

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
İŞLETME PROGRAMI

YAZILIM GELİŞTİRME SÜREÇLERİNDE SCRUM’I TERCİH
EDEN İŞLETMELERİN ÇEVİK DÖNÜŞÜMÜNE ETKİ EDEN
FAKTÖRLERİN İNCELENMESİ

YÜKSEK LİSANS TEZİ

ERHAN ÖZLÜ

HAZİRAN-2019

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
İŞLETME PROGRAMI

**YAZILIM GELİŞTİRME SÜREÇLERİNDE SCRUM’I TERCİH
EDEN İŞLETMELERİN ÇEVİK DÖNÜŞÜMÜNE ETKİ EDEN
FAKTÖRLERİN İNCELENMESİ**

YÜKSEK LİSANS TEZİ

ERHAN ÖZLÜ

DANIŞMAN

DR. ÖĞR. ÜYESİ M. FEVZİ ESEN

HAZİRAN-2019

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim.

İmza
Erhan Özlü

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığını taahhüt ederim.

Dr. Öğr. Üyesi M. Fevzi Esen

İMZA SAYFASI

Erhan ÖZLÜ tarafından hazırlanan “Yazılım Geliştirme Süreçlerinde Scrum’i Tercih Eden İşletmelerin Çevik Dönüşümüne Etki Eden Faktörlerin İncelenmesi” başlıklı bu yüksek lisans tezi, İşletme Anabilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ

İMZA

Tez Danışmanı

[Dr. Öğr. Üyesi M. Fevzi Esen]

.....

Kurumu: İstanbul Medeniyet Üniversitesi

Üyeler:

[Dr. Öğr. Üyesi Arafat Salih Aydın]

.....

Kurumu: İstanbul Medeniyet Üniversitesi

[Dr. Öğr. Üyesi Yakup Çelikkalek]

.....

Kurumu: İstanbul Gelişim Üniversitesi

Tez Savunma Tarihi: 28/06/2019

ÖNSÖZ

Günümüzde teknolojik alanlarda çok hızlı gelişim ve değişim yaşanmaktadır. Teknoloji alanında yaşanan değişime en fazla etkisi olan ve yine bu değişimden en fazla etkilenen sektörlerden birisi de yazılım geliştirme sektörüdür. Yazılım geliştirme sürecinde, yazılımların ilk ortaya çıktığı yıllardan günümüze kadar sürekli olarak daha sistematik ve etkili metodoloji arayışları olmuştur. Bu metodolojilerden en fazla öne çıkanlar, çevik yöntemler içerisinde “Scrum”, geleneksel yöntemler içerisinde “Şelale” metodolojileridir. Günümüzün artan rekabet koşullarında bilişim sektöründe faaliyet gösteren işletmeler, yazılım geliştirme süreçlerinde daha verimli çalışmak için bu metodolojilerden kendilerine daha uygun olanları tercih etmektedir. Yazılım geliştirme süreçlerinde belirli bir yöntemi tercih eden işletmeler, zaman içerisinde verimliliğini arttırmak için farklı yöntem arayışlarına da girmektedir. Özellikle son yıllarda geleneksel yöntemi tercih eden işletmelerin, geleneksel yöntemleri terk edip çevik yöntemlerden Scrum’ı tercih etmeye başladığı sıklıkla gözükmemektedir.

Bu çalışmada, yazılım geliştirme süreçlerinde çevik yöntemlerden Scrum’ı tercih eden işletmelerin dönüşüm süreçlerini etkileyen faktörler ve bu faktörlerin işletmenin çevik dönüşümünün başarısına olan etkisinin araştırılması amaçlanmıştır. Uygulanan anket çalışması ile sektör çalışanlarına çalıştıkları işletmelerde Scrum uygulama ve dönüşümleri ile ilgili sorular sorulmuş, verilen cevaplara göre değişime etki eden faktörlerin ağırlıkları ve dönüşüm sürecine olan etkileri araştırılmıştır. Sonuç olarak ise, çevik dönüşüm sürecine giren işletmelerde, çevik dönüşüm sürecinde hangi faktörlere önem ve öncelik verilmesi gerektiği konusunda çeşitli çıkarımlarda bulunulmuştur.

Tez konusunun belirlenmesi, tez sürecinin planlanması ve nihai hale gelmesinde yorum ve katkılarıyla her türlü desteği esirgemeyen danışmanım Sayın Dr. Öğr. Üyesi M. FEVZİ ESEN’e teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖNSÖZ	iii
İÇİNDEKİLER	iv
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	vii
KISALTMALAR LİSTESİ	viii
ÖZET	ix
ABSTRACT	xi
BÖLÜM I	1
1.1. GİRİŞ	1
BÖLÜM II	5
2.1. YAZILIM GELİŞTİRME YÖNTEMLERİ	5
2.1.1. Geleneksel Yazılım Geliştirme Yöntemleri	6
2.1.2. Çevik (Agile) Yazılım Geliştirme Yöntemleri	11
BÖLÜM III	18
3.1. ÇEVİK YAZILIM GELİŞTİRME YÖNTEMLERİNDEN SCRUM	18
3.1.1. Scrum Rollerleri	20
3.1.2. Scrum Etkinlikleri	22
3.1.3. Scrum Eserleri	24
BÖLÜM IV	27
4.1. GELENEKSEL YÖNTEMDEN ÇEVİK YÖNTEME GEÇİŞ	27
4.1.1. Scrum'ın Uygulanabilme Başarısı	28
4.1.2. Scrum Dönüşümünün Başarısını Etkileyen Faktörler	30
BÖLÜM V	37
5.1. ARAŞTIRMA YÖNTEMİ	37
5.1.1. Anket İçeriği ve Anket Formunun Oluşturulması	37
5.1.2. Örneklem Seçimi	38

5.1.3. Anket Sonuçları	38
BÖLÜM VI	55
6.1. SONUÇ VE ÖNERİLER	55
KAYNAKÇA	59
ÖZGEÇMİŞ	63
EKLER	65

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1. SDLC Döngüsü (Seker, 2015:19).....	5
Şekil 2. Şelale Modeli (Royce, 1970)	8
Şekil 3. Sarmal Model Yaklaşımı (Wasson, 2005)	9
Şekil 4. V-Şekli Model (Fowler, 2015).....	11
Şekil 5. Kanban Süreci (Karuna, 2015)	14
Şekil 6. XP aşamaları (Pressman, 2005: 99).....	15
Şekil 7. Test Güdümlü Proje Geliştirme Süreci (Dönmez, 2009: 30).....	17
Şekil 8. Scrum Çevresi (Schwaber ve Sutherland, 2013)	18
Şekil 9. Scrum'ın Aşamaları (Kääriäinen ve Välimäki, 2008).....	20
Şekil 10. Çevik Yöntemin Tercih Edilme Dağılımı.....	41
Şekil 11. Çevik Yöntem Kullanım Dağılımı.....	42
Şekil 12. Geleneksel Yöntem Kullanım Dağılımı.....	43

TABLÖLAR LİSTESİ

Sayfa

Tablo 1. Başlıca Çevik Yöntemlerin Uygulanma Oranları (Gencer ve Kayacan, 2017).	12
Tablo 2. Katılımcıların Medeni Durum ve Eğitim Durumuna Göre Dağılımı	38
Tablo 3. Katılımcıların Deneyim Süresi Göre Dağılımı	39
Tablo 4. Katılımcıların Mesleklerine Göre Dağılımı	40
Tablo 5. İşletmelerin Bilgi İşlem Departmanı Çalışan Sayısı Dağılımı.....	40
Tablo 6. Katılımcıların Çalıştığı Sektör Dağılımı.....	41
Tablo 7. Çevik Yöntem Kullanım Süresi Dağılımı.....	43
Tablo 8. İşletmede Çevik Yöntem Kullanım Yaygınlığı Dağılımı	44
Tablo 9. Scrum Kullanmayan Çalışanların Scrum Hakkında Görüşleri	45
Tablo 10. Scrum dönüşümü - İnsan Faktörleri.....	46
Tablo 11. Scrum dönüşümü - Süreç ve Uygulama Faktörleri.....	48
Tablo 12. Scrum dönüşümü - Organizasyonel faktörler	49
Tablo 13. Sektörlere Göre Çevik Yöntem Tercihi	51
Tablo 14. Sektöre Göre Çevik Yöntem Kullanım Süresi Dağılımı	51
Tablo 15. Çevik Yöntem Kullanımının Sektöre Göre Sayıları	52
Tablo 16. Çalışan Sayısına Göre Çevik Yöntem Tercihi	53
Tablo 17. Hipotez Testleri.....	53

KISALTMALAR LİSTESİ

FDD	: Feature Driven Development
XP	: Extreme Programming
SDLC	: Software Development Life Cycle
PMI	: Project Management Institute

ÖZET

YAZILIM GELİŞTİRME SÜREÇLERİNDE SCRUM’U TERCİH EDEN İŞLETMELERİN ÇEVİK DÖNÜŞÜMÜNE ETKİ EDEN FAKTÖRLERİN İNCELENMESİ

Erhan Özlü

İSTANBUL MEDENİYET ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
İŞLETME ANABİLİM DALI İŞLETME PROGRAMI

Yüksek Lisans Tezi

Tez Danışmanı: Dr. Öğr. Üyesi M. FEVZİ ESEN

Günümüzde teknolojik gelişmelerle birlikte, müşterilerin ihtiyaç ve beklentilerinde hızlı bir değişim yaşanmaktadır. Bu değişimlere ve gelişmelere ayak uyduramayan işletmeler zamanla rakipleri ile rekabet edemez duruma gelmekte ve faaliyetlerini sonlandırmaktadır. Bu durum yazılım geliştirme yapan işletmeler için de geçerlidir. Yazılım geliştirme yapan işletmelerin yoğun rekabet ortamında hayatta kalabilmeleri için bilişim sektöründeki gelişmelere ayak uydurmaları ve değişen müşteri ihtiyaçlarına hızlı yanıt verebilmeleri gerekmektedir. Yazılım sektöründe projeler bir yıl veya bir yılı aşkın bir sürede geliştirilebilmektedir. Söz konusu projeler geleneksel yöntemlerle geliştirildiğinde, proje süresince müşteri ihtiyaç ve çevre koşullarında olabilecek değişimlerden dolayı proje sonunda ortaya çıkan ürünlerin, müşteri ihtiyacını karşılayacak güncellikte olmaması ile sonuçlanabilmektedir.

Müşterilerin, yazılım geliştirme yapan işletmelerden beklentileri; değişen ihtiyaçlara hızlı uyum sağlamaları ve proje teslimatlarını daha kısa sürede gerçekleştirmeleridir. Bu

sebepten, deęişen müşteri ihtiyalarına daha hızlı yanıt vermek ve müşteri memnuniyetini sağlamak amacıyla evik yazılım yöntemleri, yazılım sektöründeki işletmelerce uygulanmaya başlanmış olup, gün getike daha fazla işletme tarafından tercih edilir hale gelmiştir.

evik yazılımda en ok kullanılan yöntemlerden biri olan Scrum, Türkiye’de ve dünyada birçok işletme tarafından geniş bir uygulama alanına sahiptir. Buna rağmen, Scrum’ın her zaman başarılı bir şekilde uygulanamadığı ve organizasyonların Scrum’a geiş sürecinde bir takım problemler yaşadığı durumlarla sıklıkla karşılaşmaktadır. Bunun sonucu olarak, Scrum yönteminin organizasyona getireceğı fayda ve geleneksel yöntemlere göre üstünlüğü sık sık tartışılmaktadır.

Bu tez alışmasında, Türkiye’de yazılım projelerinde evik yöntemlerin kullanımı ve söz konusu yöntemlerden biri olan Scrum uygulamalarının ne derecede gerçekleştirildiğı araştırılmıştır. Bu doğrultuda, Türkiye’de eşitli sektörlerde faaliyet gösteren işletmelerin yazılım geliştirme departmanlarında farklı pozisyonlarda alışanların, alıştıkları işletmelerdeki evik yöntem kullanımları incelenmiştir. Bununla birlikte, söz konusu alışanların Scrum ve alıştıkları işletmelerde Scrum’ın uygulanma derecesi hakkındaki düşünceleri irdelenmiştir İşletmelerin evik dönüşümdeki başarısı evik dönüşüm ve Scrum konularında uzman olan evik koların görüşleri doğrultusunda literatürde önemli görülen eşitli faktörlerle birlikte değerlendirilmiştir. Bu araştırmada, evik dönüşüm sürecine organizasyonun tamamının dahil edilmesi ve gerektiğinde deęişimin organizasyon kültürüne yayılması, dönüşüm için organizasyon içerisinde gerekli bilginin, deęişim ortamının ve üst yönetimin desteęinin sağlanmasının, evik dönüşüm sürecinin başarılı olarak gerçekleştirilmesinde önemli olarak görüldüğü sonucuna ulaşılmıştır.

Anahtar Kelimeler: Yazılım Geliştirme, evik Süreler, Scrum, Yazılım Proje Yönetimi, evik Dönüşüm

ABSTRACT

FACTORS THAT AFFECTING THE AGILE TRANSFORMATION OF THE ORGANIZATIONS WHICH PREFER SCRUM IN THEIR SOFTWARE DEVELOPMENT PROCESSES

Erhan Özlü

INSTITUTE OF SOCIAL SCIENCES BUSINESS ADMINISTRATION PROGRAM

Master Thesis

Thesis Supervisor: Dr. Öğr. Üyesi M. FEVZİ ESEN

Nowadays, there is a rapid change in customer needs and expectations with the latest technological developments. Companies that cannot keep up with these changes and developments have problems to compete with their rivals in time. This fact also seen for companies engaged in software development. Software development companies need to keep up with developments in the technology and respond quickly to changing customer needs in order to survive in an intense competitive environment. However, in the software sector, projects be developed over a period of one year or more. When companies develop these projects with traditional methodologies, customer needs and environmental conditions frequently change. In this case, the project may result in updated products at the end of the project and these products do not have the ability to meet the customer needs.

Customers expect from companies adapt to the changing needs and deliver project deliveries in a shorter time. For this reason, agile software methodologies have been preferred by many companies in the software sector in order to respond faster to changing customer needs and to ensure customer satisfaction.

Scrum is one of the most widely used method of agile software and it has a wide range of applications by many businesses in Turkey and the world. Nevertheless, unsuccessfully Scrum implementation and companies face some problems in the transition to Scrum are often encountered. As a result, it is the advantage of the Scrum and its superiority over traditional methods stated frequently.

In this study, the use of agile methodologies and the success of Scrum transformation in software projects in Turkey is researched. In this regard, the use of agile methodologies and Scrum usage was examined in different sectors. In additioin, the opinions of the employees in question about Scrum and the degree of implementation of Scrum in the companies they work in were examined. Moreover, the success of organizational agile transformation is evaluated with various factors that are considered important in the literature in accordance with expert opinions.

As a result of this study, it has been concluded that the agile transformation process has been realized more successfully in organizations that include the entire organization in the process of agile transformation, spread the change to the whole organization, change the culture of the organization and provide the support of the management.

Key Words: Software Development, Agile Processes, Scrum, Software Project Management, Agile Transformation

BÖLÜM I

1.1. GİRİŞ

Günümüzde yazılım geliştirme projelerinde farklı sektör ve uygulama alanlarında farklı yöntemler tercih edilmektedir. Söz konusu yöntemler aynı işletme içerisinde değişen proje ihtiyaçlarına göre farklılaşabildiği gibi; organizasyon kültürü, rekabet ortamı, geliştirilen ürünün karakteristik özellikleri, müşterilerin beklentileri, çevre koşulları gibi faktörlerden de etkilenmektedir. Bu sebeple, işletmelerin değişen ihtiyaçlarına ve çevre koşullarına göre zaman içerisinde yazılım geliştirmeye yönelik bir çok yöntem ortaya çıkmıştır. Bu yöntemler, özelliklerine ve kullanım alanlarına göre incelendiğinde temel olarak geleneksel yöntemler ve çevik yöntemler olarak ikiye ayrılmaktadır.

İnşaat projelerinin planlanma süreçlerinden ortaya çıkan geleneksel yazılım geliştirme yöntemleri, yazılım projelerinin ilk olarak geliştirilmeye başlandığı 1970’li yıllara kadar uzanmaktadır. Çevik yöntemlerin kullanımı ise 2000’li yıllarda geleneksel yöntemlere alternatifler aranmaya başlanmasıyla hız kazanmıştır. Uzun yıllar başarılı bir şekilde kullanılan ve günümüzde de kullanılmaya devam edilen geleneksel yazılım geliştirme yöntemlerine alternatifler aranmasının sebepleri, yazılım projelerinin inşaat projelerine göre daha değişken bir yapıda olması, içerdiği risk ve belirsizliklerin diğer projelere kıyasla daha fazla olmasıdır (Baytam ve Kalıpsız, 2011:18).

Müşteri ihtiyaçlarının net bir şekilde önceden belirlendiği ve teknolojik değişimin hemen hemen hiç bulunmadığı ortamlarda geliştirilen büyük ve karmaşık projelerde, geleneksel yöntemler uzun süredir başarılı bir şekilde kullanılmaktadır (Baytam ve Kalıpsız, 2011:6). Çevre koşulları ve müşteri ihtiyaçlarının hızla değiştiği günümüzde ise, yazılım geliştirme yapan işletmelerin söz konusu ihtiyaçlara daha hızlı yanıt vermesi beklenmekte olup; yazılım geliştirme süreçlerinde çevik yöntemlerin kullanılması gerekli hale gelmeye başlamıştır (Çetin, 2016:12). Türkiye’de yazılım geliştirmede çevik yöntemlerden biri olan Scrum’ın uygulama alanı gittikçe artmakta olup, birçok işletmede proje geliştirme süreçlerinde söz konusu yöntem tercih edilmektedir. Bunlara örnek olarak Türkiye’de; Microsoft, Yahoo, Google, Electronic Arts, High Moon Studios, Lockheed Martin, Philips, Siemens, Nokia, Capital One, BBC, Intuit, Intuit, Nielsen Media, First American

Real Estate, BMC Software, Ipswitch, John Deere, Lexis Nexis. Sabre, Salesforce.com, Time Warner, Turner Broadcasting, Oce, Türk Telekom, Turkcell, Avea ve TAI firmalarının uygulamaları gösterilebilir (Erdil, 2013:25).

Yazılım geliştirme süreçlerinde Scrum'ı tercih eden işletmelerden bazılarının çevik dönüşümü başarıyla gerçekleştirmesine rağmen, bazı organizasyonların dönüşüm aşamasında başarılı olamadığı görülmektedir. Çevik dönüşüm sürecini başarıyla gerçekleştiren organizasyonların, dönüşüm sürecini başarılı bir şekilde gerçekleştiremeyen veya beklediği başarıyı yeterince sağlayamayan organizasyonlara göre nispeten daha az olduğu belirtilmektedir (Mandanis, 2015).

2015 yılında yayınlanan çevik organizasyonlar araştırma raporuna göre çevik dönüşüm süreçlerinde organizasyonların dönüşüm süreci sonucunda bekledikleri başarıyı sağlayanların oranının tüm organizasyonlar içerisinde sadece %27 olduğu tespit edilmiştir (McKinsey, 2015). Tüm bunlar organizasyonel değişim sürecinin kolay olmadığını, çevik dönüşümü gerçekleştirmek isteyen işletmelerin dönüşüm süreci içerisinde birçok zorlukla karşılaştığını ve birçok işletmenin söz konusu zorlukları aşamadığını göstermektedir.

İşletmeler için dönüşüm sürecinden en büyük beklenti proje başarı yüzdesinin artışı iken; yazılım projelerinde ise hedeflenen amacın veya ürünün gerçekleştirilmesi tek başına yeterli bir proje başarı göstergesi değildir. Proje Yönetimi Enstitüsü'ne (PMI) göre başarılı projeler uygun maliyette ve zamanda tamamlanan, istenilen düzeyde performans sağlayabilen ve kalite olarak müşteri tatmininin sağlandığı projeler olarak nitelendirilmektedir (Jeffery K. ve Dennis P. 1988).

Bir işletmede Scrum dönüşümün başarılı olarak gerçekleştirilmesi, proje başarı oranındaki artış, üretilen yazılımların ve geliştirilen ürünlerin daha kaliteli olması, değişen müşteri ihtiyaçlarına daha hızlı yanıt verilebilmesine bağlı olarak müşteri memnuniyetinde artış ve geliştirme maliyetlerinde iyileşmelerin gözlemlenmesiyle ilişkilidir. Scrum'a geçiş sonrası belirtilen kriterlere bağlı olarak proje başarı yüzdesinde bir artış gözlemleniyorsa, organizasyonda Scrum dönüşümünün başarılı bir şekilde gerçekleşiyor olduğu söylenebilir. Bu kriterlerde beklenen başarıyı sağlayamayan veya bir önceki durumun gerisinde kalan işletmeler için dönüşüm sürecinin sağlıklı olarak gerçekleşmediği ifade

edilebilir. Dönüşüm aşamasında organizasyonel veya çevresel boyutta sorun yaşayan bu tür işletmeler belirli bir süre sonra Scrum'ı terk etme eğilimi içerisinde girmektedir.

Bir organizasyonda Scrum uygulamaya başlandığı andan itibaren çevik dönüşüm sürecinin başladığı ifade edilebilir. Çevik proje yöntemi metodolojilerinden biri olarak Scrum, karmaşık yazılım süreçlerinin basitleştirilerek yönetilmesidir. Bunun için de karmaşık bütünü, basit parçalara ayıran ve birbirini tekrarlayan bir yöntem izlemektedir. Düzenli geri bildirim ve planlamalar ile başlangıçta büyük ve karmaşık olan hedefe daha kolay ulaşılmasını sağlamaktadır.

Türkiye'de birçok işletmenin çevik yöntemleri tercih etmesiyle ilişkili olarak, Scrum'a olan ilgi gittikçe artmakta, özellikle iş dünyasında bununla ilgili farklı çalışmalar ve etkinlikler yürütülmektedir. Buna karşın, İşletmelerin Scrum yöntemine geçiş sürecini nasıl yönettiğini ve uygulama sonrası başarılarının hangi yönde gerçekleştiğini inceleyen akademik veya kurumsal bir çalışma bulunmamaktadır. Bu tez çalışmasında, işletmelerin Scrum dönüşümünü başarılı bir şekilde gerçekleştirme derecesi yapılan anket çalışmasıyla incelenmiş, işletmelerin çevik dönüşüm süreçlerini etkileyen faktörler araştırılmıştır.

