

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**ÇEVİK YAZILIM GELİŞTİRME PROJELERİNDE KRİTİK BAŞARI
FAKTÖRLERİNİN MODELLENMESİ: ÇEVİK PROJELERİN
TÜRKİYE'DEKİ UYGULAMALARI**

DOKTORA TEZİ

Burcu BİNBOĞA

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

HAZİRAN 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**ÇEVİK YAZILIM GELİŞTİRME PROJELERİNDE KRİTİK BAŞARI
FAKTÖRLERİNİN MODELLENMESİ: ÇEVİK PROJELERİN
TÜRKİYE'DEKİ UYGULAMALARI**

DOKTORA TEZİ

**Burcu BİNBOĞA
(507152102)**

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Çiğdem ALTIN GÜMÜŞSOY

HAZİRAN 2024

İSTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**MODELLING CRITICAL SUCCESS FACTORS IN AGILE SOFTWARE
DEVELOPMENT PROJECTS: APPLICATIONS OF AGILE PROJECTS
IN TÜRKİYE**

Ph.D. THESIS

**Burcu BİNBOĞA
(507152102)**

Department of Industrial Engineering

Industrial Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Çiğdem ALTIN GÜMÜŞSOY

JUNE 2024

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 507152102 numaralı Doktora Öğrencisi Burcu BİNBOĞA, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “ÇEVİK YAZILIM GELİŞTİRME PROJELERİNDE KRİTİK BAŞARI FAKTÖRLERİNİN MODELLENMESİ: ÇEVİK PROJELERİN TÜRKİYE’DEKİ UYGULAMALARI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Çiğdem ALTIN GÜMÜŞSOY**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Başar ÖZTAYŞI**
İstanbul Teknik Üniversitesi

Dr. Belma ŞIPKA
Hitit Computer Services

Prof. Dr. Emre ÇEVİKCAN
İstanbul Teknik Üniversitesi

Prof. Dr. Hüseyin Selçuk KILIÇ
Marmara Üniversitesi

Teslim Tarihi : 16 Mayıs 2024
Savunma Tarihi : 4 Haziran 2024





Anneme, babama, abime ve anneanneme



ÖNSÖZ

Bu tezin hazırlanmasında bana yol gösteren saygıdeğer hocam Doç. Dr. Çiğdem ALTIN GÜMÜŞSOY'a en içten teşekkürlerimi sunarım. Aynı zamanda, değerli görüşleriyle bu çalışmaya katkıda bulunan saygıdeğer izleme komitesi üyeleri Prof. Dr. Başar ÖZTAYŞI ve Dr. Belma ŞIPKA'ya teşekkürlerimi iletmek isterim. Emekleriniz ve rehberliğiniz sayesinde bu çalışma şekillenmiş ve tamamlanmıştır. Ayrıca tez sürecimde samimi desteğiyle bana yol arkadaşlığı eden değerli dostum Dr. Nevcihan TORAMAN'a en içten teşekkürlerimi sunarım.

Bu yolculukta yanımda olan ve bana sabır gösteren canım aileme minnettarım. Onların sevgisi, cesaretlendirmeleri ve anlayışı olmasaydı, bu çalışma bu denli tatmin edici bir sonuca ulaşamazdı. İyi ki varsınız.

Bu süreçte yanımda olan tüm sevdiklerime ve arkadaşlarıma da gönülden teşekkür ederim.

Edindiğim bilgi ve tecrübeler, gelecekte yapacağım çalışmalara ışık tutacaktır.

Mayıs 2024

Burcu BİNBOĞA
(Endüstri Mühendisi)



İÇİNDEKİLER

Sayfa

ÖNSÖZ	ix
İÇİNDEKİLER	xi
KISALTMALAR	xv
SEMBOLLER	xvii
ÇİZELGE LİSTESİ.....	xix
ŞEKİL LİSTESİ.....	xxi
ÖZET	xxiii
SUMMARY	xxv
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	5
3. ARAŞTIRMA METODOLOJİSİ.....	25
4. SİSTEMATİK LİTERATÜR TARAMASI.....	31
4.1. Araştırma Soruları	31
4.2. Veri Kaynakları	32
4.3. Arama Kriterleri	33
4.4. Makale Seçim Süreci.....	33
4.5. Seçilen Makalelere Ait Yayınların İncelenmesi.....	35
4.5.1. Yıllara göre seçilmiş yayınların dağılımı	36
4.5.2. Seçilen makalelerin kalitesi	37
5. ÇEVİK YAZILIM BAŞARI MODELİ.....	41
5.1. Çevik Başarı Ölçütleri	44
5.1.1. Süreç verimliliği.....	44
5.1.1.1. Projenin planan zamanda bitirilmesi.....	46
5.1.1.2. Planlanan bütçe dahilinde tamamlama.....	46
5.1.1.3. Ürünün belirlenen kapsam dahilinde teslim edilmesi	47
5.1.2. Sürdürülebilir yazılım ürün kalitesi	47
5.1.2.1. Yazılım güvenilirliği	48
5.1.2.2. Sistem güvenliği.....	48
5.1.2.3. Bakım yeteneği ve korunabilirlik.....	49
5.1.2.4. Taşınabilirlik	50
5.1.2.5. Performans etkinliği	50
5.1.2.6. Uyumluluk	51
5.1.2.7. Fonksiyonel uygunluk.....	52
5.1.2.8. Kullanılabilirlik.....	53
5.1.2.9. Teslim edilen yazılımın genel kalitesi.....	53
5.1.3. Paydaş memnuniyeti	54
5.1.3.1. Müşteri memnuniyeti	55
5.1.3.2. Ekip memnuniyeti	55
5.1.3.3. Üst yönetim memnuniyeti.....	56
5.2. Çevik Kritik Başarı Faktörleri.....	56

5.2.1. Müşteri faktörleri.....	60
5.2.1.1. Proje boyunca güçlü müşteri desteği.....	60
5.2.1.2. Güçlü müşteri ilişkisi	60
5.2.1.3. Müşterilerin BT alanındaki profesyonel yeteneklerinin geliştirilmesi	61
5.2.1.4. Müşterilerin kendi alanındaki deneyimi.....	62
5.2.2. Ekip faktörleri	63
5.2.2.1. Çevik ekip üyelerinin yetenekleri	63
5.2.2.2. Çevik ekip üyelerinin yazılım geliştirme konusundaki tecrübesi ..	63
5.2.2.3. Çevik ekip üyelerinin mesleki yeteneklerinin teknik eğitim yoluyla geliştirilmesi	64
5.2.2.4. Proje yöneticisinin bilgisi	65
5.2.2.5. Proje yöneticisinin tecrübesi	66
5.2.2.6. Ekip üyelerinin motivasyonu	66
5.2.2.7. Kendi kendini örgütleyen ekip üyeleri	67
5.2.3. Organizasyonel faktörler	68
5.2.3.1. Çevik tarzda çalışma ortamı	68
5.2.3.2. Hiyerarşik olmayan işbirlikçi yönetim tarzı	69
5.2.3.3. Kurumsal olarak kabul edilen değişim yönetimi.....	70
5.2.3.4. İyi tanımlanmış vizyon ve misyon	71
5.2.3.5. Üst yönetim desteği	72
5.2.4. Çevik süreç faktörleri	73
5.2.4.1. Ekip üyeleri arasında güçlü iletişim ve koordinasyon	73
5.2.4.2. Çevik etkinlikleri doğru şekilde uygulamak	73
5.2.4.3. İlerleme izleme ve kontrol mekanizması	74
5.2.4.4. Doğru miktarda dokümantasyon	75
5.2.5. Teknik faktörler.....	76
5.2.5.1. Etkin yazılım test süreci	76
5.2.5.2. Programlama standartlarına uyum	77
5.2.5.3. Yeniden düzenleme etkinlikleri	78
5.2.5.4. Çalışan yazılımın düzenli teslimatı	79
5.2.5.5. Doğru araç ve teknoloji seçimi.....	80
5.2.5.6. Etkin risk analizi yönetimi	80
5.2.5.7. Sadelik	81
5.2.6. Proje faktörleri.....	82
5.2.6.1. Proje kritikliği	82
5.2.6.2. Aciliyet	83
5.2.6.3. Teknik karmaşıklık.....	84
5.2.6.4. Teknolojik belirsizlik	85
5.2.6.5. Kapsam değişikliği	85
5.2.6.6. Büyük proje boyutu.....	86
6. ÇEVİK PROJE BAŞARI MODELİNİN ANALİZİ	87
6.1. Açıklayıcı Faktör Analizi	88
6.2. Doğrulayıcı Faktör Analizi.....	89
6.3. Yapısal Model	94
6.4. Çevik Yazılım Projelerinde Sektörel ve İş Rolü Farklılıklarının Etkisi.....	95
7. SONUÇLARIN TARTIŞILMASI	99
8. ARAŞTIRMA KISITLARI VE GELECEK ÇALIŞMALAR	103
9. SONUÇ	107
KAYNAKLAR.....	109

EKLER	117
ÖZGEÇMİŞ	121





KISALTMALAR

AFA	: Açıklayıcı Faktör Analizi
AGFI	: Düzeltilmiş Uyum İyiliği İndeksi
AHP	: Analitik Hiyerarşi Süreci
AVE	: Ortalama Çıkarılan Varyans
BT	: Bilgili Teknolojileri
CFI	: Karşılaştırmalı Uyum İndeksi
DFA	: Doğrulamalı Faktör Analizi
GFI	: Uyum İyiliği İndeksi
IJPM	: International Journal of Project Management
IST	: Information and Software Technology
JEIM	: Journal of Enterprise Information Management
JSS	: Journal of Systems and Software
KBF	: Kritik Başarı Faktörleri
KMO	: Kaiser-Meyer-Olkin
MANOVA	: Çok Değişkenli Varyans Analizi
NFI	: Normlaştırılmış Uyum İndeksi
PMI	: Project Management Institute
RMSEA	: Kök-Ortalama-Yaklaşım Karesi
SLT	: Sistematik Literatür Taraması



SEMBOLLER

p : İstatiksel anlamlılık düzeyi katsayısı





ÇİZELGE LİSTESİ

Sayfa

Çizelge 4.1 : Seçilen makalelerin dağılımı.	35
Çizelge 4.2 : H-indeksine dayalı yayınların kalitesi.	38
Çizelge 4.3 : En çok atıf alan 10 seçilmiş makale.	39
Çizelge 5.1 : Çevik uygulayıcıların demografik profili.	42
Çizelge 5.2 : Çevik başarı ölçütleri.	45
Çizelge 5.3 : Çevik kritik başarı faktörleri.	57
Çizelge 6.1 : Katılımcıların demografik profili.	88
Çizelge 6.2 : Standartlaştırılmış öge yükleri, açıklanan varyans, ve Cronbach's alpha değerleri.	90
Çizelge 6.3 : Doğrulamalı faktör analizi model uygunluk istatistikleri.	91
Çizelge 6.4 : Yakınsak geçerlik sonuçları.	92
Çizelge 6.5 : Ayrıştırıcı geçerlik sonuçları.	93
Çizelge 6.6 : Yapısal model tahmini.	95
Çizelge 6.7 : Çok değişkenli test sonuçları.	97
Çizelge 6.8 : Bağımsız gruplar arası etki testleri.	98
Çizelge B.1 : Anket soruları.	121



ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1 : Çalışma modeli.....	29
Şekil 4.1 : Makale seçme sürecine ait akış.	35
Şekil 4.2 : SLT aracılığıyla seçilen makalelerden üç veya daha fazlasını yayınlayan dergilerdeki makalelerin dağılımı.	36
Şekil 4.3 : Seçilen makalelerin yıllara göre dağılımı.....	37
Şekil 5.1 : Çevik yazılım modeli.	43



ÇEVİK YAZILIM GELİŞTİRME PROJELERİNDE KRİTİK BAŞARI FAKTÖRLERİNİN MODELLENMESİ: ÇEVİK PROJELERİN TÜRKİYE’DEKİ UYGULAMALARI

ÖZET

Son yıllarda hızla gelişen teknoloji ile değişen müşteri ihtiyaçlarına kaliteden ödün vermeden hızlı bir şekilde cevap vermek çok önemlidir. Günümüzde, piyasada her ihtiyaç duyulan ürünün muadili bulunmaktadır. Ürünleri, muadillerinden farklılaştıran bir çok özellik örneğin müşteri odaklılık, ihtiyaçlara hızlı cevap verebilme, kalite ve güvenilirlik gibi kavramlar sadece müşteri memnuniyetini artırmakla kalmaz aynı zamanda tüm paydaşların memnuniyetini artıracaktır.

Yazılım geliştirmede kullanılan proje yönetimi metodolojileri de teknolojik gelişmelerle değişime uğramaktadır. Önceki yıllarda daha çok şelale yöntemleri diye adlandırdığımız klasik yazılım geliştirme metodolojileri kullanılırken günümüzde çevik yazılım proje yönetimi yaklaşımları tercih edilmektedir. Çevik proje yazılımı, hızla değişen gereksinimlere uyum sağlayabilmek için esnek ve dinamik bir yaklaşım sunan bir proje yönetim metodolojisidir. Kısa geliştirme döngüleriyle çalışan ekipler, müşteri geri bildirimlerini doğrudan kullanarak yazılım geliştirme sürecini yönlendirir. Böylece yazılımın kalitesi, müşteri memnuniyeti ve proje verimliliği artar. Sürekli iyileştirme ve işbirliğini teşvik eden bu metodoloji, özellikle belirsizlik ve sürekli değişimle karşılaşılan sektörlerde projelerin başarılı bir şekilde yürütülmesine yardımcı olmaktadır. Bu nedenle çevik yazılım geliştirme metodolojileri son yıllarda müşteri ve iş gereksinimlerini hızlı ve etkili bir şekilde karşılamak amacıyla giderek daha popüler hale gelmektedir.

Yazılım gereksinimlerindeki belirsizlikler ve değişiklikler, şirketleri yazılım geliştirme projelerinde daha çevik olmaya zorlamaktadır. Günümüzde çevik bir ortamda şirketler ilerleyebilmek için projelerine çevik yazılım geliştirme metodolojilerini entegre etmektedir. Ancak çevik metodolojinin uygulanma şekli projenin başarısını belirleyebilir. Bu tez kapsamında, çevik uygulayıcıların bakış açısından çevik projelerin başarısını etkileyen kritik başarı faktörleri ve çevik başarı ölçütlerinin belirlenmesi amaçlanmaktadır. İlk olarak Çevik Manifesto, Çevik İlkeler ve Scrum Kılavuzu’nun detaylı incelenmesiyle kapsamlı bir sistematik literatür taraması yapılarak Çevik Yazılım Proje Başarı Modeli geliştirilmiştir. İkinci olarak altı çevik proje uygulayıcısı ile bire bir görüşmeler yapılarak kritik başarı faktörleri ve çevik başarı ölçütleri güncellenmiş ve ardından model üzerinde fikir birliğine varmak için katılımcılarla grup toplantısı yapılarak nihai Çevik Yazılım Proje Modeli geliştirilmiştir.

