübelerle birlikte, teknoloji dünyasında yaşanan değişimler ve gelişimler de bu evrimin bir parçası olmuştur. Bunla berabere bireyin teknoloji okur yazarlığının artması da bu dönüşümün hızlanmasını sağlayan etkilere biridir.

1450'lerde matbaanın icadı sonrası basılı reklamlar ile başlayan pazarlama teknolojileri ilkel evrelerini geride bırakarak, bilgisayarların yaygınlaşması ile daha modern bir yapıya kavuşmuştur. Bu bağlamda pazarlama faaliyetlerinin teknoloji ile bir potada erimesi en temelde kurumsal kaynak ve iş zekası yönetim programları ERP ve CRM'lerin kendi içlerinde yer alan iş modellemeleriyle otomatize edilmiştir. Modern çağın getirdiği değişimler her aşamada pazarlama faaliyetlerini doğrudan değişime zorlamış ve gelişimi zorunlu kılmıştır.

Teknolojinin devrimi, internet çağı ve mobil devrim tüm dünyayı ve tüm endüstriyel faaliyetleri baştan yaratırken pazarlama faaliyetlerini de kökten değiştirmiştir. Ticaret; internet ve mobil dünyada faaliyetlerini sürdürme çabası içerisindeyken markaların değer yaratma isteğinin bu çabadan bağımsız olması beklenemez. CRM sistemlerinin hantal yapısının bağımlı ve antiesnek oluşu, bu değeri yatabilecek kararlılık ve başarı görünümünü ortaya koymaktan uzaktır.

İnternet çağı ile birlikte dijital pazarlama yöntemleri ve akıllı telefon teknolojileri, markaların faaliyetlerini bu doğrultuda geliştirmeleri gerekliliğini ortaya koymuştur. Bu sebeptir ki artık CRM'ler yerini B2B sistemlerine devretmeye başlamıştır. B2B sistemleri markaların tüm müşteri ilişkilerini ve pazarlama faaliyetlerini online ortama dönüştüren yeni bir yapılanma olarak ortaya çıkmıştır. Markalar müşterileri ile olan pazarlama faaliyetlerini bu platformlar üzerinden yürütmekle hem kendi iş akış süreçlerini hızlandırmış ve kayıt altına almış hem de müşterilerinin esneklik ve pratiklik kazanmasını sağlamıştır.

B2B'ler yapısal olarak firmaların kendi bünyesinde çalışan ve kendi müşterilerine hizmet veren yapılardır. Her marka kendi B2B sistemini kurabilir ve faaliyetlerini burada devam ettirebilir.

Günümüzde B2B faaliyetlerinin kısıtlamalarından kurtulma çabası, yeni pazarlama trendi olan B4B yapısının gelişmesine yol açmıştır. B2B'den farklı olarak müşterilerine bir ticaret alanı sunan platform olarak ortaya çıkan B4B'de amaç müşterilerine ihtiyaçları doğrultusunda satış yöntemi oluşturmaktır. Bu sayede markalar kendi müşterilerini yönetmek dışında yeni pazar olanakları ile satış ağını geliştirebilirler.

B2B'ler B4B'ye evrilirken firmalar bu değişim sürecine uyum sağlamak adına faaliyetlerini yeniden düzenleyebilirler. Ancak birçok firma B2B sistemlerini kendi bünyesinde tekil olarak müşteri faaliyetlerini göz önüne tutup düzenlemektedirler. Bu durum tezde ele alacağımız çözüm önerilerinin çıkış noktasını oluşturmaktadır. Markaların gerek B2B gerekse B4B iş akışlarını içeren tüm faaliyetlerini yürütebilecekleri ortak bir SAAS platformu oluşturup hizmete sunarak firmalara yeni pazarlama yöntemi sunmak ve yeni bir teknoloji servisi hizmeti vermek bu tezde temel amaç olacaktır.

Tüm bunları dikkate alarak tezde B2B, B4B ve SAAS terimlerinin detaylarını açıklamaya başlayıp sonrasında çevik metodoloji kavramlarını inceleyeceğiz. Kullandığımız teknolojileri sebepleri ile sebepleriyle birlikte detaylandırıp sonrasında oluşturacağımız B4B karma sisteminin detaylarını, mantığını, mimari yapısını ve çevik metodolojileriyle uyarlamasını aktarıp en sonunda da elde ettiğimiz sonuçları açıklayacağız.



İKİNCİ BÖLÜM

B2B, B4B, SAAS YAPISI VE ÇEVİK METOT KAVRAMLARI

Bu bölümünde tez kapsamında yer alan ve kullandığımız kavramların ne ifade ettiğini açıklamaya çalışacağız.

2.1 B2B

Açılımı “İşletmeden İşletmeye” olan B2B terimi temelde şirketler arası pazarlama, satış uygulamalarını kapsayan sürecin kısa tanımı olmakla beraber, bu sistem gelişen dijital pazarlama teknikleri, iş süreçlerdeki dönüşüm, ihtiyaçların değişmesi, elektronik ticaret ortamlarındaki teknolojik ilerleme ile kendini daha kompleks bir yapıya dönüştürebilmiş uygulama ve/veya uygulamalar bütünüdür.

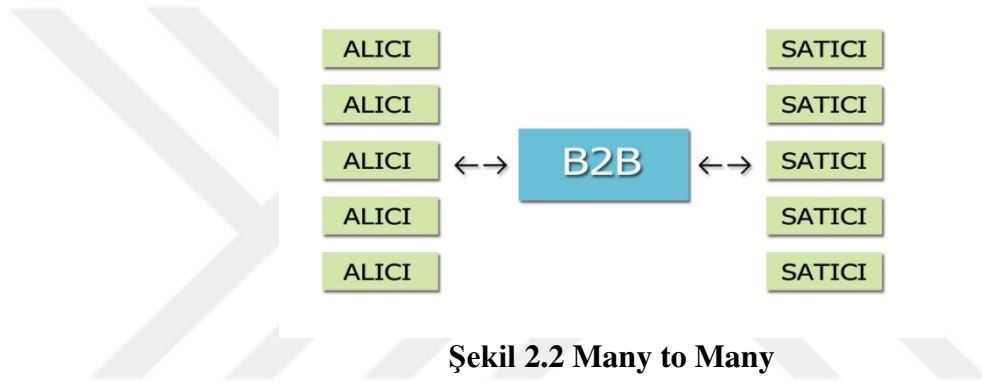
B2B sistemleri pazarlama temelinde 2 detayda ele alınır;

- **Many to One / One to Many:** Bu tarz satış kanalı, alıcı ve satıcılardan bir tarafın baskın ve yoğun olduğu yapılardır. Zaten hazır olan bir müşteri kitlesine hitap ettikleri için, arz ve talep bazında belli bir miktar kar oranına erkenden ulaşılabilir ancak karlılık oranı yüksek olmayabilir. Bu satış kanalı tipine satıcılar ve alıcılar farklılaşabileceği gibi, hitap edilen sektörde farklılık gösterebilir.



Şekil 2.1 Many to One

Many to many: Bu tarz satış kanalı, çok sayıda alıcı ve satıcının belli bir kurala bağlı olmaksızın, alıcı ve satıcı dengesinin tarafların birinde planlı yada plansız olarak baskın ve yoğun olduğu, etkileşimli, genişleyen bir yapıdır. Halihazırda yapılan uygulamalar ve elde edilen sonuçlar ile bu tarz satış kanalı sistemin alıcı yoğun bir satış kanalına evrildiği gözlemlenmektedir. Sistemin kurulması ve yoğunluğa ulaşması oldukça zahmetli ancak kritik yoğunluğa ulaştıktan sonra karlılık oranı yüksek sistemlerdir.



Şekil 2.2 Many to Many

2.1.1 B2B Sisteminin Hizmet Yapısı ve Özellikleri

Bu bölümde B2B sistemlerinin hizmet yapısı ve özelliklerini ele alacağız.

2.1.1.1 B2B Hizmet Yapısı

İnternet ve bilgi teknolojilerindeki ilerleme ile yüklenen trend haline gelen ve bir standart oturtmuş olan online pazarlama dünyasında B2B birçok şirketin kullanımı ile yerini sağlamlaştırmıştır.

En basit bir sipariş iletiminden kompleks satış sürecinin bütün basamaklarını tümüyle kapsayabilen B2B’ler şirketlere oldukça fayda sağlamaktadır.

Standart ticaret yönetiminde ürünün müşteriye tanıtımından, sipariş alınmasına, stok yönetiminden tedarik ve teslim sürecine tüm işler bir hiyerarşi ve manuel müdahalelerle gerçekleşir. Otomatize edilen süreç ve/veya süreçler, bunların birbirleri ile entegre çalışması durumları geniş bir organizasyon gerektiren maliyetli işlerdir. Ayrıca bu organizasyon şemasında hata oranı yüksek ve operasyonlar arası geçişlerde risk oldukça yüksektir. Örneğin: bir satış siparişi oluşturulurken stok bilgisinin satıcılar tarafından bilinmesi gerekir yoğun satış ortamlarında stok durumunu güncel tutmak başlı başına bir iştir. Burada yaşanacak bir belirsizlik durumu siparişin tedarik ve teslim sürecini etkileyebilir. Buda kurumsal bir şirket için fazla maliyet ve dahi müşteri kaybı riski taşımaktadır.

B2B sistemleri ürünlerin kategorilerinin sunulması, ürünlerin online yayınlanması , reklam ve pazarlama süreci, siparişin alınması, kampanya yönetimi, sipariş alınırken step by step stok kontrolü, depo yönetimi, sipariş teslim süreci, faturalama ve muhasebe gibi süreçlerin bir bütün içerisinde entegre çalışarak alıcı ve satıcılar için uçtan uca bir çözüm sunması ile firmaların ve kısmende olsa alıcıların yaşayabileceği risk ve ekstra maliyetleri azaltmaktadır. B2B'nin satıcı firmaya sunduğu entegre yapının büyüklüğü ve maliyeti, bu yapının bire bir benzerinin bahsedilen klasik bir ticaret sürecinde yürütmeye çalışacağı organizasyonun büyüklüğü ve maliyeti ile kıyaslanamaz düzeyde bir fark yaratacağı kesindir. Klasik ticaret yöntemi kar/maliyet marjı ile B2B sisteminin kar/maliyet marjı arasında oldukça fark vardır.

2.1.1.2 Uçtan Uca B2B Çözümünün Temel Özellikleri

- B2B uygulamalarında ürünlerin kategorize edilmiş olması önemlidir. Firmanın ürünlerinin kategorileri yoksa iyi bir kategori yapısı oluşturulmalı ve B2B sisteme bu kategori yapısı ile atılmalıdır. B2B sistemi bu kategorik yapıyı implement edebilir bir yapıda olmalıdır.

- B2B Sistemleri kurumsal kaynak yönetimi programları (ERP, CRM) ile entegre olabilmelidir. B2B uygulaması ile programlar arasında çift yönlü bir iletişim olmalı, B2B uygulaması bazı verileri bu programlardan alırken, sipariş stok ve benzeri verileri de iletebilir durumda olmalı.
- B2B uygulamasına ürüne ait tüm özellikler eklenebilir olmalı stok kartları, renk-beden-boyut-cins, fiyatlar, kampanyalar.
- B2B uygulamaları müşterilerine raporlama ve yönetim yetkileri verebilmelidir.
- B2B uygulamaları tek bir platform ve kullanım kolaylığı sağlamalıdır. Use-friendly yapıya sahip olmalıdır. Karmaşık ve kompleks süreçlerden kaçınılmalıdır. Hızlı ve seri işlem yapabilme esnekliğine sahip olmalıdır.
- B2B uygulaması mevcut teknik donanımları azami derecede tüketebilir olmalıdır. Az kaynak – çok iş prensibi ile çalışmalıdır.
- Her aşamada yapılan işlemlerin izine erişilebilir olmalı satış , stok , sipariş transactionlarının izlerinin takip edilebilir olması gerekir.

2.1.2 B2B Neden Önemli

İnternet ve e-ticaretin son dönemlerdeki yükselişi, kullanıcıların ve şirketlerin online alışverişe yönelmesi, teknolojik gelişmeler gibi birçok farklı etken klasik ticaret anlayışının da değişmesinde önemli bir rol oynuyor. B2B, internet üzerinden çok geniş kitlelere ulaşabilme ve farklı ürün seçeneklerine ulaşabilme gibi birçok avantajı da beraberinde getiriyor.

Artan pazar rekabetinde önem kazanan unsurlardan biride hız faktörü. Ürünün siparişi ile teslimatı arasında geçen süre ne kadar minimumunda tutulursa rekabet içerisinde bir adım öne çıkma şansı doğar. B2B sistemler satıcılara bu süreci minimum indirme şansı verir. Sipariş alınması, siparişin onaylanması, depoya bildirimi, ürün ve/veya ürünlerin hazırlanması, teslimat ve takibi sürecindeki kayıt izi bu akışın kontrolüne olanak verir, klasik yöntemlerde olabilecek birçok aksaklığın önüne geçilerek sürecin uçtan uca aksamadan hızlanmasını sağlar. Bu da firmaların iş hacminin artmasını ve kısa sürede sonuç almasını sağlar.

Daha önceden gerçekleştirdiğimiz bir projede uluslararası gözlük firmasının Türkiye temsilcisi olan müşterimiz için bir B2B uygulaması geliştirdik. Firma Türkiye genelinde birçok perakende gözlük satıcısına gözlük tedarik etmekteydi, firma yenilerde dahil olma üzere tüm ürün portföyünü müşterilerine satıcıları ile ulaştırmakta, pazarlama faaliyetlerini bu şekilde gerçekleştirip sipariş toplamaktaydı. Stok takibini telefonda merkezdeki personeller aracılığı ile takip ediyorlardı. Şirket kurumsal kaynak programı ERP kullanmaktaydı. Bu klasik yöntemde birbirlerine bağlı unsurlar sürecin akış hızını, bu unsurların hata payı da sürecin aksamasına neden olmaktadır, müşteri ziyareti esnasında stok durumu için merkeze ulaşamama, sipariş kaydının e-mail, sms ve telefonla eksik, yanlış girilmesi, satıcıların birbirlerinden kopuklukları gibi unsurlar gerek müşteri memnuniyeti, gerek zaman, gerekse maliyet açısından firmaya sıkıntılar yaratmaktaydı.

B2B uygulamasını geliştirip canlıya aldığımız da müşterilere bir account ile mobil ve internet ortamında ürünlere ulaşma detaylı bilgi alma stok durumunu güncel olarak görebilme imkanı sağladık. Bu da firmanın satıcı maliyetini düşürdü ve müşteriler ürün portföyüne online ulaşarak sipariş verebildiler, sipariş takibi yapıp tüm süreci izleyebildiler. Sistem firmadaki ERP'ye siparişleri otomatik yazdı ve muhasebeleşme öncesi onaya düşürdü. Bu Şekilde eksiksiz sipariş süreci işlemiş bulundu. Sipariş onaylanıp muhasebeleşince yine B2B üzerinden depo birimine iletildi ve siparişler müşterilere gönderildi. Müşteriler siparişin tüm aşamalarını kendilerine teslim edilene dek B2B üzerinden izleyebildiler. Bu Şekilde firma ciddi bir maliyetten kurtulduğu gibi aynı zamanda ticaret hacmini genişletmiş oldu.

Yukarıda uygulanmış canlı bir örnekte görüldüğü gibi B2B klasik ticaret süreçlerini otomatize ederek hata oranını en aza indirmiş, sağladığı hız ve azalttığı kaynak tüketimi ile firmaya fayda sağlamıştır

2.2 B4B Yapısı

B4B, açılımı Business for Business olan ve dünyada son dönemlerin yükselen değeri konumuna gelen, ülkemizde yeni yeni ilgi alanı oluşturmaya başlayan iş modeli. Klasik B2B modellerinin dışında sadece müşteri ve/veya müşterilerine ürün satmak, hizmet oluşturmak dışında, değer katma, fırsat yaratma amacıyla oluşturulan iş modelini kapsayan platformlardır.

B4B modelinde temel amaç müşterileriyle kurumsal bir ortaklık kurmak, birlikte çalışarak ortaya bir katma değer koymak, kurumların ihtiyaçlarını iyi analiz ederek başarılı bir sonuca doğru götürmektir.

Ürünün pazardaki konumu, olası yeni pazarlama fırsatları, genişletilebilir müşteri ağları, satış etkileşimi ile elde edilen bigdata ile revize edilmiş geri dönütler ve düzeltmeler ile müşterilerin yani kurumsal ortakların başarısı ve dolaylı olarak modelin başarısı hedeflenmektedir. En temelinde kazan – kazan temelli bir ortaklıktan söz edilebilir.

Özet olarak; yeni nesil iş modelinde şirketler artık şirketler ve tüketiciler için var. Müşterisi ile empati kurabilen ve ona değer/fayda sağlayan şirketler öne çıkıyor.

2.3. SAAS ve Bulut Bilişim

Bulut bilişim kavramı altında bir model olarak konuşlandırılan Saas (software as a service / hizmet olarak yazılım) temelde bir sorunun çözümü üzerine çıkmış bulut teknolojisi çözümüdür. SaaS, uzak sunucular üzerinde host edilmiş uygulamaların internet ağı üzerinden kullanımına imkan sağlayan yazılım dağıtım çözümüdür. IaaS (hizmet altyapısı) ve PaaS (hizmet platformu) modellerinin üzerinden konumlandırılmış, son kullanıcılara gerekli altyapı ve platformlar üzerinde uygulama kullanabilme imkanı sunar.

SaaS mantığını kavrayabilmek için genel çerçevede ve temel hatlarıyla bulut bilişimin ne olduğunu incelemek, mantığı ve faydaları üzerinde durmak önem arz etmektedir.

2.3.1 Bulut Bilişim Nedir?

**Bulut Bilişim, düşük yönetim çabası veya servis sağlayıcı etkileşimi ile, hızlı alınıp salıverilebilen ayarlanabilir bilişim kaynaklarının paylaşılr havuzuna, istendiğinde ve uygun bir şekilde ağ erişimi sağlayan bir modeldir.*

***TANIM: ABD ULUSAL STANDARTLAR VE TEKNOLOJİ ENSTİTÜSÜ**

Bulut bilişim, bir bilgisayar için gerekli olan tüm programların, kuruluma ihtiyaç duyulmaksızın internet üzerinden erişilerek kullanılması fikrine dayanmaktadır. Bulut bilişim, bilgisayarların yeteneklerini genişleten, kullanıcıların bir dizi yazılım ve servise internet üzerinden erişimlerine olanak tanıyan bir teknolojidir [Rayport and Heyward, 2009]

Bulut bilişim, bilgi teknolojileri servisleri için, internet tabanlı yeni bir dağıtım ve destek modelidir. Bu model, uzak cihazlara ve sunucu hizmetlerine internet üzerinden kolaylıkla erişim sağlama fikrinin bir ürünüdür. Tipik bulut bilişim, bir sunucu üzerinde tutulan veri ve yazılımlara web servisi ya da web tarayıcı benzeri yazılımlarla erişebilmeyi sağlar [Gruman and Knorr, 2008].

Web'in ikinci büyük dalgası olarak adlandırılan ve gelecekte iş dünyasına yön verecek en önemli kavramlardan biri olan bulut bilişim, internet üzerinden bir servis olarak sunulan bilgi teknolojileri kaynaklarının kullanımı ve geliştirmedir. Bulut bilişim, gerçek zamanlı, internet tabanlı teknoloji servis ve kaynaklarını ifade eden bir kavramdır [Turhan, 2009]

Klasik BT yapısında clientlerin kullandığı bilgisayarlar, uygulamalar, güvenlik sağlayıcıları ve tüm bu teknolojilerin üzerinde koştuğu sunucular, diskler ve bunlar için gerekli ups, iklimlendirme ve koruma yöntemlerini birleştiren maliyetli bir yapıyı içerir. Buna karşın bulut bilişim ortak bir data center üzerinde maliyetleri minimize edilmiş (maliyeti ortak kullanıcılar arasında bölüşürülen) ortam kullanımı için oluşturulmuş ortak servis sağlayıcılarını içeren bir kavramdır.

2.3.2 Bulut Bilişim Servis Modelleri?

2.3.2.1 Servis Olarak Yazılım (Software as a Services, SaaS)

Giriş olarak başlangıçta değinilen SaaS herhangi bir kurulum ve kaynak tüketimi olmaksızın, internet aracılığı ile bulut ortamda koşturulan uygulamalar bütünüdür.

2.3.2.2 Servis Olarak Platform (Platform as a Service, PaaS)

Servis sağlayıcı, kullananların kendi uygulamalarını geliştirebilecekleri, çalıştırıp test edebileceği bir ortam sunmakla beraber kullanabilecekleri servisler ve teknolojik altyapıyı da sağlar. Kullanıcının bu platformu oluşturan bileşenler üzerinde bir tasarrufu olmamaktadır. yalnızca kendi uygulaması ve bu uygulama için gerekli servis sağlayıcılarını içerir.

2.3.2.3 Servis Olarak Altyapı (Infrastructure as a Service, IaaS)

Kullanıcının ihtiyaçlarını bir bulut çözümü olarak sunan, işlemci, disk, ram gibi altyapı maliyetlerini aza indirgeyen, bu ihtiyaçların miktarını ve yapılandırmasını kendisine sunan bir çözümdür. İşletim sistemi dışında tüm yönetim tam bir kontrolle kullanıcıya bırakılmıştır. Yedekleme, güvenlik ve barındırma hizmeti gibi yükler kullanıcılardan alınmıştır.

2.3.3 Bulut Bilişim Üzerine Çözüm Türleri

Bulut Bilişim kavramı çözüm türleri bakımından bazı farklılıklar barındırmaktadır. Bu farklılıklar üzerine kısa bir değerlendirme yapacağız.

2.3.3.1 Genel Bulut Çözümleri

Bilgi işlem faaliyetlerini içeren tüm operasyonları ve bu operasyonların altyapısı için gerekli olan tüm teknik ve çözümlerin, 3. parti firmalar tarafından kurulmuş bulut kaynaklarla giderilmesi , buradaki kaynakların tamamının ve/veya bir kısmının kiralama yöntemiyle elde edilmesi sürecidir. Bu çözüm türünde firmalar tüm bilgi işlem yapısını 3. parti firmaların bulut sistemleri üzerine taşırlar, operasyonel olarak insan kaynağı desteği de bu hizmete mahsustur. Yüksek güvenlik gerektirmeyen düşük maliyet operasyonlara ihtiyaç duyan firmalar için idealdir. Ödeme türü genellikle belirlenen kiralama bedeli veya kullandıkça artan kira yöntemi ile çözümlenir. Salesforce ve amazon en uygun örneklerdir.

2.3.3.2 Özel Bulut Çözümleri

Firmaların bilgi işlem operasyonlarının şirket içerisinde kurulmuş bulut ortamlardan giderilmesi ile oluşturulmuş çözüm süreçlerini kapsar. Çok büyük şirketler ve yüksek güvenlik gerektirilen işlerde kullanılan yöntemdir. Dışarıya kapalı şirket bünyesine hizmet veren yapılardır. Genel bulut yöntemi kadar tasarruf sağlanmasa da grup şirketlerini içeren büyük kurumsal yapılarda veya yüksek güvenlik işlemi gerektiren büyük yapılarda önemli ölçüde tasarruf sağlamış olur.

2.3.3.3 Hibrit Bulut Çözümleri

Hibrit bulut çözümleri yapısı, genel ve özel bulut çözümlerinin karma bir şekilde kullanımı esasına dayanan süreçleri içerir. Şirketlerin (veya grup şirketlerinin) yüksek güvenlik gerektiren operasyon ve servislerini yönetmek için kurdukları özel bulut yapısı ile güvenlik gerektirmeyen basic yapıdaki operasyon ve servislerin yönetildiği genel bulut çözümlerinin bir karması olarak ortaya çıkar, sistemler servisler aracılığı ve uzak bağlantı izinleri ile birbirleri ile konuşabilirler. Bu yapıda hem güvenlik hem de maliyet optimizasyonu sağlanabilir.

2.3.4 Bulut Bilişim Avantajları

- **Düşük donanım maliyeti:** Bilgi işlem için kullanılan kaynakların temel maliyetleri bulut sistemi kullanan paydaşlara bölünmüştür.
- **Yüksek Depolama Kapasitesi:** Bulut sistemleri kullanıcılarını yüksek miktarda depolama alanına uygun fiyata sunabilirler.
- **Veri Güvenliği:** Bulut sistemlerde raid ve yedekleme yöntemleri ile veri güvenliği yüksek düzeyde sağlanır.
- **İşletim Sistemleri Arasında Gelişmiş Uyum:** Bulut bilişimde veriler uzak serverlarda tutulur, herhangi işletim sistemi ile dosyalara sorunsuz bir şekilde ulaşabilirsiniz.

- **Çoklu Erişim:** Dosya, uygulama, veritabanı gibi ortak alanlara birden fazla kullanıcının aynı anda erişimi söz konusudur. Her kullanıcı için ayrı bir yapıya ihtiyaç yoktur.

2.3.5 Bulut Bilişim Dezavantajları

- **Sürekli internet erişimi:** Gerekli verilere erişim için sabit internet bağlantısı gerekir.
- **İnternet Kalitesi Gereksinimi:** İnternet bağlantısında yaşanan hız, kesilme gibi problem durumlarında çalışmalar aksayabilir. Ayrıca upload ve download hızınız dosyaları ekleme ve alma durumlarında aksaklığa neden olmaması için geniş bant bağlantısına ihtiyacınız olacaktır.
- **Sunucu durumu:** Uygulamanın konuştuğu server eğer yoğunsa yada yavaşsa erişimlerinizde ciddi problemlere yol açabilir.
- **Güvenlik açıkları:** Genelde büyük bulut servislerine düzenlenen toplu ataklar veri kaybına neden olabilir.
- **Sistem Güncellemeleri:** Kullandığınız bulut sistemi üzerinde yapılacak güncellemelerin kullandığınız uygulamaları bozma riski vardır. Genelde güncelleme durumu sizin kontrolünüzde değildir.

