

T.C.
ALTINBAŞ UNİVERSİTESİ
Fen Bilimleri Ensititüsü/ Bilişim Teknolojileri

ÇEVİK YAZILIM GELİŞTİRME, ÇEVİK DÖNÜŞÜM VE VERİMLİLİK: BANKACILIK IT SEKTÖRÜNDE ÇEVİK DÖNÜŞÜM VE VERİMLİLİK

Gözde ÖZER

Yüksek Lisans Tezi

Danışman

Prof. Dr. Oğuz BAYAT

İstanbul, 2021

ÇEVİK YAZILIM GELİŞTİRME, ÇEVİK DÖNÜŞÜM VE VERİMLİLİK: BANKACILIK IT SEKTÖRÜNDE ÇEVİK DÖNÜŞÜM VE VERİMLİLİK

Gözde ÖZER

Bilişim Teknolojileri

Fen Bilimleri Enstitüsü'ne

Lisansüstü derece

Tezi olarak sunulmuştur.

ALTINBAŞ ÜNİVERSİTESİ

2021

Bu tezi okuduğumuzu; kapsam ve nitelik bakımından Yüksek Lisans tezi olarak yeterli bulduğumuzu beyan ederiz.

Prof. Dr. Oğuz BAYAT

Danışman

Sınav komitesi: (Yazılan ilk isim jüri başkanı, ikinci isim danışman olmalıdır.)

Prof. Dr. Oğuz BAYAT	Mühendislik ve Doğa Bilimleri, Altınbaş Üniversitesi	
Dr. Öğr. Üyesi İzzet Paruğ Duru	Meslek Yüksek Okulu, Gedik Üniversitesi	
Dr. Öğr. Üyesi Adil Deniz DURU	Spor Bilimleri, Marmara Üniversitesi	
Dr. Öğr. Üyesi Tuğçe Ballı ALTUĞLU	MDBF /Yazılım Mühendisliği, Üsküdar Üniversitesi	
Yrd.Doç.Dr. Muhammad Ilyas	Mühendislik ve Doğa Bilimleri, Altinbaş Üniversitesi	

Bu tezin Yüksek Lisans. tezi olarak bütün şartları sağladığını beyan ederim.

Prof. Dr. Oğuz ATA
Anabilim Dalı Başkanı

Fen Bilimleri Enstitüsü kabul tarihi:

____/____/____

Prof. Dr. Oğuz BAYAT
Enstitü Müdürü

Bu tezdeki tüm bilgilerin akademik kurallara ve etik davranışlara uygun olarak edinildiğini ve sunulduğunu beyan ederim. Ayrıca, bu kuralların ve davranışların gerektirdiği şekilde, bu çalışmada, orijinal olmayan tüm materyalleri ve sonuçları tamamen alıntı yaptığımı ve referans gösterdiğimi de beyan ederim.

Gözde ÖZER

ÖNSÖZ

Öncelikle yüksek lisans eğitimim boyunca desteği, anlayışını her zaman hissettiren sevgili anneme, babama ve kardeşime sonsuz teşekkür ederim.

Tez çalışmam da emeğini ve ilgisi esirgemeyen, bu sürecin sağlıklı ilerlemesinde desteğini, yönlendirmelerini ve yorumları esirgemeyen, çalışmanın ilerlemesine katkı sağlayan eş danışmanlarım ve değerli hocalarım Doç. Dr. Adil Deniz DURU' ya, İzzet Paluğ DURU' ya ve Danışmanım Prof. Dr. Oğuz Bayat'a teşekkürü bir borç bilirim.

ÖZET

BANKACILIK IT SEKTÖRÜNDE ÇEVİK DÖNÜŞÜM VE VERİMLİLİK

Gözde, ÖZER

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi

Danışman:Prof.Dr.Oğuz Bayat

Eş Danışman: Doç Dr.İzzet Paluğ Duru

Tarih: Haziran, 2021

Sayfa: 96

Bu tez çalışmasında waterfall çalışan bankacılık IT firmasının Agile (Çevik) metot ile dönüşümü ve verimliğe katkısı ne olmuştur sorusu irdelenmiştir. Değer, Tahmin edilebilirlik, Üretkenlik, Kalite, istikrar çerçevesinde ele alınmıştır. Bunun için farklı grafikler ve takımların örnek dataları kullanılmıştır. Çalışanlara etkisi için hem Kanban hem Scrum takımlarının geçmiş dataları da örnek alınarak verimlilik gösterilmiş ve sonuçlar analiz edilmiştir.

Çalışmanın sonuç kısmında Çevik yazılım geliştirme yöntemi ile çalışan firmaların eski çalışma şekillerini bırakarak, dünyada da gelişen çevik yazılım metotlarıyla çalıştıklarında verimliliğ in önemli olduğu ve dönüşürken gelişimin ve esnekliğin faydalı olduğu ve çalışan memnuniyeti ve müşteri memnuniyetinin arttığı gözlenmiştir.

Anahtar kelimeler: Agile (çevik),manifesto, Kanban, Scrum, Verimlilik.

ABSTRACT

[AGILE TRANSFORMATION IN BANKING IT SECTOR AND EFFICIENCY]

ÖZER, Gözde

[M.Sc.], [MIS], Altınbaş University,

Supervisor:Oğuz Bayat

Co-Supervisor: Doç Dr.İzzet Paluğ Duru

Date: [July, 2021]

In this thesis, the question of what has been the transformation of the waterfall-working banking IT company with the Agile method and its contribution to productivity is examined. Value, Predictability, Productivity, Quality, are handled within the framework of stability. For this, different graphics and sample data of the teams are used. For its effect on employees, efficiency was demonstrated by taking past data of both Kanban and Scrum teams as an example, and the results were interpreted.

As a result of this study, the companies that adopt the Agile software development method release their old mindset, and the productivity.

It has been observed that it is important, development and flexibility are beneficial while transforming and employee satisfaction and customer satisfaction increase..

Keywords: Agile, Kanban, Scrum,Efficiency

İÇİNDEKİLER

Sayfa

ÖNSÖZ **3**

ÖZET **4**

ABSTRACT **5**

1. YAZILIM GELİŞTİRME MODELLERİ **17**

1.1 GELENEKSEL YÖNTEMLER 17

1.1.1 Şelale(Waterfall) Model 17

1.1.1.1 Avantajları 18

1.1.1.2 Dezavantajları 18

1.1.2 Sarmal (Spiral) Model 19

1.1.2.1 Avantajları 20

1.1.2.2 Dezavantajları 20

1.1.3 V Model 21

1.1.3.1 Avantajları 22

1.1.3.2 Dezavantajları 22

1.1.4 Artırımlı Model 23

1.1.4.1 Avantajları 24

1.1.4.2 Dezavantajları 24

1.2 ÇEVİK 25

1.2.1 Çevik (Agile) Tarihçe 27

1.2.2 Çevik Bildiri (Agile Manifesto) 28

1.2.3 Yalın Düşünce 30

1.2.4 Çevik (Agile) Prensipler 30

1.2.5 Organizasyonel Çeviklik 31

1.2.5.1 Müşteri Çevikliği 31

1.2.5.2 Şirket İçi Çeviklik 31

1.2.6 Çevik Yaklaşımlar 32
1.2.6.1 Scrum 33
1.2.6.2 Extreme Programming (XP) 33
1.2.6.3 Kanban 34
1.3 SCRUM 35
1.3.1 Scrum'ın Değerleri 37
1.3.2 Scrum Takımı 38
1.3.3 Roller 39
1.3.3.1 Developers (Geliştiriciler) 39
1.3.3.2 Product Owner (Ürün Sahibi) 39
1.3.3.3 Scrum Master (Scrum Uzmanı) 40
1.3.4 Scrum Etkinlikleri (Scrum Ritülleri) 41
1.3.4.1 Sprint 41
1.3.4.2 Sprint Planlama 42
1.3.4.3 Daily Scrum (Günlük Toplantı) 42
1.3.4.4 Sprint Review 42
1.3.4.5 Sprint Retrospective 43
1.3.5 Scrum Eserleri 43
1.3.5.1 Product Backlog (Ürün iş listesi) 43
1.3.5.2 Sprint Backlog 43
1.3.5.3 Bitti Tanımı (Defination Of Done) 44
1.3.6 Scrum Metrikleri 45
1.3.6.1 Story Point Burn Down Chart (SP Burn-Down) 45
1.3.6.2 Efor burndown chart 46
1.3.6.3 Product burndown chart 47
1.3.6.4 Release burndown chart 48
1.4 KANBAN 49

1.4.1 Kanban'ın Değerleri 51
1.4.2 Kanban Rolleri 51
1.4.3 Kanban Prensipleri 52
1.4.4 Kanban Amaçları 52
1.4.5 Kanban Pratikleri 52
1.4.6 Kanban Etkinlikleri (Kanban Ritüelleri) 54
1.4.6.1 Daily 54
1.4.6.2 Replenishment 54
1.4.6.3 Retrospective, İteresyon Sunumları 55
1.4.7 Kanban Metrikleri 55
1.4.7.1 Monte Carlo Simülasyonu 55
1.4.7.2 WIP(Work –in Progress) 56
1.4.7.3 CFD (Cumulative Flow Diagram) 56
1.4.7.4 Lead Time,Cycle Time 57
1.4.7.5 Throughput (Çıktı) 58
1.5 ORGANİZASYONEL ÇEVİKLİK (ORGANIZATIONAL AGILITY) 59
1.5.1 Bir Organizasyonu Çevikleştirmek İçin Yapılması Gerekenler 61
1.6 VERİMLİLİK 66
1.6.1 Süreç Verimliliği 68
1.6.1.1 Süreç Verimliliğinin Ölçülmesi 68
1.6.1.2 Takip et 70
1.6.1.3 Akışa odaklanmak 70
1.6.2 Sık Kullanılan Metrikler 74
2. ARAŞTIRMANIN AMACI VE VERİLER 83
3. ARAŞTIRMANIN BULGULARI 103
4. SONUÇ VE ÖNERİLER 106
REFERANSLAR 108

ÖZGEÇMİŞ 112

TABLO LİSTESİ

Tablo 1-1 Çalışma Metodu Ve Büyüklüğe Göre Proje Durumu..... 13

Tablo 1-2 Agile Manifesto..... 15

Tablo 1-3 Gerçekleşmemiş Değer (Unrealized Value)..... 62

Tablo 2-1 Çeviklik ile ilgili örnek çalışmalar 84

Tablo 5-1 Geleneksel Yöntem/Agile Yöntem Karşılaştırması 91

ŞEKİL LİSTESİ

Şekil 2-1 Waterfall Model.....4

Şekil 2-2 Risk Analizi Gösterimi.....6

Şekil 2-3 V Model Gösterimi.....7

Şekil 2-4 Artırımlı Model Gösterimi.....9

Şekil 2-5 Agile.....12

Şekil 2-6 Agile Manifesto.....15

Şekil 2-7 Scrum'ın Değerleri.....23

Şekil 2-8 Scrum Etkinlikleri.....27

Şekil 2-9 SP Burndown Chart 32

Şekil 2-10 Efor Burndown Chart 33

Şekil 2-11 Product Burndown Grafiği 34

Şekil 2-12 Release Burndown Chart 34

Şekil 2-13 Kanban Değerleri 37

Şekil 2-14 Kanban Pratikleri 39

Şekil 2-15 Monte- Carlo Simülasyonu 41

Şekil 2-16 WIP 42

Şekil 2-17 CFD 43

Şekil 2-18 Lead Time-Cycle Time 44

Şekil 2-19 Ne Throughput 44

Şekil 2-20 EMB Rapor 61

Şekil 2-21 Mevcut Değer: Current Value (CV) 63

Şekil 2-22 Pazara Çıkış Süresi 64

Şekil 2-23 İnovasyon Yeteneği 66

Şekil 2-24 Verimlilik 69

Şekil 2-25 Verimlilik Başlıkları 70

Şekil 2-26 Takım Sayısı 72

Şekil 2-27 Agile Takım Çıktısı 73

Şekil 2-28 İş Tipi Dağılımı 74

Şekil 2-29 Throughput Trendi 74

Şekil 2-30 Deployment Performansı 75

Şekil 2-31 Proje Süresi 76

Şekil 2-32 Problem Sayısı 77

Şekil 2-33 Vaka 77

Şekil 2-34 Problemi Çözme Süresi 78

Şekil 2-35 Müşteri Memnuniyeti 79

Şekil 2-36 Proje Süresi 79

Şekil 2-37 Proje Adetleri 80

Şekil 2-38.Pazar Çıkış Süresi 80

Şekil 2-39 Projenin Süresi 81

Şekil 2-40 Kod Commit Adeti 82

Şekil 2-41 Kod Satır Sayısı 83

Şekil 2-42 Kodun Tarama Sayısı 97

Şekil 2-43 Kodun Built Adedi 98

Şekil 2-44 Jenkins Tarama Sayısı 99

Şekil 2-45 Deployment oranı 99

Şekil 2-46 Planlı Deployment Oranı 100

GİRİŞ

Yazılım geliştirme sürecinin başlaması için mutlaka gereksinimler vardır. Bu gereksinimler in ortaya çıkmasına neden olan şeyler ya kişi, ya patron ya da kişinin kendisi, hatta teknolojik teknolojik çalışmakta olduğumuz müşteriler ve olası müşterilerdir. Gereksinimlerin ortaya çıkmasındaki ihtiyaçların nasıl ve neden olduğundan ziyade işlerin etkin ve efektif bir şekilde yapılmasını sağlayacak çözüm yada çözümlerin bulunması, farklı fikirler üretmeniniz gerekmektedir. Konu yazılım olduğunda yeni çözüm yolları keşfetmek için beklenen sorunsuz, atik, kullanıcının kolay kullanabileceği ekranları olan bir yazılımın üretilmesidir. Müşterinin gereksinimleri için teslimat süreci genellikle çok kısa zamandır. Kısa teslimat süresi içinde bitirilmesi gerekmektedir. Yazılım geliştire sürecinde hizmet verilecek kişi ya kurumsal firmalar/organizasyonlar olabilir. Ancak hizmet alanı farklılık göstermeksizin göz ardı edilmeyecek aşamalar vardır.

Ama bu işi genellikle çok kısa bir süre içinde bitirmeniz istenir. Planlama ve hayata geçirme süresi ne kadar kısa olsa da, bu kısa zamanda geliştirecek olsanız da , ister kendiniz için, ister patronunuz için, ister müşterileriniz için yazılım geliştirin, bu geliştirmeleri yaparken göz ardı edilmemesi gereken aşamalar vardır.

Bu aşamalar iş analizi, gereksinim analizi, tasarım, geliştirme, test, devreye alma. Bu süre ç uzun süren, yorucu ve sıkıntı süreçlerdir. Yazılım projeleri genellikle Standlish grup çalışmasına da göre yönetimsel eksiklikler ya da hatalardan kaynaklanan kısmı başarı ve müşterinin memnuniyeti ile sonuçlanmaktadır. Yazılım sektöründe, yazılım projelerinin müşteriye söz verildiği zamanda teslim edilememesi, müşterinin istediği değişiklik gerektiren ihtiyaçlara/gereksinimlere hızlı dönüş sağlanmaması, yazılım hatalarının sonradan fark edilmesi ve süreç içerisinde doğan ek taleplere istinaden sistemin kendi yapısını geliştirememesi vb. sorunlar meydana gelebilmektedir.

Meydana gelen yeni sorunun çözümü için yapılan çalışmaların neticesinde 1990'ların sonlarında "Çevik" olarak adlandırılan yöntemler geliştirilmiştir.

Çevikliği bir şemsiyeye benzetirsek, bu şemsiyenin altında birden çok uygulama vardır. Bu uygulamalar; Scrum, Kanban, Scrumban, SaFe, Lean, XP vb. Bu metotların içerisinde en yaygın olarak tercih edilen Scrum çevik yazılım geliştirme metodudur. Scrum metodolojisi daha esnek bir yazılım geliştirme sürecini devam ettirebilmek ve müşterinin isteğine daha hızlı cevap verebilmek, daha adapte olabilmek adına müşterinin ve paydaşların katılımıyla geliştirilen yaklaşımdır. Bilhassa müşterinin memnuniyetinin sağlanması için geliştirmenin bazı aşamalarında da proje ve teslim edildiğinde de değişiklik yapılabilmesine, yeni gereksinimlerin dikkate alınmasına olanak verir. Neticede müşteri/müşteriler perspektifinden en ideal ürünün meydana getirilmesi ve müşteri memnuniyetinin sağlanması gerekmektedir. Bu çalışma şeklini yönetebilecek yetenekli, disiplinli, kendi kendine organize olabilen ve yönetebilen bir takımın oluşturulması oldukça önem arz etmektedir. Önemli olan devamlı değişime hızlı cevap verebilecek, değişimin getirdiği durumlarda/ortamlarda çalışabilecek takım/takımların bazı özel araç ve etkinliklerden yararlanması gerekmektedir, Scrum ve Kanban ile bahse konu araçlar ve etkinlikler sağlanabilmektedir. Yapılan bu araştırma da Bankacılık bilgi teknolojileri sektöründe yazılım geliştirme çevik dönüşüm ve dönüşen şirkette verimlilik konusunu incelenmesi ve bahse konu olan Bankacılık sektöründe verimliliğin incelenmesine yönelik bir uygulama yapılması hedeflenmiştir. Bu kapsamda çalışmanın birinci bölümünde yazılım geliştirme alanında waterfall proje yönetiminden Çevik Metotta dönüşümü ve buna istinaden yazılım geliştirme süreçleri ve modelleri açıklanmıştır. Çalışmanın devamında ise Çevik (Agile) yazılım geliştirmede yöntemi dönüşüm süreci açıklanmış ve Agile dönüşümle birlikte hem takımların verimliliği hem de şirketin verimliliği açıklanmıştır.

Çalışmanın uygulama kısmında ise Kanban ve Scrum takımlarının gerçek dataları, müşterilerden alından geri bildirimler dikkate alınıp veriler tatbik edilerek yorumlanmıştır.

Çalışmanın sonuç bölümünde Çevik yazılım geliştirme yöntemini kullanan firmaların geleneksel yöntemlerinden vazgeçmelerini dünyada da gelişen çevik yazılım metotlarıyla çalıştıklarında verimliliğin önemli olduğu ve dönüşürken gelişimin ve esnekliğin faydalı olduğu ve çalışan memnuniyeti ve müşteri memnuniyetinin arttığı gözlenmiştir.

1. YAZILIM GELİŞTİRME MODELLERİ

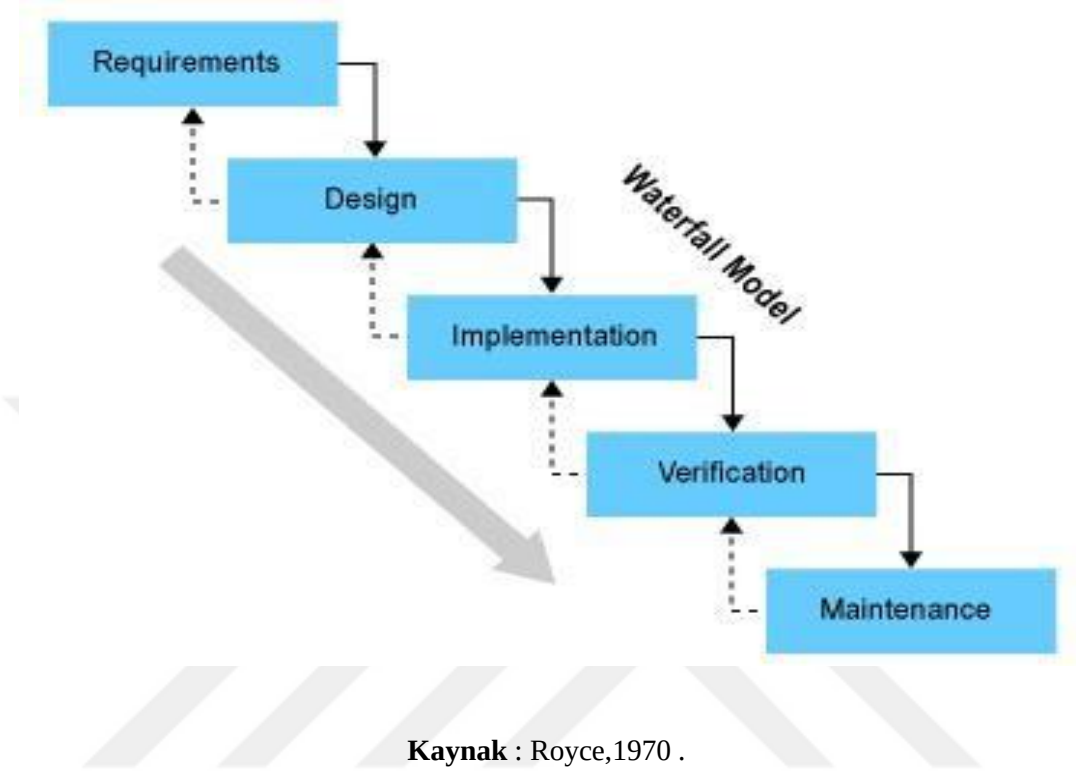
1.1 GELENEKSEL YÖNTEMLER

Klasik yazılım süreçlerinde tüm ihtiyaçlar projenin başında müşteriden temin edilmekte ve müşteriden istekleri alınmaktadır. Tüm ihtiyaçları detaylı bir şekilde tespit edilmiş projeler açısından uygun olan bu yöntem, projenin geliştirilme sırasında ihtiyaçlarda meydana gelebilecek muhtemel olan projeler açısından uygun değildir. Klasik modeller arasında en klişeleşmiş olan yazılım geliştirme modeli şelale modeli olarak dikkat çekmektedir [1] . Çalışmanın bu bölümünde ele alınan yazılım geliştirme modelleridir. Bu kapsamda klasik ve modern yazılım geliştirme modellerinin yapısı hakkında kısaca bilgi verilecektir.

1.1.1 Şelale(Waterfall) Model

Şelale modeli beş adımda gerçekleşen işlemlerdir. Bu işlemin adımları; gereksinim analizi, tasarım, kodlama, test ve bakım aşamalarıdır. Bir adım bitmeden diğer adıma geçiş olmamaktadır ve her aşamanın bitiminde dokümantasyon işlemi yapılmaktadır [1] .Bu adımların en sonunda test aşaması vardır ve bu saffaya gelindiğinde genellikle projenin sonlarına gelmiş olursunuz. Tam da bu aşamada müşteriden bir talep gelirse bunun hem maliyeti yüksek hem de her aşama adım adım olmasından kaynaklı müşterinin taleplerine hızlı cevap vermek mümkün olmayabilir. Son aşamaya bırakılan testler sırasında belirlenen hataların giderilmesi projenin sonunda giderek zorlaşmakta ve maliyet artmaktadır. Her bir aşama tamamlandıktan sonra diğer aşamaya geçilmekte, arkadan gelen aşamada sorun çıkarsa bir önceki aşamaya da dönülerek hataların giderilmesi mümkündür.

Şekil 2.1 Waterfall Model



Kaynak : Royce,1970 .

1.1.1.1 Avantajları

- Kullanmak, uygulamak ve anlamak basittir
- Yönetmek basittir.
- Projenin safhaları ayrı olduğundan iş bölümü ve iş planı projenin en başında net Bir şekilde bellidir. Bu durumun sonucu projenin yönetimini de oldukça kolay hale getirir.
- Şelale Modeli çok küçük ve gereksinimleri çok iyi belirlenmiş projelerde olumlu sonuçlar verir[4].

1.1.1.2 Dezavantajları

- Kompleks ve nesne yönelimli projeler için uygun değildir.
- Devam eden ve uzun projeler için zayıftır.
- Projede oluşabilecek her türlü değişime elverişsiz, katı bir modeldir. Yapılan her bir değişiklik maliyeti de büyük oranda etkiler.
- Müşterinin memnuniyetini sağlamak oldukça zordur; gelişim ve değişime açık

bir model olmaması etkilidir.

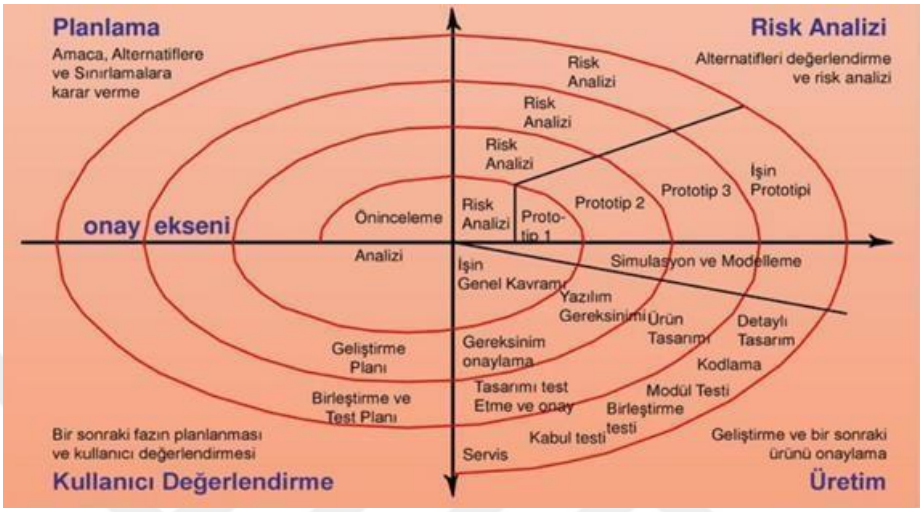
- Model adım adım ilerlediği için ürün son adımda tamamlanır, ihtiyaçların iyi tanımlanmadığı, müşterinin talebinin anlaşılmaması, projenin bittiğinde iptale Kadar giden bir yığın farklı gerginliklere de neden olur, sorunlar zinciri yaşanır [4].

1.1.2 Sarmal (Spiral) Model

1986'larda Barry Boehm' in ortaya çıkardığı bir modeldir. Spiral (Sarmal) model esas olarak dört temel aşamadan oluşmaktadır[5].İlk aşama sistem kavramsal olarak tanımlanır ve uygulama alternatifleri, nesneleri ile kısıtlılıkları hakkında inceleme ve araştırma yapılır. İkinci aşama ise alternatifler tanımlanarak, riskler tespit edilir. Geliştirme ve tanımlama işlemleri üçüncü aşamada söz konusu olur ve ardı sıra gelen fazların planlaması ise dördüncü aşamada söz konusu olur [6]. Sarmal modeli aynı safhalara geri dönülmesinin bir zorunluluk olduğunu vurgular, Sarmal modeli şelale modelinde yok sayılan riskleri dikkate alır. Proje çevrimlere ayrılır ve her bir çevrimin riskleri tek tek ele alınır. Çağdaş modellere oldukça yakındır. .

- *Planlama* : Üretilecek ara ürün için planlama, amaç belirleme, bir önceki adımda üretilen ara ürün ile bütünleştirme.
- *Risk Analizi* : Risk faktörlerinin araştırılması ve risklerin tespit edilmesi,
- *Üretim* :Ara ürünün üretilmesi
- *Kullanıcı Değerlendirmesi* :Ara ürün ile ilgili olarak kullanıcı tarafından yapılan sınama ve değerlendirmeler. Risk Analizi Olgusu ön plana çıkmıştır. Her döngü bir fazı ifade eder[7].

Şekil 2. 2. : Risk Analizi Gösterimi



Kaynak: Boehm,1998

1.1.2.1 Avantajları

- Kullanıcı kişiler sistemi erken görebilirler.
- Geliştirme küçük parçalara bölünür ve riskli kısımlar önce gerçekleştirilir.
- Pek çok yazılım modelini bünyesinde bulundurur.
- Riske duyarlı yaklaşımı potansiyel zorlukları engeller.
- Seçeneklere erken dikkate odaklanır.
- Hataları erken gidermeye odaklanır.
- Yazılım-donanım sistemi geliştirme için bir çerçeve sağlar.

1.1.2.2 Dezavantajları

- Küçük ve düşük riski olan projeler için pahalı bir yöntemdir.
- Karmaşık ve komplekstir.
- Spiral ucu sonsuza gidebilir.
- Ara adımlar fazladır bu sebeple fazla dokümantasyon gerektirir.

- Büyük ölçekte projeler gerektirir.
- Kontrat tabanlı yazılım yöntemlerine uymaz.
- Yapılan yazılımın içten geliştirileceğini varsayımlar.
- Adım adım anlaşma esnekliğini sağlamaz.
- Kurumsal risk değerlendirme deneyimine dayanır.
- Yüksek riskli işlere yoğunlaşmak, yüksek riskli işlerin doğru belirlenmesini gerektirir [8].

1.1.3 V Model

V-model(yazılım geliştirme) şelale(waterfall) modelinin gelişmiş hali olarak düşünülebilecek bir yazılım geliştirme süreci sunar. Doğrusal bir yönde ilerlemek yerine, süreç adımları kodlama evresinden sonra yukarıya doğru eğim alır ve/veya tipik V şeklini oluşturur [9]. V modeli iki taraflı bir modeldir. Bir tarafı doğrulama aşamalarını gösterirken diğer taraf ise geçerleme aşamalarını gösterir. Modelde doğrulama aşamasının her bir düzeyi, geçerleme aşamasındaki bir düzeye karşılık gelecek şekilde model oluşturulmuştur. Doğrulama aşamalarında gereksinim analizi, sistem dizaynı, mimari tasarım ve modül dizaynı bulunurken; geçerleme aşamalarında ise birim test, entegrasyon testi, sistem testi ve kullanıcı kabul testleri mevcuttur. V modelde testler sırasında belirlenen noksanlıkların ve hataların giderilmesi amacıyla ilgili seviyedeki doğrulama seviyesine geçiş yapılmaktadır [10].

Şekil 2.3. : V Model Gösterimi



Kaynak : Şeker, 2015

1.1.3.1 Avantajları

- Verification ve validation planları erken aşamalarda vurgulanır.
- Verification ve validation sadece son üründe değil tüm teslim edilebilir ürünlerde uygulanır.
- Proje yönetimi açısından takibi ve kullanımı basittir.