Modelde çevik başarı ölçütleri olarak süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti belirlenmiştir. Kritik başarı faktörleri ise müşteri faktörleri, ekip faktörleri, organizasyonel faktörler, çevik süreç faktörleri, teknik faktörler ve proje faktörleri olarak altı ana başlıkta tanımlanmıştır. Sonrasında kritik başarı ölçütleri ile çevik başarı ölçütleri arasındaki ilişki literatürden desteklenerek

Çevik Yazılım Proje Modeli tanımlanmıştır. Ayrıca kritik başarı faktörlerinin ve çevik başarı ölçütlerinin alt başlıkları da tanımlanmış ve bu alt başlıkları tanımlayan soru ölçekleri geliştirilmiştir. Tasarlanan Çevik Yazılım Proje Anketi çevik proje başarısını etkileyen kritik başarı ölçütlerinin belirlenmesinde kullanılabilecektir.

Geliştirilen modeli test etmek için çevik yazılım proje yönetimi deneyimi olan 596 katılımcıdan veri toplanarak proje başarısına etki eden faktörler belirlenmiştir. Veri analizi için yapısal eşitlik modeli kullanılmıştır. IBM Amos 20.0 ve IBM SPSS Statistics versiyon 28 yazılım programlarından faydalanılmıştır. İlk olarak açıklayıcı faktör analizi kullanılarak kritik başarı ölçütlerinin faktör yapısı ortaya çıkarılmış ve modelde belirlenen faktör sayısı ile aynı sayıda faktör elde edilmiştir. Daha sonraki adımda doğrulayıcı faktör analizi ile belirlenen faktörlerin teorik yapısının geçerliliği gösterilmiştir. En son adımda ise kritik başarı faktörleri ile çevik başarı ölçütleri arasındaki ilişki yapısal eşitlik modellemesi ile test edilmiştir. Sonuçlara göre müşteri faktörleri, ekip faktörleri, çevik süreç faktörleri ve proje faktörlerinin süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti açısından çevik proje başarısının önemli belirleyicileridir. Kritik başarı faktörleri arasında müşteri faktörleri ve çevik süreç faktörleri diğer faktörlere kıyasla süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyetini etkileyen güçlü belirleyicilerdir.

Bu çalışma ile teorik ve pratik çıkarımlar sunulmuş ve olası gelecek çalışmalar için önerilerde bulunulmuştur. Geliştirilen Çevik Yazılım Proje Başarı Modeli, projelerin başarılı bir şekilde yürütülmesi için kritik olan faktörlerin belirlenmesinde önemli bir rol oynamaktadır. Bu model, çevik yazılım geliştirme süreçlerinin daha iyi anlaşılmasını sağlamakta ve çevik metodolojilerin uygulanmasındaki etkinliği artırmak için yol gösterici olmaktadır. Araştırmanın bulguları, özellikle müşteri ve süreç odaklı yaklaşımların çevik projelerde başarıya ulaşmada kilit olduğunu vurgulamaktadır. Bu faktörler, projelerin zamanında ve bütçe dahilinde tamamlanmasını, aynı zamanda yüksek kaliteli yazılım ürünlerinin teslim edilmesini sağlamakta büyük bir rol oynamaktadır. Ayrıca çevik metodolojilerin daha etkili bir şekilde uygulanabilmesi için çevik uygulayıcılar ve proje yöneticileri için somut adımlar önermektedir. Bunun yanı sıra tasarlanan Çevik Yazılım Proje Anketi, gelecek çalışmalarda çevik proje yönetimi uygulayan farklı sektörlere de uygulanarak sektöre özel kritik başarı faktörleri belirlenmesine yardımcı olacaktır.

MODELLING CRITICAL SUCCESS FACTORS IN AGILE SOFTWARE DEVELOPMENT PROJECTS: APPLICATIONS OF AGILE PROJECTS IN TÜRKİYE

SUMMARY

Uncertainties and frequent changes in software requirements forces companies to adopt more agile methodologies in the management of software development projects. These methodologies are essential to synchronize ever-changing demands of the customers and staying current and responsive in a fast-paced software industry. Agile methodologies, like Scrum, Kanban, and Lean, are designed to foster collaboration, continuous improvement, and the delivery of high-quality products. They emphasize direct communication, customer involvement, and a flexible response to change. By breaking down projects into manageable units, agile projects enable teams to focus on high-value features, delivering them in iterative cycles called sprints. On the other hand, the success of these project management strategies often depends on how they are implemented. Therefore, understanding factors affecting the success of agile projects is important to achieve success in the agile project management.

In the current thesis, agile software project success model that combines agile critical success factors with agile project success measures and project factors (agile risk and failure factors) was developed. To determine the factors in agile project management, a systematic literature review (SLR) was conducted. Journals and conference articles were selected through a detailed literature review of the last 21 years (2000–2021). In contrast to previous studies, the Agile Manifesto, Agile Principles, and Scrum Guide were examined to identify significant factors in the application of agile software development projects. Subsequently, results obtained using the SLR, Agile Manifesto, Agile Principles, and Scrum Guide were combined to obtain a comprehensive model that identifies the significant factors affecting agile project success. Additionally, the factors were refined via individual and group meetings with six agile practitioners working in the software industry. Finally, Agile Software Project Success Model was developed. This model includes six factors explaining critical success factors: customer factors, team factors, organizational factors, agile process factors, technical factors, and project factors. Customer factors focus on maintaining a strong relationship with the customer by securing their support, aligning deliverables with their business needs, and enhancing their IT capabilities. Team factors emphasize the skills and experience of team members in software development, their motivation, and their ability to self-organize. They also involve enhancing their professional capabilities through technical training, ensuring effective and cohesive teamwork. Organizational factors encompass an agile work environment that encourages collaboration, a cooperative management style prioritizing flexibility, upper management support for agile practices, and effective change management. Agile process factors refer effective team communication, proper execution of agile events, simplified progress tracking, and maintaining essential documentation. Technical factors address the use of appropriate technologies, tools, and software architecture to

support reliable and maintainable product development. Project factors refer to factors that can influence the project's success, including criticality, urgency, technical complexity, technological uncertainty, specification changes, and project size.

Agile project success measures are also defined in three constructs: process efficiency, sustainable software product quality, and stakeholder satisfaction. Process efficiency refers to the ability to deliver projects on time, within the planned budget, and aligned with the agreed scope. Sustainability of software product quality encompasses various non-functional requirements, focusing on maintaining software reliability, security, and other quality attributes over time. It is defined through characteristics like maintainability, portability, performance efficiency, compatibility, functional suitability, and usability, which together ensure that the software remains robust and adaptable. Stakeholder satisfaction is the degree to which a project's outcomes meet the expectations and needs of key stakeholders, including customers, team members, and upper management. In the Agile Software Project Success Model, the relationships between critical success factors and agile project success measures were defined with referencing the literature.

In the final stage, Agile Software Project Questionnaire was adapted to measure the critical success factors and agile project success measures. The questionnaire consists of two parts: In the first part demographic questions were asked to agile practitioners about gender, age, project role, working industry etc. In the second step, a total of 45 items that define critical success factors and agile project measures was asked to agile practitioners on a 7 Likert (1: totally disagree...7: totally agree) scale. In the current project, a total of 596 questionnaires were collected from agile practitioners. According to demographic profile most of the respondents are working in Turkey. Furthermore, banking, stock exchange and software industry constitute a big part of the respondents. In addition, 68% of the respondents uses scrum as an agile project management methodology. In the analysis of the model, explanatory factor analysis and structural equation modeling approach were used. IBM Amos 20.0 and IBM SPSS Statistics software version 28 were used as a software for the analysis. First, explanatory factor analysis was performed to reveal the factor structure of the model. According to the results, nine factors that define critical success factors with eigen values greater 1 were revealed. This number was same as the number of factors in the Agile Software Project Success Model. Then confirmatory factor analysis was performed to validate the factors defined in the model. The initial analysis of CFA achieved model fit values greater than the threshold values. Furthermore, the convergent and discriminant validity of the constructs were also validated. Structural model analyzes the relationships defined in the Agile Software Project Success Model. According to the results, customer factors, team factors, agile process factors, and project factors have significant effects on process efficiency, the quality of software products, and stakeholder satisfaction. Among these, agile process factors have stronger effects than the other factors on sustainable software product quality. This impact underscores the importance of strong communication within teams, the effective implementation of agile events, progress tracking mechanisms, and maintaining an optimal level of documentation. Additionally, customer factors are highlighted as significant predictors of stakeholder satisfaction, software product quality, and process efficiency. This implies that strong relationships with customers, enhancing their experience throughout the project lifecycle, prioritizing responsive customer support are important.

On the other hand, technical factors such as programming standards and software testing process do not have a significant effect on stakeholder satisfaction. This outcome suggests that while technical aspects are crucial for the intrinsic quality and functionality of software products, they do not directly translate to stakeholder satisfaction. Furthermore, the effect of organizational factors such as establishment of a supportive agile culture and an appropriate structural framework also did not have a significant effect on sustainable quality of software products. This finding implies that the quality of software deliverables relies more heavily on the technical skills and operational practices employed during the development process.

The study's findings have profound theoretical and practical implications. First, this study provides Agile Software Project Success Model that defines critical success factors and agile project success measures. Agile practitioners may examine the model to define significant factors in the management of agile projects. Furthermore, the questionnaire developed to measure the perception of agile practitioners may be applied to reveal significant critical success factors. Second, the questionnaire was applied and significant predictors of agile project success were determined. All the hypotheses except two of them was approved by the data collected. The findings suggest that for organizations to harness the full potential of agile methodologies, they should invest in cultivating an agile-friendly culture that emphasizes continuous improvement, open communication, and team empowerment. Additionally, the findings recommend that product owners and scrum masters focus on enhancing customer involvement and ensuring the adaptability of processes to improve project outcomes. These findings are intended to assist project managers, scrum masters, software development teams, product owners, and internal customers in enhancing their agile practices, ultimately leading to more successful project outcomes and greater satisfaction among stakeholders. Moreover, the differential impacts of success factors suggest that organizations should conduct regular reviews of their agile practices and make adjustments based on specific project needs, team dynamics, and organizational goals.



1. GİRİŞ

Son zamanlarda teknolojinin hızla değişimi şirketlerin rekabetçi ortama ayak uydurmasını gittikçe zorlaştırmaktadır. Üretimin hiçbir zaman durmaması gerektiğini savunan klasik üretim anlayışının aksine Toyota fabrikalarında farklı bir anlayış benimsenmiştir. Her işçi montaj hattında bir problem gördüğünde iş alanının hemen yanında bulunan kolu çekerek üretim bandını durdurabilmektedir. Ayrıca Toyota üretim sisteminin iki temel taşından biri olan tam zamanında üretim ile her parça tam gerektiği anda, tam gerektiği miktarda montaj aşamasına gelmektedir. Tam zamanında üretimi gerçekleştiren Kanban sistemi 6 temel prensibe dayanmaktadır (Ohno, 2017). Bunlar;

- İşleri görselleştirme (Kanban panoları ile süreci ve işleri şeffaflaştırma prensibine dayanmaktadır)
- İş miktarına sınırlama getirme (Takımın çalıştığı işlere limit getirilmektedir)
- Süreci yönetme (Kanban panoları ile takım süreci yönetir)
- Sürece ait kuralları uygulanabilir, bilinir olması ve iş tarafına karşı bu konuda bir duruş sergilenmesi
- Geri bildirim mekanizması ile risk profili geliştirme
- Bilimsel yöntemler yardımıyla, sürekli gelişme, sürekli iyileşme ve sürekli teslimat

Yazılım geliştirmede kullanılan metodolojilerin felsefeleri de üretim sistemlerinde kullanılan tam zamanında üretim metodolojilerine benzemektedir. Şirketler arasında rekabetin artmasıyla yazılım geliştirmede klasik anlayış olan Şelale yaklaşımı şirketlerin ihtiyaçlarına cevap veremez hale gelmiştir. Böylelikle yazılım geliştirme ekipleri tarafından çevik ortama ayak uydurmak, en kaliteli ürünü tam gerektiği anda tam gerektiği miktarda ve işleri paralelize ederek minimum hata ile geliştirme yapmak için üretim sistemlerinde kullanılan Kanban, tam zamanında üretim, otomasyon gibi kavramlar da son yıllarda yazılım sektöründe kullanılmaktadır. Bu sayede yazılım

geliştirme ekibi en verimli şekilde çalışarak, minimum hata ile tam zamanında geliştirme yaparak, sürekli üretim ortamına teslimat yapmaktadır. Test otomasyonları yardımı ile her geliştirme sonrasında hata bulunup anında müdahale edilerek hatasız ürün geliştirmesi yapılmaktadır.

Yazılım geliştirmede kullanılan proje yönetimi metodolojisi, bir projenin genel başarısında çok önemli bir rol oynar. Metodoloji genellikle müşterilerin gereksinimlerine ve şirketin organizasyonel yapısına göre belirlenir. Ancak yazılım niteliklerindeki belirsizlik ve müşteri ihtiyaçlarındaki hızlı değişimler, yazılım geliştirme faaliyetlerinin daha hızlı ve çevik olmasını gerektirmektedir. Bu nedenle son yıllarda çevik yazılım geliştirme metodolojileri müşteri ve iş ihtiyaçlarına cevap vermek için tercih edilmektedir.

Çevik yazılım geliştirme, Çevik Manifesto ve Çevik İlkeler’de belirtilen temel ilkeler ve değerler üzerine kurulu kapsamlı bir dizi çerçeve ve metodolojiyi kapsar (Agile Alliance, 2023). Çevik yazılım geliştirme metodolojisi sık sürüm döngüleri, sürekli iyileştirme, hızla değişen gereksinimlere hızlı geri bildirim, yüz yüze görüşme ve yüksek kaliteli yazılım vaat etmektedir (Chan ve Thong, 2009; Klünder ve diğ, 2019). Bu nedenle son zamanlarda birçok şirket geleneksel yazılım geliştirme metodolojileri yerine çevik yazılım geliştirme metodolojilerini kullanmayı tercih etmektedir. Standish Group Chaos'un son raporu, çevik proje başarı oranlarının şelale modeline göre üç kat, şelale modelinin başarısızlık oranının ise çevik projelere göre iki kat daha fazla olduğunu göstermektedir (Johnson, 2021). Her ne kadar çevik projeler geleneksel projelerden daha yüksek başarı oranına sahip olsa da, çevik projelerin başarısını etkileyen çevik kritik başarı faktörlerini (KBF) anlamak yazılım geliştirme yönetimi için önemlidir. Projeler için kritik başarı faktörleri, “uygun şekilde sürdürüldüğünde, korunduğunda ve yönetildiğinde projenin başarısı üzerinde önemli bir etkiye sahip olabilecek özellikleri, koşulları veya değişkenleri” ifade eder (Milosevic ve Patanakul, 2005, s. 183). Genellikle kritik başarı faktörleri olarak adlandırılan başarı ve başarısızlık modelleri, bugün yazılım endüstrisi tarafından öğrenilen, uyarlanan ve kullanılan derslerin en iyi göstergelerinden bazılarıdır (Garousi ve diğ, 2019, s. 2). Kritik başarı faktörleri doğru şekilde ele alındığında, projenin başarılı olma olasılığını önemli ölçüde artırır (Nasir ve Sahibuddin, 2011).