Saas modeli bulut bilişim kavramı altında yer alır ve bulut bilişim özelliklerinin, avantaj ve dezavantajlarının çoğu saas modeli içinde geçerlidir.

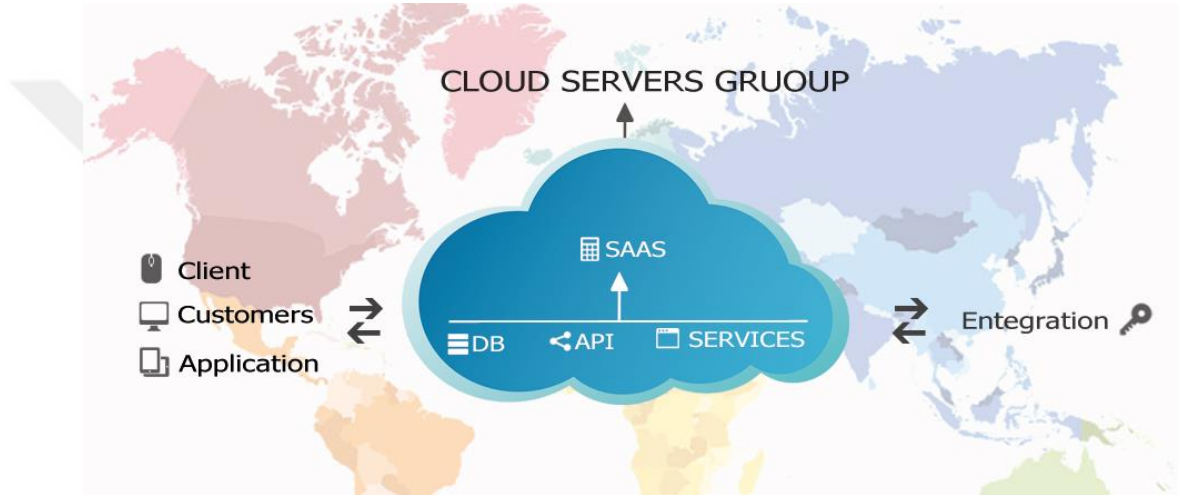
Ortak bir hizmet olarak kullanılan yazılım uygulaması (Saas uygulaması), hizmetlerin son kullanıcıya servis olarak sunulmasını sağlayan teknolojidir. Bulut sunucularda barındırılan bu uygulamayı şirketler kendi ihtiyaçları doğrultusunda kullanabilirler, bu sebeple paket programlara göre daha az maliyetle bir hizmet edinimine olanak sağlar. Şirketlerin kendi ihtiyaçlarını analiz etmesi ve bu doğrultuda saas uygulamasından ihtiyaçları doğrultusunda yararlanması saas uygulamalarının önemli bir özelliğidir. Şirketlerin ileride doğabilecek yeni ihtiyaçları için uygulamanın başka özelliklerini edinebilme imkanı vardır. Nitekim şirketler ve şirket organizasyonları yaşayan varlıklardır değişen koşullar ile operasyonel olarak yeni ihtiyaçların doğması ve geçmişte gereklilik arz eden bazı operasyonlarında kullanılmaması durumu uygulama içerisinde kullanılan modül yada akışların değişimini gerektirir. Bu açıdan da şirket için önemli olan kullanmadığı modülü bırakmak yeni ihtiyacı olan organizasyonu karşılayacak modülü elde etmektir. Bu sayede şirketler ihtiyaçlarını minimize ederek ihtiyacı kadarını satın alıp düşük maliyetlerle sistemi kullanma imkanı bulur. Ek özellikleri gerektiği zaman edinebilirler.

2.3.6 SAAS Yapısal Özellikleri

Saas uygulamaları ortak kullanıma açılmış servislerdir. Bir account ile erişime açıktır. Tek merkezli uygulama çok kullanıcıli bir mimariyle yapılandırılmıştır.

Saas servis sağlayıcısı uygulama, kendisini merkezi olarak tek bir yapı halinde sunar ancak birden fazla servis, veritabanı, api ve uygulamanın kendi bünyesindeki mimarinin yapısal olarak barındırdığı alt servisleri bulut sunucularda tutar.

Saas uygulamaları kendilerini dış kaynaklara apiler aracılığı ile açarlar, başka sistemlerin saas uygulamasına bağlanma gerekliliği varsa, açılan veya açılacak apiler bu durum için gerekli veri akışını sağlayabilirler. İletim ve veri alımın da çeşitli güvenlik mekanizmaları kullanılabilir. Account ile erişim, ip bazlı erişim kısıtlama, api güvenlik yöntemleri , ssl sertifikaları gibi güvenlik yöntemleri tercih edilebilir.



Şekil 2.3 Saas Uygulaması Yapı Haritası

2.3.7 SAAS Uygulamalarının Avantajları

- **Erişebilirlik:** Saas uygulaması kolay erişilebilir bir yapıya sahiptir. Bulut tabanlı olması platform bağımsız erişimide sağlamaktadır. İnternete bağlı herhangi bir cihazdan (mobil, pc, tablet vs) tarayıcı ile uygulamaya erişilebilir.

- **Maliyet:** Birçok paket yazılım programına şirketlerin sahip olmak için yaptığı ön yatırım oldukça yüksektir. Bununla beraber o uygulamanın bileşenlerinin barındırılacağı altyapı hizmetlerinin edinimi sonrasında bakım ve onarım giderleri, hizmetler için yetişmiş insan kaynağı düşünüldüğünde ve bu yatırımların şirketlerin büyüklükleri ile orantılı olarak artacağı göz önüne alındığında, teknoloji edinimi için yapılan yatırım devasa boyutları bulabilir. Saas uygulaması bu noktada çoklu kullanıma müsait mimari yapısıyla tüm maliyeti birden fazla kullanıcıya yayarak şirketlerin maliyetini minimize etmektedir. Şirketler düşük lisanslama ücretleri ve sonrasında kiralama yöntemiyle edindikleri saas uygulamalarla;

- Yazılım geliştirme yada tümüyle edinme maliyetini ortadan kaldırır.
- Barındırma gereksinimleri ve altyapı hizmetlerini ve bununla beraber buradaki alt uygulamaların lisans maliyetlerini ortadan kaldırır.
- Bakım-onarım, güncelleme ve bunun için gerekli insan kaynağı giderinden tasarruf ettirir.

- **Stabillik:** Saas uygulamaları bulut sunucularda tutulur, dağıtık mimari ile birden fazla bulut sunucu farklı lokasyonlarda işe koşulur. Dolayısı ile kesinti, talep dağıtımındaki yük fazlalıkları nedeniyle yaşanan kararsızlıklar ve hizmet verememe durumunda başka sunucu ve veri merkezlerini işe koşar. Kullanıcı kendi lokasyonunda eksiksiz hizmet almaya devam eder. Yedek alınan verilerle, herhangi fiziki, doğal yada siber saldırı sebebiyle çöken sistemler kısa sürede yeniden ayağa kaldırılabilir.

- **Güncelleme ve Bakım Onarım:** Servislerde ve servislerin bulundurulduğu sistemlerdeki güncelleştirme, versiyon yenileme ve yeni teknoloji implementasyonu kısa süreli, hızlı ve müşteri açısından ekstra maliyete neden olmadan gerçekleştirilir. Sistemsel yada yazılımsal bir hata durumunda hızlı dönüt ile çözüm üretilir ortak uygulama olduğu için bu düzenleme veya çözüm tüm müşterilere eş zamanlı uygulanabilir.
- **Güvenlik:** Firmaların kendi uygulamaları ve kullandıkları sistemlerin güvenliğini sağlamak için etkinleştirmesi gereken yazılım ve donanım çözümlerinin maliyeti oldukça yüksektir. Saas uygulamasında daha öncede bahsedildiği gibi bu maliyet paydaşlar arasında bölünmektedir. Büyük bir saas uygulamasının güvenliğini sağlayacak bir yapının maliyeti tekil firmaların kendi bünyelerinde sağlaması olanaklı görünmemektedir. Ayrıca yedekleme ve raid mimarileri ile verilerin kaybolması durumu oldukça zorlaştırılmıştır. Bu durum firmaların saas uygulamasını tercih etmesinde önemli bir etken olmuştur.

2.3.8 SAAS Uygulamalarının Dezavantajları

- **Bağımlılık:** Sisteme erişmek için internetin olması gerekir. Ayrıca upload ve dowland hızının yüksek olması sisteme iletişimde kolaylık sağlayacaktır.
- **Esneklik:** Firmaların yapısı, iş akışı ve operasyon yöntemleri zamanla değişmektedir. Buda yeni ihtiyaçların doğması ve kullanılan uygulamalarda değişiklik gerekliliği ortaya çıkarır. Ancak saas uygulamaları değişen firma ihtiyaçlarına aynı hızda cevap verebilecek çeviklik ve esnekliğe sahip olmayabilir. Ayrıca sistemi birden fazla kullanıcı kullandığı için firmaların bireysel değişiklik ihtiyacı karşılanamayabilir. Daha sabit akışlara sahip, uçtan uca aşamaları belirli süreçleri kapsayan saas uygulamaları kullanmak bu açıdan önemlidir.

- ***İhtiyaç Durumu:*** Sektörel bazda firmaların çalışmaları analiz edildiğinde her firmadaki ihtiyaçlar farklılık göstermektedir. Uygulamalarda modüller bir işi ortak, temel, sektör ve ülke yönetmeliklerinin bir bileşimi olan stratejiler üzerine kurar. Hizmet veren saas servisleri bu ihtiyaçların tamamını kapsar olmayabilir. Dolayısı ile saas hizmeti edinimi sırasında ihtiyacın yüzde yüzünü kapsayan bir uygulama bulunamayabilir.



2.4 Yazılım Geliştirme Metodolojileri ve Çevik Yaklaşım

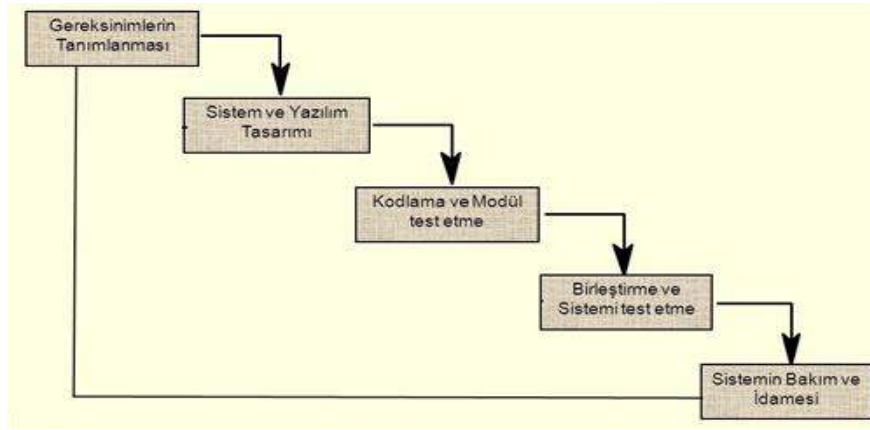
Bilgi sistem yazımlarında kullanılan metodolojiler, yazılım yaşam döngüsü boyunca kullanılacak süreç, araç, belgelendirme gibi yöntemler bütününe içeren bir disiplindir. [Saridoğan 2011]

Yazılım gelişiminin evrimleriyle bağlantılı olarak gelişim gösteren metodolojiler, günümüzde birçok yönetime bölünmüştürler. Hala gelişim ve değişimini sürdüren metodolojik yöntemler için uluslararası kuruluşlar belirli standartlaşmalar getirerek ortak bir noktaya doğru ilerleme çabası içerisindeyler.

Veri akışına, yapısına, sürece, müşteriye ve geliştiriciye bağlı olarak değişen metodolojileri yakından inceleyelim. [Saridoğan 2011]

2.4.1 Geleneksel Model (Şelale Modeli)

Bu model de kullanıcılar için gerekli tüm ihtiyaçlar önceden çözümlenir, sistem tasarlanır, uygulama geliştirilir , sistem uyarlanır test edilir, sonrasında ise bakım süreci ile takip edilir. Her aşama birbirini izleyen ardışık bir yöntemdir. Her aşama kendi içinde gerçekleştirilir ve her bir aşamadan sadece bir sonraki aşamaya geçilir. Eğer sorun varsa bir önceki aşamaya geçilir ve problem durumu düzenlenir. Bir sonraki aşamaya geçmek için bir önceki aşamanın eksiksiz tamamlanması zorunludur.



Şekil 2.4 Geleneksel Yöntem Modeli

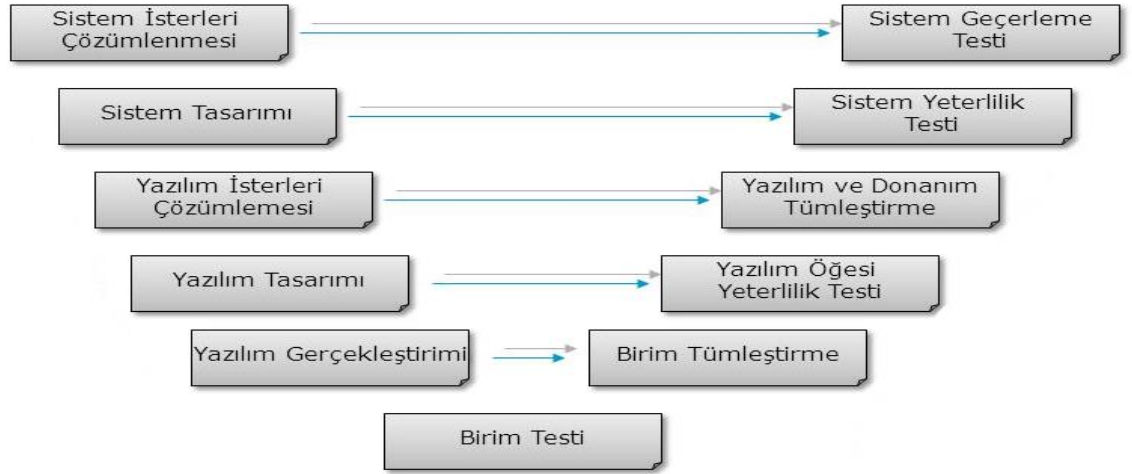
Kaynak : Yrd.Doç.Dr Yılmaz , Güray (2007) yazılım mühendisliği ve yazılım yaşam döngüsü. (05.02.2018) tarihinde <http://slideplayer.biz.tr/slide/2806563/> adresinden alınmıştır.

2.4.2 V Model

V modeli, geliştirme sürecine sistem kavramlarıyla yaklaşarak klasik modeldeki test işlemlerinin ne zaman yapılacağını ön plana çıkarır. Model, testler sırasında bulunan hataların düzeltilmesi için hangi düzeye dönülmesi gerektiğini işaret eder. [Saridoğan 2011]

V model, şelale modelinin kontrol safhaları daha organize edilmiş bir hali olarak görülebilir. [Nizam 2014]

"V" Modeli

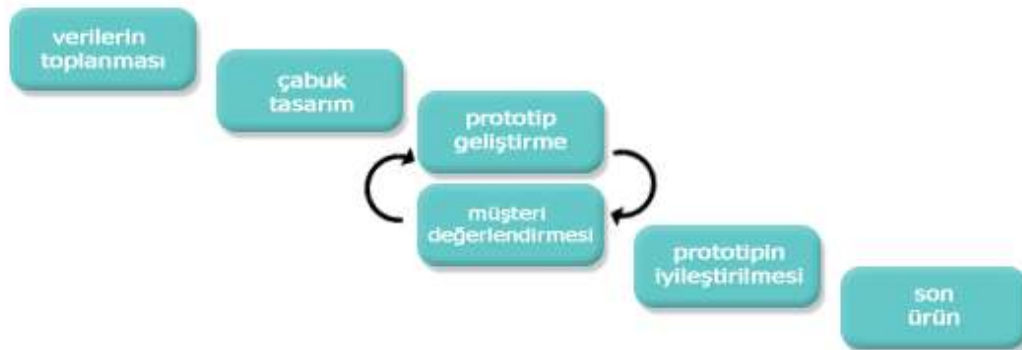


Şekil 2.5 V Modeli

Kaynak : Yrd.Doç.Dr Yılmaz , Güray (2007) yazılım mühendisliği ve yazılım yaşam döngüsü. (05.02.2018) tarihinde <http://slideplayer.biz.tr/slide/10433453/> adresinden referans alınmıştır.

2.4.3 Prototipleme Modeli

Çeşitli belirsizliklerin olduğu projelerde , istenilen yazılımın çalışan bir model prototipi yapılır ve şüpheli olan, belirsizlik görülen noktalar prototipten yola çıkılarak değerlendirilmeye tabi tutulur. Kullanıcı bu Şekilde sistemin neye benzeyeceği hakkında fikir sahibi olur.



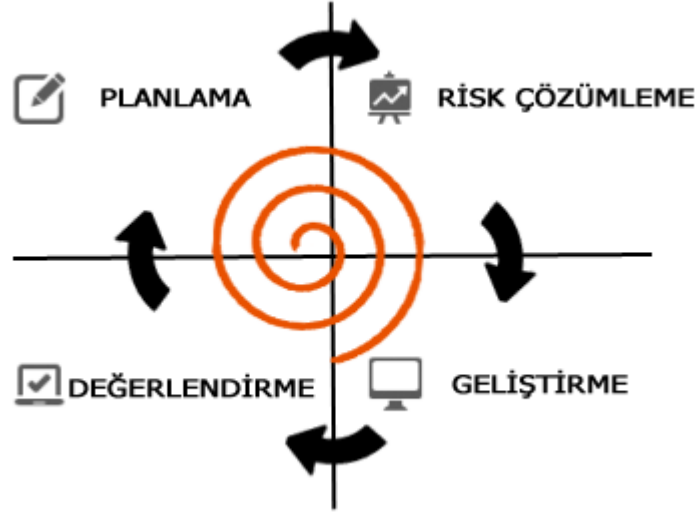
Şekil 2.6 Prototip Modeli

2.4.4 Spiral Model

Hem geleneksel (şelale) hemde prototipleme modellerini olumlu yönleri alınarak oluşturulmuş model. Öncelikle müşterinin netleştirilmiş istekleri analiz edilir, sonrasında bu ihtiyaçların gerçekleştirilmesi yapılarak ürün ortaya çıkarılır test edilir ve bir sürüm olarak teslim edilir. Sonrasında geriye kalan analizler, güncellemeler ve geliştirmeler birer sürüm olarak kullanıcıya verilir.

Spiral model aşamaları;

- *Planlama*
- *Risk çözümlemesi*
- *Geliştirme*
- *Değerlendirme*



Şekil 2.7 Spiral Model

2.4.5 Artımlı Geliştirme Modeli

Yazılım geliştirme öncesinde, yazılımdan beklenen davranışların tamamının belirlendiği ve ürünün artımlı sürümler halinde geliştirildiği modeldir. Bu modelde akışlarla ilgili temel işlevleri yerine getirebilme durumu gözetilerek geliştirilir ve ürün ortaya çıkar. Sonrasında eksik kalan işlevler için yapılan geliştirmeler ürüne eklenir. Eklenen her yeni sürümde bir öncekinden fazla işlev bulunmaktadır.

2.4.6 Evrimsel Geliştirme Modeli

Aşamalar halinde ürün ortaya çıkarmayı hedefler. Her aşamada ortaya çıkan ürün o an için tüm ihtiyaçlara cevap verir nitekite olmalı, ürünün kullanımında elde edilen dönütler, ürünün hitap ettiği alandaki değişimler , donanım ve yazılım temelli teknolojik gelişmeler neticesinde var olan yeni ihtiyaçlara cevap verecek yeni evredeki ürün bir sonraki aşama için sunulur. Her aşamada ürünün yetenekleri artmış olmalıdır.

2.4.7 Evrimsel Prototipleme Modeli

Evrimsel geliştirme modelinden farklı olarak ortaya çıkan bir ürün başlangıçta bir prototiptir. Bu prototiple birlikte müşteri deneyimlerinden elde edilen dönütler alınarak ilk evrim için ürün geliştirilir. Her evre için aynı aşamalar yani prototipler uygulanır dönütler alınır geliştirmeler tamamlanır ve ürün ortaya çıkar. Kullanıcıların tam olarak isteklerini netleştiremediği durumda kullanımı daha uygundur.

2.4.8 Çevik Yaklaşım

20. yüzyılda teknolojik gelişimlerde yaşanan ivmelenme , 21 yüzyılın ilk çeyreğinin sonu ve ikinci çeyreğinin başı itibari ile kontrol edilmesi güç bir hıza ulaşmıştır. Teknolojik gelişimdeki bu hız artışı sonucu, gelişimin merkezindeki bilgi sistemlerinin etkilemesi beklendik bir durumdur. Artan rekabet ortamı, gelişen bilgi sistemleri ile küresel rekabet olanağının artması ile daha talepkar olan müşterilerin; en yüksek kaliteyi en az maliyetle edinebilme, kısa sürede teslim alma, sürekli yaşanan değişime ayak uydurma esnekliği ve güncellenme istekleri ile karşı karşıya kalmaktayız.

Bunun dışında bu hız içerisinde oluşan devasa bilgi birikimini yönetmek, bu bilgiden kurum yararına mining operasyonlarını oluşturmak, bu know-how ile şirketin rekabet ortamına yararlı geri dönütler sunmak, kullanıcılar için önem arz etmektedir.

Tam bu noktada bir felsefe olarak *Çevik Yaklaşım*, çevik yaklaşım manifestosu ve ilkelerini kendine esas almış, dünya çapında, tüm yazılım faaliyet ve süreçlerini esnek güçlü kılmayı hedefleyen, geliştirme süreçlerini kısaltan ve müşteriye değer katmayan süreçleri reddeden bir yöntem olarak karşımıza çıkmaktadır. Temelde bir felsefeyle ve yazılım gelişimi üzerine düşünmeyi ifade eder.

Çevik gelişim teknikleri, külfet görülebilecek bazı süreç ve yapılan analizlerin uzun uzun dökümantasyonu yerine direkt olarak yazılımın kendisine yoğunlaşan, geliştirme esnasında yenilenmeye açık teknikleri ifade eder. [Nizam 2014]

Çevik yaklaşımda ortaya konan işin boyutu ne olursa olsun, proje kullanıcı beklentilerini karşılayacak en küçük parçalara ayrılır. Bu parçalar iterasyon olarak adlandırılır ve her bir iterasyon kendi içinde bir proje olarak ele alınır ve geliştirilir. Her iterasyonun 2-4 hafta aralığında gerçekleştirilebilir olması beklenir. İterasyonlar kendi içlerinde çalışan bir yapı oluşturur. Yani her bir iterasyon sonunda müşteriye genel projenin çalışır bir parçası teslim edilmiş olmalıdır. İterasyon bitiminde müşteriye projenin ne kadarının bittiği , ne kadarının kaldığı ile ilgili bilgiler verilebilir durumda olunur. İterasyonlar projenin en ufak parçalara ayrılmış biçimleri olduğu için iterasyon bitiminde alınan dönütler yada belirlenen problem durumları çok hızlı bir biçimde çözümlenebilir duruma gelir. Tüm işlemler geliştirme metodolojisinin esneklik, verimlilik, düşük hata oranı ve gelişim hızının yüksek olmasını sağlar.

2.4.8.1 Çevik Yaklaşım Manifestosu

Çevik yaklaşımın bir felsefe olarak temelde kendine çevik yaklaşım manifestosunu ilke edinmiştir.

2001 yılında Kent Beck ve 16 arkadaşı bir araya gelip, yazılım geliştirme sürecindeki verimliliği arttırma, geleneksel ağır aksak süreçlerin dışına çıkma adına bir manifesto yayınlamışlardır. Dört ana ilke ile tanımlanmış bu manifesto önceliklerin tanımlanması ve odak noktasının değişimine işaret etmektedir.

Manifesto for Agile Software Development		
1. Individuals and interactions over process and tools 2. Working software over comprehensive documentation 3. Customer collaboration over contract negotiation 4. Responding to change over following a plan That is, while there is value in the items on the right, we value the items on the left more.		
Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Tablo 2.1 Çevik Manifesto (Orginal)

Kaynak : <http://agilemanifesto.org/> adresinden alınmıştır

Çevik Yazılım Geliştirme Manifestosu		
1. Bireyler ve etkileşimler, süreç ve araçların üstündedir. 2. Çalışan program, kapsamlı dokümantasyonun üstündedir. 3. Müşteri ile işbirliği, sözleşmeler ve anlaşmaların üstündedir. 4. Değişikliklere yanıt vermek, bir planı takip etmekten daha üstündür. Burada sağda yer alan öğeler bir önem taşıırken solda yer alan öğeler bizim için daha değerlidir.		
Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Tablo 2.2 Çevik Manifesto (Türkçe)

Kaynak : <http://agilemanifesto.org/> adresinden alınmıştır

- **Bireyler ve etkileşimler**, süreç ve araçların üstündedir. (*Individuals and interactions over process and tools*)

Yazılım projesini geliştiren takımlar bireylerden oluşmaktadır. Bu takımın paydaşları müşteri, geliştirici, proje yöneticisi, testçi ve tasarımcılardır. Bir takıma ait paydaşların birbirleriyle etkileşiminin üst düzey olmaması durumunda, araç ve süreçler ne kadar iyi olursa olsun işlevleri ve etkinliklerinin yararı olmayacaktır.