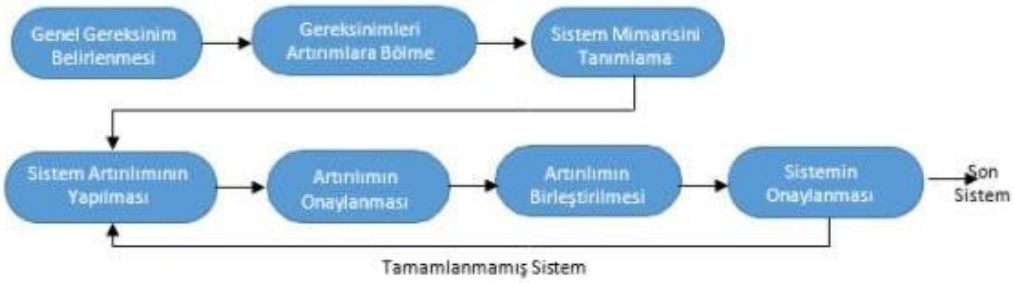
1.1.3.2 Dezavantajları

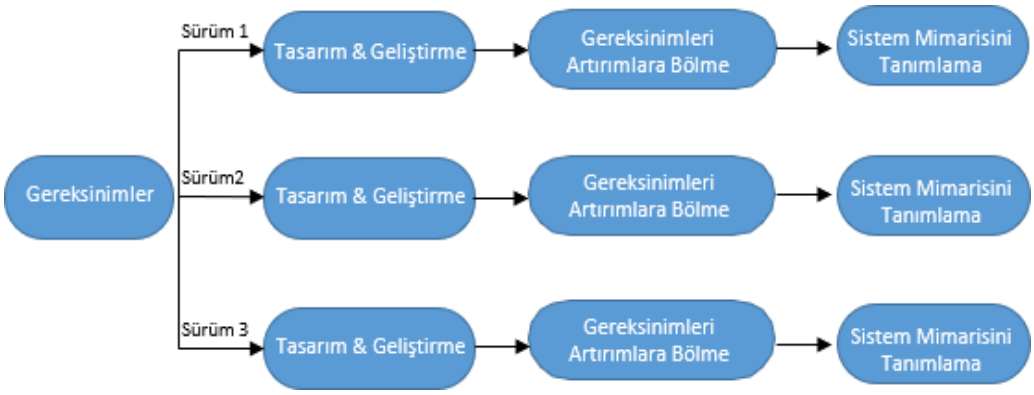
- Aynı zamanda gerçekleştirilebilecek olaylara kolay fırsat tanımaz.
- Fazlar arasında tekrarlamalar kullanılmaz.
- Risk çözümleme ile ilgili olayları içermez
- Yazılım da diğer sistemler gibi zamanla gelişir ve büyür.
- Geliştirme devam ettikçe iş ve ürün gereksinimleri de değişkenlik gösterir
- Son ürüne ulaşma düz bir çizgi ile anlatılmaz [10].

1.1.4 Artırımlı Model

Bir döngü söz konusudur. Bu modelin ilk evresi planlamadır. Planlamalar hazırlanır ve döngüye hazır hale gelinir akabinde giriş yapılmış olur. İstek analizleri, analizler, tasarımlar ifa edilir ve uygulama aşamasına geçilir. Uygulamadan sonra iki alternatif durum söz konusu olacaktır. İlk alternatif canlıya geçiş durumudur. Bu alternatif seçilir ve muhtemel hatalar her aşamada olduğu gibi bu evrede de görülür. Fakat bahse konu hatalar testlerin bir parçası olarak kabul edilebilir ve testler devam ettirilir. Tespit edilen hatalar sistemin gelişmesine imkân sağlar. Testlerden ve canlıdan elde edilen bilgilerin ışığında yeniden değerlendirme yapılır. Yazılım geliştirme süresince müşteriden ilave ihtiyaçlar ortaya çıkabilir. Bu ihtiyaçlar döngüler ile artırımlı olarak her bir iterasyonda projeye katılarak edilerek müşterin yeni istekleri karşılanmış olup müşteri memnuniyeti de sağlanmış olur[11] . Artımlı model yazılım geliştirmenin az sayıda çalışanla işin yapılmasını sağlar ve bu konuda diğer modellere göre üstünlük sağlamış olur. Özellikle çalışanlar da her artırım geçildiğinde uygulama alanına ilişkin daha fazla deneyim kazanmış olurlar. Bu modelde bir tarafta üretim yapılırken diğer taraftan kullanım yapılır. Önceki modellerde ürünlerdeki farklılıklar dikkate alınmazdı. Bu model doğal olarak yinelemelidir. Yeniden kullanılabilir bir ürün, fonksiyonellik sağlamış bir şekilde tüm döngülerin sonunda ortaya çıkar.

Şekil 2.4. : Artırımlı Model Gösterimi





Kaynak: Şeker, 2015

1.1.4.1 Avantajları

- Sistem için gerekli olabilecek bütün ihtiyaçlar müşterilerle birlikte belirlenir.
- İhtiyaçların önem durumuna göre teslim edilecek artımlar belirlenir.
- Öncelikle en önemli gereksinimleri karşılayan çekirdek bir sitem oluşturulur.
- Erken artımlar prototip gibi davranarak, gereksinimlerin daha iyi anlaşılmasını katlı sağlar.
- Tüm projede başarısız olma riski azalır, böl ve yönet yaklaşımıdır.

1.1.4.2 Dezavantajları

- Artımları tanımlamak için tüm sistemin tanımlanmasına gerek vardır.
- Gereksinimleri doğru boyuttaki artımlara atamak bazen zorlayıcı olabilir.
- Deneyimli personele ihtiyaç vardır.
- Artımların kendi içlerinde tekrarlamalara imkân vermez [8].

1.2 ÇEVİK

Yinelemeli(İteratif) geliştirme temelli bir grup yazılım geliştirme yöntemine dayanmaktadır. Burada ihtiyaçlar ve çözümler kendinden örgütlü olan farklı grupların ortak çalışmasıyla olgunlaşır. Agile metotları çoğunlukla sık denetim ve adaptasyonun teşvik edildiği disipline edilmiş bir proje yönetim sürecidir. Ekip çalışmasının, kendinden örgütlenmenin ve izlenebilirliğin özendirildiği liderlik felsefesini, kaliteli yazılımların çabuk biçimde geliştirilmesinin hedeflendiği en iyi mühendislik uygulamalarını ve yazılım geliştirme ile müşteri ihtiyaçlarını ve firmanın amaçlarını bir araya getiren business yaklaşımı destekler. Agile metotları bir yazılım geliştirme yaklaşımı değil, geliştirme süreçleri topluluğudur. Değişimin nedenlerine baktığımızda :VUCA' dan bahsedebiliriz[12].

Değişkenklik (Volatility):Çevredeki değişkenliğin sıklığıdır.

Belirsizlik (Uncertainty): Geleceği iş açısından görememe durumudur.

Karmaşıklık (Comlexity): Mevcut durumda verilerde veya durumda nedensellik analizinde zorlukların çekilmesi.

Muğlaklık (Ambiguity) :Belirsiz ortamlar

Agile Ne Değildir? Mevcut çalışma şeklimizdeki sorunları bir anda bitirecek sihirli bir değnek değildir. Bir süreç, varılacak bir hedef, amaç, görsel boardlar, post-itler, standart kullanılan araçlar yada tek bir metot değildir.

Agile Nedir? Çevik kelime anlamı Türk Dil Kurumu şu şekilde tanımlamıştır: "Kolaylık ve çabuklukla davranan, tetik, atik." Çevik bir felsefe, bir düşünce biçimidir. Bu felsefe;

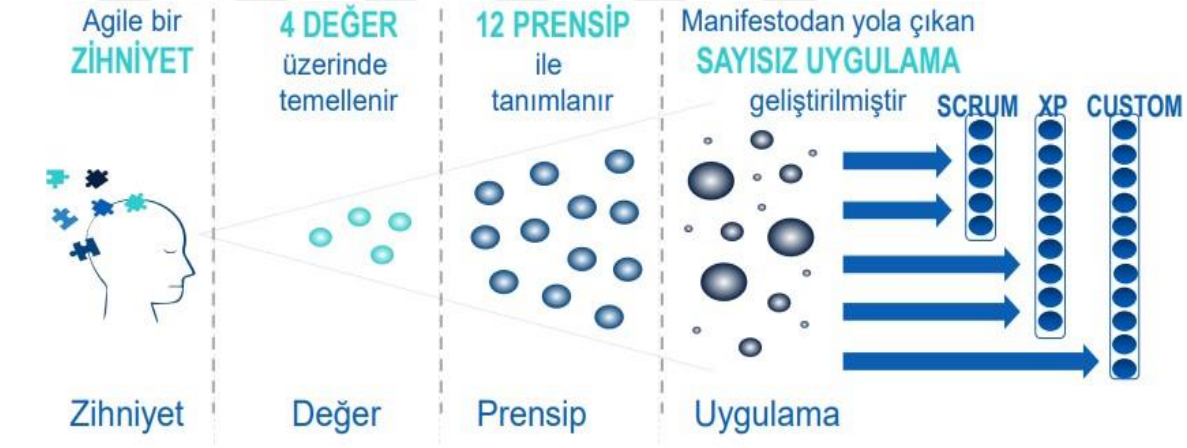
- Müşteriden alınan geri bildirimlere göre, değişen koşullara hızlı adapte olup, güvenli proje geliştirme yaklaşımı ile kısa aralıklarla değer üretmeye odaklanan bir yaklaşımdır.
- Değerler ve prensipler bütünü (Manifesto)
- Agile varılacak bir hedef değil, her gün gelişmeyi hedefleyen bir yaklaşımdır.
- Kısa planlanmış döngüler, farklılaşan gereksinimler ve birden ortaya çıkabilecek farklılıklar için organizasyonunuza esneklik olanak tanır.

- Müşteriler, ürünü oluşturmak ve çıkan problemleri kolayca çözebilmek için etkin bir biçimde geribildirim sağlar.

- Ekip katılımına öncelik verilmesi sayesinde kurulan iletişim sorunların daha çabuk çözülmesini sağlar[13].

  Proje hızla ilerletilirken ortaya çıkan her sorun bir sonraki döngüde daha iyi bir çözüm için basamak olarak kullanılır.

Şekil 2.5. : Agile



Kaynak: ttps://www.scrum.org/

Ürünü parçalara ayrıştır, her bir parçaya "değerine" göre öncelik belirle. Geliştirilen her parçayı gözden geçirerek birden fazla iterasyon ile final ulaş. Bunu yaparken müşteriyle yüz yüze iletişim, şeffaflık, yetkilendirilmiş ekiplere daha fazla özerklik ve sorumluluk verilmeli, ekiplere güven duyulmalı.

Standlisk grubunun 2015 yılında yaptığı araştırma sonucuna göre büyük ölçekli ve küçük ölçekli projelerin agile ve waterfall olmasına göre başarı sonuçlarını içeren tabloya bakıldığında küçük projeler de agile projenin walterfalle göre 1,3 daha başarılı olduğu. Büyük projeler de ise Agile'ın 6 kat daha başarılı olduğu ve genel proje dağılımlarında ise 3 kat daha başarılı olduğunun sonucunu paylaşmıştır. Bu tablo da yer alan challenged olan kısımlarda ise proje tamamlanmıştır ancak

baştaki planlanan gibi olmamıştır. Yani ya projenin bitiş tarihi ötelenmiş ya da proje kaynak planlaması başındaki gibi olmamış ama sonuç olarak proje teslim edilmiştir.

Tablo 1.1: Çalışma Metodu Ve Büyüklüğe Göre Proje Durumu

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects (x3)	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects (x6)	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects (x4)	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects (x1.3)	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

Kaynak: https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf

1.2.1 Çevik (Agile) Tarihçe

Fredrick Taylor ilk seri üretimi ilk dillendiren kişi olurken, Ford'da seri üretimi yapan ilk firmadır.

1986 yılında iki Japon akademisyen tarafından Harvard Business Review'de "The New New Product Development Game" isimli makale yayımlandı.

1993 yılında ise Scrum doğuyor. Jsutherland, Ken Schwaber ve Jeff Sutherland tarafından Agile prensiplerin yazılımda kullanımıyla ilgili araştırmalar yapıldı.1995 yılında Ken Schwaber ve Jeff Sutherland yaptıkları araştırmalar sonucunda oluşturdukları yaklaşımı "Scrum" adıyla yayınladı. Scrum'ın yayınlanması 2001' de, 17 yazılım geliştirme uzmanı, yazılım geliştirme projelerinde üretkenliği arttırmak için iki günlük uygulamalı toplantı gerçekleştirdiler. Bu toplantı sonunda 4 maddelik bir değerler topluluğu ortaya çıkarıldı. Agile Manifesto'nun yazılım geliştirme uzmanları tarafından yayınlanmış olması bunun yalnızca mühendislik alanında kullanılacağı

anlamına gelmemektedir. Yönetimsel alanlarda da Agile uygulamak mümkündür[14]. The Toyota Way: İlk doğuşu Toyota firmasında 1950'li yıllarda keşfedilen bir iş yapış biçimi. Temelinde yatan şey; değer katan şeyleri maksimum yapalım, değer katmayan şeyleri azaltalım ki, bizim çok fazla verimli çalışma yapmamızı sağlayabilelim. Değer katmayan, çöp, israf vs. yaptığımız şeyleri ne kadar yalınlaştırırsak; işimize değer katan şeylere o kadar zaman ayırabilir iz. Kaizen-"Sürekli Gelişim" ve "Saygı ile Ekip Çalışmasını içeren İnsana Saygı kavramları" üzerine inşa edilmiştir Jidoka (Otonomasyon) ve Andon (Lamba).

3 temel prensibi: MURI(aşırı yükleme), MURA(dengesiz iş yükü), MUDA(değer katmayan). Toyota Üretim Sistemi'nin sanayi sektörüne kattığı en temel ilkedir. Bu ilkeye göre her şeyi ancak müşterinin istediği anda ve miktarda üretmek esas olacak, gereksiz stoklar tümüyle ortadan kaldırılacak prensibine dayalıdır. Prensibe göre stok israftır ve sistemde hiçbir israfa yer yoktur. Her üretim aşaması ancak bir sonraki aşamanın ihtiyaç duyduğu zamanda ve miktarda üretim yapmak üzere Kanban adı verilen kartlarla tetikleniyordu. Bu mantık tedarikçi firmalar zincirinde de uygulanmıştır. Sistemde üretim müşterinin talep etmesi üzerine olur, stoklar asgariye indirilmiştir ve kaynaklar daha etkin bir şekilde kullanılabilir şekilde oluşturulmuştur. Toyota fabrikalarında geliştirilen üretim sisteminin altında yatan düşünme biçimi, Lean Enterprise Institute'un kurucusu James Womack ve arkadaşları tarafından incelenerek "Yalın Düşünce" adlı kitapta öğrenilebilir ve tekrarlanabilir bir sistem yaklaşımı olarak sunulmuştur [14].

1.2.2 Çevik Bildiri (Agile Manifesto)

Agile Manifesto dört maddeden oluşuyor. Manifestoya göre; bireyler ve aralarındaki etkileşimler süreçler ve araçlardan, çalışan yazılım kapsamlı dokümantasyondan, müşteri ile iş birliği sözleşme pazarlıklarından, değişime karşılık vermek bir plana bağlı kalmaktan daha önemli olduğunu söylüyor[8].

Şekil 2.6. : Agile Manifesto



Kaynak :scrum.org

Tablo 1.2. : Agile Manifesto

İngilizce	**Türkçe Tercümesi**
Individuals and interactions over processes and tools	Bireyler ve iletişim süreç ve araçlardan önce gelir.
Working software over, comprehensive documentation	Çalışır durumda olan program kapsamlı dokümantasyondan daha önceliklidir.
Customer collaboration over contract negotiation	Müşteri ile birlikte çalışmak sözleşme görüşmelerinden daha önceliklidir.
Responding to change over following a plan	Değişikliklere uyum sağlamak bir planı izlemekten daha önemlidir.
That is, while there is value in the items on the right, we value the items on the left more.	Sağ bölümde yer alanlar (İngilizce metinde altı çizili olmayanlar) değer taşımakla birlikte, sol bölümde altı çizili olan değerler daha önemlidir.

Bildiriyi İmzalayanlar		
Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Kent Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

Kaynak: http://agilemanifesto.org [Erişim Tarihi: 21.03.2021]

1.2.3 Yalın Düşünce

Amaç: Müşteriye değer sunumunun sürekli iyileştirilmesi ve bu değeri üreten çalışanların güçlendirilmesidir. Üretkenliği, kaliteyi ve verimliliği artırır. İsrafı azaltır ve maliyeti düşürür. Temel prensipleri; değer odaklı, sürekli iyileştirme ve akıştır. Burada maximum değer ve/veya bunu yaparken minimum israfla yapmak önemlidir. Yazılım geliştirme süreçlerinde kısmen biten işler, ek özellikler, ekstra işlemler/adımlar, task değiştirme, yazılım hataları, tekrar eden işler israflardır.

1.2.4 Çevik (Agile) Prensipler

Çeviğin temel prensibi; değişimi hoş karşılamak, değişen gereksinim taleplerini, geliştirme sürecinin sonlarında gelseler bile, her zaman hoş karşılamaktır. Çevik yaklaşımlar, müşterinin rekabet avantajını koruyabilmesi için değişimi yönetir ve yönlendirir. Sık teslimat yapılması söz konusudur. Çalışan yazılım birkaç haftada bir veya en fazla birkaç ayda bir, olabildiğince sık biçimde müşteriye kullanıma sunulabilir. Teslim periyodu ne kadar kısaysa o kadar daha iyidir. Çalışan yazılım ile müşteri tatmini hedeflenir. Birincil öncelik, müşteri için değerli olan yazılım bileşenlerini mümkün oldukça erken ve sürekli olarak geribildirim alarak, müşteriyi memnun etmektir. Müşteri ile sürekli birlikte çalışma söz konusudur. Proje devam ettikçe, iş gereksinimlerinin sahibi olan iş birimleri ve yazılım geliştirme ekibi her gün yakın, hatta yan yana çalışmalıdır. İnsanları motive et. Projeler motivasyonu yüksek kişiler tarafından hazırlanmalı, ihtiyaç duyulacak her şey için destek verilmelidir. Projeyi hazırlayan kişilere aldıkları işlerini bitireceklerine dair güven söz konusu olabilir. Yüz yüze iletişim kurulmalıdır. Bilgilerin yazılım ekibine aktarılmasının ve yazılım ekibinin kendi içerisindeki iletişimin en etkin ve verimli yolu yüz yüze konuşmaktır. İlerleme ölçütü olarak çalışan yazılım, bir projenin ne kadar ilerlediğ inin göstergesi, çalışan yazılımdır.

Sürdürülebilirliği önemse, çevik yaklaşımlar yazılım geliştirmenin sürekliliğinin korunmasını önemser. Tüm bunları yaparken teknik kaliteden ödün verme, teknik kalitenin sürekli korunması çevikliği güçlendirir. Basitleştir. Basitlik; yapılmayan işleri artırma sanatı; zaruridir. Kendi Kendine organize ol, en iyi mimariler, en iyi gereksinimler ve en iyi tasarımlar, kendi kendine

organize olabilen takımlar tarafından üretilir. Yansıt ve ilerlet. Takım, sabit aralıklarla bir araya gelip, nasıl daha etkin olabileceğini düşünür ve sonra kendi ahengini kendi yaratır. Periyodik olarak böyle yapan bir takım, kendini her defasında bir adım öteye taşır[15].

1.2.5 Organizasyonel Çeviklik

1.2.5.1 Müşteri Çevikliği

Çevik yazılım geliştirme yaklaşımlarının genel ilkesi, müşterinin de geliştirme takımıyla beraber aynı takım içerisinde çalışmasını gerekli kılar. Başlangıçta ayrıntılı gereksinim analizi yapılmamasının getirdiği sorunların giderilmesi, dizayn ve kodlama yapılabilmesi, analizdeki eksikliklerin doldurulması amacıyla müşteri ile birlikte çalışmak oldukça önemlidir. Müşteri profili bu çalışma şeklinde uygun değil ise çevik yöntemlerin kullanımı olma ihtimali söz konusu olacaktır. Örnek olarak gereksinimlerin yalnızca müşterinin üst yönetimince bilinmesi ve bu yöneticilerin de zamanlarının sınırlı olması sebebiyle birlikte alışmanın imkânı bulunmayabilir. Diğer taraftan müşteri uzak bir noktada olabilmekte ya da beraber çalışma konusunda istekli olmayabilmektedir. Bu durumda gerçek ihtiyaç sahibi müşterinin yerine projenin verilmiş olduğu alt yüklenici firma temsilcileriyle birlikte çalışılması gereken durumlar olabilir. Bu durumlarda projenin başarısı olumsuz anlamda etkilenebilecektir[15].

1.2.5.2 Şirket İçi Çeviklik

Çevik şirketlerin en temel iki özelliği esnek ve hızlı olmalarıdır. Hızlılık işi tamamlama süratini ifade ederken, esneklik ise çevrede yaşanan değişime uyum sağlama yeteneğidir. Çeviklik ise, süratli bir biçimde çalışmak ve çevrede meydana gelen değişikliklere süratle uyum sağlamak anlamına gelir. Diğer bir ifadeyle çeviklik aynı zamanda hem süratli hem de esnek olabilmekt ir. Hızlılık, şirketin faaliyetlerini sürdürme, fırsatlar kovalama, sorunları çözme, değişime yanıt verme bağlamında hızlı olmasını ifade eder. Bir siparişin tamamlanma zamanı, bir müşterinin şikâyetinin yanıtlanma zamanı, bir satın alma konusunda yöneticinin olur verme zamanı, yatırım kararının alınma zamanı için geçen süreler hızlılık ile alakalı durumları anlatmaktadır. Diğer taraftan esneklik ise, piyasada meydana gelen farklılıkları önceden görebilme yeni oluşan şartlara ve müşterilerin beklentilerine hızlı bir şeklide uyum sağlama yeteneğidir. Piyasa şartlarındaki ani değişimler ile başa çıkma yöntemleri geliştirmek, müşteriye almayı istediği ürünün niteliklerini seçme hakkı tanımak esnek olmak kavramıyla ilgilidir. Çevik şirketler, tutkulu patronları, dinamik

personeli, esnek yapılarıyla süratli karar alabilme, ani eyleme geçebilme, hızlı bir şekilde iş bitirebilme ve değişimlere hızlı şeklide uyum sağlayabilme yeteneklerine sahiptirler. Kurumsal anlamda çeviklik, ürün teslim zamanından, şikâyet yanıtlama süratine, pazarda yaşanan değişimlere göre tekrardan organize olabilme esnekliğinden, bir doğal afet anında hızlı karar alabilmeye kadar birçok açıdan kendini gösterebilmektir. Çevik şirketler süratli, heyecanlı ve her an her şeye hazır halde yaşamlarını sürdürürler. Çevik bir şirkette çalışanlarda bulunan iş bitirme azmi ve pozitif enerji fark yaratmaktadır. Ataletli şirketler ise çevik şirketlerden farklı bir yapı, kültür ve çalışma biçimine sahiptir. Bu şirketler esnek değil katı, kendini yenileyen değil kendini tekrarlayan, motive değil demotive ve dinamik değil durağan bir tarza sahiptir. Bir şirkette, piyasa şartlarının değiştiği, işlerin eskiden olduğu gibi olmadığı, yeni şartlarda ayakta kalabilmek için yeni bir şeyler yapması gerektiği görülür, bu değişimi başaramamanın bedelleri ise şirket tarafından ödenir, ancak yine de yapılması gerekenler yapılmaz ise kurumsal anlamda atalet hali meydana gelmiş demektir. Şirketlerin çevikliği ile yazılım geliştirme kapmasında çeviklik arasında benzerlikler olduğu görülmektedir. Benzerlik açısından her ikisinde de değişime ayak uydurma, esneklik ve hızlılık kavramlarının ön plana çıktığı ve başarı açısından önem arz eder[15].

Organizasyonların çevik yaklaşımda olmalarının aşağıdaki faydaları vardır.

i. Kaos ortamında dengeyi korumak,

ii. Pazara çıkış hızı ve rekabet avantajı sağlamak,

iii. Operasyonel verimlilik ve karlılığı artırmak,

iv. İnovasyon kabiliyetini artırmak,

v. Değer katan ve yetenekli çalışanı şirkete kazandırabilmek,

vi. Dijital dönüşüme zemin hazırlamak [15].

1.2.6 Çevik Yaklaşımlar

Çevik süreçler, çalışmanın önceki bölümlerinde ifade edilen öncelik ve ilkeleri benimseyen, manifestoyu kabul eden ve çalışma yöntemlerini çeviklik yaklaşımıyla oluşturmuş olan süreçler olarak ifade edilmektedir. Zamanla, bir meta model biçiminde kabul edebilecek olan çevik süreci uygulayan farklı çevik süreç çeşitleri meydana gelmiştir. Bu süreçlerin pek çoğu çevik bildiriden

sonra meydana gelmiştir. Fakat bazı süreçlerden ise çevik bildiri öncesinde de faydalanılmıştır. Bu günlerde oldukça yaygın biçimde faydalanılan en önemli çevik süreç çeşitleri çalışmanın bu bölümünde detaylıca bahsedilecektir[8].

1.2.6.1 Scrum

Scrum'ın ilk tam uygulaması, Jeff Sutherland, John Scumniotales ve Jeff McKenna'nın Easel Corporation'da Scrum'ı uyguladığı 1993 yılında gerçekleşti ve Scrum bir çevik yazılım geliştirme sürecidir. Takeuchi ve Nonaka'nın yeniliğe yeni bir bütünsel yaklaşımı rugby sporuyla karşılaştırdığı ve tüm takımın "mesafeye gitmeye çalıştığı" 1986 HBR makalesi "Yeni Yeni Ürün Geliştirme Oyunu" ndan ilham aldılar. “Topu ileri geri geçiren bir birim ” 1995 yılında, Jeff Sutherland ve Ken Schwaber, Austin, Teksas'ta düzenlenen Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) '95 konferansında "SCRUM Geliştirme Süreci" başlıklı bir makaleyi ortaklaşa sundular. 2001'de Sutherland, Schwaber ve on beş meslektaşı Snowbird, Utah'da bir araya geldiler ve Agile Manifestosu'nu hazırladılar. O zamandan beri, Sutherland, Schwaber ve gittikçe büyüyen Scrum uygulayıcıları topluluğu, dünyanın her yerindeki şirketlerde on binlerce yüksek performanslı ekip oluşturmaya devam etti. Scrum uygulamalarına diğer önemli katkılar, işin miktarını ve ekiplerin hızını ölçmenin bir yolu olarak hikaye noktalarının geliştirilmesinin yanı sıra müşteri odaklı iş hedeflerini tanımlayan bir araç olarak kullanıcı hikayelerinin geliştirilmesiyle Mike Cohn tarafından yapılmıştır. Scrum daha çok proje yönetim ve planlamayla ilgili değerlere ve pratiklere odaklanmaktadır. Yazılımın nasıl, hangi şekilde hazırlanması gerektiği ile ilgili ayrıntıları içermemektedir. Scrum bağımsız bir çevik metot şeklinde bilinmemelidir. Birçok proje kapsamında Scrum, Unified Process (UP) veya Extreme Programming (XP) ile desteklenmektedir [31].

1.2.6.2 Extreme Programming (XP)

XP, 1990’ların sonuna doğru Kent Beck, Ron Jeffries ve Ward Cunningham tarafından Chrysler şirketi için hazırlanan bir proje sonrasında oluşturulmuş bir çevik yazılım geliştirme sürecidir [17]. XP, Scrum’ dan esinlenerek geliştirilen bir yöntemdir. Fakat XP, Scrum’ın tersine daha ziyade yazılım geliştirme pratikleri üzerinde durmaktadır. Buna bağlı olarak XP ve Scrum projelerde bir arada kullanılması söz konusudur. Özellikle değişimin hızlı, ihtiyaçların belirsiz olduğu orta ve küçük çaplı yazılım geliştirme takımlarına yönelik çözümler ortaya koyar. Esas gayesi müşterilerin

taleplerine cevap veren kalitesi yüksek ürünü sağlamak amacıya müşteriye güven vermektedir[17]. XP ile oluşturulan çevik süreçte müşteri ve ihtiyaçları merkezi bir rol oynamaktadır. Yazılım esnasında XP ile tam belirli olmayan ve çabuk değişikliğe uğrayan müşteri gereksinimlerine ayak uydurulabilir. Bu konvansiyonel yazılım metotlarda mümkün değildir, çünkü proje öncesi müşteri ihtiyaçları en son detayına yazılı bir şekilde çalışılmıştır. Oluşan bir dokümantasyon baz alınarak, yazılım gerçekleştirilir. Proje ilerledikçe müşteri tarafından yapılması istenen değişiklikler in maliyeti çok yüksek olacaktır, çünkü mevcut yapı (tasarım — design) istenilen değişikler in yapılmasını engelleyebilir ya da yeni bir yapılanmaya gidilmesi gerekebilir. XP, kullanıldığı projelerde formalite ve bürokrasinin mümkün en az seviyeye çekilmesine önem verir. Çevik olabilmek için az yükle yola çıkılması gerekmektedir. Bu yüzden proje öncesi geniş çapta tasarım ve dokümantasyon oluşturulmasına izin verilmez. Genel olarak test pratikleriyle dikkati üzerine çeker. XP'nin çığır açan pratikleri; Otomatik Test (Test Automation), Sürekli Entegrasyon (Continuous Integration), Test Güdümlü Geliştirme (Test Driven Development) konusundaki uygulamalardır. Çevik model karakteri taşımayan projelerde dahi bu pratiklerden yararlanılması kazanç sağlayacaktır. XP, yazılım geliştirme konusunda esneklik ve kolaylık sağlamak adına 12 ayrı uygulama öngörmektedir. Bu uygulamalar [17].

i. Metafor,
ii. Kısa ve küçük aralıklı sürümler (release)
iii. Planlama oyunu
iv. Basit tasarım
v. Yeniden yapılandırma
vi. Önce test
vii. Ortak kod sahiplenme
viii. İkili programlama
ix. Haftada 40 saat çalışma
x. Sürekli tümleştirme (continuous integration)
xi. Kodlama standartları
xii. Ekipte müşteri [17].

1.2.6.3 Kanban

Bir sinyal ya da işaret anlamına gelmektedir. O sinyal görüldüğünde bir sonraki parçayı üretmenin zamanının gelmiştir. Kanban : Kanban methodu iş yerindeki problemleri ortaya çıkarmaya ve Lean (yalın) yaklaşımı ile onları çözümlemesine yardımcı olur. Chris Matts'in ifadesiyle "Kanban

metodu size odanın ışıklarını açıp ve içerideki kaplanları (tehlikeleri, riskleri, dengesizlikleri) görmenizi sağlar ama kaplanlarla savaşacak olan sizsiniz.

1.3 SCRUM

SCRUM, Rugby (Ragbi) oyununda 8 oyuncudan oluşan gelen bir takımdır. Takımda her bir oyuncu diğer oyuncuyla topu oyun alanında hareket ettirmek maksadıyla birlikte hareket ederler ve bu takımlar yakın olarak birbirleriyle çalışırlar[18]. Takımda roller tanımlıdır ve takım üyeleri tek bir hedefe odaklanmış bütün halde çalışırlar, tam bir takım olarak hareket ederler.