Çevik metodolojiler için başarı faktörleri (Chow ve Cao, 2008; Misra ve diğ, 2009; Wan ve Wang, 2010; Stankovic ve diğ, 2013; Silva ve Santos, 2015; Ahimbisibwe ve

diğ, 2015, 2017; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019), çevik proje başarısını etkileyen risk faktörleri (Shrivastava ve Rathod, 2015, 2017) ve çevik başarısızlık faktörleri (Chow ve Cao, 2008; Stankovic ve diğ, 2013) gibi birçok çalışma çevik yazılım geliştirme projelerindeki başarı, başarısızlık ve risk faktörlerine odaklanmıştır. Bu çalışmalar başarı faktörleri, risk ve başarısızlık faktörlerinin proje başarısı üzerindeki bireysel etkilerine dair içgörüler sunsa da bu tezin amacı çevik kritik başarı faktörlerini çevik proje başarı ölçütleriyle birleştiren kapsamlı bir çevik yazılım projesi başarı modeli geliştirmektir. Çevik yazılım proje yönetimindeki kritik faktörleri tespit etmek için detaylı bir literatür taraması yapılmıştır. Önceki çalışmaların aksine, çevik yazılım geliştirme projelerinin uygulanmasındaki önemli faktörleri belirlemek için Çevik Manifesto, Çevik İlkeler ve Scrum Kılavuzu ayrıntılı olarak incelenmiştir. Ayrıca yazılım sektöründe çalışan altı çevik uygulayıcı ile bireysel ve grup toplantıları yapılarak faktörler geliştirilmiştir. Son olarak Çevik Yazılım Projesi Başarı Modeli geliştirilmiştir. Oluşturulan modelde tanımlanan kriterleri tanımlayan anket öğeleri tanımlanarak Çevik Proje Yönetimi Anketi literatürden oluşturulmuştur. Geliştirilen model, 596 çevik uygulayıcısından toplanan geniş ölçekli bir anket verisi ile test edilmiştir. Geliştirilen model, scrum master, ürün sahipleri, geliştiriciler ve araştırmacılar tarafından süreç verimliliği, ürün kalitesi ve paydaş memnuniyeti gibi çevik proje başarı ölçütlerini etkileyen kritik faktörlerini anlamak için bir referans model olarak kullanılabilir. Ayrıca farklı sektörde çevik yazılım yönetimi benimseyen uygulayıcılar için oluşturulan anket kritik başarı faktörlerini benimsemek için kullanılabilir.

Bu tez temel olarak 9 bölümden oluşmaktadır. Bölüm 2’de kritik başarı faktörleri ve proje başarı ölçütlerine ilişkin mevcut literatürün kapsamlı bir incelemesi sunulmaktadır. Bölüm 3’te araştırma metodolojisi sunulmaktadır. Bölüm 4’te tez kapsamında yapılan sistematik literatür taramasından bahsedilmektedir. Bölüm 5’te Çevik Yazılım Projesi Başarı Modeli, çevik başarı ölçütleri ve çevik kritik başarı faktörleri açıklanmaktadır. Bölüm 6’da veri analizi ve sonuçlar sunulmaktadır. Bölüm 7’de sonuçların tartışılması sunulmaktadır. Bölüm 8’de araştırma kısıtları ve gelecekteki araştırma alanları önerilmekte, Bölüm 9’da ise sonuçlar hakkında bilgi verilmektedir.



2. LİTERATÜR ARAŞTIRMASI

Çevik yazılım geliştirme metodolojileri günümüzde giderek daha fazla yaygınlaşan ve organizasyonların rekabet avantajı elde etmelerine yardımcı olan bir yaklaşım olarak önem kazanmaktadır. Bu yaklaşım, hızlı değişen pazar koşullarına uyum sağlayabilme, müşteri memnuniyetini artırabilme ve ürün kalitesini iyileştirebilme gibi avantajlar sunmaktadır. Ancak çevik yazılım geliştirme metodolojilerinin başarısı, pek çok faktörün etkileşimine bağlıdır ve bu faktörlerin tam olarak anlaşılması önemlidir. Uzun yıllardır araştırmacılar, çevik yazılım projelerinin başarısını etkileyen faktörleri incelemiş ve bu alandaki literatürü zenginleştirmektedir.

Yeo (2002), katılımcıların “başarısız” bir projedeki deneyimlerine dayanarak bilgi teknolojileri projelerindeki kritik başarısızlık faktörlerini belirlemek için Singapur merkezli bir anket çalışması gerçekleştirmiştir. Daha sonra faktörleri analiz etmek için üçlü S çerçevesi kullanılmıştır. Yeo (2002), literatür taraması ve kendi araştırmasından elde ettiği bulguları kullanarak bilişim teknolojileri (BT) projelerinde sıkça karşılaşılan başarısızlık faktörlerini tanımlamıştır. Bu faktörler arasında, zaman çizelgesinde hatalı tahminleme, proje risk analizinin yetersizliği, gereksinimlerin belirsizliği, net olmayan vizyon&misyon ve teknolojik belirsizlikler gibi öğeler yer almaktadır.

Fortune ve White (2006), 63 makalenin sistematik literatür taramasıyla toplam 27 tane kritik başarı faktörü belirlemişlerdir. Biçimsel Sistem Modeli önerilmiş ve kritik başarı faktörleri Biçimsel Sistem Modelinin bileşenleriyle eşleştirilmiştir. Çalışmada yer alan kritik başarı faktörleri arasında üst yönetim desteği, iyi tanımlanmış vizyon&misyon, iyi müşteri iletişimi, tecrübeli ekip üyeleri, değişim yönetimine sahip olma, yetkin proje müdürü, tecrübeli proje müdürü, teknolojik gelişmelerle ilgili yetkin olma, iyi tahminlenmiş bir zaman çizelgesine sahip olma, proje ilerleme izleme ve kontrol mekanizması, yeterli bütçe, düzenli organize edilen eğitimler, tecrübeli ekip üyeleri ve proje büyüklüğü gibi öğeler yer almaktadır. Araştırmanın sonuçları, proje başarı faktörlerinin, proje başarısını etkileyen karmaşık bir sistem içinde nasıl etkileşime girdiğini ve nasıl çerçevelendiğini göstermektedir.

Chow ve Cao (2008), literatür taramasına dayanarak çevik projeler için bir ön liste hazırlamış, kritik başarı faktörlerini belirlemiştir. Çalışmada web tabanlı bir anket 25 farklı ülkeden 109 kişiye uygulanmış ve toplanan verileri analiz etmek için faktör analizi ve çoklu regresyon yöntemi kullanılmıştır. Çevik yazılım projesi için 5 kavramdan bahsedilmektedir. Organizasyonel faktörler, insan faktörleri, süreç faktörleri, teknik faktörler ve proje faktörleri çalışmada tanımlanmış olan kavramlardır. Organizasyonel faktörler için güçlü yönetim desteği, hiyerarşik değil işbirlikçi organizasyon kültürü, uygun çevik tarzda çalışma ortamı, çevik prensiplerin uygulanması gibi kritik başarı faktörlerinden bahsedilmektedir. İnsan faktörleri için motivasyona sahip ekip üyeleri, çevik süreç hakkında bilgili yöneticiler, kendi kendini organize eden ekip üyeleri ve iyi müşteri ilişkisi gibi öğelerden bahsedilmektedir. Süreç faktörleri için çevik odaklı gereksinim yönetim sürecinin takip edilmesi, günlük yüz yüze toplantılar, güçlü müşteri varlığı, çevik odaklı kontrol ve izleme mekanizması ve çevik odaklı proje yönetim sürecinin takip edilmesi gibi kritik başarı faktörlerinden bahsedilmektedir. Teknik faktörler için kodlama standartlarının takip edilmesi, sürekli teslimat, uygun miktarda dokümantasyon, ekip üyelerine uygun teknik eğitimlerin verilmesi, ekip yetkinlikleri, test süreçlerinin doğru bir şekilde yönetilmesi gibi kritik başarı faktörlerinden bahsedilmiştir. Proje faktörleri için proje büyüklüğü, risk yönetimi, düzgün maliyetlendirme çalışmasına sahip olmanın çevik proje başarısı için kritik başarı faktörü olabileceği belirtilmiştir. Çalışma sonunda yapılan regresyon analizi çalışması sonrasında 48 hipotezden yalnızca 10'unun desteklendiğini göstererek çevik yazılım geliştirme projeleri için üç kritik başarı faktörü belirlenmiştir. Bunlar teslimat stratejisi, çevik yazılım mühendisliği teknikleri ve ekip yetkinliklerini içermektedir. Chow ve Cao (2008), çevik yazılım geliştirme alanında araştırmacılar ve çevik uygulayıcılar için önemli bir referans kaynağı olarak kabul edilmektedir.

Misra ve diğ. (2009), literatür taraması yaparak elde ettikleri bulguları analiz etmiş ve çevik yazılım geliştirme uygulamalarının benimsenmesinde kritik olan faktörleri tanımlamışlardır. Çalışma sonucu bulunan faktörlerin, çevik yazılım geliştirme uygulamalarının başarıyla benimsenmesinde kritik bir rol oynadığı vurgulanmıştır. Regresyon analizi kullanılarak yapılan değerlendirmede, 14 bağımsız değişkenin çevik yazılım geliştirme başarısı olarak tanımlanan bağımlı değişkenle ilişkisi incelenmiştir. Analiz sonuçlarına göre, müşteri memnuniyeti, müşteri işbirliği, müşteri bağlılığı,

karar verme süresi, kurumsal kültür, kontrol mekanizmaları, kişisel ve toplumsal kültür faktörleri ile eğitim ve öğrenme süreçlerinin başarıyla doğrudan ilişkili olduğu belirlenmiştir. Öte yandan ekip dağılımı, ekip büyüklüğü, planlama, teknik yeterlilik, iletişim gibi faktörlerin başarı ile anlamlı bir ilişkisi bulunamamıştır. Bu sonuçların çevik yazılım geliştirme uygulamalarının başarılı bir şekilde benimsenmesinde belirleyici olan kritik başarı faktörlerinin daha iyi anlaşılmasına katkı sağlayacağı belirtilmiştir.

Howell ve diğ. (2010), proje belirsizliği ve sonuçlarına dayanan bir olumsuzluk çerçevesini geliştirmeyi amaçlamaktadır. Bu çerçeve, belirsizlik, karmaşıklık, aciliyet, ekip güçlendirme ve kritiklik olmak üzere beş seçim faktörü tanımlamaktadır. Araştırma modeli, bu faktörleri kullanarak projeler için uygulanabilir bir yönetim metodolojisi seçimi sağlayacak şekilde olumsuzluk teorisini genişletmektedir. Araştırmada, literatür taraması yoluyla yapılmış ve alternatif proje yönetimi yaklaşımları ele alınmıştır. Çalışma sonuçları, farklı projelerin farklı yönetim yaklaşımları gerektirdiğini ve bu yaklaşımların projenin kavramına göre uyarlanması gerektiğini vurgulamaktadır. Çalışmanın önemi, proje yönetiminde olumsuzluk faktörlerinin etkisini ve bu faktörlere dayalı olarak projeler için en uygun yönetim metodolojisinin nasıl seçileceğini anlamaya katkıda bulunmasıdır. Belirsizlik, karmaşıklık ve aciliyet gibi temalar, daha önceki olumsuzluk teorisi çalışmalarında da önemli faktörler olarak kabul edilmiştir ve bu çalışmada da projelerin nasıl yönetilmesi gerektiğine dair önemli ipuçları sağlamaktadır. Ekip güçlendirme ve kritiklik gibi diğer temalar ise özellikle BT projelerinde öne çıkan faktörlerdir. Çalışma, bu faktörlerin projenin başarısı üzerindeki etkisini ve yönetim süreçlerine nasıl entegre edilebileceğini ele almaktadır. Sonuç olarak çalışma, projelerin çeşitliliğini ve projelerin yönetilme biçimlerindeki çeşitliliği yansıtan kavramsal bir model sunmaktadır. Bu çalışmanın projeler arası farklılıkları tanımak ve bu farklılıklara uygun yönetim stratejileri geliştirmek için bir temel teşkil edeceği belirtilmektedir.

Wan ve Wang (2010) çalışmasında, Çin ve Hong Kong'daki bir yazılım şirketi olan P Company üzerinde çevik yazılım süreç iyileştirmesiyle ilgili kritik başarı faktörleri incelenmiştir. Şirketin süreç yönetimi durumunu ve karşılaştığı sorunları anlamak için çalışanlara kapsamlı bir anket çalışması yapılmış, 80 anket dağıtılarak 51 geçerli cevap toplanmıştır. Katılımcılar arasında üst düzey yöneticiler, proje yöneticileri, teknik uzmanlar, programcılar, test uzmanları ve sistem yöneticileri bulunmuştur. Anket

sonuçları, faktör analizi ve hipotez testleri ile değerlendirilmiştir. Araştırmada, üst yönetim desteği, organizasyonel destek, araç ve teknoloji uygunluğu, uygulama metodunun doğruluğu ve eğitim-öğretim programlarının çevik yazılım süreç iyileştirmesi üzerindeki etkileri incelenmiştir. Tüm bu faktörlerin çevik süreç iyileştirmeleri üzerinde önemli bir rol oynadığı hipotezi desteklenmiştir. Sonuç olarak, bu çalışma P Company özelinde kritik başarı faktörlerini tanımlamış ve organizasyonların çevik süreçleri daha başarılı bir şekilde uygulayabilmesi için bu faktörlerin etkin bir şekilde yönetilmesi gerektiğini önermiştir.

Sudhakar (2012), literatür taramasına dayalı olarak yazılım projeleri için kavramsal bir kritik başarı modeli geliştirmiştir. Teknik faktörler, organizasyonel faktörler, çevresel faktörler, ürün faktörleri, proje yönetimi faktörleri, ekip faktörleri, iletişim faktörleri olmak üzere 7 kavramda 35 kritik başarı faktöründen bahsedilmiştir. 35 kritik başarı faktörü arasında proje içi iletişim, üst yönetim desteği, açık proje hedefleri, çıktıların güvenilirliği, proje planlaması, takım çalışması, proje takım koordinasyonu, kalite kontrolü, müşteri kabulü, çıktıların doğruluğu, belirsizliği azaltma, istikrarı maksimize etme, gerçekçi beklentiler ve kullanıcı katılımı gibi öğeler yer almaktadır. Çalışma, proje yönetimi, ürün, takım ve iletişim faktörlerini yazılım projeleri için önemli başarı faktörü kavramları olarak vurgulamaktadır. Sudhakar (2012) çalışması, kavramsal bir nitelik taşımaktadır ve pratik uygulamalara yönelik somut önerilerde bulunarak yazılım projeleri üzerinde çalışan organizasyonlar ve proje yöneticileri için faydalı bilgiler sunmaktadır. Araştırmanın orijinallliği, kritik başarı faktörlerinin kategorize edilmesi, 35 önemli faktörün belirlenmesi ve özellikle ürün, takım ve iletişim faktörlerinin altının çizilmesiyle sağlanmıştır. Bu çalışma, yazılım projelerini yönetme sürecinde dikkate alınması gereken yeni perspektifler sunmaktadır.