- **Çalışan program**, kapsamlı dokümantasyonun üstündedir. (*Working software over comprehensive documentation*)

Projeye dahil olan tüm paydaşların tercihi sayfalarca süren sıkıcı, yorucu dökümanlar olmamalıdır. Bunu yerine hızla tamamlanmış integrasyonları içeren çalışan bir yazılım tercih edilebilir. Döküman işi tarif etme, standart belirleme, iş sonrasında eğitim faaliyetleri açısından önemli kılavuzlardır ve olmalıdırlar. Ancak geliştiricilerin birincil hedefi yazılım ortaya koymak olmalıdır. Başlangıçta belirlenen müşteri isteklerinin sayfalarca döküman haline gelmesine harcanacak enerji hızlı bir yazılım üretim sürecine harcanıp sonrasında bu yazılımın gelişmesine olanak sağlamalıdır.

- **Müşteri ile işbirliği**, sözleşmeler ve anlaşmaların üstündedir. (*Customer collaboration over contract negotiation*)

Müşterilerle karşılıklı iyi niyet ve anlayış esasına dayanan sözleşmeler olmalıdır. Ancak bu sözleşme esasında işin boyutlarının, varacağı son noktanın tarifi net yapılamayabilir. Nitekim müşterilerin kendi ihtiyaçlarının yüzde yüz tarifini yapabilecek becerileri olmayabilir. İstekler ve akışlar zamanla değişebilir. Dolayısı ile buradaki temel amaç sözleşme kısıtlaması ile yola çıkmak yerine müşteriyi paydaş olarak görüp birlikte çalışma, karşılıklı dönütlerle süreci besleme ve nihai ürünün gelişme esasında son halini bulmasını sağlamaktır.

- Değişikliklere yanıt vermek, bir planı takip etmekten daha üstündür. (Responding to change over following a plan)

Bir proje planı esnek ve değiştirilebilir olmalıdır. Müşteri istekleri, çalışma alanındaki yenilikler, teknolojik değişimler, müşteri operasyonlarının revizyonu, yazılımı geliştirdiğimiz sektörün problemlerinin yarattığı ekstra durumlar ve akışlar, yönetmelik ve idari düzendeki yöntem ve hukuki değişimler ve bu gibi akla gelebilecek proje içerisinde değişim ihtiyacı yaratan isteklere cevap veriyor olabilmek esastır.

2.4.8.2 Çevik Yaklaşım Prensipleri

Değişen müşteri istek ve talepleri ürünün gelişimi esnasında projeye implement edilir. [Saridoğan 2011]

Principles behind the Agile Manifesto
Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Business people and developers must work together daily throughout the project.
Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Working software is the primary measure of progress.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Continuous attention to technical excellence and good design enhances agility.
Simplicity--the art of maximizing the amount of work not done--is essential.
The best architectures, requirements, and designs emerge from self-organizing teams.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Tablo 2.3 Çevik Prensipler (İngilizce)

Kaynak : <http://agilemanifesto.org/> adresinden alınmıştır

Çevik Yaklaşımın Prensipleri
Önceliğimiz, faydalı yazılımların erken ve sürekli teslimatıyla müşteri memnuniyetini sağlamaktır.
Yazılımın ilerleyen dönemlerinde gelse bile talep edilen değişiklikler hoş karşılanmalıdır. Çevik süreçler, değişiklikleri müşterinin rekabetteki avantajını korumak ve sağlamak için kullanılırlar.
Kısa zaman aralıkları tercih edilerek iki haftadan iki aya kadar çalışır yazılım ortaya koyulmalıdır.
Müşteri ve yazılımcılar proje süresince beraber çalışmalıdır.
Projelerin motivasyonu yüksek bireyler tarafından yapılmasını sağla, onlara ihtiyaç duydukları ortamı ve desteği ver ve işi bitirebileceklerine inan.
Bilgi alışverişinde en verimli ve efektif yöntem takım içinde yüz yüze konuşmaktır
Çalışır durumda olan program ilerlemenin ana göstergesidir.
Çevik süreçler etkili yazılım yöntemlerini destekler. Müşteri, yazılımcılar ve kullanıcılar sabit bir tempoda beraber çalışabilmelidirler.
Teknik mükemmelliğe devamlı özen gösterilir ve iyi tasarım çevikliği kuvvetlendirir.
Sadelik (basitlik) esastır.
En iyi mimariler, gereksinimler ve tasarımlar kendi kendine organize olabilen takımlardan çıkar.
Belirli zaman dilimlerinde takım nasıl daha etkin olabileceği konusunda kendini sorgular ve edindiği bilgiler doğrultusunda çalışma tarzını adapte eder.

Tablo 2.4 Çevik Prensipler (Türkçe)

Kaynak : <http://agilemanifesto.org/> adresinden alınmıştır

- Önceliğimiz, faydalı yazılımların erken ve sürekli teslimatıyla müşteri memnuniyetini sağlamaktır. (Our highest priority is to satisfy the customer through early and continuous delivery of valuable software)

Yazılım geliştirilmesi esnasında müşteriye mümkün olan en kısa sürelerde çalışır vaziyette bir program teslimiyle müşteri memnuniyetinin sağlanması hedeflenir. Müşteri bu teslimatlarla birlikte iş hakkında gerekli kontrolleri ve değerlendirmelerini paylaşabilir, isteklerinin ne oranda gerçekleştiğini görebilir. Geri beslemelerle sorunlarının veya yeni isteklerinin uyarlanması gerçekleşebilir. Bu yöntemle proje bitiminde yazılan yazılımın müşteri istekleriyle yüksek oranda tutarlılık sağladığı görülür.

- Yazılımın ilerleyen dönemlerinde gelse bile talep edilen değişiklikler hoş karşılanmalıdır. Çevik süreçler, değişiklikleri müşterinin rekabetteki avantajını korumak ve sağlamak için kullanılırlar. (Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.)

Müşteri talep ve isteklerinin proje geliştirme esnasında dikkate alınıp, uyarlanmasını ifade eden bu prensip gelen talep projenin gelişim evrelerinin hangi aşamasında olursa olsun ister başında ister sonunda implementasyonunun yapılması gerekliliğine vurgu yapar.

- Kısa zaman aralıkları tercih edilerek iki haftadan iki aya kadar çalışır yazılım ortaya koyulmalıdır. (Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.)

Proje iterasyonlara bölünerek 2-4 hafta aralığında çalışır yazılımlar müşteriye sunulur. Her iterasyonda yazılım yeni ve müşteriye ait gerekli yetenekleri kazanmış olur.

- Müşteri ve yazılımcılar proje süresince beraber çalışmalıdır. (*Business people and developers must work together daily throughout the project.*)

Çevik yaklaşımda proje paydaşlarının tamamı birlikte olumlu bir etkileşim içerisinde çalışıyor olmalı.

- Projelerin motivasyonu yüksek bireyler tarafından yapılmasını sağla, onlara ihtiyaç duydukları ortamı ve desteği ver ve işi bitirebileceklerine inan. (*Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.*)

Projede yer alan geliştiriciler arasında bir kıdemsel fark yoktur. Ekip elemanları birbirleriyle bilgi alışverişi yaparak bilgi birikim açısından aralarındaki farkı kapatırlar. Proje dahilinde ekip elemanlarına gerekli güvenin, çalışma ortamının verilmesi gerekir. Bu sayede kendilerine olan güvenleri artar, yüksek motivasyonla işin biteceğine olan inançları sağlamlaşır.

- Bilgi alışverişinde en verimli ve efektif yöntem takım içinde yüz yüze konuşmaktır. (*The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.*)

Paydaşların birbiriyle iletişimde güven duygusunu arttırmak, anlaşılır net ve eksiksiz bilgi aktarımını sağlamak için en önemli metot olarak yüz yüze iletişim kullanılır. Bu sayede yanlış anlaşılımların önüne geçilir ve yazılı iletişimin hantallığından kurtulmuş olunur.

- ***Çalışır durumda olan program ilerlemenin ana göstergesidir. (Working software is the primary measure of progress.)***

Geleneksel yöntemlerde ürün geliştirme faaliyetlerinin sonunda ortaya çıkar. Müşteri o ana dek programla ilgili gelişimin ne aşamada olduğunu göremez. Ancak çevik metodolojide belli aralıklarla müşteriye çalışır bir yazılım teslim edilir. Müşteri açısından bu yöntemle elde edeceği ürüne ait gelişim gözlemlenmiş olur.

- ***Çevik süreçler etkili yazılım yöntemlerini destekler. Müşteri, yazılımcılar ve kullanıcılar sabit bir tempoda beraber çalışabilmelidirler. (Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.)***

Süreçte sabit bir çalışma hızı ile ilerlemelidir. Görevler eşit dağıtılmıştır. Fazla mesai durumu yaşanmamalıdır.

- ***Teknik mükemmelliğe devamlı özen gösterilir ve iyi tasarım çevikliği kuvvetlendirir. (Continuous attention to technical excellence and good design enhances agility.)***

Kalite beklentisi oldukça yüksektir. Proje için oluşturulan ekipler işlerinde uzman kişiler olmalı. Program tasarımsal ve yazılımsal olarak sürekli optimize edilir. Yeni talep ve düzenleme isteklerine anında reaksiyon verilerek yazılım implementasyonu yapılır. Tüm bunlarda çevik yaklaşımla geliştirilen yazılımın kaliteli olmasını sağlamaktadır.

- **Sadelik (basitlik) esastır.** (*Simplicity--the art of maximizing the amount of work not done--is essential.*)

Projenin mümkün olan en sadeleştirilmiş Şekille tasarlanmasının yazılımsal süreç açısından işleri kolaylaştıracağını söylemek doğru olur. O anki beklentileri eksiksiz karşılayan, anlaşılır, değiştirilmesi kolay ve maliyeti düşük çözümler tercih edilmelidir.

- **En iyi mimariler, gereksinimler ve tasarımlar kendi kendine organize olabilen takımlardan çıkar.** (*The best architectures, requirements, and designs emerge from self-organizing teams.*)

Ekipler kendini organize edebilme yeteneğine sahip olmalıdır. Ekip elemanları gereksinimleri ortaya koyar, iş planı çıkarır, iş bölümü yapar, mimari ve tasarımı optimize edebilirler. Dolayısı ile ekip bu süreçleri yapabilecek bilgi ve beceriye sahip olmalıdırlar.

- **Belirli zaman dilimlerinde takım nasıl daha etkin olabileceği konusunda kendini sorgular ve edindiği bilgiler doğrultusunda çalışma tarzını adapte eder.** (*At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.*)

Müşteri tarafından onay görmüş etkili yazılım çözümünü en verimli Şekilde ortaya çıkarabilme amacındaki ekiplerde sürekli öğrenme esastır. Belirli zamanlarda bir araya gelerek bilgi alışverişinde bulunup süreci ve ilerlemeyi değerlendirip, gerek görülürse yeniden yapılandırmaya gidebilirler.

2.4.8.3 Çevik Yaklaşım Türleri

Çevik yaklaşımda birden fazla süreç yer almaktadır. Bazı özellikleri itibarı ile birbirlerine benzeyen bu yöntemler gerçekleşme açısından kendilerine has bir yapı, terminoloji ve taktik barındırmaktadır.

2.4.8.3.1 Uç Programlama - Ekstrem Programlama - EP (Extreme Programming)

Chrysler firması için Kent Beck, Ron Jeffries ve Ward Cunningham tarafından yapılan bir proje sonrasında ortaya çıkmış bir çevik yaklaşım türüdür. Değişen müşteri istekleri için geleneksel metodun yazılım öncesi istekleri en ince ayrıntısına kadar tasarlaması ve dökümanete etmesinden imtina ederek en sade ve en basit temel bir analizle yazılımı biran önce müşteriye teslim etme çabası içerisindedir. Müşteri, geliştiriciler ve yöneticiler bir ekip halinde çalışırlar. Değişen müşteri istekleri gelişim evresi içerisinde muhakkak karşılanır.

XP dört değer üzerine kuruludur: Basitlik (*simplicity*), İletişim (*communication*), Geri besleme (*feedback*) ve Cesaret (*courage*).

2.4.8.3.2 Dinamik Sistem Geliştirme Metodu – DSGM (Dynamic System Development Model)

Dinamik sistem geliştirme metodu Büyük Britanya’da 90’lı yıllarda konsorsiyum tarafından geliştirilmiştir. Fonksiyonel gereksinimlerin temel alındığı bu metotta diğer metotlara oranla biraz daha detay içerir.

Etkin bir takım, genellikle 2 – 6 hafta arasında değişen iterasyonlarla sonlandırılmış teslimler, sıkı işbirliği ile desteklenmiş takım çalışmaları, tüm arttırım süreçlerinde sıkı test prensipleri, aktif kullanıcı etkileşimini içeren bir yöntemdir. Kullanıcı grubunun belirli olduğu ve kesin zaman kısıtlarının olduğu projelerdir. Dolayısı ile gereksinimler önem sıralamasına dizilebilir.

2.4.8.3.3 Uyarlanabilir Yazılım Geliştirme - UYG (Adaptive Software Development)

Takım üyelerinin işbirliği ile süreçten ziyade ürüne odaklandığı temel bir süreci benimser. Belirsizlik durumlarında çözüm için insanlar arasında güçlü bir iletişim kurmak, tüm paydaşları iletişime yönlendirerek çözümün yaratıcı bir unsuru olmasını sağlamak esastır. UYG süreci 3 bölümden oluşur.

- *Tahminde bulunma:* Amaç belirlenmesi ve geliştirme için gerekli bilgilerin bir plan dahilinde ortaya konması.
- *İş Birliği:* Eş zamanlı olarak bileşenlerin geliştirilmesi yapılır.
- *Öğrenme:* Müşteri ve geliştiricilerin proje başında toplantıları ve kalite incelemesi için yapılan toplantıları kapsar.

2.4.8.4 Kristal (Crystal)

Kristal, insan iş gücüne inanan, iletişimi merkezine kabul etmiş bir metotlar topluluğudur.. Bu süreçler aşağıdaki durumları temel almaktadır.

- Her proje farklı birtakım kural, uyum ya da yöneme ihtiyaç duymaktadır.
- Proje çalışmaları, insan ile ilgili konularda hassastır ve bu konuların gelişimine bağlı olarak daha da gelişir; insanlar daha iyi oldukça, grup içi çalışmalar daha da iyi olmaktadır.
- Etkin iletişim ve sık yapılan teslimler orta seviye iş ürünlerine gereksinimi azaltmaktadır.

- *Kristal süreçlerin iki kuralı bulunmaktadır.*
 - o 1 ile 3 ay arasında değişen en fazla 4 ay süren artımlarla geliştirim yapılmaktadır.
 - o Her bir artım öncesi ve sonrası başarılar ve başarısızlıkların dile getirildiği fikir seminerleri düzenlenmektedir.

2.4.8.5 Scrum

Scrum seksenli yıllarda Kent Schwaber ve Jeff Sutherland tarafından geliştirilmiş bir süreçtir. Scrum, Rugby oyunundan alınmış bir terimdir. Oyuncular bir sonraki oyun için geçici bir süre bir araya gelip fikir alışverişi yaparlar.

Farklı çevik yaklaşım metodolojileriyle rahatlıkla kullanılabilir. Kompleks iş sürecini, yazılım evrelerini küçük sprintlere (birim) bölerek adım adım geliştirmeye olanak veren bir ekosistemi tanımlar. İhtiyaçların net belirtilemediği, karmaşık projeler için etkili bir yöntemdir.

İçerisinde çeşitli süreçleri ve teknikleri kullanabileceğiniz bir çerçeve olarak nitelendirilebilecek olan scrum, ürün yönetimi ve geliştirme pratiklerinizin etkililiğini açık bir şekilde ortaya koyarak iyileştirme fırsatı sunar.

Scrumdaki önemli düşüncelerden bir tanesi ekip üyelerinin işi nasıl yapacaklarına kendilerinin karar vermeleridir. Stratejik hedeflerin yani nihai hedeflerin dışında o hedefe nasıl ulaşılacağına dair takip edilecek yol, yöntem seçimi karar verme aşamasında ekibe aittir. [Sutherland 2015]

Scrum genellikle 10-30 günlük iterasyon sürelerini kapsar. 5-10 kişilik kendi kendini düzenleyen ve yönlendiren takımlar iterasyon içerisinde dokunulmazdır. Her iterasyon müşteri güdümlüdür ve adapte olan bir planlama içerir. Genellikle 3 - 12 ay süren kısa süreli projelerde iterasyon sonrası mutlaka paydaşlara bir demo sunulur. Takım üyelerinin bir araya geldiği belirli soruların sorulduğu günlük scrum toplantıları yapılır. Kararlar kısa sürede alınır ve hemen uygulanır.

2.4.8.5.1 Scrum Temel Kavramlar

Scrum’da temelde 3 kavram vardır. Bunlar scrum takımı, zaman planlaması ve scrum araçlarıdır.



Şekil 2.8 Scrum Temel Kavramlar

2.4.8.5.2 Roller

Scrumda roller 3'e bölünmüştür. Ürün sahibi (product owner), scrum yöneticisi (scrum master) ve geliştirme takımı.

Ürün Sahibi (Product Owner) : Müşteri Tarafından Görevlendirilmiş, Proje ve gereksinimler hakkında takımı yönlendiren, ürünün öncelikli hedeflerini ortaya koyan, takımın işini ve ürün değerini en üst düzeye çıkarmayı hedefleyen kişi. Ürün sahibi bir tanedir. Sorumluluk her zaman ürün sahibindendir.

Ürün sahibinden beklenen davranışlar;

- İş listesini anlaşılır, açık ve net ifade etmek.
- İş listesinin gerçekleşmesini sağlayacak sıralamayı yapmak
- Takımın iş listesindeki her bir detayı özenle anlamasını sağlamak
- İş listesi ve takımın sonraki işlerini görünür kılmak
- Her iterasyon sonunda ürün kabulü yapmak

Scrum Yöneticisi (Scrum Master) : Scrum yöneticisi scrum takımının lideridir. Ekibin ilerlemesini destekler ve yaşanan problemlere çözüm bulmaya çalışır. Scrum takımının scrum teorisine, pratiklerine ve kurallarına uygun olarak işlenişini gerçekleştirmesini sağlamakla yükümlüdür.

Scrum Yöneticisi aşağıdaki pratiklerden sorumludur;

- Scrum etkinliklerini yönetmek.
- Geliştirme Takımının ilerlemesine engel oluşturan unsurları ortadan kaldırmak
- Organizasyondaki Scrum uygulamalarının etkililiğini artırmak üzere diğer Scrum masterlarla birlikte çalışmak.
- Takımın dış etkilerden korunmasını ve sadece kendi işine yoğunlaşarak üretkenliğinin artmasından sorumludur.

Geliştirme Takımı/Scrum Takımı (scrum team) : Projenin gelişiminden sorumlu scrum prensiplerini benimsemiş her biri alanında uzman bireylerin oluşturduğu takımlara verilen isimdir.

Geliştirme takımı, her bir sprintin (iterasyon) bitimiyle , ürünün bitmiş bir parçasığını teslim etmekten sorumludur. Her bir iterasyonu sadece geliştirme takımları geliştirirler.

Geliştirme takımları;

- Bir iterasyonu tamamlayacak bilgi ve beceriye sahip bireylerden oluşurlar.
- Scrum takımı içerisinde, hiçbir alt role izin verilmez test ve analist gibi uzmanlıklara bakılmaz. Hepsi geliştirici olarak değerlendirilir.
- Kendi kendilerini organize edebilme becerileri yüksektir.
- Geliştirme Takımı üyelerinin uzmanlaştıkları belli beceriler veya odak alanları olabilir fakat iş sorumluluğu bireysel değil geliştirme takımınınındır.
- Gerekli süre tahminini yaparlar
- 5 – 9 kişiden oluşur

Geliştirme takımları hızlı ve esnek davranabilecek kadar küçük, etkileşimi sağlayabilecek ve iterasyonlardaki tanımlanmış işleri beceri ile bitirebilecek potansiyeli taşıyacak kadar büyük olmalı. Küçük takımlarda etkileşim eksikliği olacağı gibi büyük takımlarda da çok fazla koordinasyona ihtiyaç duyarlar. Ürün sahibi ve scrum master bu ekibe dahil değildir. (sayısal olarak)

2.4.8.5.3 Scrum Etkinlikleri / Zaman Planlaması / Toplantılar (Scrum Events / Meetings)

Koşu (sprint) : Koşu , çevik yaklaşım değerleriyle iterasyon, maximum bir aylık dilimi kapsayacak şekilde, açıkça ifade edilen iş listesine uygun , kullanılabilir bir ürün ortaya çıkarmak için geliştirme takımı, scrum master ve ürün sahibinin faaliyetlerini içeren sürecin tamamının tanımını kapsar. Koşu sonunda projeye ait alt bir ürün oluşturulur. Sprintlerin süresi sabittir. Önceki sprint biter bitmez yeni sprint başlar. Her bir Sprintin, neyin üretileceğine ilişkin bir tanımı, üretime rehberlik edecek bir tasarımı ve esnek bir planı, işin kendisi ve sonuçta ortaya çıkacak olan ürünü vardır. Her bir Sprint bir aydan uzun bir ömrü olmayan bir proje olarak düşünülebilir. Projeler gibi Sprintler de bir şeyi başarmak için kullanılır.

Sprintler; Sprint Planlama, Günlük Scrumlar, geliştirme işi, Sprint Değerlendirme ve Sprint Retrospektifinden oluşur

Sprint Planlama (Sprint Planning): Sprint başlamadan önce yapılan toplantıdır. Ürün sahibi sprintin başarması gereken amaç ve hedefle gerekli iş listesini açıkça tanımlar ve bu tanımlardan hareketle ekibin tüm talepleri değerlendirdiği geliştirme ihtiyaçlarının konuşulduğu süreci tanımlar. Sprint sonunda teslim edilecek iş parçacığı olarak ne teslim edilecek ve bu iş parçacığı geliştirilirken hangi yöntemler izlenecek sorularının cevapları aranmaktadır.

Günlük Scrum (Daily Scrum): Günlük olarak gerçekleştirilen ve 15 dakika geçmeyen toplantılardır. Bu toplantılarda 3 soru sorulur;

- Geliştirme Takımının Sprint Hedefine ulaşması için dün ne yaptım?
- Geliştirme Takımının Sprint Hedefine ulaşması için bugün ne yapacağım?
- Beni veya Geliştirme Takımını Sprint Hedefine ulaştırmaktan alıkoyacak bir engel görüyor muyum?

Günlük Scrumlar iletişimi iyileştirir, başka toplantılara olan ihtiyacı ortadan kaldırır, geliştirmenin önündeki engellerin tespit edilmesini sağlar, hızlı karar almayı teşvik eder. Bu etkinlik kilit bir gözlem ve adaptasyon toplantısıdır.

Sprint Değerlendirme (Sprint Review) : Sprint bitiminde , sprintin bir değerlendirmesini yapmak, gözden geçirmek ve geri dönütler almak için, ürün parçacığının kontrol edilmesini sağlayan toplantılardır. Ürün sahibi iş listesinin ne kadarının bittiğini görür. Geliştirme takımı bir sprint boyunca neyin yolunda gidip, o ana kadar hangi problemlerle karşılaşıldığını ortaya koyar. Sprint sonundaki ürün gösterilir , tartışılır, gelen sorular yanıtlanır. İş listesiyle ilgili değerlendirme yapılır.

Sprint Retrospektifi/Geriye dönük koşu değerlendirme (Sprint Retrospective) : Sprint Retrospektifi, Scrum Takımının kendini değerlendirerek sıradaki Sprintte yapacağı iyileştirmelere ilişkin öngöründe bulunmasını sağlayan toplantılardır. Sprint Retrospektifinin sonunda, Scrum Takımı sıradaki Sprintte uygulayacağı iyileştirme alanlarını tespit etmiş olur.

Sprint Retrospektifinin amaçları şunlardır:

- Son Sprintin insanlar, ilişkiler, süreç ve araçlar bakımından nasıl geçtiğini gözlemlemek
- İyi giden noktaları ve muhtemel iyileştirme alanlarını tespit edip sıralamak
- Scrum Takımının iş yapış tarzını iyileştirecek bir plan oluşturarak bir sonraki sprintin veriminin artmasını sağlamak.

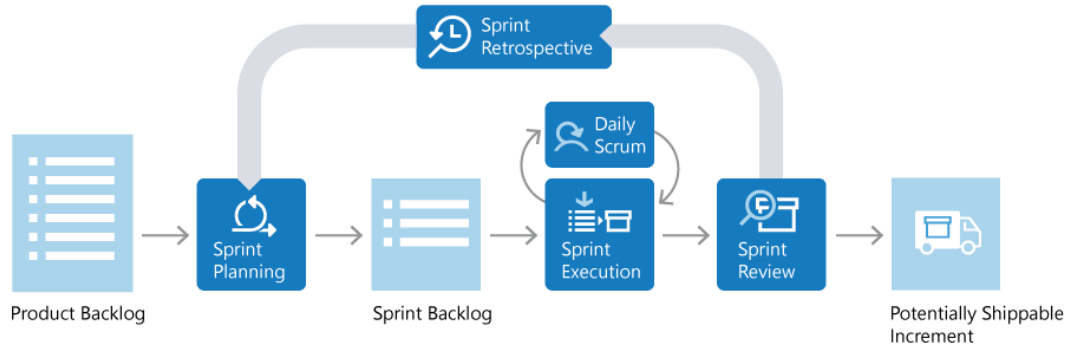
2.4.8.5.4 Scrum Kavramlar / Araçlar (Scrum Artifacts)

Ürün İş Listesi/Ürün talebi (Product Backlog) : Ürüne ait beklenti ve gereksinimlerin sıralandığı bir listedir. Ürün listesi başlangıçta bilinen ve en iyi tanımlanan temel gereksinimleri içerir. Ürün kullanıldıkça ürün iş listesi değişebilir. Ürün listesi yaşayan dinamik bir listedir. Kullanıldıkça geri bildirimlerle daha da genişleyen ve zenginleşen bir yapısı vardır.