Yazılım mühendisliği alanında SCRUM adı verilen çevik yazılım geliştirme metodu, yazılım takımlarının benzer amaçları taşımaları ve benzer stratejiyle hareket etmeleri sebebiyle Rugby takımından ilham alınarak SCRUM adını almıştır. Bu yazılım geliştirme yaklaşımı, karmaşık ortamlarda yazılım geliştirme amacıyla iteratif ve arttırımlı bir yazılım geliştirme sürecidir. SCRUM herhangi bir karmaşık durumda organizasyonun oluşan kaotik durumu adapte olacak biçimde geliştirmeyi devam ettirmesine imkân tanır. SCRUM metodolojisi, esnek bir geliştirme sürecini devam ettirebilmek amacıyla tasarlanmıştır. SCRUM çerçevesi, ürünün çıktısını planlamayı ve projenin ilerlemesi esnasında ortaya çıkabilecek değişkenleri yönetebilmeye imkân tanımaktadır. Organizasyonlara herhangi bir zaman aralığında proje ve teslimatları üzerinde değişiklik yapabilme imkânı tanımakta ve bunun sonucunda ise imkan dahilinde olan en kısa zamanda en uygun ürünün ortaya konmasını sağlamaktadır. SCRUM mevcut olan bir sistem ya da ürün prototipi için yönetme, geliştirme ve bakım metodolojisi olarak tanımlanır [31]. Scrum sadece bir çerçeve değil aynı zamanda da düşünce biçimidir. Anlaması kolay, basit ancak uzmanlaşması gibi satranç gibi zordur. Scrum, insanların, takımların ve organizasyonlar ın karmaşık problemler için adapte edilebilir çözümler aracılığıyla değer üretmesine yardımcı olan basit bir çerçevedir.

Kısaca,

- Bir Product Owner'ın karmaşık bir sorun için Product Backlog'daki işleri sıraladığı,

- Scrum Team'in, seçilen işleri Sprint boyunca bir değer Increment'ine çevirdiği,

- Scrum Team ve paydaşların sonuçları inceleyip ve bir sonraki Sprint için düzeltme yaptığı
- Ve bunları tekrarladığı bir ortamı yaşatmak için bir Scrum Master gereklidir.

Scrum basittir. Olduğu gibi deneyin ve felsefesinin, teorisinin ve yapısının hedeflere ulaşmaya ve değer yaratmaya yardımcı olup olmadığını belirleyin. Scrum çerçevesi bilinçli olarak eksik bırakılmıştır, yalnızca Scrum teorisini uygulamak için gerekli kısımlar tanımlanmıştır. Scrum, onu kullanan insanların ortak zekâsı üzerine inşa edilmiştir. Scrum'ın kuralları insanlara ayrıntılı yönergeler vermek yerine onların ilişkilerine ve etkileşimlerine rehberlik eder. Çerçeve dahilinde çeşitli süreçler, teknikler ve yöntemler kullanılabilir. Scrum, mevcut uygulamaları toparlar veya bazılarını gereksiz görüp dışarıda bırakır. Scrum, mevcut yönetimin, ortamın ve çalışma tekniklerinin göreceli etkililiğini görünür kılar. Bu sayede iyileştirmeler yapılabilir. Scrum'ın teorisi şudur: Deneysellik ve yalın düşünce. Deneysellik, bilginin deneyimden ve gözleme dayalı karar vermekten geldiğini ileri sürer. Yalın düşünce ile israfı azaltır ve temel unsurlara odaklanır. Scrum da, tahminlemeyi en üst seviyeye çıkarmak ve riski ve engelleri kontrol etmek için yinelemeli (iterative) ve artımlı (incremental) bir yaklaşım kullanır. Scrum, işi yapmak ve gerekli becerileri paylaşmak veya edinmek için toplam olarak tüm becerilere ve uzmanlığa sahip insan gruplarını bir araya getirir. Scrum, kapsayıcı bir etkinlik olan Sprint içinde gözlem ve adaptasyon için 4 resmi etkinliği bir araya getirir. Bu etkinlikler, Scrum' ın deneyselliğinin taşıyıcı kolonları olan şeffaflık, gözlem ve adaptasyonu uyguladığı için işe yarar [31].

1.3.1 Scrum'ın Değerleri

Şekil 2.7. : Scrum'ın Değerleri



Kaynak : https://www.scrum.org/

Scrum'ın başarıyla uygulanması insanların aşağıdaki beş değeri ustalıkla yaşatma becerisine bağlıdır: Taahhüt, Odak, Açıklık, Saygı ve Cesaret Scrum Team, hedeflerine ulaşmayı ve birbirin i desteklemeyi taahhüt eder. Scrum Team'in birincil odak noktası, hedeflerine doğru mümkün olan en iyi ilerlemeyi sağlamak için o Sprint'te aldıkları işlerdir. Scrum takımı ve paydaşları yapılacak işleri ve bu işleri yaparken karşılaşılan zorluklarla ilgili açıktır. Scrum takımının her bir üyesi diğer takım üyelerine yetkinliklerinde ve bağımsız insanlar olarak saygı duyar ve çalıştıkları insanlar tarafından da bu şekilde saygı görürler. Scrum Team'in üyeleri, doğru olanı yapma, zor problemler üzerinde çalışma cesaretine sahiptir. Bu değerler Scrum Team'e çalışmaları, eylemleri ve davranışları ile ilgili yön verir. Alınan kararlar, atılan adımlar ve Scrum'ın kullanılma biçimi bu değerleri azaltmamalı veya zayıflatmamalı, güçlendirmelidir. Scrum Team üyeleri, Scrum etkinlikleri ve eserleri ile çalışırken bu değerleri öğrenir ve keşfeder. Bu değerler Scrum Takımı ve birlikte çalıştıkları kişiler tarafından içselleştirildiğinde, Scrum'ın deneyselliğinin taşıyıcı kolonları olan gözlem, şeffaflık ve adaptasyon canlanarak tüm organizasyon için güven ortamı oluşturur. Scrum takımı ve paydaşları, tüm işlere ve işin gerçekleştirilmesindeki zorluklara açık olmayı kabul eder[16].

1.3.2 Scrum Takımı

Scrum'ın küçük bir gruptan oluşan temel birimidir. Takım bir Scrum Master(Scrum Uzmanı), bir Product Owner (Ürün Sahibi) ve developers (geliştiricilerden) oluşmaktadır. Bu takımda hiçbir rolde hiyerarşi yoktur, kimse kimsenin yöneticisi değildir. Bu takımdan oluşan alt takımlar da yoktur takım tektir. Takım her seferinde ortak bir hedefe, ürün hedefine odaklanır. Takım oluşurken rastgele birkaç kişiden oluşan kişilerden takım kurulmaz. Takım kurulurken oluşan takım kaynaşmış ve kendi alanlarından uzman profesyonel kişilerden oluşur. Oluşan takım çapraz fonksiyoneldir, yani üyeler her bir Sprint'te değer yaratmak için tüm becerilere sahiptir. Bunu şöyle düşünebiliriz, takım bir değer yaratacak ve başka bir takıma ihtiyaç duymadan kendi içerisinde test, yazılım, analiz geliştirmesini kendisi yapabilmelidir. Genellikle çapraz fonksiyon denilince takımlar ya da organizasyon bunu herkes her işi yapar olarak algılayabiliyor ancak burada kişilerin çapraz fonksiyonluğu değil takımın çapraz fonksiyon olması önemlidir. 2010 Kasım Scrum Guide'ı güncellenmeden önce takım self-organizedir sözü geçerli idi. Bu takım kendi kendine organize olmalı anlamına geliyordu. Ancak 2010 Kasım güncellenen guide şunu söylüyor: kendi kendilerini yönetirler, yani kimin neyi, ne zaman ve nasıl yapacağına kendileri karar verirler (2010 Kasım-Scrum guide) [16].

Scrum Takımı, hızlı davranabilecek kadar küçük ve bir Sprint içinde anlamlı bir işi tamamlayacak kadar büyük, ideal olarak 10 veya daha az kişidir. Genellikle, daha küçük takımların daha iyi iletişim kurduğunu ve daha üretken olduğunu gözlemlenmiştir. Takımdaki kişi sayısı çok büyürse, her biri aynı ürüne odaklanan birden fazla kaynaşmış Scrum Team olarak yeniden düzenlenmeyi düşünmelidirler. Bu nedenle, aynı Ürün Hedefi'ni, Product Backlog'u ve Product Owner'ı paylaşmaları gerekir. Scrum Team, paydaşlarla iş birliği, çalıştırma, deney, bakım, doğrulama , araştırma ve geliştirme ve yapılacak diğer şeyler gibi ürünle ilgili tüm faaliyetlerden sorumludur. Kendiişlerini yönetebilmek için organizasyonlar tarafından kurulan, yetkilendirilen ve desteklenen takımlardır. Sprintler halinde sürdürülebilir bir hızda çalışmak takımın odaklanmasını ve tutarlılığını geliştirir. Tüm Scrum takımı her Sprint'te değerli ve faydalı bir artış (increment) oluşturmaktan sorumludur [16].

1.3.3 Roller

1.3.3.1 Developers (Geliştiriciler)

Takımda her sprint kullanılabilir bir artırımlı bir kısım oluşturmayı taahhüt etmiş kişilerdir. İhtiyaç duydukları beceriler genellikle geniştir ve çalışma alanlarına göre değişiklik gösterebilir. Takımın ihtiyaç duyduğu yetkinlikleri geliştirmesinde developersları organizasyonlar desteklemelidir.

Sorumlulukları:

- Sprint planını oluşturmak ve planları her gün sprint hedefine göre uyumlamak.
- Her sprintin artırılmasından tüm üyeler sorumludur ve kararlara katılırlar.
- Herkes sprint hedefini bilir.
- Başarısından sorumludur. Bir bitti tanımına bağlı kalarak kaliteden sorumludur, ürünün NASIL yapılacağına karar verir.
- Ürün iş listesini oluşturmak ve sürdürmek için ürün sahibi ile birlikte çalışır ve iş birliği yapar.
- Kendi kendini yönetir ve çok işlevlidir.

Ancak önünde engeller ile karşılaşabilirler. Bireysel çalışan üyeler ve vizyon ve ekip hissenin oluşmaması, bağımsız çalışma ve dış bağımlıklar, sprint hedefi hakkında net olmamak, sorulara açıklayıcı ve net olmayan bir ürün sahibine sahip olmak.

1.3.3.2 Product Owner (Ürün Sahibi)

Takımın çalışması sonucu oluşan ürünün değerini en üst seviyeye çıkarmaktan sorumludur. Ürünün değerini en üst seviyeye nasıl çıkarılacağı ise bireyler, takımlar ve organizasyonlar arasında farklılık gösterebilir. Product Owner, aynı zamanda Product Backlog'u etkili bir şekilde yönetmekle sorumludur, şunları içerir:

- Ürün Hedefi'ni oluşturmak ve açıkça iletişimini yapmak,
- Product Backlog maddelerini oluşturmak ve açıkça iletişimini yapmak,

- Product Backlog maddelerini sıralamak,
- Product Backlog'un görünür, şeffaf ve anlaşılır olmasını sağlamak

Product Owner, yukarıdaki işleri kendisi yapabilir veya bu yükümlülüğü başkalarına devredebilir. Ancak sorumluluk her zaman Product Owner'dadır. Product Owner'ın başarılı olabilmesi için aldığı kararlarının organizasyondaki herkesten saygı görmesi çok önemlidir. Product Owner'ın kararları, Product Backlog'un içeriği, sıralaması ve Sprint Review'da sunulan gözlemlenebilir Increment aracılığıyla görünürdür. Product Owner tek bir kişidir; bir komite olamaz. Product Owner, birçok paydaşın ihtiyaçlarını öncelikleindirip Product Backlog'a yansıtabilir. Product Backlog'u değiştirmek isteyenler bunu Product Owner'ı ikna etmeye çalışarak yapabilirler. Son söz product ownerdadır [18].

Product Owner sprintin ilerlemesinden sorumlu kişi değildir, takımın deneticisi, iş dağıtıcısı ya da takımın yöneticisi de değildir. Geliştiricilerin önündeki engeller gibi Product Owner'ın da önünde engeller olabilir. Bunların en belirgin olanları: Takım ile işbirliği yapamamak, Scrum Master ile sorumlulukların karıştırılması, müşterilerin ihtiyacını anlamamak, hem Scrum Master hem de Product Owner olmak.

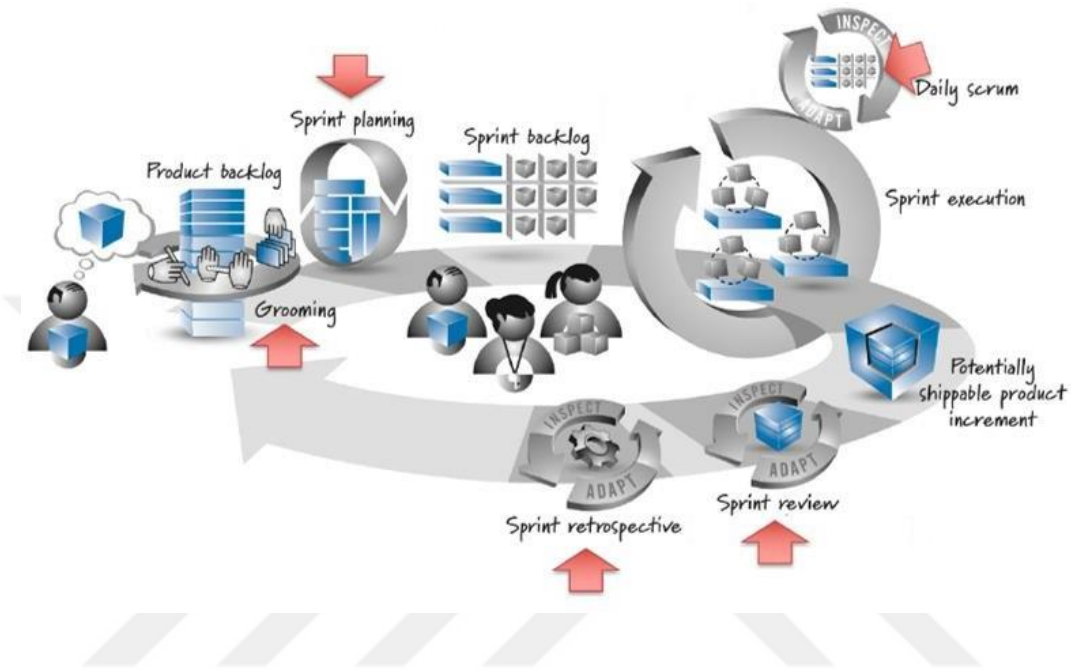
1.3.3.3 Scrum Master (Scrum Uzmanı)

Scrum Kılavuzu'nda tanımlandığı gibi Scrum'ın benimsenmesinden sorumlu olan kişidir. Bunu hem takım, hem de organizasyon içindeki tüm kişilerin Scrum teorisini ve pratiklerini anlamasına yardımcı olarak yapar. Takımın etkililiğinden sorumludur. Bunu, pratiklerini Scrum çerçevesi dahilinde iyileştirmesini sağlayarak yaparlar. Scrum Masterlar, takıma ve organizasyonun bütününe hizmet eden gerçek liderlerdir. Scrum Master, takım üyelerine kendini yönetme ve çapraz fonksiyonluluk konularında koçluk yapmaktan, bitti tanımına uyan en yüksek değerli ürün oluşturmasındaki odaklanmasına destek olmak, takımın ilerlemesinin önündeki engellerin kaldırılmasını sağlamaktan, tüm etkinliklerinin gerçekleşmesini, bunların pozitif, üretken olmasını ve zaman sınırları içinde tutulmasını sağlamaktan. Product Owner'a ise ürün hedef tanımına, ürün iş listesinin sıralanması konusunda yardımcı olmaktan sorumludur [18].

Engelleri ise; kendisini geliştirmek istemeyen bir takımın olması, ürün sahibi ile kötü bir ilişki içerisinde olmak, çok hiyerarşik bir yapının olması, çevik olmayan ekipler ile olan blokelerdir.

1.3.4 Scrum Etkinlikleri (Scrum Ritülleri)

Şekil 2.8. : Scrum Etkinlikleri



Kaynak : https://www.scrum.org/

1.3.4.1 Sprint

Fikirlerin değerlere dönüştüğü Scrum'ın kalbi olan etkinliktir. Tutarlılık yaratması adına sprint süreleri sabit değildir. Bu süre maksimum bir ay ya da daha az olabilir. Bir sprint bittiğinde direk diğer sprint başlar. Sprint süresi çok uzun olursa sprint hedefi geçersiz hale gelebilir , karmaşıklaşabilir ve sprint için riskler artabilir bu sebeple sprint süresi gözlemlenmelidir[18].

Bir sprintin içerisinde; planlama, review, daily, refinement, retrospective etkinlikleri ve ürün hedefini yakalamak için gerekli olan tüm işler vardır. Sprint süresince o sprintin hedefini tehlikeye sokacak hiçbir değişiklik yapılmaz. Kalite hedefleri düşmez, ürün/hizmet iş listesinde iyileştirmeler yapılabilir. Bir sprint, sprint hedefi kaybolmuşsa, anlamını yitirmişse iptal edilebilir. Ancak bu iptal yetkisi sadece Product Owner' dadır[18].

1.3.4.2 Sprint Planlama

Ürün sahibi tarafından yazılmış iş listesidir. O sprintte yapılacak işlerin planlandığı, sprint ile başlayan etkinliktir. Bu planlama tüm scrum takımının iş birliği ile yapılır. Ürün sahibi, katılımcıların en önemli ürün iş listesindeki maddelerini ve hedef ile ilişkisini hazır olarak getirmelidir bu etkinliğe gelirken. Bu etkinliğe takım tavsiye almak amacıyla ile başkalarını da davet edebilir. Bu planlama etkinliğinde bu sprintin neden değerli olduğu, sprintte nelerin tamamlanabileceği, seçilen işlerin nasıl yapılacağı ve sprintin hedefi nedir bu konular konuşulur. Planlama etkinliğini süresi, bir aylık bir sprintler için sekiz saat ile sınırlıdır. Eğer takımın daha kısa sprintleri var ise bu sürenin daha kısa olması beklenir[18]. Süre uzuyorsa takım bu planlama toplantısının neden uzun sürdüğü ile ilgili düşüncelerini retrospective toplantılarında konuşabilir.

1.3.4.3 Daily Scrum (Günlük Toplantı)

Sprintin hedefine doğru ilerlemeyi gözeden geçirmek, planlana alınmış işlerin gereken şekilde düzenlemesinin yapıldığı sprint backloga adaptasyonun sağlandığı, günlük 15 dk.yı geçmeyen toplantıdır. Karmaşıklığın olmaması adına her gün aynı yer ve zamanda olması önemlidir. Tüm takım katılabilir ancak ürün sahibi sprint backlog üzerinden aktif iş çekmemişse sadece gözlemci olarak katılabilir. Bu toplantıda takımın önünde bir engel var ise bu paylaşılmayabilir. Scrum'ın değerlerinden olan şeffaflık burada önemlidir takım şeffafça bu toplantıyı gerçekleştirebilir.

1.3.4.4 Sprint Review

Sprintin çıktısını gözden geçirmek ve gelecek sprintteki alınacak ön görülen işlerin konuşulduğu toplantıdır. Takım yaptıkları işlerin çıktılarını tüm paydaşlara sunar ve ürünün hedefine doğru ilerlemede hangi noktadalar bu durum görüşülür. Bu toplantıya katılan katılımcılar çıktılar hakkında geri bildirim verebilirler. Takımda bu verilen geri bildirimlerden ürün iş listesinde iyileştirme fırsatı görürse değerlendirebilir. Bir sprintte bu toplantının süresi bir aylık sprint süresi var ise 4 saati geçmemelidir. Daha kısa sprint süresi var ise bu saat daha az olmalıdır. Bu toplantıda da şeffaflık ve güven önemlidir. Takım şeffafça paylaşım yapmalı ve paydaşlarda takımın paylaşımlarına güven duyulmalıdır.

1.3.4.5 Sprint Retrospective

Scrum takımının geçen sprint özelinde, kişileri, etkileşimleri, süreçleri, araçları gözden geçirir. Nelerin iyi gittiğini, ne tip problemlerle karşılaştıklarını ve bu problemleri nasıl çözdüklerini yada çözemediklerini konuşurlar[19]. Takımın iyileştireceği çıktılar var ise bunları çıkarırlar hatta bu çıktılar takımın sprint backloguna da eklenebilir. Bir aylık sprint süresi olan takımlarda bu etkinliğin süresi maksimum 3 saattir. Ancak daha kısa süreli sprinti varsa takımın 3 saatten daha kısa sürmelidir bu etkinlik. Sprint, retrospective ile sona erer ve hemen yeni sprint başlar.

1.3.5 Scrum Eserleri

Yapılan işi ya da üretilen değeri temsil etmektedir eserler. Ana amaç şeffaflıktır. İlerlemeyi ölçmek, şeffaflık ve odaklanmayı artırmak için taahhüt içerir. Scrum eserleri; ürün iş listesi için ürün hedefi, sprint backlog için sprint hedefi, artırımlı için ise bitti tanımıdır[19].

1.3.5.1 Product Backlog (Ürün iş listesi)

Ürünü geliştirmek için nelere ihtiyaç duyulduğunun sıralı ve ürün hizmet yaşadığı sürece yaşayan bir listesidir. Scrum takımı tarafından üstlenilen işlerin tek kaynağıdır. Scrum takımı tarafından bir Sprint içerisinde bitirilebilecek ürün iş maddeleri, Sprint Planning etkinliğinde seçilmeye hazır kabul edilir. Genellikle iyileştirme aktiviteleri sonrasında bu derecede şeffaflık elde ederler. Product Backlog iyileştirme, Product Backlog maddelerini daha küçük, daha kesin parçalara ayırma ve detaylandırma aktivitesidir. Bu, açıklama, sıra ve büyüklük gibi detaylar eklemek için gerçekleştirilen sürekli bir aktivitedir. Bu bileşenler genellikle çalışma alanına göre değişkenlik gösterir. İşi yapacak olan Developers işlerin büyüklüğünü belirlemekten sorumludur. Product Owner, detaylarda nelerin yapılıp nelerden vazgeçilebileceğini anlamalarına yardımcı olarak Developers'ı etkileyebilir. Karakteristiği geri bildirimlere göre güncellenebilir esnektir. Gereksinimler ne kadar ayrıntılı olursa, ürün tanımını o kadar etkiler.

1.3.5.2 Sprint Backlog

Sprint Hedefi (goal-neden), o sprint için seçilen Product Backlog maddeleri (ne) ve Increment'i ortaya çıkarabilmek için yapılacak, hayata geçirilebilir bir plandan (nasıl) oluşur. Sprint Backlog, Developers tarafından, Developers için hazırlanan bir planlamadır. Sprint Hedefi'ne ulaşmak için Developers'ın Sprint sırasında başarmayı planladıkları işin son derece görünür, gerçek zamanlı bir

resmidir. Sonuç olarak Sprint Backlog, daha fazla bilgi edindikçe Sprint boyunca güncellenir. Daily Scrum'da Developers'ın ilerlemeyi gözlemleyebilmesi için yeterli ayrıntıya sahip olabilmelidir. Sprint hedefi her sprint için tektir. Developers'ın bu sprint hedefi ile bu taahhüdü olmasına rağmen, bunu başarmak için gereken iş açısından esneklik sağlar. Sprint Hedefi, takımın tek bir hedefe odaklanmasını ve tüm takımın anı işi sahiplenmesini yaratarak, Scrum takımı ayrı hedeflerin yerine birlikte çalışmaya teşvik eder. Sprint Hedefi, Sprint Planning etkinliği sırasında yaratılır ve ardından Sprint Backlog'a eklenir. Developers Sprint sırasında çalışırken, Sprint Hedefi'ni akıllarında tutarlar. İşler beklenenden farklı çıkarsa, Sprint Hedefi'ni etkilemeden Sprint Backlog'un kapsamını müzakere etmek için Product Owner ile iş birliği yaparlar. Artırımlı ise, Product Hedefi'ne doğru atılan somut bir adımdır. Her Increment, önceki Incrementlerin üzerine eklenir ve tüm Incrementlerin birlikte çalışmalarını temin edecek şekilde tamamen doğrularlar. Değer sağlamak için Increment kullanılabilir halde bulunmalıdır. Bir Sprint içinde birden fazla Increment oluşturulabilir. Incrementlerin toplamı Sprint Review'da sunularak deneysellik desteklenir. Ancak bir Increment Sprint sona ermeden önce de paydaşlara teslim edilebilir. Sprint Review değerin teslim edilebilmesi için bir mekanizma gibi görülmemelidir. Bitti Tanımı'na uymadığı sürece yapılan iş Increment'in parçası olarak kabul edilemez[19].

1.3.5.3 Bitti Tanımı (Defination Of Done)

Hizmet/Ürün için gerekli kalite gereksinimlerini sağladığı halinin hem takımın hem de organizasyonun kabul ettiği resmi bir tanımıdır. Bir ürün iş listesi (Product Backlog) maddesi Bitti Tanımı'nı karşıladığı anda Increment doğar. Bitti Tanımı, Increment'in parçası olarak hangi işlerin bittiğine dair herkes için ortak bir anlayışı sağlayarak şeffaflık yaratır. Bir ürün iş listesi maddesi Bitti Tanımı'nı karşılamıyorsa, kullanıma alınamaz, üretime dâhil edilemez ve hatta Sprint Review'da sunulamaz. Bunun yerine, ilerideki sprintlerde değerlendirilmek üzere Product Backlog'a geri döner. Bir Increment için Bitti Tanımı organizasyonun standartlarının bir parçasıysa, tüm Scrum takımları olabildiğinde olarak bitti tanımlarına uyabilmelidir. Organizasyonel bir standart değilse, Scrum takımının ürüne/hizmetine göre uygun bir Bitti Tanımı oluşturulabilmelidir. Developers'ın Bitti Tanımı'na uyması gerekir. Bu maddeler bir takımın işinin kalitesidir. Bir hizmet/ürün üzerinde birlikte çalışan birden fazla Scrum Takımı mevcutsa, bu takımlar karşılıklı olarak aynı Bitti Tanımı'nı birlikte tanımlamalı ve bu tanımlara birlikte uyulabilmelidir. Bir sprint içerisinde geliştirilen görevlerin tamamlanmış sayılması için yerine

getirmesi gereken maddeleri kapsar. Bu,yapılması gerekenler olarak kabul edilir. Ekip, gelişmelerin istenen kaliteyi karşılaması için hangi koşulların karşılanması gerektiğini kabul eder. Kaliteyi belirler [19].

1.3.6 Scrum Metrikleri

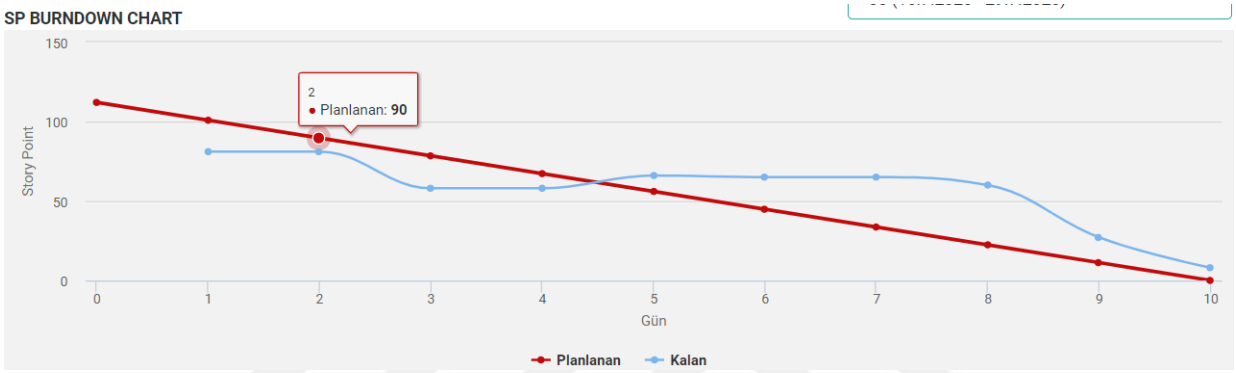
Agile takımları başarıya götüren ve çevikliğin temelinde olan maddelerden birisi de "Deneyimsel" süreçtir. Deneyimsel süreç oluşturabilmek içinde de Plan – Do – Check – Act döngüsünü uygulayabilmek çok önemli olmaktadır. Önce plan yapılır, bu plan hayata geçirilir, metrikleri takip ederek değerlendirmeler yapılır ve süreçte iyileşme yapılacak noktaları var ise yeniden planlamalar yapılabilir. Yapılan planlamaların alınan kararların doğru ilerleyip ilerlemediğini görebilmek için veriler ön plana çıkmaktadır. Scrum takımlarının kullanabileceği verilerden birisi de metriklerdir. Metriklerin başında ise Story Point gelir. Story Point takımdan takıma değişen, herhangi bir birimi olmayan bir metriktir. Herhangi bir birimi olmadığından Story Point ifadesini farklı isimlerden de karşılaşabiliriz. Velocity, Jelibon, armutlar, elma, bonibon vb gibi kullanımları vardır. Organizasyonlar/kurumlar Story pointlere istediği ismi koyabilirler isim de kesinlik yoktur. Bu değerler takımdan takıma bile değişmesi söz konusu olabilir. Story Point, fibonacci serisi kullanılarak PBI'lara verilen tahminleme yöntemlerinden birisidir. Her PBI'ya fibonacci serisinden Story Point tahminleme yapılır, sprint sonunda ise tamamlanan PBI'a DONE kabul edilmektedir. Takımında burada kullanacağı metrik de DONE olan PBI'larıdır. Story Point metriğini kullanarak elde edilen burndown chart'lar vardır. Fibonacci serisinin yanında t-shirt size(XS,S,M,L,XL), yada 1'den 100'e kadar sayı serileri de kullanılabilir. Takımın başta hangi metriği kullanıyorsa onunla ilerlemesi önemlidir. Spirinte bu kullanılan değerlerinden bir diğerine geçek doğru değildir [20].

1.3.6.1 Story Point Burn Down Chart (SP Burn-Down)

Her sprint için ayrı tutulur. Takımın DONE(o sprintte tamamladığı) yaptığı işleri gösteren burndown chart'tır. Story point burn down metriğindeki amaç takımın bir sprint boyunca kalan PBI'ların story point cinsinden gösterilmesidir ve takımın sprint içerisindeki gidişatının yorumlanmasını sağlamak amaçlanır Takımın sprint içinde waterfall yaklaşımla mı ilerledi, uzmanlık alanlarının paralel bir şekilde çalışıyor mu veya işlerin daha küçük parçalara bölünme ihtiyacı var mıydı, yoksa küçük parçalara bölebildi mi sorularının değerlendirildiği bir chart

olmaktadır[20]. Takımın hızını gösterir. Ancak bu chart sadece takıma özeldir, çünkü her takımın hızı birbirinden farklıdır ve ekipler arası karşılaştırılma yapılmamalıdır. Ekibin çalışır bir yazılım üretme yeteneğidir. SP Burn-down metriği o takımın bir sprintte ne kadar büyüklükte iş alabileceğinin tahminlemesinde yardımcı olmaktadır. Bu metrik ile takımın zaman içerisinde nasıl geliştiğini de incelemek için kullanılır.