Stankovic ve diğ. (2013) tarafından gerçekleştirilen çalışma, Eski Yugoslavya'daki BT şirketlerinde çevik yazılım projelerinin başarısını etkileyen kritik başarı faktörlerini belirlemeyi amaçlamaktadır. Stankovic ve diğ. (2013), Chow ve Cao (2008) tarafından yapılan benzer bir çalışmadan esinlenerek, çoklu regresyon analizi kullanmıştır. Anket yöntemiyle elde edilen veriler, bölgedeki çeşitli BT şirketlerinden üst düzey geliştiriciler ve proje yöneticileri arasında toplanmıştır. Araştırmanın sonuçları, önceki çalışma tarafından geliştirilen modeli tamamen doğrulayamamıştır. Ancak çalışma eski Yugoslavya ülkelerinde çevik yazılım geliştirme üzerine ek bilgiler sağlayarak,

bu bölgenin dış kaynak kullanımı projeleri için bir referans olarak kullanılmasına olanak tanımıştır. Çalışmanın temel bulguları, bölgedeki çevik yazılım projelerinin başarısında etkili olan spesifik faktörleri ve bu faktörlerin çeşitli başarı boyutlarına olan etkilerini ortaya koymuştur. Bunlar arasında, proje yönetimi süreci, müşteri katılımı ve teknik uygulamalar gibi faktörler öne çıkmıştır. Ayrıca Stankovic ve diğ. (2013) çalışmasında ANOVA kullanarak Scrum ve diğer yöntemleri kullanan katılımcılar arasında proje başarısı açısından önemli bir fark olmadığını doğrulamıştır. Benzer şekilde çevik proje yürüten şirketlerde çeşitli pozisyonlarda çalışan katılımcılar arasında da anlamlı bir fark bulunamamıştır. Chow ve Cao (2008) çalışmasını referans alan ilk çalışmalardan biri olduğu için çevik yazılım geliştirme alanında araştırmacılar ve çevik uygulayıcılar için önemli bir kaynak olarak kabul edilmektedir.

Sheffield ve Lemétayer (2013), çalışmalarında yazılım geliştirme projelerinde çevikliğe yol açan temel unsurları tanımlamış ve analiz etmişlerdir. Çalışma kapsamlı literatür taraması içermektedir. Ayrıca PRINCE2, PMI ve çevik uygulama topluluklarının üyeleriyle görüşmeler ve anket çalışması yapılmıştır. Kart sıralama tekniğinin kullanıldığı görüşmeler ön cevaplar sağlamıştır. 127 kişinin katıldığı anket verileri çoklu regresyon analizi ile test edilmiştir. Analiz sonucunda bir proje ortamı faktörü (kurum kültürü) ve bir proje faktörü (proje ekibinin güçlendirilmesi) başarılı projelerin yazılım geliştirme çevikliği ile ilişkilendirilmiştir.

Ahimbisibwe ve diğ. (2015), yazılım geliştirme projeleri için kritik başarı faktörlerini belirlemeyi, kategorize etmeyi ve geleneksel plan-tabanlı ile çevik metodolojiler arasında bir uyum modeli geliştirmeyi amaçlamaktadır. Literatürü sistematik bir şekilde gözden geçirmiş ve 148 makaleden yazılım geliştirme projeleri için toplam 37 kritik başarı faktörü elde etmiştir. Ardından, kritik başarı faktörleri kategorize edilmiş, bu faktörler organizasyonel, takım ve müşteri faktörleri olarak üç ana kategoriye ayrılmıştır, geleneksel ve çevik metodolojileri karşılaştıran bir model geliştirilmiştir. Bulgular, kritik başarı faktörlerinin üç ana kategorisi içinde, bireysel faktörlerin önceki çalışmalarda ne sıklıkla bahsedildiğine göre sıralanmıştır. Araştırma, kavramsal ve meta-analitik bir odaklanmaya sahiptir. Gelecek çalışmaların, geliştirilen uyum modelini istatistiksel veriler kullanarak test etmesi gerektiği vurgulanmıştır. Bu araştırma, geleneksel plan odaklı ve çevik metodolojileri karşılaştırarak yazılım geliştirme proje başarısı üzerine tam bir uyum uygunluk perspektifi geliştirmeyi amaçlamıştır. Müşteri desteği, üst yönetim desteği, organizasyonel kültür, iyi

tanımlanmış vizyon&misyon, deęişim yönetimi, ekip üyelerinin tecrübesi ve bilgisi, proje büyüklüęü, gereksinim deęişiklikleri, proje kritiklięi, aciliyet, kaynak yetersizlięi, müşteri eğitimleri, teknik karmaşıklık gibi faktörler çalışmada önerilen kritik başarı faktörlerinden bazılarıdır.

Ahimbisibwe ve dię. (2017), Ahimbisibwe ve dię. (2015) çalışmasına dayanan bir olumsuzluk modeli geliştirmiş ve modeli deneysel olarak doğrulamıştır. Çalışmada, dış kaynaklı yazılım geliştirme projelerindeki kritik başarı faktörleri arasındaki farkları karşılaştıran ve geleneksel plan-tabanlı ve çevik metodolojileri test eden bir uyum modeli geliştirilmiştir. Literatür taraması üzerinden teorik bir model ve 54 hipotez oluşturulmuştur. Önerilen modeli test etmek için anket kullanılarak 984 katılımcıdan veri toplanmıştır. Katılımcılar, dünya çapında uluslararası dış kaynaklı yazılım geliştirme projelerine dahil olan kıdemli yazılım proje yöneticileri ve uygulayıcılardan oluşmaktadır. Yapısal eşitlik modeli kullanılarak analiz çalışması yapılmıştır. Sonuçlar, çeşitli kritik başarı faktörlerinin çevik ve geleneksel plan-tabanlı metodolojiler arasında anlamlı şekilde farklılık gösterdiğini ve çeşitli proje başarı ölçütleri için farklı şekillerde etkilendiğini göstermektedir.

Garousi ve dię. (2018), Ahimbisibwe ve dię. (2015) çalışmasındaki başarı modelini referans almıştır. Mevcut literatürden elde ettikleri kritik başarı faktörlerine dayanan bir araştırma modeli kullanarak kritik başarı faktörleri yazılım projelerinin başarısıyla ilişkilendirilmiştir. Çalışmada 101 uzmana anket uygulanmıştır. Anket, Türkiye'deki yazılım endüstrisindeki katılımcılara uygulanmıştır. Çalışma, projelerin başarısı ile kritik başarı faktörleri arasındaki korelasyonu incelemeyi amaçlamıştır. Bu kapsamda, proje yönetimi uygulamaları ve ekip işbirlięi gibi belirli KBF'lerin yazılım projelerinin başarısıyla güçlü pozitif korelasyona sahip olduęu doğrulanmıştır. Çalışmada proje yönetimi uygulamaları, etkili iletişim, yetenekli ve deneyimli proje ekibi, net gereksinimler ve hedefler ve üst yönetimin desteęi en etkili KBF'lerden olduęu belirtilmiştir.

Serrador ve Pinto (2015), çevik yöntemlerin proje başarısına olan etkisini anlamak için 1002 projeden elde edilen verilerin incelenmesi yapılmıştır. Araştırmacılar, anket yöntemiyle veri toplayarak projelerin başarı düzeylerini ve çevik yöntemlerin başarı faktörleri üzerindeki etkilerini ölçmeyi hedeflemiştir. Ankette, katılımcılardan projelerinde çevik yaklaşımlarını ne ölçüde kullandıklarını ve projelerinin başarı durumlarını belirtmeleri istenmiştir. Başarı, verimlilik ve paydaş memnuniyeti olarak

iki boyutta değerlendirilmiştir. Bunun yanında, projenin vizyon/kalitesi, proje karmaşıklığı ve ekip deneyimi gibi değişkenlerin de çevik yöntemlerin etkisi üzerindeki rolü analiz edilmiştir. Analizler için toplanan veriler üzerinde korelasyon ve regresyon analizleri uygulanmıştır. Bu analizler, çevik yöntemlerin her iki başarı boyutunda da pozitif etkileri olduğunu ortaya koymuştur. Özellikle proje vizyonunun/kalitesinin çevik yöntemlerle birlikte proje başarısına pozitif bir katkı yaptığı gözlemlenmiştir. Sonuç olarak çevik yöntemlerin projelerde başarıyı artırdığı, ancak bu etkinin çeşitli proje faktörleri tarafından da şekillendiği görülmüştür. Çalışma, proje yönetiminde esneklik ve paydaş katılımının önemini vurgulayarak, çevik yöntemlerin projelere değer katabileceğini göstermiştir.

Shrivastava ve Rathod (2015), dağıtık çevik geliştirme projelerindeki risk faktörlerini sınıflandırarak ve bu risklerin yönetilmesi için kullanılan yöntemleri belirleyerek dağıtık çevik geliştirme projelerinde karşılaşılan riskleri azaltmayı amaçlamaktadır. Araştırma, niteliksel veri toplama yöntemleri kullanarak yapılmıştır ve 13 farklı bilgi teknolojisi organizasyonundan elde edilen 28 proje üzerinde uygulamalı çalışmalar ve 13 uygulayıcı ile yapılan görüşmeler içermektedir. Analiz sonuçları, 45 dağıtık çevik geliştirme risk faktörünü beş ana kategori altında toplamıştır. Bu kategoriler şunlardır: yazılım geliştirme yaşam döngüsü, proje yönetimi, grup farkındalığı, dış paydaş işbirliği ve teknoloji kurulumudur. Örneğin, grup farkındalığı kategorisi, takım üyeleri ve diğer paydaşlar arasında gerekli iletişim, koordinasyon, işbirliği ve güven eksikliğinden kaynaklanan riskleri kapsamaktadır. Bu kategoriler, Leavitt'in Organizasyonel Değişim Modeli'nin dört bileşeni olan görev, yapı, aktör ve teknoloji ile ilişkilendirilmiş ve bu model, risklerin gerçek dünya uygulamalarında kolaylıkla yönetilmesini sağlamak için kullanılmıştır. Araştırma sonuçları, dağıtık çevik geliştirme projelerinde karşılaşılan riskleri etkin bir şekilde yönetmek için politika, prosedür ve yönergeler geliştirilmesine yardımcı olacak teorik bir çerçeve sunmuştur. Çalışma, BT şirketlerinin dağıtık çevik projelerde karşılaşılabilecekleri riskleri daha iyi anlamalarına ve bu riskleri etkin bir şekilde yönetmelerine yardımcı olacağını belirtmektedir.

Shrivastava ve Rathod (2017), Shrivastava ve Rathod (2015) çalışmasındaki modeli referans almıştır. Bu çalışma, Shrivastava ve Rathod (2015) çalışmasının devamı niteliğindedir ve dağıtık çevik geliştirme projelerinde karşılaşılan riskleri daha iyi anlamayı ve yönetmeyi hedeflemiştir. Risk faktörlerini sınıflandırmak ve dağıtık çevik

geliştirme projelerindeki etkilerini ölçmek için kapsamlı bir literatür taraması ve alan araştırmaları yapılmıştır. Araştırmacılar, endüstri uygulayıcılarının dağıtık çevik geliştirme projelerindeki riskleri nasıl algıladığını ve bu riskleri nasıl yönettiklerini değerlendirmek için anket kullanmışlardır. Riskler, etkilerine göre sıralanmış ve en problemli olanlar belirlenmiştir. En problemli risk faktörleri arasında çoklu tedarikçiler arasında zayıf koordinasyon ve çoklu tedarikçiler arasında yazılımsal entegrasyondaki riskler bulunmaktadır. Bunlarla birlikte yazılım geliştirme yaşam döngüsü kategorisinde yer alan gereksinim belirleme, test ve entegrasyon ve kodlama ve entegrasyon gibi risk alanların yüksek derecede risk taşımakta olduğu tespit edilmiştir.

Silva ve Santos (2015), çevik yazılım projelerinin başarısını etkileyen temel unsurları çeşitli perspektiflerden ele almıştır. Araştırma, çevik yazılım geliştirme bağlamında insanlar, süreçler ve teknoloji ile ilgili kritik faktörleri tanımlamayı ve analiz etmeyi amaçlamaktadır. Çevik yazılım geliştirme projelerinde kritik başarı faktörleri üzerine yapılan sistematik literatür taramasında 1387 makale arasından 23 makale seçilmiştir. Silva ve Santos (2015), bu 23 makaleyi analiz ederek çevik yazılım projelerinde insan, süreç ve teknoloji kavramına göre 14 kritik başarı faktörü belirlemiştir. İnsan kavramı; yeterlilik ve uzmanlık, üst yönetimin desteği, ekip ve müşteri motivasyonu, ekip büyüklüğü, kullanıcı katılımı olmak üzere beş kritik başarı faktöründen oluşmaktadır. Süreç kavramı; çevik pratiklerin düzgün bir şekilde uygulanması, öncelikli özelliklerin ilk teslimatı, doğru miktarda dokümantasyon ve güçlü iletişimden oluşmaktadır. Teknoloji kavramı; uygun teknik eğitim, entegrasyon testi, basit tasarım, araç desteği ve iyi tanımlanmış kodlama standartları olmak üzere beş kritik başarı faktöründen meydana gelmektedir.

Inayat ve diğ. (2015), sistematik literatür çalışması yapmış ve nihai 21 makale belirlenmiştir. 21 makalenin analizi sonucunda, geleneksel gereksinim mühendisliği uygulamalarının karşılaştığı beş temel zorluk; iletişim sorunları, aşırı kapsam, gereksinimlerin doğrulanması, gereksinimlerin dokümantasyonu ve yazılım süreçlerine nadir müşteri katılımı olarak belirlenmiştir. Çevik gereksinim mühendisliği uygulamalarının karşılaştığı sekiz zorluk; minimum dokümantasyon, müşteri erişilebilirliği, yazılım mimarisi, bütçe ve zaman tahmini, işlevsel olmayan gereksinimleri ihmal etme, müşteri yetersizliği ve anlaşmazlık, sözleşmeye dayalı sınırlamalar ve gereksinimlerin sürekli değişimi olarak belirlenmiştir. Ayrıca

sistematik literatür çalışması sonucunda elde edilen 17 çevik pratikten bahsedilmektedir. Bunlar; yüz yüze iletişim, müşteri katılımı, kullanıcı hikayeleri, yinelemeli gereksinimler, gereksinimlerin önceliklendirilmesi, değişim yönetimi, çapraz işlevsel takımlar, prototipleme, test süreçleri, gereksinimler modellemesi, gereksinim yönetimi, gözden geçirme toplantıları ve kabul testlerinin uygulanması, kod yeniden düzenleme faaliyetleri, ortak kavramsallaştırma, eşli kodlama, retrospektif toplantıları ve sürekli planlama toplantılarıdır.