Sprint İş Listesi/Koşu talebi (Sprint Backlog) : Sprint iş listesi günlük scrum faaliyetlerinin gerçekleşmesi ve iterasyonun nihai hedefe ulaşmasında sprint için seçilen ürün iş listesi kalemlerini içerir. Ürün parçacığında hangi fonksiyonların olması gerekliliğini tanımlar. Sprintin kendisinden beklenen hedeflerini ve buna ulaşma planlarını içerir.

Release Burndown : Kalan gereksinimler / geçen zaman grafiğidir. Proje başlamadan product backlog içerisindeki tüm gereksinimlerin bir grafikte dikey olarak yazılır. Ardından her sprint bittikten sonra yatay bölüm biten gereksinimlerle güncellenir ve böylelikle projenin hayat sürecinde yukarıdan aşağıya doğru giden bir yatay çizgi oluşur.

Sprint Burndown: Sprint için geçerli olan kalan gereksinimler/geçen zaman grafiğidir. Sprint boyunca takım üyeleri sorumlu oldukları görevler hakkında kalan saat/gün Şekilinde bilgi verirler günlük olarak. Ve bu bilgiler bu grafikte günlük olarak güncellenir. Bu sayede bir anda sprint gidişatı hakkında bilgi sahibi olabilir ve olası sorunları önceden görme şansımız olabilir.



Şekil 2.9 Scrum Yaşam Döngüsü

Scrum sürecinin getirileri;

Scrum özellikleri itibarı ile tüm dünyada etkin bir şekilde kullanımını sürdüren, çevik metotlar içerisindeki en etkin yöntem olarak öne çıkmaktadır.

Scrum aşağıdaki avantajları beraberinde getirir;

- İterasyon ile projenin parçalara ayrılarak karmaşıklığın azaltılması sağlanır.
- Ürün sahibinin takımın bir parçası gibi sürekli işbirliği içinde çalışması, tüm beklentilerin sürece aktarımını buda müşteri memnuniyetinin artmasını sağlar.
- Öncelikli isteklerin ürün sahibi tarafından belirtilerek bunların çözülmesi ve rekabet avantajının müşteriye verilmesi sağlanmaktadır.
- Önceden risklerin fark edilerek azaltılması sağlanabilir.
- Hata durumlarının erken tespiti ve anında müdahalenin yapılmasıyla, gelişim süreci hızı artmış olur.

2.4.8.5.5 Çevik Yaklaşım Avantajları ve Dezavantajları

Avantajları:

- Kısa planlanmış döngüler, değişen gereksinimler ve ani ortaya çıkabilecek değişikliklerin implementasyonu için organizasyonunuza esneklik imkânı tanır.
- Müşterileriniz ürünü oluşturmak ve çıkan problemler kolayca çözebilmek için aktif bir şekilde geribildirim sağlar.
- Ekip katılımına öncelik verilerek kurulan iletişim sayesinde sorunların daha hızlı çözülmesini sağlar.
- Proje hızla ilerlerken ortaya çıkan her problem bir sonraki döngüde daha iyi bir çözüm için basamak olarak kullanılır.
- Yapılan toplantılar ve iletişimin sürekliliği takım içi özgüveni artırır.
- Organizasyon içi gelişim olanağı sağlar

Dezavantajları:

- Ürün gereksinimleri sürekli değişebileceğinden maliyetler peşinen tahmin edilemez.
- Müşterilerden sürekli geri bildirim almak sizi alaşağı edebilir ya da bazı müşterilerin zamanı ve hiç ilgisi olmayabilir.
- Müşteri ile iletişimi zor olan büyük Kurumsal yapılarda uygulamak zor olacaktır.
- İyi tasarlanmamış bir sprint iş listesi sprintin çıktısını olumsuz etkileyeceği gibi, motivasyon kaybına neden olur.

2.4.8.5.6 Scruma Başlamak

[Sutherland 2014]

1-Bir ürün sahibi olmalı ; Bu kişi sizin yapacağınız ya da başaracağınız şeyle ilgili vizyona sahip kişidir. Riskleri, ödülleri, olması gereken işleri, neler yapılabileceğini ortaya koyan ve açıkça ifade etmesi gereken kişidir.

2-Bir Ekip Oluşturmalı ; Bu ekibin ürün sahibinin ortaya koyduğu vizyon ve gereklilikleri gerçeğe dönüştürebilecek yetenek ve beceriler ortaya koyması gerekir. Ekipler küçük olmalı 3 ila 9 arası idealdir.

3-Bir Scrum Ustası Olmalı ; Ekibin kalanına Scrum'un çatısını oluşturma konusunda önderlik ederek, tüm üyelerin scrum prensiplerine bağlı kalmalarını ve eksiksiz uygulamalarını sağlar .

4-Bir Ürün İş Listesi Oluşturmalı ; Ortaya konan vizyonun gerçek kılınması için gereken her şeyin listesidir. Bu iş listesi ürünün yaşam süresince var olur ve evrim geçirir; ürünün yol haritasıdır. Sadece bir tane ürün listesi olabilir, bu da ürün sahibinin ürün yelpazesine bakarak önceliklendirdiği ekip tarafından tamamlanacak her şeyi ifade eder.

5- Ürün iş Listesini İyileştirme ve Önceden Kestirme ; İş Listesindeki işleri tamamlayacak olan ekip elemanlarının , ne kadar emek harcayarak işi bitirebileceklerini tahmin etmeleri önemlidir. Ekip iş listesinin her bir kalemi için gerçekleştirmenin mümkün olup olmadığını kestirmelidir.

Bu kestirim için cevap aranan sorular;

- Her bir kalemi tamamlamak için ekip gerekli bilgiye sahip mi?
- Öngörülebilir bulunulacak kadar küçük bir parçaya ayrılmış mı iş ?

- Herhangi bir şeyin “tamamladığının” bir göstergesi oluşturuldu mu ? Herkes bu gösterge için fikir birliğinde mi? Şimdiye kadar tamamlananların göstergesi nedir? Tanımı nedir ?
- Tamamlanan her iş için ürün yetenekleri açısından bir değer artışı var mı?

Her bir kalem iş, bitişinde, gösterilebilir, demo edilebilir ve teslim edilebilir olmalı

6-Sprint Planlama ; Ekip , scrum master ve ürün sahibinin bir araya geldiği scrum toplantılarının ilkidir. Toplantıda ürün listesinin(ki bu liste öncelik sırasına göre listelenmiştir) en tepesinden başlayarak bakılır ve yapılacak sprintte bunlardan ne kadarının yapılabileceği tahmin edilir. Sprintler 1 aylık süreyi geçmeyecek şekilde planlanmalı. Sprint planlamada , ekip neleri tamamlayabileceklerine karar verdikleri anda nokta konulmalıdır.

7-İşin Görünür Kılınması ; Bunu sağlamak için 3 sütuna bölünmüş bir scrum tahtası hazırlanmalı “yapılacaklar”, ”devam edenler”, “tamamlananlar” . tamamlanacaklar sütununa yapılacak her iş kalemi bir yapışkan kağıtla eklenir. Sonrasında her bir kalemde ilerleme sağlanırsa tahta üzerinde o işe ait kağıt ilerletilir. İşin görünür kılmaya yarayan bir diğer yöntem BurnDown şemasıdır. Bir eksende ekibin sprinte dahil ettiği sayılar , diğerinde gün sayısı vardır. Scrum master her gün tamamlanan iş kalemlerini sayar ve onları şemaya işler. İdeali sprintin son gününde sıfır iş kalemi kalacak şekilde aşağı doğru dik eğimli bir çizgidir?

8- Günlük Scrum Önemlidir ; Her gün 15 dakikayı geçmemeye özen göstererek , ekip ve scrum ustasının bir araya geldiği ve aşağıdaki 3 adet soruya cevap arandığı mini toplantılardır.

- Ekibin sprinti tamamlaması için dün ne yaptın
- Ekibin sprinti tamamlaması için bugün ne yapacaksın
- Bireysel yada ekip açısından, sprint hedeflerine ulaşmanızı sağlayan engeller var mı?

Ekip otonomdur, kendi aralarında yapılan bir etkinliktir. Gecikme yada engellemelerin görünür kılınması ve kaldırılması açısından önemli bir görüşmedir. Bu toplantıların bir diğer önemli yanı , ekip üyelerinin sprintteki yerini tam olarak anlamalarını sağlamasıdır.

9- Sprintin Gözden Geçirilmesi ve Demo ; Sprint süresince ekibin neleri gerçekleştirdiğini gösteren toplantılardır. Herkes katılabilir, Scrum master, ekip, ürün sahibi dışında hissedarları, yönetim, müşteri. Bu toplantı ekibin sprint süresince tamamlananlar sütununa ne/neleri koyduğunu gösteren toplantıdır. Ekip biten işleri yani teslim edilecek işleri gösterir. Tamamlanmış bir ürün olmayabilir ancak bir ürünün tamamlanmış bir özelliğinin sunulmuş olması gerekir.

10-Sprint Geçmişine Bakış ; Ekip son sprint'le tamamlanan işleri gösterdikten sonra, bir sonraki sprinte hazırlık açısından , bir önceki sprintte nelerin iyi gittiğini , nelerin süreci aksattığını ortaya koyması ve bu bilgi birikimden olumlu sonuçlar çıkarıp sonraki sprintlerde daha etkin ve başarılı işlemler gerçekleştirme çabasını ortaya koymasısıdır.

Anahtar nokta bu gözden geçirme, sadece sürecin aksayan yanlarını görmek, bir fotoğraf almak ve perspektif geliştirmektir. Ana unsur budur. Kesinlikle suçlu aranmaz. Dolayısı ile bu toplantıların sağlıklı olabilmesi adına duygusal olgunluk ve güven atmosferi şarttır. Çözüm odaklı olmak önemlidir. Dolayısı ile bireysel olarak ekip içinde bizi rahatsız eden konuları mazeret olarak değil, çözülmesi gereken bir sorun olarak özgüven ve cesaretle ortaya koymak doğru olacaktır.

11- Ekibin engel ve aksaklıklarla olan tecrübe ve süreç gelişimlerini dikkate alarak, vakit kaybetmeden bir sonraki sprint döngüsüne başlanması gerekir.

BÖLÜM 3

UYGULAMAYI GELİŞTİRİRKEN KULLANILAN TEKNOLOJİLER

Uygulamamızı Visual Studio 2017 platformu üzerinde ASP.NET altyapısı ve MVC mimarisiyle geliştireceğiz. Asp.net kullanmamızın temel nedeni ekibe dahil olan bütün yazılımcıların bu konuda gerekli ve yeterli bilgi birikime sahip olmasıdır. Bunun yanı sıra dilin esnekliği, hazır kütüphaneler, editörün kullanışlı olması bizim için bir avantaj olmaktadır. MVC mimarisi kodlama ve tasarım olarak bize oldukça güçlü bir altyapı sunmaktadır. Kullanışlı SEO yapısı ile URL yönetim kolaylığı, HTML yapısına yüzde yüz hakim olma MVC tercih etmemizde oldukça etkili oldu.

Veritabanı olarak Microsoft SQL Server 2012 kullanacağız. Daha önceki tecrübelerimiz MSSQL sistemi üzerinde yoğunluk bulunduğu için tercihimizi bu yönde kullandık. Stabil olması ve detaylı kaynaklar bulunması tercihimizi etkileyen unsurlar olarak öne çıkmıştır.

Sistemi besleyen web servisler WCF mimarisi kullanılarak geliştirilmiştir. Dışa açılan servisler ise RESTFUL yapısında WEB API'ler olacaktır. Buradaki temel nokta bu sistemlerin dış alıcı ile XML ve JSON gibi genel geçer standartlar ile iletişim kurabilmesidir. Bir standart oluşturan bu servisler platform bağımsız birçok yapı ve programlama diliyle stabilize çalışmaktadır. Bu sistemler güçlü güvenlik imkanları sunmaktadır.

Uygulamamızı, servisleri ve veritabanları Microsoft Server 2012 üzerinde koştan IIS8 servislerinde host edeceğiz. Genel olarak microsoft ürünlerini tercih etmemizin nedeni; döküman erişimi, teknik destek ve bu uygulamalar için gerekli tecrübeye sahip olmamızdır.

Uygulama Altyapısı;

- ASP.NET MVC mimarisi
- Html5
- CSS3
- Bootstrap3
- AngularJS
- JQuery
- Ajax

Servisler;

- ASP.NET MVC mimarisi
- WCF
- WEP API

Veritabanı;

- MS SQL Server

Yardımcı Programlar;

- Photoshop
- Soap UI Service Test Application
- Postman Service Test Application
- Paint.NET 4.0.19
- <https://cacao.com/>
- <https://www.scrumdo.com/>

Uygulama altyapısında kullanılan HTML5 temel web programlama dilinin en son sürümüdür. Web tabanlı browser üzerinde koşan uygulamalar için standart bir dildir. CSS3 dili ise stil olarak web düzeni sayfalarının tasarımsal olarak geliştirilmesini sağlar ve en son css türüdür. Bootstrap3 ise css kütüphanesidir ve çok geniş bir düzenleme imkanı sunan hazır kod bloklarını yapısında barındırarak büyük bir avantajla tasarım geliştirmesi sağlar.

AngularJS, JQuery ve ajax betik dillerini gerekli yerlerde tasarımsal yada yazılımsal esneklik gerektiren noktalarda kullanıyoruz. Hazır kütüphaneler birçok yerde pratik çözümler sunarak elimizi oldukça kuvvetlendirmektedir.

Photoshop ve Paint.net tasarımsal olarak gerekli olan image ve iconları geliştirmemize olanak sağlayan grafik düzenleme yazılımlarıdır.

Soap ui ve postman servis testlerimizi gerçekleştirmemize olanak sağlayan , test programlarıdır.

Tez kapsamında eklenen birçok Şekil ve grafik cacao.com online tasarım modülünde geliştirilmiştir.

Scrum yapısını kullanırken , süreçleri dökümante ve otomatize edecek bir programa ihtiyacımız vardı. <https://www.scrumdo.com/> bir saas uygulaması olarak basit ara yüzü, kullanım kolaylığı ve belli oranda ücretsiz olması scrum için tercih sebebimiz oldu.

BÖLÜM 4

BULGULAR VE B4B KARMA SİSTEMİ

B2B'nin temelde şirketler arası pazarlama, satış uygulamalarını kapsayan süreci otomatize eden uygulama ve/veya uygulamalar bütünü olduğundan bahsetmiştik. Şirketler satış ile ilgili tüm operasyonlarının modellendiği bu uygulamalarla daha etkin bir satış ağı yönetimi oluşturacaklarından bahsetmiştik. Manuel sürecin getirdiği ağır yapı, operasyon akışlarındaki kesintiler bu uygulamalarla minimize edilir.

B4B ise B2B sistemlerinden farklı olarak şirketlere yeni bir dünya sunmaktadır. Artık şirketler için yeni pazarlama yöntemi sunan bir parametre olmaktadır. Sistemi kullanan müşteriler kendi mevcut tüketicilerinden farklı olarak yeni müşteri edinme imkanına sahip olacaklardır.

Bu çalışmamızda B2B ve B4B sistemlerinin karması melez bir uygulama üzerinde çalışacağız. B2B ve B4B uygulamalarının akışlarını içerisinde barındıran karma sistem bir SAAS uygulaması olarak çalışacaktır. Sistem tek bir yerden çalışacak ve firmalar tek uygulama üzerinden kendilerine ait accountlarla sistemi kullanabileceklerdir. Bu bölümün başlangıcında sistemi geliştirirken kullanacağımız yazılım geliştirme metodolojisinin uyarlaması ele alınacak sonrasında teknik detaylar ve sistemin kendisi anlatılacaktır.

4.1 Çevik Metodolojilerin Projede Uygulanması

Bu bölümde projemiz için seçtiğimiz çevik yaklaşım metodolojisinin uygulanma safhaları hakkında detay içeren bilgiler aktarılacaktır.

4.1.1 Neden Çevik Yaklaşım

Geliştireceğimiz projede çevik yaklaşım metodolojisini kullanmamızın temel sebeplerini, çevik yaklaşım manifestosundaki maddelerle açıklayacak olursak

4.1.1.1 Bireyler ve Etkileşimler, Süreç ve Araçların Üstündedir

Çevik yaklaşımda, bireyler arasındaki etkileşimin, projelerin gelişim aşamasına katkısı oldukça yüksektir. Bu madde özelinde bireylerin etkileşim ile gerek kendi içlerinde birbirlerinin yeterliliklerini yükseltmesi, gerek projenin tüm paydaşlar tarafından bilinir kılınması ve projeye tüm paydaşların katkısının sağlaması, eksikliklerin anında tespiti ve etkileşimde kolay çözümü projemizde olmasını beklediğimiz katkıyı bize vermesi açısından elzemdir.

4.1.1.2 Çalışan Program, Kapsamlı Dokümantasyonun Üstündedir

Karma sistemimizin kapsamı oldukça geniştir. Projeye başlamadan kapsamlı bir döküman hazırlama çalışmaları oldukça uzun süreceği gibi aynı zamanda tüm detayları önceden tahmin edebilme yeteneğine bağlıdır. Karma sistemi oluşturan tüm süreçler aslında yaşayan bir operasyondur ve bütün süreçler göz önüne alındığında çok kapsamlı, eksiksiz bir döküman hazırlamak oldukça zordur. Bunun yerine sistemimizi fazlara bölüp , her fazın alt iterasyonları gerçekleştikçe çalışan bir sistemin prod ortamda kullanıma alınması ve bu çalışan sistem üzerinden geri dönütler alıp, mevcut iterasyonun eksiklerini giderme ve bir sonraki faz için geçiş aşamasını planlamak çalışmamız açısından daha uygun olacaktır.

4.1.1.3 Müşteri İle İşbirliği, Sözleşmeler ve Anlaşmaların Üstündedir

Sistemimiz müşterilerin ortak kullandıkları bir saas platformudur. Tek bir müşteri isteği dışında birden fazla müşteri ile çalışma imkanımız olacak. Bu da farklı akışların uygulamamıza implementasyonunu sağlayacaktır.

4.1.1.4 Değişikliklere Yanıt Vermek, Bir Planı Takip Etmekten Daha Üstündür

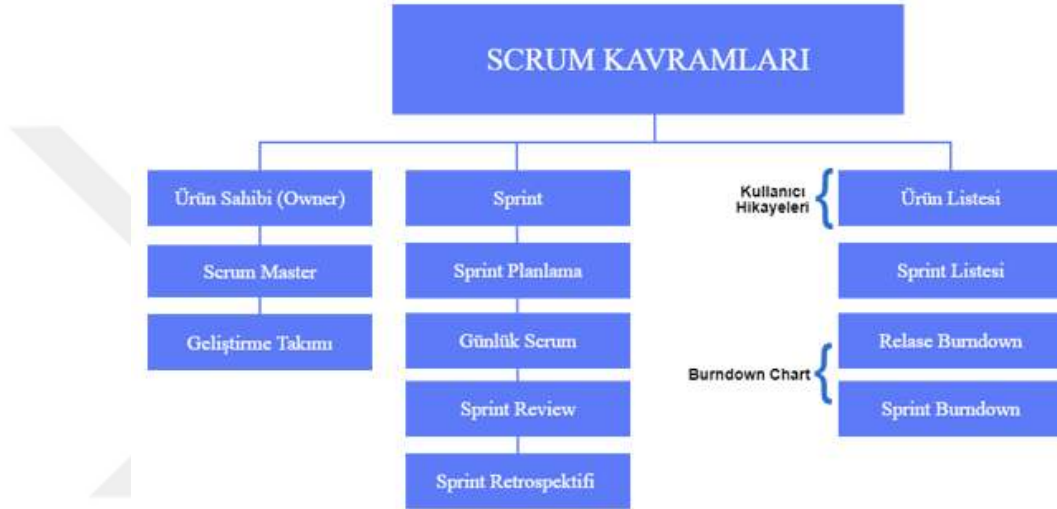
Ele alacağımız proje firmaların yaşayan sistemlerinin bir modellemesi gibi düşünülebilir. Tüm firmaların pazarlama faaliyetleri gelişim ve değişim içerisinde olacaktır. Bununla beraber gelişen teknolojinin bize sunduğu imkanlar değişimi kaçınılmaz kılacaktır. Bu sebeple müşterilerden gelecek değişiklik talepleri belli bir standartta düzenlenerek uygulanabilir. Yazılımsal ve altyapısal değişiklikler sürüm olarak sistemimize uyarlanır. Bu noktada bizim açımızdan önemli olan farklı müşterilerden gelen talepler veya var olan bir problem durumunun çözümü gerçekleştirildiğinde tüm müşterilere hızlıca uygulanabilir olması. Dikkat edilmesi gereken tüm talepler bir standarda dönüştürülerek sisteme uyarlanmalı.

Çevik yaklaşımın temel 4 değeri projemizde gerçekleştirmek istediğimiz tüm detayları bizim açımızdan görünür kılmaktadır. Tüm avantajları ile sürece oldukça katkı sunarak çok miktarda işi daha az efor ve daha sistematik bir düzenle gerçekleştiriyor olacağız.

Geliştirilecek projede temel bazda yapılacaklar dökümanite edilecektir. Genel bir çerçeve belirginleştirilecektir. Ancak projemizin son hali yani nihai çalışacak uygulama tüm bu çevik yaklaşım süresinde elde edilecek bilgiler ile derinleştirilecektir. Çevik yaklaşımı tercih etmemizin bir diğer sebebi olan bu durumda projemiz sahadan gelen dönütlerin de uyarlanmasıyla son halini alacaktır.

4.1.2 Çevik Yaklaşımın Projede Uygulanması

Çevik yaklaşım metodolojilerinden scrum kullanarak, çevik felsefe temelleriyle projemize Şekil vereceğiz. Bu bölümde projemizin scrum uyarlamasının bilgilerini paylaşacağız.



Şekil 4.1 Scrum Temel Kavramları Listesi

4.1.2.1 Scrum Metodolojisinin Projeye Uyarlanması

Scrum metodu projemizde uygulanırken öncelikli olarak ekibimiz oluşturulmuştur. Ekipte scrum kurallarına uygun olarak ürün sahibi, scrum master ve geliştirme takımı yer almaktadır. Yeterlilikleri yazılım mühendisi(2), tasarım uzmanı, analist-test uzmanı ve IT uzmanlığı olan bireylerden oluşan 5 kişilik geliştirme ekibimizde scrum süresince her bir üye başka bir ünvan ile anılmaz herkes ekibin bir üyesidir. Scrum takımımız oluştuktan sonra ürün sahibinin ürün ihtiyaçlarını baz alarak önceliklendirdiği ürün listesi üzerinde çalışarak kullanıcı hikayeleri ve temel ürün ekran prototipleri oluşturur.

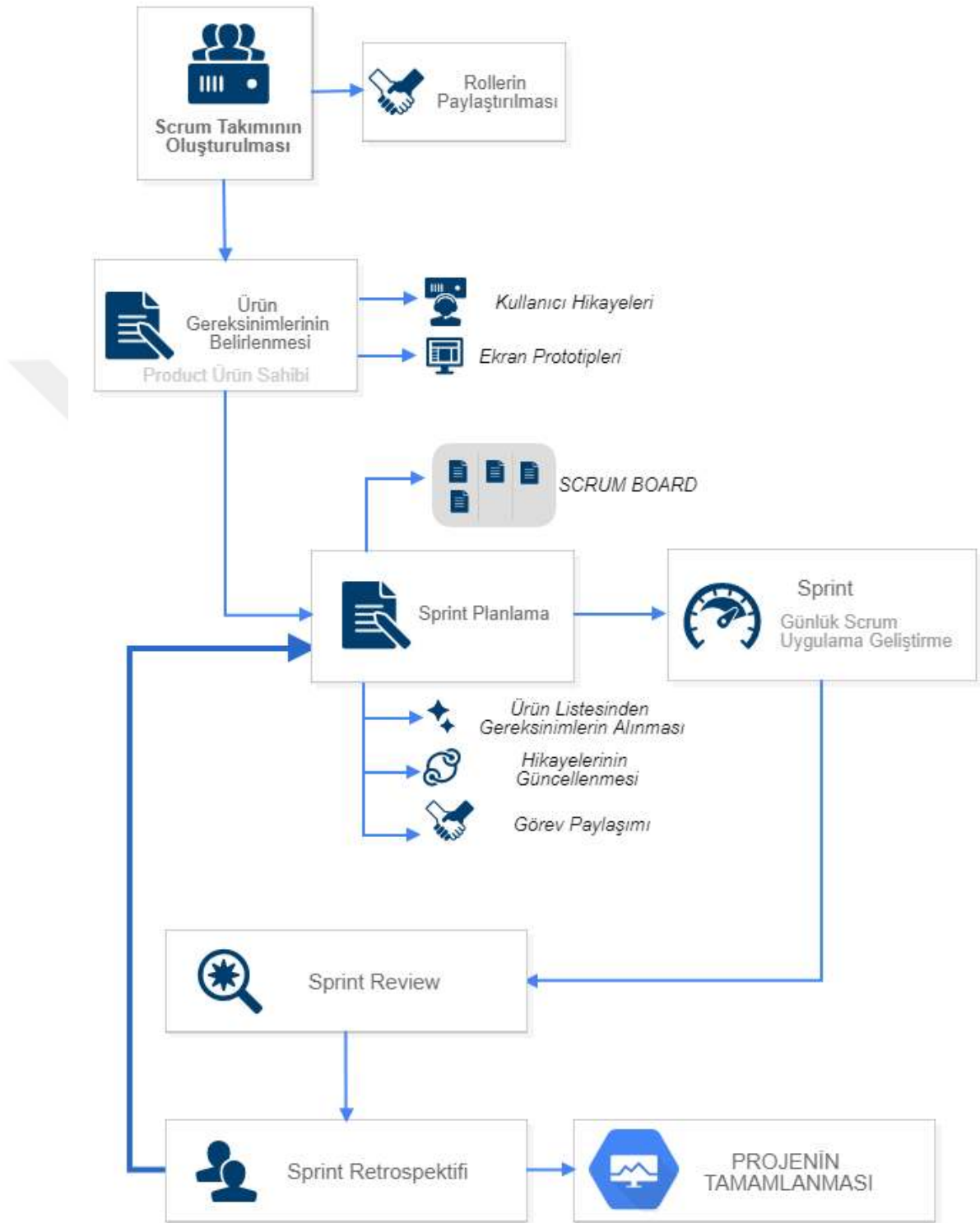
Sprint planlama aşamasında takım bir araya gelerek sprint planlama toplantıları gerçekleştirir. Ürün listesindeki öncelik sırasına göre en önemli maddeler belirlenerek sprinte dahil edilir önemli olan sprintin 2-4 hafta aralığında gerçekleştirilebilecek bir iş planı ortaya konabilmesi. Bu aşamada kullanıcı hikayeleri ekip tarafından güncellenerek görevler dağıtılır. Daha sonra scrum boarda oluşturulur. İş gereksinimleri boarda aktarılır.