Literatürde, çevik dönüşüm sürecini etkileyen faktörler incelendiğinde, organizasyonların dönüşüm sürecini etkileyen birçok faktör bulunduğu görülmektedir. Bu çalışmada söz konusu faktörlerden; çalışanların Scrum farkındalığı ve Scrum hakkındaki yaklaşımlarını ölçen insan faktörü, yöntemin uygulanma başarısını ölçen, süreç ve uygulamalara ilişkin uygulama faktörleri ve yönetim desteği ile organizasyonel değişimleri ölçen organizasyonel faktörler ele alınmıştır. İnsan faktörleri başlığında, çevik dönüşüm sürecinde bireylerin rol, sorumluluk ve tutumlarına ilişkin bileşenlere temas edilmiş olup; işletmenin çevik dönüşümünü doğrudan etkilediği düşünülen çalışan yetkinlikleri, sorumlulukları ve Scrum farkındalıkları değerlendirilmiştir. Uygulama faktörleri başlığında ise takımların eğitim ve gelişimleri, sorumlulukları ve danışman desteği alabilmeleri üzerinde durulmuştur. Organizasyonel faktörler olarak ise, çevik yönetime üst yönetimin desteği, çevik değişimin tüm organizasyona yayılımı ve organizasyonel kültürde yarattığı değişim ele alınmıştır. Çalışmada ayrıca, Scrum'ı tercih eden

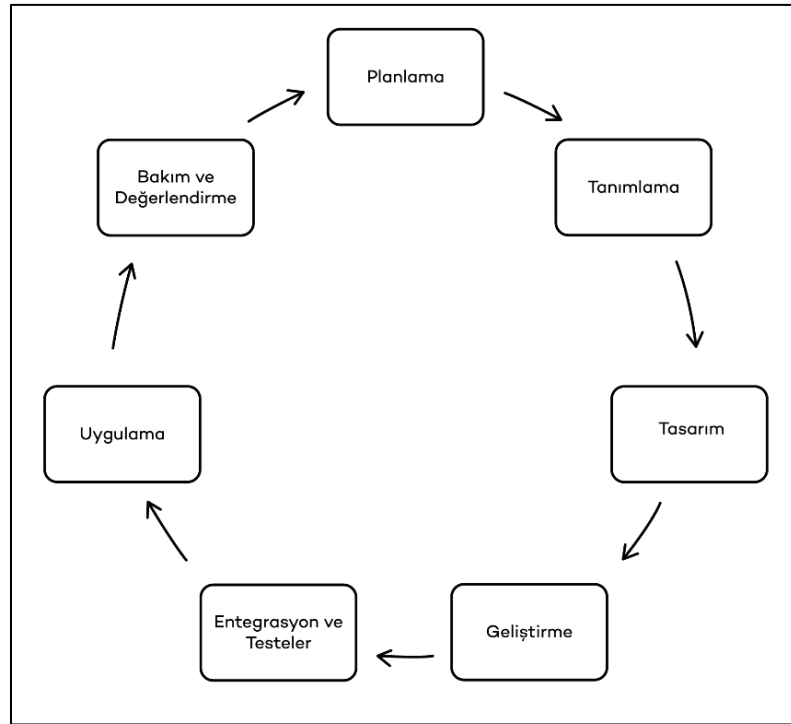
iřletmelerin evik dnřm srecinde bu faktrleri nasıl ynettięi ve bunun dnřm bařarısına olan etkisi anket alıřması ile incelenmiřtir.

1 Ocak - 15 řubat 2018 tarihleri arasında sektr alıřanları ile gerekleřtirilen anket alıřmasında, sonular temel istatistiksel tablolar ve testlerle elde edilmiřtir. Arařtırma sonucunda iřletmelerin yaygın olarak Scrum'ı tercih ettięi sonucuna ulařılmasına raęmen, iřletmelerin bu dnřmden bekledikleri verim, kalite, maliyet iyileřtirmelerini kısmen saęlayabildikleri tespit edilmiřtir. alıřmada Scrum yntemini tercih eden iřletmelerin, yazılım geliřtirme srelerinde verimlilik artıřı ve yazılım geliřtirme maliyetlerinde azalma olmadıęı sonuları ıkmıřtır. Bunun yanı sıra, bařarılı dnřm srecinin bir gereklilięi olan organizasyon kltrndeki deęiřimin yeterli seviyelerde gerekleřmedięi sonucuna ulařılmıř olup; bu baęlamda organizasyonların Scrum dnřmn bařarılı olarak gerekleřtirme noktasında sıkıntılar yařadıęı dřnlmektedir. Buna raęmen, dnřm srecine giren iřletmelerin, deęiřen mřteri ihtiyalarına daha hızlı yanıt verebildięi, mřteri memnuniyetinde artıř saęladıęı ve retilen yazılım kalitesini kısmen arttırdıęı sonuları ıkmıřtır.

BÖLÜM II

2.1. YAZILIM GELİŞTİRME YÖNTEMLERİ

Yazılım ürünlerinin üretim aşaması ve kullanım süreci boyunca geçirdiği tüm aşamalar "Yazılım Geliştirme Yaşam Döngüsü" – SDLC olarak tanımlanmaktadır (Şeker, 2015:18). Yazılım geliştirme süreçlerinde söz konusu döngü sırasıyla; planlama, tanımlama, tasarım, geliştirme, entegrasyon test ve uygulama aşamalarından oluşmaktadır. Bu adımların nasıl gerçekleştirileceğine yönelik ise işletme ihtiyaçlarına ve yazılım geliştirme koşullarına göre çeşitli model ve yöntemler geliştirilmiştir. Söz konusu yöntemler temel olarak geleneksel ve çevik yöntemler olarak ikiye ayrılmaktadır.



Şekil 1. SDLC Döngüsü (Seker, 2015:19)

Şekil-1’de SDLC döngüsüne ait aşamalar verilmektedir. SDLC döngüsü planlama ile başlayıp birbirini takip eden ve sonunda döngüye tekrar başlayan aşamalardan oluşmaktadır. Planlama aşaması müşteri ihtiyaçlarının belirlendiği, gereksinimlerin toplantığı, risklerin belirlendiği ve ürün fizibilite çalışmasının yapıldığı en önemli ve en temel aşamadır. Planlama yapıldıktan sonraki adımda ise ürün gereksinimlerinin açıkça

tanımlandığı ve belgelendiği tanımlama aşaması gelmektedir. Tanımlama aşamasında ürün gereksinimleri belgelenerek müşterilerden onay alınmaktadır. Tasarım aşamasında ise ürün mimarisi için birden fazla tasarım önerilmekte ve bunlar içinden geliştirilecek ürün için en iyi mimari tasarım seçilmektedir. Tasarım aşaması tamamlandıktan sonra geliştirme aşaması başlamakta ve bunla birlikte ürün oluşturulmaya başlanmaktadır. Entegrasyon ve test aşaması, geliştirilen ürünün doğrulandığı aşamadır ve bu aşamada geliştirme sonunda çıkan ürün ile tanımlama aşamasında tariflenen ürünün aynı olduğu teyit edilmektedir. Ürünün test ortamında doğrulanması sağlandıktan sonra ürün, kullanım için gerçek ortama sürülmektedir. Dağıtım, ürünün özelliğine ve işletmenin stratejisine göre birden veya kademeli yapılabilir. Son aşamada ise canlıya gönderilen ürün gerçek ortamda takip edilerek çıkan problemler düzeltilmektedir.

2.1.1. Geleneksel Yazılım Geliştirme Yöntemleri

Geleneksel yazılım geliştirme yöntemleri; müşteri gereksiniminin belirlenmesi, analiz dökümanının oluşturulması, tasarım çalışmalarının yapılması, kodlamanın yapılması, testlerin gerçekleştirilmesi, teslimat ve bakım çalışmalarının yapılması adımlarını içermektedir. Bu adımların en önemli özelliği, bir sürecin tamamlanmadan bir diğerine başlanamamasıdır. Dolayısıyla bu süreçler birbiri ile bağımlı olarak çalışmakla birlikte, bir süreç grubunun çıktısı kendisinden sonraki süreç grubunun girdisi olduğu için, süreç grupları arasında bir bağımlılık söz konusudur (Pressman, 2005:410).

Geleneksel yazılım süreçlerinde yazılım gereksinimlerinin tamamı başlangıçta müşteriden alınmakta ve müşteri ihtiyaçlarının tamamı net olarak ifade edilmeden bir sonraki adıma geçilememektedir. Bu yüzden geleneksel yöntemler müşteri ihtiyaçlarının tamamının proje başında detaylı olarak belirlenebildiği projeler için uygun olmakla birlikte, değişen müşteri ihtiyaçlarının bulunduğu ve ihtiyaçların proje başında net olarak belirlenemediği ortamlar ve projeler için uygun olmamaktadır (Pressman, 2005:99).

Geleneksel yazılım geliştirme yöntemlerinin ilk ortaya çıkışı, inşaat sektöründe kullanılan proje yönetim planına dayanmaktadır (Baytam ve Kalıpsız, 2011:18). Söz konusu yöntemler ilk olarak 1956 yılında Benington tarafından Amerikan Donanması'nın düzenlediği bir matematik sempozyumunda sunulmuştur (NMCAP, 1956). Bu tarihe

kadar yazılım projelerinin herhangi bir y nteme baėlı kalınmaksızın geliřtirilmeye  alıřıldığı g zlemlenmektedir. Uzun yıllar bařarılı bir řekilde kullanılan ve g n m zde hala kullanılmaya devam eden geleneksel yazılım geliřtirme y ntemlerine alternatifler aranmasının sebebi, yazılım projelerinin inřaat projelerine g re daha deėiřken bir yapıda olması ve belirsizliklerin yazılım projelerinde daha fazla bulunması olarak a ıklanmaktadır (DeCarlo, 2015:55).

Geleneksel yazılım y ntemlerinden en  ok tercih edilenlerinden bazıları; řelale (Waterfall) y ntemi, Sarmal (Spiral) model ve V řekli (V-Shaped) modeli olarak sıralanmaktadır. Bu modellerin diėer geleneksel yazılım y ntemlerine g re daha yaygın kullanıldığı belirtilmektedir (Pressman, 2005: 38).

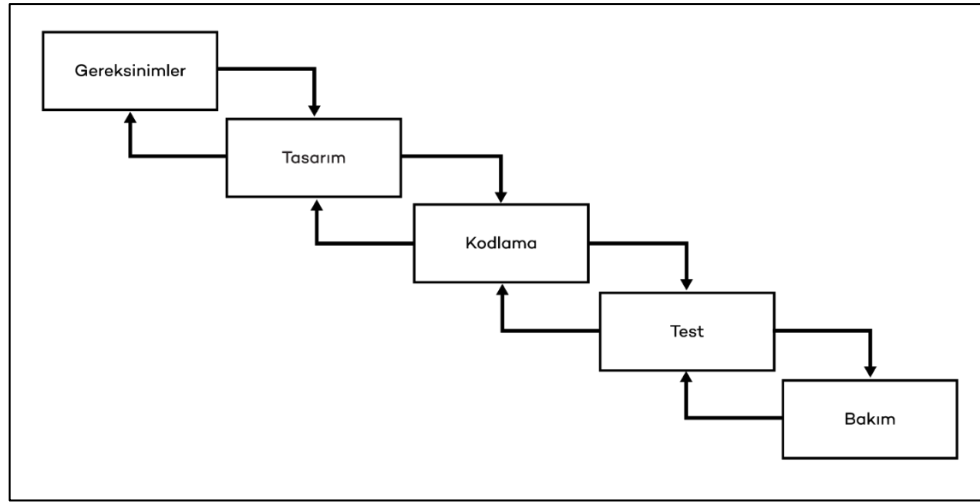
2.1.1.1. řelale (WaterFall) Y ntemi

Geleneksel y ntemlerden en  ok tercih edilen řelale y ntemi literat re Royce (1970) ve Benington (1983) tarafından tanıtılmıştır. řelale y nteminde temel olarak birbirini takip eden beř ařama bulunmaktadır. řekil 2’de g sterildiėi  zere, bu ařamalar sırasıyla; gereksinimlerin belirlenmesi, analiz - tasarım d k manlarının hazırlanması, kodlamanın yapılması, testlerin ger ekleřtirilmesi ve bakım adımlarından oluřmaktadır.

řelale y nteminin ilk ařamasında m řteri gereksinimleri toplanmakta ve  r n gereksinimleri ortaya  ıkartılarak d k mante edilmektedir. Sonraki s re ler  r n gereksinim d k manında yazılanlara g re yapılacaėı i in, d k manın nihai formu m řterilerden onay alınmaktadır. Tasarım ařamasında;  r n n karakteristik  zellikleri,  r n gereksinimleri, yazılım geliřtiren firmanın kořulları ve teknoloji gibi kriterler dikkate alınarak mimari tasarım belirlenmektedir. Tasarım ařaması tamamlandıktan sonra geliřtirme ařaması bařlamakta; daha  nceden d k mante edilmiř gereksinimler belirlenen mimari tasarım i erisinde programlanmaya bařlanmaktadır. Programlamalar tamamlandı  a  ıkan program par acıkları test edilerek doėru  alıřıp  alıřmadıėı kontrol edilmektedir. Test ařaması tamamlandıktan sonra  r n canlıya alınmakta ve burada bakımı devam etmektedir. S z konusu adımlardan biri tamamlanmadan diėerine ge ilememekte ve her s recin sonucunda ilgili adımın detaylı dok mantasyonu yapılmaktadır (řeker, 2015: 24).

Şelale yöntemi, dökümantasyon ağırlıklı olması ve süreçlerinin birbirinden bağımsız olması açısından kolay kontrol edilebilme ve daha iyi yönetilebilme gibi pozitif yönlere sahiptir. Bu yüzden büyük ve karmaşık projelerde, müşteri ihtiyaçlarının net bir şekilde belirli olduğu durumlar ile değişimin hemen hemen hiç bulunmadığı statik ortamlarda geliştirilen yazılım projelerinde söz konusu yöntem sıklıkla tercih edilmektedir (Baytam ve Kalıpsız, 2011: 17).

Günümüzde, yazılım projelerinin birçoğunda süreçler hızlı iletişim ve sürdürülebilir gelişmeye odaklanmaktadır. Müşteri ihtiyaçları, hızlı değişen çevre koşullarından dolayı sürekli değişmekte, proje sonucunda ortaya çıkan ürünler kısa bir süre sonra güncelliğini kaybetmektedir. Bunun yanı sıra, müşteriler proje başında ne istediklerinden tam olarak emin olamamakta veya isteklerini eksik tarifleyebilmektedir. Ayrıca proje geliştirme sürecinde çıktılar gözle görülmeye başlandıkça, müşteri isteklerinde sıklıkla değişimler olmaktadır. Bu durum, geleneksel yöntemlerden biri olan şelale yönteminin, yazılım projelerin başarısındaki etkisini tartışılır hale getirmekle birlikte, organizasyonların yeni arayışlar içine girmesine sebep olmaktadır (Adetokunbo ve diğerleri, 2013: 432).

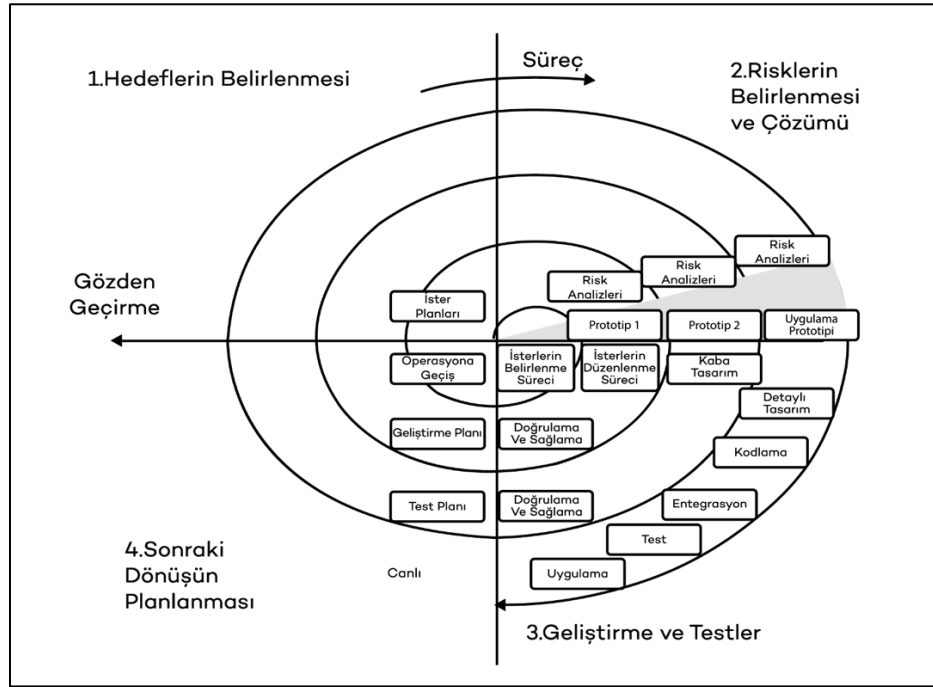


Şekil 2. Şelale Modeli (Royce, 1970)

2.1.1.2. Sarmal (Spiral) Model

Sarmal model Boehm (1986) tarafından oluşturulup literature kazandırılmış bir model olup; şelale modelindeki düz akıştan farklı olarak yazılım geliştirmenin kademeli olarak yapıldığı bir yöntemdir. Sarmal modelin temeli, yazılım geliştirmeyi tekrarlayarak niceliği

ve etkinliđi arttırmaya dayanmaktadır. Tekrarlayarak arttırmada bir konu tekrar tekrar ele alınmakta ve her tekrarda konu daha derinlemesine ele alınmaktadır. Bilgiler önceki tekrarlarda öğrenilenlere dayalıdır ve bu bilgilerin genişlemesi ve derinleşmesi sağlanmaktadır. Spiral model, diğerk modellerden farklı olarak süreci oluşturan aşamalardan tekrarlı bir şekilde geçilmesini ve her geçişte projenin ilerleme kat etmesini hedeflemektedir (Boehm, 1986).



Şekil 3. Sarmal Model Yaklaşımı (Wasson, 2005)

Şekil 3’de gösterildiđi üzere, sarmal model temel olarak dört ana bölümden oluşmaktadır. İlk aşama olan planlamada, üretilecek ara ürün için işin planlanması ile gereksinimlerin, kısıtların ve alternatif çözümlerin belirlenmesi yapılmaktadır. Ayrıca bir önceki adımda üretilen ürün ile yeni oluşturulacak geliştirmelerin birleştirilmesi yapılmaktadır. İkinci aşama olan risk analizinde ise projeye ve ürüne ait risk faktörleri belirlenip risk planı oluşturulmakta; alternatifler değerlendirilerek risk analizi yapılmaktadır. Üretim aşamasında ise, planlanmış ara ürün geliştirilmekte; son aşama olan kullanıcı değerlendirmede ise kullanıcıların ara ürünü test etmesi ve ürün hakkına değerlendirme yapması sağlanmaktadır (Wasson, 2005).

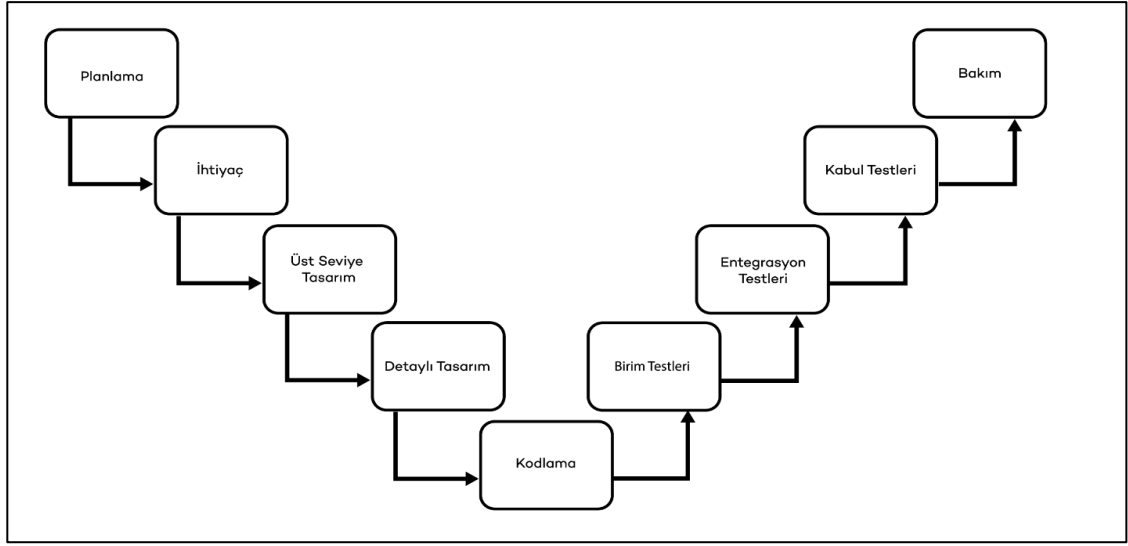
Sarmal modelin pratikte çok fazla tercih edilmemesinin nedeni, uygulama maliyetinin yüksek ve uygulamasının zor olmasıdır. Örneğin sarmal modelde, risk analizi oldukça özel uzmanlık gerektirmekte ve proje başarısı, büyük ölçüde risk analizi aşamasına bağlı olmaktadır. Ayrıca, sarmal modelde proje başarısında çok önemli yeri olan proje kilometre taşlarının tanımlanması zor olabilir. Bu yüzden sarmal model, küçük projelerde ve düşük riskli projelerde hemen hemen tercih edilmeyen bir yazılım geliştirme yöntemidir (Virginia Tech Courses, 2018).

2.1.1.3. V- Şeklinde (V- Shaped) Model

Bu modeldeki adımlar V şeklini oluşturduğu için “V-Şeklinde Model” adını almaktadır. Model temel olarak şelale yöntemine benzemekle birlikte, şelale yönteminden temel farkı, test odaklı bir model olmasıdır. Söz konusu model, testlere dokümantasyon safhasından itibaren başlanıldığı için sonraki süreçlerde karşılaşılabilecek hataların daha erken tespit edilmesini sağlamaktadır.

V - şeklinde modelde Şekil 4’de gösterildiği üzere sırasıyla; planlama, ihtiyaç belirleme, üst seviye tasarım ve detay tasarım adımları bulunmaktadır. Bu adımlar tamamlandıktan sonra kodlamaya geçilmektedir. Kodlamadan sonra, iş paketlerinde yukarı doğru çıkılarak, sırasıyla birim testleri yapılmakta; entegrasyon testleri gerçekleştirilmekte, kullanıcı kabul testleri gerçekleştirilerek müşteri kararı için ilgili müşterilere gidilmektedir. Bakım aşamasında ise müşteriler tarafından kullanılmaya başlanılan ürünün bakımı ve problem düzeltmeleri yapılmaktadır (Fowler, 2015).

Fowler (2015)’e göre V-Şekli Model çok katı ve az esnek bir modeldir. Bu nedenle ihtiyaçların sürekli değiştiği ortamlar için uygun değildir. Ayrıca bu yöntemin uygulandığı süreçlerde yazılım, uygulama aşamasında geliştirilmeye başlanmakta ve erken safhalarda çalışan yazılım veya prototip üretimi yapılmamaktadır. V-Şekli model yaklaşımını seçmek için yüksek müşteri güveni gereklidir ve proje başında hiçbir prototip üretilmediğinden müşteri beklentilerini karşılama konusunda çok yüksek risk bulunmaktadır.



Şekil 4. V-Şekli Model (Fowler, 2015)

2.1.2. Çevik (Agile) Yazılım Geliştirme Yöntemleri

Çevik yazılım geliştirme yöntemleri, 1950’den sonraki yıllarda verimliliğin artırılması için geliştirilen yalın yaklaşımların, 70’li yıllardan itibaren yazılım sektöründeki uygulama alanı olarak ortaya çıkmış olup, 90’lardan sonra kullanımı hız kazanmıştır (ACM Yazılım Çözümleri, 2017).