Joslin ve Müller (2015) çalışmasında, proje yönetimi metodolojileri ile proje başarısı arasındaki ilişkiyi ve bu ilişki üzerinde proje yönetişiminin etkisini incelemektedir. Araştırma, iki araştırma sorusu etrafında şekillenmiş ve bu sorulara yanıt arayarak yönetişim yapılarının proje yönetimi metodolojileri kullanımı üzerindeki etkilerini değerlendirmiştir. Araştırmanın ilk sorusu, yönetişim yapıları ile proje başarısı arasında pozitif bir ilişki olup olmadığını sorgulamaktadır. Çalışma, bu soruya yönelik olarak, yönetişim yapılarının proje başarısı üzerinde pozitif bir etkisi olduğunu doğrulamıştır. Yönetişim yapılarının, proje süreçlerini düzenleyici ve destekleyici rol oynadığı, bu sayede proje yönetimi metodolojilerinin daha etkin kullanılmasını sağladığı görülmüştür. İkinci araştırma sorusu, proje yönetişiminin, proje yönetimi metodolojisi ve başarı arasındaki ilişkide bir moderatör rolü oynayıp oynamadığını ele almaktadır. Bu soru kısmen desteklenmiştir. Çalışma proje yönetişiminin bazı durumlarda moderatör olarak işlev görebileceğini, ancak bu etkinin her zaman belirgin olmadığını ortaya koymuştur. Özellikle, yönetişim yapılarının, uygulanan proje yönetimi metodolojileri ve proje başarısı arasındaki ilişkiye indirgenemeyen bir etkisi olduğu, bu etkinin duruma göre değişebileceği belirtilmiştir. Joslin ve Müller (2015)'in bu çalışması, proje yönetimi uygulamalarının ve başarılarının nasıl optimize edilebileceği konusunda proje yönetimi literatürüne katkı sağladığı vurgulanmaktadır.

Joslin ve Müller (2016) çalışması, proje yönetimi ve proje başarısı arasındaki ilişkiyi, özellikle proje yönetişiminin bu ilişkideki rolünü değerlendirmek için yapılmıştır. Çalışma, proje yönetişimi modellerini beş başarı kavramı (proje verimliliği, kurumsal faydalar, proje etkisi, paydaş memnuniyeti, gelecek potansiyeli) ve iki yönetişim kavramı (hissedar odaklılığa karşı paydaş odaklılık ve davranışa karşı çıktılarının kontrolü) üzerinden incelemiştir. Bu çalışma, dünya çapında yapılan çapraz kesitsel bir çevrimiçi anket aracılığıyla veri toplamıştır. Anket, 254 kişiden toplanmış ve faktör analizi ile regresyon analizleri uygulanmıştır. Bu analizler, proje yönetişiminin proje

başarısı ile küçük ancak anlamlı bir korelasyona sahip olduğunu göstermiştir. Özellikle, yönetim modellerinin paydaş odaklılığı, proje başarısıyla pozitif bir ilişki içindedir. Ancak yönetişimin davranışsal kontrolü ve çıktı kontrolü arasındaki ilişki proje başarısıyla anlamlı bir korelasyon göstermemiştir.

Engelbrecht ve diğ. (2017), işletme yöneticilerinin BT yetkinliği bileşenleri ile BT proje başarısı arasında bir ilişki olup olmadığını araştırmıştır. Yetkinlik iki ana kavramla açıklanmıştır. Bunlar, BT bilgisi ve BT deneyimidir. BT bilgisi; işletme yöneticilerinin- teknoloji bilgisi, uygulama bilgisi, sistem geliştirme bilgisi, BT yönetimi bilgisi, BT bilgisine erişim olarak tanımlanmıştır. BT deneyimi; işletme yöneticilerinin- BT projelerinde deneyimi, BT yönetiminde deneyimi erişim olarak tanımlanmıştır. BT yetkinliğinin iki ana bileşenini içeren yedi bağımsız değişkenle bir anket yapılmıştır. Ankete dünya çapında müdür seviyesinde 108 kişi katılmıştır. Yapısal eşitlik modellemesi kullanılmıştır. BT bilgisi kavramında, işletme yöneticilerinin BT yetkinliğinin genel etkisi önemli bulunmuştur. Özellikle işletme yöneticilerinin uygulamalar hakkındaki bilgilerinin BT proje başarısı üzerinde önemli bir etkisi olduğu belirlenmiştir.

Perkusich ve diğ. (2017) tarafından yapılan çalışma, scrum projelerinin sürekli iyileştirilmesine yardımcı olacak metrikler ve Bayes ağları kullanarak bir model geliştirmiştir. Çalışma, literatür taraması ve alan uzmanlarından elde edilen bilgilerle bir Bayes ağı oluşturarak başlamıştır. Bayes ağı, Scrum projelerinin kalitesini değerlendirmede ve iyileştirmede kullanılmak üzere tasarlanmıştır. Bu çalışma, iki aşamalı bir süreç izlemiştir. İlk aşamada, literatürden ve uzman görüşlerinden yararlanarak Bayes ağının temelini oluşturan metrikler belirlenmiş ve bu metrikler arasındaki ilişkiler tanımlanmıştır. Bu metrikler; ürün kalitesi, günlük scrum kalitesi, geliştirme takımı yetkinliği, yazılım mühendisliği teknikleri kalitesi, kod kalitesi, sprint planlama kalitesi, sprint & iş listesi kalitesi, ürün sahibi genel özellikleri, çalışma doğrulama kalitesi, ürün vizyon kalitesi, ürün iş listesi yönetim kalitesi, ürün iş listesi sıralama kalitesi kavramlarını içermektedir. İkinci aşamada ise bu Bayes ağının çıkarımları on adet simüle edilmiş senaryo ile doğrulanmış ve ağın ilk versiyonundaki sınırlamalar dikkate alınarak güncellenmiştir. Analiz sonuçları, Bayes ağının Scrum projelerini yeterli bir şekilde temsil ettiğini ve scrum master'ların projeleri daha etkin bir şekilde yönlendirmesine yardımcı olabilecek veriler sağladığını vurgulamıştır. Çalışmanın bulguları, tüm senaryolar için çıkarım doğruluğunun

artırıldığını ortaya koymuştur. Ayrıca çalışma, Bayes ağlarının esnekliğini ve farklı scrum takımlarına uyum sağlayabilme kapasitesi vurgulanmaktadır.

Fatema ve Sakib (2018) tarafından gerçekleştirilen çalışma, çevik yazılım geliştirme takımlarının verimliliğini etkileyen faktörleri belirlemek ve analiz etmek üzerine odaklanmıştır. Araştırma, Bangladeş'teki yazılım şirketleri üzerinde yapılan bir durum çalışması olarak tasarlanmıştır. Çalışmanın temel amacı, çevik ekip verimliliğini etkileyen ana faktörleri belirlemek ve bu faktörlerin etkileşimlerini ve yapılarını anlamaktır. Araştırma, nitel bir yaklaşım kullanarak yapılmıştır. İlk olarak literatür taraması ve alan uzmanlarıyla görüşmeler yoluyla çevik takımların verimliliğini etkileyen faktörler belirlenmiştir. Bu bilgilerle bir etki diyagramı ve nedensel döngü diyagramı oluşturulmuş, bu sayede faktörler arasındaki ilişkiler ve etkileşimler model olarak kurulmuştur. İkinci aşamada, bu model, çeşitli senaryolar kullanılarak doğrulanmış ve rafine edilmiştir. Toplamda 12 yazılım şirketinden anket için dönüş yapılmıştır. Yarı yapılandırılmış mülakatlar ve yüz yüze görüşmeler için 12 uzman ile çalışılmıştır. Çevik takım verimliliğini etkileyen bir dizi faktör ortaya konmuştur. Bunlar arasında ekip etkinliği (ekibin işbirliği ve iç dinamikleri, ekip üyeleri arasındaki etkileşim ve görevleri yerine getirme yeteneği), ekip yönetimi (ekibin yönetilme biçimi, liderlik tarzı ve yöneticilerin ekip üyeleri üzerindeki etkisi), motivasyon (ekip üyelerinin motivasyon düzeyleri, işe olan bağlılıkları ve görevlerini yerine getirme isteklilikleri) ve müşteri memnuniyeti (müşterilerin projeye ve sonuçlara olan memnuniyeti, müşteri geri bildirimlerinin ekip üzerindeki etkisi, müşteri ile etkileşim) gibi faktörler bulunmaktadır. Araştırma, bu faktörlerin çevik takım çalışmalarını nasıl etkilediğini detaylı bir şekilde açıklamış ve yöneticilerin bu faktörleri nasıl iyileştirebileceklerine dair önerilerde bulunmuştur.

Rashid ve Khan (2018) çalışmasında, çevik yöntemler kullanılarak yeşil ve sürdürülebilir yazılım geliştirilmesi için başarı faktörlerini belirlemeyi amaçlamıştır. Bu kapsamda yapılan sistematik literatür incelemesinin ardından, elde edilen bulguların doğrulanması için bir anket çalışması gerçekleştirilmiştir. Çalışma, küresel yazılım geliştirme sektöründeki uzmanlar arasında yürütülmüş ve 25 farklı ülkeden toplam 106 uzman ankete katılmıştır. Yeşil ve sürdürülebilir yazılım geliştirme süreçlerinde çevik yöntemlerin başarısını etkileyen 16 faktör tespit edilmiştir. Bunlar; hızlandırılmış teslimat, sürekli entegrasyon, sürekli doğrulama, zaman ve bilişim kaynaklarının etkin kullanımı, e-atık azaltma, değişim konusunda esneklik, ürün

yaşam döngüsünün yeşil ve sürdürülebilir yönetimi, iyileştirilmiş kalite, minimum dokümantasyon, iteratif geliştirme, minimum yeniden mühendislik çalışmaları, süreçlerin optimizasyonu, optimize edilmiş kod, polimorfik tasarım, azaltılmış maliyet, güçlü iletişim ve iş birliği olarak belirlenmiştir. Anket verileri analizinde frekans analizi kullanılmıştır. Bu analizler, başarı faktörlerinin önem sıralamalarının literatür taraması ve anket sonucu elde edilen veriler arasında tam olarak örtüşmediğini ortaya koymuştur, ancak her iki veri setinde de başarı faktörlerinin yüksek oranda olumlu geri bildirim aldığı belirlenmiştir. Örneğin, zaman ve bilişim kaynaklarının etkin kullanımı sistematik literatür taramasında birinci sırada yer alırken, anket verilerinde yedinci sırada yer almıştır. Bu çalışma, yeşil ve sürdürülebilir yazılım geliştirme pratiğinde hangi faktörlerin öne çıktığını belirleyerek ve ilgili faktörlere odaklanarak yazılım geliştirme süreçlerinin daha çevreci ve sürdürülebilir hale getirilmesine yardımcı olacağı vurgulanmıştır.

Menezes ve diğ. (2018) çalışmasında, yazılım geliştirme projelerinde risk faktörlerini belirlemek amacıyla kapsamlı bir sistematik literatür taraması gerçekleştirmiştir. Araştırma sonucunda, Software Engineering Institute tarafından geliştirilen yazılım geliştirme taksonomisine göre sınıflandırılan 148 farklı risk faktörü tespit edilmiştir. Çalışmada yer alan risk faktörleri ürün mühendisliği, program kısıtlamaları ve geliştirme ortamı olarak sınıflandırılmıştır. Ürün mühendisliği sınıfında ilk 10'da yer alan risk faktörleri; gereksinim belirsizliği, gereksinim değişiklikleri, yeni teknolojinin tanıtılması, dışarıdan temin edilen bileşenlerdeki eksiklikler/kötü arayüzler, teknik karmaşıklık, eksik gereksinimler, dokümantasyonun kalitesi, iç sistem entegrasyonunun eksikliği, yanlış fonksiyon ve özelliklerin geliştirilmesi ve testlerin eksikliğidir. Program kısıtlamaları sınıfında ilk 10'da yer alan risk faktörleri; gerekli becerilere sahip olmayan personel, kullanıcı/müşterinin eksik taahhüdü, kararsız organizasyonel ortam, kaynakların kötü tahmin edilmesi, gerçekçi olmayan zaman çizelgesi, deneyimsiz personel, yüksek personel devri, müşteri ile anlaşmazlık, üst yönetimin projeye yeterince destek vermemesi ve bütçe dalgalanmalarıdır. Geliştirme ortamı sınıfında ilk 10'da yer alan risk faktörleri; planlama eksikliği veya yetersiz planlama, personel taahhüdünün düşük olması, yetersiz disiplin ve standardizasyon, proje hedeflerinin belirsiz olması, ekip üyeleri arası iletişim eksiklikleri, kapsam değişiklikleri, proje ilerlemesinin yeterince yakından izlenmemesi, geliştirme metodolojisinin projeye uygun olmaması, ekip üyeleri arasında çatışmalar ve etkili

proje yönetim metodolojisinin eksikliğidir. Çalışma, özellikle yazılım gereksinimleriyle ilgili risk faktörlerinin en sık tekrar eden ve en çok atıfta bulunulan faktörler olduğunu vurgulamaktadır. Ayrıca personelin teknik becerilerinin eksikliğinin tüm sınıflar arasında en sık bahsedilen risk faktörü olduğu belirlenmiştir. Çalışmada yazılım geliştirme projelerinde başarısızlık düzeyini azaltmanın temel unsurları olarak bu faktörlerle ilgili daha fazla çalışma yapılması gerektiği söylenmektedir.

Abrar ve diğ. (2019), büyük ölçekli çevik yazılım geliştirme projelerinde yönetim perspektifinden motivasyon sağlayıcı faktörleri belirlemek amacıyla sistematik literatür çalışması gerçekleştirmiştir. Araştırma, çevik metodolojilerin büyük ölçekli projelerde nasıl uygulanabileceği konusunda motivasyon sağlayıcı faktörleri tanımlamak için gerçekleştirilmiş ve toplamda 58 araştırma makalesinden 21 motivasyon faktörü tespit edilmiştir. Sistematik literatür çalışması yardımıyla elde edilen faktörler; üst yönetim desteği, çevik geliştirme ortamında eğitim ve öğrenim, çevik geliştirme uzmanlığı, ekip yetkinliği, üst yönetimin çevik yöntemler hakkında bilgilendirilmesi, işbirlikçi organizasyonel kültür, müşteri memnuniyeti, müşteri ile güçlü işbirliği, sürdürülebilir planlama, otomatikleştirilmiş yazılım araçlarının kullanımı, ekip üyeleri için planlanmış eğitimler, liderliğin güçlü taahhüdü, ekip otonomisi, çevik yönelimli gereksinim yönetim süreçlerinin kullanımı, değişim yönetimi mekanizması, bilgi paylaşımı yönetimi, kaliteli üretimde çift programlama kullanımı, yönetim için adanmışlık, deneyimsizlik durumunda pilot proje, çevik geliştirme ortamı ve ekip teşviki olarak belirlenmiştir. Araştırma sonuçlarının doğruluğunu ve uygulanabilirliğini artırmak amacıyla gelecekteki çalışmalarda anket ve deneysel çalışmalar yapılması planlanmaktadır. Bu çalışmalar, teorik bulguları pratik deneyimlerle destekleyerek çevik yazılım geliştirme uygulamalarının etkinliğini artırmayı amaçlamaktadır.