Bu aşamadan sonra sprint başlatılmış olur. Günlük scrum toplantılarıyla geliştirmeler gerçekleştirilerek iterasyon(sprint) içerisinde yer alan gereksinimler gerçekleştirilir.

Sprint tamamlandığında takım sprint review aşamasında toplanarak ürünü ve süreci değerlendirir. Ürün sahibi ürüne ait sprintte yer alan sprint listesinin ne kadarının gerçekleştirildiğini görür, sprint sonunda sprintte tamamlanan ürün parçacığı çalışır vaziyette değerlendirilir.

Bir sonraki aşamada Sprint Retrospektifi ile birlikte ekip kendini değerlendirir, bir önceki sprintte karşılaşılan olumlu ve olumsuz tüm değerlendirmeler , bir sonraki sprinte olumlu referans olarak aktarılacak şekilde değerlendirilir. Bu aşamadan sonra artık yeni sprint için sprint planlamaya geçilebilir.

Tüm sprintler, fazlar gerçekleştirildiyse proje tamamlanacak ve prodada alınacaktır.



Şekil 4.2 Scrum Metodunun Proje Uyarlama Şeması

4.1.2.2 Scrum Kavramları

Scrum metodolojileri uyarlanması ile ilgili bilgiler bölüm daha önce özet olarak aktarılmıştı. Uyarlanan projeyi daha netleştirmek için kavramları teker teker ele alarak detaylandırıp, scrum planlarını control management üzerinde projelendireceğiz. Management için online bir tool scrumdo kullanacağız.

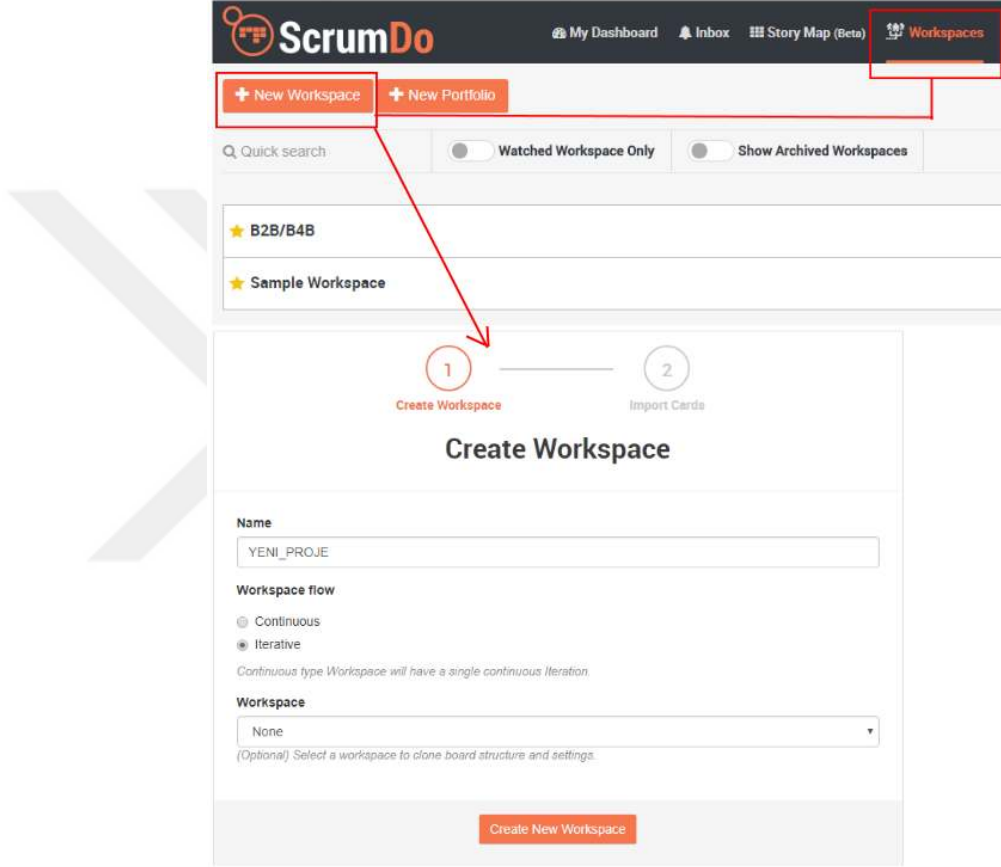
ScrumDo Tools;

ScrumDo tools internet üzerinde çalışmakta ve herhangi bir kurulum gerektirmemektedir. <https://www.scrumdo.com/> adresinden erişim sağlanan tool basit bir registerdan sonra kullanıma hazır hale gelecektir.

The image shows a web browser window displaying the ScrumDo sign-up page. The browser's address bar shows a secure connection (Güvenli) to the URL https://app.scrumdo.com/account/signup/. The page features the ScrumDo logo, which consists of an orange circle with a grid pattern inside, followed by the text "ScrumDo" in a bold, sans-serif font. Below the logo, the heading "Sign Up" is centered, followed by the instruction "Please enter your email address to get started in ScrumDo." A text input field labeled "Email Address" is provided for the user to enter their email. Below the input field is an orange button labeled "Create Account". At the bottom of the page, a line of text reads "Free Trial, Cancel Any Time, No Credit Card Needed".

Şekil 4.3 ScrumDo Kullanıcı Kayıt Ekranı

ScrumDo Tools, uygulamasına başlarken Workspace alanı belirlenir. Workplace alanı tüm scrum akışlarını yöneteceğimiz bir proje olarak düşünebiliriz workspace tanımlanmadan projeyi başlatamayız.



Şekil 4.4 ScrumDo Workspace Oluşturma

4.1.2.3 Ürün Sahibi

Daha önce B2B sistemlerinin uygulamalarını sahada yönetmiş bir arkadaşımızı seçtik. Bir çok e-ticaret ve B2B sistemlerinin uyarlamasında görev alması tüm saha ihtiyaçlarımızı karşılayacaktır.

4.1.2.4 Scrum Master

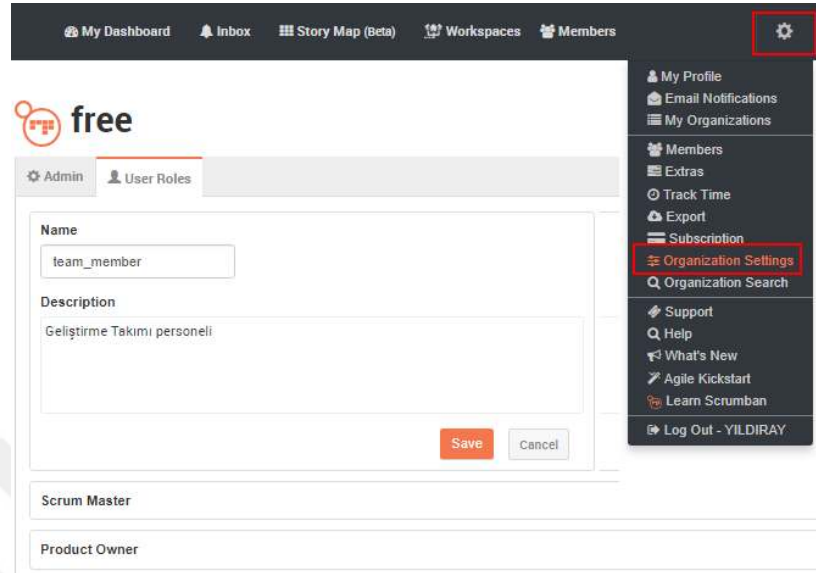
Scrum master vizyonunu projeye uygulama görevini kendim üstlendim ve toolda bu görevin atamasını bu doğrultuda gerçekleştirdik.

4.1.2.5 Geliştirme Takımı

Geliştirme takımımız ürün sahibi ve scrum master hariç 5 kişiden oluşmaktadır

- Yazılım mühendisi(2)
- Tasarım uzmanı
- Test ve analiz uzmanı
- IT uzmanı

ScrumDo da workspace tanımlamasından sonra kullanıcı rolleri için düzenleme yapıp scrum takımını oluşturacağız.



Şekil 4.5 ScrumDo Rollerin Oluşturulması

Şekil 4.5 de görüldüğü üzere organization settings yönlendirmesiyle açılan ekranda add user butonu ile yeni kullanıcı rolü ekliyoruz.



Şekil 4.6 ScrumDo Scrum Takımı

Geliştirme takımımız(Team), müşterimiz(Customer) ve team içinde yer alan scrum masterımızla birlikte scrum rollerimizi tamamlanmış bulunuyoruz.

Organization Members		
	Gonca Ayaz @Gonca <syazgonca@gmail.com>	team_member
	@huseyin.syz25@gmail.com <huseyin.syz25@gmail.com>	team_member
	sadik tek @teksadik <teksadik@gmail.com>	team_member
	tek yaz @Tknor <yazilim.teknoser@gmail.com>	team_member
	Burak Yildirim @mamoste <burakyildirim545@gmail.com>	team_member
Organization Customers		
	urun sahibi @ProductOwnerMine <yildirayyildirim87@gmail.com>	
Organization Staff		
	urun sahibi @ProductOwnerMine <yildirayyildirim87@gmail.com>	Product Owner
	YILDIRAY YILDIRIM @YILDIRAY <cambazyilo@gmail.com>	Scrum Master

Şekil 4.7 ScrumDo Scrum Takımı Detayları ve Roller

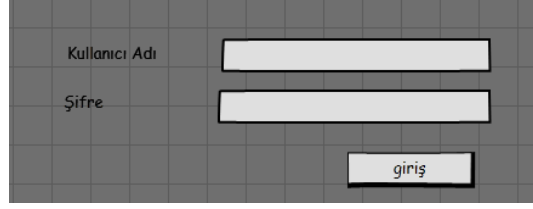
4.1.2.6 Ürün Listesinin Oluşturulması (Product Backlog)

Scrum Takımını oluşturup roller belirlendikten sonra ürün sahibinin önceden analiz edip belirlediği ürün listesi kalemleri üzerinde çalışır. Ürün sahibi gerekliliklerin tamamını net ve açık olarak ifade etmeli. Bu kalemler öncelik listesine göre sıralanmıştır ve genel olarak hikaye haline getirilmiştir.(beklentiler listelenmiştir) Daha sonra bu kalemler sprint planlama toplantılarında öncelik sırasına göre sprinte alınacak ve hikayeler karşılıklı görüş alış veriş i ile güncellenecektir. Aşağıda tablolarda ürün listelerimizi vereceğiz.

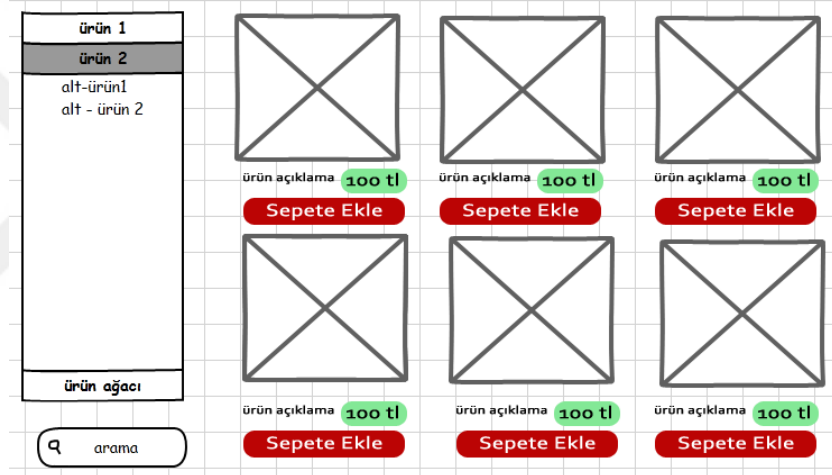
B2B SİSTEM ÜRÜN İŞ LİSTESİ				
Numara		Ürün Listesi Maddeleri	Hikayeler	öncelik
1	A	Database serverların, geliştirme ortamlarının, kullanıcı yapılandırmalarının yapılması yetki ve kısıtlamaların verilmesini içeren It mimarisinin geliştirilmesi.	Tüm it sistemlerinin ekip ve bizim tarafımızdan üzerinde çalışılabilir hale getirilmesini istiyorum	1
2	A	Database oluşturulması, gerekli olan basic, must, transaction ve log tablolarının geliştirilmesi.	Tüm database operasyonlarının yapılmasını istiyorum manuel veri girişi bile yapılabilir?	1
3	A	Tasarım modellemesi (site tasarımı)	Sitenin tasarlanmasını istiyorum tasarımcılar belli aralarla bizimle resimleri paylaşabilirler?	1
4	A	Tasarım modelinin arayüze dönüştürülmesi , temel arayüz geliştirilmesi	Tasarımların artık web browserda görüntülenebilir bir HTML yapıya bürünmesini istiyorum	2
5	F	Yönetici ekranı -Temel Ayarların Yapılmasına izin veren ekran	Yönetici ekranları yapılınsın ve bir yandan da biz kullanıcı, ürün ekleme, kategori düzenleme işlemlerimizi yapalım istiyorum	3
6	F	Kullanıcı ekleme ve silme modülü - yetki yapılandırması modülü -kullanıcı güncelleme modülü	//	3
7	F	Kategori Yönetimi Modülü	//	3
8	F	Ürün Yönetim modülü (ekleme/silme/Güncelleme)	//	3
9	O	login işlemleri - şifre değiştir - şifremi unuttum	Kullanıcıların sisteme giriş yapması için ilk olarak login işlemlerinin yapılmasını elzem görüyorum	4
10	U	Menu Geliştirmeleri	Menülerin nasıl yansıdığını görmek istiyorum	5
11	U	Karşılama ekranı düzeninin yapılandırılması, ürün listeleme modülü	Ana sayfadaki ürün listeleri ve öne çıkanların(ürün) nasıl görüneceğini test etmeliyiz	5
12	U	Arama Modülü	ürün araması yapılabilirmeli.	5
13	U	Ürün detayı ekranı ve sipariş verme	Ürünlere tıklanınca detayının ekrana gelmesi gerekir. Ekranı çok yormadan gerekli tüm detayları verelim	6
14	U	Sepet Uygulaması	Kullanıcılar sipariş olarak sepete eklediği ürünleri sepetinden tekrar kontrol ediyor olmalı, sepet detayı önemli	6
15	U	Sipariş Detay ve Takibi	Verilen siparişin durumunu müşteriler takip edebilmeli	6
16	U	Faturalarım ve Ödeme	Sipariş sonrası firmalara kesilen faturlar ve bu faturaların ödeme ekranları, ayrıca müşteri eft havale seçenekleriyle de ödemeli	7
17	U	Ayarlarım	İlk etapta temel olarak , şifre değiştir , kişisel bilgileri değiştir şeklinde sınırlı bir veri yenileme alanları oluşturabiliriz.	7
18	F	Yönetici ekranı -Siparişler ve Teslimatlar	Siteyi kullanan satıcıların sipariş takibi yaptığı ekranlar liste ve detay şeklinde olabilir, popup kullanılabilir	8
19	F	Yönetici ekranı -Faturalar ve Ön Muhasebe	Firmaların siparişlerine gelen ödemeleri gördüğü ekranlar yapılmalı, kullanıcı rapor alabilmeli	8
20	F	Yönetici ekranı -Bilgilendirme ve mesaj sistemi	Genel olarak sms ve mail bilgilendirmeleri için otomatik talimatların girildiği sayfalar	9
21	F	Yönetici ekranı -Otomatik İş Listeleri	Otomatize edilmiş iş listeleri, ürün ekleme , kaldırma, fatura hatırlatma vb (sprintte detaylandırılabilir)	9
22	A	Entegrasyon Apileri	ERP ve CRMler için Api desteği	10
*** A[site geliştirme için temel işlem] *** F[Firmaların Kullandığı Ekranlar] *** U[Userların Kullandığı Ekranlar] *** O[ortak Kullanılan Ekranlar]				

Tablo 4.1 Ürün İş Listesi (Product Backlog)

Ürün iş listesi ile birlikte ekran prototiplerini de beklentilerimizi ekibin anlayacağı düzeyde basit toollarla ifade edebilecek çizimler iletebiliriz.



Şekil 4.8 Ekran Prototipi Örneği (Giriş Sayfası)



Şekil 4.9 Ekran Prototipi Örneği (Ürün Listesi)

4.1.2.7 Sprint Listesi (Product Backlog)

Sprint iş listesi günlük scrum faaliyetlerinin gerçekleşmesi ve iterasyonun nihai hedefe ulaşmasında sprint için seçilen ürün iş listesi kalemlerini içerir. Bu listenin detayları sprint planlama toplantısında netleştirilir.

4.1.2.8 Sprint Planlama ve Sprint

Sprintin, maximum bir aylık dilimi kapsayacak şekilde, scrum takımının ürün iş listesinden öncelik sırasına göre seçtiği iş kalemlerini gerçekleştirmeye yönelik faaliyetler olduğundan bahsettik.

Scrum takımımız bir araya gelerek, ürün listesinden önceliklere göre belirlediği işleri sprintlere ayırmıştır. Bu sprint düzenlemesinde grubumuzun yetenekleri ve yapabileceği iş kapasitesi dikkate alınmıştır.

B2B SİSTEM ÜRÜN İŞ LİSTESİ - SPRINT BAZLI			
Numara	Sprintler	Ürün Listesi Maddeleri	Hikayeler
1	<i>Sprint 1</i>	Database serverların, geliştirme ortamlarının, kullanıcı yapılandırmalarının yapılması yetki ve kısıtlamaların verilmesini içeren İt mimarisinin geliştirilmesi.	Tüm it sistemlerinin ekip ve bizim tarafımızdan üzerinde çalışılabilir hale getirilmesini istiyorum
2	<i>Sprint2</i>	Database oluşturulması, gerekli olan basic, must, transaction ve log tablolarının geliştirilmesi.	Tüm database operasyonlarının yapılmasını istiyorum manuel veri girişi bile yapılabilir?
3		Tasarım modellemesi (site tasarımı)	Sitenin tasarlanmasını istiyorum tasarımcılar belli aralarla bizimle resimleri paylaşabilirler?
4		Tasarım modelinin arayüze dönüştürülmesi , temel arayüz geliştirilmesi	Tasarımların artık web browserda görüntülenebilir bir HTML yapıya dönüştürülmesini istiyorum
5	<i>sprint 3</i>	Yönetici ekranı - Temel Ayarların Yapılmasına izin veren ekran	Yönetici ekranları yapılınsın ve bir yandan da biz kullanıcı, ürün ekleme, kategori düzenleme işlemlerimizi yapalım istiyorum
6		Kullanıcı ekleme ve silme modülü - yetki yapılandırması modülü - kullanıcı güncelleme modülü	//
7	<i>sprint 4</i>	Kategori Yönetimi Modülü	//
8		Ürün Yönetim modülü (ekleme/silme/Güncelleme)	//
9	<i>sprint 5</i>	login işlemleri - şifre değiştir - şifremi unuttum	Kullanıcıların sisteme giriş yapması için ilk olarak login işlemlerinin yapılmasını elzem görüyorum
10		Menu Geliştirmeleri	Menülerin nasıl yansıdığını görmek istiyorum
11		Karşılama ekranı düzeninin yapılandırılması, ürün listeleme modülü	Ana sayfadaki ürün listeleri ve öne çıkanların(ürün) nasıl görüneceğini test etmeliyiz
12		Arama Modülü	ürün arama yapılabilmeli.
13	<i>sprint6</i>	Ürün detayı ekranı ve sipariş verme	Ürünlere tıklanınca detayının ekrana gelmesi gerekir. Ekranı çok yormadan gerekli tüm detayları verelim
14		Sepet Uygulaması	Kullanıcılar sipariş olarak sepete eklediği ürünleri sepetinden tekrar kontrol ediyor olmalı, sepet detayı önemli
15		Sipariş Detay ve Takibi	Verilen siparişin durumunu müşteriler takip edebilmeli
16		Faturaların ve Ödeme	Sipariş sonrası firmalara kesilen faturalar ve bu faturaların ödeme ekranları, ayrıca müşteri eft havale seçenekleriyle de ödemeli
17	<i>sprint 7</i>	Yönetici ekranı -Siparişler ve Teslimatlar	Siteyi kullanan satıcıların sipariş takibi yaptığı ekranlar liste ve detay şeklinde olabilir, popup kullanılabilir
19		Yönetici ekranı -Faturalar ve Ön Muhasebe	Firmaların siparişlerine gelen ödemeleri gördüğü ekranlar yapılmalı, kullanıcı rapor alabilmeli
18	<i>sprint 8</i>	Ayarların	İlk etapta temel olarak , şifre değiştir , kişisel bilgileri değiştir şeklinde sınırlı bir veri yenileme alanları oluşturabiliriz.
20		Yönetici ekranı -Bilgilendirme , mesaj ve Otomatik iş Listeleri sistemi	Genel olarak sms, mail ve tekrarlanan iş akışları için otomatik talimatların girildiği sayfalar
21	<i>sprint 9</i>	Entegrasyon Apileri	ERP ve CRMler için Api desteği

Tablo 4.2 Sprintlere Ayrılmış Ürün İş Listesi

Tablo 4.2 projemizde sprint planlama ile ürün iş listesinin sprintlere bölünmüş listesini içermektedir. Her bir sprint 2-4 hafta sürecek bir planlamayı içermektedir. Sprint Planlama, bir aylık Sprint için 8 saatle sınırlıdır. Ürün olarak ne teslim edilecek ve bu ürün nasıl yapılacak cevaplanması gereken bu iki soru bu toplantıların esas temellerini oluşturur. Yeni ihtiyaçlar veya istekler ile sprint sonunda beklenen ürün belirginleşir. Her sprint için ürün iş listesinde yer alan kalemler detayları ile ele alınır ve alt maddeler belirlenebilir. Bu aşama ise ürün listesinden seçilen iş kalemleri alt iş kalemlerinin de eklenmesiyle sprint listesi oluşturulur.

Projemizden bir örnek sprint oluşturma süreci örneklemini ele alacak olursak;

<i>sprint 4</i>	Kategori Yönetimi Modülü	
	Ürün Yönetim modülü (ekleme/silme/Güncelleme)	

Tablo 4.3 Sprint 4 İçin Ayrılmış Ürün İş Listesi

Tablo 4.3 de yer alan sprint 4 için ayrılmış iş kalemlerini görmekteyiz. Bu sprintte firmaların kendi kategori sistemlerini girmelerini ve sonrasında ürün ekleyerek girilen ürünlerin bu kategori ile eşleşmesini sağlamak hedefimizdir. Ürün sahibi tarafından iletilen iş kalemleri için sprint planlama toplantılarında güncellenen hikayeler;

Hikaye 1: *Kategori Yönetimi Modülü, ürün ağacının gerektirdiği kategorileri sisteme aktarabilmek istiyoruz. Ekleme, silme ve güncelleme işlemleri yapabilmeliyiz*

Hikaye 2: *Ürün yönetim modülü, Yeni ürün ekleyebilmek, silmek ve güncellemek istiyoruz. Ürün eklenirken fiyat, ürün resmi açıklama gibi detaylar muhakkak olmalı*

Ürün olarak ne teslim edilecek? Bu sorunun cevabı hikayenin içinde yer almakta ürün kategori işlemleri ve ürün işlemlerini yapabileceğimiz birer ekran. Ekip hikayeler üzerinden istenilen ürünü daha da netleştirmek için karşılıklı etkileşim ile ürün sahibine soru sorabilir.