Şekil 2.9. :SP Burndown Chart

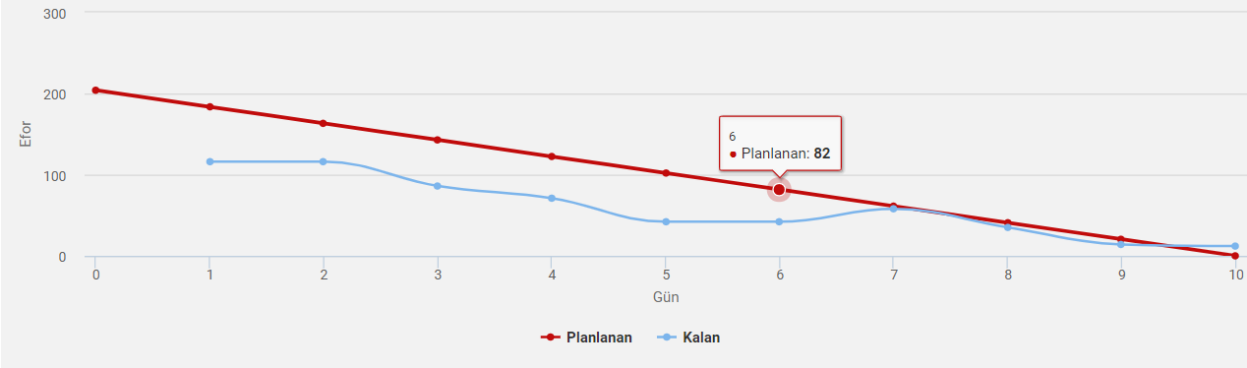


Kaynak : Teze konu olan şirketin xxx isimli takımı

1.3.6.2 Efor burndown chart

Sprint bazlı her takım için ayrı çizilen bir burndown chart'tır. Takımların o sprinte aldıkları PBI'ların altındaki task'lar için saat cinsinden belirttikleri eforlar kullanılarak çizilerek oluşan grafiklerdir. Task'larda ki toplam kalan saatler çizerek takımın sprint başında belirlediği toplam task eforunun günlük olarak takibini sağlamaktadır. Daily sırasında yapılabilir, grafiğin güncel tutulması bu metriği kullanacaklar için önemlidir. Bu metrik sayesinde takımın sprint başında planlanan zamana sprint boyunca uygun olup olmadığı gözlemlenebilir ve takımın ihtiyacı olan kararı alır. Örneğin; takımın toplam kalan task eforu baseline'ın üzerinde ilerliyorsa takım önceliği en az olan PBI'ların NOT DONE olacaklarını belirleyip önceliği yüksek olan işlere odaklanabilirler veya baseline'ın altında ilerliyorsa da sprinte, product ownerın da bilgisi dâhilinde ek işler almaya karar verebilir [20].

Şekil 2.10. : Efor Burndown Chart

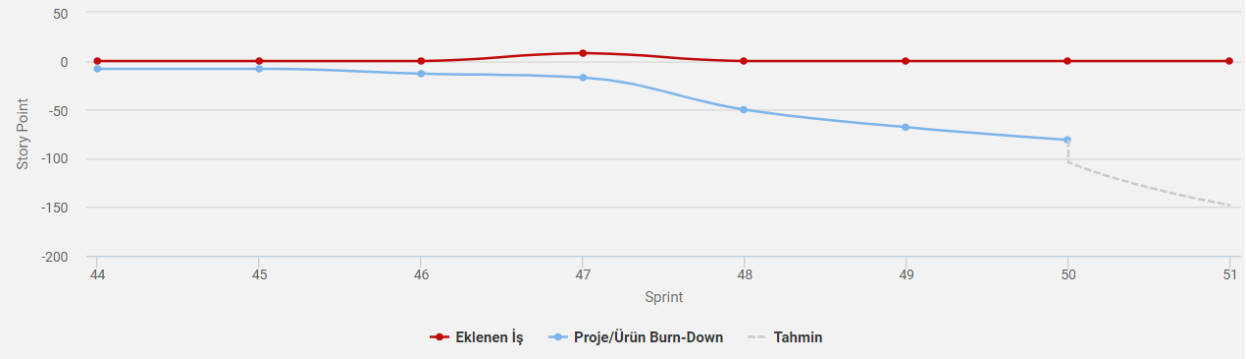


Kaynak : Teze konu olan şirketin xxx isimli takımı

1.3.6.3 Product burndown chart:

Takımın sorumluluk alanındaki ürünün kapsamında nelerin yapılacağının ve ne zaman tamamlanacağının öngörüsünün verildiği garafikler/chartlardır. Takımın tamamlayacağı bir hedef vardır, bu hedefin takıma ne kadar uzaklıkta olduğunu ve hedefe de ne kadar hızla yaklaştığının göstermede product burndown chart yardımcı olur.. Bu grafiği okuyarak ürünün ne zaman tamamlanacağı öngörüsünü tüm paydaşlarla paylaşılır ve dinamik bir chart olduğundan da değişimleri hızlı bir şekilde gösterir [20]. Böylece takım scrumın da değerlerinden olan şeffaflık değerinin de uygulayarak bu metrikle paydaşlar bir karar alması gerekiyorsa alması gereken olası kararlar içinde veri sağlamış olur.

Şekil 2.11. : Product Burndown Grafiği

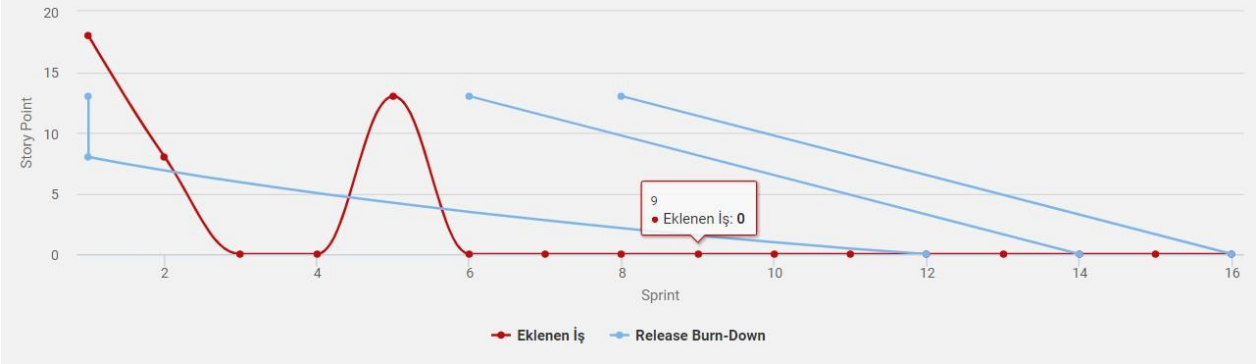


Kaynak : Teze konu olan şirketin xxx isimli takımı

1.3.6.4 Release burndown chart:

Product burndown charttaki tüm ürünün kapsamının yerine sadece release edilecek kapsamın ele alınarak yapılan chart'tır. Product burndown charttaki kullanılan bilgilerin sadece release edilecek kısımlarını release kapsamı için düşünülebilinir [20]. Burn Down Chart ayrıca, scrum değerlerine de hizmet etmektedir. Bu metrik ile takım ne kadar işi kaldı, tamamlayabilmek için ne kadar zamana ihtiyacı var net bir şekilde görebilir ve böylece scrumın değerlerinden birisi olan taahhüt ettikleri işlere odaklanmayı sağlar.

Şekil 2.12. : Release Burndown Chart



Kaynak : Teze konu olan şirketin xxx isimli takımı

1.4 KANBAN

Kanban: profesyonel hizmetlerde işleri/çalışmaları organize etme ve yönetme, değer akışını optimize ederek daha verimli, etkin ve ön görülebilir süreç akışını sağlayan metottur. İşaret, sinyal, büyük görsel board, sipariş kartı olarak tanımlanır. İşlerimizin nasıl işlediğini bize gösterir. Ürün teslim sürecini iyileştirir. Sürekli gelişimi harekete geçirir. Amaca uygun bir iş modeli geliştirir.Kanban da amaç, süreçlerinizdeki aksaklıkları tanımlamak ve sistemdeki aksaklıkları düzeltmektir, bu sayede işlerin en uygun hızda veya verimde gerçekleşebilir olmasını görünür kılar. Kanbanın kullanımı 1940'lı yılların sonlarından itibaren Toyota tarafından ilk defa uygulanmıştır. İlk Kanban sistemi, Japonya'da Toyota otomotiv için Taiichi Ohno (Endüstri Mühendisi ve İş Adamı) tarafından geliştirilmiştir. Üretimin her aşamasında işleri ve envanterleri en iyi şekilde kontrol etmek ve yönetmek, akışı görselleştirmek olarak basit bir planlama sistemi olarak oluşturulması amaçlanmıştır. Kanban'ın gelişmesinin temel nedeni, Toyota'nın Amerikan otomotiv rakiplerine kıyasla yetersiz üretkenliği ve verimsizliğiydi. Kanban ile Toyota, hammadde, yarı mamul ve nihai ürünlerin maliyet yoğun envanterini azaltırken, üretkenliği artıran esnek ve verimli tam zamanında üretim kontrol sistemine ulaştı. Her istasyonda üretim sistemini kontrol altında tutulması, ara stokların önüne geçmesi ve sistemde israf varsa bunların ortaya çıkması hedeflenir. Kanban ürünü ve bilgi akışını yönetir, buna ek olarak stok yönetimi yapılmasının gerektirmez. Sadece güvenlik stoğu vardır. Güvenlik stoğu ise kanban kadardır. Kanban sisteminin uygulandığı organizasyonlarda tam zamanında üretim felsefesiyle ve çekme sistemine göre üretim gerçekleştirilmektedir. Kanban tam zamanında üretimin (Just In Time-JIT) bir parçası üretimi kontrolde tutmak için kullanılan görsel bir metottur. Push sistem (Çekme sisteminin) bir parçasını içerir. Kanban sistemine göre sonraki iş önceki işten sadece kullandıklar ı kadar ve zamanda malzeme çekmesidir. Kanban sisteminde genelde kart kullanılır, ancak bazen kutu da kullanılmaktadır [21]. Kanban sisteminde kullanılan kartlar, ekibin kendi tasarımına göre değişir ve bu kartta iş ile ilgili bilgiler taşınır. İşin detayı, ne yapılacağı, ne zaman yapılacağı, kimin yapacağı, işin tipi ve kimin yapacağı ile ilgili bilgiler içerebilir.

Kanban sistemi optimum olacak şekilde üreticiden sonuncu kullanıcıya değin bütün değer akışını çek eder. Bu sayede, üretim sürecinde yer alan farklı basamaklarında israfı engeller. Kanban, işleyişi daima takip edilmelidir. Üretim süreçlerini zayıflatabilecek etkenlerden kaçınmak için dikkat gösterilmelidir. Amacı, daha kısa teslimat süresi ile daha yüksek verim elde edebilmektir. Zamanla Kanban, çeşitli üretim sistemlerinde kullanılmaya, daha etkili olmuştur [22]. Kanban, Taiichi Ohno tarafından imalat endüstrisinde tanıtılırken, konsepti 2004 yılında genel olarak bilgi teknolojileri ve yazılım geliştirme ilk uygulayan David J. Anderson'dır. David, çalışmalarını Taiichi Ohno'nun çalışmaları üzerine inşa etti, Eli Goldratt, Edward Demmings, Peter Drucker ve diğerleri, çekme sistemleri, kuyruk teorisi ve akış gibi kavramlarla Kanban Metodunu tanımlayacaklardır. Kanban ile ilgili ilk kitabı 2010 yılında yayınlanan "Kanban: Teknoloji İşiniz İçin Başarıyla Evrimsel Değişim"dir, demiştir.

Kanban sistemi itme sistemini tersine olarak işlemesidir, tersten okumak gerekir. Sonraki aşamadan iş çekildikçe bir önceki aşamadakine sıra gelir ve oradan iş çekilir. Arada kalan blokları bu sayede ortadan kaldırmayı sağlamaktadır. Aralarda bekleme süresini aza indirgemektir hedef. Müşteriden istenilen talebi üretime almak için çekme sistemiyle geriden işleyerek bu şekilde çekilir. Eğer sistemin bir yerinde arıza, hata, işleyişi bozan bir durum oluşmuşsa süreç sonraki adıma geçemez. Tamamlanmamış veya fazladan kısım oluşturulmaması konusunda dikkatli olunmalı , önce hatalar, sistem aksaklıkları giderilip ardından sonraki sürece, önündeki sürece devam edilir. Kanban tahtasında üç durak /istasyon bulunmaktadır. Bu istasyonlar; İstenen, Devam Eden ve Tamamlanandır. Bu istasyonlar, bu akış sayesinde darboğazların görünür olması sağlanır. Verimliliğin artmasına neden olur. Kanbanı uygulamak kolay, ucuzdur. Kullanıldığı alan, yapılan işin ismi, işin tanımları, numarası, iş sayısı / kanbanın konulduğu şifrenin numarası veya ismi, bir önceki iş istasyonun adı, teslim edileceği iş istasyonun ismi, operasyonun ismi, teslimat süresi şeklinde tahtalar tasarlanır. Ancak her ekibin kendine özgü kanban tahtası olmalıdır [23]. Çünkü o işi, işin akışı takımlar kendileri tasarladıklarında hem sahiplik hissiyatı oluşur hem d sistemi yürütecek kişiler tarafından oluşmasından kaynaklı dar boğazlar, iş akışını bozan kısımlar da daha kolay aksiyon alınır.

1.4.1 Kanban'ın Değerleri

Kanban da scrumın değerlerine benzer değerleri vardır, ancak farklı olunan değerlerde vardır Kanban'ın değerleri şeffaflık, anlaşma, denge, saygı, anlayış, liderlik, işbirliği, müşteri odaklılık, akıştır.

Şekil 2.13. : Kanban Değerleri



Kaynak : https://kanban.university/

1.4.2 Kanban Rolleri

Scrum'da belirli roller vardır. Kanban'da yoktur. Ancak olmaması olmayacağı anlamına gelmez. İstenirse Kanban'da da, Scruma benzer roller kullanılabilir. Rolleri tanımlarken hassas olmalısınız, yeni tanımlanan rolün gerçekten ekibe değer sağlayacağından emin olmalısınız. Halihazırdaki aşamalara, rollere ve sorumluluklara saygı gösterilmesi gerekir. İlk başlarken hiç bir organizasyonel değişiklik olmamalıdır. Kanban, halihazırdaki aşamalara, rollerin ve sorumlulukların bir değeri olacağını ve bunu korumanın değerli olduğunu benimsemektedir. [24] Genellikle Kanban uygulayan şirketlerde roller; Kanban Master, Product Owner ve develeoperslardan oluşur. Kanban Master'ın görevi aynı Scrum Master da olduğu gibi Kanban uygulanması konusunda takıma destek verir. Kanban uzmanı, kanban bekçisi diyebiliriz. Product

Owner, ürünün sahibi, diğer takımlarla, müşterilerle iletişimi kuran, bağımlılığı yöneten kişidir. Developers ise iterasyonun ilerlemesinden sorumlu kendi alanında uzman, yetkin kişilerdir.

1.4.3 Kanban Prensipleri

Değişim yönetim prensipleri ve servis/hizmet prensipleri şeklinde iki grupta Kanban Prensiplerini sayabiliriz. Değişim yönetim prensipleri: Şuan ne yapıyorsan onunla başla, hayatını bir anda değiştirme mevcut senin için nasılsa öyle devam et der. Sürekli gelişimi destekle ekibin, ürünün gelişimi ve sistemin iyileşmesi için gelişimin önemli olduğunu vurgular. Üçüncü değer ise her kademede liderliği desteklemektir, herkes kendinin lideridir liderlik her alanla ve her kademede desteklenmelidir. Servis /Hizmet prensiplerindeki değerler ise müşterinin ihtiyaçlarını önceliklendir müşterini anla, neyi ne zaman istiyor, acili yeti ne, kişiyi değil akışı yönet. Müşteri ve paydaşlarına göre stratejilerini ve çalışma kurallarını geliştir.

1.4.4 Kanban Amaçları

Sürdürebilirlik; sabit tempoda ve sürekli gelişime odaklanmak. Hizmet odakları olmak, müşteri memnuniyetini her zaman öncelikli tut. Dayanıklılık, rekabetçi ve organizasyonel çevikliği sürdürmeyi amaçlar.

1.4.5 Kanban Pratikleri

İş akışını görselleştir. Akışı görselleştirmek hem şeffaflığı sağlar hem de sistemin akışında iyileşim noktaları varsa bunların görünür olmasına olanak sağlar.

İş sayısını limitlemek, işleri bitirmeye odaklanmayı sağlamakta, Push sistemleri pull sisteme çevirmede, iş akışında kapasite boşluğunu göstermede, müşterinin bekleme süresini kısalttmada, kaliteyi artırmasında yol gösterebilir.

Akışı yönet. Akışı yönetmekle işi zamanında bitirilemezse etkisi ne olur sorusunun cevabı bulunur. İşin tipine göre kural seti oluşturulabilir. İş tipine göre farklı renklendirmeler kullanılabilir. Önceliklendirme konusunda fikir verir. Kanban tahtasında işlerin sürtünmesiz ve blokesiz bir şekilde sağ tarafa doğru akması ve müşteri isteklerinin hızlı bir şekilde (asgari iş teslim sürelerinde) bitmesi en ideal durumdur. Acaba Kanban sistemimiz ne kadar verimli işliyor? Verimlilikten önce israf noktaları üzerinde konuşmamız gerekir. Aynı anda birden fazla işe

başlamak verimsizlik getirecektir çünkü Kanban tahtasında günler geçtikçe dokunulmamış işler olacaktır.

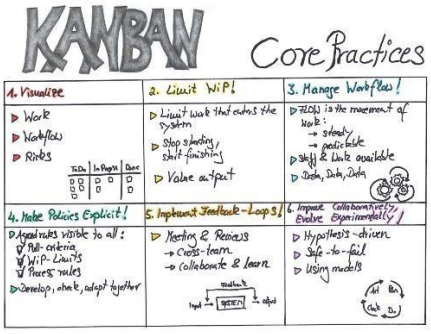
Süreçleri belirgin hale getir. Süreç kuralları çıkarılmış olur. Limitler daha net belirlenir, karar verme süreçlerini kolaylaştırır, herkes için net ve görünür olur. Süreç içerisinde geliştirilebilir ve güncellenebilir yerleri görünür kılar.

Geri bildirim döngüleri oluştur. Dailylerde boardu sağdan sola bakışla herkesin aynı noktadan bakmasını sağlar. İş çekme ve görev dağılımlarında takım birbirine geri bildirim verir, ortada engeller, riskler varsa bunlar konuşulur. Sürekli iyileşme, metrikler, süreçler ve limitler ile geri bildirimler oluşturarak iyileşme ihtimali artar.

İşbirliği içinde gelişimi destekle. Her nokta da iş birliği gelişmeli ki süreçler, kişiler gelişsin. Takımı geliştirmezsen gelişimi desteklemezsen sistem de de iyileştirme noktalarını yapabilmek içselleşmeyebilir.

Kanban, takımdaki üyelerin her biri içinde lider olma ruhunu ve karar vermeyi desteklemektedir. Ekibe dahil olan en yeni ve en düşük kıdemde olan alt ekip üyesinin fikri varsa, değerlendirilme li, uygunsa kabul edilmeli ve önemsenmelidir. Organizasyondaki herkesin ideal performansa ulaşabilmesi için her bir bireyin sürekli iyileştirme (Kaizen) fikrine teşvik etmesi gerekmektedir.[24].

Şekil 2.14. : Kanban Pratikleri



Kaynak : https://kanban.university/

1.4.6 Kanban Etkinlikleri (Kanban Ritüelleri)

Scrum'da Döngülerin Süresi sabittir. Kanban'da ise süreler sınırlı döngülerle kısıtlanmamıştır. Planlama, süreç geliştirme ve teslimi ne zaman yapmanız gerektiğini kanban takımı olarak kendiniz seçebilirsiniz. Kanban'da aktiviteleri tekrar eden bir şekilde yapmayı da seçebilirsin iz, ama zorunlulukları yoktur. Örnek olarak her Salı ürünü ya da hizmeti teslim edebilirsiniz ya da bir ürün parçacığı tamamlandığında direk teslim edebilirsiniz, müşteriye sunabilirsiniz.

1.4.6.1 Daily

Board üzerinden doğru ilerlemeyi gözeden geçirmek, planlana alınmış işlerin gereken şekilde düzenlemesinin yapıldığı backloga adaptasyonun sağlandığı, günlük 15 dakikayı geçmeyen toplantıdır. Karmaşıklığın olmaması adına her gün aynı yer ve zamanda olması önemlidir. Sistemde darboğazlar var ise bunlar konuşulur. Board kanban da sağdan sola doğru okunur.

1.4.6.2 Replenishment

Refinement etkinliği önceliklendirme, detaylandırma ve PBI büyüklük tahmini (yüksek seviye) içerir. Product Owner ve Development Team tarafından sürekli ve eş güdüm içerisinde yapılır. Development takımını kapasitesinin Maximum. %10 kısmını bu çalışma için ayırmalı ihtiyaç durumunda diğer paydaşlar da bu toplantıya katılabilir. Refinement sonucu Definition of Ready tanımına uyan maddeler sprint planlama toplantıları aracılığıyla sprinte dahil edilir. DoR demek; bir işin sprinte alınabilir duruma gelmesi yani çalışılmaya hazır olması demektir. İş tipleri öğeleri, Takım tarafından yapılan işlerin çeşidini belirtir. Örneğin analiz, kodlama, test, 3 farklı iş tipi öğesini ifade eder. İş tipleri öğelerinin ekibin kabiliyetine göre seçilmesi son derece önemlidir. İş akış durumları, değer akış ırmağındaki istasyonlar gibidir. En basit haliyle, Yapılacaklar, Yapılıyor, Yapıldı şeklinde iş akış durumları Kanban boardda ortaya konulabilir.

Gelen iş kaynaklarının farklılığı, takıma kaç farklı kaynaktan, hangi aralıklarla işler akmaktadır. Bunun belirlemesi son derece önemlidir. Çünkü takımın verdiği servisin hızı ve kalitesi bu sorunun cevabıyla doğrudan orantılıdır. WIP limitleri, iş kalitesini artırmak için önemli bir kuraldır. Aynı zamanda Little yasası ile tahminleme yapılabilmesi için önemlidir. Dolum zaman ve politikaları

Sırada bekleyen işler, boardda ne kadar iş var, WIP Limit kaç olmalı, hizmet grupları, iş tipleri hangi hizmet grubuna girmektedir. Acil işler var mı, müşteri öncelikli neleri bekliyor konuları

konuşabilir. Verimli olması önemlidir, verimsizlik oluşuyorsa bu verimsizlikler, israflar belirlenip iyileştirme yapılmalıdır.

1.4.6.3 Retrospective, İteresyon Sunumları

Kanban bu etkinlikleri scrumdaki gibi bir süreye ve zorunluluğa bağlamaz ancak yapılmasında demez takım isterse yapabilir. Çünkü değerlerinden birisi geribildirim döngüsü kurmaktır. Hem retrospective toplantıları hem de iterasyon sunumları geri bildirim döngüsü açısından önemlidir.

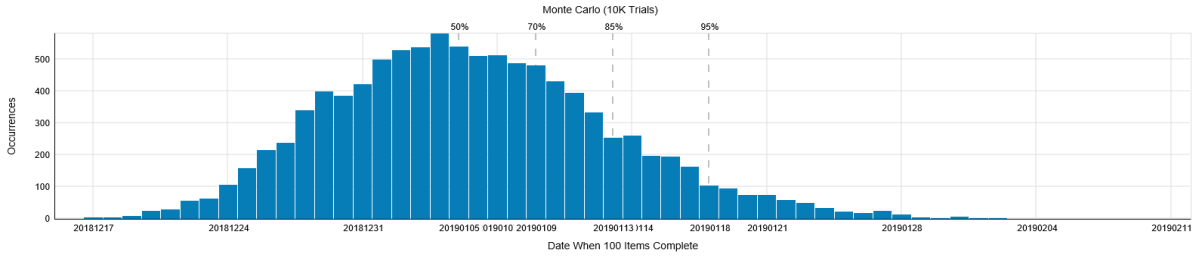
1.4.7 Kanban Metrikleri

Monto Carlo Similasyonu, WIP(work-in Progress), CFD (Cumulative Flow Diagram), Lead time, Cycle time ve Throughtput (çıktı) metrikleri vardır.

1.4.7.1 Monte Carlo Simülasyonu

Riski hesaba katmanıza ve veriye dayalı kararlar vermenize yardımcı olan matematiksel bir tekniktir. Monte Carlo; olayın belirli olasılıklarla gelecekte nasıl çıktılar elde edileceğinin önceden belirlenmesine, tahminlemesine yönelik kullanılan bir yöntemdir. Belirlenen olayın parametreleri/girdileri sisteme veri olarak giriş yapılır ve model kurulur. Tesadüfü zamanlarda veriler girilir ve/veya sonuca yönelik belirli ihtimaller ortaya çıkar. Böylece minimum maliyet ile sistemi üretime almadan bize getireceği sonuçları görmüş oluruz. Değişkenler/parametreler değişik ihtimal dağılımları ile gösterilir. İhtimal dağılımını kullanarak risk analizindeki belirsizliğ i tanımlarken sık ve yaygın kullanılan başarılı yöntemdir.

Şekil 2.15. : Monte- Carlo Simülasyonu



Kaynak : https://kanbanize.com/kanban-resources/kanban-analytics/monte-carlo-simulation

1.4.7.2 WIP(Work –in Progress)

Çok fazla parçalı tamamlanmış işlerin olması israftır ve pahalıya mal olmaktadır. Lead time'ın uzamasına sebep olduğu gibi organizasyonun müşterilerine ve fırsatlara hızlı yanıt verebilmesine engel olmaktadır. Kanban pratiklerinden birisi de limitledir. Work In Progress metriği de buradan geliyor. Takımların aynı anda birçok işle ilgilenerek hepsinde azar azar ilerleme katetmelerinin yerine limitleyerek daha odaklı ilerlemesi hedeflenmektedir. Bu sebeple her bir değer akışını ayrı ayrı bir limiti olmalı, takımlar deneyimleyerek değer akışında yer alan istasyonlar için ideal WIP limitini belirleyebilir. WIP iş sayısını gözlemlemek, limitlemek ve optimize Kanban metodunun temel başarısıdır. Servisin lead time süresini azaltır, kaliteyi ve iş teslim oranlarını artırır. Tam tersine efektif olmayan yönetim anlayışı insanların ve kaynakların maksimum kullanımına odaklanır ve insanların dolu olduğundan emin olmak ister. Sonuç olarak insanlar bunalır, kendilerine tanımları net yapılmış işlerin verilmesini isterler ve büyük resimdeki organizasyona ve müşteriye olan katkısını gözden kaçırırlar. Her bir adıma bir limit koyulanabilir.

Şekil 2.16. :WIP

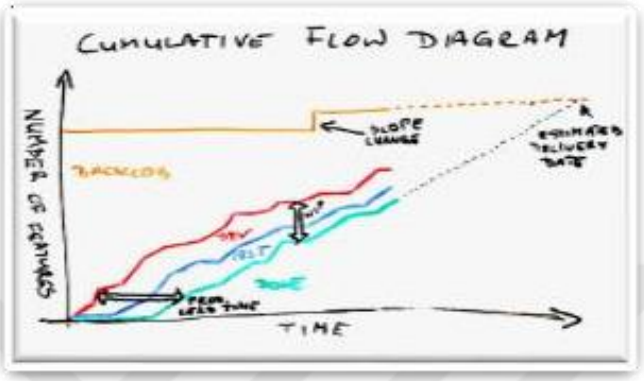


Kaynak : Teze konu olan şirketin xxx isimli takımı

1.4.7.3 CFD (Cumulative Flow Diagram)

Biri kapsam değişikliğidir. Kümülatif Akış Şemasına dayanarak, yalnızca ne kadar işinizin sürdüğünü tam olarak öğrenemezsiniz; bu bir çeşit yaklaşımdır. Oldukça iyi ama sadece tahmin. Yine de WIP'in zaman içinde nasıl değiştiğinin çok iyi bir göstergesidir. Nihai teslim süresi göstergesi olarak kullanılmamalı, ancak ne kadar ön süremiz olduğunu ve zaman içinde nasıl değiştiğini oldukça iyi gösterir. Son olarak, teslimat süresini kabaca tahmin etmek için yapılan eğrinin eğimini tahmin edebiliriz. Elbette kapsam değişirse teslimat süresi de değişecektir, dolayısıyla kapsam çizgisi (turuncu olan) da yaklaşıktır.

Şekil 2.17. :CFD



Kaynak : Teze konu olan şirketin xxx isimli takımı

1.4.7.4 Lead Time,Cycle Time

Talep sahibinin ihtiyacı ilettiği andan itibaren lead time'ın sayacı başlamakta, talebin tamamlanması ile birlikte de sayaç durmakta ve lead time sonlanmaktadır. Talep sahibi talebi ilettikten sonra takım hemen ele alamasa da sayaç işlemektedir.

Cycle Time ise, Geliştirme Takımı'nın bu geliştirme fikri üzerinde çalışmaya başlaması ve ürünü müşteriye teslim etmesine kadar ki geçen süredir. Bu süre içerisinde çeşitli sebeplerden dolayı takım bloke olduğu ilerleyemediği talepler olacaktır. Tavsiyem bu tip blokeli işlerde de cycle time sayacının işlemesidir. Böylece cycle time ile ilgili iyileştirme noktalarını da farkındalığı artmış olacaktır.

Lead time ve Cycle time ölçümlerinde dikkat edilecek noktalardan birisi de iş tipleri göre ayrımlar yapılmasıdır. Bakım, Problem, Talep gibi iş tiplerini tek bir lead ve cycle time olarak ölçümlemek ve hepsinin ortalamasına bakmak yanlış yönlendirecektir. Bu sebeple ortalama lead ve cycle time belirlenirlen iş tipine göre belirlenmesi israf ve iyileştirme noktalarının belirlenmesi açısından daha sağlıklı olacaktır. Ortalama lead/cycle time takip edilirken bir periyodun belirlenmedir. Her bir iterasyon sonunda ortalama lead/cycle time'daki değişiklikler takip edilebilir. Takip edilmesi

için çizgi veya bar grafiği kullanılabilir. Herbir PBI'ın lead/cycle time incelenirken aykırı olan taleplere odaklanılabilir. Taleplerin çoğunluğunu toplandığı bir bölge olacaktır. Ama aykırı dediğim bazı talepler ise bu bölgenin üzerinde bulunacaktır. Öncelikli amaç aykırı olan taleplerin süresinin uzun olmasına sebep olan etkenlerin belirlemesidir.

Lead Time = Teslim Tarihi – Fikrin / İhtiyacın Oluştuğu Tarih

Cycle Time = Teslim Tarihi –Takımı'nın bu işte çalışmaya başladığı tarih.