Yazılım projelerinin değişken bir yapıda olması ve çok fazla belirsizlik barındırması, geleneksel yöntemlerin hızlı değişen çevre koşulları ve müşteri ihtiyaçlarına tam olarak cevap veremediği durumlarda kullanılan çevik yazılım yöntemlerini ortaya çıkarmıştır. Söz konusu yöntemler geleneksel yöntemlerde ön planda olmayan bazı değerlerin, ön plana çıkarılarak daha başarılı ve kaliteli yazılımların elde edilebileceği fikriyle birlikte ortaya çıkmıştır (Baytam ve Kalıpsız, 2011:15). Çevik yazılım yöntemlerinin temel maddeleri olarak:

- Bireyler ve etkileşimi, süreç ve araca tercih etmek,
- Çalışan bir yazılımı, detaylı belgelendirmeye tercih etmek,
- Müşteri ile işbirliğini, sözleşmedeki kesin kurallara tercih etmek,
- Değişikliklere uyum sağlayabilmeyi, belirli bir plana tercih etmek,

şeklinde belirlenmiştir (Agile Manifesto, 2001). Devam eden çalışmalar neticesinde, çevik bildirim yeni özellikler kazanarak 12 temel prensip üzerinde şekillenmiştir (Agile Manifesto, 2001)

Schindler'e (2008) göre çevik yöntemler arasında en yaygın kullanılan yöntemler Scrum ve Scrum'ın hibritlerinden oluşan Scrum/Kanban karma, Scrum/XP karma yöntemleridir. Bunların haricinde farklı organizasyonlarda kanban, sınırsal programlama, test güdümlü geliştirme, özellik güdümlü geliştirme ve dinamik sistem geliştirme modeli gibi yöntemler de kullanılmaktadır. Söz konusu yöntemlerin kullanımı Scrum'a oranla oldukça azdır (Scrum Alliance, 2018).

Tablo 1'de VersiyonOne, Forresoter ve AgileTurkey tarafından ayrı olarak yapılmış çalışmalarda başlıca çevik yöntemlerin kullanım oranları verilmiştir. Scrum metodolojisinin Türkiye ve dünya genelinde açık ara en popüler metodoloji olduğu görülmekle birlikte, organizasyonların kendi ihtiyaçlarına göre Kanban ve XP yöntemlerini tek başına ve Scrum ile birlikte tercih ettiği durumlar gözlemlenmektedir.

Tablo 1. Başlıca Çevik Yöntemlerin Uygulanma Oranları (Gencer ve Kayacan, 2017)

Metodoloji	VersionOne	Forrester	AgileTurkey
Scrum	% 58	% 86	% 65
XP	% 1	% 29	% 7
Kanban	% 5	% 57	% 32
Scrum/XP Karma	% 10	-	% 8
Scrum/Kanban Karma	% 8	-	% 7
Çoklu Karma Uygulamalar	% 8	-	% 8

Ayrıca, çevik prensipleri üzerine oturmuş farklı çevik yazılım geliştirme yöntemleri bulunmasına rağmen bu yöntemlerden en çok tercih edileni ve kullanılanları aşağıdaki gibi belirtilmektedir (Gencer ve Kayacan, 2017):

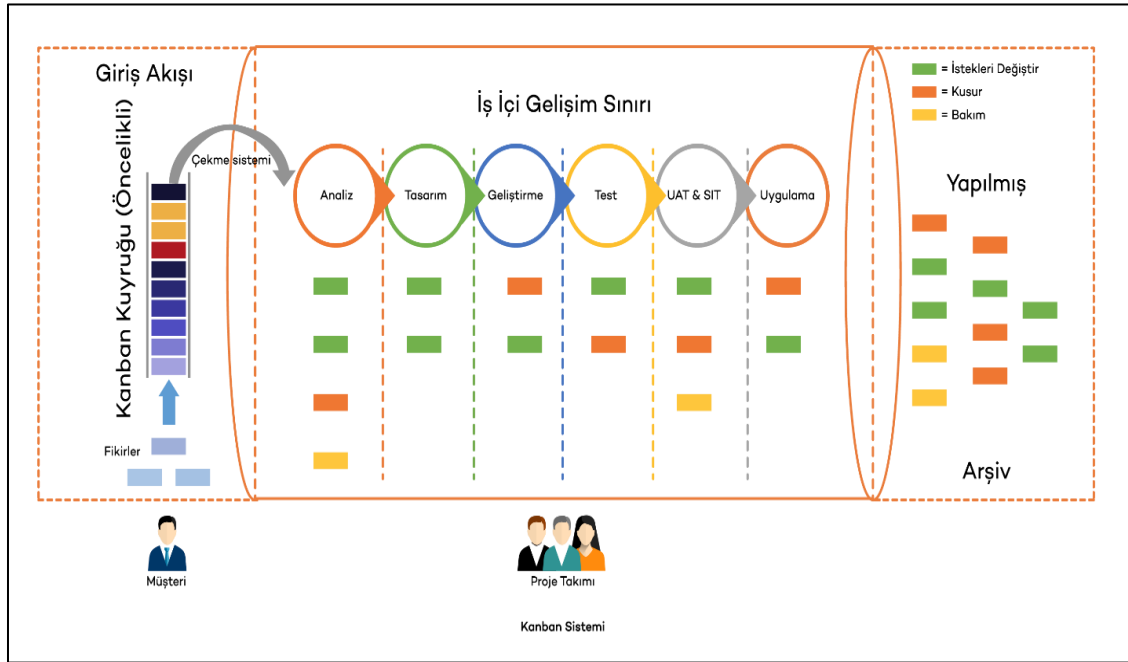
- Scrum
- Kanban
- Sınırsal Programlama (XP)
- Test GÜdümlü Geliştirme
- Dinamik Sistem Geliştirme Metodu
- Özellik GÜdümlü Geliştirme (Feature-Driven Programming)
- Çevik Birleştirilmiş Süreç (AGILE UnifiedProcess),
- Uyumlu Yazılım Geliştirme (Adaptive Software Development),
- Akılcı Birleştirilmiş Süreç (RationalUnifiedProcess).

2.1.2.1. Kanban

Kanban ilk olarak 1950'li yılların başında TaiichiOhno tarafından geliştirilmiş olup, Toyota'nın üretim sisteminde, tedarik zinciri yöntemini kontrol altında tutmak için kullanılmaya başlanmıştır (Ahmad ve Ovio, 2013: 9). Kanban sisteminin yazılım geliştirme süreçlerindeki uygulaması ise ilk kez 2004 yılında David J. Anderson tarafından yapılmış olup, aynı yıl Microsoft'un yazılım geliştirme süreçlerinde kullanılmasıyla birlikte, birçok firma ve organizasyon yazılım geliştirme süreçlerinde Kanban'ı tercih etmeye başlamıştır (Digite, 2018).

Kanban, işi görselleştirmek, akışı geliştirmek, kayıpları azaltmak ve müşteri değerini en üst düzeye çıkarmak amacıyla çekme sistemiyle çalışan bir sistem olarak tanımlanmaktadır (Ahmad ve Ovio, 2013: 13). Söz konusu sistem, yazılım geliştirme sürecinde işlerin devamlılığını ve sürekli teslimatını sağlamak için kullanılmaktadır. Sistem, anlık takip edilen iş sayısını kısıtlayarak birikme ve tıkanmaları engellemeye çalışmakta olup, değişen koşullara daha hızlı cevap verilmesini ve daha sık yazılım teslimatları yaparak müşteri memnuniyetinin en üst seviyeye çıkartılmasını amaçlamaktadır (Ahmad ve Ovio, 2013:13).

Şekil 5’de Kanban sisteminin adımları verilmiştir. Giriş akışı kısmında, yapılacak işlerin önceliklendirilmiş şekilde listelendiği bir iş listesi bulunmaktadır. Bu iş listesi, müşteri tarafından sürekli yenilerinin eklendiği ve önceliklendirilmelerin güncellendiği bir listedir. Yazılım geliştirme ekibi, bu listeden en öncelikli işleri alarak geliştirmeye başlamakta ve tamamlanan işleri “bitti” kısmına alarak müşteriye teslim etmektedir. Kanban uygulamalarında karşımıza çıkan en önemli dezavantajların başında, kanban tahtasında zamanında güncelleme yapılmadığında geliştirme ekibinin yanlış geliştirme sürecine girdiği durumlar yaşanabilmesi gelmektedir. Bu durum, öncelikli olmayan işlerin önceliklendirilmesi gibi durumlara sebep olmaktadır. Bunun yanı sıra, Kanban’ın uygulama problemlerinden bir diğeri ise, işlerin ne zaman başlanacağına ve biteceğine yönelik zamanlama planının bulunmamasıdır. Bu da müşterinin ürününü ne zaman teslim alacağını bilmemesi gibi bir duruma sebep olmaktadır.

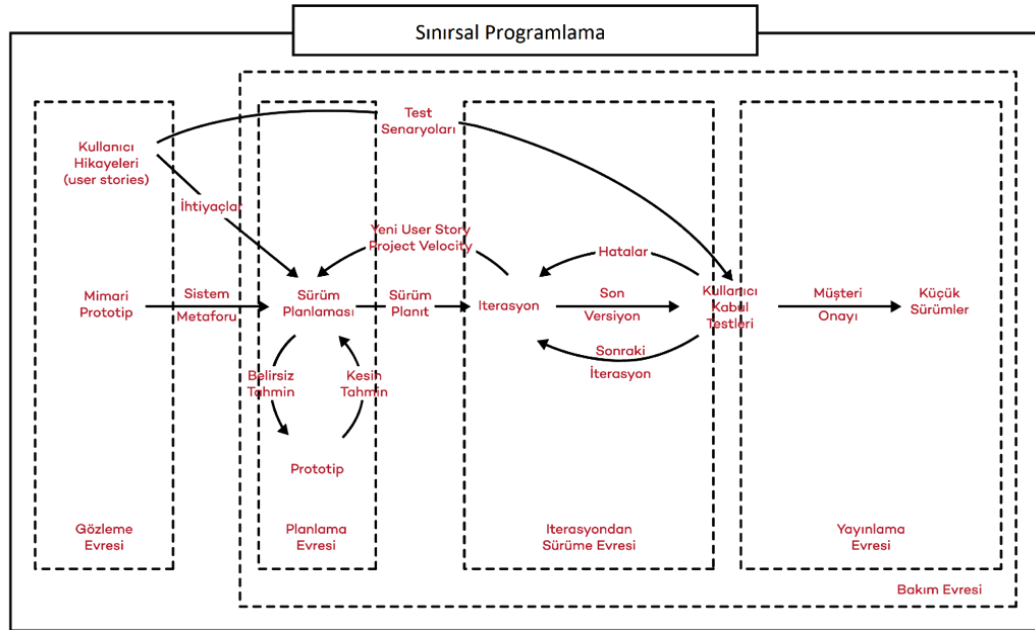


Şekil 5. Kanban Süreci (Karuna, 2015)

2.1.2.2. Sınırsal Programlama (XP)

İşbirliği çerçevesinde, hızlı ve erken yazılım geliştirmeyi ve kaliteli ürünler geliştirilmesini hedefleyen sınırsal programlama, yazılım sektöründe sıklıkla kullanılan çevik yöntemlerden biridir. Yöntem ilk olarak 1999 yılında Beck tarafında geliştirilmiş olup, 2004 yılında revize edilmiştir (Abdullah ve Abdelsatir, 2013: 442).

Sınırsal programlama temel olarak iletişim, sadelik, geri bildirim ve cesaret ana değerleri üzerinde kurulmuş olup, çevik yazılım geliştirme ortamında müşteri gereksinimleri merkezi bir rol oynamaktadır (Beck, 2005). Sınırsal programlama yönteminde, yazılım geliştirme sürecinde tam olarak belli olmayan veya sık değişikliğe uğrayan müşteri gereksinimlerine uyum sağlanmaktadır. Çevikliği sağlayabilmek için kısa sürede tamamlanabilecek seviyede bir işle yola çıkılmakta, proje öncesi geniş çapta tasarım ve dökümantasyon oluşturulmamaktadır.



Şekil 6. XP aşamaları (Pressman, 2005: 99)

Sınırsal programlama yöntemiyle geliştirilen bir proje Şekil 6'da verilmiştir. Buna göre, sınırsal programlamada gözleme evresi, planlama evresi, iterasyondan sürüm evresi ve yayınlama evresi bulunmaktadır. Gözleme evresinde, müşteri ihtiyaçlarına göre kullanıcı hikayeleri belirlenmekte ve kullanıcı hikayelerine göre mimari tasarım için prototip

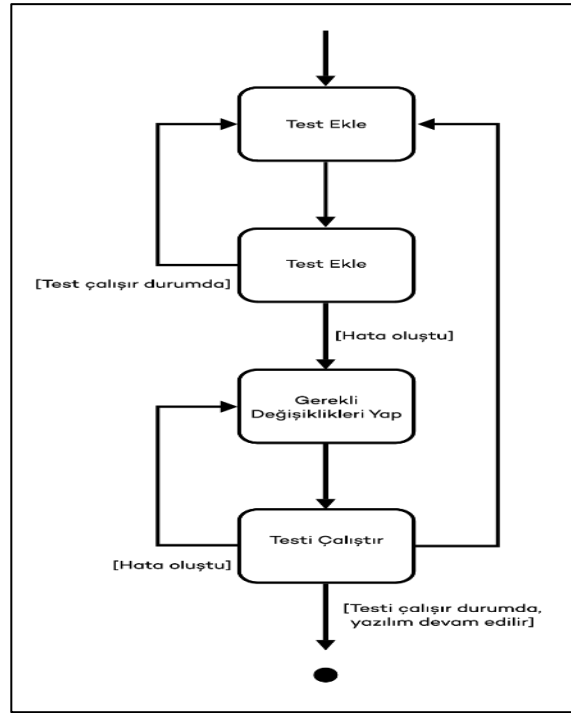
oluşturulmaktadır. Gözleme evresinden sonra planlama evresinde ise, geliştirme sürümlerinin planlaması yapılmaktadır. İterasyon evresiyle birlikte ürün geliştirilmeye başlanmakta; bu aşamada çıkan hatalar düzeltilerek yayınlama evresinde ürün canlıya alınmaktadır.

Sınırsal programlama metodolojisinde, mimari tasarımdan daha çok, yazılan koda odaklanılmaktadır. Bu da bazen en az geliştirilen kod kadar önemli olan mimari tasarımların yetersiz olmasına sebep olmaktadır. Ayrıca dökümantasyon az olduğu için, benzer hatalarla karşılaşılmasında çözüm tekrar baştan aranmakta ve bu durum organizasyonun öğrenilmiş birikimlerinin olmamasına neden olmaktadır.

2.1.2.3. Test GÜDÜMLÜ Geliştirme

Test GÜDÜMLÜ Geliştirme, kod yazılmaya başlanmadan önce test senaryolarının ve birim testlerin yazılması, sonrasında söz konusu senaryoya bağlı kalacak şekilde yazılım geliştirilmesidir. Test senaryolarının yazılım geliştirmeden önce yapılması, kodun daha iyi kurgulanmasına olanak sağlamaktadır (Beck, 2002:81). Test güdümlü geliştirme sürecinde atılması gereken adımları sırasıyla şu sıralanabilir (Dönmez, 2009: 99):

- Test senaryosunu oluşturmak,
- Senaryoyu koş ve bunun çalışmadığını görmek,
- Testin çalışması için gerekli değişiklikleri yapmak,
- Testleri tekrar koş ve hepsinin hatasız çalıştığını görmek,
- Tekrarları yok etmek.



Şekil 7. Test Güdümlü Proje Geliştirme Süreci (Dönmez, 2009: 30)

Test güdümlü yazılım geliştirme yöntemi ile birlikte oluşturulan tasarım, yazılan testler şekillendikçe oluşan bir tasarım modelidir ve bu model Şekil 7’deki gibi gösterilmektedir.

2.1.2.4. Diğer Çevik Süreçler

Kullanım alanları çok sınırlı olmakla birlikte, diğer çevik yazılım geliştirme yöntemleri aşağıdaki gibidir (Azizi ve Taqi, 2015):

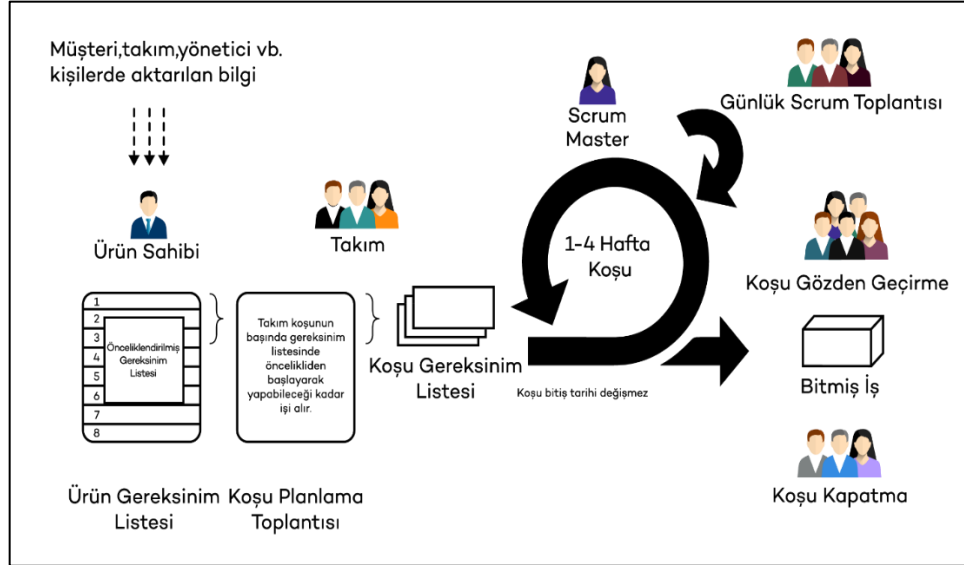
- Dinamik Sistem Geliştirme Metodu (DynamicSystem Development Method),
- Özellik Güdümlü Geliştirme (Feature-Driven Programming),
- Çevik Birleştirilmiş Süreç (AGILE UnifiedProcess),
- Uyumlu Yazılım Geliştirme (Adaptive Software Development),
- Akılcı Birleştirilmiş Süreç (RationalUnifiedProcess).

Belirtilen yöntemlerin Scrum, Kanban ve XP gibi çevik yöntemlere göre çok daha az tercih edilmesinin en önemli sebeplerinden birisi, uygulama kolaylığının basit olmamasıdır. Özellikle Scrum ve Kanban diğer çevik yöntemlere göre organizasyonlara çevik metodolojiyi uygulama aşamasında avantajlar sağlamaktadır (Azizi ve Taqi, 2015).

BÖLÜM III

3.1. ÇEVİK YAZILIM GELİŞTİRME YÖNTEMLERİNDEN SCRUM

Tez çalışmasının ana odağı olan Scrum modeli, çevik yazılım geliştirme modellerinden en yaygın olarak kullanılanı olarak gösterilmektedir (Scrum Alliance, 2018). Agile Türkiye'nin 2013 yılında yayınlamış olduğu Yazılım Üretkenlik Raporu sonuçlarına göre Türkiye'de çevik yöntemlerin kullanım oranı %64 civarında olup, Scrum yönteminin en yaygın olarak kullanılan çevik yöntemlerin başında geldiği belirtilmektedir (Çetin, 2016: 17). Scrum, Schwaber ve Sutherland tarafından geliştirilmiş, karmaşık ürün ve projelerin geliştirilmesini ve sürdürülmesini sağlayan çerçeve olarak tanımlanmıştır (Schwaber, 2004). Bu yönetime çerçeve denilmesinin sebebi, yöntemi net tariflemek yerine yöntemin sınırlarını belirtmesi, organizasyonlara Scrum çevçevesi içerisinde kendi yöntemlerini uygulama fırsatı sunmasıdır. Scrum'daki roller, etkinlikler ve eserler Şekil-8'de Scrum çerçevesi üzerinde gösterilmiştir.



Şekil 8. Scrum Çevresi (Schwaber ve Sutherland, 2013)

Scrum, felsefe olarak deneyciliğe dayanmaktadır ve Scrum yönteminin doğru şekilde işleyebilmesi için şeffaflık, gözlem ve adaptasyon sağlanabilmesi önem taşımaktadır. Şeffaflık, yapılacak işlerdeki ilerleme durumunun açık bir ortamda herkes tarafından görülebilmesini sağlamaktadır. Gözlem ise ürün parçalarının belirli aralıklarla müşteriye

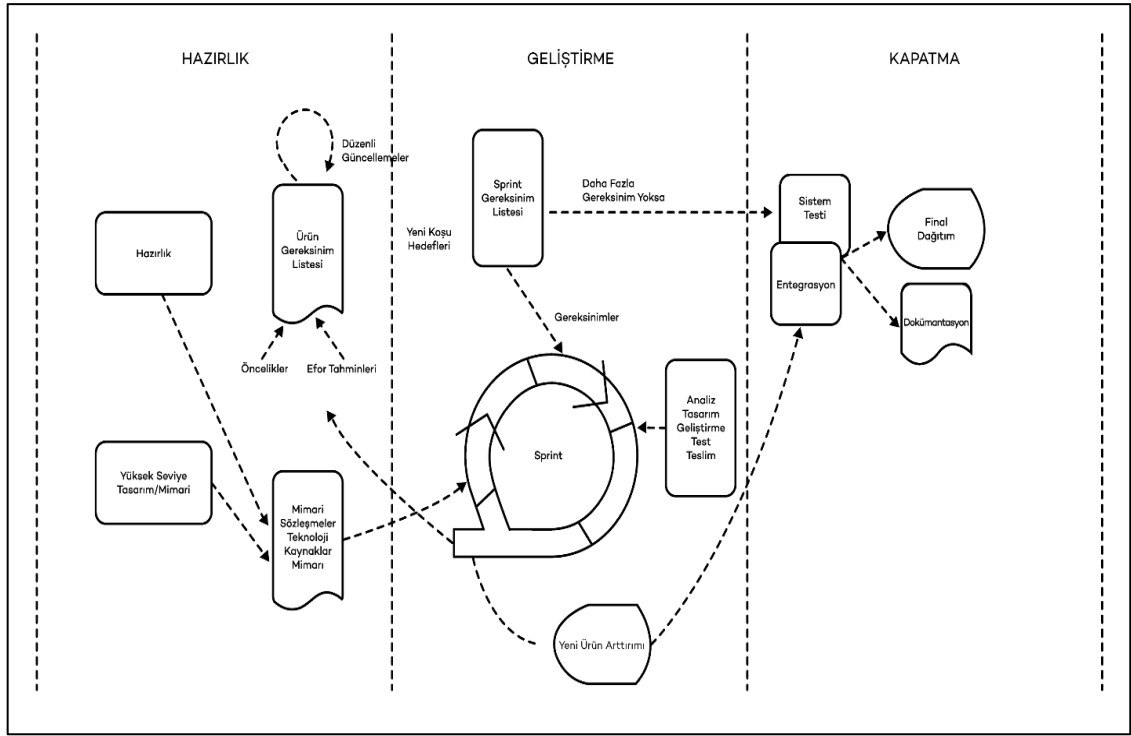
teslim edilerek gerekli düzeltmelerin yapılması ve kalan işlerin sürekli güncel tutulmasını sağlamaktadır. Adaptasyon ise geleneksel yöntemlerde olan gereksinimlerin en baştan bir defalığına belirlenmesi yerine, her teslimat sonrasında tekrar tekrar değerlendirilmesini ve duruma göre ihtiyaç duyulması halinde revizyonlar yapılmasını sağlamaktadır (Schwaber ve Sutherland, 2013).