Anthony (2019) çalışması, yazılım risklerinin önceliklendirilmesi için analitik hiyerarşi süreci (AHP) tabanlı bir modelin kullanılabilirlik özelliklerini doğrulamaya yönelik bir araştırmadır. Araştırma, yazılım geliştirme projelerinde risk önceliklendirme modelinin kullanılabilirliğini ve etkinliğini değerlendirmek için yapılmıştır. Çalışma, bir literatür taraması ve anket çalışması içermektedir. Araştırmanın birincil yöntemi, yazılım risk faktörlerinin nasıl değerlendirildiğini ve önceliklendirildiğini göstermek için geliştirilen AHP tabanlı yazılım risk

önceliklendirme modelinin kullanılabilirliğini doğrulamak amacıyla tasarlanmış bir anketten elde edilen verilerin analizidir. Anket, Malezya'daki yazılım uygulayıcılarına uygulanmış ve toplamda 108 katılımcıdan veri toplanmıştır. Analiz için yapısal eşitlik modellenmesi kullanılmıştır. Bu analiz, modelin yazılım risk faktörlerini önceliklendirme konusunda yüksek bir kullanılabilirlik düzeyine sahip olduğunu göstermiştir. Ayrıca modelin özellikle teknik beceri eksiklikleri gibi belirli risk faktörlerini etkili bir şekilde ele aldığı tespit edilmiştir. Modelde ele alınan bazı öncelikli risk faktörleri şunlardır: teknik yeteneklerin eksikliği, yetersiz proje yönetimi, değişen müşteri gereksinimleri, zaman ve bütçe sınırlamalarıdır. Araştırma, yazılım risklerini değerlendirme ve önceliklendirme konusunda AHP yaklaşımı kullanarak risk faktörü seçimi yapılmasını sağlayan bir model geliştirmiştir. Pratik uygulamalar açısından bu çalışmanın yazılım şirketlerine, yazılım risklerini değerlendirme ve önceliklendirme konusunda rehberlik eden risk bilgileri sağladığı belirtilmiştir.

Arcos ve diğ. (2019) çalışması kalite odaklı kapsamlı bir sistematik literatür araştırmasını içermektedir. Bu çalışmada kritik başarı faktörleri, kaliteyi ölçmek için ölçütler, kalite nitelikleri, çevik uygulamalar, kalite modelleri ve çevik ilkeler belirlenmiştir. Sistematik literatür araştırması ile belirlenen 71 çalışma sonucunda toplam 118 kritik başarı faktörü, 137 çevik uygulama, 102 ölçüt ve 70 kalite niteliği elde edilmiştir. Çalışma sonucunda elde edilen kritik başarı faktörlerinden bazıları; güçlü yönetici desteği, hiyerarşik olmayan işbirlikçi organizasyonel kültür, yüz yüze iletişime büyük değer veren sözlü kültür, iletişim ve müzakere, müşterilerle iletişim, çevik metodolojinin evrensel olarak kabul edilmesi, tüm takımın bir arada bulunması, uygun çevik tarzı çalışma ortamına sahip organizasyon, çevik geliştirme için uygun ödül sistemi, yeniliği destekleme, paydaşlarla güçlü ilişkiler, uyumluluk, müşteri memnuniyeti, müşteri işbirliği, müşteri taahhüdü, eğitimler, tecrübe, modüler tasarım olarak belirtilmiştir. Proje başarı ölçütlerinden bazıları; yazılım geliştirme yaşam döngüsü aşamalarına harcanan çaba, eşli programlamanın gerekçesi, müşteriler tarafından gönderilen hatalar, müşteriler tarafından istenen değişiklikler, bir müşteriden elde edilen gelir miktarı, ertelenen hatalar sayısı, kodlama standartları sapmalarının yoğunluğu, olgunluk seviyesi yüksek ekip, sürüm süresi, iterasyon süresi, iterasyon sayısı, harcanan geliştirme çabası, eklenen ürün iş listesi öğeleri yüzdesi, değiştirilen ürün iş listesi öğeleri yüzdesi, reddedilen ürün iş listesi öğeleri

yüzdesi, tahmin kayması, öncelik kayması, zaman tahmini, sürüm dönemi, sprint dönemi olarak belirtilmiştir. Sistemik literatür taraması sonucunda bulunan çevik pratiklerden birkaçı; küçük ekipler, çok fonksiyonlu ekipler, günlük toplantı, hedef toplantıları, eşli programlama aktiviteleri, ekip içi iletişimde çeviklik, insanların işbirliği yetenekleri, ekip iletişim yetenekleri, kişilerin yetkinliği, ekip üyeleri arasında güven, liderlerin katılımı, lider tanıma, müşteri ve geliştiriciler arasında günlük işbirliği, müşterilerle etkileşim sıklığı, müşteri motivasyonları, sorumluluk ve yetki, vardiyalı çalışma, kendi kendini organize eden ekipler, profesyonellerin eğitimi, müşteri memnuniyeti ve yazılım projelerinde yenilik yönetimi olarak belirtilmiştir. Çalışma sonucunda elde edilen kalite niteliklerinden bazıları; doğruluk, sağlamlık, genişletilebilirlik, uyumluluk, verimlilik, zamanında, doğrulama ve onay, kullanım kolaylığı, maliyet etkinliği, sadelik, genellik, çalışma zamanında modülerlik, ölçeklenebilirlik, esneklik, etkililik, üretkenlik, yapılabilirlik, performans, yararlılık, karmaşıklık olarak belirtilmiştir.

Perera ve Perera (2019) çalışmasında, Sri Lanka'daki yazılım endüstrisinde müşteri katılımının çevik proje başarısı üzerindeki etkilerini incelemiştir. Çalışmada 369 katılımcıdan toplanan verileri analiz etmek için güvenilirlik analizi, faktör analizi ve regresyon analizi kullanmıştır. Çalışmada müşteri katılımını temsil eden bağımsız değişkenler ve çevik projelerin başarısını etkileyen faktörler; müşteri iletişim düzeyi, müşteri işbirliği, müşteri etkisi, müşteri endüstri bilgisi ve müşteri esnekliği olarak belirtilmiştir. Ayrıca organizasyon kültürü moderatör değişken olarak kullanılmıştır. Analiz sonuçları, müşteri katılımının çeşitli yönlerinin çevik projelerin başarısı üzerinde pozitif bir etkisi olduğunu ortaya koymuştur. Araştırmada test edilen tüm hipotezler kabul edilmiştir.

Hughes ve diğ. (2020) çalışmasında, bilgi sistemleri proje başarısı faktörlerini incelemektedir. Başarı faktörlerinin birbiriyle olan bağımlılıklarını ve etkileşimlerini anlamak için yorumlayıcı yapısal modelleme kullanarak kavramsal bir araştırma yapılmıştır. Çalışma Birleşik Krallık'ta gerçekleştirilmiş ve uzman katılımcıların görüşleriyle desteklenmiştir. Araştırmanın veri seti, bilgi teknolojileri alanında çalışan dokuz uzmanın katılımıyla elde edilmiş görüşler üzerinden oluşturulmuştur. Bu uzmanlar, çeşitli sektörlerde (finans, kamu sektörü, danışmanlık, ve Savunma Bakanlığı), çeşitli rollerde (proje ve değişim yönetimi, iş analizi ve sistem geliştirme) görev almaktadırlar. Katılımcılar, başarı faktörleri arasındaki potansiyel ilişkiler

üzerine yapılandırılmış tartışmalara katılmışlar ve bu faktörler arasındaki bağlantıları değerlendirmişlerdir. Çalışmada elde edilen başarı faktörleri; katılımcı ve taahhütlü sponsorluk, projedeki kullanıcı katılımı, yetenekli kaynakların kullanılması, proje yöneticisinin becerileri, tecrübesi ve stili, proje kapsamının yönetimi, proje denetim sürecinin yer alması, proje yönetim metodolojisinin kullanılması, net iş durumunun olması, direnç yönetim süreci, kuruluşun proje olgunluğu, formalize edilmiş rol tanımları, gerekli araçlar ve altyapı, resmi risk yönetimi, kısa geliştirme süresi, etkili fayda yönetim süreci, entegre değişim ve proje yönetimi, teknik bir arızayla sonuçlanan olayların yönetim süreci olarak belirlenmiştir. Bu çalışma, bilgi sistemlerinde başarı faktörleri arası ilişkilerin modellemesinin, diğer bağlantılı faktörlere potansiyel etkisini gösterebileceğini ve böylece proje başarısının şansını artırabileceğini vurgulayan yorumlayıcı bir yaklaşım göstermiştir.

Kanjanda ve Tuan (2020) çalışmasında, Zimbabwe'deki BT projelerini tehdit eden risk faktörlerini belirlemeyi ve tanımlanan risk faktörlerinin birbirleriyle nasıl bağlantılı olduğunu gösteren bir sistemsel model oluşturmayı amaçlamaktadır. Çalışma, sistemsel düşünceye dayanarak, risk faktörlerini tanımlamak için Etkileşimli Yönetim adında bir yaklaşım benimsemiştir. Etkileşimli Yönetim oturumu, dört ana aşamada ilerlemiştir: fikir üretimi, fikir açıklaması, fikir yapılandırması ve yapılandırılan fikirlerin yorumlanması. Bu sürecin çıktısı, tanımlanan riskler arasındaki nedensel ilişkileri gösteren bir yönlendirme grafiği çizilmesidir. Zimbabwe'de gerçekleşen çalışmada ilgili paydaşların etkileşimli bir şekilde sistemsel modeli üretmeleri istenmiştir. Etkileşimli Yönetim süreci sonucunda altı risk faktörü tanımlanmıştır: sınırlı bilgisayar okuryazarlığı, zayıf iletişim, yeterli yönetici desteğinin olmaması, tasarım karmaşıklığı, bürokrasi ve çalışan devir hızıdır. Araştırmanın amacı, Zimbabwe'deki BT projelerinde risk yönetiminin önemini vurgulamak ve bu projelerde başarısızlık oranını düşürmeye yönelik stratejiler geliştirilmesi için katkıda bulunmaktır.

Tam ve diğ. (2020) çalışmasının amacı devam eden çevik yazılım geliştirme projelerinin başarısını etkileyen insan faktörlerini belirlemek ve modellemektir. Bu çalışma, Portekiz'de gerçekleştirilmiş ve çevik metodolojilerle çalışan 216 profesyonel üzerinden anket verileri toplanmıştır. Anket verileri yapısal eşitlik modellemesi kullanılarak analiz edilmiştir. Çalışma, beş insan faktörünü (kişisel özellikler, eğitim ve öğrenme, toplumsal kültür, ekip kabiliyeti ve müşteri katılımı)

çevik yazılım geliştirme projelerinin başarısını etkileyen temel unsurlar olarak tanımlamıştır. Bu faktörler, projenin maliyet, zaman ve müşteri memnuniyeti açısından başarısını ölçmekte kullanılmıştır. Çalışmanın bulgularına göre, ekip kabiliyeti ve müşteri katılımı devam eden çevik projelerin başarısına önemli katkıda bulunan ana faktörler olarak belirlenmiştir. Çalışmanın analiz sonuçları, projenin başarısı üzerinde doğrudan etkili olan ekip kabiliyeti ve müşteri katılımı dışında, kişisel özellikler ve toplumsal kültür gibi diğer faktörlerin de dolaylı yollarla çevik projenin başarısını etkileyebileceğini göstermiştir. Bu bağlamda çalışma aynı zamanda bu insan faktörlerinin, projenin başarısına olan dolaylı etkilerini de ortaya koymuştur.

Rashid ve diğ. (2021), küresel yazılım geliştirme için Yeşil Çevik Olgunluk Modeli geliştirmiştir. Yeşil Çevik Olgunluk Modelini oluşturmak için çalışma dört aşamadan oluşmuştur. Faz 1’de yeşil ve sürdürülebilir yazılım geliştirilmesinde kullanılan çevik metodolojilerle ilişkili başarı ve risk faktörlerini belirlemek amacıyla bir sistematik literatür taraması yapılmıştır. Bu tarama, çevik metodların yeşil yazılım geliştirmedeki etkisini destekleyen başarı faktörlerini ortaya çıkarmak için yapılmıştır. SLT sonucunda elde edilen başarı faktörleri; hızlandırılmış teslimat, sürekli entegrasyon, sürekli doğrulama, zaman ve bilişim kaynaklarının etkin kullanımı, e-atık azaltma, değişim konusunda esneklik, ürün yaşam döngüsünün yeşil ve sürdürülebilir yönetimi, iyileştirilmiş kalite, minimum dokümantasyon, iteratif geliştirme, minimum yeniden mühendislik çalışmaları, süreçlerin optimizasyonu, optimize edilmiş kod, polimorfik tasarım, azaltılmış maliyet, güçlü iletişim ve iş birliği olarak belirlenmiştir. Faz 2’de sistematik literatür taramasının sonuçlarına dayanarak, 25 farklı ülkeden 106 çevik yazılım projeleriyle ilgilenen uzmanın katılımıyla bir anket çalışması gerçekleştirilmiştir. Bu anket, Faz 1’de belirlenen faktörlerin doğrulamasını yapmıştır. Faz 3’de önceden tanımlanan başarı faktörleri, Yeşil-Çevik Olgunluk Modeli’ne dahil edilmek üzere yedi yeşil-çevik olgunluk seviyesine kategorize edilmiştir. Bu model, çevik ve yeşil yazılım geliştirme uygulamalarını entegre ederek, sürdürülebilir yazılım geliştirmeyi teşvik etmeyi amaçlamaktadır. Faz 4’de, Yeşil-Çevik Olgunluk Modeli’nin yazılım endüstrisindeki yapısını ve etkinliğini değerlendirmek için beş vaka çalışması yapılmıştır. Bu vaka çalışmaları, modelin pratikteki uygulanabilirliğini ve çevik-yeşil olgunluğu ölçme kapasitesini test etmek için tasarlanmıştır. Faz 4’de gerçekleştirilen beş vaka çalışması, Yeşil-Çevik Olgunluk Modeli’nin yapısal ve etkililik değerlendirmesi için yapılmıştır. Bu vaka çalışmaları, farklı küresel yazılım

geliştirme organizasyonlarında Yeşil-Çevik Olgunluk Modeli'nin uygulanabilirliğini ölçmüş ve modelin gerçek endüstri ortamında nasıl işlediğini gözlemlemiştir. Vaka çalışmaları sonucunda elde edilen bulgular, Yeşil-Çevik Olgunluk Modeli'nin, küresel yazılım geliştirme organizasyonlarında yeşil ve sürdürülebilir yazılım geliştirme konusunda çevik olgunluklarını değerlendirmede kullanışlı olduğunu doğrulamıştır. Bu sonuçlar, farklı şirketlerin Yeşil-Çevik Olgunluk Modeli'ni kullanarak yeşil-çevik olgunluk seviyelerini başarıyla ölçtüklerini ve bu değerlendirmelerin, modelin yapısal sağlamlığı ve endüstriyel uygulanabilirliği açısından olumlu geri bildirimlerle desteklendiğini göstermektedir.