Şekil 2.18. : Lead Time-Cycle Time

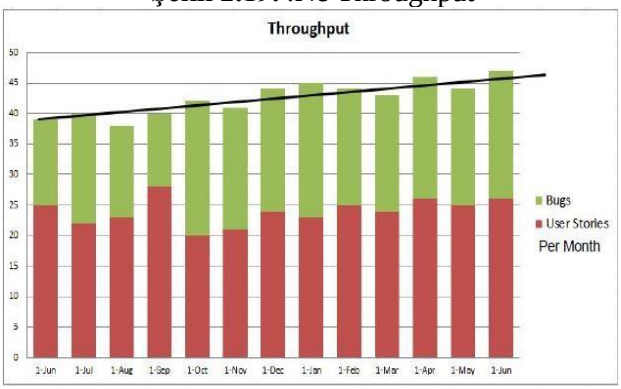


Kaynak : Teze konu olan şirketin xxx isimli takımı

1.4.7.5 Throughput (Çıktı)

İş sayısının ortalama yazılım teslim süresine oranı birim zamanda biten iş sayısını verir. Buda Throughputtur. Değer akışından geçen tamamlanmış PBI sayısıdır. Throughput'un da Türkçe karşılığı çıktıdır. Bu metriği de etkin bir şekilde kullanılabilmesi için iş tipine göre bakılması önemlidir. Belli bir periyotlarda da iş tipine göre tamamlanmış PBI sayısı takip edilebilir.

Şekil 2.19. :Ne Throughput



Kaynak : https://kanbanize.com/

1.5 ORGANİZASYONEL ÇEVİKLİK (ORGANIZATIONAL AGILITY)

Bir organizasyonun, gelmekte olan veya var olan bir tehlike yada imkana uygun zaman da tedbir alabilmesine, cevap verebilmesine, adapte olabilmesidir. Bugün var olan değişen sosyoekonomik şartlar, büyük rekabetin olması, artan maliyetler, belirsizlikler ve güvenlik adına alınan tedbirleri, değişime uğrayan müşterinin taleplerini ve/veya müşterinin isteklerini organizasyonlar ın faaliyetlerini sürdürmelerini güçleştirmekte, pazardaki yeri koruyabilme ve/veya bu yeri geleceğe taşıyabilmedeki muğlaklıkları doğurmaktadır. Organizasyonların sürekliliğini devam ettirebilmesi için, örgütsel olarak değişime çabuk uyum sağlayabilmek, yeni kauslar olmadan atik davranmak, hızlı ve çabuk yanıt vermek gerekir. Buna ayak uydurabilen organizasyonlar, olası riskler karşısında en az zararla süreci geçirir, avantajlara çabuk şekilde cevap vererek piyasadaki çekişmede belirgin hale gelir dolayısıyla pazar payını artırır. Çabuk yanıt verebilmek yönetim biliminde "Organizasyonel Çeviklik" olarak tanımlanır. Organizasyonel, değişiklikleri görünür kılmak ve bunların yaşatacağı negatif etkilerin ve/veya imkanların varlığını öğrenmek, organizasyonel çevikliktir. Değişimlere anda cevap bulmak yanıt verebilmek, doğru çözümler ve bulunan çözümleri hayata geçirebilme yetileri ise çevikliği (agility) gösterir. Organizasyonel Çeviklik için yönetsel ve pazarlama stratejileri, teknolojileri, çalışanları, süreçlerini, araştırma geliştirmeye daha farkında olabilmeyi doğurmaktadır. Çeviklik kavramında hız, belirsizlik, risk ve/veya esneklik bulunmaktadır. Yine de sadece hızlı ve esnek olmak, çeviklik için yeterli olmamaktadır. Organizasyonel Çeviklik için, organizasyonun stratejileri, vizyonu ve değerlere önem verilmeli, hızlı ve/veya atik davranmakla yapılacak kısa ve/veya uzun dönemde stratejik

plan ve hedefler organizasyonun hedefleri faydalı katkı sağlayacak fikirler alınabilmelidir. Ortaya çıkan fikirler zamanı da dikkate alarak ve organizasyonun yalnız hızlıca gelişmesini değil, bu geliştirmelerin doğru uzantı da olmasını, ihtiyaç halinde de başa dönüp yeniden rotasını değiştirmesini ve bu değişime ayak uydurması da değerlendirilmelidir. Özet olarak, yeni ürünü pazara sürmek, yeni ürünü geliştirmek ve bununla ilgili girişimlerde bulunurken, mevcuttaki işleri de durdurmamalıdır. Bunları yaparken güncel teknoloji ve sistemlerini kullanmalı, takımlar ve alanında uzman kişilerden yararlanmalı, süreçlerini çevikliğe uygun hale getirmeli, organizasyon biçimlerini uyumlandırmalı, iş birliğine dayalı çalışma sistemini yaratmalı ve bütünleştiric i olabilmektedir. 1980 senesinin sonundan 1990'lı yılların ilk çeyreğindeki dönemler de, dünyadaki gelişen sosyo-ekonomik ve politik gelişmeler ve/veya yeni doğan sektörleri etkilerini anlamak için çok fazla efor tüketilmiştir. Geçen süre de bilim adamları, bu değişimlerin geleneksel organizasyonlardan daha çabuk ilerlemesi sebebiyle, organizasyonların değişime direnç göstermesi ya da uyum sağlayamama gibi negatif etkileri olan problem yaşadıkları gözlemlenmiştir. Organizasyonlar için meydana gelen bu imkanların artı yanlarını göz ardı etmiş ve değişikliğe ayak uydurmamıştır. Bu sebeplerle uzun periyotlar da pazardan çekilmeye ve kaybetmeye ya da piyasa koşullarına ayak uyduramama durumunda kalmışlardır. Çabuk ve tahmin edilemeyen, hesaba katılamayan durumlar halinde artan değişimler, organizasyonların başlıca sorunları olmuştur. Taraf olmayan organizasyonlarda meydana gelen teknolojik değişim ve gelişimin ilerleyen bütün zaman dilimlerinde fark edilebilir olması da organizasyonları çevik olmayı mecburiyete itmiştir. Şirketlerin nasıl çalıştıkları konusunda bilgilidir; Resmi ve gayri resmi kanallarla işini nasıl yaptıracağını bilir; Temel politika, uygulama ve prosedürlerin ortaya çıkış noktalarını ve arkalarında yatan sebepleri anlar; Şirketlerin kültürlerini anlar farklılaşan durumların firmalar için çeviklik oldukça başarı sebebi olarak gözlemlenmiştir. Kendi çıkarları doğrultusunda manevralar yapmak için çok fazla zaman harcayabilir; Önemsiz konular üzerinde çok fazla zaman ve enerji harcayabilir. Belirsizliğin ve değişim hızının üstesinden gelinebilmes i için kurum ile piyasa arasında arasın da sürekli bir ilişki yaratılması gerekir. İş çevikliğinin temelinde, değişen ve gelişen ortamlarda etkili bir biçimde performans gösterebilmek için kurum stratejisini, organizasyonunu ve operasyonların, ürünlerin sürekli olarak gözden geçirip adapte edebilme, adapte olabilme yetkinliği ve esnekliğidir. İş çevikliği, organizasyonların değişen, gelişen piyasa koşullarına hızla uyum sağlamasına, yeni iş fırsatlarından faydalanmasına, anı değerlendirmeye, yeni ve alternatif dağıtım kanallarını benimsemesine veya tedarik etmesine,

geliştirmesine , maliyetleri azaltmasına veya süreçteki gelir akışlarını artırmasına olanak tanımaktadır.

1.5.1 Bir Organizasyonu Çevikleştirmek İçin Yapılması Gerekenler

Şirketin çevikliği kendi istikrarı dışında olan parametrelerle etkileşim içinde olmayı sağlamaktadır. Çeviklik, uzayan ve takip edilemeyen değişim ve değişkenlerin baskına olan etki alanında gelişme ve/ veya büyüme becerisidir [21]. Kidd' e göre (1994) çevik üretim taktiklerinin odak noktaları şunlardır:

- Çok yönlü müşterinin isteklerine ,atik bir şekilde geri dönüt sağlayabilmek,
- Projenin oluşturulması ve prosesif sürecinin azaltılması ,
- Ucuz olmayan girdilerin daha verimli kullanılması,
- Ürün yelpazesi çeşitliliği ve kişiye özel ürünler vb. [12].

Organizasyonun, insanların ve teknolojinin kendilerine aynı doğrultuda ve ilişkili olarak geliştirilmesini ve çevikleştirilmesini günümüzde zorunlu kılınmıştır. Böylelikle çeviklik organizasyonun kültürü haline gelmiş olacaktır. Çeviklik modelinin dört temel başlığı bulunmaktadır. Alıcıya olabildiğince çok değerin sunumu, pazarda çekişme düzeyini çoğaltmak için değişik organizasyonlarla sağlanacak olan iş birlikleri, belirsizliğin ve değişim etkili bulunduğu alanlarda üretimi yapmak ürün/ hizmet sunabilmek ve/veya gelişebilmek için uygun faaliyetlerin olması ve grup içindeki kişilerin bilgi, beceri ve etki alanlarının, karar alma yetkinliklerinin arttırılması. Şirketi çevikleştirmekte dokuz temel alanın çevikleştirilmes i gerekir.[7]. İlgili alanlar aşağıda sıralanmıştır. Bu alanlar ;

• Ürünün/hizmetin tasarlanması,

• Sürecin tasarlanması,

• Üretimin programlanması, planlanması ve kontrol edilmesi,

• Tesisin tasarımı ve konumlandırılması,

• Verilecek olan hizmet veya ürünün lojistik süreci,

• Bilgi iletişim merkez ve sistemleri,

• Tedarik zincirleri,

• Beşeri etkileri,

• İş uygulamaları ve süreci .

Yukarıdaki alanlarda çevikliğin sağlanması durumunda organizasyonun kültürüne yansıdığından söz etmek mümkün olacaktır.

Farkındalık: Organizasyonlar değişen piyasa koşullarını anlarlar veya bunu zamanla kavrarlar. Organizasyondaki belirsizlikleri öngörebilmesi, canlı, etkin iletişimi kurabilmesi ve feedback i kurumsallık seviyesine çıkarabilmek için ekiplere ar-ge için gerekli bilgi ve zamanı sağlamasıdır. *Esneklik:* Organizasyonlar kapsamlı piyasaların koşullarına yanıt verebilirler. İnsan kaynakları ve dış girdilerde yararlanmak, organizasyon esnekliği durumunu şekillendirmek, hızlıca uyum sağlamak.

Üretim: Organizasyonlar belirgin piyasa değişikliklerinde efektif ve verimli bir şekilde işletilir ler. *Teknoloji,* şirket yapılarındaki en karışık ve üretim sürecinde ; çevikliğe en ihtiyacı olan belirsiz piyasa değişkenlerindendir. Süreçlerin tanımı, süreçlerin sıralaması, iyi desteklenen bir iş birliği ve çabuk öğrenen çalışanlar çevik organizasyonlar ın oluşturmasına yardımcı olur.

Bilinen veya bilinmeyen değişime doğru ve zamanında cevap verebilmektir. Değişiklikler i avantaja çevirerek, bu değişikliklerden kendi artıları doğrultusunda faydalanmaktır. Organizasyonun çevikliği için dört temel özellik bulundurması önerilir: *Cevap Verebilme*: Farklılıkları belirleme, çabuk cevap verebilme ve kullanma yeteneği.

Yetkin olabilme: Şirketin hedefleri ve/veya amaçları doğrultusunda etkinliği ve verimliliği etkiler. *Esnek olabilme:* Çeşitli hizmetleri ve/veya ürünleri farklı şekillerde, her ürüne, gelişen teknoloji ve insan kaynağına adapte olabilme, adaptasyon uyumluluğundaki işleme yeteneğidir. *Hızlı olabilme:* Olabildiğince çabuk bir biçimde işlevlerin ve/veya görevlerin uygulanabilme yeteneği olarak tanımlanır.

Organizasyonel Çeviklikte organizasyonları göz ardı etmemesi gereken unsurları vardır.: Stratejik vizyonu, yeterli teknolojik becerileri, uygun teknolojileri (yazılımsal ve donanımsal), ürün ve

hizmet kalitesini, maliyeti, verimliliğini, yeni ürünü müşteriye sunmayı, değişimi yönetebilmeyi, yeterli ve güçlendirilmiş bireyleri, takım etkinliklerini, hem takımda hem şirket bünyesinde iş birliği, düzen, üretim hacmi esnekliği, ürünün model ve şekil esnekliği, birey esnekliği, pazara yeni ürün sunma hızı, ürün ve atikliği ve dakikliği, hızı takım içerisinde güçlü bir iletişim kurulması önerilebilir. Merkezi bir otoriteden karar almak yerine, kararların fonksiyonel noktalarda alınmasıdır. Kısa vadeli basit planlarını yapılması önerilebilir. Kişiselleştirilmiş ürünü üretebilme kapasitesi sağlanması önerilebilir. Ürün ve hizmet kalitesi artırılması önerilebilir . Ürünün teslim süresi kısaltılabilir. Bu hizmetlerin sürekliliği ve organizyonel çevikliğin kurumsallaşarak bir yönetim kültürü haline dönüşmesi önerilebilir. Kurumsal çeviklik, oldukça önemli bir kavramsal değerdir fakat ölçülebilir değildir. Organizasyonların piyasadaki çekişme yeteneğini sürdürebilmeleri, çekişme gücü ve değişikliklere karşı hazır olabilmesi için, organizasyonlar içerisinde çevikliğin düşük seviyede olduğu alanları tespit etmek ve bu alanlarda gerekli iyileştirmelerin planlanması oldukça önemlidir.

Yazılım yönetim metotları teorik olarak alt yapısını 1911'de Fidelik ve ismin Taylor tarafından yayınlanmıştır. Principal of Scientific Management çalışması kaynaktır Taylor bu çalışmada seri üretimin yapılması için işlerini tamamlayanların veya yapanların farklı özelliklere sahip olması gerektiğini savunmaktadır. Böylece seri üretim için iş tanımları mikro düzeydeki ayrıntısına kadar yöneticiler vasıtasıyla belirlenecek ve iş görenler de aynı işi sürekli yaparak iş üzerinde uzman kazanacak üretim bu şekilde en iyi şekilde yapılacaktır. Taylor'ın çalışması sanayi devrimi dönemi için gerçekten yeni bir akım yaratmıştır. Bu sayede üretim bandı kavramını geliştirmiş ve seri üretim hız kazanmıştır. Principal of Scientific Management'ın yaklaşımı kendi döneminde şartlar içerisinde oldukça anlamlıdır aynı ürün işler tekrar edilerek seri şekilde üretebilmektedir gereksinimlerin değişim hızlı sıklığı çok azdır ve adaptasyon için yeterli süre vardır rekabet sordukları bugüne kıyasla çok daha hafiftir sistemi işlerin sürekli tekrarı uzmanlaşma ve en ince aynısına kadar tarif edilebilirlik yolunda verimlilik ve faydalanma üzerine inşa edilmiştir tandır sisteminde bir işçinin ya da bir makinanın boşta kalması Üretimin azalmasına neden olacak bir birim ürün maliyetini artıracağı için faydalanma oranını maksimize edilmesi gerektiğini söylemektedir benzer şekilde yazılım sanayisinde metodolojik için yaklaşım 1970 yılında doktor Winston W. Royce tarafından yapılmıştır. Ross makalesinde yazım sayfalarına bir şelaleye benzeterek bu adımların analiz tasarım kodlama test gibi izlenmesi yoluyla Şelale metodunun başarıyla çalışabileceğini söylemiştir bununla birlikte tarif ettiği makalesinde bu sayfalara geri

dönme ihtiyacı olabileceğini örneğin tasarımdan analize kodlamadan tasarıma veya testten diğer sayfaları dönüş ihtiyacı ulaşabileceğini öngörmüştür böyle bir ters yönde akış durumunun zorluklarını ve metodun bu yönden risk içerdiğini de açıkça belirtmiştir nedense Ross'u makalesinden en az önem verilen kısımda bu nokta olmuştur. Tüm bu sorunlara bir çözüm hatta Şelale metodundan bir alternatif olarak ortaya çıkmıştır ortak değerlere baktığımızda ağırlığ ın insan faktörüne çalışan yazılıma işbirliğine ve Agile felsefenin temellerini oluşturan değişime karşılık vermeye değer verildiğini görülür. Sprint içerisinde tüm takımların maksimum fayda sağlayabilmesi için üzerinde çalışabileceğiniz bazı ipuçları vardır:

1. Öncelikle ürün listesindeki öncelikli işleri yani diğer üreten küçük parçalara bölmeye çalışın böylece Sprintin hemen ilk günlerinde analizi yapılmış kodlaması bitmiş ve teste başlanılmış olabilecektir bu durum takımın kendi içinde birlikte çalışma pratiğini olumlu yönde tetikleyecektir.

2. Otomasyon ve mühendislik pratiklerine yatırım yapmayı her Sprintin her işin bir parçası haline getirin.

3. Test-first gibi yaklaşımlar kullanın.

4. Kişilerin yetkinliklerini artırmaları teşvik edin teknolojiden anlayan iş gereksinimlerin den anlamayan geliştirici kodlar anlamayan mühendisinin artık işi zor ve bu gerçekçi değildir. Güçlü takımlar yetkin kişilerin Bir arada çalışabilmesi ile mümkündür.

5. Pear-Programming eşli çalışma gibi paylaşımı tetikleyen teknikler kullanın.

6. Takım içinde 100 yüze iletişimi artırın işlerin sık sık birlikte konuşup çözümüne birlikte tartışmasını sağlayın ve unutmayın yazılım problemleri lineer ya da Deterministik (belirlenimc i) değil Stokastik (rastgele) yani değişken ve rastlantısal olan bir durumdur: Boştaki çalışana değil boştaki işe yoğunlaşan kişilerin ve sistemin dolu görünmesi İllüzyon da kanmayın. Önemli olan kişilerin takımların sistemin ne kadar yoğun ve dolu olduğu değil ne kadar üretkendir. Kişileri yönetmek ve onların nefes alamayacak kadar dolu olmasını sağlamaya çalışmak yerine, kişilere oyun olanı yaratın ve onları takımlarıyla birlikte güçlendirin kompleks sistemlerde hızlı ilerlemenin yolu yavaşlamakta ve odaklanmaktan geçer. Organizasyonel değişim de önemli olan Çevik'li artırmak çevik bir yöntem uygulamak değil çevikliğin devamlılığı için organizasyonlar ın alt yapılarında stratejilerinde performans yönetiminde hiyerarşik yapılandırılmasında

bütçelenmesinde planlamasında uygun çalışma alanlarında kullanılan araçlarda değişiklikler in yapılması gerekir. Bu tür değişiklikleri organizasyondaki birçok kişinin çıkarlarına etkilediğinden zor değişiklikler olarak adlandırabiliriz. Alt yapısal değişiklikler davranışlar da desteklemeli ve davranışlar da kültür değişimin yolunu açabilmeli organizasyonun çevikliğini sürdürebilmesi için alt yapısal değişikliğin yapılabileceğini denenmelidir. Organizasyonel dönüşüm birçok alternatifin olduğu bir süreçtir ve bu alternatiflerin bazı zor kararları bol deneme yanılma ile olmaktadır . Çeviklik dönüşümleri nihayetinde şirket kültürüne dokunduğundan uzun zaman süren değişimlerdir. Liderlik ve değişim uzmanı John P. Koter değişim inisiyatiflerin başarı oranı %30 olarak belirtmiştir bununla birlikte çoğu zaman yeni forma uyum sağlayamayan çalışanların bir kısmı da bu süreçte şirketin ayrılabilir. Bu durumu doğal seleksiyon olarak değerlendirebilir. Aciliyet duygusu ve kavramını yaratmak kurumsal şirketlerde çalışan birçok insan birçok konu ile ilgili değişmesi gereken bir şeyler olduğunu düşünür ve kimi zamanda dedikodu şeklinde bunu dile getirmeye çalışırlar. Ancak çokta değişimlerin mola aralarında konuşularak gerçekleşmesi çok zordur. Öncelikle organizasyon yoğun bir değişime ihtiyacın ne olduğunu organizasyonu değiştirebilecek kritik kişilere kabul edip başlaması gerekir. İlk adım bu farkındalığın çarpıcı şekilde oluşturulmasıdır. Bu noktada mutlaka ve mutlaka her bir bireyin bu değişimin bana ve yakın çevreme ne faydası olacaktır sorusuna cevap bulması gerekir. İkincisi değişime yön verecek lider birliğin oluşturulmasıdır. Kritik kişilerde farkındalık oluşmasıyla bu kişilerin birlikte hareket edecekleri aynı jargonu ve aynı iletişim stratejilerini benimseyecekleri bir liderler takım oluşturulur her ne kadar gerçekçi bir takım olsa da bir süre değişim ve dönüşüm adımlarının yönetimini yapacak olan ekip budur. Üçüncüsü değişim vizyon ve yol haritasının oluşturulmasıd ır. Liderler takımı değişimin organizasyonun nereye götüreceğini ile ilgili hedef noktasında hemfikir olabilmeli vizyonun doğru iletişimi ile organizasyonun geri kalanında farkındalık yaratmak ve insanları bir amaç uğruna bir araya toplamak kolaylaşır. Vizyonun belirlenmesinden sonra kısa ve uzun vadeli stratejik hedefler ortaya konulabilmelidir. Bu hedeflerden aksiyon planları oluşturabilmesi, planlar aksiyonları hayata geçirilmesi ve aksiyonları sonuçlarına göre değişe bilmelidir. Dördüncüsü vizyonun iletişimidir. Bu aşamada birçok araç kullanılabilir farkındalık seansları bildiriler birebir görüşmeler ve çeşitli kanallardan yayınlanır. Bu iletişim bir seferlik değil sürekli olmalı, vizyon iletişimi yapılırken eski yöntemlerle kullanılan jargonlara yer verilmesine dikkat edilmeli, bunun yerine yeni kültürü sembolize eden jargonlar kullanılmasına özen gösterilebilir. Örneğin; rapor kaynaklanma gibi insanların kullandığı itici hale gelen ifadeler

delege etmek gibi jargonlar tercih edilmeyebilir. Beşincisi değişime destek verenleri güçlendirmektir. Organizasyonda değişim karşında duran birçok kişi olacağı gibi değişimi destekleyenler de olacaktır. Bu kişileri güçlendirmek ve onların da aksiyona geçmesini kolaylaştırmak önündeki engelleri kaldırmakta önemli adımlardandır. Bunun için lider takımın mutlaka engelleri aşması, ikna olması, benimsemesi, içselleştirmesi ve bu kişilerle yakın çalışması yakın çalıştığının da görünür olması gerekmektedir. Altıncısı kısa vadeli kazanımlar oluşturmaktır. Vizyon ve strateji gibi konuların iletişimi yapıldıktan sonra araya fazla zaman girmeden aksiyona geçilmesi ve küçük kazanımların oluşması değişime olan inancı güçlendirir küçük kazanımlar değişim konusunda kararsızlar da ikna etme konusunda kolaylık yaratacaktır. Bunun dışında değişim içerisindeki insanları da büyük motivasyon sağlayacaktır. Kazanımların etkili bir şekilde organizasyondaki diğer insanları aktarmakta oldukça kritik öneme sahiptir. Yedincisi vizyon doğrultusunda pes etmeden ilerlemektir. Küçük kazanımlar için çalışılırken kazalar ya da başarısızlıklar yaşanabilir. Her başarısızlıktan öğrenilmiş olan dersler organizasyonu başarıya biraz daha yaklaştıracaktır. Dolayısıyla öğrenme böylesine kompleks bir sürecin en önemli yapıtaşlarından birisidir. Hatta değişimin iletişimi yapılırken bu başarısızlıkların yaşanabileceği de vurgulanabilir. Sekizincisi yeni bir kültürdür. Organizasyonda başarı ulaşma yöntemi değiştirilebilir. Bir deneme yanılma önemli bir iş yapış şekli haline gelmekte yeni değil yeni başarı ve başarısızlık tanımları organizasyon tarafından benimseniyor olabilir. Bu değişiklikler organizasyon tarafından büyük bir heyecanla yaratılmalı, ancak unutulan ya da yeterince vurgulanmayan önemli bir haftası da olabilir. Buda geri bildirim döngüleri kurmaktır. Sistemin çalışıp çalışmadığını nasıl iyileştirebilir olacağını hakkında kişileri organizasyona ve takımlara geri bildirim döngüleri olmalıdır.

1.6 VERİMLİLİK

Verimlilik ve etkinlik ölçülerdir. Herhangi bir organizasyonun müşterilerine ne kadar iyi değer sunabileceğidir. Etkililik doğru olanı yapmaya odaklanırken, verimlilik, bu değerin ne kadar hızlı olduğuna ve ne teslim edildiğine odaklanır. Daha hızlı teslimat maliyeti düşürür birçok ortamda üretim küçük bir indirim maliyet açısından karlılıkta önemli bir kazanç sağlayabilir. Ayrıca etkinliği artırabilir. Toyota'daki mühendisler, kârlılığın% 95'ini (1) çok istenen bir ürünü tasarlamak ve (2) Pazar penetrasyonunu artırmak için düşük maliyetli tasarım[24]. İsraf düşük

miktarda atık olması anlamına gelir, üretim sürecidir. Yani, verimlilikte bir artış doğrudan israfların azalmasıyla bağlantılıdır.

İsraflar, ürüne değer katmayan herhangi bir faaliyet olarak tanımlanır. Verimliliğe bu şekilde odaklanmak Toyota Üretim Sisteminin [25] merkezinde yer almaktadır.

Verimliliğe odaklanmak çok önemliyken, verimliliği çok düşük olan ekiplerde verimliliğe odaklanan faktörler de vardır. [26]. Bu, odaklanmak anlamına gelir. Verimlilik değeri 1 gün veya 6 yıl içinde sunmak arasındaki fark olabilir. Geliştirme ekiplerine verimlilikleri hakkında fikir vermek için Süreç Verimliliği bir ölçüt sunar ve Scrum çalışma biçiminde tanımlanır [27]. Scrum şu gözlemlerden türemiştir:

Yalın donanım ekipleri [28]. Süreç Verimliliği, Yalın Üretimde onlarca yıldır kullanılan standart bir metrikten türetilmiştir - katma değerli çalışma süresinin saat zamanına bölünmesidir. [29].

Uygulamada, Süreç Verimliliği% 25'i aşmaktadır. Süreç verimliliğini kullanımıyla geliştirilm iş süreçler vardır. Bunlar; Yalın yöntemler [29] ortalama Scrum takımı, Ürün İstek Listesini tamamlamak için Süreç Verimliliği, Anketlere göre ürün% 5-10 seviyesindedir.

ABD ve Avrupa'da Scrum sınıflarına katılanlar vardır. Bunlar;[30]. Süreç Verimliliğ inin Ölçülmesi, önemli ölçüde Scrum takımlarının performansını olduğu gibi iyileştirmek, takım performansındaki artışla doğrudan ilişkili [27]. Aynı zamanda bağımsız olma avantajına da sahiptir.

Ekipler, teknolojiler veya çalışma alanıdır. Bu belge, Sürecin nasıl ölçüleceğini tanımlamaktadır. Verimlilik, herhangi bir Agile ekibinin onu kullanmaya başlamasına izin verir. Verimlilikler i hakkında fikir edin. Agile yöntemlerle çalışan şirketinde çalışmalar [27], Süreç Verimliliğinin iyileştirildiğini gösterir. Bu göstergeler; bir hikâyenin olup olmadığını belirleyen kontrol listesi. Bir sprint'e "hazır". Yayınlanan bu etkiyle ilgili araştırmalar, Scrum Kılavuzunda [31] "Hazır" kavramı ve Hazırın Tanımı" adlı bir modelin yayınlanması, ScrumPlop.org [32] olabilir. Mevcut verilere dayanarak sorunun analizi ve Metrik Süreç Verimliliği tanımlanabilir. Süreç Verimliliğ ini ölçmek ve hesaplama ile başlanabilir. İlk Süreç Verimliliği ile ilgili ölçümler, ekiplerin üretim süreçlerini incelemelerine izin vermek, son düşüncelerle sonuçlandırılabilir.

1.6.1 Süreç Verimliliği

Metrik ve ekiplerin bunu nasıl ölçebileceğidir. Bir ekipte % 0 ile% 100 arasında bir ölçekte metrik mutlak karşılaştırmalar yapın. % 0 verimlilik asla üretime geçmeyen ürünleri temsil eder, % 100 tam kesintisiz odak anlamına gelir. Baştan sona müşteri için değer katmak. Süreç Verimliliğ i herhangi bir geliştirme için ilgi çekicidir. Performanslarını iyileştirmek isteyen ekipler, o sprinte ya da iterasyon da Yalın'da iyi bilinen bir metriği uyarlar. Üretimde ise bunu doğal bir şekilde yapar. Çevik takımlar, takımlara miktar konusunda iç görü sağlayarak değer kattıkları zaman ve miktarı boşa harcadıkları zamanı kullanırlar. Bir takımın hesaplamak için ölçmesi gereken unsurlar; Kullanıcı hikâyeleri için Süreç Verimliliği, Süreç Verimliliğini tanıtma hedefi, geliştirme ekipleri, hikâyeleri daha hızlı yerine getirmeye odaklanır ve geri bildirim döngüleri ile bu ekiplerin müşterilerinin ihtiyaçları paylaşılır. Böylece, Süreç Verimliliği, odaklanan ekiplere çok iyi sonuçlar ortaya çıkar. Kaynak yerine Aktarım Hızı Maksimizasyonu, Kullanım Maksimizasyonu. İstenildiği gibi yan etkiler, Süreç Verimliliği oğul vermeyi [32] ve geliştirmedeki atıkların detaylı analizidir.

1.6.1.1 Süreç Verimliliğinin Ölçülmesi

Ekiplerin israfı azaltmasına ve teslimat hızı, ölçüme odaklanmaları gerekir ve Süreç Verimliliğ in i optimize etmektir. İşlem Döngüsü Verimlilik, Yalın'dan gelen bir ölçüdür. Üretim ve göreceli miktarı açıklar. Değer katmakla değer katmamak için harcanan zaman ürün (veya hikaye). [29] 'dan bu Süreci alıyoruz.

Döngü Verimliliği = Katma Değer / Toplam Harcanan Zamanı

Bunu bir geliştirme ortamına uyarlayacak olursak: yazılım geliştirme ekipleri için bir tanım oluşturmak, birlikte çalışmak. Sezgisel olarak, iki şey olmalı. Süreç Verimliliğini hesaplamak için ekipler tarafından ölçülür.

1. Hikâyenin devam ettiği toplam süre, hikâye başarıyla yerleştirildiğinde üretime alındığında biter.

2. takımın yasaklandığı toplam süre hikâyeye değer katmaktan. İlk ölçü Döngü Süresidir. Devir süresi gerçekleştirmek için harcanan toplam süreyi temsil eder. Döngü Süresi saat cinsinden ölçülür ve yalnızca çalışma saatleri sayılır. Yani, Pazartesi günü saat 4'te başlayarak çalışın

öğleden sonra ve cuma günü saat 10'da biten bir döngü vardır 2 saatlik zaman (sabah 9'dan akşam 5'e kadar çalıştığı varsayılarak) Döngü Süresini hesaplamak için ekipler takip etmelidir. İş üzerinde çalışmaya ne zaman başladıklarını ve ne zaman geldiklerini, üretime ne zaman aldıklarını.

Döngü Süresi: Hikâyenin içinde bulunduğu toplam süredir, ilerleme, çalışma saatleriyle ölçülür. Şu şekilde hesaplanır:

Üretimde Zaman Hikâyesi - Hikâyenin Başlangıç Zamanı = İkinci ölçü Kesinti Süresidir.