Scrum yöntemi temel olarak üç aşamadan oluşmaktadır. Bunlar sırasıyla hazırlık, geliştirme ve kapatma aşamalarıdır. Hazırlık aşamasında ürün gereksinim listesi oluşturulmakta, kullanılacak mimari, teknik detaylar belirlenmektedir. Geliştirme aşamasında, ara dağıtımlar yapılarak ürün parçalar halinde müşterilere sunulmaktadır. Kapatma aşamasında ise son testlerle birlikte müşteriye ürünün sunumu yapılmaktadır. Şekil-9'da söz konusu aşamalar detaylı olarak gösterilmiştir.

Scrum yönteminde her aşamanın belirli adımları bulunmaktadır. Hazırlık aşamasının adımları, planlama ve üst seviye mimari tasarımdan oluşmakta olup, planlama adımıyla ise geliştirilecek ürünün maliyeti ve süre belirlemelerinin yapıldığı iş gereksinim listesi oluşturulmaktadır. Mimari ve yüksek seviye tasarım aşamasında ise çıkarılan iş listesinin elemanlarının nasıl yapılacağı belirlenmektedir (Schwaber ve Sutherland, 1995).

Geliştirme aşaması yinelemeli geliştirme adımlarından oluşmaktadır ve bu aşamada zaman, maliyet, rekabet, ihtiyaç ve kalite gibi değişkenler göz önüne alınarak yeni versiyonun geliştirilmesi yapılmaktadır. Söz konusu yenilemeli döngüler tamamlanmadan kapatma safhasına geçilmemektedir. Ürün iş listesinde işlerin tamamı tamamlanınca geliştirme aşaması tamamlanmış olmaktadır (Schwaber ve Sutherland, 1995).

Scrum'ın kapatma aşamasıyla birlikte proje tamamlanmaktadır. Bu aşamada; tüm sistem ve kullanıcı testleri tamamlanıp, ürün son kullanıcı tarafından kullanılabilir hale getirilmektedir. Ayrıca bu aşamada eğitim, kullanım ve pazarlama dokümanları gibi dokümanlar hazırlanmaktadır (Schwaber ve Sutherland, 1995).



Şekil 9. Scrum'ın Aşamaları (Kääriäinen ve Välimäki, 2008)

3.1.1. Scrum Roller

Scrum uygulamaları; Ürün Sahibi, Geliştirme Takımı ve Scrum Master olmak üzere üç ana rol üzerinden işlemektedir. Bu rollerin süreç içerisinde aldığı görev ve sorumluluklar Scrum takımının başarısı için kritik önem taşımaktadır. Scrum'da bunların dışında roller veya ünvanlar oluşturmak, çevik değişimin tam olarak anlaşılmasına ve karmaşıklığı artırarak takımların gerçek bir çevik takım olma yönünde önlerine engel olarak çıkmasına yol açmaktadır (Schwaber ve Sutherland, 2013: 4). Bu rollerin görev ve sorumlulukları aşağıdaki gibi özetlenebilir.

3.1.1.2. Ürün Sahibi

Ürün sahibi, ürün kapsamının yönetiminden sorumlu kişidir. Ürün gereksinim listesinin sorumluluğu ürün sahibinde olup, geliştirme ekibi, proje sürecinde ürün sahibinin yönlendirmesine göre hareket etmektedir. Ayrıca geliştirme döngüleri sonunda, ürün çıktılarının onayı da yine ürün sahibi tarafından yapılmaktadır. Ürün Sahibi tek bir kişiden oluşmakta olup bir komite veya grup değildir. Diğer kişiler ürün sahibi ile ürün hakkında

bilgi alışverişinde bulunabilirler fakat ürün üzerinde yapılacak değişiklik ancak ürün sahibinin onayıyla gerçekleşebilir (Cohn, 2007). Sonuç olarak, ürün sahibi Scrum takımlarının yaptığı iş ve üründen sorumlu olan tek kişidir.

Ürün sahibi'nin sorumluluğu ürünün değerini ve geliştirme takımının işini en üst seviyeye çıkarmak olup, ürün sahibinin bunu nasıl yapacağı ise; organizasyon, takım ve kişi bazlı olarak farklılık göstermektedir (Schwaber ve Sutherland, 2013: 5).

3.1.1.1. Geliştirme Takımı

Scrum takımı, projenin geliştirmesinden sorumlu takımdır ve bu takım geliştirme sürecindeki işleri çapraz bir şekilde gerçekleştirebilen kişilerden oluşmaktadır. Geliştirme takımı, ürün geliştirme sürecinde işlerin büyüklüğünü ve karmaşıklığını belirleyerek, sprint sürecinde işleri önceliklendirebilmektedir. Ayrıca takım, içinde iş ataması olmadığı için kendi kendine organize olabilen bireylerden oluşmaktadır (Kır, 2014: 9; Schwaber ve Sutherland, 2013: 4).

Scrum takımı için kişi sayısının çok az olması takım üyeleri arasındaki etkileşimi azaltmakta ve üretkenlik artışını sınırlandırmaktadır. Takım üyesinin çok fazla olması ise karmaşıklığı arttırmakta, takımın organize olmasını zorlaştırmaktadır. Bu yüzden Scrum takımlarının 5-9 kişiden oluşması ideal olarak kabul edilmektedir (Baytam ve Kalıpsız, 2011: 49).

3.1.1.3. Scrum Master

Scrum Master'ın görevi, takım içerisinde Scrum'ın anlaşılmasını ve uygulanmasını sağlamaktır. Bu görevi de, takımın Scrum'ın teorisine, pratiğine ve kurallarına uygun olarak gerçekleştirmesini sağlayarak yerine getirmektedir.

Judy ve Krumins (2008)'e göre Scrum Master geliştirme sürecinde takımının karşılaştığı engellerin aşılması için takıma hizmet ederek, Scrum takımı için ideal geliştirme ortamının sağlanmasına yardımcı olmaktadır. Ürün sahibi tarafından hazırlanan iş listesi bazen takım tarafından tam ve doğru anlaşılmamaktadır. Bunları takım için anlaşılır hale getirmek ve ürün sahibini daha anlaşılır ürün gereksinim listesi hazırlaması konusunda yönlendirmek Scrum Master'ın görevlerindendir. Scrum Master projede yönetici olarak

nitelendirilmediği için, takımı yönlendirirken talimat verme ve müdahala etme şeklinde bir yetkiye sahip değildir (Kır, 2014: 9).

Scrum Master bir danışman koç gibi ekibine liderlik ve rehberlik edip başarılarını takdir eden kişi olarak görülmektedir. Süreç içerisinde Scrum kurallarını sıkı takip ederek bir hakem gibi sürecin doğru ilerlemesini takip etmektedir (Baytam ve Kalıpsız 2011: 48).

3.1.2. Scrum Etkinlikleri

Scrum içinde tanımlanmamış toplantı ve etkinlik sayısını en aza indirmek için Scrum kılavuzu içinde belirlenmiş bir takım etkinlikler bulunmaktadır. Söz konusu etkinlikler belirli bir sıraya göre gerçekleştirilmekte ve belirli bir zaman içinde tamamlanması gerekmektedir. Her bir etkinliğin amacı net olarak belirlenmiş olup, tüm etkinlikler için belirlenmiş maksimum süreler bulunmaktadır. Scrum sürecinde amaç, içinde bulunulan etkinliğin belirtilen sürede amacına ulaştırılıp bir sonrakine geçilmesidir. Zaman kısıtı bulunmasının sebebi, gerekli etkinliğe süreç içinde uygun miktarda sürenin harcanıp zamanın etkin ve verimli bir şekilde kullanımının sağlanmasıdır (Pichler, 2010: 98). Ayrıca Scrum içerisindeki her bir etkinlik gözlem ve adaptasyon için fırsat sağlamaktadır. Bu etkinlikler takımın şeffaf bir ortamı sağlayabilmesi için gereken özelliklerde tasarlanmıştır. Dolayısıyla, herhangi bir etkinliğin gereken kriterlere uygun yapılmaması istenen şeffaflığın kaybolmasına ve böylelikle adaptasyon fırsatlarının azalmasına sebep olmaktadır. Scrum çerçevesindeki bu etkinlikler sırasıyla aşağıdaki gibi açıklanmaktadır (Schwaber ve Sutherland, 2013: 7).

3.1.2.1. Sprint (Koşu) Planlama Toplantısı

Her sprint başlangıcında yapılan toplantılara sprint planlama toplantısı denilmektedir. Bu toplantının süresi bir aylık sprint için 8 saat, daha kısa süreli sprintler için daha kısa olacak şekilde belirlenmiştir (Schwaber ve Sutherland, 2013: 8).

Sprint planlama toplantısında, başlayan sprintte ürün parçasından ne tamamlanıp teslim edileceği ve bu teslimatı yapabilmek için hangi işlerin gerekli olduğu sorularının cevabı aranmaktadır. Bu iki sorunun cevabının arandığı sprint planlama toplantısı iki aşamada gerçekleştirilmekte olup, ilk aşamada takım üyeleri önceliklendirilmiş iş listesi üzerinden

işlerin büyüklüklerini belirlemekte ve iş kapsamında yapılacak işleri tartışmaktadır. İkinci aşamada ise sprint boyunca tamamlanabilecek büyüklükte işlerden sprint iş listesinin belirlenmesi gerçekleştirilmektedir. İkinci aşamada ise alınan işlerin üzerinden detaylı bir şekilde geçilerek yapılacak görevlerin belirlenmesi sağlanmaktadır (Pichler, 2010: 99).

3.1.2.2. Günlük Scrum Toplantısı

Günlük Scrum toplantısı her bir çalışma günü için yapılmakta olup, tüm Scrum takımı üyelerinin bir önceki gün neler gerçekleştirdiğini, bugün ne gerçekleştireceğini ve işi gerçekleştirirken muhtemel olarak karşılaşılabilecek sorunları tespit edip, bunları cevaplandırabileceği maksimum 15 dakika ile kısıtlanan toplantıdır (Schwaber ve Sutherland, 2013: 10).

Takım için faydalı olan bu toplantılarda, takım üyeleri problemlerini aktarma fırsatı bulmaktadır. Söz konusu problemler için o an çözüm aranmamaktadır. Buna yönelik, Scrum master, takım üyelerinin problemlerini not alarak günlük Scrum toplantısı sonucunda takım üyesi ile detaylı konuşarak problemlere çözüm aramaktadır (Pichler, 2010: 102).

3.1.2.3. Sprint Değerlendirme Toplantısı

Her sprint sonunda sprint sürecinde geliştirilen ürünün sunumunun yapıldığı toplantılara Sprint değerlendirme toplantısı denilmektedir. Toplantının süresi bir aylık Sprint için 8 saat, daha kısa süreli sprintler için daha kısa olarak gerçekleşmektedir (Schwaber ve Sutherland, 2013: 10).

Sprint değerlendirme toplantısına Scrum takımının yanı sıra, ürünün paydaşları, konu ile ilgili uzmanlar ve ürünün son kullanıcıları katılmakta, kişiler konu ile ilgili görüşlerini iletmektedir. Sprint değerlendirme toplantısının çıktılarından biri de, bir sonraki sprint planlama toplantısının girdisi olmaktadır. Bu toplantıda alınan geri bildirimler ürün gereksinim listesinin güncellenmesini sağlamaktadır (Pichler, 2010: 103).

3.1.2.4. Sprint Geçmiş Değerlendirme Toplantısı

Sprint geçmiş değerlendirme toplantısı, sprintin en sonunda yapılan ve takım üyelerince bir önceki sprintte hangi noktalarda problem yaşandığının ve bu problemler için ne gibi

değişiklikler yapılabilceğinin sorgulandığı etkinliktir. Söz konusu etkinliğin süresi bir aylık Sprint için 3 saatle sınırlandırılmış olup, daha kısa süreli sprintler için daha az olmaktadır (Schwaber ve Sutherland, 2013: 11).

Geçmiş değerlendirme toplantısında sprintin insan, ilişki ve süreçler bakımından nasıl geçtiği değerlendirilmektedir. İyi giden ve iyileştirilmesi gereken noktalar tespit edilip, Scrum takımının daha iyi iş yapmasını sağlayacak planlar oluşturulmaya çalışılmaktadır.

3.1.3. Scrum Eserleri

Scrum'ın eserleri, şeffaflığın yanı sıra gözlem ve adaptasyon fırsatları sunmak için yapılan işi veya üretilen değerleri temsil etmektedir. Bu eserler, her bir paydaşın anlayabileceği düzeyde ve standartta, bilginin şeffaflığını en üst seviyeye yükseltecek şekilde tasarlanmaktadır.

3.1.3.1. Ürün İş Listesi

Ürün gereksinim listesi, ürünün tamamlanması için yapılması gerekenlerin olduğu bir liste olup, söz konusu liste ürün sahibinin sorumluluğunda oluşturulmakta ve kontrol edilmektedir. Ürün iş listesi, sürekli gelişen ve değişen dinamik bir yapıya sahiptir. Bu listede ürünün özellikleri, işlevleri, iyileştirmeleri ve düzenlemeleri bulunmaktadır. Listede ayrıca, her bir iş kaleminin önceliği, büyüklüğü, tahmini maliyeti ve değeri yer almaktadır (Schwaber ve Sutherland, 2013: 14).

Geliştirme takımı sprint planlama toplantısında ürün gereksinim listesindeki işlerden en öncelikli olanları seçerek sprint sürecinde yapılacak ürün parçasını belirlemektedir. Bir ürün sahibinin sorumluluğundaki iş listesi üzerinde birden fazla takım çalışabilmektedir. Ürün gereksinim listesine herhangi bir takım üyesi madde ekleyebilmekte fakat bunun önceliklendirmesi ürün sahibi tarafından, büyüklüğünün belirlenmesi ise takım tarafından yapılmaktadır. Ürün gereksinim listesi içerisinde yer alan maddelere ekleme ve çıkartma yapılabilmekte olup, ürün büyüklüğünün belirlenmesinde genellikle fibonacci sayıları (1-3-5-8-13-21...) yaygın olarak kullanılmaktadır (Hundermark, 2009).

3.1.3.2. Sprint İş Listesi

Sprint iş listesi, takımın o sprintteki hedefine ulaşabilmesi için tamamlaması gereken işlerin daha görünür kılındığı, sprint planlama toplantısında ürün gereksinim listesi içerisinde seçilen ve bir sprint boyunca geliştirilecek işlerin olduğu liste olarak tanımlanmaktadır (Schwaber ve Sutherland, 2013: 13). Ürün gereksinim listesinin sorumluluğu ürün sahibinde olmasına rağmen, sprint gereksinim listesinin sorumluluğu geliştirme takımındadır. Sprint sürecinde takımın onayı olmadan ürün sahibi bu listeye madde ekleyememektedir. Geliştirme sürecinde takım bu listeyi sürekli olarak güncelleyerek, tamamlananları “Bitti” olarak işaretlemekte, üzerinde çalışmaya devam edilenlerin tahmini kalan süresi sürekli olarak güncel tutulmaktadır. Takımın hedefi, sprint sonuna kadar sprint listesindeki bütün işleri “Bitti” statüsüne getirmektir (Pichler, 2010: 101).

3.1.3.3. Ürün Parçası

Scrum takımı her sprint sonunda bir ürün veya çalışabilir bir yazılım parçası çıkartmayı hedeflemektedir. Scrum’da, müşteri için anlamlı ve fonksiyonel olan bu yazılıma “Ürün Parçası” denilmektedir. Sprintin sonunda, yeni ürün parçası “Bitti” olarak işaretlenmelidir. Bu da, tamamlanan parçanın takımın “Bitti” tanımındaki koşulları sağladığı anlamına gelmektedir (Schwaber ve Sutherland, 2013: 14).

3.1.3.4. Ürün Kalan Zaman Grafiği

Ürün kalan zaman grafiği, ürün iş listesindeki işlerin ne kadar sürede yapılacağını tahmin edilip hesaplanmasıyla oluşturulmaktadır. Ürün geliştirme süreci boyunca iş listesinde sürekli güncelleme yapıldığı için ürün zaman grafiği de sürekli güncellenmektedir (Çetin, 2016: 29). Ürün kalan zaman grafiği, proje süresinin net bitiş tarihini vermemekte olup, projenin tahmini olarak ne kadar sürede tamamlanabileceğinin ön görülmesini sağlamaktadır.

3.1.3.5. Sprint Kalan Zaman Grafiği

Sprint planlama aşamasında sprint listesindeki işleri tamamlamak için ihtiyaç olan süreler tahmin edilmektedir. Günlük Scrum toplantılarında dün ne yaptım sorusuna istinaden

ilgili görevlerin kalan süreleri sürekli olarak güncellenmektedir. Gün bazlı kalan işlerin tamamlanması gereken toplam süreler, grafik üzerinde gösterilmektedir. Söz konusu grafik, sprint sürecinde kalan işlerin sürelerini göstermektedir.

Sprint süresi içerisinde listedeki işlerin ne kadarının yapıldığını, ne kadarının kaldığını ve hedefe ulaşmak için tahmini sürenin ne kadar olduğunu karşılaştırmalı olarak gösteren grafiğe sprint kalan zaman grafiği denilmektedir (Çetin, 2016: 29).

BÖLÜM IV

4.1. GELENEKSEL YÖNTEMDEN ÇEVİK YÖNTEME GEÇİŞ

Geleneksel yazılım geliştirme yöntemleri geçmişten günümüze kadar birçok işletme tarafından başarılı bir şekilde kullanılmış ve kullanılmaya devam etmektedir. Günümüzde söz konusu yöntemleri başarılı bir şekilde uygulayan ve yazılım geliştirme projelerini bu yöntemlerden birini kullanarak yürüten birçok işletme bulunmaktadır (DeCarlo, 2004). Son yıllarda ise hızlı değişen çevre koşulları ve müşteri ihtiyaçlarının bulunduğu ortamlarda geleneksel yöntemler, özelliklerinden dolayı işletmelerin ihtiyaç ve beklentilerini karşılamamaya başlamıştır. Buna bağlı olarak geleneksel yöntemlerin günümüz koşullarında işletmelerin beklentilerini karşılamak konusunda kısmen yetersiz kaldığının düşünülmesine sebep olmuştur. Geleneksel yazılım geliştirme yöntemlerinin yetersiz kabul edilmesinin temelinde, proje geliştirme aşamasında bir aşama tamamlanmadan diğer aşamaya geçilmesine izin verilmemesi yatmaktadır. Buna bağlı olarak geleneksel yöntemlerde, müşteri ihtiyaçlarının tamamının ilk aşamada belirlenmesi gerekmektedir. Fakat müşterilerin proje başlarında ne istediklerini tam olarak açıklayamaması en sık karşılaşılan problemlerden birisi olarak karşımıza çıkmaktadır. Ayrıca müşteriler, değişen çevre koşullarından dolayı isteklerinde değişiklik yapma eğiliminde olmaktadır. Bu sebeplerden dolayı geleneksel yöntemlerle yürütülen projelerde, geliştirme sürecinde oluşabilecek değişikliklere uyum sağlanması konusunda zorluklar çıkmaktadır (Baytam ve Kalıpsız, 2011: 7).

Geleneksel yazılım geliştirme süreçlerinde karşılaşılan problemlerden bir diğeri de, geleneksel yöntemlerin birbirini takip eden safhalardan oluşmasıdır. Bu aşamalardan birisi tamamlanmadan bir sonraki aşamaya geçilmediği gibi, bir aşamanın çıktısının kendinden sonra gelen aşamanın girdisi olmasıdır. Sonraki aşamalarda örneğin test aşamasında fark edilecek herhangi bir hata tüm yapılan işin değiştirilme riskini barındırmakta, bu durum ise projenin hayata geçme süresini uzatmaktadır (Baytam ve Kalıpsız, 2011: 7). Tüm bu sıkıntılar dolayısıyla, proje başarı oranında düşme ve yazılım geliştirme sonunda çıkan ürünün müşteriye memnun etmemesi gibi sonuçlar ortaya çıkmakta olup, işletmeler çözüm olarak çevik yazılım geliştirme yöntemlerini tercih

etmeye başlamaktadır. Çevik yazılım yöntemlerine geçişle birlikte, işletmeler dökümantasyon fazlalığını ortadan kaldırmaya çalışmakta, değişimlere daha hızlı uyum sağlayarak proje başarı oranlarını arttırmayı amaçlamakta ve müşteriye teslim edilen çalışmalarda daha olumlu sonuçlar almayı beklemektedirler (Başar ve diğ., 2015: 39).

4.1.1. Scrum'ın Uygulanabilme Başarısı

2000'li yıllardan sonra birçok işletme, projelerinde başarı yüzdesini arttırmak ve müşterilerini daha memnun etmek için yazılım geliştirme süreçlerinde Scrum yöntemini kullanmaya başlamıştır. Scrum çerçeve olarak basit bir yöntem olmasına rağmen geçiş sürecinde dönüşümü yönetmek ve başarıyla sonuçlandırmak açısından zor ve karmaşık bir iş olarak ifade edilmektedir (Scrum Alliance, 2018). Bu durum günümüzde hala geleneksel yöntemlerin birçok büyük işletme tarafından çeşitli uygulama alanlarında kullanılmaya devam edilmesinin sebebini açıklamaktadır (Çetin, 2016: 17).

İşletmelerin dönüşüm sürecinde çeşitli problemlerle karşılaşmalarının temel sebepleri, dönüşüm süreci içerisinde işletmelerin eski alışkanlıklardan vazgeçememesi ve değişim sürecinde daha önce hiç karşılaşılmayan ve öngörülemeyen problemlerle karşılaşılmasıdır (Ellis, 2016). Dönüşüm sürecinde karşılaşılan söz konusu problemlerin çözümü ve dönüşüm sürecinin başarısı için organizasyon içerisinde Scrum hakkında bilgi eksikliğinin bulunmaması, Scrum ile ilgili yeterli bilgi ve deneyime sahip danışman koçların istihdam edilmesi gerekmektedir. Scrum basit bir çerçeve olmasına rağmen organizasyonlarda bu kültürü uygulamak ve dönüşümü gerçekleştirmek uzun sürmekte, dönüşüm için zaman ve emek harcanması gerekmektedir. Dönüşüm sürecinin uzun sürmesi belirli bir süre sonra yönetimin dönüşüme olan desteğinin azalmasına veya sonlanmasına sebep olabilmektedir. Yönetim desteğinin bulunmadığı ortamda ise başarılı bir dönüşümün gerçekleşmesi mümkün görünmemektedir (Schindler, 2008).

Scrum dönüşümünden beklenen faydayı sağlamanın en önemli ölçütlerinden birisi, işletmelerin yazılım geliştirme süreçlerinde proje başarı metriklerinde artış sağlamasıdır. PMI'a göre başarılı projeler uygun maliyette ve zamanda tamamlanan, istenilen düzeyde performans sağlayabilen ve kalite olarak müşteri tatmininin sağlandığı projelerdir (Jeffery ve Dennis, 1988). Scrum yöntemi, çalışan yazılımın sürekli müşteriye gösterilmesi ile

yenilemeli bir yapıya sahiptir. Bu sebeple müşterinin istemediği yöne fazla ilerlemeden müdahale edilmesini sağlamaktadır. Bu yüzden Scrum'ı başarılı olarak uygulayabilen organizasyonlarda verimliliğin ve müşteri memnuniyetinin arttığı gözlemlenmektedir (Ellis, 2016).