Literatür taraması sonucunda çevik projeler için kritik başarı faktörlerini belirlemeye yönelik kapsamlı çalışmalar bulunmaktadır (Chow ve Cao, 2008; Misra ve diğ., 2009; Wan ve Wang, 2010; Stankovic ve diğ., 2013; Silva ve Santos, 2015; Ahimbisibwe ve diğ., 2015, 2017; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019; Garousi ve diğ., 2019). Chow ve Cao (2008), literatürden çevik projeler için kritik başarı faktörlerinin bir ön listesini oluşturmuş ve daha sonra bu faktörleri beş kavramda gruplandırmıştır: organizasyon, insanlar, süreçler, teknik ve proje. Çevik proje başarı faktörlerinin proje başarısı üzerindeki etkileri, 25 farklı ülkedeki 109 çevik yazılım projesinden toplanan verilerle analiz edilmiştir. Stankovic ve diğ. (2013) çalışmasında, Chow ve Cao (2008) tarafından sunulan kritik başarı faktörleri modelini kullanmış ve kritik başarı faktörlerinin etkilerini analiz etmiştir. Ahimbisibwe ve diğ. (2015) yazılım geliştirme projeleri için literatürden toplam 37 kritik başarı faktörü çıkarmış ve bu faktörleri üç gruba ayırmıştır: müşteri, organizasyonel ve ekip faktörleri. Daha sonra geleneksel proje yönetimi metodolojilerini çevik olanlarla karşılaştırmak için KBF'ler ile bir olumsuzluk uyum modeli geliştirmişlerdir. Ahimbisibwe ve diğ. (2017), dış kaynaklı yazılım geliştirme projeleri için geleneksel proje yönetimi ve çevik metodolojileri kritik başarı faktörleri açısından karşılaştırmak için Ahimbisibwe ve diğ. (2015) tarafından sunulan uyum modelini kullanmıştır. Garousi ve diğ. (2019), yazılım projesinin başarısı için kritik başarı faktörlerinin göreceli önemini değerlendirmek için aynı modeli kullanmıştır.

Kritik başarı faktörleri ile ilgili literatür incelendiğinde, bazı çalışmalarda örgütsel ve insanlarla ilgili faktörler (Misra ve diğ., 2009); insanlar, süreç ve teknolojiyle ilgili faktörler (Silva ve Santos, 2015); çevik metodolojiler kullanarak yeşil ve sürdürülebilir yazılım geliştirmeye yönelik başarı faktörleri (Rashid ve Khan, 2018); müşteri ile ilgili

faktörler (Perera ve Perera, 2019); çevik geliştirme süreci üzerindeki kaliteyle ilgili faktörler (Arcos-Medina ve Mauricio, 2019); ve insanlarla ilgili faktörler (Tam ve diğ., 2020) gibi belirli kavramlarla sınırlı kaldığı görülmektedir. Misra ve diğ. (2009) çevik yazılım geliştirme uygulamalarının benimsenmesiyle ilişkili başarı faktörlerini tanımlamış, örgütsel ve insanlarla ilgili faktörlere odaklanmıştır. Silva ve Santos (2015) çevik yazılım projeleri için kritik başarı faktörlerini insan, süreç ve teknoloji kavramları olarak gruplandırmıştır. Her bir kavram bir dizi öge ile açıklanmıştır. Rashid ve Khan (2018) küresel ölçekte faaliyet gösteren yazılım geliştirme tedarikçileri için çevik metodolojiler kullanarak yeşil ve sürdürülebilir yazılım geliştirmeye odaklanmıştır. Perera ve Perera (2019) müşteri iletişim düzeyi, müşteri işbirliği gibi müşteri katılımının çevik proje başarısı üzerindeki etkisini araştırmıştır. Arcos-Medina ve Mauricio (2019) kritik başarı ölçütlerini, kalite niteliklerini ve çevik uygulamaları belirlemek için çevik geliştirme sürecinde kalite üzerine sistematik bir literatür taraması yapmıştır. Tam ve diğ. (2020) çevik yazılım geliştirme projesinin başarısını etkileyen müşteri katılımı, eğitim ve öğrenme, toplumsal kültür, ekip kabiliyeti ve kişisel özellikler gibi insanlarla ilgili faktörlere odaklanmıştır.

Ayrıca karmaşık sistemlerin geliştirilmesinde, risk tanımlama sürecinin eksikliği proje başarısızlığının başlıca nedenlerinden biridir (Reeves ve diğ., 2013). Dolayısıyla risk ve başarısızlık faktörleri yazılım geliştirme projelerinin başarısını önemli ölçüde etkilemektedir. Önceki yapılan çalışmalarda risk faktörleri ve bunların yazılım geliştirme projeleri üzerindeki etkileri incelenmiştir (Shrivastava ve Rathod, 2015, 2017; Menezes ve diğ., 2019; Kanjanda ve Tuan, 2020). Shrivastava ve Rathod (2015, 2017) çalışmalarında dağıtık çevik yazılım geliştirme projelerinin performansını etkileyen bir dizi risk faktörü elde edilmiş ve bir risk yönetimi çerçevesi geliştirilmiştir. Chow ve Cao (2008) ve Stankovic ve diğ. (2013) sadece çevik KBF'leri değil, aynı zamanda yazılım projesi başarısızlıklarının ana nedenlerini de incelemişlerdir.

Bu çalışmalar, Chow ve Cao (2008), Misra ve diğ. (2009), Wan ve Wang (2010), Stankovic ve diğ. (2013), Silva ve Santos (2015), Ahimbisibwe ve diğ. (2015), (2017), Rashid ve Khan (2018), Arcos-Medina ve Mauricio (2019), Garousi ve diğ. (2019), Perera ve Perera (2019) ve Tam ve diğ. (2020), çevik projeler için kapsamlı bir başarı faktörleri listesi sunsa da, tez kapsamında yapılan bu çalışmada çevik kritik başarı faktörleri ile yazılım projesi başarı ölçütleri arasındaki ilişkileri kapsamlı bir bakış

açısıyla açıklayan bir Çevik Yazılım Projesi Başarı Modeli önerilmektedir. Ayrıca literatür taramasına ek olarak, literatürde bahsedilmeyen olası faktörleri ortaya çıkarmak için Çevik Manifesto, Çevik İlkeler ve Scrum Kılavuzu ayrıntılı olarak incelenmiştir. Model ve faktörler, yazılım geliştirme sektöründe çalışan çevik uygulayıcılar tarafından doğrulanmıştır. Dolayısıyla mevcut çalışma yeni bir model oluşturarak literatüre katkıda bulunmakta, çevik kritik başarı faktörlerini müşteri faktörleri, ekip faktörleri, organizasyonel faktörler, çevik süreç faktörleri, teknik faktörler ve proje faktörleri (çevik risk ve başarısızlık faktörleri) olarak gruplandırırken, çevik proje başarı ölçütlerini süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti olarak gruplandırmaktadır. Ayrıca modelde tanımlanan ilişkiler 596 çevik uygulayıcıdan veri toplanarak test edilmiş ve doğrulanmıştır. Katılımcılar farklı demografik özelliklere sahiptir. Farklı ülkelerden katılan katılımcılar farklı çalışma modellerine (uzaktan çalışma, hibrit çalışma ve ofis tabanlı çalışma), farklı meslek dallarına (yazılımcı, iş analisti, test uzmanı ve diğerleri) ve farklı deneyim seviyelerine sahiptir.

3. ARAŞTIRMA METODOLOJİSİ

Mevcut çalışmada kullanılan araştırma metodolojisi Şekil 3.1’de gösterilmektedir. Araştırma modelinde ilk adımda Sistemik Literatür Taraması (SLT) yaklaşımı ile çevik yazılım projesi başarı modelinin ilk versiyonu geliştirilmiştir. Başarı modelinde yer alan faktörlerin geliştirilmesi, çevik metodolojiler konusunda uzman altı uygulayıcı ile bire bir toplantılar yapılarak titiz bir süreçle gerçekleştirilmiştir. Daha sonrasında çevik yazılım projesi başarı modelinde fikir birliğine varmak için uzman kişilerle toplu toplantı yapılmıştır. Sonrasında geliştirilen Çevik Yazılım Proje Modeli’nde belirtilen ilişkileri test etmek için Çevik Yazılım Proje Anketi literatürde var olan ögeler dikkate alınarak oluşturulmuştur. Nihayetinde 596 çevik uygulayıcıdan anket yoluyla elde edilen veri seti kullanılmış ve böylece araştırma modelinde tanımlanan ilişkiler açıklığa kavuşturulmuştur.

Şekil 3.1’de gösterildiği gibi çalışma modelinin ilk aşamasında “Kritik Başarı Faktörleri ve Başarıyı Tanımlayan Ölçütlerin Belirlenmesi” yer almaktadır. Bu kapsamda tez çalışmasında ilk olarak sistemik literatür taraması yapılmıştır. Sistemik literatür taraması, belirlenen araştırma sorularına yanıt aramak ve çalışmaya objektiflik sağlamak için yapılan kapsamlı bir literatür inceleme sürecini içermektedir. SLT, araştırma alanında mevcut bilgi birikimini sistemik ve tekrarlanabilir bir biçimde incelemeyi amaçlamaktadır. Tez kapsamında Kitchenham ve Charters (2007) tarafından önerilen SLT yaklaşımı kullanılmıştır. Literatür taraması için öncelikle tez çalışmasına ait araştırma soruları belirlenmiştir. Araştırma soruları, literatür taramasının odak noktasını belirlemektedir ve araştırma sürecinin yönünü çizmektedir. İkinci adım, veri kaynaklarının belirlenmesidir. Veri kaynaklarının belirlenmesi adımı, kullanılacak literatürün hangi veri tabanlarından elde edileceğini belirlemektedir. Güvenilir ve geçerli veri tabanlarının seçimi, çalışmanın kalitesini doğrudan etkilemektedir. Bu yüzden bu adımda titizlikle veri tabanları seçilmiştir. Adım 3, arama kriterlerinin belirlenmesidir. SLT’de kullanılacak anahtar kelimeler, tarih aralıkları gibi kriterler bu aşamada tanımlanmıştır. Arama kriterleri, araştırma sorularına en uygun literatürün saptanmasında önemli bir rol oynamaktadır. Adım 4,

makale seçim sürecidir. Elde edilen literatür arasından, belirlenen ekleme ve çıkarma kriterlerine göre makale seçimi yapılmıştır. Makale seçim süreci, araştırmanın objektifliğini artırmaktadır. Adım 5 ise seçilen makalelere ait yayınların incelenmesi adımıdır. SLT kapsamında seçilen makalelerin yayınlandıkları dergilerdeki dağılımı, yayınlandıkları yıllara göre dağılımı, seçilen makalelerin atıf sayıları, kalitesi gibi kavramlar titizlikle analiz edilmiştir. Bu aşamada, seçilen makalelerin bilimsel güvenilirlik ve geçerlilik açısından değerlendirilmesi yapılmıştır. SLT, tez çalışmasının güvenilirliğini artırmak için çok önemlidir. Sistematik literatür taraması süreci, dördüncü bölümde detaylı bir şekilde sunulmaktadır.

Şekil 3.1’de gösterildiği gibi, SLT aşamasından sonra çevik yazılım projelerinde başarıyı belirleyen faktörleri daha iyi analiz etmek amacıyla, çevik uygulayıcılarla bire bir ve grup görüşmeleri gerçekleştirilmiştir. Bu görüşmeler, çevik yazılım başarı modelinin geliştirilmesi için kritik öneme sahiptir. Bire bir görüşmelerde seçilen altı çevik uygulayıcı ile çevik uygulamaların kullanımı, çevik prensipler ve etkin yazılım test süreçleri gibi konular ele alınmıştır. SLT sonucunda elde edilen çevik kritik başarı faktörleri ve başarı ölçütleri detaylıca tartışılmıştır. Grup görüşmesinde ise, başarı ölçütleri ve KBF’lerin son hali belirlenerek, Çevik Yazılım Başarı Modeli oluşturulmuştur. Çevik uygulayıcılar ile yapılan görüşmeler, çevik uygulamaların farklı yönlerinin anlaşılmasını sağlamış ve Çevik Yazılım Başarı Modelinin daha etkili bir şekilde oluşturulmasına yardımcı olmuştur. Çevik uygulayıcılarla yapılan görüşmeler ve Çevik Yazılım Başarı Modeli, beşinci bölümde detaylı bir şekilde tartışılmaktadır.

Şekil 3.1’de gösterildiği üzere, anket hazırlama süreci, araştırma modelinin pratikteki uygulanabilirliğini ve geçerliliğini test etmek için önemli bir süreçtir. Bu süreç, “Çevik Yazılım Projesi Başarı Modeli” gibi teorik modellerin, gerçek verilerle doğrulanmasını sağlamaktadır. Literatürden alınan referanslarla (ISO, 2011; Stankovic ve diğ., 2013; Ahimbisibwe ve diğ., 2017; Garousi ve diğ., 2019; Hughes ve diğ., 2020) desteklenen ve çevik KBF’leri ile başarı ölçütlerini içeren 48 soruluk bir anket, modelin kapsamlı bir değerlendirmesini sunmaktadır. Bu sorular aracılığıyla, çeşitli demografik özelliklere sahip çevik uygulayıcıların model üzerindeki kavram ve öğelerle ilgili algıları ölçülmüştür. Anket hazırlama ve uygulama süreci, çevik yazılım projelerinde başarıyı etkileyen faktörleri belirlemek ve önerilen modelin pratikteki

uygulamalarına ışık tutmak açısından kritik olduğu düşünülmektedir. Anket soruları EK B’de sunulmaktadır.

Şekil 3.1’de de gösterildiği üzere, anket yoluyla veriler toplandıktan sonra “Çevik Yazılım Projesi Başarı Modeli” veri analizi süreci devam ettirilmiştir. Gözlemlenen değişkenler arasındaki ilişkileri keşfetmek ve bu değişkenleri daha az sayıda temel faktöre indirgemek için açıklayıcı faktör analizi kullanılmıştır. Açıklayıcı faktör analizi ile Çevik Yazılım Başarı Modeli’nde kullanılan çevik kritik başarı faktörleri ve başarı ölçütleri altında yatan faktörlerin belirlenmesi amaçlanmıştır. Açıklayıcı faktör analizi, bu ölçütlerin birbirleriyle olan ilişkilerini anlamak ve hangi ölçütlerin birlikte gruplandığını belirlemek için tez çalışması kapsamında kullanılmıştır. Açıklayıcı faktör analizine ilişkin detay bilgiler, altıncı bölümde sunulmaktadır.