Kesinti Süresi tüm zamanların toplamını temsil eder, takımın değer katmaktan kesintiye uğradığın ı Onların işidir. Kesinti Süresi ayrıca ölçülür. Saatler ve yalnızca çalışma saatleri sayılır. Devir süresi yerine kesinti süresi kullanma nedeni ve katma değeri ölçmeye çalışmak iki yönlüdür. İlk hepsinden, bu ölçümlerden biri yeterince iyi, bunlar sadece toplam süreyi oluşturur. İkinci bunun nedeni, ekiplerdeki insanların çalışırken kesinlikle ölçmeyi severler. Takım yaşadıklarını da ölçmekten hoşlandıkları deneyimdir. İşe değer katan faaliyetler katma değerli zaman olarak kabul edilir. Yapan faaliyetler işe değer katmayan işler israftır azaltılabilmelidir. Takımlar birçok kişiden oluştuğu için herhangi bir zamanda bazı kişilerin değer, diğerleri ise değildir. Olmayan insanlar değer katmak o zaman değer katamaz, yani israftır. Ancak ekleyebilecek bazı kişiler değer katma değer değildir, bu israf olarak kabul edilir. Bu nedenle, bir kesinti kısmi olabilir, değer katabilecek insanlar değer katmıyor. Uygun araçlar olmadan, ölçmek için gereken çaba miktarı uygulamada bu çok yüksektir (bir parçası olan takımlar için olarak rapor edilen ilk ölçümler problem veya potansiyel problem). Kısmi kesintilere yaklaşırken, kesintilerin uzunluğunu ölçmeyi önermek, bir takımın bir takımın kesintiye uğradığını tanımlamak, Bu yaklaşım izin verdiği için sistemi oynamanın daha az yoludur.

Kesinti Süresini hesaplamak için ekipler, değer katmaktan ne zaman kesintiye uğradıklarını takip etmelidir. İşteki söz konusu kesintinin ne zaman başladığını yazılmalı, ne zaman tekrar çalışmaya başlayabileceklerini not etmelidir. Örnekler şunları içerir, ancak bunlarla sınırlı değildir : Toplantılar, Manuel Test, Manuel Dağıtım, o işten başka bir şey üzerinde çalışmak, Ekibi yasaklayan Ekip üyesi olmayanlar, değer katmaktan (ör. Güvenlik kontrolleri).

1.6.1.2 Takip et

Çok sayıda takımın veri toplama ve planlamaya katılmasıdır. Süreç Verimliliği. Beklenti, bu seviyelerin süreç verimliliği, birbirlerinden yaklaşık 10 faktörlü takımlar. Verilerin toplanması meta verileri içerecektir. Verilerdeki belirli özniteliklerin belirli süreç seviyeleriyle güçlü bir şekilde ilişkilendirilir. Verimlilik. Süreç Verimliliği, kıyaslamaya uygun bu türden bir sabit aralık (% 0-100). Bu yüzde değil işin boyutuna veya karmaşıklığına bağlıdır. Bu ekip büyüklüğüne, kullanılan teknolojiye veya ekibin etki alanına bağlıdır. Yapabilecek şey ise süreci ölçmektir.

Çok sayıda farklı ekibin ve birçok farklı işleri olabilir. Yeterli veri noktamız varsa, süreç verimliliğinin ne kadar değiştiğini görülebilir. Birçok farklı ekip ve birçok farklı iş. Bu tür verilerden, herhangi bir takım kendi mevcut süreç verimliliği, takımlar ve ileri diğerleriyle karşılaştırır. Bundan böyle herhangi bir ekip ne kadar verimli ve ne kadar alanda gelişme var takip edebilir. Sonuçlara ve deneyime dayalı olarak kazanılan iyileşim noktaları daha sonra ekiplere yardımcı olmak için süreç verimliliğinde önemli ölçüde kazanç sağlar. İyileşmeleri daha küçük bir ekip grubunda uygulamak, onlardan ve sonuçlarından yararlanmaya dayanabilen işlem verimliliğ i ile ilgili iyileşimi uygulamak yaygınlaşmak açısından önemlidir. Pilotlama olarak düşünüle bilinir.

1.6.1.3 Akışa odaklanmak.

David Anderson, Kanban sürecini resmileştirdi. 2010'da yazılım geliştirme için [39] ve Henrik Kniberg [40] temel özellikleri güzel bir şekilde özetledi. Kanban ve bunları Scrum ile karşılaştırdı. Kanban var. Üç ana özellik - çalışmayı görünür kılma, sınırlama devam eden çalışma ve döngü süresini ölçmek. Scrum Ekip, Ürün Sahibi, Günlük Scrumlar, Sprint Planlama, Sprint İncelemesi ve Retrospektifler. Temel Scrum etkinlikleri ile yüksek performans elde etmek için Kanban'ın bir ekibe sahip olması gerekir. Günümüzde yüksek performanslı Scrum'lar sürekli teslimat yapıyor, esasen her bir biriktirme listesi öğesini olduğu gibi dağıtmak, tamamlandı, bu yüzden Kanban'a benziyor. Scrum Patterns Group, bir tampon için nasıl tampon kurulacağını göstermiştir. Scrum ekibi kesintiye uğramadan çalışmayı mümkün kılar. performans kaybı [41]. Böylece mevcut yaklaşım Maksimize etme akışı artık DevOps olarak adlandırılıyor. DevOps, Scrum kullanarak dağıtımı otomatikleştirmek anlamına gelir. Amazon'da olduğu gibi [41].

Kanban, döngü süresini ölçmektedir. Bir işin birim zamanda tamamlanması için gereken süredir. Bu bir işin birim seviyesinde iyi bir süreç verimliliği ölçüsüdür. Devam Eden Çalışmaları bunun

maksimum sınırını ve o iş için gereken döngü süresini, bir takımın ne kadar gelişime açık olduğu ile ilgili tahminleme konularında yardımcı olmaktadır. Süreç Verimliliği, kendi kendini organize olan takımların yaratılmasına yardımcı olur. Ekiplere net bir iç görü sağlayarak zamanlarını nerede harcadıklarını ve nasıl, nerede iyileşim noktaları olabileceğini tahminler. "İşleri Halletmek" [42] ile benzerdir. Süreç-Verimlilik, insanların ne kadarının zamanlarını alıyor, işi tamamlamaktan başka ekip nelere zaman harcadığı konusunda tahminlemeye yardımcı olduğu önerilir. Ekibin hedefleri nelerdir, takım hedefleri doğrultusunda mı iş yapıyor sorusuna yardımcı olur. Düşük performansa sahip takımları, nelerde tıkanıklık yaşıyorlar, ekipte iyileşimler için aksiyonlar belirlenebilir. Ekibin hangi hedeflere sahip olduğunu görerek üretkenlik benzer durumlarda başarıyla gerçekleştirildiğini desteklemektedir. Süreç verimliliği, ekibe iş durmaksızın baştan "tek parça sürekli akış" Toyota Üretiminin temel hedefidir. [43]. Süreç Verimliliği, ekibe yardımcı olan bir metriktir. Ekipler ne kadar verimli olduklarına dair fikir edinirler. Görevlerini ve işlerini tamamlayabilmek için akışa odaklamak sonuca ulaşmak için yardımcı olur. Lean ile birlikte Proses Verimliliğindeki uygulamalar, bir takıma çalışma şeklini iyileştirme konusunda yardımcı olur ve daha verimli hale gelmesini sağlar.Takım verimliliği birçok takımı takip ederek ve Süreç Verimliliği için bir kriter oluşturabilir. Verimlilik ile bir kıyaslama mevcut olduğunda, takımlar daha sonra ne kadar iyi yaptıklarını görebilir ve süreçlerindeki gelişimin ne kadar büyük olduğunu belirlemede yol gösterebilir. Süreç Verimliliği ekiplerin şu anda ne kadar iyi durumdalar, işlerini ne kadar iyi yapıyor anlamasına yardımcı olur. Bu, takımlara bir Story Puanlarının asla veremeyeceği bir bakış açısını katar. İyi ölçümlenen metrikler iyileştirmeleri teşvik eder. İyi ölçümler, takımın kendi ölçümlerini ölçmesine yardımcı olur. Performans ve gerçeklere dayalı değişiklikler yapılabilir. Story'nin aksine noktalar, bu ölçütler, akışı iyileştirme de anlamlıdır. Takım, yönetime, elmadan elmaya kadar birden çok takımın performansı verir. Ayrıca için tutarlı veri toplama, böylece ölçülen hiper üretkenlik açık ve genel olarak herkes tarafından anlaşılır olacak, iyi metrikler, Scrum Master'a sağlamlık kazandıracak şekilde bir çerçeve oluştururlar. Takım herhangi bir değişikliğin etkisini netleştirirler, yeni araçların ve teknolojilerin tanıtımından süreçteki veya hatta takım kompozisyonundaki değişikliklere kadar. Bu, haklı göstermesi gerekebilecek Scrum Masters'a yardımcı olabilir ek kaynak talepleri, bu kaynakların nasıl uygulanacağı ve etkisi, şirketin talep edilmesi durumunda bekleyebileceğini, bütçeyi. Temel metrik için formüller şu şekildedir:

1. Hız: Kabul edilen tüm çalışmaların orijinal tahminlerinin $\Sigma'\ddot{u}$.

2. İş Kapasitesi: Sırasında rapor edilen tüm çalışmaların toplamı. Sprint'in gidişatı hangi iş uygulandı ya da bitmedi.

3. Odaklanma Faktörü : *Hız ÷ İş Kapasitesi*

4. Kabul Edilen Çalışma Yüzdesi :

Σ (Kabul Edilen Çalışmanın Orijinal Tahminleri) ÷(Sprint için Orijinal Tahmin)

5. Bulunan İşin Yüzdesi

Σ (Bulunan Çalışmanın Orijinal Tahminleri) ÷ (Sprint için Orijinal Tahmin)

6. Tahmin Doğruluğu:

1- (Σ (Tahmin Deltaları) ÷ Toplam Tahmin)

7. Tahmin Doğruluğu

(ΣOrijinal Tahminler) Σ (ΣOrijinalTahminler + Σ Kabul Edilen Çalışma + Σ Bulunan İş)

8. Hedeflenen Değer Artışı (TVI +)

Mevcut Sprint'in Hızı ÷ Orijinal Hız

9. Ölçekte Başarı

Fibonacci Ölçeğindeki (Fp) her Nokta için, formül şudur:

(Σ No. Kabul Edilen Ölçek Girişimleri Fp) ÷ (No. Tüm Girişimlerin Ölçek Fp)

10. Galibiyet / Zarar Kaydı: Her Sprint yalnızca şu durumlarda bir Galibiyettir:

a) Orijinalin minimum% 80'i Tahmin Kabul Edildi

b) Sprint Sırasında Bulunan + Kabul Edilen Çalışma

Orijinalin% 20 veya daha azında kalır. Tahmin Hedeflenen Değer Katkısı arttı dediğimizde% 200, net ve kanıtlanabilir olmasını istiyoruz. Anlamı: Takımın becerisinin ikiye katlanması istenen karmaşıklığı başarıyla çözdü. TVC +(Hedeflenen Değer Katkı artışı),karlı bir şekilde uygulanan

artışı karşılaştırmak. Gelirdeki artışla birlikte ekibin beygir gücü. Ürün Sahibi'nin iş yığını tarafından oluşturulur. Doğal olarak Takımın Çalışmasının Kapasite (tam beygir gücünün ölçüsü), Hızlarından daha yüksek olabilmeli (ölçüsü çabalarını istenen ve onaylanan hale getirme yeteneği Değer). İşaretler İş Kapasitesi genellikle eşittir veya üzerindedir.

Hız: İş Kapasitesi nadir durumlarda düşebilir, Hızın altında. Bunun nedeni, Hızın Orijinal iş Tahminlerine göre hesaplanır. İş Kapasitesi toplama göre hesaplanırken Odaklanma Faktörü gittiğinde çok yüksekse, genellikle Takımın yeteneklerini tahmin edebilmesi önerilir. Bu grafiklerin "mükemmel" görünüyor veya diğer kurumsal yapıları görmezden geliyor olması yakın gelecekte patlayabilecek sorumluluklarında done olarak kullanılması önerilir. Bulunan Çalışma, bir parça ile ilişkili çalışmadır. Eskinin üstünde ve ötesinde olan Tahmin Çalışması başlangıçta beklenen ancak tamamlanması gereken orijinal çalışma öğesini teslim edilmelidir. Benimsenen Çalışma, öne sürülen çalışmadır. Ürün İş Listesinden Takım orijinalini tamamladığı için Sprint Erken tahmin edilir. Orijinal Tahminin yüzdesi olarak, bu ikisi birlikte eklendiğinde değerler %20'yi geçmemelidir. Ortalama bir durumda orijinal Tahmin, ortalama performansı değerlendirmek için Sprintin içine bakılır. Burada sunulan ölçümler. Tahmin Doğruluğu, Takımın şunları yapma becerisini ifade eder:

Sprint Planlama Toplantısında bir araya gelmeli, seçmeli ve kendilerini bir çalışma grubuna adamalıdır. % 80 doğruluk, elde edebileceklerini temsil eder. Önümüzdeki Sprint sırasında. Başaran herhangi bir Takım % 100 Doğruluk muhtemelen bir tür harici baskı ve bu nedenle düşük tahmin çalışması misilleme korkusu veya benzeri nedeniyle, işlevsiz dinamik. Bu numara gittiğinde% 90'ın üzerinde, Scrum Master'ın Ekibin ortamı hissettiklerinden emin olmak için daha fazla işte iyi niyetle çaba sarf etmek, hatta Tahminin en alt% 20'si kabul edilmediğinde Sprint' in sonunda. Tahmin Doğruluğu% 75-80'in altına düştüğünde genel olarak, Takımın yoğun şekilde rastgele seçilmesi veya Scrum tarafından yeterince korunmadığı söylenebilir. Sprint sırasında ustalaşın. Özellikle senaryolarda birden fazla Takımın Sprint Planlama yaptığı aynı gün, genellikle bir Ürün Sahibi ekiplerden biri, ekibinin ihtiyaçlarını başka bir ekibin Ürün Sahibi ile koordine edemiyor ve sonuçta bir Takımın Sprint' in başlarında omuzlar. İyi bir Scrum Ustası Sprint' in bundan korunduğundan emin olacak davranıştır. Harika bir Scrum Master, Ürün Sahipleri, Sprint Planning' den önce koordine edildi, böylece takımların Sprint Planlama toplantıları etkilidir. Organizasyonun başarısının yordayıcıları Sprint geliyor. Tahmin Doğruluğu takımın becerisini

yansıtır. Sprint sırasında işin gövdesini doğru bir şekilde tahmin etmek için Planlama. Bu sayı yine ortalıkta kalabilmelidir. Sağlıklı takımlarda% 80 iş. Tahmin Doğruluğu çok yükseldiğinde (ortalamada% 88'in üzerinde), muhtemelen takımın aşırı muhafazakâr olmak ve aşırı harcama yapmak istemediği, planlama, sindirme, araştırma ve zaman miktarı çoktur. Bu senaryolarda, ekip şunları yapabilir: biraz daha fazla risk kabul edin ve başarmak için daha fazla zaman harcamalı, iş, yapmaktan çok planlamaya zaman ayırmamalı. Bu genellikle daha kısa, daha yüksek verimlilik ve (daha kısa tercih edenler için toplantılar) daha mutlu takımlardır. Tahmin doğruluğu çok düşük olduğunda (ortalama% 72' nin altında), Scrum Master' ın takım üzerindeki baskıları araştırmaya başlamalı, sıklıkla Kullanıcı Hikayelerinin / PBI' larının çok zayıf olması durumu Ürün Sahibi' nin erişemeyeceği anlaşılmıştır. Takımın Sprint sırasında oldukları teknolojiyi veya ürünü anlaması önemlidir. Ürünü oluşturması / değiştirmesi isteniyor veya gereksinimlerin Sprint sırasında değişiyor. Ayrıca Bilgi Vampirine sahip olma potansiyeli sistem açısından kritik bilgileri biriktiren ekip ve herkesi karanlıkta tutmak, tüm bu yapısal açıklar düzeltilebilir ve bu metrik, Scrum Master' ın böyle olduğunda düzeltmeler gerekli ve düzeltmeler ne zaman istenen güven durumunu verir.

1.6.2 Sık Kullanılan Metrikler

Scrum ve Kanban kısımlarında belirtilen tüm metrikler verimlilik için kullanılabilir. Ek olarak sık kullanılan metrikler aşağıdaki gibidir.

Evidence-Based Management (Gelişim ve Çeviklik için Değeri Ölçme), organizasyonlar ın belirsizlik durumlarında müşterilerine sundukları değeri, organizasyonel yeteneklerini ve iş sonuçlarını sürekli olarak geliştirmelerine yardımcı olan deneysel bir yaklaşımdır. Kanıta Dayalı Yönetim, kuruluşların ürün teslimatlarından elde ettikleri değeri ölçmelerine, yönetmelerine ve artırmalarına yardımcı olmak için kullanabilecekleri bir çerçevedir. Evidence Based Management sonuçları iyileştirmeye, riskleri azaltmaya ve yatırımları optimize etmeye odaklanır.

Ken Schwaber ve Scrum.org tarafından geliştirilmiş ve sürdürülmüştür. Deneyler oluşturarak, deneyin sonucunu ölçerek, hedefleri ve sıradaki adımları gözlemleyip adapte ederek, gelişim yollarına karar vermeleri konusunda organizasyonlara yardımcı olur.[45]. Çevik kuruluşlar, sonuçların sık sık incelenmesinin riski sınırlandırdığını ve değer sağlama yeteneğini geliştirdiğ ini bilir. Liderleri yatırımları yatırım getirisine ve değere göre yönetir. Aynı zamanda, rakiplerinden

önce fırsatlardan yararlanmasına olanak tanıyan uyarlanabilir bir kültür yaratmak için kurumu etkilemek için çalışırlar.

Kanıta Dayalı Yönetim, kuruluşların doğru yerlere yatırım yapmak, daha akıllı kararla r almak ve yinelemeli ve artımlı bir yaklaşım kullanarak riski azaltmak için doğru önlemleri almalarına yardımcı olur.

Çevik ilkelerin ve değerlerin yanı sıra bu ampirik yöntem, organizasyon için başarılı değişim adımları sağlar.

Evidence Base Management Dört temel değer alanına odaklanır. Gerçekleşmemiş değerler, İnovasyon yeteneği, mevcut değer ve pazara çıkış süresidir. Bu dört boyuta odaklanmak, organizasyonların nerede olduklarını ve nereye gitmeleri gerektiğini daha iyi anlamalarını sağlar.

Şekil 2.20. :EMB Rapor



Kaynak : https://www.scrum.org/resources/evidence-based-management-guide

Şekil: EBM dört Temel Değer Alanına(Key Value Areas) odaklanır. [45]

Gerçekleşmemiş Değer (Unrealized Value): Organizasyonun tüm potansiyel müşterileri veya kullanıcılarının ihtiyaçlarını karşılaması durumunda gerçekleştirebileceği gelecekteki potansiyel değerleridir. Organizasyonlar gerçekleştirilmemiş değeri maksimuma çıkartmak için şu soruları sorulabilir: 1. Organizasyonumuz tarafından bu pazarda veya diğer pazarlarda herhangi bir ek değer yaratılabilir mi? 2. Kullanılmayan bu fırsatların peşinden koşmak için bir çaba harcamaya ve riske girmeye değer mi? 3. Ek Gerçekleşmemiş Değer elde etmek için daha fazla yatırım yapılmalı mı? Bu sorularla şu metrikler çıkabilir.

Tablo 1.3.: Gerçekleşmemiş Değer (Unrealized Value)

Pazar Payı Market Share	Ürünün kontrolü dışındaki pazarın göreceli yüzdesidir; müşteri ihtiyaçlarının daha iyi karşılanması durumunda ürünün elde edebileceği potansiyel pazar payıdır.
Müşteri veya Kullanıcı Memnuniyet Farkı Customer or User Satisfaction Gap	Bir müşteri veya kullanıcının arzu ettiği deneyim ile mevcut deneyimi arasındaki fark.
İstenilen Müşteri Deneyimi veya Memnuniyeti Desired Customer Experience or Satisfaction	Müşterinin sahip olmak istediği deneyimi gösteren bir ölçüm.

Kaynak : https://www.scrum.org/resources/evidence-based-management-guide

*Mevcut Değer: Current Value (CV) :*Bir organizasyonun müşterilerine ve paydaşlarına şu anda sunduğu değeri anlamaktır. Gelecekte var olabilecek değerleri değil, yalnızca şu anda var olanı dikkate alırlar. Organizasyonların Mevcut Değer için sürekli olarak yeniden değerlendirmeler i gereken sorular sorabilirler. Bu sorular:

1. Kullanıcılar ve müşteriler bugün ne kadar mutlu? Mutlulukları artıyor mu yoksa azalıyor mu?
2. Çalışanlarınız bugün ne kadar mutlu? Mutlulukları artıyor mu yoksa azalıyor mu?

3. Yatırımcılarınız ve diğer paydaşlarınız bugün ne kadar mutlu? Mutlulukları artıyor mu yoksa azalıyor mu? Sorularıdır. Bu sorularla şu metrikler çıkabilir.

Şekil 2.21. : Mevcut Değer: Current Value (CV)

Mevcut Değer: Current Value (CV)

KVM	Ölçüm:
Çalışan Başına Gelir Revenue per Employee	Bu oran (brüt gelir / çalışan sayısı), bir sektördeki önemli bir rekabet göstergesidir. Bu oran, sektörden sektöre önemli ölçüde değişiklik gösterir.
Ürün Maliyet Oranı Product Cost Ratio	Ölçülen ürün veya sistem için, operasyonel maliyetler de dahil olmak üzere, toplam gider ve maliyetlerin gelire oranıdır.
Çalışan Memnuniyeti Employee Satisfaction	Çalışan bağlılığını, enerjisini ve işe olan heyecanını ölçmeye yardımcı olacak bir tür duygu analizi.
Müşteri Memnuniyeti Customer Satisfaction	Ürünle ilgili müşteri bağlılığını ve mutluluğunu ölçmeye yardımcı olacak bir tür duygu analizi.

Kaynak : https://www.scrum.org/resources/evidence-based-management-guide

Pazara çıkış süresi (Time to Market) : Organizasyonun yeni özellikleri, hizmetleri veya ürünleri piyasaya hızlı bir şekilde sunma yeteneğidir. Organizasyonların Time-to-market için sürekli olarak yeniden değerlendirmeleri gereken sorular şunlardır: 1. Organizasyon yeni deneylerden ve bilgilerden ne kadar hızlı öğrenebilir? 2. Edinilen bilgiye göre ne kadar hızlı adapte olabilirsin iz?
3. Yeni fikirleri müşterilerle ne kadar hızlı test edebilirsiniz? . Bu sorularla şu metrikler çıkabilir.

Şekil 2.22. : Pazara Çıkış Süresi

Derleme ve Entegrasyon Sıklığı Build and Integration Frequency	Belli bir dönem içerisinde entegre ve test edilmiş derlemelerin sayısı. Kullanıcılarına düzenli olarak ürünün yeni versiyonlarını sunabilen bir takım için, bu ölçümün yerini gerçek sürüm ölçümleri alır.
Sürüm Frekansı Release Frequency	Belli bir dönem içerisindeki sürüm sayısı, ör. sürekli, günlük, haftalık, aylık, üç aylık vb. Bu, yeni ve rekabetçi ürünlerle müşteriyi memnun etmek için gereken süreyi yansıtmaya yardımcı olur.
Sürüm Stabilizasyon Periyodu Release Stabilization Period	Geliştiricilerin ürünün piyasaya sürülmeye hazır olduğunu söylediği nokta ile müşterilere gerçekten sunulduğu nokta arasındaki ürün sorunlarını düzeltmek için harcanan zaman. Bu, kötü geliştirme pratikleri ve altında yatan tasarım ve kaynak kod etkisini göstermeye yardımcı olur.
Ortalama Onarım Süresi Mean Time to Repair	Bir hatanın tespit edildiği andan düzeltildiği ana kadar geçen ortalama süre. Bu, organizasyonun bir hatayı düzeltme etkinliğini göstermeye yardımcı olur.
Müşteri Döngü Süresi Customer Cycle Time	Bir sürümde çalışmanın başladığı andan gerçekte yayınlandığı ana kadar geçen süre. Bu ölçü, organizasyonun müşterisine ulaşma yeteneğini göstermeye yardımcı olur.
Akış Süresi Lead Time	Bir fikrin önerildiği veya bir hipotezin oluşturulduğu andan müşterinin bu fikirden yararlandığı ana kadar geçen süre. Bu ölçü, müşteri ve ürüne göre değişebilir. Müşteri memnuniyetine katkıda bulunan bir faktördür.
Değişiklikler için Akış Süresi Lead Time for Changes	Kodun teslim edildiği andan canlı ortamda kodun başarıyla çalıştırılmasına kadar geçen süre. Daha fazla bilgi için bkz. DORA 2019 report.
Dağıtım Sıklığı Deployment Frequency	Organizasyonun, ürünün yeni sürümünü müşterilere / kullanıcılara dağıtma (yayınlama) sayısı. Daha fazla bilgi için bkz. DORA 2019 report.
Hizmet Geri Yükleme Süresi Time to Restore Service	Hizmet kesintisinin başlaması ve hizmetin tekrar tamamen geri yüklenmesi arasındaki geçen süre. Daha fazla bilgi için bkz. DORA 2019 report.
Öğrenme Süresi Time-to-Learn	Bir fikri veya iyileştirmeyi tasarlamak, geliştirmek, kullanıcılara sunmak ve kullanıcı deneyiminden öğrenmek için gereken toplam süre

Kaynak : https://www.scrum.org/resources/evidence-based-management-guide

İnovasyon Yeteneği (Ability to Innovate):

Bir organizasyonun müşteri ihtiyaçlarını daha iyi karşılayabilecek yeni özellikler sunma etkililiğidir. Organizasyonlar inovasyon yeteneğini artırmak için için sürekli olarak yeniden değerlendirmeleri gereken sorular şunlardır:

1. Organizasyonun yeni değer sunmasını engelleyen şeyler nelerdir?

2. Müşterilerin veya kullanıcıların bu yenilikten yararlanmasını engelleyen şeyler nelerdir? Soruları sorulabilmeli ve sorularla şu metrikler çıkabilir.

Şekil 2.23. : İnovasyon Yeteneği

İnovasyon Oranı Innovation Rate	Yeni ürün özelliklerine harcanan çaba veya maliyetin toplam ürün eforuna veya maliyetine yüzdesel oranıdır. Bu, organizasyonun yeni ürün özellikleri sunma kapasitesi hakkında fikir verir.
Hata Trendi Defect Trends	Son ölçümden bu yana hataların değişim trendinin ölçümü. Hata müşteri, kullanıcı veya organizasyonun kendisi için ürünün değerini düşürür. Hatalar genellikle amaçlandığı gibi çalışmayan şeylerdir.
Ürün Üzerinde Çalışma Endeksi On-Product Index	Takımların ürün ve değer üzerinde çalışmak için harcadığı sürenin yüzdesi.
Yüklü Version Endeksi Installed Version Index	Mevcutta destek verilen ürünün sürümlerinin sayısı. Bu, organizasyonun eski yazılım sürümlerini desteklemek ve sürdürmek için harcadığı çabayı yansıtır.
Teknik Borç Technical Debt	"Hızlı ve kirli" çözümlerin daha sonra iyileştirmelere ihtiyaç duymasından kaynaklı ekstra geliştirme ve test çalışmalarını yansıtan bir programlama kavramıdır. Teknik borç, değer üretimi üzerinde istenmeyen bir etkiye ve önlenebilir israf ve risk artışına yol açar.
Üretim Vaka Sayısı Production Incident Count	Belirli bir dönem için, Geliştiricilerin yayında olan üründeki bir sorunu düzeltmek için bölünme sayısı. Üretim Vaka sayısı ve sıklığı, ürünün stabilitesini göstermeye yardımcı olabilir.

Kaynak : https://www.scrum.org/resources/evidence-based-management-guide

Market share chart: Pazar payı, bir şirket veya kuruluş tarafından kazanılan bir pazarın bölümünü veya yüzdesini ifade eder. Pazar payı hesaplaması, bir şirketin belirli bir dönemdeki toplam satışlarını ve o dönem içinde faaliyet gösterdiği sektörün toplam satışlarını dikkate alır. Pazar payı, şirketin dönem içindeki satışlarının aynı dönemdeki sektörün toplam satışlarına bölünmesiyle hesaplanır. Bu metrik, bir şirketin pazardaki yerine ve rakiplerine göre büyüklüğü hakkında bir fikir vermek için kullanılır.

Innovation Rate: yeni ürün özelliklerine harcanan çaba veya maliyetin toplam eforuna veya maliyetine yüzdesel oranıdır. Bu organizasyonun yeni ürün özellikleri sunma kapasitesi hakkında fikir verir.

Unplanned work: Ürün Sahibi, Geliştirme Takımının çalışmasından kaynaklanan ürünün değerini maksimize etmekten sorumludur. Bunun nasıl gerçekleştiği her kuruluş için farklıdır. Ama insanları PO'ya yönlendirerek talebi Ürün İş Listesi ve Sprint İş Listesindeki diğer her şeyle karşılaştıracaktır. PO, mevcut Sprint İş Listesinin yeni talepten daha ağır bastığını hissederse, Ürün İş Listesine gider ve uygun şekilde plana dâhil edilebilir. Acil ilgilenilmesi gerektiğini yeterince önemli hissederlerse, PO ile developer Ekibi arasında bir konuşma başlar. Mevcut sprint' e girebileceklerini ve işi Sprint Hedefini tehlikeye atmadan tamamlayabileceklerini hissederse, bu gerçekleşir. Sprint Hedefini tehlikeye atabileceğini düşünürlerse, Ürün İş Listesine konulacak ve daha yüksek sıraya konulacaktır. Bir PBI' ın değiş tokuş edilmesini gerektireceğini ve bu takasın Sprint Hedefini tehlikeye atmayacağını düşünüyorlarsa, o zaman o takas yapılır. Ancak tüm bunlar, Scrum Takımının mevcut olmasıyla olur. Bunu genellikle Günlük Scrum' ın hemen ardından yaparız.

Employee engagement score: çalışanın bağlılığını, enerjisini ve işe olan heyecanını ölçmeye yardımcı olacak bir tür duygu analizi yapılması önerilebilir.