Scrum yöntemi çevik prensipleri gereği kapsamlı dokümantasyon yerine, çalışan yazılıma değer vermektedir. Bu da, Scrum yönteminde yazılım takımlarının, uzun dokümantasyon hazırlamak yerine bir an önce çalışan yazılım üretilmesine odaklanmasını sağlamaktadır. Söz konusu durumu başarabilen takımlar Scrum yöntemi ile işlerini daha kısa sürede tamamlamakta ve daha hızlı sonuca ulaşmaktadır. Yazılım geliştirme takımlarının teslimat sürelerindeki iyileşme, Çevik dönüşümün başarılı olarak uygulanmakta olduğunun göstergelerinden biridir (Thuy, 2013: 30).

Thuy'a (2013) göre Scrum metodolojisinin başarılı uygulandığı işletmelerde yazılım geliştirme süreçlerinde verimliliğin artması ve yazılım teslimat sürelerinin iyileşmesi beklenmektedir. Bu durum dolaylı olarak yazılım geliştirme maliyetlerinin azalmasını sağlamaktadır. Çevik dönüşümü başarılı olarak gerçekleştiren bir işletmede, yazılım geliştirme maliyetlerinin azalması ve yazılım teslimatlarının daha kısa sürelerde yapılması beklenmektedir.

Scrum'ın geleneksel yöntemlerden en önemli farklarından birisi kalitenin her zaman ön planda tutulmasıdır. Çevik dönüşümün başarılı bir şekilde sağlanması, işletmede daha düşük maliyetle daha yüksek kaliteli yazılımlar üretilmesini sağlamaktadır. Yeterli kaliteye sahip olmayan ürün parçaları, Scrum takımının belirlediği "Bitti" kriterine uymadığı için söz konusu kalite şartlarını sağlamadan üretim ortamına gönderilmemektedir. Dolayısıyla başarılı bir çevik dönüşümün olduğu ortamlarda yazılım ve ürün kalitesinde artış beklenmeli, üretim ortamında geleneksel yöntemlere göre daha az hata ile karşılaşılmalıdır (Kır, 2014: 19).

Scrum yöntemi, çevik prensiplerine bağlı olarak belirli bir plana ve sözleşmelere bağlı kalınmasından çok, değişime karşılık verilmesini önemsemektedir. Bu da, geliştirilen ürünün değişen müşteri ihtiyaçlarına göre sürekli değişebileceği anlamına gelmektedir. Geleneksel yöntemlerde müşteri ihtiyaçlarının en baştan belirlenmesi istenmekte ve

yazılımın üretildiği aşamada ihtiyaçlar değişime kapatılmaktadır. Bu durum, süreç sonucunda ortaya çıkan ürünün müşteri ihtiyaçlarını karşılamaması ve müşteri memnuniyetinde düşüş yaşanması ile sonuçlanmaktadır. Scrum'ın başarılı olarak uygulandığı bir ortamda müşteri ihtiyaçlarındaki değişime daha hızlı yanıt verilerek, müşteri memnuniyetinde artış gözlemlenmelidir (ACM Yazılım Çözümleri, 2017). Scrum geçişi sonrası müşteri ihtiyaçlarına hızlı yanıt verme ve müşteri memnuniyetinde artış sağlanması gibi değişimler gözlemlenmişse bu işletmede çevik dönüşümün başarılı bir şekilde gerçekleşiyor olduğu söylenebilir.

Aşağıdaki bölümde işletmelerin Scrum tercihleri sonrası çevik dönüşümünü etkileyen faktörler incelenmiş olup, söz konusu faktörlerin işletmenin çevik dönüşümüne olan etkileri değerlendirilmektedir.

4.1.2. Scrum Dönüşümünün Başarısını Etkileyen Faktörler

Literatürde, işletmelerin yazılım geliştirme süreçlerinde geleneksel yöntemlerden Scrum yöntemine geçişlerini etkileyen faktörlerin sınıflandırılmasıyla ilgili olarak çeşitli çalışmalar bulunmaktadır. Örneğin, Chow ve Cao (2008) Scrum yöntemi ile geliştirilen projelerin başarısını etkileyen kritik faktörleri; organizasyonel yapı, süreçler, insanlar, kullanılan teknolojiler ve tasarım olmak üzere beş ana grup altında toplamıştır. Bir başka çalışmada Nerur ve diğerleri (2005), çevik dönüşümün başarısını etkileyen faktörleri, yönetim, organizasyon yapısı, kişi, süreç ve teknoloji şeklinde sınıflandırmıştır. Silva ve Santos (2015) yapmış olduğu çalışma da ise, Scrum'da yazılım geliştirme projelerini etkileyen faktörleri insan, süreç, teknoloji başlıkları altında gruplayarak incelemektedir. Boehm ve Turner (2005) ise Scrum yönteminin başarılı bir şekilde uygulanmasında karşılaşılan engelleri mevcut yazılım geliştirme süreçlerinin çevik değerlerle çelişmesi, iş süreçlerin çevik değerlere uyum sağlayamaması ve insandan kaynaklanan faktörler olarak üç farklı gruba ayırmaktadır (Paulk, 2009: 10).

Bu çalışmada ise çevik dönüşüme etki eden faktörler, çalışanların Scrum farkındalığı ve Scrum hakkındaki yaklaşımları ile ilgili faktörler insan faktörü, yöntemin uygulanma süreci, takımların eğitim ve gelişimleri, sorumlulukları ve danışman desteği alabilmeleri ile ilgili faktörler uygulama faktörleri ve organizasyonel değişim, değişimin tüm işletmeye

yayılabilmesi, yönetim desteği ile ilgili faktörler de organizasyonel faktörler altında ele alınmıştır.

4.1.2.1. İnsan Faktörü

Thuy (2013)'a göre insan faktörü çevik dönüşümdeki en önemli faktörlerden biridir. Scrum'daki tüm rollerin, görevlerini yerine getirmek konusunda istekli ve yeterli bilgiye sahip olması, yazılım geliştirme projesinin başarısında en büyük etkiye sahiptir. Çevik dönüşümünde insan faktörünün önemi çevik bildirimde de vurgulanmış olup; bireyler ve etkileşimlerine, müşteri ile işbirliğine ve müşteriler tarafından önerilen değişikliklere yanıt verebilmenin önemine vurgu yapılmaktadır (Scrum Alliance, 2018). Scrum dönüşümü etkileyen insan faktörü:

- Takım profili,
- Disiplin,
- Motivasyon ve katılım,
- Teknik yeterlilik,
- Uyum,
- İletişim becerileri,
- Kişisel karakteristikler ,
- Sosyal kültür

gibi birçok alt nitelikten oluşmaktadır (Hameed ve Diğerleri, 2016: 85; Ozierańska, 2015: 82). Bu tez çalışmasında insan faktörü grubu altındaki bu alt faktörlerden olan takım profili, disiplin ve motivasyon üzerinde durulmuştur.

İnsan faktörünün alt niteliklerinden biri olarak incelenen takım profili, Scrum takımlarının kendi organizasyonunu sağlayabilecek düzeyde kalifiye olmasını ifade etmektedir. Ürün geliştirme takım tarafından yapıldığı için takım, çevik prensiplerin temeli olan ürünü geliştirebilecek kapasitede olmalı ve değişikliklere cevap verebilecek nitelikte donanımına sahip olmalıdır. Eğer geliştirme takımları bu koşulları sağlamıyorsa Scrum dönüşümün başarılı olacağından söz edilemez (Silva ve Santos, 2015: 51).

Bir diğ er alt nitelik ise, takım  yelerinin disiplini dir. Scrum ba arısı Scrum takımının disiplini ile doğ ru orantılıdır. Scrum'da en temel disiplin Scrum toplantılarına ge  kalmamak ve toplantıya katılamama durumunda takım  yesinin herhangi bilgi eksikliğı ya amamasının saėlanmasıdır (Ozierańska, 2015: 87). Bu t r disiplin sorunları takım  yelerinin daha  nceki alışkanlıklarının yanı sıra; ki isel bir disiplin eksikliğinden kaynaklanabilmekte olup, s z konusu durumun Scrum toplantılarının verimliliğine ve takım motivasyonu  zerine olumsuz etkileri bulunmaktadır. Takım  yelerinin toplantıya gecikmesi toplantıların veriminde d   kl ğ e sebep olmakla birlikte, bu durum takım  yelerinin birbirlerine kar ı olumsuz tutumları ile sonu lanmaktadır (Ozierańska, 2015: 88). Bu sebeple,  evik d n    mde ve Scrum'ın uygulanma ba arısında takım  yelerinin disiplini  ok  nemli bir yer tutmaktadır.

 nsan fakt r  altında incelenen bir diğ er nitelik ise takım  yelerinin Scrum uygulamaları konusundaki motivasyonu ve  evik d n    m  benimsemeleridir. Scrum'da ba arı i in en  nemli etkenlerden biri, ekip  yelerinin yalnızca Scrum yapmak istemeleri deėil,  evik olmak istemeleri ve deėi ime a ık olmalarıdır (Thuy, 2013: 43). Scrum takımının motivasyonunu en  ok etkileyen nitelikler, Scrum etkinliklerinin verimli bir  ekilde ge irilebilmesi ve Scrum master'ın takımını Scrum konusunda motive edebilme becerisidir. Scrum takımının motivasyonu takımın  retkenliğini ve performansını doėrudan etkilediğı i in, Scrum d n    m  s recinde motivasyonun  nemi  ne  ıkmaktadır. Konboy ve diğ erlerine g re (2011) Scrum takımının motivasyonunun y ksek olması, takımın hedefine adım adım yakla masına ve kaliteli bir  r n ortaya  ıkartılmasına b y k katkı saėlamaktadır.

4.1.2.2. Uygulama Fakt rleri

Scrum d n    m n  etkileyen uygulama fakt rleri:

- Rol ve sorumluluklar,
- Eėitim ve geli im,
- Scrum etkinlikleri,
- Scrum ara ları,
- Ko  ve Danı man desteėi,

- Takım yapısı,
- Sprint süresi,

olarak sayılabilir (Hameed ve diğerleri, 2016; Misra, 2009; Ozierańska, 2015).

Uygulama faktörleri içerisinde incelenen ilk alt nitelik Scrum takımının rol ve sorumluluğudur. Ozierańska'a (2015) göre Scrum takımında bireysel rollerin doğru bir şekilde anlaşılması ve uygulanması Scrum metodolojisinin başarılı bir şekilde uygulanabilmesine önemli derecede etki etmektedir.

Scrum Master'ın takımın varoluşuna ve başarısına olan adanmışlığı, Scrum takımının başarısını ve doğrudan çevik dönüşümün başarısını etkilemektedir. Scrum yönteminde ürün sahibinin rolünün önemi de büyük olduğundan, ekibin ürün sahibi ile koordineli çalışabiliyor olması dönüşüm sürecinin başarılı olmasındaki en önemli kriterlerden biri olarak görülmektedir (Ozierańska, 2015: 89). Schwaber ve Sutherland (2013)'e göre Scrum master Scrum takımı için "hizmet veren bir lider" rolündedir. Bu sebeple, Scrum master ekibe hizmetkar liderlik sağlayabilecek nitelik ve nicelikte bilgi ve donanımına sahip olmalıdır. Ayrıca bu kişi hizmetkar lider olduğu için Scrum konusunda takımı sürekli motive edebilmelidir.

Çevik dönüşümlerde ve Scrum uygulamalarında uygulama faktörleri altında dikkat çeken bir diğer alt başlık ise eğitim ve sürekli gelişmenin sağlanmasıdır. Scrum'da eğitim ve takımın sürekli gelişmesi Scrum başarısı için vazgeçilmez öğedir. Scrum takımlarının tümü tarafından Scrum ilkelerinin ve değerlerinin anlaşıldığı bir ortamın sağlanması takımın birçok konuda daha uyumlu hareket etmesi için önemlidir. Lindvall ve diğerleri (2002) bir organizasyonda çevik uygulamaları başarılı bir şekilde uygulayabilmek için eğitim faktörünün önemine dikkat çekmiş olup, organizasyonun çevik dönüşüm öncesinde ne kadar bir eğitime ihtiyacı olduğunun belirlenmesi gerektiğini vurgulamaktadır. Bir başka çalışmada ise, Misra(2007) sürekli öğrenmeyi, çevik yöntemlerle geleneksel yöntemler arasındaki en temel farklardan biri olarak görmüş ve insanların birbiriyle sürekli bilgi paylaşmasının ve öğrenmeye istekli olmasının çevik dönüşümün başarıya ulaşma şansını arttıracığını belirtmiştir .

Scrum uygulama aşamasında Scrum etkinliklerinin doğru ve etkili bir şekilde uygulanması çevik dönüşümünün başarısını etkileyen bir diğer faktördür. Geliştirme takımları bu pratikleri uygularken genel prensiplerin dışına çıkmadan, projenin neden yapıldığını ve nasıl yapılacağını bilerek uygulamalıdır. Linders (2013)'e göre Scrum'ın bir başarı faktörü de, çevik felsefe ve Scrum ilkelerinin dönüşüme başlamadan önce geliştirme takımı tarafından anlaşılmasının sağlanmasıdır. Bu felsefe ve ilkeler yeterince öğrenilmeden Scrum uygulanmaya başlanması dönüşüm sürecinin başarısızlıkla sonuçlanmasına neden olabilmektedir. Scrum ilkelerinin anlaşılması için düzenlenen Scrum toplantılarının bir amacı da Scrum'da denetim ve uyum yeteneğinin sağlanmasıdır. Söz konusu toplantılar Scrum sürecinde hedefe ulaşma başarı derecesinin ve yolda duran sorunların belirlenmesine yardımcı olmaktadır. Scrum toplantılarının kurallarının ihlal edilmesi, Scrum'ın temel prensiplerine uyulmaması anlamına geldiği gibi, bu durum takım üyelerinin çalışma motivasyonunu ve kordinasyonlarını olumsuz etkilemektedir (Ozierańska, 2015: 88).

Scrum araçları, Scrum takımının bulunduğu konumu, ilerlediği yolu ve gideceği hedefi açık bir şekilde görülmesini sağlayan araçlardır. Geliştirme takımının ürün listesinde yapılacak bütün iş ve değişiklikler tanımlanmalı ve ilgili liste sürekli güncel tutulmalıdır. Takım, her sprint planlama toplantısında ürün gereksinim listesini dikkate alarak sprint listesini oluşturmalıdır. Sprint kalan zaman grafiği güncel tutulmalı, bu grafik üzerinden sprintte tamamlanmak üzere alınan işlerden, tamamlanan ve henüz tamamlanmamış işlerin büyüklüğü net bir şekilde görülebilmelidir. Ürün kalan zaman grafiği, takımın sprint hedefini gerçekleştirmeye ne derece yakın olduğunu görebilmesi için ve toplam kalan işi izleyerek ilerleyebilmesi için önemlidir (Ozierańska, 2015: 88).

Uygulama faktörleri grubunun altındaki son alt başlık, Scrum konusunda uzman kişilerden ve kendini bu işe adanmış çevik koçlardan destek alınmasıdır. İyi bir çevik koçluk hizmeti, geçiş zorluklarıyla karşılaşıldığında geliştirme takımına yardımcı olacaktır. Scrum dönüşümünde başarı, danışmanlığın geçiş öncesi ve geçiş sırasında sağlanmasıyla mümkün olmaktadır (Gandomani, 2014: 263). Scrum'ın uygulanmaya başlanması aşamasında iyi bir koçluk ve danışmanlık hizmeti sağlanması, ekip üyelerinin

cesaretinin kırıldığı yerlerde cesaretlendirilmesi ve ekibin karşılaştığı geçiş zorluklarının aşılması açısından önemlidir. Thuy (2013)' a göre organizasyonların değişime açık olması ve çevik olmaya istekli olması gerekmektedir. Bunun mümkün olabilmesi için, organizasyonun doğru bir şekilde değiştirildiğini anlayan ve ihtiyaç halinde değişimi yönlendirebilen deneyimli koçlara ihtiyaç duyulmaktadır. Değişim, istikrarlı bir şekilde ve adım adım gerçekleştirilmelidir. Bu sebeple, Scrum dönüşümü aşamasında organizasyonların alacağı iyi bir koçluk ve danışmanlık hizmeti, değişim sürecinin başarısını yakından etkilemektedir.

4.1.2.3. Organizasyonel Faktörler

Scrum dönüşümünü etkileyen organizasyonel faktörler altında:

- Müşterilerin ve iş birimlerinin dönüşüme katılması ve dönüşüm sürecinde müşteri ile iş birliğinde bulunulabilmesi,
- Değişim sürecine üst yönetimin destek olması,
- Scrum dönüşümü ile birlikte organizasyon kültürde istenilen değişimin sağlanabilmesi,
- Değişimin kısıtlı kalmasından ziyade tüm organizasyona yayılabilmesi,

gibi çeşitli alt faktörler bulunmaktadır (Hameed vd., 2016: 85; Ozierańska, 2015: 84).

Bir işletmede Scrum dönüşümünün organizasyonun en üst kademesinden başlaması gerektiği için, üst yönetimin desteğinin sağlanması önemlidir. Üst yönetimin çevik prensiplerini benimsemek anlamında yetersiz kalması ve dönüşüm için yeterli desteği sağlamaması, çevik dönüşümün başarılı olmasının önündeki en büyük engellerden biridir (Gandomani ve Nafchi, 2015: 262). Üst yönetim desteği bulunmayan çevik dönüşümlerde beklenen başarıların sağlanması mümkün olmamakla birlikte, organizasyonun üst kademesinden itibaren başlayan çevik prensipleri benimsetme ve yaygınlaştırma faaliyeti takım seviyesine yayılmalı ve çalışanların çevik dönüşüme uyum sağlayabileceği ortamlar oluşturulmalıdır.

Organizasyonel faktör altındaki bir diğer nitelik ise, Scrum dönüşümü ile birlikte organizasyonel kültürün değiştirilebilmesidir. VersionOneInc (2013) tarafından

yayınlanan “Yedinci Yıllık - Çevikliğin Durumu” raporuna göre organizasyonel kültürü değiştirme yeteneği, Scrum dönüşümünü gerçekleştirmeye çalışan işletmelerin karşılaştığı en büyük problemlerin başında gelmektedir. Lindvall ve diğerlerine (2002) göre çevik uygulamalar müşteri işbirliğine ve geri bildirimine önem verdiğinden, organizasyon kültürünün değişikliklere uyarlanabilir olması gerekmektedir (Gansamoni, 2016: 258). Buna göre, organizasyonel kültürün, işbirliğine dayalı bir ortamda çalışmayı destekleyecek şekilde değişmesi gerekmektedir. Ayrıca, Scrum’ı uygulamak isteyen işletmeler hızlı iletişim kültürüne ve insanlara güvenin olduğu, takım kararlarının desteklendiği ve değişen müşteri gereksinimlerinin teşvik edildiği bir ortama uyum sağlamalıdır.

Organizasyonel faktörler altında incelenen son alt nitelik ise değişimin tüm organizasyona yayılmasıdır. Bir işletmede üst yönetim desteği ile Scrum dönüşümü gerçekleşiyorsa bu dönüşüm organizasyonun tamamında gerçekleşmelidir. Çünkü bir organizasyondaki ekipler birbirine bağlı olarak iş yapabilmekte ve sıklıkla bir projede farklı görev ve sorumluluklarda yer almaktadır. Eğer çevik dönüşümü organizasyonun tamamında gerçekleştirmek yerine, sadece belirli birimlerde uygulanmaya çalışılmak istenirse, dönüşümü gerçekleştirmek isteyen ekip, diğer ekiplere bağımlılığından dolayı gerçek anlamda çevik bir süreç olmayacağı gibi, çevik ekipler geleneksel yöntemlere göre çalışan ekipler nedeniyle birçok problemle karşılaşacaktır (Misra, 2009: 2).

BÖLÜM V

5.1.ARAŞTIRMA YÖNTEMİ

5.1.1. Anket İçeriği ve Anket Formunun Oluşturulması

Bu tez çalışmasında veri toplama yöntemi olarak anket çalışması uygulanmıştır. Anket soruları hazırlanırken Shahane (2014), Çetin (2016), VersionOne (2013), Paulk (2013) ve Scrum Alliance (2015) çalışmaları ile birlikte Scrum konusunda uzun süredir tecrübeleri bulunan uzman çevik koçların görüşlerinden yararlanılmıştır. Soruların bir kısmı içinde bulunulan zaman, anakütle özellikleri ve organizasyonların Scrum'a geçiş sürecini etkileyen faktörler incelenerek, katılımcıların bu faktörlerin geçiş sürecini etkilemesi ile ilgili düşünceleri göz önünde bulundurularak yeniden şekillendirilmiştir. Anket çalışması, Türkiye'de farklı sektörlerde bulunan işletmelerin yazılım geliştirme departmanlarında çalışan ve farklı mesleklerden olan 105 kişiye, Ocak 2018 – Şubat 2018 tarihleri arasında online olarak uygulanmıştır. Hazırlanan anket formu EK1'de verilmiştir.

Anketin ilk bölümünde katılımcılara, demografik bilgilerin toplanması amacıyla sorular yönetilmiştir. Bu bölümde katılımcılara eğitim düzeyi, cinsiyet, iş deneyimi, çalıştığı işyerinin bulunduğu sektör ve işyerinde görevi gibi sorular yöneltilerek katılımcılar hakkında demografik bilgiler toplanmıştır. Demografik bilgiler oluşturulurken Çetin (2016)'ın çalışmasından ağırlıklı olarak yararlanılmıştır. Ankette ilk bölümün sonunda sorulan soruyla birlikte, Scrum deneyimi olan ve olmayan katılımcılar ayrıştırılmıştır. Scrum tecrübesi bulunmayan katılımcılara sadece Scrum yöntemi hakkında düşünceleri sorulmuş olup, Scrum yönteminin başarısı ve çalıştıkları işletmelerde uygulanabilirliğine dair sorular sadece Scrum tecrübesi olan katılımcılara yöneltilmiştir.

Anketin ikinci bölümünde katılımcıların çevik yöntem tecrübesini ve geleneksel yöntem tecrübesini belirleyen sorular yönetilmiştir. Çevik yöntem tecrübesi araştırılırken katılımcıların hangi yöntemlerde ve ne kadar süre ile tecrübesinin olduğu soruları yönetilmiştir. Sorular oluşturulurken VersionOne (2013) araştırma anket çalışmasından yararlanılmıştır. Katılımcıların geleneksel ve çevik yöntem tecrübesini araştıran sorulardan sonra, katılımcılara Scrum yöntemi ile ilgili düşüncelerini tespit eden sorular

yöneltmiştir. Ayrıca katılımcılara, Scrum'ın uygulama derecesi ile işletmenin çevik dönüşümünün başarısını etkileyen faktörlerin gerçekleşme derecesini araştıran sorular da yöneltmiştir.

5.1.2. Örneklem Seçimi

Araştırmada anakütle, Türkiye’de farklı sektördeki firmaların yazılım departmanlarındaki çalışanlar olarak belirlenmiştir. Anakütle özelliklerini taşıyan örneklemin tespiti, Türkiye’de yazılım sektöründe çalışanların kısıtlı sayıda olması durumu ve bu konuda daha önce yapılmış herhangi bir araştırma bulunmadığı faktörleri de göz önünde bulundurularak, yargısal örnekleme ile gerçekleştirilmiştir. Buna göre, örneklem hacminin yeterli olup olmadığına veri toplama süreci içerisinde karar verilmiş olup, toplamda 105 kişiye ulaşılmıştır. Verilerin kodlanması, düzenlenmesi ve analizi Excel 2016 kullanılarak gerçekleştirilmiştir.