- **Kategori için belirli bir sayı var mıdır? Sınırsız kategori ekleyebilmeli mi?**
- *Hayır Maximum alt kategori oluşturabilir?*
- **Kategori güncelleme ile güncellenecek alanlar?**
- *Kategori yalnızca kategori adı güncellenmesi doğru olur.*
- **Kategori silinirken kategorinin ilişkili ürünleri varsa silmeye izin verilecek mi?**
- *Kategori silinirken ilişkili ürün varsa uyarı verelim, ayrıca kategori silinirken alt kategorisi varsa yine uyarı vermeliyiz.*
- **Ürün eklerken kategori seçmek zorunlu olacak mı?**
- *Kesinlikle, bir ürün eklenecekse kategori muhakkak seçilmeli*
- **Ürün eklenirken hangi detaylar kullanıcıdan istenecek?**
- *Ürün Kodu*
- *Ürün Adı*
- *Ürün açıklaması*
- *Birim fiyat*
- *Ürün resim*

Karşılıklı etkileşim ile detayları ortaya çıkan sprintte gerçekleştirilmesi beklenenler son halini bu Şekilde alır.

sprint 4	Kategori Yönetimi Modülü	
	Maximum 5 adet alt kırılım olacak şekilde kategori ağacı oluşturulabilecek, sililecek ve güncelleyebilecek ekran tasarımı	
	Database tablolarının oluşturulması	
	DB bağlantıları ve verilerin DB'ye işlenmesi	
	gerekli sınıf ve kod bloklarının yazılması	
	Ürün Yönetim modülü (ekleme/silme/Güncelleme)	
	Ürünlerin Ekleneceği/silineceği ve güncelleneceği ekran tasarımları (- Ürün Kodu - Ürün Adı - Ürün açıklaması - Birim fiyat - Ürün resmi)	
	Database tablolarının oluşturulması	
	DB bağlantıları ve verilerin DB'ye işlenmesi	
	gerekli sınıf ve kod bloklarının yazılması	

Tablo 4.4 Sprint İş Listesi “Yapılması Gereken Görevler Belirtilmiştir”

İş listelerinin oluşmasından sonra geliştirme ekibi bu maddelerin büyüklüklerini belirlerler. Bunun için uygulanan yöntemlerden biride poker planlama oyunudur. Ürün sahibi iş maddesini açıkladıktan ve sorular sorulup iş maddesi herkes tarafından anlaşılır ve net hale geldikten sonra geliştirme ekibinden herkes bu maddeye elindeki kartlardan birini seçerek bir hikaye puanı verir. Kartlardaki puanlar “0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?” değerlerinden birin olabilir. En yüksek ve en düşük puan verenler nedenlerini birbirlerine aktararak birbirlerini ikna yoluna giderler, temel amaç orta nokta bulup iş maddesinin büyüklüğüne ekipçe bir karar vermektir.

Sprint listesi maddesi: Maximum 5 adet alt kırılım olacak şekilde kategori ağacı oluşturulabilecek, sililecek ve güncelleyebilecek ekran tasarımı

Yukarıdaki madde için gelen en düşük değer 5 en yüksek değer 13 iki puan sahibi ekip üyesi arasında karar vermeyi kolaylaştıracak şöyle bir konuşma geçmesi beklenir.

- *X: Bence kategori ekranları alt kırılımlı yapı olduğu için iş yükü olarak geniş bir kurguyu gerektirir. bunun için 13 puan verdim*
- *Y: Anlıyorum ancak benzer uygulamaları daha önce yaptık, yeni kurgu için çok düşünmemize gerek olmadığı kanaatindeyim, bunun için 5 puan verdim*
- *X: Haklısın yeterli tecrübemiz var hızlıca geçebiliriz. O zaman 8 diyelim makul gibi görünüyor*
- *Y: Tamam anlaştık.*

Bu Şekilde ekip bir iki dakikayı geçmeyecek Şekilde tartışarak bir büyüklük göstergesinde anlaşır. Eğer anlaşılamazsa tekrar oynamaya devam edilir. Son aşamada da ortak bir karar alınamaz ise ya madde de anlaşılmayan noktalar vardır yada iş maddesi yeterince küçük birimlere ayrılmamıştır.

Bu yöntemle tek tek maddelerin büyüklükleri belirlendikten sonra sprintin toplam yükü(hızı) bulunacaktır. Sprint planlama toplantısı sonucu bir sprinte ait büyük resmi tüm detaylarıyla görebiliyor olmamız gerekir.

	Sprint İş Listesi	Hikaye Puanı
sprint 4	Kategori Yönetimi Modülü Görevleri	
	<i>Maximum 5 adet alt kırılımı olacak şekilde kategori ağacı oluşturulabilecek, silenecek ve güncelleyebilecek ekran tasarımı</i>	13
	<i>Database tablolarının oluşturulması</i>	8
	<i>DB bağlantıları ve verilerin DB ye işlenmesi gerekli sınıf ve kod bloklarının yazılması</i>	40
	Ürün Yönetim modülü Görevleri (ekleme/silme/Güncelleme)	
	<i>Ürünlerin Ekleneceği/sileneceği ve güncelleneceği ekran tasarımları (- Ürün Kodu - Ürün Adı - Ürün açıklaması - Birim fiyat - Ürün resim)</i>	13
	<i>Database tablolarının oluşturulması</i>	5
	<i>DB bağlantıları ve verilerin DB ye işlenmesi gerekli sınıf ve kod bloklarının yazılması</i>	40
	toplam sprint Puanı(Hızı)	119

Tablo 4.5 Sprint Dökümanı

Sprint Listesindeki her bir görev bir kullanıcı hikayesi biçiminde scrum tahtası (scrum kanban) üzerinde story bloguna yerleştirilir.

Sprint :	4	Öncelik :	3	Hikaye Puanı :	40
Ürün İş Listesi	<i>Kategori Yönetimi Modülü</i>				
Sprint İş Listesi Kalemi (Görev):	<i>DB bağlantıları ve verilerin DB ye işlenmesi gerekli sınıf ve kod bloklarının yazılması</i>				
Açıklama :	<i>kategori ekleme, silme ve güncelleme ekranlarının kod bloklarını yazacağız, başlamadan önce tablo stabilitesi kontrol edilmeli</i>				
Test Senaryosu :	<i>insert işlemi için injection kontrolü</i>				
	<i>insert işleminde validation kontrolleri</i>				
	<i>kayıt eklemede sorun olmadığından emin olun Db kontrolü yapın</i>				
	<i>güncelleme işlemi için kaydı kontrol edin Db kontrolü yapın</i>				
	<i>silme işlemi sırasında silinen kategoriye ait alt kategori yada ürün listesi varsa uyarı verilmesi</i>				

Tablo 4.6 Örnek Bir Kullanıcı Hikayesi

Tüm bu scrum planlama neticesinde oluşan sprint listeleri iş kalemlerini yani görevleri birer kullanıcı hikayesi formatında scrum tahtası (scrum kanban) üzerine yerleştirebiliriz. Scrum tahtasını projemizde 5 Parçaya ayırdık.

- Stories : Hikayeler görevler yer alır
- To Do : Yapılacak görevler
- Doing : Yapılmakta olan görevler
- Reviewing / Testing : Gözden geçirme test etme
- Done : Tamamlanan görevler

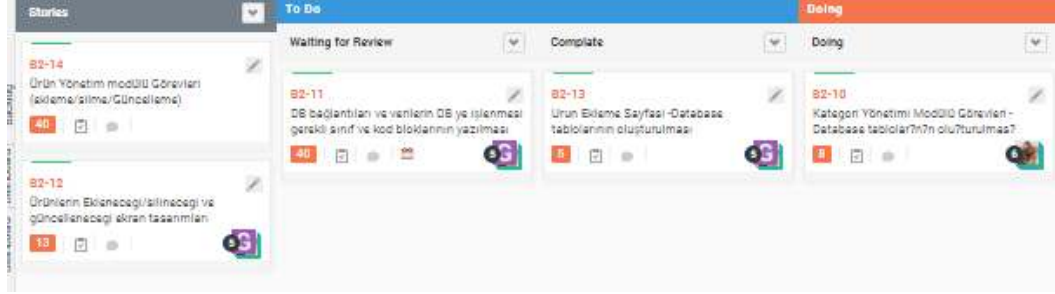
Her görev ilgili alanda işlemi bitince bir sütun yan tarafa kaydırılır. Şimdi ScrumDo tolls üzerinde bu işlemi gerçekleştireceğiz

Şekil 4.10 ScrumDo Üzerinde Story Tanımlama

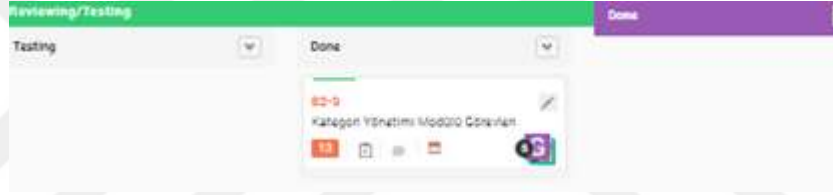
Şekil 4.10 de görüldüğü gibi kullanıcı hikayelerini sisteme giriyoruz , sisteme girerken hikaye puanlamasını yapıyoruz.

Şekil 4.11 ScrumDo Üzerinde Sprint Tanımlama

Sprint tanımlamalarından sonra , ilgili kullanıcı hikayelerini sürükle-bırak yöntemiyle 5 sutundan birine yerleştiriyoruz.

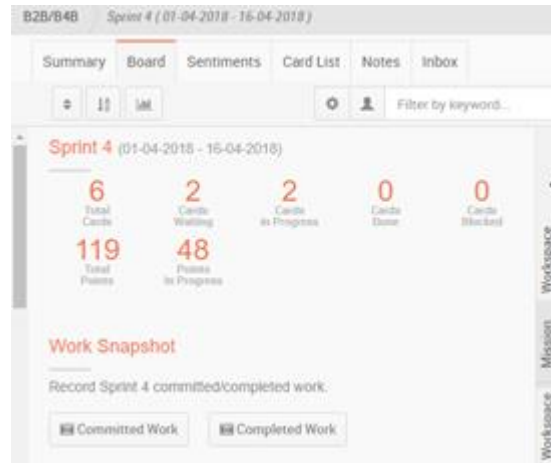


Şekil 4.12 ScrumDo Üzerinde Scrum Board Görüntüsü -1



Şekil 4.13 ScrumDo Üzerinde Scrum Board Görüntüsü -2

Tüm Planlamanın ardından Sprint 4 için görüntülenebilir istatistik verisi;



Şekil 4.14 ScrumDo Üzerinde Sprint Detayları

4.1.2.9 Günlük Scrum

Günlük scrum işleyişinde 24 saatte bir olmak koşulu ile her gün aynı saatte 15 dakikayı geçmeyecek şekilde ufak bir toplantı yapılır. Toplantıda takım üyelerine 3 adet soru sorulur.

- Dün ne yaptım?
- Bugün ne yapacağım?
- Önümde engel varmı?

4.1.2.10 Sprint Review

Sprint etkinliklerinin sonunda gerçekleştirilen ürün parçacığının ürün sahibine gösterilmesiyle birlikte sprint süresince isteklerin ne kadar karşılandığının değerlendirildiği , gerek görülürse düzeltmelerin yapıldığı toplantılardır. 4 haftalık bir sprint için 4 saati geçmeyecek toplantılar düzenlenebilir. Bu değerlendirme toplantılarının sahibi “Ürün Sahibidir”. Önemli olan bir diğer detay ürün sahibi sprint sonunda sunulan ürünü , önceden tanımladığı sprintin başarı kriterine yaklaşp yaklaşmadığının yani kendi açısından “Bitti” tanımına uyup uymadığını değerlendirmesidir.

Sprint 4 toplantısı sonucunda ürün sahibi tarafından gerçekleştirdiğimiz modüllere ilişkin olarak aldığımız geri bildirim:

“ürün eklenirken dosya boyutu için sınır verilmemiş, bu uyarıyı vermeliyiz.”

“Çoklu ürün silme eklenebilir”

4.1.2.11 Sprint Retrospektifi

Sprint Retrospektif etkinliđinin amacı Scrum Takımı'nın gemiř Sprint'lere bakarak önündeki Sprint'lerde yapabileceđi iyileřtirmeleri belirlemesidir. Sprint Retrospektif, 4 haftalık Sprint'lerde 3 saat, 2 haftalık Sprint'lerde 1.5 saatle sınırlıdır. Sprint retrospektif etkinliđine, ürün sahibi, geliştirme takımı ve scrum masterın katılması zorunludur. scrum takımı, iletişim, araç, süreç, iş yapış Şekil, çalıştığı ortam gibi scrum takımı'nı etkileyen farklı faktörleri göz önünde bulundurarak iyileřtirme aksiyonları belirlemeye ve bu aksiyonları hayata geçirebilmek için planlama yapmaya çalışır.

Sprint 4 için yapılan Sprint Retrospektif toplantısında

- Ön kabul testlerine dikkat edilmesi
- Etkileřimin daha da artması
- Süre açısından daha dikkatli olunması

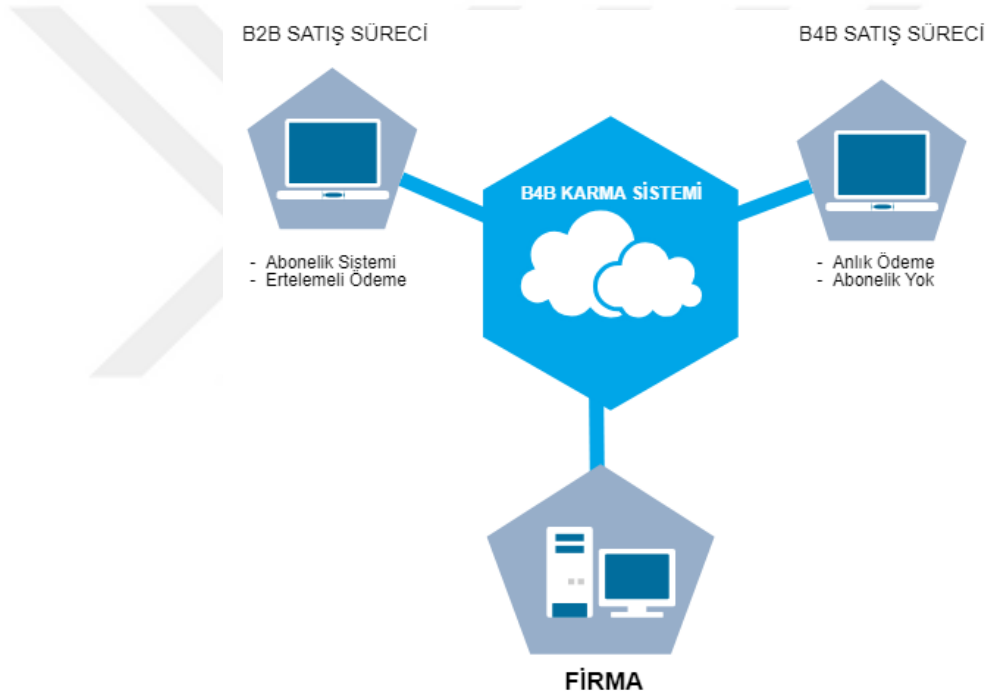
Konularına öncelik verilmesi kararlařtırılmıřtır.

4.1.2.12 BurnDown Chat

Sprint Retrospektif etkinliđinin amacı Scrum Takımı'nın gemiř Sprint'lere bakarak önündeki Sprint'lerde yapabileceđi iyileřtirmeleri belirlemesidir. Sprint Retrospektif, 4 haftalık Sprint'lerde 3 saat, 2 haftalık Sprint'lerde 1.5 saatle sınırlıdır.

4.2 B4B Karma Sistemi SAAS Uygulaması

Karma sistemimizin, tek bir uygulama üzerinden çalışan bir saas uygulaması olduğundan ve B2B - B4B sistemlerinin iş akışlarını içerisinde barındırdığından bahsetmiştik. Kullanıcılar sistemde iki farklı pazarlama mantığının operasyonel süreçlerini birarada yönetebilecekler. Ürünlerini ister abonelerine isterlerse de anlık ödeme ile kendilerine abone olmayan kullanıcılara satabilirler.



Şekil 4.15 B4B Karma Sistemi Seması Genel Görünüm

B4B Karma Sistemi Temelde;

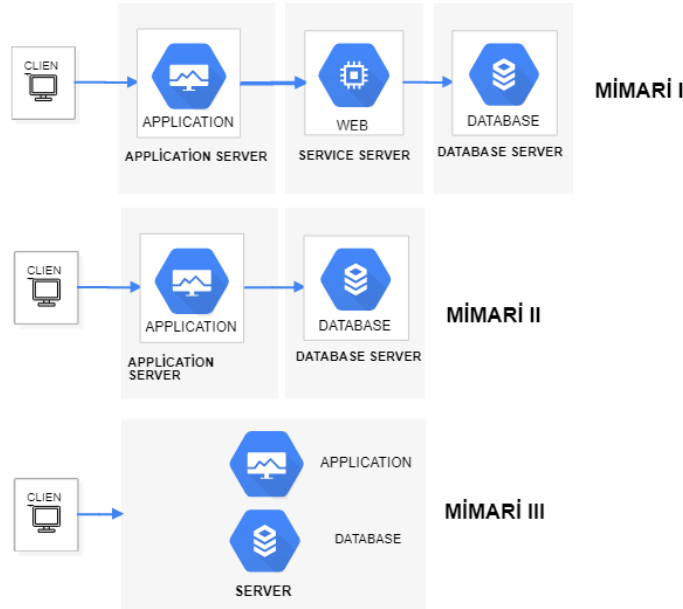
- İsteddiği ürünleri bir yetkiyle sadece abonelerinin satış siparişi geçmelerine izin verdikleri iş akışlarının yürütülmesi ile B2B operasyonlarını

- Yetkiyle genel satışa çıkardıkları ürünleri, ödeme şartıyla aboneli olmayan salt kullanıcılara direkt satışı ile de B4B operasyonlarını ifade eder.

B4B operasyonlarında kullanıcı ürünlerini kendi belirttiği şartlarda satışa açabilir yada kapayabilir. Bu koşullar renk, beden, adet, seri bazlı ürün grubu gibi operasyonel yetkileri kapsayabilir. Bu yapıda(B4B) kullanıcıların sipariş verdiği satıcıların daha önceden müşterisi olma gerekliliği bulunmamaktadır.

4.2.1 B4B Karma Sistemi Mimari Yapısı

B4B karma sistemi mimari yapısı B2B ve B4B sistemlerinin bir birleşimi baz alınarak oluşturulmuştur. Türkiye’de firmaların kullanmış olduğu B2B sistemleri genellikle kendi yönettikleri serverlarda konuşlandırılmıştır. Tüm uygulama aynı serverda olacağı gibi veritabanı ve uygulama ayrı sunucularda da barındırılabilir. Aşağıdaki şekilde görüldüğü üzere cluster yapısı kullanılmadan tek sunucu yada her uygulama için bir sunucu mantığı yer almaktadır.



Şekil 4.16 B2B Sistemi Mimarisi

B4B karma mimarisi B2B mimarileri üzerinde kullanılmaya gerek duyulmayan dağıtık mimari ve cluster yapıları üzerine inşa edilerek çoklu işlemler ve yüksek performans elde edilmesi amaçlanmıştır. B2B nin sorunlarından biri de bu sayede giderilmiş olmaktadır. Mimaride üç ayrı cluster yapısı bulunmakta. Detay vermeden önce clusterın ne olduğunu inceleyelim.

***Cluster:** Cluster mimarisi ya da cluster yapılar en yalın tanımıyla; belirli konfigürasyonlar ile yapılandırılarak aynı amaç için birlikte ya da yedekli çalışan sunucu kümeleridir. Cluster oluşturmak için gerekli sebepler değişkenlik göstermektedir. Biz projemizde yük dağıtımli clusterlar (Load-Balancing clusters) kullanacağız. Bu tarz clusterlarda birden fazla sunucu belirli bir servis ya da hizmeti oluşturan işlem yükünü ortak bir biçimde paylaşarak kullanırlar. Bir sunucuda hizmet kesilmesi sistemin çalışmasını engellemez.*

En alt katmanda bulunan cluster veritabanı cluster olarak konumlandırılmıştır. 3 farklı sunucuda çalışan veritabanı cluster gelen istekleri load balancer ile farklı serverlara ileterek yük dağılımını dengeler. Bir üst katmanda yer alan cluster web servislerimizi barındıran clusterdır. Her bir serverın üzerinde konuşturduğumuz IIS servislerinde barındırılan web servis uygulamalarına istekler yine aynı şekilde balancer ile iletilmektedir. Son olarak çalışma prensibi tek bir sunucunun görevini herhangi bir donanım yada yazılım problemi oluştuğunda diğer bir sunucunun otomatik olarak devralması esasına dayanan yüksek erişilebilirlik clusterı olarak kullandığımız ve uygulamamızı barındıran clusterdır.

Servisler ve veritabanı katmanlarında dağıtık mimari kullanmaktayız. Bu sayede uygulama altyapımızdaki veri akışında hız ve erişebilirlik sağlamış olmaktadır.

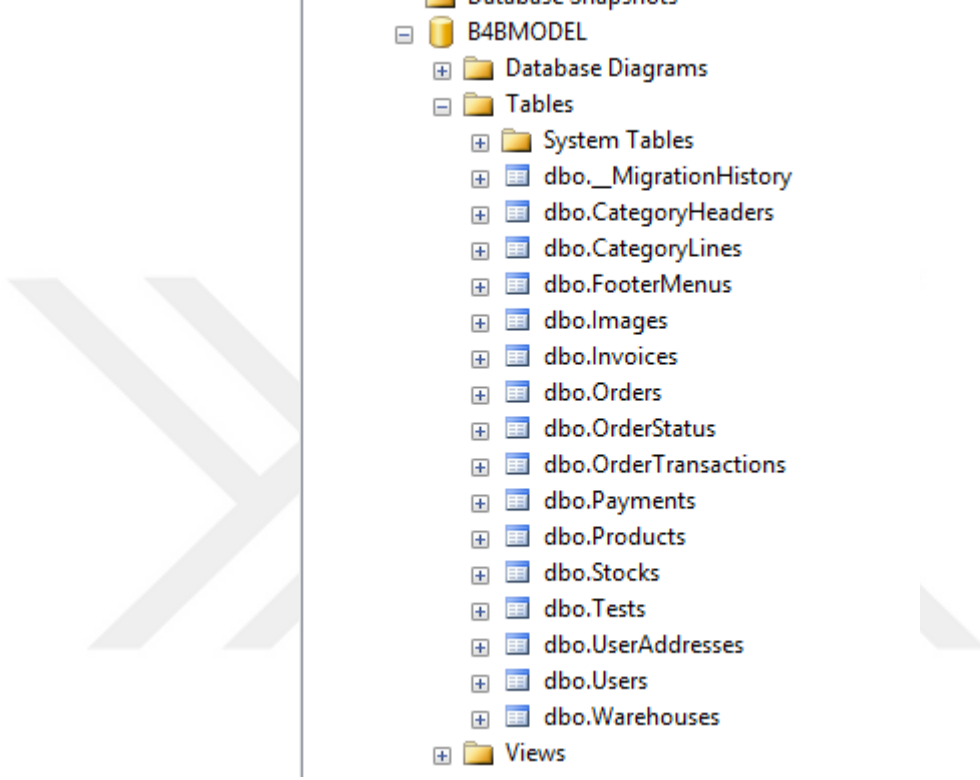


Şekil 4.17 B4B Karma Sistemi Mimari

4.2.2 Veritabanı Mimari

Veritabanı uygulaması olarak MS SQL Server 2012 kullanılır. Oluşturduğumuz B4BMODEL veritabanı tablo geliştirmelerini code first yapısıyla, web servisimiz içerisinde gerçekleştiriyoruz.

İhtiyaç olması durumunda Microsoft SQL Server Management Studio ortamında veritabanı müdahalelerimizi gerçekleştirebiliyoruz.



Şekil 4.18. Veritabanı Tablo Listesi

B4BMODEL Veritabanı Tabloları;

- ***Users;*** Kullanıcıya ait genel bilgilerin tutulduğu tablomuzdur.
- ***UserAddresses;*** Kullanıcının sipariş esnasında, teslimat için seçeceği adresleri barındıran tablomuzdur. Bir kullanıcının birden fazla adresi olabilir.
- ***Warehouses;*** Firmaya ait depoların eklendiği tablo. Bir firmanın birden fazla deposu bulunabilir.

- **Stocks;** Stokların bulunduğu depodur. Hangi product hangi depoda kaçane, kaç arızalı , kaç şüpheli bilgilerinin tutulduğu tablodur.
- **Products;** Ürünlerin Tutulduğu tablodur. Ürünün detayları, resim bilgileri, kategorileri, marka bilgileri ve satış yetkilerinin tutulduğu tablodur.
- **Images;** Ürünlere ait resimlerin bilgi ve adreslerinin tutulduğu tablodur.
- **FooterMenus;** Footer bölümünde yer alacak menü bilgileri bu tabloda tutulur.
- **CategoryHeaders;** Ana kategorilerin tutulduğu tablodur, aynı zamanda uygulamadaki menü buradan beslenir.
- **CategoryLines;** CategoryHeaders tablosunda yer alan her bir kategorinin alt kırılımları burada tutulur. Örnek: Ayakkabılar [CategoryHeaders] > Erkek Ayakkabısı [CategoryLines].
- **Orders;** Siparişlerin tutulduğu tablodur. Hangi ürünün kim tarafından , hangi adrese gönderildiği bilgisi tutulur.
- **Invoices;** Siparişe ait fatura bilgisinin tutulduğu tablodur.
- **OrderStatus;** Siparişlerin durum geçişlerinin tutulduğu tablodur. Örnek [Onayda, Teslim Edildi].
- **OrderTransactions;** Sipariş verilirken olası hata durumunda tutulan log tablosu.
- **Payments;** Fatura bilgisi ve toplam sipariş ücret tutarının tutulduğu tablodur.

Tablolarımız ilişkisel veri modelini kullanmaktadır. Bu sayede tablolardaki verilerin güvenliğini ve bütünlüğünü sağlamış olacağız.