2. ARAŞTIRMANIN AMACI VE VERİLER

Değişen pazarda, piyasa koşullarında, taleplerde olan ve teknolojide meydana gelen gelişim, değişim değişimler organizasyonları meşgul hale düşürmekte ve farklı yönelimlere sokmaktadır. Organizasyonların varlıklarını sürdürebilmesi için şuan ki durumlarını hem korumak hem de avantaja çevirebilmek için agile (çevik) organizasyon modelini geliştirip, uygulamaya başlamışlardır. Birçok kaynağa göre agile üretim tarihi seri üretim ve yalın üretimden sonra gelmektedir. Agile metot şuan ki organizasyon yapısını kendiliğinden doğurmuş ve sürüklemiştir. Dünyamızda oluşan inovasyonları takip edebilmek, değişimi kolayca uyum sağlayabilmek ve/veya üstelikte değişimlerin öncüsü rolünü sahiplenmek üzere organizasyonlar agile tercih sebebidir. Hızlıca gelişen müşterilerin talepleri, kalite ve/veya teslimat tarihinden taviz göstermeden yerine getirmek, pazarda yarışın içerinde yer almak ve/veya pazardan büyümek için gereklilik haline gelmiştir. Belirsizliğin içerisinde gerekli kararları alabilmek agile organizasyonlarda mümkün olur. Odağına müşteriyi almak, koşullara hızlı cevap verebilen üretim içeren çevik organizasyonlar, takım içindeki iletişime ve ürünün değerine katkı sağlamaktadır. Karar verme merci iş biriminden teknoloji birimlerine indirmektedir. Bu da müşterinin taleplerini karşılama süresini azaltmakta ve organizasyonlara oldukça katkı sağlar. Agile ile alakalı değişik incelemeler de değişik tanımlar ve agile pratikleri vardır. Organizasyonlar değişimi yakalayabilmek, ayak uydurabilmek ve değişim de öncü olabilmek için organizasyon yapısını çevikleştirmelidirler. Bir banka örneğinde; potansiyel banka müşterisini, yatırımlarını iyi değerlendirmeyi talep eden potansiyel müşterileri var. Müşteri kitlesinin amacı, var olan yatırımın ı en iyi şekilde değerlendirmek ve yatırımını en iyi şekilde yönetmek istemektedir. Bu hedef kitlesinde olan müşteri açısından ideal ürününü, hizmeti ve çözümü baştan sonra oluşturup ilerletebilecek bir ekip olsun. Bu ekibin bünyesinde yazılım geliştirmeden operasyona, satın almadan pazarlamaya kadar farklı yetkinliklerin bir araya gelir. Bu takımın, her bir üyesinin de aynı ortak hedef için çalışacak, bir ile dört haftalık süreler içerisinde ürünü geliştirerek yapılan ürünü test edecek ve sürekli değer üreten üretimi gerçekleştirecektir. Geleneksel yöntemlerdeki organizasyon yapılarınsa ise bir fikrin pazarlamadan çıkıp, projelendirilip analizinin yapılması ve yazılıma geçmesi, operasyona aktarımı oradan satışa geçme sürecini düşünürsek yeni kurulan bu çevik takımın ne kadar hızlı bir şekilde fikirler üretip test ederek öğrenen, geri bildirim ile gelişen bir organizasyona dönüşeceği gözlemlenir.

Tablo 2.1: Çeviklik ile ilgili örnek çalışmalar

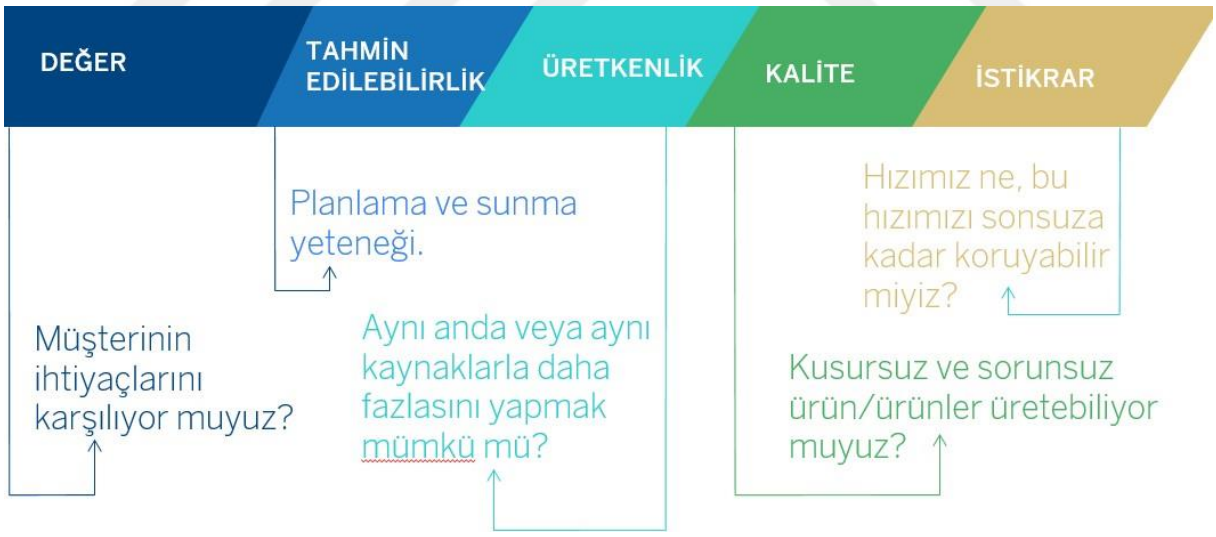
YAZAR	YIL	KONU	SONUÇ
Buğra Kocatürk	2009	Planning Agile Software projects with reduced guess estimation	Çevik yazılım sürecindeki belirsizlikler ve yaklaşımlar incelenmiş, proje oluşturulmuştur.
Emrah Aşan	2014	Agile and collaborate system enginnering	Çeviklik hakkında iki temel sonuca varılmıştır. Bunlar modellemedeki eş zamanlılık ve çapraz işbirliğidir.
Berker Sazan	2019	Çevik Yazılım Geliştirme ve Çevik Dönüşüm: IT Sektöründe bir uygulama	Müşteri memnuniyetinin sağlanmasına yönelik bir uygulama yapılması hedeflenmiş, bu bağlamda çevik yazılım geliştirme ve SCRUM metodu ile müşteri memnuniyeti arasındaki ilişki araştırılmıştır.

Yapılan araştırmalar da yukarıdaki sonuçlara ulaşılmıştır. Çevik Yazılım Geliştirme de çevik dönüşüm ve verimlilik bankacılık IT sektöründeki çevik dönüşüm ve verimlilik konuları hakkında herhangi bir çalışma tespit edilmemiştir. Bu sebeple yapılan çalışmanın alanında ilk olma özelliği taşıması ve güncel kavramları incelemesi açısından ilk olma özelliğini taşımaktadır. Teze konu olan bankanın verimlilik ile ilgili kullandığı metrikler ve dataları aşağıdaki gibidir.

Bu datalar şirketin Agile olmadan önceki ve Agile olduktan sonraki karşılaştırmalı datalarıdır. Bu datalar da

- Değer,
- Tahmin edilebilirlik,
- Üretkenlik,
- Kalite,
- İstikrar açısından incelenmiştir. Bu incelemeler yapılırken takım sayılarından, takımların çıktılarından, time-to- market verilerinden, proje deployment zamanlarına, kalite metriklerine, müşteri memnuniyetlerine, devops metriklerine bakılarak grafikler oluşturulmuştur.

Şekil 2.24. : Verimlilik



Kaynak : Teze bahsi olan şirketin verimlilik başlığı

Şekil 2.25. : Verimlilik başlıkları

DEĞER

DEĞER	Müşterinin ihtiyaçlarını karşılıyor muyuz?	1. Projeler 2. Maintenance ve problem kayıtları zamanında, kaliteli ihtiyaç karşılanıyor mu? **Açılan ve kapanan problem kayıtları adet ve süreleri.** **Açılan ve kapanan projeler ve adetleri.** Yapılan geliştirmeler ile **iş süreçleri ve zamanlarında elde edilen kazanç.** **Inovasyon projeleri oranı.**

TAHMİN EDİLEBİLİRLİK

TAHMİN EDİLEBİLİRLİK	Ne durumdayız? Planlama ve sunma yeteneği.	**Planlamada tahminlenen SP ve Efor** ile **gerçekleşen SP ve Efor** karşılaştırması. Süreçlerin başında **talep edilen proje adetleri** ile **gerçekleşen proje adetleri.** **Lead/cycle time scrum ve kanban** her ikisi için kullanılabilir mi? **Proje ve maintenance** işlerin planlamasının ve tahminlemesinin verimli yapılarak takım tarafından işlerin verimli yönetilmesi : **Takımların proje ve diğer işlere ayırdığı zamanların quarter bazında karşılaştırılması.**

KALİTE

KALİTE	Kusursuz ve sorunsuz ürün/ürünler üretebiliyor muyuz?	**Kod kalitesi -** Takımların Code Review sonuçları. **Test bulgu adeti.** Test kalitesi - **Üretime alındıktan sonra yaşanan test bulgu adeti.** **Test otomasyon adetleri.** **Agile olgunluk kalitesi :** Sertifika sayıları (PSM, PSPO, PAL) **Çevik Ayna Sonuçları (Takımlar kendilerini nasıl görüyor) :** Agile **ritüeller** bazında. Agile **roller** bazında. Quarter bazında **bug tekrar adet** karşılaştırması.

ÜRETKENLİK

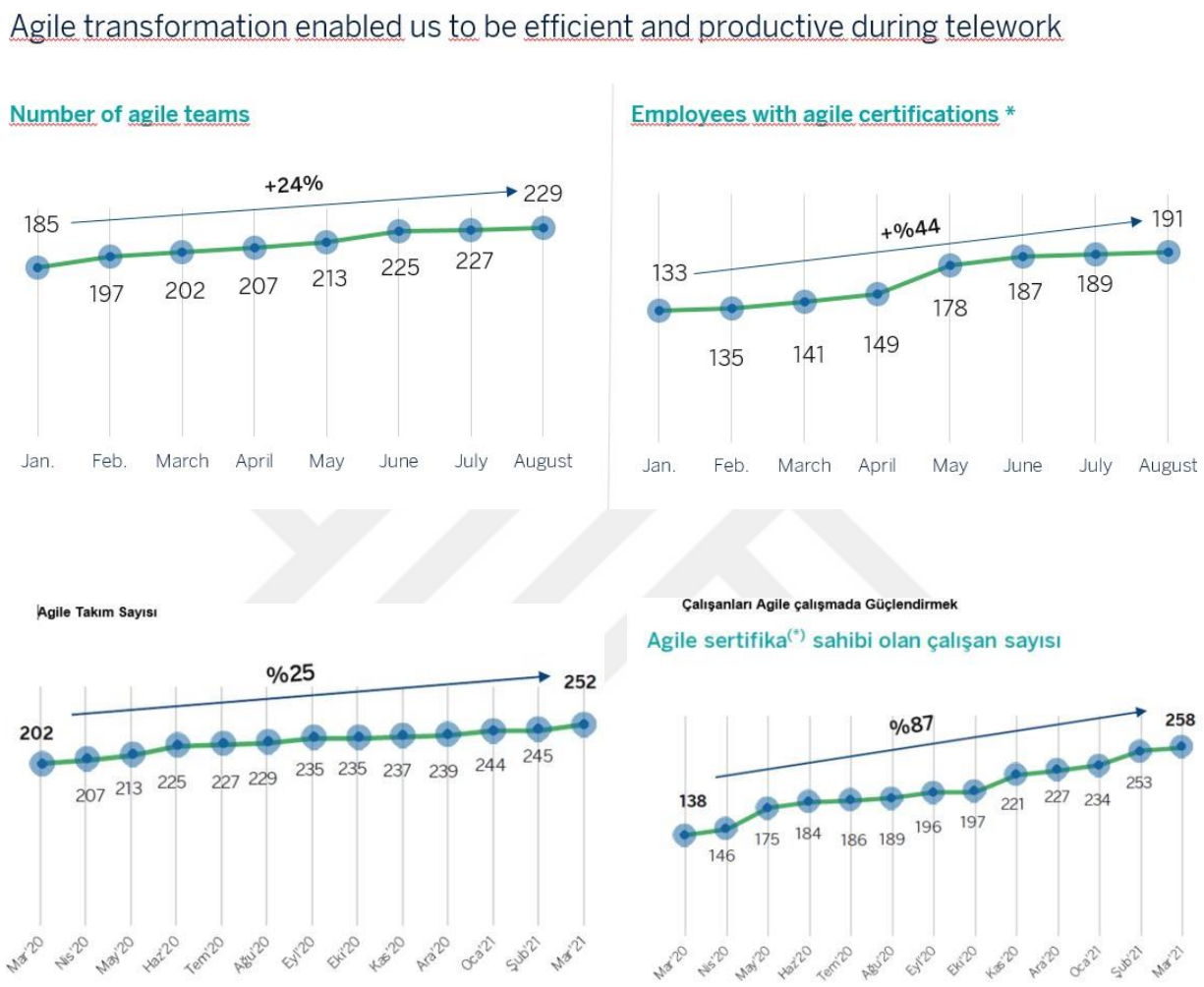
ÜRETKENLİK	Aynı anda veya aynı kaynaklarla daha fazlasını yapmak mümkü mü?	**Gerçekleşen SP ve Efor sprint/iterasyon ilerleyiş** takibi. **Release Yönetimi :** Quarter bazında takımların **release sıklığı** karşılaştırılması. Quarter bazında takımların **release iş adedi** karşılaştırılması. **Çalışan üretkenliği :** Quarter bazında kullanılan **mazeret, rapor izin oranları. Çalışan motivasyonunun** üretkenlik üzerinde etkisi. **Deployment sayıları -** prolemleri ayırarak new feature bazında.

İSTİKRAR

İSTİKRAR	Hızımız ne, bu hızımızı sonsuza kadar koruyabilir miyiz?	**Takımların quarter bazında sprint/iterasyon SP ve Efor değerleri.** **Kod ve test kalitesindeki istikrar.** **Sürdürülebilirlik analizi** için quarter bazında fazla mesai saat oranları. **VC call ve webex davetleri verilerinden** yola çıkarak subject bazında **Agile ritüel toplantı sürelerinin karşılaştırılması.**

Kaynak : Teze bahsi olan şirketin verimlilik başlığı

Şekil 2.26. : Takım Sayısı

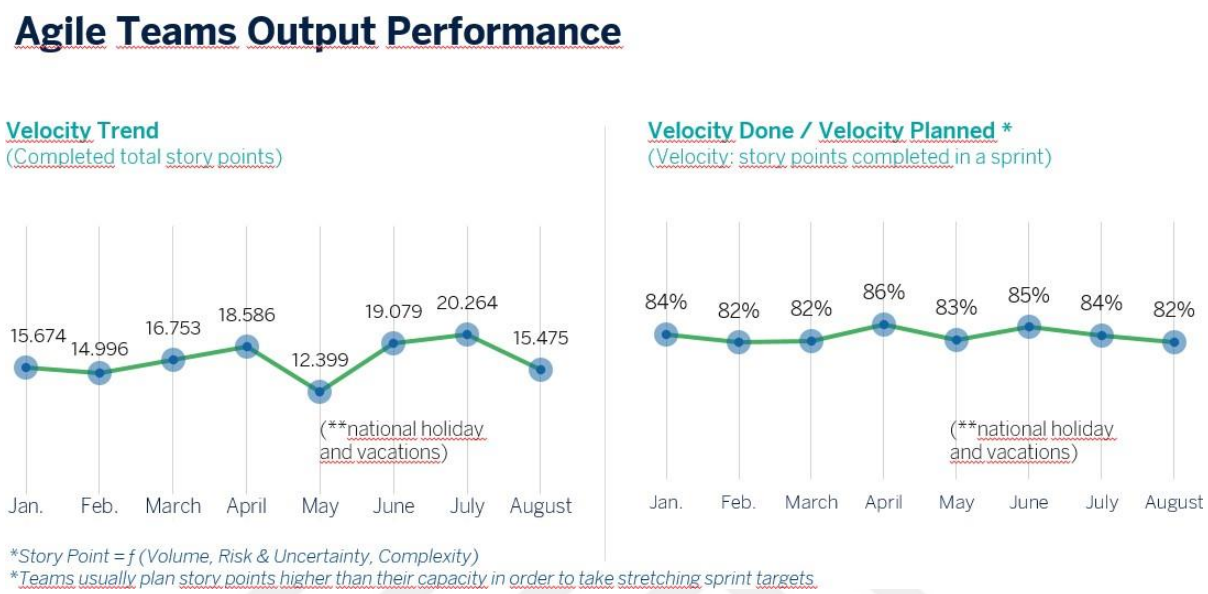


Kaynak : Teze bahsi olan şirket

Number of Agile Teams grafiğinde şirket dönüşümünde Ocak 2020'de 185 olan takım sayısını Ağustos 2020'de 229 olduğunu ve Mart -2020-Mart 2021 grafiğine bakıldığında ise bu sayının 252 olduğu görülmüştür. Özetle; zamana bağlı olarak takım sayısının artış yaptığı gözlenmiştir.

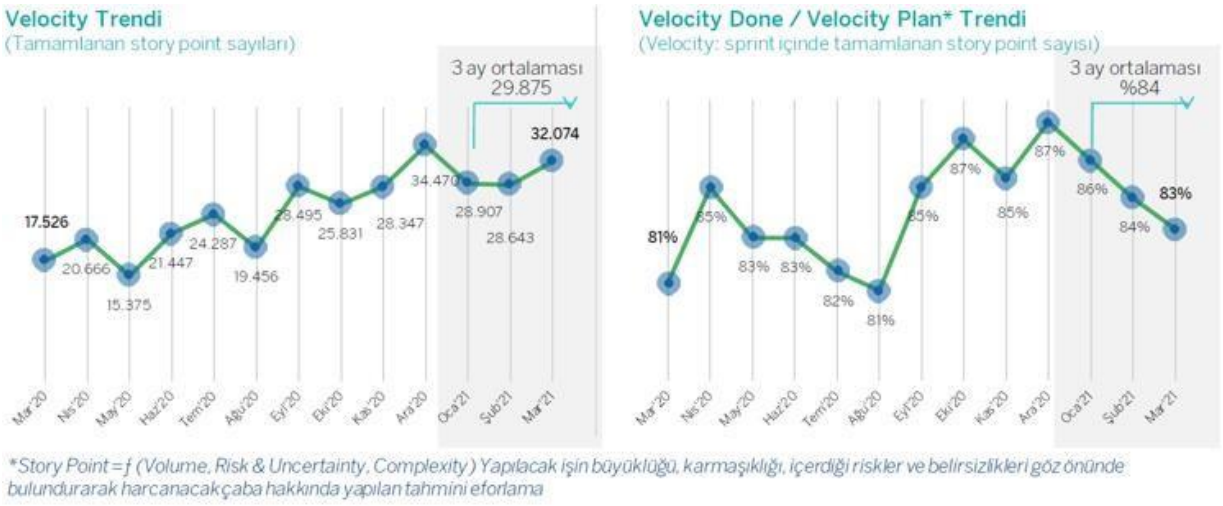
Employees with Agile Certification grafiğinde şirketin çalışanlarını sertifikasyonla desteklenmesinin sonucu gözlemlenerek Ocak-2020'de 133 olan sertifikalı çalışan sayısının Ağustos 2020'de 191 olduğunu ve %44 artığı, Mart 2020- Mart 2021 grafiğinde ise bu sayısının 258 olduğu gözlemlenmiştir.. Bu sertifikalar ise Scrum.org ve eşleniği Uluslararası sertifikalardır.

Şekil 2.27. : Agile Takım Çıktısı



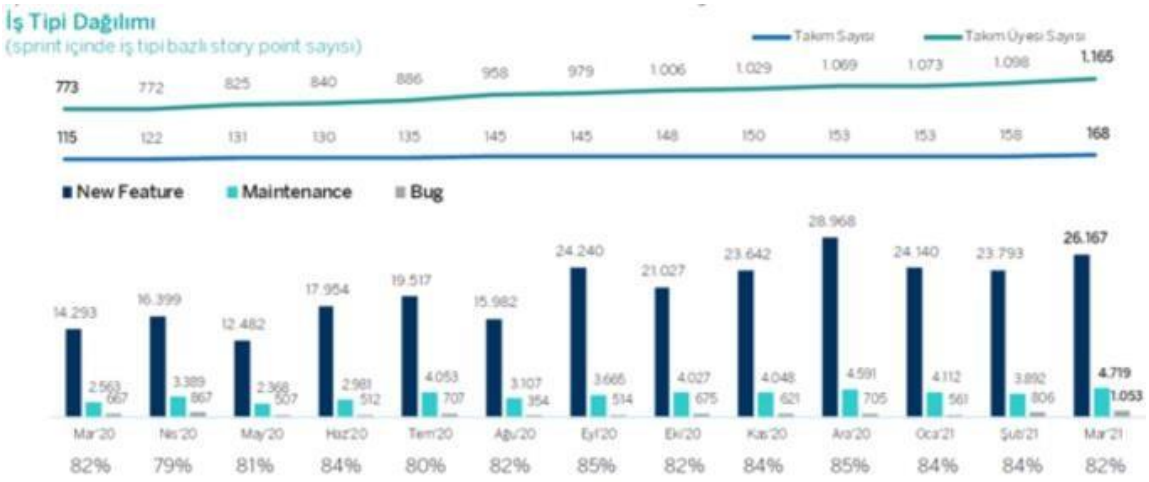
Kaynak : Teze bahsi olan şirket

Agile takımların hız grafiğinde her takımın bir sprintte veya bir iterasyonda yaptıkları çıktılar bakıldığında zamanla artış gözlemlenmiştir. Waterfall çalışan şirketin agile çalışma şekline geçtiğinde ise bu velocity grafiğinin Ocak -2020 ve Ağustos 2020'de %82 oranında yapılan işlerin çıktısına katkı sağladığı gözlemlenmiştir.



Scrum takımlarında 2020 Mart ayından, 2021 Mart ayına göre story point bazında bu artış %83 olmuştur.

Şekil 2.28. :İş Tipi Dağılımı



Kaynak : Teze bahsi olan şirket

Scrum takımlarının 'New Feature' olarak yaptıkları işlere bakıldığında işe sürekli artışın olduğu gözlemlenmiştir. Özellikle bu sayı Mart-2021'de takımların aldıkları işlerin %82'sinin yeni iş üretmeye efor harcadığı gözlemlenmiştir.

Şekil 2.29. : Throughput Trendi



Kaynak : Teze bahsi olan şirket

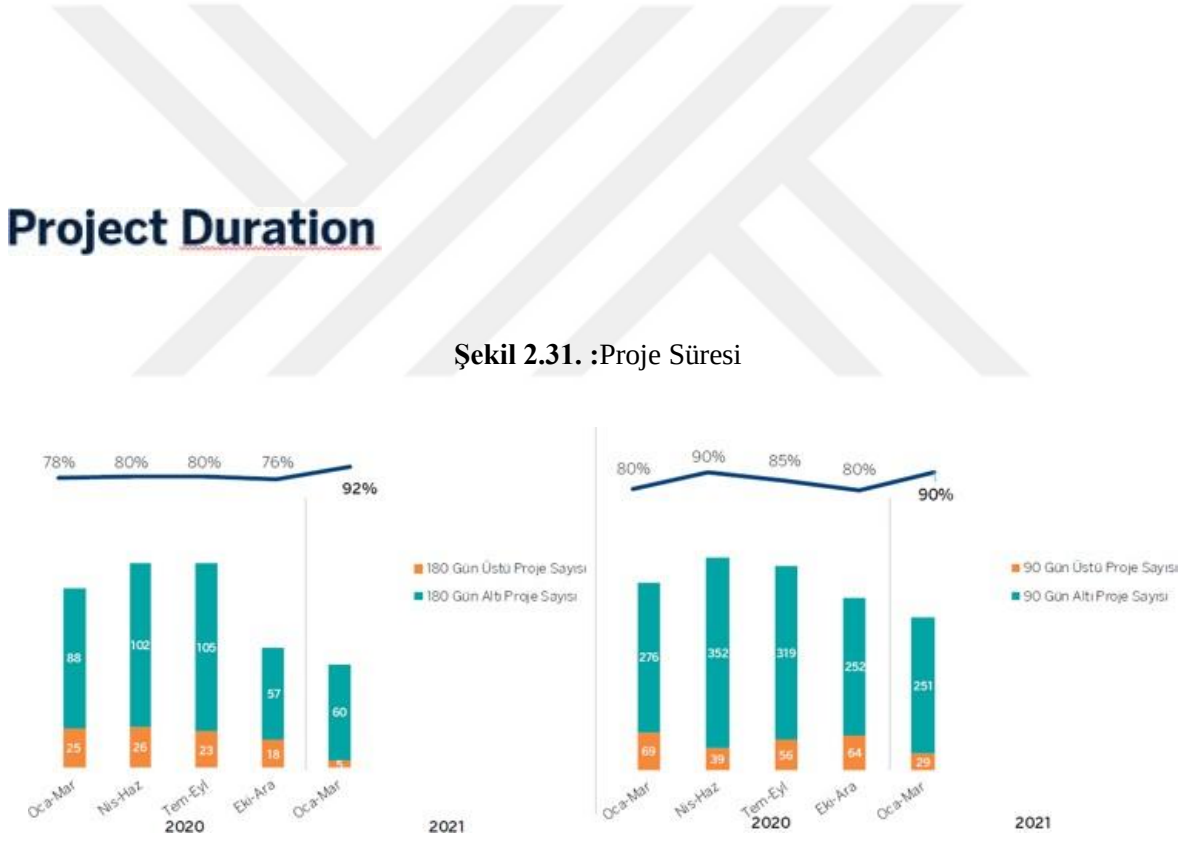
Kanban takımlarında ise Throughput grafiklerine Haziran-2020 ve Mart-2021 bakıldığında yapılan işlerde artışın olduğu gözlemlenmiştir.

Şekil 2.30. : Deployment Performansı



Kaynak : Teze bahsi olan şirket

Şirketin genelinde yapılan projeler ölçümlendiğinde bu projeler için müşterilere sık teslimat için deployment sayısının artış yaptığı gözlemlenmiştir.



Şekil 2.31. : Proje Süresi

Kaynak : Teze bahsi olan şirket

180 günden kısa zamanda tamamlanan büyük projelerin ve 90 günde tamamlanan küçük projelerin MVP yaklaşımı ve Time to Market'te artan bir performansın olduğu gözlemlenmiştir.

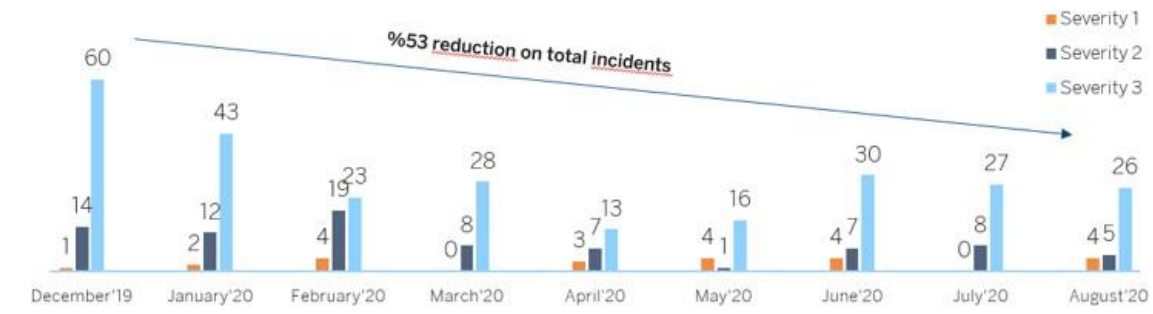
Şekil 2.32. : Problem Sayısı



Kaynak : Teze bahsi olan şirket

Kalite metriklerinde ekiplere gelen problem kayıtlarına bakıldığında Ocak ayı ile Ağustos ayı arasındaki dönem de problem kayıtlarının azaldığı görülmüştür. Açılan problem kayıtlarının ise hızlı bir şekilde çözüldüğü açılan problem kayıt oranı ile çözülen problem kayıt oranın bire bir örtüştüğünü grafikler göstermiştir.

Şekil 2.33. : Vaka



Kaynak : Teze bahsi olan şirket

Kalite metriklerinden Acil- olaylar metriğine bakıldığında; etkili ekip seferberliği ve olaylara sanal olarak hızlı müdahalenin öncelikli 1,2,3 olarak gruplandığında bu oranın Aralık 2019'a oranla %53 oranında azaldığını Ağustos 2020 'de gözlemlenmiştir. Bu grafikte acil olayları ATM, Mobil Bankacılık, POS, Para Transferleri, Ödeme Sistemleri birimlerinin dataları incelenerek grafik oluşturulmuştur.

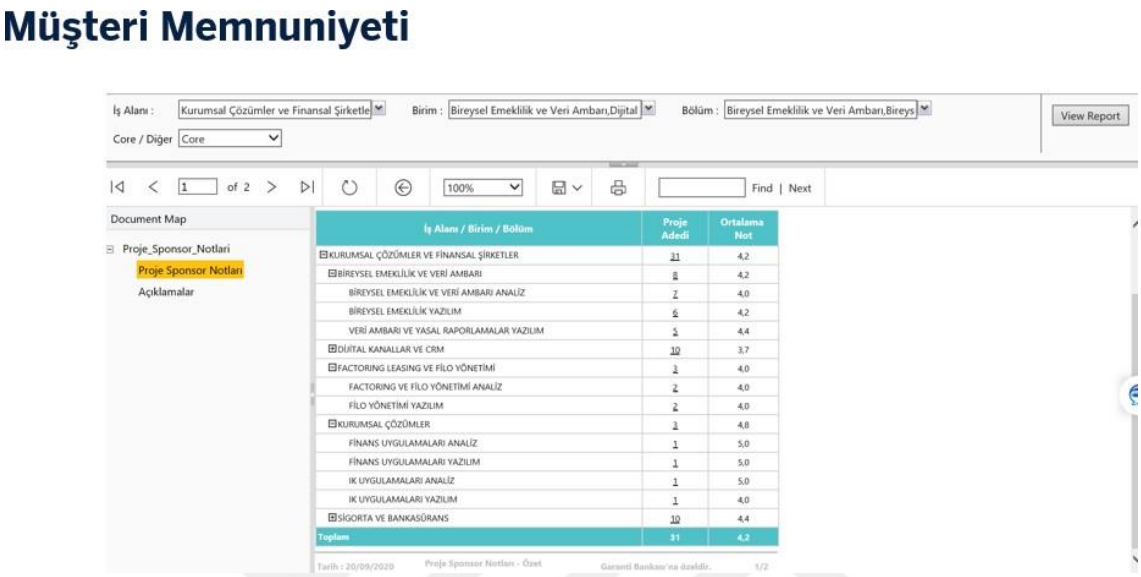


Şekil 2.34. : Problemi Çözme Süresi

Kaynak : Teze bahsi olan şirket

Şirketin SLA(Service Level Agreement) metriği ölçümlerinde; bu sürenin hedeflerine kısa sürede ulaştığını ve kesintisiz hizmet ile müşteri memnuniyetini artırdığı gözlemlenmiştir.