5.1.3. Anket Sonuçları

Anket, katılımcılara internet ortamından online olarak uygulanmıştır. Çalışma kapsamındaki katılımcılara ait bilgiler temel grafik ve tablolar halinde sunulmuştur.

Tablo 2’de belirtildiği üzere, ankete katılanların %41’i kadınlardan ve %59’u erkeklerden oluşmaktadır. Katılımcıların eğitim düzeyine bakıldığında, herhangi bir lisans programından mezuniyeti olanların oranının yüksek olduğu ve lisans düzeyi altında bir eğitim düzeyinin olmadığı görülmektedir.

Tablo 2. Katılımcıların Medeni Durum ve Eğitim Durumuna Göre Dağılımı

Değişken		Frekans (Kişi)	Yüzde (%)
Cinsiyet	Erkek	62	59,0
	Kadın	43	41,0
	Toplam	105	100
Eğitim Düzeyi	Üniversite	66	62,8
	Yüksek Lisans	37	35,2
	Doktora	2	2,0
	Toplam	105	100

Tablo 3 'de katılımcıların toplam deneyimleri ve son iş yerlerindeki deneyimleri birlikte verilmiştir. Ankete katılanların toplam iş deneyimine bakıldığında, katılımcıların ağırlıklı olarak 5 yıldan fazla süredir çalışmakta olduğu gözlemlenmektedir. Katılımcıların toplam iş deneyimlerinin yüksek çıkmasına rağmen, çoğunluğunun yakın dönemde iş değişikliği yaptığı göze çarpmaktadır. Buradan da, yazılım sektöründe iş değiştirme sıklığının yüksek olduğu bilgisi çıkarılabilir.

Tablo 3. Katılımcıların Deneyim Süresi Göre Dağılımı

Toplam deneyim süresi dağılımı	Mevcut İşteki Deneyim Süresi						
	Süre	1 yıldan az	1-3 yıldan az	3-5 yıldan az	5-10 yıldan az	10 yıl ve +	Toplam Kişi Sayısı
	1 yıldan az	1					1
	1-3 yıldan az	5	4				9
	3-5 yıldan az	4	10	6			20
	5-10 yıldan az	10	13	10	8		41
	10 yıl ve +	5	7	7	11	4	34
	Toplam Kişi Sayısı	25	34	23	19	4	105

Tablo 4 'de ankete katılanların meslek dağılımı yüzdesel olarak sunulmuştur. Ankete katılanlar yönetici, danışman, uzman ve diğer görev sınıfı olarak dört farklı görev kategorisi içerisinde gruplandırılmıştır. Yönetici grubu içerisinde, uygulama geliştirme yöneticisi, BT müdürü, takım lideri ve ürün sahibi; danışman grubu içerisinde, agile koç ve proje yöneticisi; uzman grubu içerisinde, yazılımcı, iş analisti ve test uzmanı; diğer görev sınıfı içerisinde ise akademisyenler, yönetici ortak ve ücret-yan haklar yetkilisi meslekleri şekilde gruplandırma yapılmıştır. Buna göre, ankete katılanların %66,7'si uzman, %17,1'i yönetici grubu içerisinde yer almaktadır. Ayrıca, ankete katılanların

%30,48'nin yazılımcı, %24,76'sının iş analisti pozisyonunda çalışan kişiler olduğu; geriye kalanının ise yukarıda belirtilen yönetici,danışman, uzman ve diğer kategoriler içerisinde belirtilen çeşitli pozisyonlarda olduğu tespit edilmiştir.

Tablo 4. Katılımcıların Mesleklerine Göre Dağılımı

Değişken	Pozisyon	Frekans (kişi)	Yüzde (%)
Katılımcıların görev/meslek dağılımı	Yönetici	18	17,1
	Danışman	14	13,3
	Uzman	70	66,7
	Diğer	3	2,9
	Toplam	105	100

Tablo 5 'te katılımcıların çalıştıkları sektöre ait dağılım sunulmuştur. Katılımcıların çalıştığı işletmelerde bilgi işlem departmanı çalışan sayısı; 101-500 arasında olanlar %28,6 ve 501-1000 arasında olanlar %27,6 ile birbirine en yakın gruplar olarak tespit edilmiştir. Bilgi işlem departmanı çalışan sayısı 50'den az kişi bulunanların oranı ise %13,3 olarak bulunmuştur. Buna göre, araştırmaya katılanların yaklaşık %77,2'sinin bilgi işlem departmanındaki çalışan sayısı 100'ün üstünde olan işletmelerden olduğu tespit edilmiştir.

Tablo 5. İşletmelerin Bilgi İşlem Departmanı Çalışan Sayısı Dağılımı

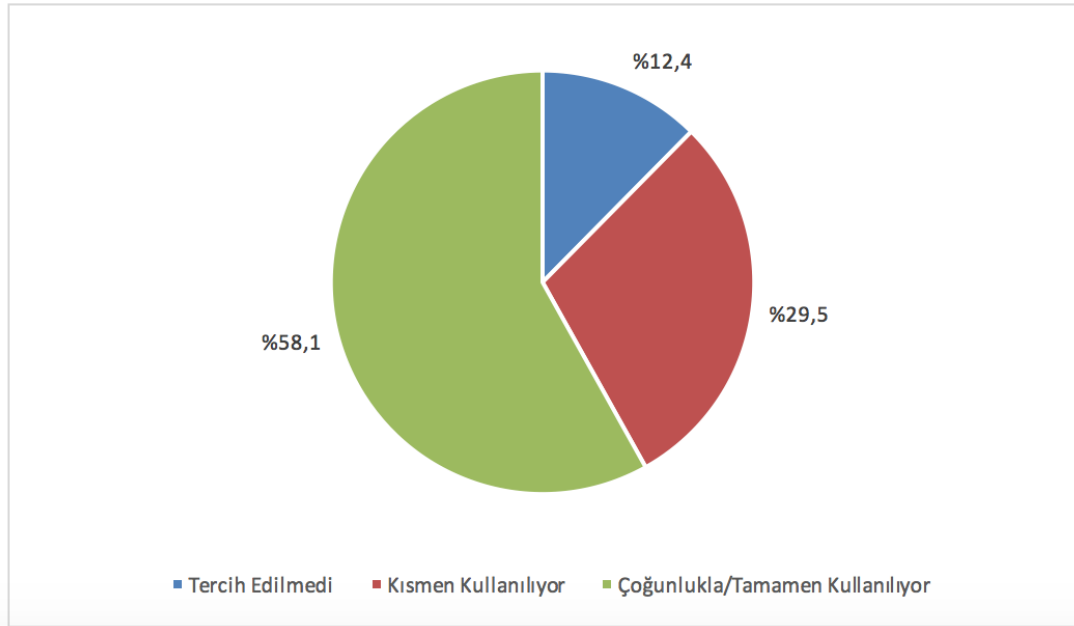
Değişken	Çalışan Sayısı (kişi)	Frekans (kişi)	Yüzde (%)
Katılımcıların çalıştığı işletmelerde bilgi işlem departmanı çalışan sayısı	<50	14	13,3
	50-100	10	9,5
	101 - 500	30	28,6
	501-1000	29	27,6
	>1000	22	21,0
	Toplam	105	100

Tablo 6 'da ankete katılanların çalıştıkları sektöre göre dağılımları sunulmuştur. Buna göre, araştırmaya en fazla katılımın %50,5 ile finans sektöründen olduğu görülmektedir. Katılanların %31,4'ünün ise iletişim - bilgi teknolojileri sektöründen olduğu tespit edilmiştir.

Tablo 6. Katılımcıların Çalıştığı Sektör Dağılımı

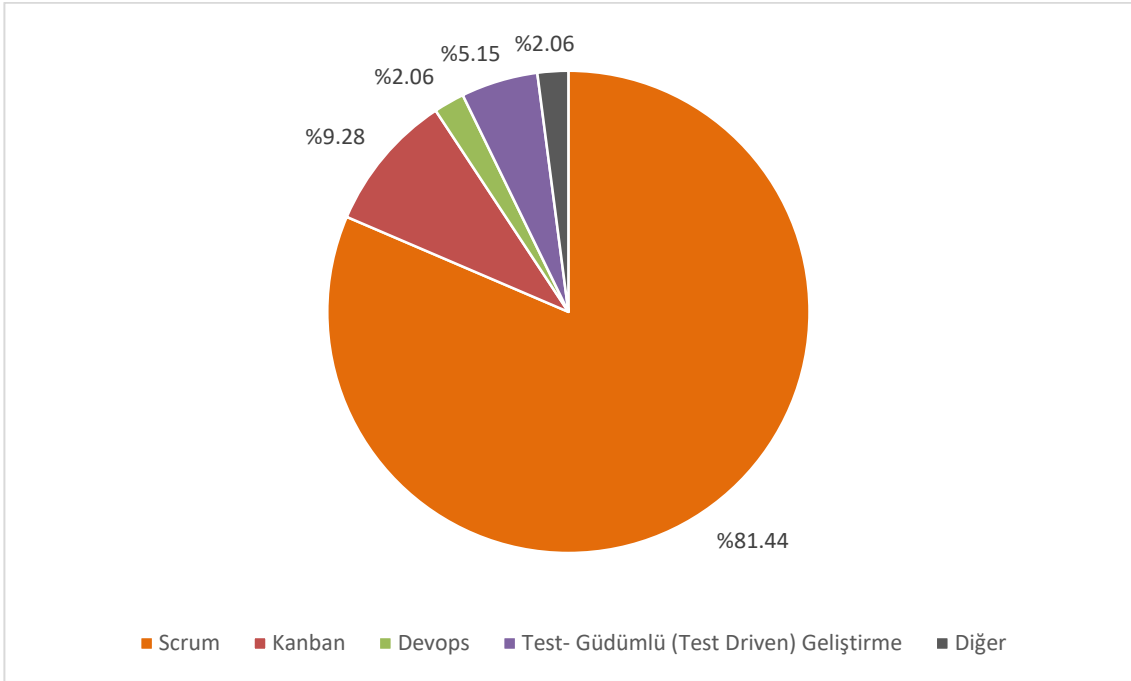
Değişken	Sektör	Frekans (kişi)	Yüzde (%)
Katılımcıların çalıştığı işletmelerin faaliyet gösterdiği sektör	Bankacılık - Finans	53	50,5
	İletişim - Bilgi	33	31,4
	Diğer	19	18,1
	Toplam	105	100

Şekil 10'da ise katılımcıların çalıştığı işletmelerin çevik yöntem tercihinine ilişkin dağılım sunulmuştur. Ankete katılanların büyük bir çoğunluğunun işletmelerinde çevik yöntem tecrübesinin olduğu tespit edilmiş olup, 13 kişinin çalıştıkları işletmelerde çevik yöntemin tercih edilmediği; bunların içerisinde 8 kullanıcının ise çalıştığı işletmelerde çevik yöntem tecrübesinin hiç olmadığı tespit edilmiştir.



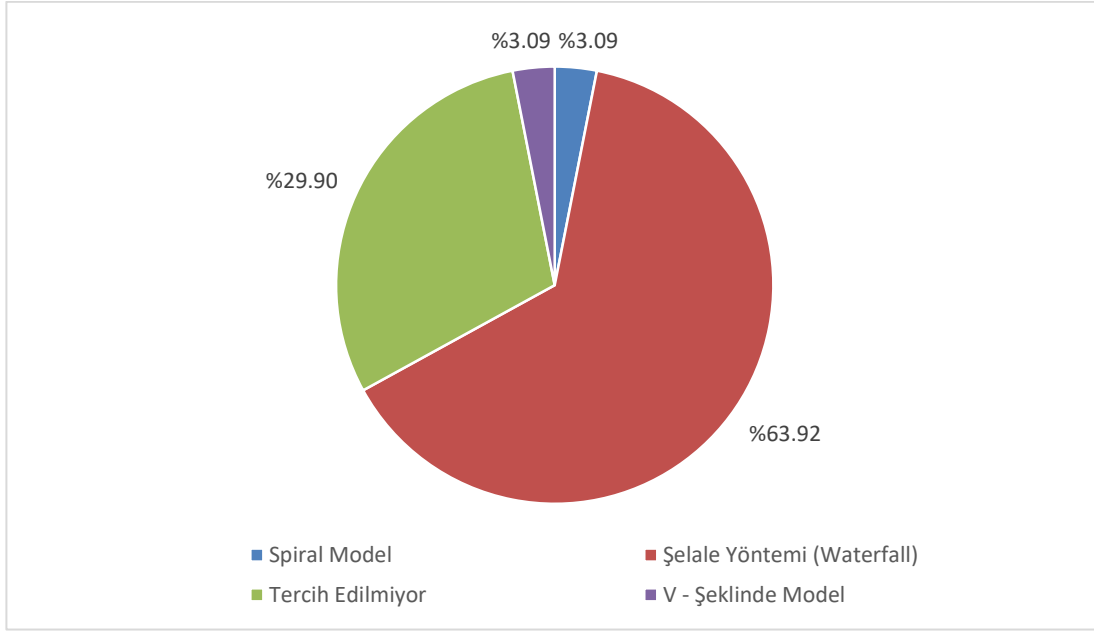
Şekil 10. Çevik Yöntemin Tercih Edilme Dağılımı

Şekil 11’de çevik yöntemin tercih edildiği işletmelerde çevik yöntemlerden hangisinin kullanıldığına dair dağılım sunulmaktadır. Çevik yöntemlerden en fazla tercih edilenin %81,4 ile Scrum olduğu açıkça gözlemlenmektedir. Ayrıca, Scrum’ın dışında Kanban ve Test-Güdümlü geliştirme yöntemlerinin de tercih edildiği tespit edilmiş olup, bunların oranının düşük olduğu görülmektedir.



Şekil 11. Çevik Yöntem Kullanım Dağılımı

Şekil 12’de ise geleneksel yöntemlerin kullanım dağılımı sunulmuştur. Buna göre, geleneksel yöntemlerden en fazla tercih edilen %63,9 ile şelale yöntemi olup, ankete katılanların %29,9’nın çalıştığı işletmelerde hiçbir geleneksel yöntemin kullanılmadığı tespit edilmiştir. Ayrıca, “Çalıştığınız işletmede yazılım projelerinde çevik yöntemler tercih edilmekte midir?” sorusuna, “Hiç tercih edilmedi” şeklinde cevap verenlerin tamamında, tercih edilen geleneksel yöntemin şelale yöntemi olduğu sonucuna ulaşılmıştır.



Şekil 12. Geleneksel Yöntem Kullanım Dağılımı

Tablo 7 'de ankete katılanların çalıştıkları işletmelerde çevik yöntemlerin ne kadar süredir kullanıldığına ilişkin dağılım sunulmuştur. Ankete katılanların çalıştıkları işletmelerde çevik yöntem kullanım süresinin en fazla %58,7 oran ile 1 - 3 yıl arasında olduğu tespit edilmiştir. Bu durum, işletmelerin büyük bir kısmının çevik yönetime geçişlerinin son 3 yıl içerisinde gerçekleştiğini göstermektedir.

Tablo 7. Çevik Yöntem Kullanım Süresi Dağılımı

Değişken	Süre	Frekans (kişi)	Yüzde (%)
Çevik yöntemler ne kadar süredir kullanılıyor?	1 -3 yıl arasında	57	58,8
	1 yıldan az	12	12,4
	3 - 5 yıl arasında	11	11,3
	5 yıldan fazla	10	10,3
	Bilgim yok*	7	7,2
	Toplam	105	100

Not: * Çevik yöntemler ne kadar süredir kullanılıyor sorusuna, “bilgim yok” diyen katılımcıları ifade etmektedir.

Tablo 8 'de çevik yöntemlerin işletme içerisinde kullanım yaygınlığına ait dağılım sunulmuştur. Ankete katılanların %39'unun çalıştığı işletmede ekiplerin tamamının / neredeyse tamamının Scrum yöntemini tercih ettiği tespit edilmiştir.

Tablo 8. İşletmede Çevik Yöntem Kullanım Yaygınlığı Dağılımı

Değişken	Oran	Frekans (kişi)	Yüzde (%)
Yazılım geliştirme yapan takımların ne kadarı çevik yöntemi kullanıyor?	Tamamı / Neredeyse tamamı	41	39,0
	Yarı yarıya	9	8,6
	Yarıdan daha azı	29	27,6
	Yarıdan daha fazlası	18	17,1
	Cevapsız	8	7,7
	Toplam	105	100

Anket çalışmasında, katılımcılar hakkında demografik bilgilerin yanı sıra; katılımcıların Scrum hakkında düşünceleri ve Scrum'ı benimseme derecelerini anlamaya yönelik sorular da sorulmuştur. Sorulan sorularda, Scrum yöntemindeki dökümantasyon yeterliliği ve yöntemin uygulanmasının büyük işletmeler için çok uygun olmadığı gibi sorunlar üzerine odaklanılmıştır. Konu hususunda, ankete katılanlar tarafından verilen cevaplar katılımcıların Scrum tecrübesinin olması ve olmamasına göre ayrı ayrı değerlendirilmiştir.

Tablo 9'de yazılım geliştirme süreçlerinde Scrum kullanmayanların Scrum hakkında görüşlerini belirten insan faktörleri dağılımı sunulmuştur. Scrum kullanmayan takımlar, "Scrum yöntemini kullanmak daha düşük kalitede yazılımlar üretilmesine neden olur" ifadesine %84,62 ve "Scrum'ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler" ifadesine ise %80,77 oranında katılmıyorum/kesinlikle katılmıyorum cevaplarını vermiştir. Benzer şekilde, "Scrum'da takımlar yeterli analiz yapmazlar" ifadesine %84,62 ve "Scrum'da takımlar yeterli mimari tasarım yapmazlar" ifadesine ise %88,46 oranında katılmıyorum/kesinlikle katılmıyorum cevaplarının ağırlıklı olarak ön plana çıktığı görülmektedir.

Tablo 9. Scrum Kullanmayan Çalışanların Scrum Hakkında Görüşleri

Sorular		1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum	Toplam
Scrum yöntemini kullanmak daha düşük kalitede yazılımlar üretilmesine neden olur.	Frekans (kişi)	9	13	2	2	0	26
	Yüzde (%)	34,62	50,00	7,69	7,69	0,00	100
Scrum'ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler.	Frekans (kişi)	8	13	2	3	0	26
	Yüzde (%)	30,77	50,00	7,69	11,54	0,00	100
Scrum'da takımlar yeterli dökümantasyon yapmazlar.	Frekans (kişi)	5	8	4	7	2	26
	Yüzde (%)	19,23	30,77	15,38	26,93	7,69	100
Scrum'da takımlar yeterli analiz yapmazlar.	Frekans (kişi)	6	15	1	3	1	26
	Yüzde (%)	23,08	57,69	3,85	11,54	3,85	100
Scrum'da takımlar yeterli mimari tasarım yapmazlar.	Frekans (kişi)	6	17	1	2	0	26
	Yüzde (%)	23,08	65,38	3,84	7,69	0,00	100
Scrum yönteminde, yazılım geliştiren takımlar geleneksel yöntemlerle çalışan takımlara göre daha disiplinsiz çalışır.	Frekans (kişi)	6	16	1	3	0	26
	Yüzde (%)	23,08	61,54	3,85	11,54	0,00	100
Scrum yönteminde, yazılım geliştirme maliyetleri daha yüksek olur.	Frekans (kişi)	5	15	3	3	0	26
	Yüzde (%)	19,23	57,69	11,54	11,54	0,00	100
Scrum yönteminin tüm organizasyonda uygulanma düşüncesi, işletmeler için sadece geçici bir hevestir.	Frekans (kişi)	6	10	4	4	2	26
	Yüzde (%)	23,08	38,46	15,38	15,38	7,69	100
Scrum yöntemi sadece küçük organizasyonlar için uygundur.	Frekans (kişi)	3	14	4	3	2	26
	Yüzde (%)	11,54	53,85	15,38	11,53	7,69	100
Scrum yönteminde projeleri yönetmek, geleneksel yöntemlere göre daha zor ve karmaşıktır.	Frekans (kişi)	6	12	6	2	0	26
	Yüzde (%)	23,08	46,15	23,08	7,69	0,00	100

Tablo 10’da ise, yazılım geliştirme süreçlerinde Scrum kullananların, Scrum hakkında görüşlerini belirten insan faktörleri dağılımı sunulmuştur. Yazılım geliştirme süreçlerinde Scrum tercih eden işletmelerde çalışanlar, “Scrum yöntemini kullanmak daha düşük kalitede yazılımlar üretilmesine neden olur” ifadesine %79,75 ve “Scrum’ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler” ifadesine ise %75,95 oranında katılmıyorum/kesinlikle katılmıyorum şeklinde cevap vermiştir. Benzer şekilde, “Scrum’da takımlar yeterli analiz yapmazlar” ifadesine katılmayanların oranı %68,35 iken, “Scrum’da takımlar yeterli mimari tasarım yapmazlar” ifadesine %70,89 oranında katılmıyorum/kesinlikle katılmıyorum şeklinde cevap vermiştir.

Tablo 10. Scrum dönüşümü - İnsan Faktörleri

Sorular		1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum	Toplam
Scrum yöntemini kullanmak daha düşük kalitede yazılımlar üretilmesine neden olur.	Frekans (kişi)	27	36	9	6	1	79
	Yüzde (%)	34,18	45,57	11,39	7,59	1,27	100
Scrum’ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler.	Frekans (kişi)	23	37	6	11	2	79
	Yüzde (%)	29,11	46,84	7,59	13,92	2,54	100
Scrum’da takımlar yeterli dökümantasyon yapmazlar.	Frekans (kişi)	12	27	10	29	1	79
	Yüzde (%)	15,19	34,18	12,66	36,71	1,27	100
Scrum’da takımlar yeterli analiz yapmazlar.	Frekans (kişi)	19	35	12	12	1	79
	Yüzde (%)	24,05	44,30	15,19	15,19	1,26	100
Scrum’da takımlar yeterli mimari tasarım yapmazlar.	Frekans (kişi)	17	39	12	8	3	79
	Yüzde (%)	21,52	49,37	15,19	10,12	3,80	100
Scrum yönteminde, yazılım geliştiren takımlar geleneksel yöntemlerle çalışan takımlara göre daha disiplinsiz çalışır.	Frekans (kişi)	25	37	7	7	3	79
	Yüzde (%)	31,65	46,84	8,85	8,86	3,80	100

Scrum yönteminde, yazılım geliştirme maliyetleri daha yüksek olur.	Frekans (kişi)	16	39	13	9	2	79
	Yüzde (%)	20,25	49,37	16,46	11,39	2,53	100
Scrum yönteminin tüm organizasyonda uygulanma düşüncesi, işletmeler için sadece geçici bir hevestir.	Frekans (kişi)	19	33	12	9	6	79
	Yüzde (%)	24,05	41,77	15,19	11,39	7,60	100
Scrum yöntemi sadece küçük organizasyonlar için uygundur.	Frekans (kişi)	16	30	15	10	8	79
	Yüzde (%)	20,25	37,97	18,99	12,66	10,13	100
Scrum yönteminde projeleri yönetmek, geleneksel yöntemlere göre daha zor ve karmaşıktır.	Frekans (kişi)	17	36	13	8	5	79
	Yüzde (%)	21,52	45,57	16,46	10,13	6,32	100

Tablo 11 'de yazılım geliştirme süreçlerinde Scrum kullanan ekiplerin, süreç ve uygulama faktörlerini gerçekleştirme derecelerini gösteren dağılım sunulmuştur. Tablo 11 'de gösterildiği üzere, ankete katılanların yaklaşık %80'i Scrum yöntemini kullanmanın verimliliği arttırmadığını ve bu durumun maliyetleri azaltmadığını belirtse de, ankete katılanların yarısından fazlası Scrum ile işlerin daha kısa sürelerde tamamlandığını ve bunun müşteri memnuniyetini arttırdığını, müşteri ihtiyaçlarına daha hızlı yanıt verilebildiğini belirtmektedir. Bununla birlikte, Scrum yöntemini kullanan işletmelerde üretilen yazılım ve ürünlerin kalitesinde artış sağlandığına ve üretim ortamında karşılaşılan hata sayısının azaldığına dair net bir kanaate ulaşılamamıştır.