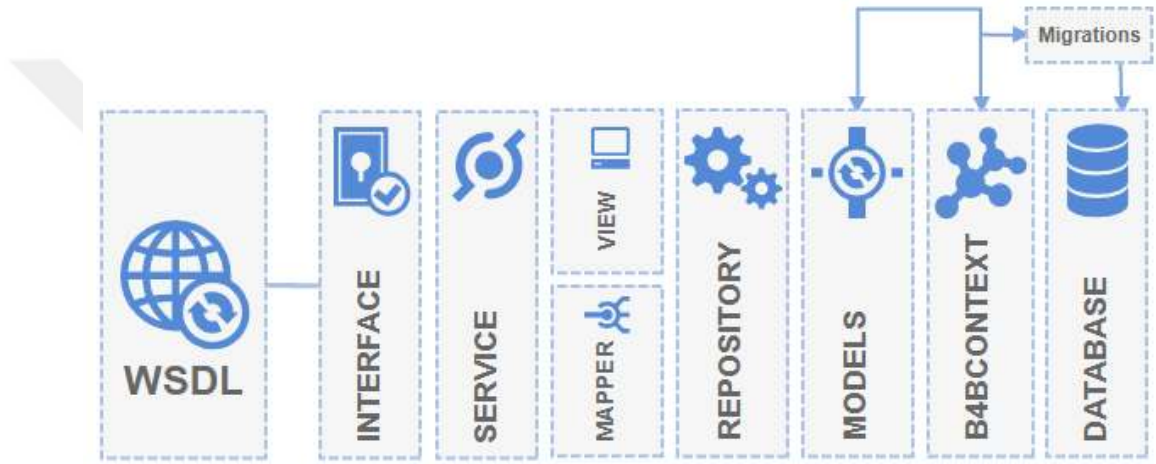
Her tablonun kendi indexlerini oluşturarak performans sorununu ortadan kaldıracaktır.



Şekil 4.19. Veritabanı İlişkisel Veri Modeli

4.2.3 Uygulama Servisi

Yapısal olarak veritabanı katmanının hemen üzerinde yer alan uygulama servisimiz, sistemin tüm veritabanı işlemlerini karşılayan birer metot içinde barındırır. Uygulama web servis aracılığı ile gerekli transaction işlemlerini gerçekleştirir. Servis uygulamamızı geliştirirken WCF web servis framework apisini kullandık.



Şekil 4.20. Web Servis Mimarisi

Servis mimarisinin en alt katmanında veritabanı bulunmakta. Şekil 6.2 de yer alan veritabanı cluster ile bağlantı sağlayarak verileri işlenmesini sağlamaktayız. Veritabanı erişiminin sürekliliği, yük dağılımını otomatize edilmesi ve performans için veritabanı server önünde konumlandırılmış bir balaster yapımız mevcut.

Veritabanı katmanının üzerinde yer alan ve tablolarının bilgilerini tutan ve veritabanı erişim katmanı olarak kullanılan “Models” ve “B4Bcontext” sınıflarımız mevcuttur. Bu iki sınıf veritabanı classlarıdır.

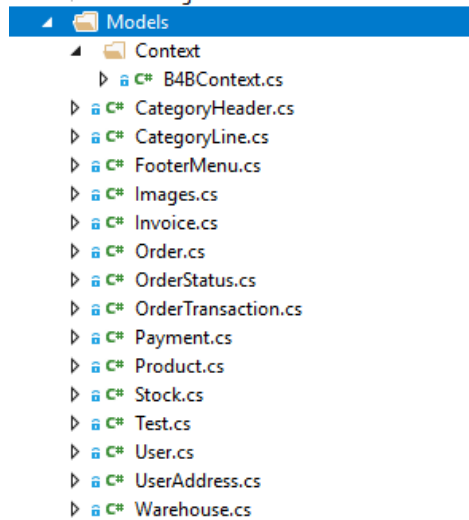
```

6 references | yildiray.yildirim, 6 days ago | 1 author, 1 change
public class CategoryHeader
{
    1 reference | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public int Id { get; set; }
    1 reference | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public string CategoryHeaderID { get; set; }
    1 reference | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public string CategoryHeaderName { get; set; }
    1 reference | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public string CategoryHeaderDescription { get; set; }
    2 references | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public bool Lock { get; set; }
    1 reference | yildiray.yildirim, 6 days ago | 1 author, 1 change
    public ICollection<CategoryLine> CategoryLine { get; set; }
}

```

Şekil 4.21. Models Altında Yer Alan Tablo Sınıfı Örneği

İhtiyacımız olan bütün tabloları models altında bir sınıf olarak yaratıp, yarattığımız tüm sınıfları da B4Bcontext sınıfına tanıtmamız gerekir. Tüm tanımlamalar yapıldıktan sonra “package manager console” ile migration işlemlerini yapıp veritabanımızın güncellenmesini sağlıyoruz. Migration yapısını kod olarak oluşturduğumuz sınıfların veritabanına işlemlerini sağlayan bir yöntem olarak düşünebiliriz. Migration içerisinde veritabanı settings metotlarını tutan bir “Configuration” sınıfı mevcuttur.

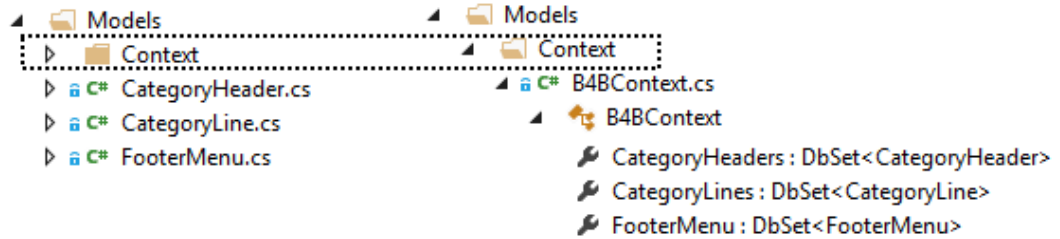


Şekil 4.22. Models ve Context Genel Görünüm

Model klasörü altında yer alan her bir sınıf bir tablo olarak veritabanı generate edilecektir. Ayrıca her bir sınıf context klasörü altında B4Bcontext classının içerisine tanımlanmış olmalıdır. B4Bcontext sınıfı “System.Data.Entity.DbContext” sistem sınıfını miras almıştır. Şekil 6.8 de görüldüğü gibi her bir tablo sınıfı B4Bcontext sınıfı altında DbSet ile tanımlanmıştır.

```
public partial class B4BContext : DbContext
{
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Test> Tests { get; set; }
    2 references | 0 changes | 0 authors, 0 changes
    public DbSet<Product> Products { get; set; }
    1 reference | 0 changes | 0 authors, 0 changes
    public DbSet<Image> Images { get; set; }
    8 references | 0 changes | 0 authors, 0 changes
    public DbSet<User> Users { get; set; }
    1 reference | 0 changes | 0 authors, 0 changes
    public DbSet<CategoryHeader> CategoryHeaders { get; set; }
    1 reference | 0 changes | 0 authors, 0 changes
    public DbSet<CategoryLine> CategoryLines { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<FooterMenu> FooterMenu { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<OrderStatus> OrderStatus { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Order> Order { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<UserAddress> UserAddress { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<OrderTransaction> OrderTransaction { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Invoice> Invoice { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Payment> Payment { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Warehouse> Warehouse { get; set; }
    0 references | 0 changes | 0 authors, 0 changes
    public DbSet<Stock> Stock { get; set; }
}
```

Şekil 4.23. B4Bcontext Sınıfı Tanımları



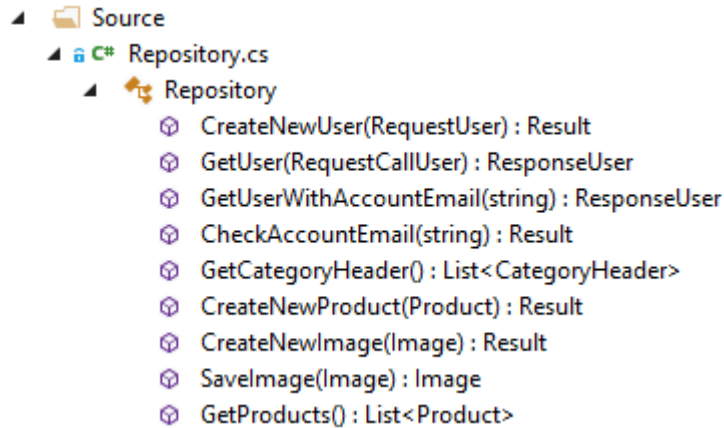
Şekil 4.24 .Models ve Context Sınıf İlişki Yapısı Örneği

Models içerisinde yer alan tablo sınıfları oluşturulduktan ve context sınıfı tanımlaması yapıldıktan sonra “package manager console” ile veritabanı işlenmesi sağlanır. Bunun için ilk olarak “add-migration updateImages” komutu sınıflar oluşturulur. Buradaki updateimages migration işlemine verdiğimiz isimdir. Son olarak “update veritabanı” komutu ile veritabanı güncellenir.

```
PM> add-migration updateImages
Scaffolding migration 'updateImages'.
The Designer Code for this migration file includes a snapshot of your current Code First model. The migration. If you make additional changes to your model that you want to include in this migration
PM> update-database
Specify the '-Verbose' flag to view the SQL statements being applied to the target database.
Applying explicit migrations: [201805120941133_updateImages].
Applying explicit migration: 201805120941133_updateImages.
Running Seed method.
```

Şekil 4.25. Models ve Context Tablolarının Veritabanına İşlenmesi

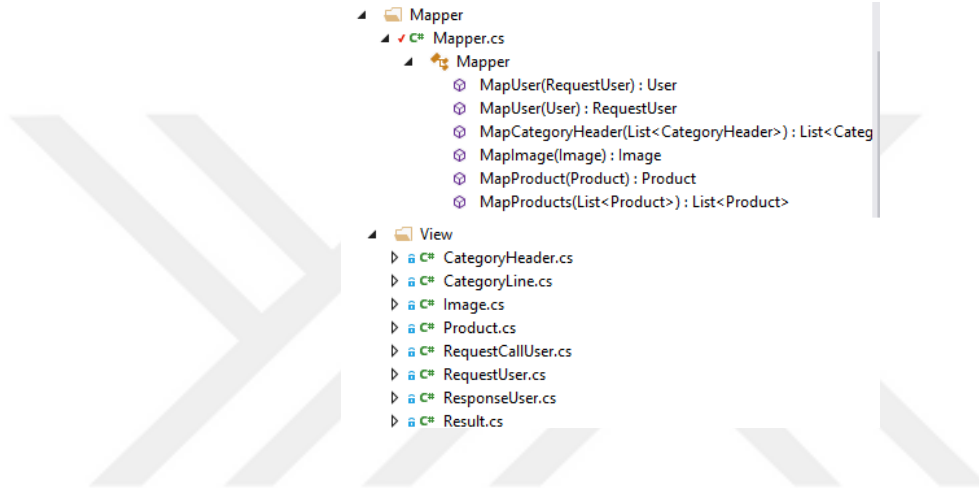
Models üzerinde repository katmanını bulunmaktadır. Repository katmanını gelen taleplerin işlendiği ve veritabanı iletildiği bağlantı sınıfı olarak tasarlandı.



```
Source
└─ C# Repository.cs
   └─ Repository
      ├── CreateNewUser(RequestUser) : Result
      ├── GetUser(RequestCallUser) : ResponseUser
      ├── GetUserWithAccountEmail(string) : ResponseUser
      ├── CheckAccountEmail(string) : Result
      ├── GetCategoryHeader() : List<CategoryHeader>
      ├── CreateNewProduct(Product) : Result
      ├── CreateNewImage(Image) : Result
      ├── SaveImage(Image) : Image
      └── GetProducts() : List<Product>
```

Şekil 4.26. Repository Sınıfı Örneği

Repository katmanı üzerinde “mapper” ve “view” sınıflarını konumlandırdık. View katmanı servislere gelen talepleri karşılayan ve dönen sınıfları içerir. Mapper ise servise istek yapan clientın ilettiği view sınıflarını veritabanının istediği sınıf türlerine, veritabanından okunan verileri clientlara dönecek sınıflara çeviren metotları içerir. Bu katman bir karşılama ve dönüşüm katmanıdır.



Şekil 4.27. Mapper Metotları Örneği ve View Sınıfları

Service katmanı interfaceden önce, dışarıya açılan servislerin metotlarını barındıran sınıftır. Interface katmanını servisin dışarıya açılan metotlarını, contact ve operation contact , message contract, policy and binding izinlerini içerir WSDL servis kullanıcılarının servisi call etmeleri için gerekli endpoint adresini indexler.

```

public interface I@bservice
{
    [OperationContract]
    View.Result CreateNewUser(RequestUser RequestUser);
    [OperationContract]
    View.ResponseUser GetUser(RequestCallUser RequestCallUser);
    [OperationContract]
    View.ResponseUser GetUserWithAccountEmail(string ApiKey, string AccountEmail);
    [OperationContract]
    View.Result CheckAccountEmail(string ApiKey, string AccountEmail);
    [OperationContract]
    List<View.CategoryHeader> GetCategoryHeader(string ApiKey);
    [OperationContract]
    View.Result CreateNewProduct(View.Product RequestProduct);
    [OperationContract]
    View.Result CreateNewImage(View.Image RequestImage);
    [OperationContract]
    List<View.Product> GetProducts(string ApiKey, int Id);
}

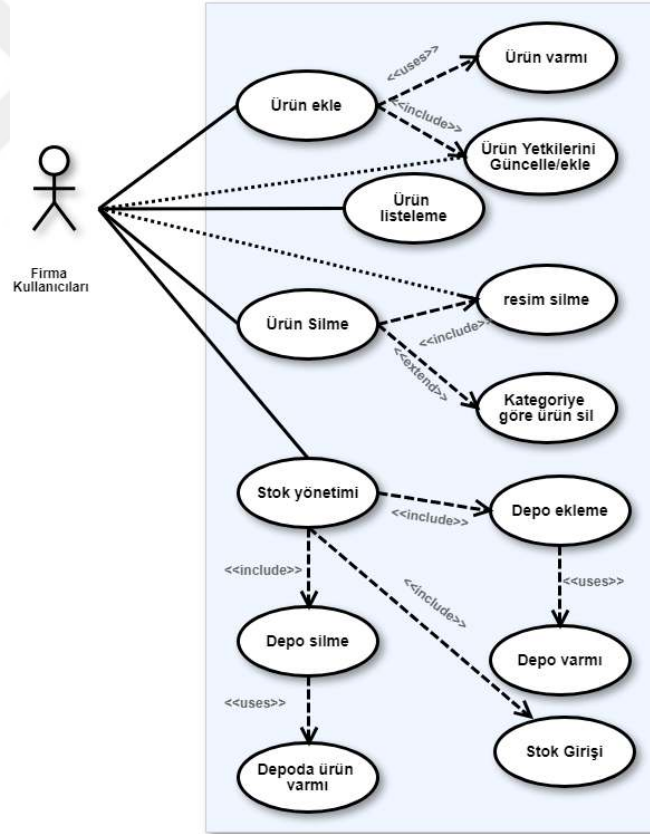
public class Service1 : I@bservice
{
    [reference | yildirim.yildirim, 5 days ago | 1 author, 2 changes]
    public View.Result CreateNewUser(RequestUser RequestUser)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 2 changes]
    public View.ResponseUser GetUser(RequestCallUser RequestCallUser)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 1 change]
    public View.ResponseUser GetUserWithAccountEmail(string ApiKey, string AccountEmail)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 1 change]
    public View.Result CheckAccountEmail(string ApiKey, string AccountEmail)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 1 change]
    public List<View.CategoryHeader> GetCategoryHeader(string ApiKey)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 2 changes]
    public View.Result CreateNewProduct(View.Product RequestProduct)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 1 change]
    public View.Result CreateNewImage(View.Image RequestImage)
    [reference | yildirim.yildirim, 5 days ago | 1 author, 1 change]
    public List<View.Product> GetProducts(string ApiKey, int Id)
}

```

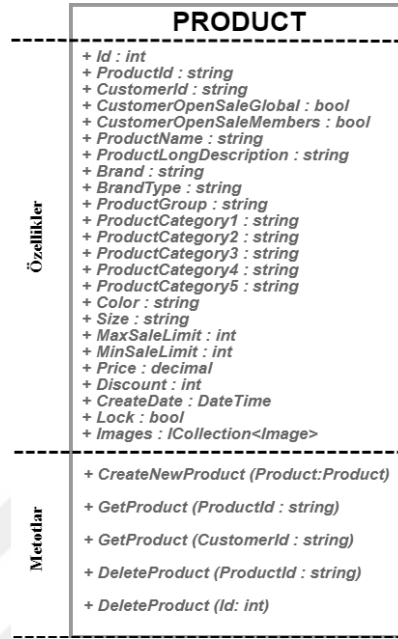
Şekil 4.28. Inteface ve Service Sınıfları

4.2.4 UML Diagrams ve Usecase

Uygulamamızın tüm akış ve operasyonel işlemlerinin veritabanı ilişkisini uygulama servisinden kontrol etmekteyiz. Tabloların insert, update ve delete işlemleri ile birlikte sorgulamalar, raporlar ve dosyalama işlemlerini servisten yürütüyor olacağız. Bu sebeple işlem gerektiren tüm süreçlerin diyagramları ve kullanıcı senaryoları (usecase) servis ve uygulamanın tamamı düşünülerek tasarlandı. Uml diyagramlar ve use caseler içlerinde hem web app hem de servis metotlarını barındırır bir mantıkla ortaya konmaya çalışıldı.



Şekil 4.29. Ürün ve Stok Yönetimi İçin Örnek Bir Use Case



Şekil 4.30. Ürüne Ait Class Diagramı

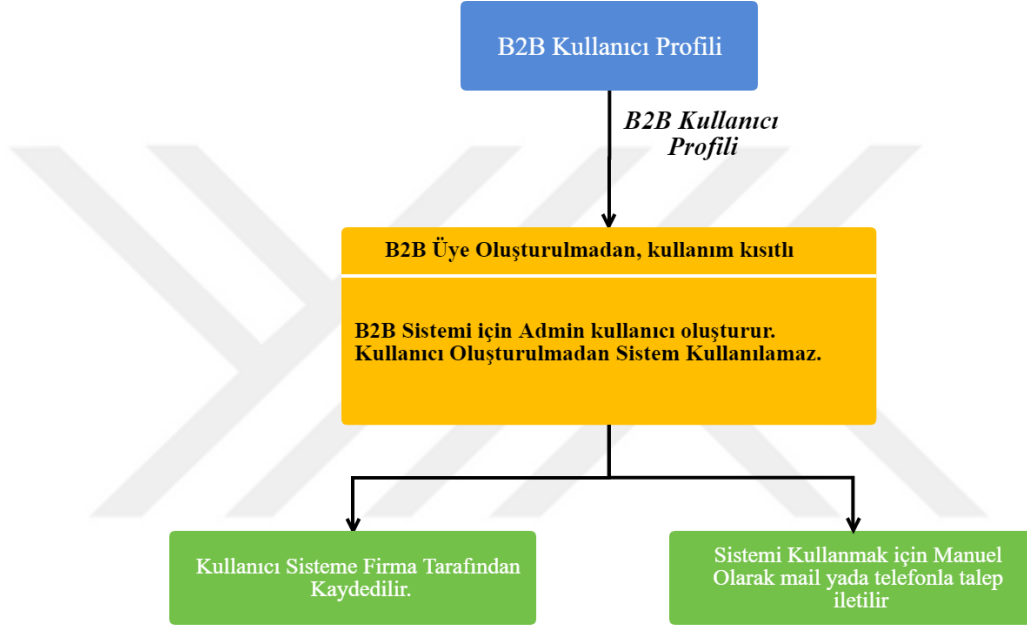
4.2.5 B4B Karma Sistemine Genel Bakış

B4B karma sistemi B2B ve B4B organizasyonlarının tüm operasyonel süreçlerini içermekte ve bu nedenle oldukça kapsamlı iş akışlarını yapısında barındırmaktadır. Tüm bu iş akışlarını bir düzen içerisinde uygulama görüntüleriyle detaylandırmak hem net fotoğrafı çekebilmek hem de sistemin çalışma mantığının daha iyi kavranması açısından önemlidir.

4.2.6 Kullanıcı Profilleri

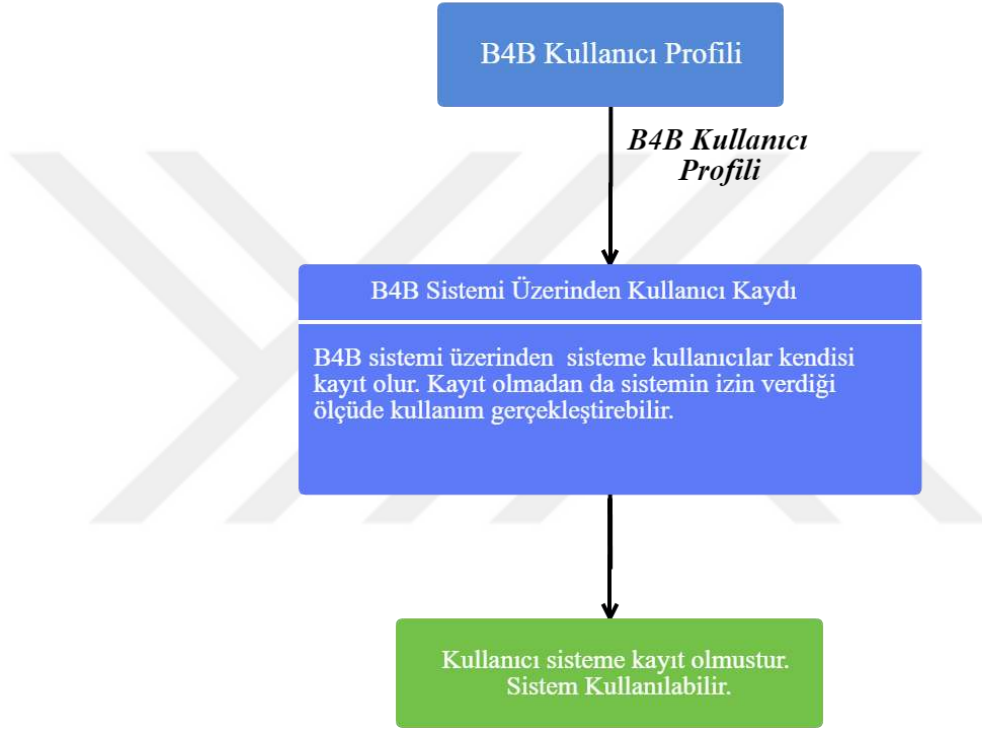
B2B, B4B ve B4B karma sistemlerini birbirinden ayıran ana unsurlardan biri de kullanıcı profillerinin sistem üzerindeki mimariye etkisi , işleyiş mimarisi açısından kullanıcı rolleri sistemin çalışma mantığını belirlemekte. Öncelikle B2B ve B4B sistemlerindeki kullanıcı rollerini inceleyerek, B4B karma sisteminde bu kullanıcı mimarisine getirdiğimiz yeniliği vermeye çalışacağız.

B2B kullanıcı yapısı, uygulamayı kullanan firma tarafından kullanıcıların oluşturulması mantığına dayanır. Yani B2B sisteminde kullanıcıya bir account açılır. Bu account açılmadan yapı gereği kullanıcı sisteme erişemez



Şekil 4.31. B2B Kullanıcı Profili İş Akışı

B4B Kullanıcı mantığı genel geçer e-ticaret yapısı mantığında işler kullanıcı kaydını kendisi oluşturur. Kullanıcı oluşturmada sistem belli kısıtlarla kendini herkese kullanma imkanı tanır.



Şekil 4.32. B4B Kullanıcı Profili İş Akışı

Sistem, alıcı ve satıcı profilleri üzerinde, temelde iki iş akışı üzerinden ilerlemektedir. Bu iki temel akış kendi içinde ayrılıklar göstermekle beraber sistemimizin tüm yapısının, akışının belirlenmesinde ana unsurdur. Bu iki temel profil birbirlerinin rollerini alabilmektedir. Bir satıcı aynı zamanda alıcı, bir alıcı ise satıcı olabilir. Alıcı ve satıcı profilleri birbirleri ile abonelik yapısı üzerinden etkileşimde bulunurlar.

Bir alıcının , herhangi bir satıcının müşterisi olması sistemimizde abonelik olarak değerlendirilir. Abone olunan firmalar alışverişle ilgili kıstaslarını buna göre düzenleyebilirler. Profillerin detaylarını inceleyecek olursak;

Alıcı satıcının satışa çıkardığı ürünlerden yine satıcı firmanın belirlediği toptan miktar ve fiyat üzerinden sipariş verebilir. Satıcı profili oluşturmak sadece servis sağlayıcı olarak bizim sorumluluğumuzdadır. Sistemde satıcı olmak isteyen bir firma bizim müşterimiz olacak ve gerekli anlaşmalardan sonra kendisine bir müşteri profili yaratılıp bir account açılacaktır. Müşteri bu accountla kendisine bağlı alt firmalar için abonelik izni verip sipariş alabilir.

Uygulamadaki ikinci temel alıcı profilidir. Alıcı profilini satıcı firmalar oluşturabileceği gibi, kendileri de sisteme kayıt yoluyla üye olabilirler. Sistemimizde uygulamaya kayıt olunmadan sadece herkesin görebildiği menü ve ekranlara ulaşılabilir. Ürün sipariş vermek yada abonesi olduğu firmaların ürünlerine erişip satış siparişi geçmek için siteye giriş yapması gerekmektedir.

Bu bağlamda iki profil için;

Satıcı Profili;

- Sistemde satıcı olabilmek için servis sağlayıcısı tarafından firma profilinin tanımlanması, lisans verilmesi ve account açılması gerekmektedir. Tüm bunlar için karşılıklı yapılan bir sözleşmeyle sürecin resmileşmesi gerekmektedir.
- Satıcı sistemi kullanır hale gelince ürünlerini sisteme ekleyebilir, ürün için satış yetkilerini düzenleyebilir, kendi bayileri için kullanıcı isteği oluşturabilir.
- Satışlarını ve siparişlerini takip edebilir.
- Raporlama yapabilir.
- Gelen abonelik taleplerini yönetebilir.
- Muhasebe yönetimi yapabilir.