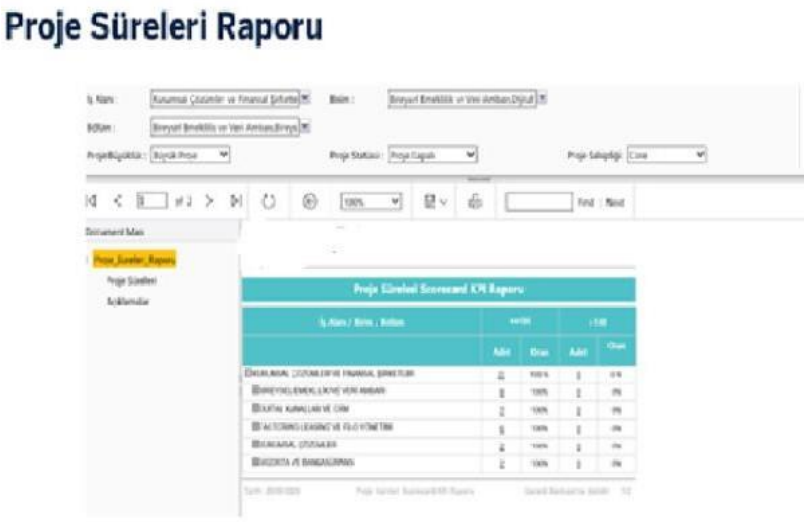
Şekil 2.35. :Müşteri Memnuniyeti



Kaynak : Teze bahsi olan şirket

Müşteri Memnuniyeti ölçümlerinde birim bazlı paydaşlardan geri bildirimler alınıp bu grafik oluşturulmuştur. Müşterilere beş üzerinden sunulan ankette memnuniyetin 4,2 olduğu grafiklerde görülmüştür.

Şekil 2.36. : Proje Süresi



Kaynak : Teze bahsi olan şirket

Şekil 2.37. : Proje Adetleri



Kaynak : Teze bahsi olan şirket

Proje süreleri ve Proje adetleri de metrik olarak şirkette tutulmuştur. Bu metrikler de ekiplerin büyük projelerini 180 günde küçük projeleri ise 90 günde tamamlandığı gözlemlenmiştir.

Şekil 2.38. : Pazara Çıkış Süresi

Time-To-Market (Lead Time)

	Average of Kapanış/Açılış Farkı (Gün)	Distinct Count of Talep No
2013	1.555	71
2014	1.284	208
2015	807	451
2016	466	1.076
2017	370	1.554
2018	336	1.126
2019	200	912
2020	101	405

Kaynak : Teze bahsi olan şirket

Şekil 2.39. : Projenin Süresi

Time-To-Project

	Average of Proje Açılış - Talep Açılış (Gün)
2017	102
2018	110
2019	104
2020	40

Kaynak : Teze bahsi olan şirket

Şirketin time-to market ve time to Project metriklerinde projelerin müşterilere sık teslimat yapıldığını ve projelerin açılış ile kapanış farkının azaldığı gözlenmiştir.

Şekil 2.40. : Kod Commit Adeti



Kaynak : Teze bahsi olan şirket

Kod geliştirme de Git-Commit adetlerin de ve teste verme de artan performansın olduğu gözlemlenmiştir.

Şekil 2.41. : Kod Satır Sayısı



Kaynak : Teze bahsi olan şirket

Kod satır sayısında (LineofCode) bakıldığında Mart2020-Ağustos2020 aralığında kod sayısının artığı yeni kod geliştirme de artan performansın olduğu gözlemlenmiştir.

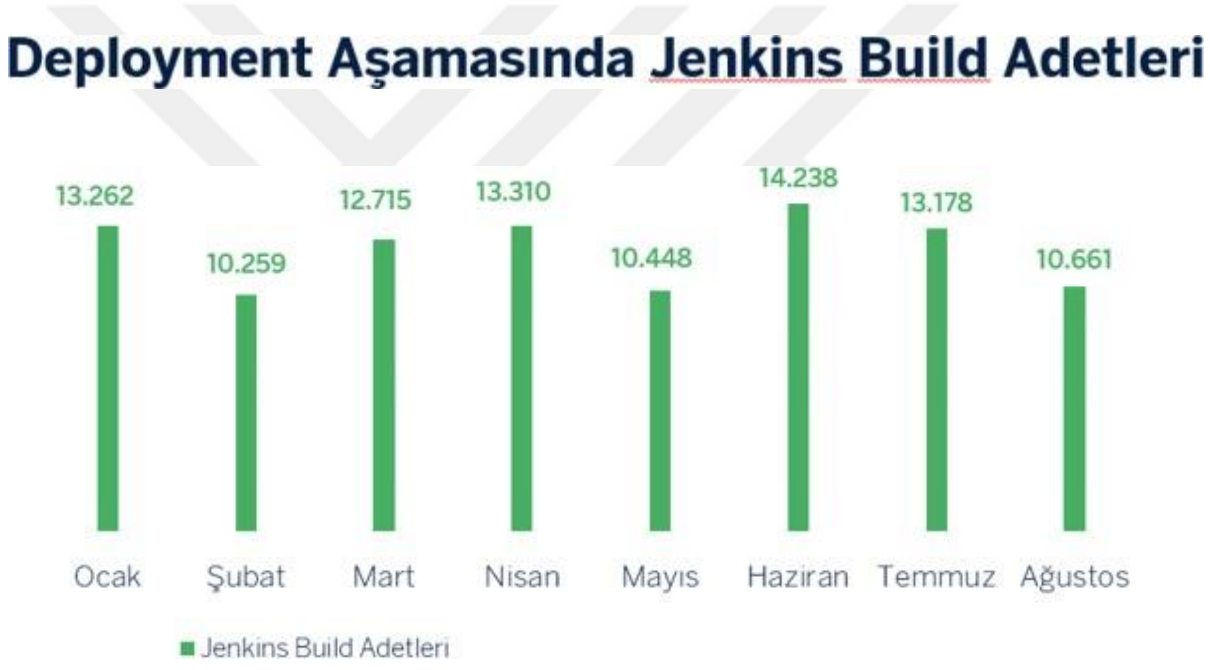
Şekil 2.42. : Kodun Tarama Sayısı



Kaynak : Teze bahsi olan şirket

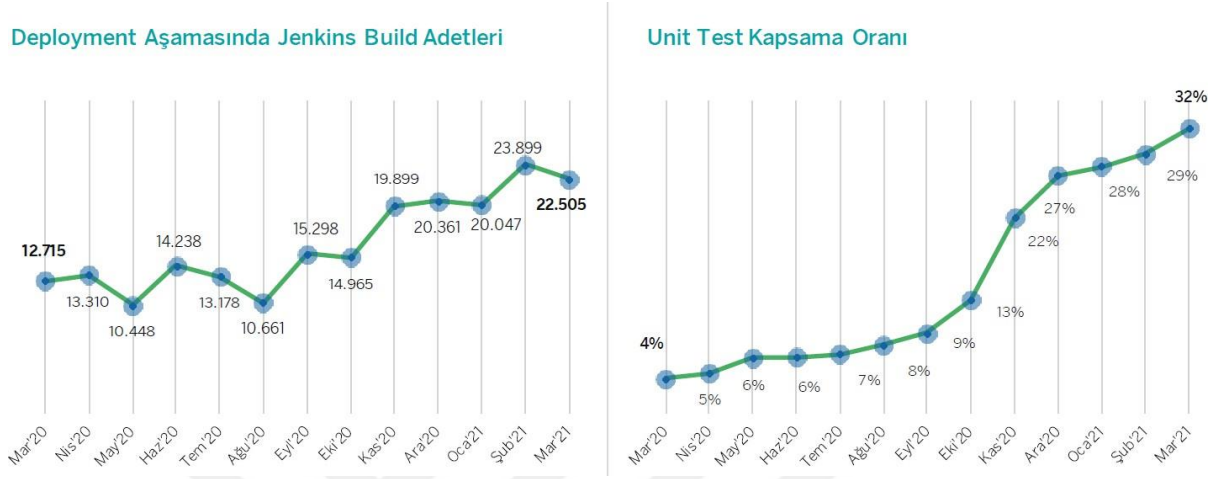
Teste verme aşamasında SonarQube tarama adetleri teste vermede artan performans sergilediği görülmüştür.

Şekil 2.43. : Kodun Built Adedi



Kaynak : Teze bahsi olan şirket

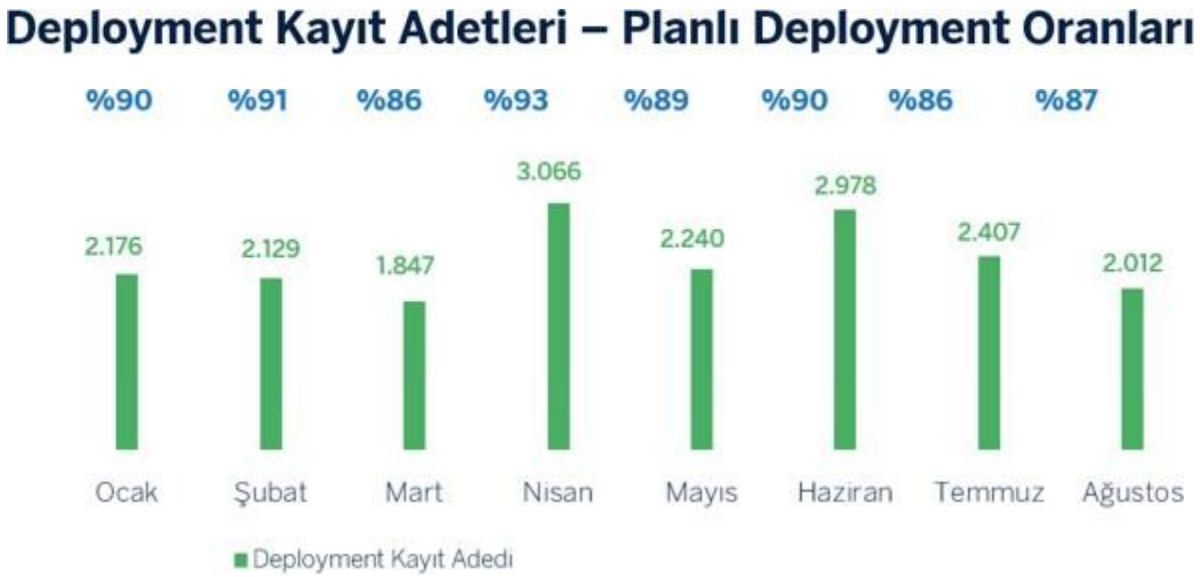
Şekil 2.44. : Jenkins Tarama Sayısı



Deployment aşamasında Jenkins Build adetlerinin ve unit test kapsama oranın sürekli entegrasyonda artan performansın olduğu gözlemlenmiştir.

Kaynak : Teze bahsi olan şirket

Şekil 2.45. : Deployment oranı



Kaynak : Teze bahsi olan şirket

Şekil 2.46. : Planlı Deployment Oranı



Kaynak: Teze bahsi olan şirket

Üretkenlik metriğinde Planlı deployment oranlarında time to markete etkisi olmuştur. Şirketin planlı deployment oranın %93 seviye de olduğu gözlemlenmiştir.

3. ARAŞTIRMANIN BULGULARI

Bu çalışma ile Bankacılık bilgi teknolojileri sektöründe yazılım geliştirme ve çevik dönüşüm konusunun araştırılması ve teze bahis olan sektörde dönüşümün nasıl ilerlediği, müşteri memnuniyeti sağlanmasında ve şirkete kattığı verimliliğe yönelik incelemelerin yapılması amaçlanmıştır. Bu kapsamda çevik yazılım geliştirme ve Agile metodu ile bankacılık yazılım sektöründe dönüşüm ve dönüşüm sonucu verimliliğin nasıl olduğuna yanıt alınması amaçlanmıştır. Sonuç olarak, yeni dijital dünyada adapte olabilen ve değişimi temel esas alabilen ve benimseyen şirketlerin dünyası olacaktır. Bu sebeple artık yeni dönemi çevikleşen şirketler pazarda yerini almaya devam edecektir. Öte taraftan çevikliği kazanmak çok kolay değildir. Çevikleşebilmek için mevcut şirket yapısının ve şirket kültürünün, iş yapış şekillerinin baştan değişmesi gerekmektedir. Dönüşümü gerçekleştiren tezde bahsi olan şirkette dönüşümün başında ve ortalarında sıkıntı, zorluklar da yaşamıştır. Şirkette yer alan yöneticilerin varlığını nasıl devam ettireceğinin olması, organizasyonda yönetici altı çalışanların süreci bilmemeleri çeşitli endişelere yol açmıştır. Ancak burada yapılan Agile Yaklaşımlar ile ilgili şirketteki tüm birimlere ve her kademedeki kişilere bilgilendirmenin yapılması, üst yönetim desteği ve dönüşümü anlatmak olumlu etki olarak doğurmuştur. Şirket, Agile çalışmada rolleri (Scrum Master, Kanban Master, Lider, Developers) güçlendirmenin önemini fark etmiş ve bu rolleri güçlendirmek için güçlendirme programları düzenlemiş ve bu rollerdeki çalışanlara güçlendirme eğitimleri verilmiştir. Şirketin gelişimi destekleme kapsamında Agile düşünce şeklinde de yer alan sürekli gelişim odaklı yaklaşmak çalışanlara Temel Agile Scrum ve Kanban eğitimleri ve sertifikasyon programlarını sunması ile de artı değer üretmiştir. Bu temel eğitimler, sertifikasyon eğitimleri ve güçlendirme eğitimlerinin şirkete yüklü maliyetleri vardır. Ancak şirket kendi bünyesinde kurmuş olduğu Çevik Metot Ekibiyle bu eğitimlerin şirket içerisinde verilmesine olanak sağlayarak bütçeyi azaltmıştır. Kurulan Çevik Metot Ekibinin (Agile Koç) farklı fikirlerden, farklı birimlerden ve farklı şirketlerden olması kıymetli olmuştur. Çevik Metot ekibi, Agile farkındalığın oluşması, gelişmesi, çalışanların temel eğitimleri, takım oluşumları, takımların Agile yolculuklarındaki temel ihtiyaçlarına cevap vermek, güçlendirme ve sertifikasyon program eğitimlerinin içeriğinin tüm ekip üyelerinin verebilecekleri şekilde olabilmesi tasarlanarak hazırlanması ve verilmesi,

organizasyonel çeviklik, sistem alt yapısının tasarlanması ve hayata geçirilmesi, verimlilik, agile sürdürebilir olması açısından çalışmalarına devam etmektedir. Takım dönüşümleri sırasında takımlara Scrum ya da Kanban çalışılması şeklinde diretmektense takımın yöneticileri, takımda yer alacak kişiler ve Çevik metot yönetimi (Agile Koç) ile birlikte oluşacak takımların iş yapış şekilleri, kültürleri konuşularak belirlenmiştir. Ortak kararla belirlemek, sebeplerini şeffaflıkla konuşmak oluşacak direnci kırmış, takımların Agile'ı benimsemesine katkı sağlamıştır. Takım dönüşümü sırasında oluşacak takımlar yönetim kadrosu tarafından yapılmıştır. Takımın Çapraz fonksiyonlu olmasına özen gösterilmiştir. Takım bir başka takıma bağımlığını en aza indirgeyecek şekilde kendi içerisinde yazılım sürecini uçtan uca tamamlayacak haliyle oluşturulmuştur. Kurulan ekiplere kendi kendilerine organize olabilme yetkisi verilmiştir. Şirkette kullanılan toollar çevikleştirilmiştir, çevikleşme için bu alana yatırım yapılmış, toolu kullanan şirket çalışanlarından sürekli geri bildirim ile geliştirmeleri devam etmektedir. Şirketin yaptığı değişikliklerden bir tanesi de performans sistemidir. Yeni oluşturulan performans sistemi çevikliğe uygun olacak şekilde tüm takımın birbirini değerlendirdiği, müşterilerden ve sponsorlardan alınan geri bildirime göre sürekli gelişimi destekler hale gelmiştir. Ancak takımın sayısal verileri (Hız grafikleri, bitti sayısı) performansa yansıtılmamıştır. Çünkü şirket her takımın farklı dinamiklerinin olduğunu, takımların hızlarını karşılaştırılması gerektiğini farkındadır. Organizasyon takımların iyileştirme noktalarını manipüle edilecek dataların olabileceğini farkındadır, burada da çevikliğin değerlerinden birisi olan güveni benimsemiş, içselleştirmiştir ve çalışanına, takımdaki üyelere güvenmiştir. Şirket çevikliği canlı tutacak aktiviteler düzenlemiş ve düzenlemeye devam etmektedir. Geri bildirim mekanizmasına önem verip bu kültürü desteklemek için şirketin herkes tarafından görülecek alanlarına Plus & Deltalar koymuştur. Çalışanlar bu alanlara hem Agile dönüşüm hem de bir birleriyle ilgili geri bildirimler vermektedir. Ayrıca her çalışan düzenli olarak bağlı olduğu liderle kısa zaman aralıklarıyla bir araya gelip geri bildirim alıp vermişler, yönetici çalışanın paylaşımlarıyla birlikte gelişimi desteklemiştir. Şirket teşekkür ve takdir kültürünü de benimseyerek Kudo Walllar yapıp, çalışanların birbirlerine teşekkür etmesini sağlayacak ortam sağlamıştır. Üst yönetim her zaman takımları desteklemiş, takımlara desteğini göstermek, takımların ihtiyaçlarını konuşmaya fırsat vermek ve ürünü yerinde görebilmek için Gemba Walk ismi verilen aktivitelerle takımlarla buluşmuştur. Farklı firmalardan Agile ile çalışan kişilerle bir araya gelinip, deneyim paylaşımı ve soru, cevap etkinlikleri yapılmıştır. Bahse konu olan Bankada verimlilikte her zaman önemli olmuştur. Verimliliği ölçümleyebilmek ve gözlemleyebilmek adına

toollar ile desteklenmiştir. Oluşturulan bu tool anlık ölçümler yapabildiği gibi, takımların kendi istedikleri sürelerde ve metriklerde veri sunmaktadır. Bu ölçümlemeler şeffaftır, takım üyeleri, yöneticiler, paydaşlar, sponsorlar ve müşteriler tarafından erişilebilirdir. Takımların kendi Agile olgunluk seviyelerini kendilerinin değerlendirdiği, takip edebilecekleri alanlarda mevcuttur. Şirketteki Agile dönüşümün gidişatını, takımların taleplerini ve sahayı dinlemek, müşterilerden geri bildirim almak amacıyla Çevik Metot ekibi takımlarla iletişimde, bunu anket yoluyla yapmakta ve anket sonuçlarını da tüm organizasyonla açık, şeffaf bir şekilde paylaşmaktadır. Şirkette Agile metriklerle ölçümlemeler de hem projenin canlıya geçiş süreleri, müşteri memnuniyeti, verimlilik, kalite ve işbirliği ile ilgili ölçümleri yukarıdaki şekillerde paylaşıldığı üzere takip etmiştir ve etmeye de devam etmektedir.

4. SONUÇ VE ÖNERİLER

Bu tez çalışmasında Türkiye'deki bir Bankanın, teknoloji sektöründe yazılım geliştirme ve çevik dönüşüm konusunu incelenmesi ve bu bankanın hem dönüşüm için neler yaptığı hem de Agile metodu ile çalışarak eski çalışma şekli olan waterfalle göre verimlilik arasındaki ilişkisi araştırılıp veriler ile analizi yapılmıştır.

Agile (Çevik) yazılım geliştirmesinin en önemli özelliğinin müşteri memnuniyetini attırmak, müşterinin ihtiyaçlarına hızlı cevap verebilmek ve değişen koşullara hızlı adapte olabilmek hedefi değerlendirirken bir taraftan da şirkete kattığı verimlilik açısından değerlendirilmeye çalışılmıştır.

Şirket dönüşümü sırasında mutlaka çalışanın gelişimi her kademe de desteklenmelidir. Agile çalışmadaki roller güçlendirilmelidir. Üst yönetim desteği önemlidir ve mutlaka üst yönetimden destek alınabilmelidir. Üst yönetimin takımlara desteği olmalı ve bu desteği gösterebilmek için çeşitli aktiviteler düzenleyebilirsiniz. Takımların dönüşümü sırasında çalışanlara eğitimler ile sürekli gelişim desteklenmelidir. Takımlara Scrum ya da Kanban çalışmalısınız şeklinde diretmek yerin takımlar, yöneticiler ile bir araya gelerek iş yapış şekillerini konuşup, takımın yapısına göre Kanban ya da Scrum olarak takımları dönüştürmelisiniz. Sadece çalışma şekli değil, kültürü, alt yapısı, sistemleri, çevikliği destekleyecek şekilde olmalı, Scrum'ın değerlerine sahip çıkmalı, içselleştirmelisiniz. Sözde Agile olmak değil, değerler desteklemelisiniz şirketinizi.

Organizasyonlarınızda yapabiliyorsanız ya çevik metotlarda dönmenin hem şirket, hem çalışan hem de sistemler üzerinden daha verimli, daha kazançlı sonuçları gözlemlersiniz. Ancak başarılı bir dönüşümü yapabilmek için bir şirketin direk örneğini alıp onu kendi şirketinizde uygulamak yerine, kendi şirketinizin kültürünü, kendi şirketinizin çalışanlarını ve kendi şirketinizin stratejilerini göz önünde bulundurarak iyi pratiklerini uyumlayarak denemelisiniz. Başarılı olup olmadığımızı anlamak için ölçümleme yapabilirsiniz.

Sonuç olarak aşağıdaki tablodan yaralanabilirsiniz.

Tablo 5.1. : Geleneksel Yöntem/Agile Yöntem Karşılaştırması

Geleneksel Yöntemle çalışan organizasyon/şirketlerde	Agile (Çevik) Metot ile çalışan organizasyon/şirketlerde
Fonksiyonalite temelli yapılanma.	Değer odaklı ve farklı yetkinliklerin birlikte çalışmasıyla oluşan takımbazlı.
Hiyerarşik, ast-üst ilişkisine dayalı yönetim anlayışı.	İş birlikçi, temelinde takım oyunu olan yönetimyapısı.
Kişileri pushlayarak çalıştıran yönetimve yöneticiler.	Takıma destek olan liderler.
Verimlilik ve operasyon odaklı.	Üretkenlik ve değer odaklı.
Proje planları ve bu proje planına sadık kalma kontrolü.	Sürekli daha iyisini planlama, değişen koşullara adapte olabilme.
Detaylı planlamalar, mükemmel ürün / hizmetin tek bir seferde (en sonda) üretime alınması.	Minimum değerde, ufak adımlarla ve öncelik sıralaması yüksekdeğerin daha öncelikli olarak sürekli, sık üretime alma.
Hiyerarşik raporlama, varsayımlar ve hissiyata bağlı karar mekanizmaları.	Radikallik, şeffaflık, veriye dayalı karar alma mekanizması.
Uzun süren onay alma süreçleri, otorite onaylarıyla ilerleme.	Onaya ve karar mekanizmasına bağlı, baştan sonra sorumluluğun takımda olduğu, kendi kararlarının alabilen öğrenmeye açık akımlar.
Hata yapma müsamahasının kısıtlı olduğu	Deneyselliğe ve hatalardan öğrenmeye dayalı.
Mekanik, hantallaşmış organizasyonel yapı.	Yaşayan, şirketin stratejilerine paralel şekilde değişime açık organizasyonel yapı.
Bireysel performansa bağlı performans sistemi.	Stratejiler ve takımın iş hedeflerine bağlı takım performans sistemi.
Odağın yatırımcıda olduğu.	Odağın müşteride ve çalışanda olduğu.
Detaylı planlamalar, mükemmel ürün / hizmetin tek bir seferde (en sonda) üretime alınması.	Minimum değerde, ufak adımlarla ve öncelik sıralaması yüksek değerin daha öncelikli olarak sürekli, sık üretime alma.
Hiyerarşik raporlama, varsayımlar ve hissiyata bağlı karar mekanizmaları.	Radikallik, şeffaflık, veriye dayalı karar alma mekanizması.
Uzun süren onay alma süreçleri, otorite onaylarıyla ilerleme.	Onaya ve/veya karar vermeye bağlı, baştan sonra sorumluluğun takımda olduğu, kuvvetli ekipler.
Hata ihtimalinin belirli ölçütlerde olduğu	Deneyselliğe, hatayasürekli öğrenmeye dayalı.
Mekanik, hantallaşmış organizasyonel yapı.	Yaşayan, şirketin stratejilerine paralel şekilde değişime açık organizasyonel yapı.
Bireyselperformansa bağlı performans sistemi.	Stratejiler ve takımın iş hedeflerine bağlı takım performans sistemi.
Odağın yatırımcıda olduğu	Odağın müşteride ve çalışanda olduğu.

REFERANSLAR

[1] Chakravorty, Pragnan (2018). "What is a Signal? [Lecture Notes]". IEEE Signal Processing Magazine. 35 (5): 175–177. Bibcode:2018ISPM...35.175C. doi:10.1109/MSP.2018.2832195.

[2] Yagup Macit, Eray Tüzün, "Uygulama Yaşam Döngüsü Yönetimi Karşılaştırmalı Süreç İncelemesi", 9'uncu Ulusal Yazılım Mühendisliği Sempozyumu, 2015

[3] Ayşe Betül Karagöz, Fatma Molu, "Finans Sektörü Yazılım Süreçlerinde Şelale Modelinden Scrum Modeline Geçiş", 2nd Internatıonal Symposium On Innovatıve Technologıes In Engıneerıng And Scıence, 2014

[4] https://tr.wikipedia.org/wiki/V-Model_[Yaz%C4%B1l%C4%B1m_geli%C5%9Ftirme)

[5] http://ceng.gazi.edu.tr/~hkaracan/source/YPY_H3.pdf

[6] http://newton.cs.concordia.ca/~paquet/wiki/images/e/ea/Spiral_model.gif

[7] http://ecomputernotes.com/software-engineering/formal-methods-model

[8] Şeyda Ocak, Medine Yıldıztekin, "Güvenli Yazılım Geliştirme Süreç Modellerinin Karşılaştırılması Uygulamaları", Elektrik-Elektronik Bilgisayar Sempozyumu, 2011

[9] Sanjana Taya, Sheveta Gupta, "Comparative Analysis of Software Development Life Cycle Models", International Journal of Computer Science and Technology Vol. 2, Issue 4, 2011

[10] L.Alexander and A. Davis, 1991. "Criteria For Selecting Software Process Models". In Proceedings of COMPSAC'91 Tokyo, Japan, 09-11-1991 to 09.13.1991, pp. 521-528

[11] Craig Layman and Victor Basili, "Iterative and Incremental Development: A Brief History", IEEE Computer, 2003.

[12] Prateek Sharma, Dhananjaya Singh, "Comparative Study of Various SDLC Models on Different Parameters", International Journal of Engineering Research, 2015

[13] https://www.cs.odu.edu/~zeil/cs350/f14/Public/processModels/

[14]http://www.isanalizinedir.com/index.php?option=com_content&view=article&id=54:agile&catid=39:yazilim-gelistirme-metodoloji&Itemid=56

[15] Amlani,Radhika D., "Analysis and comparative study of various software development process models", thesis PhD, Saurashtra University, 2013

[16] https://agileportal.net/scrum-metrikleri/

[17] Güney, S. (2017) Yönetim ve Organizaston El Kitabı, Nobel Yayınevi, Ankara.

[18] İşcan, Y , Karabey, A . (2015). Bilgi Teknolojilerinin Benimsenmesi İle Örgütsel Atiklik

[19] Arasındaki İlişki Üzerine Bir Araştırma. Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 0 (26), 1-17.

[20] Hormozi A.M, (2001). Agile manufacturing: the next logical step", Benchmarking: An International Journal, 8 (2), 132-143.

[21] Singh, J.,Sharma, G., Hill, J., Schnackenberg, A, K. (2013). Organizational agility: What it is, what it is not, and why it matters".Article in Academy of Management Annual Meeting Proceedings.

[22] Kasap, G , Peker, D . (2009). çevik üretim: otomotiv ana sanayinde faaliyet gösteren bir işletmenin çevikliğinin ortaya konmasına yönelik bir araştırma. Elektronik Sosyal Bilimler Dergisi, 8 (27), 57-78.

[23] Sharifi H, Zhang Z (1999). A Methodology for achieving agility in Manu Factoring organizations:introduction, International Journal OF Production Economics, 62(1) 7-22.

[24] T. Sakai, The Secrets Behind the Success of Toyota: Privately Published, 2018.

[25] T. Ohno, Toyota Production System: Beyond Large-Scale Production: Productivity Press,1988.

[26] W. Myers, "Why Software Developers Refuse to Improve," Computer, vol. 31, pp.

112, 110-111, 1998.

[27] C. R. Jakobsen and J. Sutherland, "Scrum and CMMI Going from Good to Great," in Agile Conference, 2009. AGILE '09., 2009, pp. 333-337.

[28] H. Takeuchi and I. Nonaka, "The New New Product Development Game," Harvard Business Review, 1986.

[29] M. L. George, Lean Six Sigma: Combining Six Sigma Quality with Lean Speed: McGraw Hill, 2002.

[30] ScrumInc. (2018, 15 Jun 2018). Scrum Courses and Training. Available: https://www.scruminc.com/scrumtraining/scrum-courses-list/

[31] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive Guide to Scrum, The Rules of the Game," scrumguides.orgMarch 2017.

[32] J. Sutherland, N. Harrison, and J. Riddle,"Teams that Finish Early Accelerate Faster: A Pattern Language for High Performing Scrum Teams," in 2014 47th Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI, USA, 2014.

[33] J.-D. Lovelock, A. O'Connell, W. L. Hahn,A. Adams, D. Blackmore, R. Atwal, S. H. Hong, N. Gupta, and P. Niketa, "Forecast Alert: IT Spending, Worldwide, 1A18 Update," Gartner, Stamford, CT2018.

[34] J. Sutherland and J. Sutherland, Scrum : the art of doing twice the work in half the time, First Edition. ed. New York: Crown Business, 2014.

[35] N. Modig and P. Åhlström, This is Lean: Resolving the Efficiency Paradox: Rheologica Publishing, 2014.

[36] S. Shingo, Fundamental Principles of Lean Manufacturing: Productivity Press, 2017.

[37] N. Forsgren, J. Humble, and G. Kim,Accelerate: Revolution Press, 2018.

[38] C. Ladas, Scrumban and Other Essays on Kanban Systems for Lean Software Development. Seattle: Modus Cooperandi Press, 2008.

[39] D. J. Anderson, Kanban: Successful Evolutionary Change for Your Technology Business: Blue Hole Press, 2010.

[40] H. Kniberg, Kanban and Scrum - making the most of both: lulu.com, 2010.

[41 ScrumPlop.org. (2018, 26 Aug). Published Patterns: Illigitimus Non Interruptus.

[42] R. Monica, "Amazon Deployment Data from Roy Monica, Head of Engineering Amazon Devices Demand Forecasting," Seatlle: Amazon.com, 2018.

[43] D. Allen, Getting Things Done: The Art of Stress-Free Productivity, Revised Edition. New York: Penquin Books, 2015.

[44] J. Coplien, "An Alternative to Kanban: One-Piece Continuous Flow," in Scruminc.com vol. 2018, ed. Cambridge, MA: Scrum Inc.,2011.View publication

[45] https://www.scrum.org/resources/evidence-based-management-guide?gclid=CjwKCAjwhYOFBhBkEiwASF3KGYDlsfV-EzzLx2UKev1_PzaMTS-hyYwG-oHFt0xk5EsOfTPYFhupxBoCjCUQAvD_BwE