Tablo 11. Scrum dönüşümü - Süreç ve Uygulama Faktörleri

Sorular		1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum	Toplam
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak verimliliğimizi artırdı.	Frekans (kişi)	27	36	9	6	1	79
	Yüzde (%)	34,18	45,57	11,39	7,59	1,27	100
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanarak, işleri daha kısa sürede bitirebiliyoruz.	Frekans (kişi)	7	12	16	40	4	79
	Yüzde (%)	8,86	15,19	20,25	50,63	5,07	100
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak, yazılım geliştirme maliyetlerini azalttı.	Frekans (kişi)	5	17	30	25	2	79
	Yüzde (%)	6,33	21,52	37,98	31,65	2,53	100
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak, üretilen yazılım ve ürünlerin kalitesini artırdı.	Frekans (kişi)	5	14	24	30	6	79
	Yüzde (%)	6,33	17,72	30,38	37,97	7,59	100
Scrum yöntemi ile geliştirdiğimiz projelerde, üretim ortamında daha az hata ile karşılaşmaktadır.	Frekans (kişi)	7	17	29	23	3	79
	Yüzde (%)	8,86	21,52	36,71	29,11	3,80	100
Scrum yöntemini kullanmak, çıkan ürünlerle ilgili müşteri memnuniyetini artırdı.	Frekans (kişi)	5	10	20	37	7	79
	Yüzde (%)	6,33	12,66	25,32	46,84	8,85	100
Scrum yöntemini kullanarak, değişen müşteri ihtiyaçlarına daha hızlı yanıt verebildiğimizi düşünüyorum.	Frekans (kişi)	5	4	8	46	16	79
	Yüzde (%)	6,33	5,06	10,13	58,23	20,25	100
İşletmenizde Scrum uygulamasının başarılı bir şekilde gerçekleştiğini/gerçekleşmekte olduğunu düşünüyorum.	Frekans (kişi)	12	12	22	27	6	79
	Yüzde (%)	15,19	15,19	27,85	34,18	7,59	100
İşletmenizde Scrum yönteminin tercih edilmeye devam edileceğini düşünüyorum.	Frekans (kişi)	7	6	18	35	13	79
	Yüzde (%)	8,86	7,59	22,78	44,30	16,47	100
İşletmenizde Scrum yöntemini tercih etmenin, proje başarı yüzdesinde artış sağladığını düşünüyorum.	Frekans (kişi)	6	10	20	35	8	79
	Yüzde (%)	7,59	12,66	25,32	44,30	10,13	100

Tablo 12 'de yazılım geliştirme süreçlerinde Scrum kullanan ekiplerin, organizasyonel faktörleri uygulama derecelerini gösteren dağılım sunulmuştur. Katılımcıların yarısından fazlası çalıştıkları işletmelerde gerçekleşen Scrum dönüşümü sürecine, yeterli üst yönetim desteğinin olduğunu belirtmiştir. Çevik dönüşümü sonrası işletmedeki hızlı iletişim, güven, takım kararlarının desteklenmesi ve değişen müşteri gereksinimlerinin teşvik edildiği organizasyon kültüründe yeterli değişimin olup / olmadığı kanaatinin net olmadığı sonucu ortaya çıkmıştır. Benzer şekilde, ankete katılanların yarısından fazlasında çevik dönüşümün tüm işletmeye yayılamadığı fakat dönüşümü sağlayan takımların kendi organizasyonlarını sağlayabildiği ve Scrum disiplinine bağlı kaldıkları sonucuna ulaşılmıştır. Bununla birlikte, işletme içinde takımlara Scrum uygulamaları konusunda yeterli düzeyde eğitim sağlanamadığı, fakat geliştirme takımlarının yeterli donanım, bilgiye sahip olduğu ve takım üyelerinin Scrum uygulamaları konusunda istekli ve değişime açık olduğu tespit edilmiştir. Ayrıca, çevik dönüşüm gerçekleşen işletmelerde Scrum'ın uygulanması sürecinde karşılaşılan problemlerle ilgili danışılabilir uzmanların bulunduğu sonucuna da ulaşılmıştır.

Tablo 12. Scrum dönüşümü - Organizasyonel faktörler

Sorular		1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum	Toplam
İşletmenizde Scrum dönüşümü sürecinde, üst yönetim dönüşüme her zaman destekçi olmuştur ve takımlarının dönüşümünü sürekli desteklemektedir.	Frekans (kişi)	5	9	18	36	11	79
	Yüzde (%)	6,33	11,39	22,78	45,57	13,93	100
Scrum sonrası, işletmenizde organizasyon kültürü; hızlı iletişim kültürüne, insanlara güvenin olduğu, takım kararlarının desteklendiği, değişen müşteri gereksinimlerinin teşvik edildiği şekilde değişim göstermiştir.	Frekans (kişi)	10	10	26	28	5	79
	Yüzde (%)	12,66	12,66	32,91	35,44	6,33	100

Scrum dönüşüm sürecinde işletmenizdeki tüm takımlar dönüşümü gerçekleştirmiştir / gerçekleştirmeye devam etmektedir.	Frekans (kişi)	8	23	12	26	10	79
	Yüzde (%)	10,13	29,11	15,19	32,91	12,66	100
İşletmenizdeki Scrum takımlarında, takımlar kendi organizasyonunu sağlayabilmektedir	Frekans (kişi)	5	10	21	35	8	79
	Yüzde (%)	6,33	12,66	26,58	44,30	10,13	100
İşletmenizdeki takımlar Scrum uygulamaları konusunda yeterli donanım ve bilgiye sahiptirler.	Frekans (kişi)	4	12	21	35	7	79
	Yüzde (%)	5,06	15,19	26,58	44,30	8,87	100
İşletmenizde çalışanlar, Scrum disiplinlerine bağlı kalmaktadırlar. (Örnek: etkinliklerine zamanında katılıma özen göstermekte ve katılamama durumunda erkenden bilgilendirme yapmaktadır.)	Frekans (kişi)	4	12	12	44	7	79
	Yüzde (%)	5,06	15,19	15,19	55,70	8,86	100
İşletmenizde Scrum uygulamaları konusunda takım üyeleri isteklidirler ve değişime açıktırlar.	Frekans (kişi)	5	9	18	42	5	79
	Yüzde (%)	6,33	11,39	22,78	53,16	6,34	100
İşletmenizde Scrum yöntemi ile ilgili yeterli düzeyde eğitim sağlanmakta ve sağlanmaya devam etmektedir.	Frekans (kişi)	6	11	13	35	14	79
	Yüzde (%)	7,59	13,92	16,46	44,30	17,73	100
İşletmenizde Scrum etkinlikleri (Sprint planlama, değerlendirme, gözden geçirme vs. toplantıları) etkin ve verimli şekilde yapılabilmektedir.	Frekans (kişi)	8	9	16	37	9	79
	Yüzde (%)	10,13	11,39	20,25	46,84	11,39	100
İşletmenizde Scrum uygulanması sürecinde karşılaştığınız sıkıntılarla ilgili danışabileceğiniz uzmanlar bulunmaktadır.	Frekans (kişi)	5	9	14	37	14	79
	Yüzde (%)	6,33	11,39	17,72	46,84	17,72	100

Tablo 13’de, ankete katılanların çalıştıkları işletmelerin bulunduğu sektöre ve işletmedeki çevik yöntem tercihi dağılımına bakıldığında, çevik yöntem tercihinin ağırlıklı olarak bankacılık ve finans sektöründe olduğu gözlemlenmektedir. İletişim ve bilgi teknolojileri ve diğer sektörlerde çevik yöntem tercihinin bankacılık ve finans sektörüne göre daha düşük olduğu sonucuna ulaşılmıştır.

Tablo 13. Sektörlere Göre Çevik Yöntem Tercihi

Yazılım Projelerinde Çevik Yöntemler Tercihi ve faaliyet gösterilen Sektör		Sektör			Toplam
		Bankacılık ve Finans	İletişim ve Bilgi Teknolojileri	Diğer	
Çevik Yöntem Tercihi	Hayır	5	3	3	11
	Kısmen Evet	13	13	7	33
	Evet	35	17	9	61
	Toplam	53	33	19	105

Tablo 14’de, yazılım projelerinde çevik yöntem kullanım süresi ve faaliyet gösterilen sektör dağılımına bakıldığında, çevik yöntem tercihinin tüm sektörlerde ağırlıklı olarak 1-3 yıl arasında tercih edildiği görülmektedir.

Tablo 14. Sektöre Göre Çevik Yöntem Kullanım Süresi Dağılımı

Yazılım Projelerinde çevik yöntem kullanım süresi ve faaliyet gösterilen sektör		Sektör			Toplam
		Bankacılık ve Finans	İletişim ve Bilgi Teknolojileri	Diğer	
Çevik Yöntem Kullanım Süresi	1 yıldan az	2	9	1	12
	1 - 3 yıl arasında	31	15	11	57
	3 - 5 yıl arasında	10	4	1	15
	5 yıldan fazla	5	2	1	8
	Bilgim yok*	6	2	5	13
	Toplam	54	32	19	105

Tablo 15’de, yazılım geliştirme yapan takımlarda çevik yöntem kullanan oranının sektöre göre dağılımına bakıldığında, bankacılık ve finans sektöründe yazılım geliştirme ekiplerinin daha yüksek oranda çevik yöntemleri kullandığı görülmektedir. Bankacılık ve finans sektöründe dönüşümün tamamen gerçekleşme oranı %50 civarındayken diğer sektörlerde dönüşüm oranının %50’nin altında kaldığı sonucuna ulaşılmıştır.

Tablo 15. Çevik Yöntem Kullanımının Sektöre Göre Sayıları

Sektöre ve Yazılım geliştirme yapan takımlarda çevik yöntem kullanıma göre adetleri		Sektör			Toplam
		Bankacılık ve Finans	İletişim ve Bilgi Teknolojileri	Diğer	
Yazılım geliştirme yapan takımlarda çevik yöntem kullanımı	Hiç kullanmayan	5	3		8
	Yarıdan daha azı	11	13	5	29
	Yarı yarıya	2	4	3	9
	Yarıdan daha fazla	11	5	2	18
	Neredeyse tamamı / Tamamı	26	9	6	41
	Toplam	55	34	16	105

Tablo 16’da, işletmelerin bilgi işlem - yazılım geliştirme departmanındaki çalışan sayısına göre çevik yöntem tercihi dağılımına bakıldığında, çalışan sayısı 100’ün altında olan işletmelerde çevik yöntem tercihinin daha düşük olduğu görülmektedir. Çalışan sayısı 500’den fazla olan işletmelerde ise çevik yöntem tercihinin yüksek olduğu sonucuna ulaşılmıştır.

Tablo 16. Çalışan Sayısına Göre Çevik Yöntem Tercihi

Çalışan Sayısı ve Çevik Yöntem Tercihi		Çevik Yöntem Tercihi			Toplam
		Hayır	Kısmen Evet	Evet	
Bilgi İşlem - Yazılım geliştirme departmanındaki çalışan sayısı	100'den az	7	8	9	24
	101-500	2	12	16	30
	501+	2	13	36	51
	Toplam	11	33	61	105

İşletmelerin faaliyet gösterdikleri sektör ve çevik yöntem tercihi; faaliyet gösterilen sektör ve çevik yöntem kullanım süresi; faaliyet gösterilen sektör ve ilgili departmanlarda çevik yöntem kullanan personel sayısı arasında bir ilişki olup olmadığının tespit edilmesi amacıyla, kategorik değişkenlerde kullanılan ve parametrik olmayan bir istatistik test olan ki-kare bağımsızlık testi uygulanmıştır. Oluşturulan hipotezler ve elde edilen ki-kare skorlarıyla, hipotez test sonucu Tablo 17’de verilmiştir.

Tablo 17. Hipotez Testleri

Hipotezler	χ^2	Serbestlik Derecesi	Anlamlılık Seviyesi (p)*	Hipotez Kabul/Red
H_0 : Faaliyet gösterilen sektör ile yazılım projelerinde çevik yöntem tercihi arasında bir ilişki yoktur.	3,471	4	0,482	Kabul
H_0 : Faaliyet gösterilen sektör ile yazılım projelerinde çevik yöntem kullanım süresi arasında bir ilişki yoktur.	13,930	4	0,008	Red
H_0 : Faaliyet gösterilen sektör ile bilgi işlem - yazılım geliştirme departmanlarında çevik yöntem kullanan personel sayısı arasında bir ilişki yoktur.	9,268	6	0,159	Kabul
H_0 : Yazılım projelerinde çevik yöntem tercihi ile bilgi işlem - yazılım geliştirme departmanlarında çalışan sayısı arasında bir ilişki yoktur.	15,051	4	0,005	Red

***0,01 anlamlılık seviyesinde**

Uygulanan ki-kare test sonuçlarına göre, faaliyet gösterilen sektör ile yazılım projelerinde çevik yöntem tercihi arasında %99 güven düzeyinde anlamlı bir ilişki bulunamamıştır. Yani, tüm sektörler yazılım projelerinde çevik yöntem tercihi konusunda benzerlik göstermektedir. Faaliyet gösterilen sektör ile yazılım projelerinde çevik yöntem kullanım süresine bakıldığında, değişkenler arasında ilişki olduğu tespit edilmiştir. Buna göre, farklı sektörlerde çevik yöntem kullanım süreleri açısından farklılık bulunmaktadır. Ayrıca, işletmenin faaliyet gösterdiği sektör ile bilgi işlem - yazılım geliştirme departmanlarında çevik yöntem kullanan personel sayısı yani çevik yöntemi kullanan personel yaygınlığı arasında %99 güven aralığında bir ilişkinin olmadığı sonucuna ulaşılmıştır. Buradan, farklı sektörlerde yazılım projelerinde, çevik yöntem kullanan personel oranlarının benzerlik gösterdiği sonucu çıkarılabilir. Yazılım projelerinde çevik yöntem tercihi ile bilgi işlem - yazılım geliştirme departmanlarında çalışan sayısı arasında anlamlı bir ilişkinin olup olmadığına bakıldığında ise, H_0 hipotezinin reddedildiği; yani, %99 güven düzeyinde iki değişken arasında anlamlı bir ilişkinin olduğu sonucuna ulaşılmıştır. Buna göre, işletmelerdeki bilgi işlem - yazılım geliştirme departmanındaki çalışan sayısına göre çevik yöntem tercihinin değiştiği anlaşılmaktadır.

BÖLÜM VI

6.1. SONUÇ VE ÖNERİLER

Bu çalışmada, yazılım geliştirme süreçlerinde, çevik yöntemlerden Scrum'ı tercih eden işletmelerin çevik dönüşüm süreçlerini etkileyen faktörler ve bu faktörlerin işletmenin çevik dönüşüm sürecine olan etkileri, çeşitli sektörlerden çalışanlara anket uygulanarak araştırılmıştır. Araştırma sürecinde, işletmelerin çevik dönüşümlerinin birçok faktörden etkilendiği tespit edilmiş ve bu faktörler; insan, uygulama ve organizasyonel faktörler adı altında gruplanarak incelenmiştir. İnsan faktörü başlığı altında; çalışan yetkinlikleri, sorumlulukları ve çalışanların Scrum farkındalıkları değerlendirilmiştir. Uygulama faktörleri başlığı altında ise takımların eğitimleri, gelişimleri, sorumlulukları ve danışmanlık desteği alabilmeleri üzerinde durulmuştur. Son olarak organizasyonel faktörler başlığı altında, çevik dönüşüme üst yönetimin desteği, çevik değişimin tüm organizasyona yayılabilmesi ve dönüşümün organizasyonel kültürde yarattığı değişimler ele alınmıştır.

Uygulanan anket çalışmasında sektör çalışanlarına ilk olarak çevik dönüşümü etkileyen insan faktörünü ölçecek sorular yöneltilmiştir. Katılımcılara bu bölümde, “*Scrum yöntemini kullanmanın daha düşük kalitede yazılımlar üretilmesine neden olacağı*” ile ilgili görüşleri sorulmuştur. Scrum tecrübesi olan ve olmayan katılımcı grubunun her ikisi de ağırlıklı olarak bu görüşe katılmamıştır. Benzer şekilde katılımcılar çoğunluklu olarak “*Scrum'ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler*” görüşüne katılım göstermiştir. Bir başka soruda katılımcılara, “*Scrum'da yeterli dokümantasyon yapılmadığı*” ile ilgili düşünceleri sorulmuş olup, katılımcıların çoğunlukla bu görüşe de katılmadıkları sonucuna ulaşılmıştır. Dokümantasyonun en yoğun olduğu analiz ve mimari tasarım aşamasında katılımcıların “*Scrum'da takımlar yeterli analiz yapmazlar*” ve “*Scrum'da takımlar yeterli mimari tasarım yapmazlar*” ifadelerine de ağırlıklı olarak katılmadıkları görülmektedir. Ayrıca, anket sonuçlarına göre çalışanların Scrum yönteminde yazılım kalitesi ve planlama ile ilgili çekincelerinin bulunmadığı görülmüştür. Bu sonuçlara bakılarak, çalışanların Scrum yöntemi hakkında düşüncelerinin olumsuz olmadığı kanaatine ulaşılabilir.

Sonuç olarak, çalışanların Scrum farkındalığına ve Scrum yaklaşımlarına bakıldığında, katılımcıların ağırlıklı olarak Scrum metodolojisi hakkında olumlu görüş bildirdiği gözlemlenmiştir. Scrum hakkındaki olumlu yaklaşım belirtilen katılımcıların daha fazla bulunduğu ortamlarda Scrum dönüşümünden beklenen faydanın daha fazla sağlandığı ve dönüşüm sürecinin daha başarılı yürütüldüğü görüşü ön plana çıkmaktadır. Buna bağlı olarak, başarılı bir dönüşüm için takım üyelerinin Scrum'a inanma ve Scrum'ı gerçekleştirme konusunda istekli olmasının önemi ortaya çıkmaktadır.

Anket çalışmasının bir diğer kısmında katılımcılara çalıştıkları işletmede Scrum ilkelerinin uygulanma derecesini ölçecek sorular yöneltilmiştir. Çalışanların çoğunluğu, çalıştıkları işletmelerde Scrum uygulamaları konusunda yeterli düzeyde eğitim sağlandığını düşünmektedir. Benzer şekilde katılımcıların çoğu sprint planlama, sprint değerlendirme, sprint gözden geçirme gibi Scrum etkinliklerinin etkin ve verimli şekilde yapılabildiğini ve işletmede Scrum uygulanması sürecinde danışılabilen uzmanlar bulunduğu görüşüne katılmışlardır. Çevik dönüşümden vazgeçen veya kısmi dönüşümün yaşandığı işletmelerde söz konusu kriterlere katılımın daha düşük çıktığı gözlemlenmiştir. Bu durum çevik dönüşüm sürecinde Scrum'ı uygulamaya çalışan işletmelerin dönüşüm sürecindeki bu faktörlerden önemli derecede etkilendiğini göstermektedir.

Anketin son kısmında ise, katılımcılara çalıştıkları işletmede çevik dönüşümü etkileyen organizasyonel faktörlere dair sorular yöneltilmiştir. Ankete katılanlar ağırlıklı olarak Scrum uygulanma sürecine, yeterli üst yönetim desteğinin olduğunu belirtmişlerdir. Fakat çalışanlar çevik dönüşüm sonrası, çalıştıkları işletmede hızlı iletişim, güven, takım kararlarının desteklenmesi yönünde organizasyon kültüründe yeterli değişimin net olarak gerçekleşmediğini belirtmiştir. Bunun yanı sıra, dönüşümden vazgeçen veya kısmi dönüşümün yaşandığı işletmelerde, yönetim desteğinin daha düşük olduğu gözlemlenmiştir. Buradan yola çıkarak, dönüşümün başarısı için üst yönetimin desteğinin şart olduğu sonucuna bir kez daha ulaşılabilir. Sonuç olarak, katılımcıların çalıştıkları işletmelerde Scrum dönüşümü ile birlikte organizasyonel kültürde istenilen değişimlerin yeteri kadar sağlanamadığı hususu ön plana çıkmış olup, bu durumun çevik dönüşümden beklenen sonuçlara ulaşamamasına neden olduğu düşünülmektedir. Araştırmada, çevik

dönüşüme başlayan işletmelerin yarısından fazlasının çevik dönüşümü tüm işletmeye yayamadığı fakat buna rağmen dönüşümü sağlayan takımların kendi organizasyonlarını sağlayabildiği sonucu da çıkmıştır. Ayrıca işletmelerin bir kısmı dönüşümü tüm organizasyona yayabilmişken, bir kısmında dönüşümün sadece belirli takımlarda kısıtlı kaldığı belirtilmiştir. Bu sonuçlara bakılarak, Scrum dönüşümünün tüm işletmede uygulanmadığı ve işletme organizasyonunun buna uygun hale getirilmediği sürece, Scrum uygulamalarından istenilen verimin alınamadığı görülmektedir.

Ek olarak araştırma kapsamında, çevik dönüşümü başlatan işletmelerin bu dönüşümü ne ölçüde başarılı olarak gerçekleştirdikleri de araştırılmıştır. Araştırmaya katılan çalışanların çalıştığı işletmelerden %8’inde çevik dönüşüm sürecine hiç başlanmadığı sonucu çıkmıştır. Dönüşüm sürecine dahil olanlardan %3 gibi çok az oranda dönüşümün kesinlikle başarısızlıkla sonuçlandığı ve dönüşümün terk edildiği görülmektedir. Ayrıca, bu katılımcılardan gelen cevaplarda Scrum’a inancın ve Scrum’ın başarı getireceği düşüncesinin, diğer katılımcılara oranla daha düşük olduğu gözlemlenmektedir. Bir organizasyonda çalışanların Scrum hakkında olumlu düşüncelere sahip olması ve Scrum hakkında bilgi düzeylerinin yeterli olması, organizasyonlardaki Scrum dönüşümünü etkileyen kriterler olduğu görülmektedir.