- Kampanya yönetimi gerçekleştirebilir.
- Stok yönetimi yapabilir.
- Satıcı profili aynı zamanda alıcı olarak da faaliyet gösterebilir.
- Ürünleri global satışa açmak ve belirlenen miktarlarda satışını gerçekleştirmeye olanak sağlar ki buda B4B süreçlerinin iş akışına dahil olan bir süreçtir. Buradaki önemli nokta global satışta ödemeler önceden alınır ve sipariş sonrasında onaylanıp teslim süreci başlatılır.
- Ürünlerin sadece abone firmalara satılmasını kapsayan ürün yetkilendirilmesi ise B2B sürecini ifade eder ve burada satış siparişi geçilir. Sipariş onaylanır ve faturalanarak muhasebeleşir. Burada ödeme alınması zorunlu değildir ve alıcı firmalar zaten satıcı üyenin sürekli çalıştığı müşterileridir.

Alıcı firma profili;

- Alıcı profili müşterisi olduğu satıcı firma tarafından iletilebileceği gibi, alıcı olmak isteyen bir firma tarafından da register olunarak elde edilebilir.
- Satıcı profilin alıcı olabileceği gibi, alıcı profilinde satıcı olabilme imkanı mevcuttur. Ancak alıcı profilin satıcı olabilmesi için servis sağlayıcısı tarafından yetkilendirilmesi gerekmektedir.
- Bir alıcı profil birden fazla firmaya abone olabilir. Bunun için abonelik talebi ilettiği satıcı firmanın bu talebi olumlu karşılaması ve onaylaması gerekir.
- Alıcı firma abone olma isteği iletimi esnasında, abonelik verecek olan alıcı firmanın sözleşme ön şartlarını kabul eder. Bunun yanı sıra vergi levhası örneği, mersis ve vergi numarası içeren kesilmiş kaşeli bir fatura örneği zorunlu olmak şartıyla satıcı firmaların abonelik talep ettiği evrakları sisteme girmeli ve satıcıya iletmelidir.

- Satıcı firma tarafından abonelik talebi gelmesi durumunda ki bu talep unıq bir mail adresinden iletilmeli, bu mail üzerinden kayıtlı olan bir alıcı firma varsa abonelik talebini kabul edebilir. Eğer mail adresi kayıtlı değilse sisteme kayıt olabilir.
- Alıcı firmalar herhangi bir satıcıya abone değilse sadece satıcıların global satışa çıkardığı ürünleri , satıcıların belirlediği koşulda ödeme yaparak alabilirler.
- Alıcı firma abonesi olduğu firmalardan ürün siparişi geçebilir. Bu siparişin oluşması için bir ödeme önkoşul değildir zira alıcı satıcının bir abonesidir.
- Alıcı firma sipariş takibi ve raporlama yapabilir
- Hem alıcı hem de satıcı ortak yönetim panelini kullanacaklardır. Menüler ve yapılan işlemler yetkisel olarak girilen profile göre ekrana yüklenecektir.

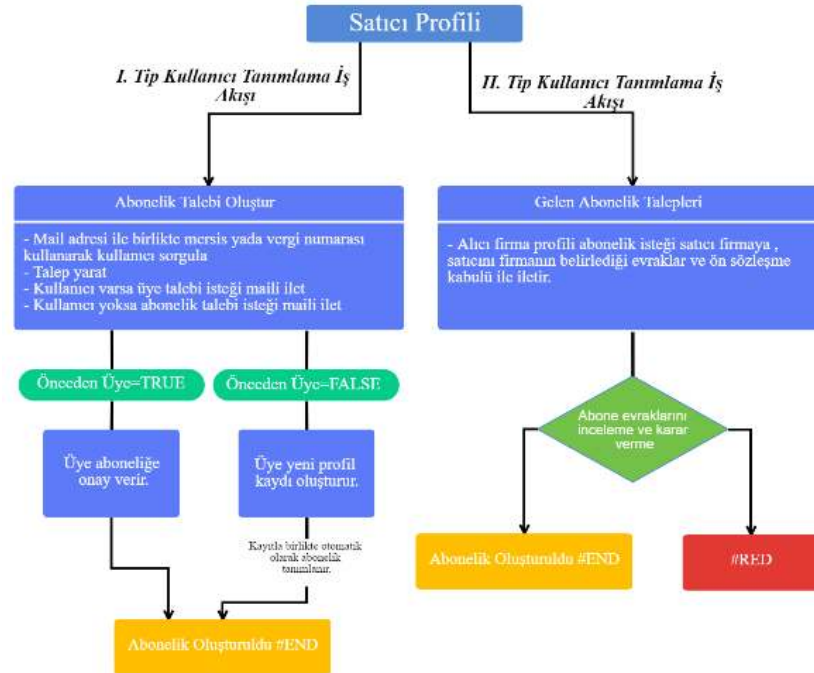
Uygulamamızda kullanıcı profilleri ve abonelik sistemi geriye kalan tüm akışların belirlenmesinde etken olduğu için akışların buna göre düzenlendiğinden bahsetmiştik. Sisteme satıcı profile kayıtlı bir abonelik açarak yada gelen abonelik taleplerini değerlendirerek , farklı satış stratejileri izleyebilir. Hakeza alıcı profile abonelik isteği ile firmaların müşterisi olma yada global satıştan yararlanma yoluna giderek kendi tedarik stratejilerini düzenleyebilir.

Alıcı ve satıcı için abonelik süreçlerinin iş akışlarının yapısını daha yakından değerlendirecek olursak; Satıcı profili için iki türlü akıştan söz edilebilir.

- Abonelik talebi oluşturma
- Gelen abone isteklerini değerlendirme

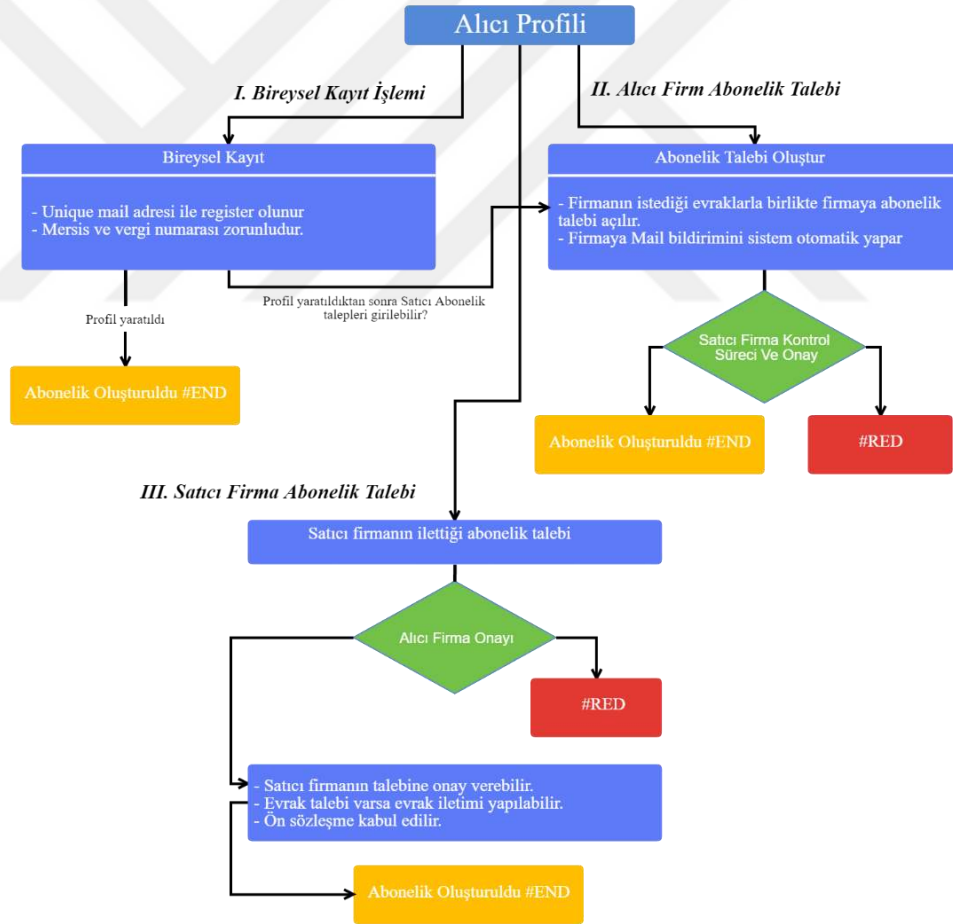
Abonelik talebi oluşturmak için, eklenecek firmanın unique bir e-mail adresi ile mersis numarası veya vergi numarası kullanılarak sorgulama yapılır. Eğer firma mevcutsa aboneye abonelik onayı için bir istek yönlendirilir. Bu istek mail olarakta firmaya iletilir. Karşı firma onay verdiği anda abonelik gerçekleşmiş olur. Şayet firma önceden sisteme üye değilse üyenin bilgileri girilerek kendisine, sisteme kayıt olması için bir davetiye iletilir bu davetiyeye istinaden aday firma sisteme kaydını gerçekleştirerek aboneliğini aktifleştirmiş olur.

Gelen abone istekleri ise satıcı firmaya mail yoluyla bildirilir. Aynı zamanda kendine ait yönetim panelinden gelen abonelik isteklerini raporlayabilir. Alıcı firma adayı abonelik başvurusu esnasında satıcı firma tarafından belirlenen evrakları sisteme yüklemek durumundadır. Satıcı firma gerek görürse bunların bir nüshasını posta yoluyla talep edebilir. İstenen evraklar sisteme yüklenip ön sözleşme metni onaylanırsa talep satıcı firmaya iletilir. Satıcı firma , aday alıcı firmayı inceleyip talebe onay yada red verebilir.



Şekil 4.33. Satıcı Profili Abonelik Tanımı İş Akışı

Alıcı profili adayı sisteme bireysel olarak kayıt olabilmeyi yanı sıra yukarıda belirtildiği gibi satıcı tarafından da davet edilebilir. Bireysel kayıt yapan alıcı firmalar abonelik talebi ile bir satıcıya üye değillerse sadece satıcı firmaların global satışa çıkan ürünlerini alabilirler. Bir satıcıya abone olmak isteyen alıcılar, satıcı firmanın profiline girerek istenen evrakları sisteme yükledikten sonra ön sözleşmeyi imzalayıp abonelik talebini iletebilir. Talep satıcı firma tarafından incelenerek onay yada red alabilir. Eğer satıcı tarafından talep alıcıya iletilmişse alıcı onay verip aboneliği kabul edebilir, isteniyorsa evrak talebine dönüş yapılabilir yada talebi reddedebilir.



Şekil 4.34. Alıcı Profili Kayıt ve Abonelik Talebi

BÖLÜM 5

SONUÇ VE ÖNERİLER

Teknolojinin gelişmesi, internet ve mobil dünyasında devrim niteliğindeki değişiklikler, teknoloji okur yazarlığının artması, sosyal medya ve dijital ticaretteki dönüşüm bilgi üretimini ve yayılımını hızlandırmıştır. Bu dijital hızlanma ile firmalar, ürettikleri değerlerin paylaşım yöntemlerinin de aynı nitelik ve donanımına sahip olması durumunda endüstriyel rekabette kendilerini bir adım öne çıkaracaklarını bilmektedirler.

Bu bilinçle yola çıkan firmalar üretimlerini mevcut alıcı firmalarla buluştururken teknolojinin getirilerinden faydalanmaktadırlar. Türkiye’de birçok firma satış pazarlama yöntemlerini dönüştürerek internet ve mobil teknolojinin desteğiyle B2B elektronik platformlara taşımışlardır. B2B platformları firmaların bireysel olarak oluşturduğu tek yönlü olarak kendi müşterilerine satış gerçekleştirdiği bir platformudur. Ancak rekabet düzeyinin artışı ve dünyadaki dijital dönüşüm trendleri ile B2B sistemler B4B sistemlere evrilmektedirler. B4B sistemleri B2B sistemlerden farklı olarak sadece tek yönlü satış kanalı olmanın dışında firmanın pazarlama ağını büyüten, satış paradigmalarını değiştiren iş ortağı mantığını ortaya koyar.

Tez kapsamında yapılan çalışmalar neticesinde elde edilen bulgular ve yapılması planlanan uygulama, ticari işletmelerin klasik bir B2B elektronik ticaret platformundan B4B platformuna geçiş trendini kapsar iş akışları ile yola çıkacak ve tüm bir uygulamayı tam bir SAAS platformu haline getirecektir. Bu ortak uygulama ile birden çok satıcı birden çok alıcıya hizmet verebilir ve sistemin içerisindeki yetkilendirme yapısıyla alıcılara kısıtlama koyabilirler. Bu sayede tezde oluşturmayı hedeflediğimiz sistem ile işletmeler gelişmiş bir “Karma Sistem” platformu kullanabilme imkanına sahip olabileceklerdir.

B2B ve B4B sistemlerinin ne olduđu ile ilgili literatür çalışmalarımızı yaparken elde ettiğimiz bulgularla daha önce çalıştığımız ve müşterimiz olan B2B firmalarının yapılarını inceledik. B2B sistemlerinin geliřtirmesini yapan firmaları analiz ettik. Bu ve benzeri firmaların çoğunun tekil satış kanalı olan B2B platformunu kullandığını tespit ettik. Tek kanallı satış yapan platformların genellikle řirket bünyesinde teknik olarak konumlandırılan sunucularda çalıştığını, teknik aksaklıklar dışında bazı kuruluşlarda güvenliğinde sıkıntı yarattığını gördük. Sonrasında uygulamamızın bir servis olarak hizmet vermesini sağlayacak olan SAAS mimarisinin nasıl uyarlanacağı konusunda çalışmalar yaptık. SAAS kullanmadaki temel amacımız kullananlar için tek bir noktadan erişim sağlayan, güvenilir ve firmaya ekstra bir teknik külfet getirmeyecek düzeyde stabil bir ortam oluşmasını sağlamaktı.

“Karma Sistem” platformunu geliştirirken çevik yaklaşım metotlarını kullandık. Bu durum ekibimizin iş gücü olarak daha ekonomik çaba ile yüksek seviyede performans göstermesini sağlamaktı. Çevik yaklaşım ile tezimizin amacı olan uygulamanın geri dönütlerde beslenen ve geliřtikçe büyüyen bir yapıda olması hedeflenmiştir. Projemiz de 22 adet her biri bir veya birkaç modülü kapsayacak düzeyde geniş maddeden oluşan product listesini 9 sprint ile geliştirme hedefindeyiz. Her sprint 4 hafta süre ile toplamda işin 36 haftayı kapsar şekilde yapılması planlanmıştır. Sprint sonrası toplantılar ve bir sonraki sprint planında yer alacak maddelere ekleme yapılması durumunda sprint sayısının artabileceğı ve proje bitim süresinin değıřkenlik gösterebileceğı öngörülmüştür.

Çalışma sonunda elde edilen bulgulara dayanarak aşağıdaki sonuçlar ortaya çıkmaktadır, ortaya koyduğumuz sonuçlar firmalar için bir öneri ve çözüm yöntemidir.

- Firmalar karma sistemimizi kullanarak çok büyük ve geniş operasyon akışını bir platform üzerinde elde edebilecekleri gibi, karma sistemin esnekliği iki platformu bir arada kullanabilme imkanı kazanacaklardır. Bu da firmaların tüm süreçlerini sorunsuz otomatize etmesini sağlar.
- Abonelik yönetimi ile sistemi kullanan firma bir B2B SAAS uygulaması edinmiş olur.
- Bir SAAS uygulaması edinimi ile firma server, teknik bakım, teknik personel ve genel bakım ve onarım maliyetlerinden kurtulur. Maliyetlerini düşürüp, kar marjını arttırmak isteyen firmalar sistemi tercih edebilirler.
- Firma uygulama ile bir muhasebe ve stok yönetim programı edinmiş olur.
- Firma sistemin geniş erişim ağı ile büyük bir pazara kendini açmış olur.
- Web tabanlı bir uygulama olduğu için sorunsuz olarak tüm platformlarda ve mobil cihazlarda çalışabilme özelliğine sahiptir ve online yapısıyla her an erişilebilir bir yapıdadır.
- Arttırılmış güvenlik yapısı ile firmaların operasyonları ve veri akışları güvende olacaktır.
- Yedekleme yöntemleri ile veri güvenliği sağlanacaktır.
- Api ve web servislerle firmalar kendi sistemleri ile uygulamayı entegre edebilirler

Sonuç olarak verilen bütün özellikler ve sistem detayları aslında firmalar için bir öneri olarak değerlendirilebilir. Firmalar sistemi kullanarak bu önerileri kendi avantajlarına kullanabilirler.

Tez kapsamında ortaya konan bulgular, elde edilen analizler ve geliştirilen ürün göz önüne alınacak olursa bir sonraki aşama için yapılacak değerlendirmede;

- B2C iş akışları sisteme implement edilebilir.
- Sistem üçüncü parti alışveriş sistemleriyle entegre edilebilir.
- Yapay zeka ve görüntü işleme çalışmalarıyla akıllı arama sistemleri geliştirilebilir.



KAYNAKÇA

Egemen Baş, A., B2B ve B2C Pazarlamada Müşteri Sadakati Programı Uygulaması, Yüksek Lisans Tezi, Kocaeli Üniversitesi, Kocaeli, 2016

Turkoğlu, Ü., B2B E-Business And Development Of A B2B E-Business Software, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, İzmir, 2006

Burak Arda, R., Endüstriyel Pazarlarda Faaliyet Gösteren Kobi'ler Açısından Müşteri İlişkilerinin Geliştirilmesi ve Yönetimi, Yüksek Lisans Tezi, İstanbul Üniversitesi, İstanbul, 2006

Bayoğlu, B., Effects of B2B E-Marketplaces to the Marketing Strategies of Companies, Yüksek Lisans Tezi, Yedi Tepe Üniversitesi, İstanbul, 2005

Tansu Turdağ, Ü., Firmadan-Firmaya E-Ticaret, E-Pazaryerleri ve Bir E-Pazaryeri Firmasının İş Planı, Yüksek Lisans Tezi, İstanbul Üniversitesi, İstanbul, 2001

Demir, N., A Research and Application About Integration of ERP (Enterprise Resource Planning) And B2B (Business-to-Business), Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, İzmir, 2006

Özceylan, D., Kobi'lerin Küresel Pazarlara Ulaşmasında E-Ticaret: OSB'lerde B2B Modeli, Yüksek Lisans Tezi, Sakarya Üniversitesi, Sakarya, 2004

Coşkun, N., Elektronik Ticaretin Gelişiminde Temel Dinamikler ve Gelişimi Önündeki Engeller, 01.2004, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü Dergisi Cilt 13, Sayı 2, 243-258, (<http://dergipark.gov.tr/cusosbil/issue/4370/59783>), Erişim Tarihi: 05/03/2018.

Kuşat, N., İşletmeden İşletmeye (B2B) Elektronik Pazaryerleri ve Gelişmekte Olan Ülkelerin Dış Ticaretinde Uygulanabilirliği, Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi 23, sayı 3, 691-709, (<http://dergipark.gov.tr/yonveek/issue/26778/281808>), Erişim Tarihi: 12/03/2018.

Sutherland, J., “SCRUM iki katı işi yarı zamanda yapma sanatı” (3. Baskı), Buzdağı Yayınevi, Ankara, 2014

Sarıdoğan, E., “Yazılım Mühendisliği Temelleri” (1. Baskı), Papatya Yayınevi, İstanbul, 2014

Nizam, A., “Yazılım Proje Yönetimi” (1. Baskı), Papatya Yayınevi, İstanbul, 2014

Highsmith, J. “Agile Software Development Ecosystems”, Pearson Education Corporate Sales Division, Boston, 2002

Keith, Clinton J. “Agile game Development with Scrum”, Pearson Education Corporate Sales Division, Boston, 2010

Shore, J., Warden, S., “The Art of Agile Development”, Oreilly, United States, 2008

Cockburn, A., “Agile Software Development”, Agile Software Development Series, Addison-Wesley Publishers Limited, 2001

Ali Babar, M., Brown, Alan W., Mistrik, I., Agile Software Architecture : Aligning Agile Processes And Software Architectures, Elsevier, 2014

Cockburn, A., Agile Software Development Software Development as a Cooperative Game, Pearson Education Corporate Sales Division, United State, 2007

Cockburn, A., A Human-Powered Methodology for Small Teams Including the Seven Properties of Effective Software Projects, Humans and Technology inc, United State, 2004

Akbayır, D., Bir Çevik Yazılım Geliştirme Sürecinin Uyarlanması ve Uygulanması, Yüksek Lisans Tezi, Maltepe Üniversitesi, İstanbul, 2010

Alparslan, G., Çevik Süreç İle Model Yönelimli Yazılım Geliştirme, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul, 2015

Yıldırım, Sinan., Çevik Yazılım Geliştirme İlkelerinin Proje Süreçlerine Etkileri, Yüksek Lisans Tezi, Marmara Üniversitesi, İstanbul, 2010

Dönmez, Ercan., Çevik Yazılım Geliştirme Sürecinde Kritik Başarı Faktörlerinin Belirlenmesi Ve Önceliklendirilmesine Yönelik Bir Örnek Çalışma, Yüksek Lisans Tezi, Bahçeşehir Üniversitesi, İstanbul, 2009

Duru, İsmail., Çevik Yöntemlerle Mobil Uygulama Tasarımı Ve Gerçekleştirilmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul, 2009

Acar, Ö., “Extreme Programming”(1 Baskı), Pusula Yayıncılık, İstanbul, 2009

Elibol, M., Selçukcan Erol, Ç., Scrum Metodu Kullanılarak Bir Mobil Uygulama Geliştirme Sürecinin Gerçekleştirilmesi. Bilişim Teknolojileri Dergisi 10, sayı 2, 169-176, (<http://dergipark.gov.tr/gazibtd/issue/28916/309299>), Erişim Tarihi: 12/02/2018.

Çetiner, Y., Bulut Bilişim Ve Örnek Bir SAAS Uygulaması, Yüksek Lisans Tezi, Kırıkkale Üniversitesi, Kırıkkale, 2014

Allahverdi, M., Bulut Muhasebe Sistemleri ve Bir SWOT Analizi, Muhasebe ve Finansman Dergisi, 92-105, (<http://dergipark.gov.tr/mufad/issue/35846/402447>) Eriřim Tarihi: 10/03/2018.

Gürbüz, A., “Yazılım Test Mühendislięi” (1. Baskı), Papatya Yayınevi, İstanbul, 2010

Silahtaroglu, G., “Sistem Analizi ve Tasarımı” (2. Baskı), Papatya Yayınevi, İstanbul, 2014

Taşdelen, A., “C++, JAVA ve C# ile UML ve Dizayn Paternleri” (1. Baskı), Pusula Yayınevi, İstanbul, 2014

Kılan, K., Yazılım Mühendislięinin Geliřimine Bir Bakıř, Türkiye Biliřim Vakfı Bilgisayar Bilimleri ve Mühendislięi Dergisi Cilt 8, Sayı 2, 31 - 36, (<http://dergipark.gov.tr/download/article-file/399019>), Eriřim Tarihi: 01/04/2018.

Kurtel, K., Eren, ř., Yazılım Mimarisinin Kalite Gereksinimleri:Yazılım Güvenilirlięi, Türkiye Biliřim Vakfı Bilgisayar Bilimleri ve Mühendislięi Dergisi, sayı 1, (<http://dergipark.gov.tr/tbbmd/issue/22243/238777>), Eriřim Tarihi: 01/04/2018.

Gruman, G., Knorr, E., 2008. What cloud computing really means. (<https://www.infoworld.com/article/2683784/cloud-computing/what-is-cloud-computing.html>). Eriřim Tarihi: 01.05.2018.

Turhan, Ö., 2009. Bulutların üzerinde iř yapmak; Cloud Computing. (<https://www.workcube.com/blog/bulutlarin-uzerinde-is-yapmak-cloud-computing/>) Eriřim Tarihi: 30.01.2013.

EK KAYNAKLAR

<https://www.visualstudio.com/wp-content/uploads/2017/04/agile-scrum-lifecycle-diagram.png> (2018)

[https://msdn.microsoft.com/tr-tr/library/dd381412\(v=vs.108\).aspx](https://msdn.microsoft.com/tr-tr/library/dd381412(v=vs.108).aspx)(2018)

[http://bidb.itu.edu.tr/eskiler/seyirdefteri/blog/2013/09/06/model-view-controller-\(model-g%C3%B6r%C3%BCn%C3%BCm-kontrol%C3%BC\)\(2018\)](http://bidb.itu.edu.tr/eskiler/seyirdefteri/blog/2013/09/06/model-view-controller-(model-g%C3%B6r%C3%BCn%C3%BCm-kontrol%C3%BC)(2018))

<https://azure.microsoft.com/tr-tr/services/sql-database/>(2018)

<https://developers.google.com/web/updates/2017/11/overscroll-behavior> (2018)

https://books.google.com.tr/books?id=bop3qVqOLzwC&pg=PA152&lpg=PA152&dq=get+cookiecontainer+from+web+references&source=bl&ots=YaD4-_nWcB&sig=e19SP9uziFEfKLLF9wOVsf0tDwE&hl=tr&sa=X&ved=0ahUKEwi0laup4fraAhVjJJoKHUgzCr8Q6AEIajAG#v=onepage&q=get%20cookiecontainer%20from%20web%20references&f=false, (Jamsa, K., .NET Web Services Solutions, 2018)

ÖZGEÇMİŞ

16 Nisan 1987, İzmir ili, Karşıyaka ilçesi doğumluyum. İlk, orta ve liseyi İzmir ili, Çiğli ilçesinde okuduktan sonra, Yıldız Teknik Üniversitesi, Eğitim Fakültesi, Bilgisayar ve Teknoloji Öğretimi Bölümüne kaydoldum. 2012 yılında mezun oldum 2012 yılından beri çeşitli özel sektörlerde yazılım uzmanı, yazılım sorumlusu olarak çalıştım 2013 yılında Beykent Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladım.

Yabancı dilim ingilizcedir.

Aday : YILDIRAY YILDIRIM