Çevik yöntemlerin ağırlıklı olarak bankacılık ve finans sektöründe tercih edildiği, diğer sektörlerde çevik yöntem tercihinin bankacılık ve finans sektörüne göre daha düşük oranda çıktığı, çalışma kapsamında tespit edilen diğer sonuçlardan biridir. Araştırma sonucuna göre, yazılım geliştirme yapan takımlarda çevik yöntem kullanım oranının en yüksek bankacılık ve finans sektöründe olduğu sonucu istatistiksel olarak da tespit edilmiştir. Çevik yöntem tercihinin bankacılık ve finans sektöründe diğer sektörlerle göre yüksek çıkmasının nedeni, bu sektördeki müşteri ihtiyaçlarının daha yüksek hızda değişiyor olması olarak düşünülebilir. Bu düşünceden hareketle, bankacılık dışındaki sektörlerde müşteri ihtiyaçlarındaki değişim hızının bankacılık ve finans sektörüne göre daha düşük seviyelerde olduğu sonucuna varılabilir. İleriki çalışmalarda kullanıcılara müşteri ihtiyaçlarındaki değişim oranını ölçecek sorular yöneltilerek, sektör ile müşteri ihtiyaçları değişimi arasında bir ilişkinin var olup olmadığının araştırılması önerilebilir.

Ayrıca araştırma sonucunda çevik yöntemin tüm sektörlerde kullanım süresinin ağırlıklı olarak 1-5 yıl arasında olduğu tespit edilmiştir. Buradan, çevik yöntemlerin ülkemizde son 5 yıldır popülerliğinin arttığı ve kullanım alanının yaygınlaşmaya başladığı sonucuna ulaşılabilir. Araştırmada, yazılım geliştirme departmanındaki çalışan sayısının düşük olduğu işletmelerde çevik dönüşüm kullanımının daha düşük olduğu sonucu çıkmıştır. Bu sonuç, dönüşümün özellikle bankacılık ve finans sektörlerinde olduğu ve bu sektörlerdeki çalışan sayısının diğer sektörlerdeki çalışan sayılarından yüksek olmasına bağlanabilir. İlerleyen çalışmalarda, Scrum tercih etmeye başlayan işletmelerde kullanım tercihinin hangi faktörlerle ilişkili olduğu ve işletmenin çevik dönüşümden ne beklediğinin araştırılması önerilmektedir. Gelecekteki çalışmalarda, çalışma kısıtlarından biri olan örneklem sayısının arttırılarak, işletmelerde çevik dönüşüme etki eden faktörlerin daha büyük bir örneklemle tespit edilmesi ve çevik dönüşümün gerçekleşme derecesinin nitel ve nicel olarak tespit edilebilecek araştırma modellerinin geliştirilmesi önerilmektedir.

KAYNAKÇA

- Abdullah, E. ve Abdelsatir, E.-T. B. (2013). Extreme Programming Applied in a Large-scale Distributed System. *2013 International Conference On Computing*, 442-446.
- ACM Yazılım Çözümleri. (2017). Çevik Yazılım Geliştirme Agile. <http://www.acm-software.com/Pdf/AboutAgile.pdf> adresinden erişildi.
- Ahmad, M. O., Markkula, J. ve Ovio, M. (2013). Kanban in software development A systematic literature review. *39th Euromicro Conference Series on Software Engineering and Advanced Applications*, 9-16.
- Azizi, N. ve Taqi, M. A. (2015, Haziran). *Applying Agile methodologies within the context of traditional project governance*. (Yayımlanmamış yüksek lisans tezi). Jönköping University.
- Baytam, V. ve Kalıpsız, O. (2011). *Scrum Yazılım Geliştirme Metodolojisi için Yönetim Sistemi Tasarımı ve Gerçeklenmesi*. (Yayımlanmamış yüksek lisans tezi). Yıldız Teknik Üniversitesi, İstanbul.
- Benington, H. D. (1983). Production of large computer programs, *Annals of the History of Computing*, 5, 350-361.
- Cho, J. J. (2010). *An Exploratory Study on Issues and Challenges of Agile Software Development with Scrum*. (Yayımlanmamış doktora tezi). Utah State University, Utah.
- Chow, T. ve Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *The Journal of Systems and Software*, 81, 961-971.
- Çetin, E. (2016). *Yazılım Geliştirme için Harmanlanmış Scrum Modeli*. (Yayımlanmamış yüksek lisans tezi). Kocaeli Üniversitesi, Kocaeli.
- DeCarlo, D., 2004. Leading and managing extreme projects. <http://onlinelibrary.wiley.com/doi/10.1002/ltl.101/epdf> adresinden erişildi (18 Kasım 2018).
- Digite (2018). What is Kanban? <https://www.digite.com/kanban/what-is-kanban/> adresinden erişildi (12 Kasım 2018).
- Hundermark, P. (2009). Do Better Scrum. ScrumSense Report. <http://www.scrumsense.com> adresinden erişildi (29 Mart 2019).
- Dönmez, E. (2009). *Çevik yazılım geliştirme sürecinde kritik başarı faktörlerinin belirlenmesi ve önceliklendirmesine yönelik bir örnek çalışma*. (Yayımlanmamış yüksek lisans tezi). Bahçeşehir Üniversitesi, İstanbul.
- Ellis, G. (2016). Agile Project Management: Scrum, eXtreme Programming, and Scrumban. *Project Management in Product Development*, Butterworth-Heinemann, 223-260.

- Erdil, A. ve Erbiyık, H. (2013a, Ekim). Yalın Çevik (Agile) yaklaşımıyla yazılım geliştirme Scrum Uygulama Örnekleri. 6.Endüstri Mühendisliği Bahar Konferansları –Yalın Dönüşüm. <http://www1.mmo.org.tr/> adresinden erişildi (18 Mayıs 2018).
- Fowler, K. R. (2015). Developing and Managing Embedded Systems and Products.
- Gandomani, T. J. ve Nafchi, M. Z. (2016). Agile transition and adoption human-related challenges and issues A Grounded Theory approach. *Computers in Human Behavior*, 62, 257-266.
- Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M. ve Parizi, R. M. (2014). The impact of inadequate and dysfunctional training on Agile transformation process A Grounded Theory study. *Information and Software Technology*. <http://dx.doi.org/10.1016/j.infsof.2014.05.01> adresinden erişildi (20 Mayıs 2018).
- Gandomani, T. J., Zulzalil, H., Ghani, A. A. A., Sultan, A. B. M. ve Sharif, K. Y. (2014). How Human Aspects Impress Agile Software Development Transition and Adoption. *International Journal of Software Engineering and Its Applications*, 8(1), 129-148.
- Gencer, C. ve Kayacan, A. (2017). Yazılım Proje Yönetimi Şelale Modeli ve Çevik Yöntemlerin Karşılaştırılması. *Bilişim Teknolojileri Dergisi*, 10(3), 335-352.
- Hameed, T. A. E., Latif, M. A. E. ve Kholief, S. (2016). Identify and Classify Critical Success Factor of Agile Software Development Methodology Using Mind Map. *International Journal of Advanced Computer Science and Applications*, 8(5), 83-92.
- Judy, K. H. ve Krumins-Beens, I. (2008). Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership. *Proceedings of the 41st Hawaii International Conference on System Sciences*.
- Kır, G. (2014, Ağustos). *Scrum Adoption in Kuveyt Turk IT*. (Yayımlanmamış yüksek lisans tezi). Doğu Üniversitesi, İstanbul.
- Mandanis, G. (2015). *Agile Transformation : What Went Wrong ?* <https://www.linkedin.com/pulse/agile-transformation-what-went-wrong-pradeep-bindra/> adresinden erişildi. (23 Kasım 2017).
- Misra, Subhas Chandra. (2007, Nisan). *Adopting Agile Software Development Practices Success Factors, Changes Required, and Challenges*. (Yayımlanmamış doktora tezi). Carleton University, Ottawa.
- Nerur, S., Mahapatra, R. ve Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications Of The Acm*, 48(5), 72-78.
- Ozierańska, A., Skomr, A., Kuchta, D. ve Rola, P. (2016). The critical factors of Scrum implementation in IT project. *Journal of Economics and Management*, 25(3), 79-97.

- Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M. (2018). Large-scale agile transformation at Ericsson: a case study. *Empirical Software Engineering*, 23(5), 2550-2596.
- Paulk, D. M. C. (2009, Nisan). On Empirical Research into Scrum Adoption. 5000 Forbes Avenue.
- Paulk, M. C. (2013). Scrum adoption survey. *Software Quality Professional*, 15(2), 27-34.
- Pichler, R. (2010). *Agile Product Management with Scrum: Creating Products that Customers Love*. Addison-Wesley Professional.
- Pinto, J. K. ve Slevin, D. P. (1988). Project success definitions and measurement techniques. <https://www.pmi.org/learning/library/project-success-definitions-measurement-techniques-5460> adresinden erişildi (27 Kasım 2017).
- Pressman, R. S. (2005). *Software Engineering A Practitioner's Approach*. Seventh Edition.
- Schindler, C. (2008). Agile Software Development Methods and Practices in Austrian IT Industry: Results of an Empirical Study. Computational Intelligence for Modelling Control and Automation, 2008 International Conference, sunulmuş bildiri, Viyana.
- Schwaber, K. (2004). *Agile Project Management with Scrum*. Redmond, Washington.
- Schwaber, K. ve Sutherland, J. (1995). Business Object Design and Implementation. OOPSLA, sunulmuş bildiri, Texas.
- Schwaber, K. ve Sutherland, J. (2013, Temmuz). Scrum Kılavuzu™. *Scrum.org*.
- Scrum Alliance. (2018). *State of Scrum 2017-2018 Report*. <https://www.scrumalliance.org> adresinden erişildi (12 Mart 2018).
- Shahane, D., Jamsandekar, P. ve Shahane, D. (2014). Factors Influencing the Agile Methods in Practice Literature Survey & Review. *IEEE, 2014 International Conference on Computing for Sustainable Global Development*, 556-560.
- Silva, K. M. B. ve Santos, S. (2015). Critical Factors in Agile Software Projects According to People, Process, and Technology Perspective. *2015 6th Brazilian Workshop on Agile Methods*, 48-54.
- Şeker, S. E. (2015). Yazılım Geliştirme Modelleri ve sistem/Yazılım Yaşam Döngüsü. *YBS Ansiklopedi*, 2(3), 18-29.
- Thuy, D. H. ve Khanh, N. N. T. (2013). *Scrum's success factors and scrum's benefit levels to diffrent project and processes*. (Yayımlanmamış tez). Lathi University.
- VersionOne. (2013). *Seventh Annual State of Agile Development Survey*. <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf> adresinden erişildi (11 Şubat 2018).

VersionOne. (2016). *10th annual “state of agile development”* survey. <https://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf> adresinden erişildi.

Virginia Tech Courses (2018). Software Engineering The Spiral Model. <http://courses.cs.vt.edu/csonline/SE/Lessons/Spiral/index.html> adresinden erişildi (23 Şubat 2018).

ÖZGEÇMİŞ

ERHAN ÖZLÜ

Emek Mah. 67. Sok Göktürk sitesi evleri A6 / D.12 Çayırova/Kocaeli

E-mail: ozlue88@gmail.com

KİŞİSEL BİLGİLER

Doğum Tarihi : 25/03/1988
Doğum Yeri : Türkiye – Muğla
Medeni durumu : Evli

EĞİTİM

2015- İstanbul Medeniyet Üniversitesi , İşletme (Tezli)
2006- 2011 İstanbul Teknik Üniversitesi , Endüstri Mühendisliği
2002-2006 Köyceğiz Anadolu Lisesi, Fen bilimleri

İŞ DENEYİMİ

09/2017 – : IBTECH - İş Analisti
09/2013 –09/2017 : AKBANK - İş Analisti Scrum Master
12/2010 – 09/2013 : IBTECH - İş Analisti
06/2010 – 12/2010 : IBTECH - Stajyer
06/2009 – 08/2009 : Finansbank A.Ş. - Stajyer
02/2009 – 06/2009 : Hüsnü Özyeğin Vakfı - Gönüllü Çalışan

SERTİFİKA VE EĞİTİMLERİM

- PSM - I (Professional Scrum Master) sertifikası, Scrum.org - 11.2015
- Scrum ile Agile Yazılım Geliştirme, BTAKADEMİ - 19.10.2014-21.10.2014
(16 Saat)

- ManagingProjectsInAgileWithScrum, BTAKADEMİ- 17.09.2014-19.09.2014 (16 Saat)
- Uluslararası Mobil Ekran Tasarımı ve Kullanılabilirlik Eğitim – UXservices – 14.02.2013-16.02.2013
- BAP - Uluslararası İş Analisti Eğitimi - BA-WORKS – 07.06.2012 – 09.06.2012
- Uygulamalı Süreç Yönetimi - Bankalar Birliği 26.03.2012 – 28.03.2012
- MS Visio - Bilge Adam –05.03.2012
- ISTQB Uluslararası Sertifikalı Yazılım Test Uzmanı(Foundation Level), BA-WORKS – 07.06.2012 – 09.06.2012

YABANCI DİL

İngilizce: İleri seviye

Almanca: Başlangıç seviye

BİLGİSAYAR

Microsoft Office (Word, Excel, Acces PowerPoint, Visio), SQL, İnternet, C programla, XML, SQL Server, HTML, TOAD

İLGİ ALANLARI

Spor (Masa tenisi, basketbol, yürüyüş, futbol), sinema, seyahat, bağlama

EKLER

EK1: ANKET FORMU

İstanbul Medeniyet Üniversitesi Lisansüstü Eğitim Enstitüsü İşletme bölümünde yürütülen yüksek lisans tezi kapsamında, Türkiye'de BT çalışanlarının Scrum yöntemine olan yaklaşımlarını, Scrum'ın tercih edildiği işletmelerde uygulanabilme başarısını ve Scrum'ın uygulama başarısını etkileyen faktörlerin gerçekleşme derecesini araştıran çalışmanın bir parçasıdır. Araştırma sonuçlarının sizinle de paylaşılmasını isterseniz e-posta adresime mail atmanız yeterli olacaktır. Değerli zamanınızı ayırdığınız için teşekkür ederiz.

1- Lütfen size uygun cevabı işaretleyiniz.

Cinsiyetiniz	Kadın ()	Erkek ()			
Eğitim seviyeniz?	Ortaöğrenim / Lise ()	Yüksek Okul ()	Üniversite ()	Yüksek Lisans ()	Doktora ()
Toplam iş deneyiminiz?	1 yıldan az ()	1 - 3 yıl arasında ()	3 - 5 yıl arasında ()	5 -10 yıl arasında ()	10 yıl ve üzerinde ()
Mevcut iş yerinizdeki deneyiminiz?	1 yıldan az ()	1 - 3 yıl arasında ()	3 - 5 yıl arasında ()	5 -10 yıl arasında ()	10 yıl ve üzerinde ()
Çalıştığınız işletmedeki göreviniz?	Takım Lideri ()	Proje Yöneticisi ()	BT Müdürü ()	Yazılımcı ()	Test Uzmanı ()
	Ürün Sahibi ()	Operasyon Destek ()	Diğer (lütfen belirtiniz)		
Çalıştığınız işletmenin faaliyette bulunduğu sektör?	Bankacılık - Finans ()	İletişim – Bilgi Teknolojileri ()	Diğer (lütfen belirtiniz)		
Çalıştığınız işletmenin Bilgi İşlem - Yazılım geliştirme departmanındaki çalışan sayısı nedir?	<50 ()	50-100 ()	101 - 500 ()	501-1000 ()	>1000 ()
Çalıştığınız işletmede yazılım projelerinde çevik yöntemler tercih edilmekte midir?	Tercih edilmedi ()	Kısmen kullanılıyor ()	Çoğunlukla / Tamamen kullanılıyor ()		

2- Katılımcı çevik yöntem tecrübesi

İşletmenizde yazılım geliştirme süreçlerinde çevik yöntemlerden hangisi tercih edilmektedir. (Birden fazla olması durumunda en çok tercih edileni seçiniz.)	Scrum ()	Kanban ()	Sınırsal Programlama (XP) ()	Test-Güdümlü (Test Driven) Geliştirme ()	Diğer____
İşletmenizde yazılım geliştirme süreçlerinde geleneksel yöntemlerden hangisi tercih edilmektedir. (Birden fazla olması durumunda en çok tercih edileni seçiniz.)	Tercih Edilmiyor ()	Şelale Yöntemi (Waterfall) ()	Spiral Model ()	V - Şeklinde Model ()	Diğer____
İşletmenizde çevik yöntemler ne kadar süredir kullanılıyor?	1 yıldan az ()	1 -3 yıl arasında ()	3 - 5 yıl arasında ()	5 yıldan fazla ()	Bilgim yok ()
İşletmenizde yazılım geliştirme yapan takımların ne kadarı çevik yöntemi kullanıyor?	Yarıdan daha azı ()	Yarı yarıya ()	Yarıdan daha fazlası ()	Tamamı / Neredeyse tamamı ()	

3- Katılımcı geleneksel yöntem tecrübesi

İşletmenizde yazılım geliştirme süreçlerinde geleneksel yöntemlerden hangisi tercih edilmektedir. (Birden fazla olması durumunda en çok tercih edileni seçiniz.)	Şelale Yöntemi (Waterfall)()	Sarmal (Spiral) Model ()	V - Şeklinde Model ()	Diğer____	
İşletmenizde çevik dönüşümün ne zaman başlayacağını düşünüyorsunuz?	1 yıldan kısa bir süre içinde ()	1 - 3 yıl arasında ()	3 yıldan uzun bir süre sonra ()	Hiçbir zaman ()	Fikrim yok ()
İşletmenizde çevik dönüşüme olan ihtiyaç ile ilgili düşünceleriniz nelerdir?	İhtiyaç bulunmuyor ()	Denenmesin de fayda olacaktır ()	Kesinlikle denenmelidir ()	Fikrim yok ()	

4- Scrum yöntemi ile ilgili aşağıdaki ifadeleri değerlendiriniz.

Lütfen bu soruları kendi düşünceleriniz doğrultusunda; 1-Kesinlikle Katılmıyorum, 2-Katılmıyorum, 3-Kararsızım, 4-Katılıyorum, 5-Kesinlikle Katılıyorum, seçeneklerinden birini tik (X) koyarak cevaplandırınız.	1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum
Scrum yöntemini kullanmak daha düşük kalitede yazılımlar üretilmesine neden olur.	()	()	()	()	()
Scrum'ı tercih eden takımlar yeterli planlama yapmazlar ve önlerini net olarak göremezler.	()	()	()	()	()
Scrum'da takımlar yeterli dökümantasyon yapmazlar.	()	()	()	()	()
Scrum'da takımlar yeterli analiz yapmazlar.	()	()	()	()	()
Scrum'da takımlar yeterli mimari tasarım yapmazlar.	()	()	()	()	()
Scrum yönteminde, yazılım geliştiren takımlar geleneksel yöntemlerle çalışan takımlara göre daha disiplinsiz çalışır.	()	()	()	()	()
Scrum yönteminde, yazılım geliştirme maliyetleri daha düşük olur. *	()	()	()	()	()
Scrum yönteminin tüm organizasyonda uygulanma düşüncesi, işletmeler için sadece geçici bir hevestir.	()	()	()	()	()
Scrum yöntemi sadece küçük organizasyonlar için uygundur.	()	()	()	()	()
Scrum yönteminde projeleri yönetmek, geleneksel yöntemlere göre daha zor ve karmaşıktır.	()	()	()	()	()

5- İşletmenizde Scrum'ın başarı kriterlerine etkileri ile ilgili aşağıdaki ifadeleri değerlendiriniz.

Lütfen bu soruları kendi düşünceleriniz doğrultusunda; 1-Kesinlikle Katılmıyorum, 2-Katılmıyorum, 3-Kararsızım, 4-Katılıyorum, 5-Kesinlikle Katılıyorum, seçeneklerinden birini tik (X) koyarak cevaplandırınız.	1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak verimliliğimizi arttırdı.	()	()	()	()	()
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanarak, işleri daha kısa sürede bitirebiliyoruz.	()	()	()	()	()
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak, yazılım geliştirme maliyetlerini azalttı.	()	()	()	()	()
Yazılım geliştirme süreçlerinde Scrum yöntemini kullanmak, üretilen yazılım ve ürünlerin kalitesini arttırdı.	()	()	()	()	()
Scrum yöntemi ile geliştirdiğimiz projelerde, üretim ortamında daha fazla hata ile karşılaşmaktadır.*	()	()	()	()	()
Scrum yöntemini kullanmak, çıkan ürünlerle ilgili müşteri memnuniyetini arttırdı.	()	()	()	()	()
Scrum yöntemini kullanarak, değişen müşteri ihtiyaçlarına daha hızlı yanıt verebildiğimizi düşünüyorum.	()	()	()	()	()
İşletmenizde Scrum uygulamasının başarılı bir şekilde gerçekleştiğini/gerçekleşmekte olduğunu düşünüyorum.	()	()	()	()	()
İşletmenizde Scrum yönteminin tercih edilmeye devam edileceğini düşünüyorum.	()	()	()	()	()
İşletmenizde Scrum yöntemini tercih etmenin, proje başarı yüzdesinde artış sağladığını düşünüyorum.	()	()	()	()	()
Eklemek istediğiniz görüşleriniz varsa, lütfen belirtebilir misiniz?					

6- Çevik dönüşümün başarısını etkileyen aşağıdaki faktörlerin, işletmenizde gerçekleşme derecesi ile ilgili aşağıdaki soruları cevaplayınız.

Lütfen bu soruları kendi düşünceleriniz doğrultusunda; 1-Kesinlikle Katılmıyorum, 2-Katılmıyorum, 3-Kararsızım, 4-Katılıyorum, 5-Kesinlikle Katılıyorum, seçeneklerinden birini tik (X) koyarak cevaplandırınız.	1- Kesinlikle Katılmıyorum	2- Katılmıyorum	3- Kararsızım	4- Katılıyorum	5- Kesinlikle Katılıyorum
İşletmenizde Scrum dönüşümü sürecinde, üst yönetim dönüşüme her zaman destekçi olmuştur ve takımlarının dönüşümünü sürekli desteklemektedir.	()	()	()	()	()
Scrum sonrası, işletmenizde organizasyon kültürü; hızlı iletişim kültürüne, insanlara güvenin olduğu, takım kararlarının desteklendiği, değişen müşteri gereksinimlerinin teşvik edildiği şekilde değişim göstermiştir.	()	()	()	()	()
Scrum dönüşüm sürecinde işletmenizdeki tüm takımlar dönüşümü gerçekleştirmiştir / gerçekleştirmeye devam etmektedir.	()	()	()	()	()
İşletmenizdeki Scrum takımlarında, takımlar kendi organizasyonunu sağlayabilmektedir.	()	()	()	()	()
İşletmenizdeki takımlar Scrum uygulamaları konusunda yeterli donanım ve bilgiye sahip değillerdir.*	()	()	()	()	()
İşletmenizdeki çalışanlar, Scrum disiplinlerine bağlı kalmaktadırlar. (örnek: etkinliklerine zamanında katılıma özen göstermekte ve katılamama durumunda erkenden bilgilendirme yapmaktadır.)	()	()	()	()	()
İşletmenizde Scrum uygulamaları konusunda takım üyeleri isteklidirler ve değişime açıktırlar.	()	()	()	()	()
İşletmenizde Scrum yöntemi ile ilgili yeterli düzeyde eğitim sağlanmakta ve sağlanmaya devam etmektedir.	()	()	()	()	()
İşletmenizde Scrum etkinlikleri (Sprint planlama, değerlendirme, gözden geçirme vs. toplantıları) etkin ve verimli şekilde yapılabilmektedir.	()	()	()	()	()
İşletmenizde Scrum uygulanması sürecinde karşılaştığınız sıkıntılarla ilgili danışabileceğiniz uzmanlar bulunmaktadır.	()	()	()	()	()
Ekleme istediğiniz görüşleriniz varsa, lütfen belirtebilir misiniz?					