Açıklayıcı faktör analizinden sonra önceden tanımlanmış faktör yapısının veri seti ile ne kadar uyumlu olduğunu test etmek amacıyla doğrulayıcı faktör analizi yapılmıştır. Tez çalışması kapsamında, Çevik Yazılım Başarı Modeli gibi kapsamlı bir modelde, teorik olarak önerilen faktör yapılarının veri ile uyumunu doğrulamak için doğrulayıcı faktör analizi kullanılmıştır. Modelin yapısal bütünlüğünü ve ölçüm modellerinin güvenilirliğini sağlamlaştırmak için doğrulayıcı faktör analizinin yapılması önemlidir. Ayrıca faktörler arası korelasyonları ve her bir değişkeninin kendi faktörüne olan katkısını net bir şekilde ortaya konulması çalışmanın değerleri arasındadır. Doğrulayıcı faktör analizine ilişkin bilgiler altıncı bölümde gösterilmektedir.

Yapısal eşitlik modellemesinde ölçüm modelleri ve yapısal modeller arasındaki ilişkileri eş zamanlı olarak değerlendirerek, değişkenler arasındaki karmaşık ilişkiler modellenabilir. Tez çalışması kapsamında, doğrulayıcı faktör analizi ile doğrulanmış faktör yapısını kullanarak, bu faktörler arasındaki ilişki test edilmiştir. Çevik Yazılım Başarı Modeli’nde, çeşitli başarı faktörlerinin çevik proje başarısı üzerindeki etkisini anlamak için yapısal eşitlik modeli uygulanmıştır. Yapısal eşitlik modellemesi aracılığıyla teorik olarak önerilen hipotezlerin doğruluğu test edilmiştir. Çevik Yazılım Başarı Modeli’nin analizine ilişkin çalışmalar altıncı bölümde detaylı bir şekilde sunulmaktadır.

Çevik Yazılım Başarı Modelinin analizinde en son aşamada, sektörel farklılıkların ve iş rollerinin çevik yazılım projelerinin başarısı üzerindeki etkisini incelemek için çok değişkenli varyans analizi (MANOVA) testi yapılmıştır. Sektörel farklılıklar ve iş

rolleri gibi deęiřkenler, evik yazılım projelerinin başarısı üzerinde karmařık etkiler yaratabilir. Finans sektr ve dięer sektrler gibi baęımsız deęiřkenlerin paydař memnuniyeti, srdrlebilir yazılım rn kalitesi ve sre verimlilięi gibi baęımlı deęiřkenler zerindeki etkilerini deęerlendirmek iin MANOVA uygulanmıřtır. alıřmalarda demografik farklılıklardan kaynaklanabilecek etkilerin test edilmesi nemlidir. Finans sektr gibi sıkı denetim ve dzenlemelere tabi sektrlerin zel ihtiyaları ve iř rollerinin projeler zerindeki etkilerini belirlemek, daha etkili evik uygulama stratejileri ve politikalar geliřtirilmesine yardımcı olacaęına inanılmaktadır. ok deęiřkenli varyans analizi, bu tr stratejik kararlar iin gerekli ięrrleri saęlamaktadır. evik yazılım projelerinde sektrel ve iř rol farklılıklarının etkisine ynelik yapılan MANOVA alıřması altıncı blmde sunulmaktadır.



1. KRİTİK BAŞARI FAKTÖRLERİ VE BAŞARIYI TANIMLAYAN ÖLÇÜTLERİN BELİRLENMESİ

- **Literatür Taraması** - Science Direct, IEEE, SPRINGER, WILEY ve EMERALD veritabanları taranmıştır
- **Bire bir Görüşme**
 - *Faktör ve Kriterlerin belirlenmesi*
- **Toplu Görüşmeler**
 - *Faktör ve Kriterlerin belirlenmesi*
- **Faktör ve başarı ölçütlerinin yalınlaştırılarak nihai modelin oluşturulması**



2. ANKET UYGULAMASI

- **Demografik bilgi**
- **Çevik başarı faktörlerinin incelenmesi**
- **Çevik başarıyı tanımlayan ölçütlerin incelenmesi**
- **Kullanılan çevik yazılım metodolojisi hakkında bilgilendirme**
- **Metin alan**



3. UYGULANAN TEKNİKLER

- **Açıklayıcı Faktör Analizi**
- **Yapısal Eşitlik Modellemesi**
- **MANOVA**

Şekil 3.1 : Çalışma modeli.



4. SİSTEMATİK LİTERATÜR TARAMASI

SLT aşamasında literatür tarama sürecini standardize etmek için Kitchenham ve Charters (2007) tarafından önerilen SLT yaklaşımı kullanılmıştır. Kitchenham ve Charters tarafından geliştirilen SLT modeli, araştırmacılara karmaşık bir konuda eksiksiz bir literatür tarama ve sentezleme süreci sunmaktadır. Bu model, araştırmacılara belirli bir araştırma sorusuna yanıt bulmak için adım adım bir yaklaşım sağlar. İlk adım, araştırmanın kapsamını belirlemeyi içerir; burada araştırmacılar, hangi konuların dâhil edileceğini ve dışlanacağını tanımlarlar. Ardından literatür tarama stratejileri geliştirilir ve belirli veri tabanları ve kaynaklar üzerinde sistematik bir arama yapılır. Bu süreçte, belirlenen kriterlere uygun olan ve araştırma sorusunu yanıtlamaya en uygun olan çalışmalar belirlenir. Seçilen çalışmaların kalitesi ve uygunluğu titizlikle değerlendirilir. Son adımda ise seçilen çalışmalardan elde edilen veriler sentezlenir ve analiz edilir. Bu analiz, mevcut bilgiyi özetlemeyi, eğilimleri belirlemeyi ve gelecekteki araştırmalar için önerilerde bulunmayı içerir. SLT modeli, araştırmacılara güvenilir ve kapsamlı bir şekilde bilgiye erişme ve araştırma sorularını yanıtlama imkanı sunar. Bu nedenle bilimsel araştırmalarda yaygın olarak kullanılan bir yöntemdir.

Literatür tarama süreci, İTÜ Elektronik Kütüphane veritabanlarını içermektedir. İlk olarak ilgilenilen alanla ilgili araştırma soruları formüle edilmiştir. Arama kriterleri; kritik başarı faktörleri, proje başarı ölçütleri, proje başarısızlığına neden olan faktörler ve proje risklerini kapsayacak şekilde oluşturulmuştur. Ardından proje başarı ölçütleri olan kritik başarı faktörlerini belirlemek için veri kaynakları, arama kriterleri ve makale seçim süreci uygulanmıştır.

4.1.Araştırma Soruları

Kitchenham ve Charters (2007), SLT modelini geliştirirken araştırmacıların kullanabileceği yönlendirici sorular önermiştir. Bu sorular, araştırmacıların çalışmalarını şekillendirmelerine ve literatür tarama sürecini daha etkili bir şekilde yönetmelerine yardımcı olur. Örneğin araştırma soruları arasında belirli bir konuda

mevcut olan literatürün ne kadarının bulunduğunu belirlemeye yönelik sorular bulunmaktadır. Bunun yanı sıra hangi veri tabanlarının veya kaynakların kullanılacağı, hangi arama stratejilerinin benimseneceği gibi pratik konuları da ele alacak sorular yer almaktadır. Ayrıca araştırmacıların belirli bir araştırma sorusuna yanıt bulabilmek için gereken kapsamı belirlemelerine yardımcı olan sorular da bulunmaktadır. Bu şekilde Kitchenham ve Charters'ın önerdiği araştırma soruları, araştırmacıların SLT modelini kullanarak daha odaklanmış, sistemli bir literatür taraması yapmalarına olanak tanır. Kitchenham ve Charters (2007)'a göre "SLT, belirli araştırma soru(ları) veya ilgi alanı ile ilgili mevcut tüm araştırmaları tanımlamak, değerlendirmek ve yorumlamak için sistematik bir yöntemdir". Bu çalışmanın araştırma soruları, kritik başarı faktörlerinin, proje risk faktörlerinin, proje başarısızlık faktörlerinin ve proje başarı ölçütlerinin belirlenmesine ve kategorize edilmesine odaklanmıştır:

Araştırma Sorusu 1. Çevik yazılım projelerinin kritik başarı faktörleri nelerdir?

Araştırma Sorusu 2. Çevik yazılım projelerinin başarı ölçütleri nelerdir?

Araştırma Sorusu 3. Kritik başarı faktörleri hangi kavramlara aittir?

Araştırma Sorusu 4. Proje başarı ölçütleri hangi kavramlara aittir?

4.2. Veri Kaynakları

Literatür taraması, Scencedirect, IEEE, SPRINGER, WILEY ve EMERALD veritabanlarından 2000-2021 yılları arasında son 21 yılı içerecek şekilde yapılmıştır. Literatür taraması için bitiş tarihi 26 Eylül 2021 olarak belirlenmiştir. SLT aşamasından sonra anketin veri toplama aşaması yaklaşık bir takvim yılı sürmüştür. Bu arada sürekli literatür taraması devam ettirilmiş, 2022 ve 2023 yıllarına ait yayınlar titizlikle okunmuş ve analiz edilmiştir. Böylelikle güncel literatür çalışmaya dahil edilmiştir. Makale tipi için literatür taraması, araştırma ve konferans makaleleri dikkate alınmıştır.

Scencedirect, çeşitli disiplinlerde bilimsel makalelere erişim sağlayan bir akademik yayın platformudur. Scencedirect bünyesinde makaleler, kitaplar ve konferans bildirileri bulundurmaktadır. IEEE (Institute of Electrical and Electronics Engineers), elektrik, elektronik ve bilgisayar mühendisliği gibi teknik alanlarda faaliyet gösteren yayın platformudur. IEEE, makaleler, teknik yayınlar, konferans bildirileri gibi yayınları bünyesinde barındırır. Springer, çeşitli disiplinlerde yayın yapan bir

platformdur. Springer bünyesinde makaleler, kitaplar ve konferans bildirileri bulundurmaktadır. Wiley, çeşitli disiplinlerde yayın yapan bir başka önemli yayınevidir. Wiley'in araştırma dergileri, kitaplar, veritabanları ve eğitim materyalleri gibi çeşitli yayınları bulunmaktadır. Son olarak Emerald Publishing, sosyal bilimler, işletme yönetimi ve mühendislik gibi alanlarda yayın yapan bir yayınevidir. Emerald da diğer bahsedilen yayınevleri gibi bünyesinde dergiler, kitaplar, konferans bildileri gibi çalışmaları bulundurmaktadır.

4.3.Arama Kriterleri

Kitchenham ve Charters (2007) çalışmasına göre arama kriterlerinin belirlenmesi, araştırmacıların odaklanmak istedikleri konuyu ve literatürü tanımlamalarına dayanır. Bu süreçte araştırmacılar genellikle araştırma sorularını ve hedeflerini dikkate alarak belirli anahtar kelimeler ve terimler oluştururlar. Ayrıca araştırmacılar belirli veri tabanlarını veya kaynakları seçerken, araştırmanın kapsamını belirlemek ve kaynaklara erişim sağlamak için dikkatli bir değerlendirme yaparlar. Bu süreçte araştırmacılar genellikle belirli dergi ve konferanslara da odaklanarak ilgili literatürü taramayı planlarlar. Kitchenham ve Charters'in önerdiği SLT modeli, araştırmacıların bu adımları sistemli bir şekilde uygulamalarını ve arama kriterlerini belirlemelerini sağlar, böylece araştırma süreci daha etkili ve yönlendirici bir şekilde ilerler. SLT için bir arama kriterine sahip olmak, sonraki adımlar için çok önemlidir. Arama dizisi formatı, araştırma sorularının ana terimlerine göre tasarlanmıştır. Bu çalışmada kullanılan arama dizesi aşağıdaki gibidir:

((agile OR scrum) AND software) AND (“project success” OR “critical success factor” OR “project failure” OR “project risk”).

4.4.Makale Seçim Süreci

Makale seçim sürecinde yukarıda belirtilen arama dizesi kullanılmıştır. Şekil 4.1, makale seçim sürecinin adımlarını göstermektedir. İlk adımda, seçilen veri tabanı kaynaklarından 3702 makale çıkarılmıştır. İkinci adımda başlıklar, özetler, girişler ve sonuçlar okunarak dahil etme ve kapsam dışı bırakma kriterleri uygulanmıştır.

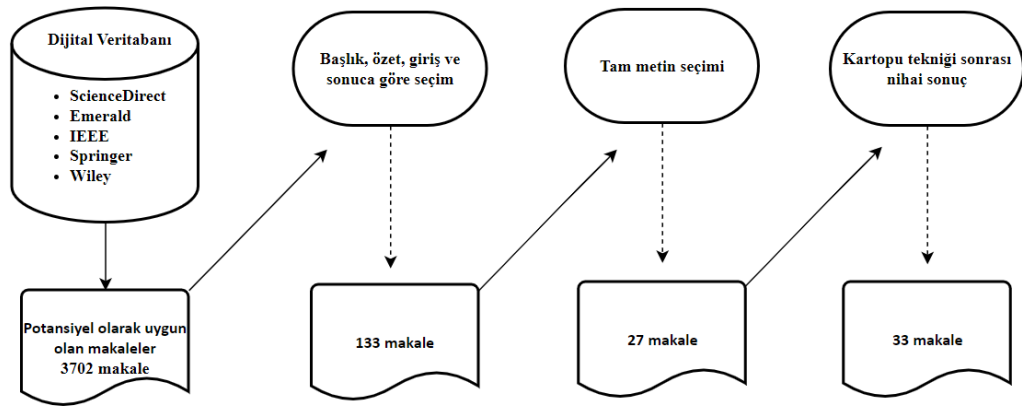
Makalelerin çalışmaya dahil edilme kriterleri aşağıdaki gibidir:

- İngilizce yazılmış makaleler
- 2000-2021 yılları arasında yayınlanan makaleler (26 Eylül 2021 tarihinden sonra yayınlanan makaleler dahil edilmemiştir)
- Çevrimiçi olarak mevcut makaleler
- Çevik yazılım projelerinde veya genel bağlamda kritik başarı faktörlerinden bahseden makaleler
- Çevik yazılım projelerinde veya genel bağlamda başarısızlık faktörlerinden bahseden makaleler
- Çevik yazılım projelerinde veya genel bağlamda risk faktörlerinden bahseden makaleler
- Akademi ve endüstri ile ilgili makaleler
- Yazılım geliştirmede başarı faktörleri ve başarı ölçütleri üzerine ampirik, teorik veya uzman görüşü içeren makaleler.

Kapsam dışı bırakma kriterleri aşağıdaki gibidir:

- Yazılım geliştirme ile ilgili olmayan makaleler (örneğin, üretim endüstrisinde Kanban panolarının uygulanması)
- Kısa makaleler (altı sayfadan az).

Bu kriterler uygulanarak makale sayısı 133'e indirilmiştir. Bu makaleler titizlikle okunduktan sonra 27 tanesi kısa listeye alınmıştır. İnternet aramaları ve kısa listeye alınan makalelerin referansları temel alınarak altı makale daha çalışmaya eklenmiştir. Son olarak, 33 makale belirlenmiştir. SLT sonucu seçilen makaleler Ek A'da gösterilmiştir. Çizelge 4.1, seçilen makalelerin dağılımını göstermektedir.



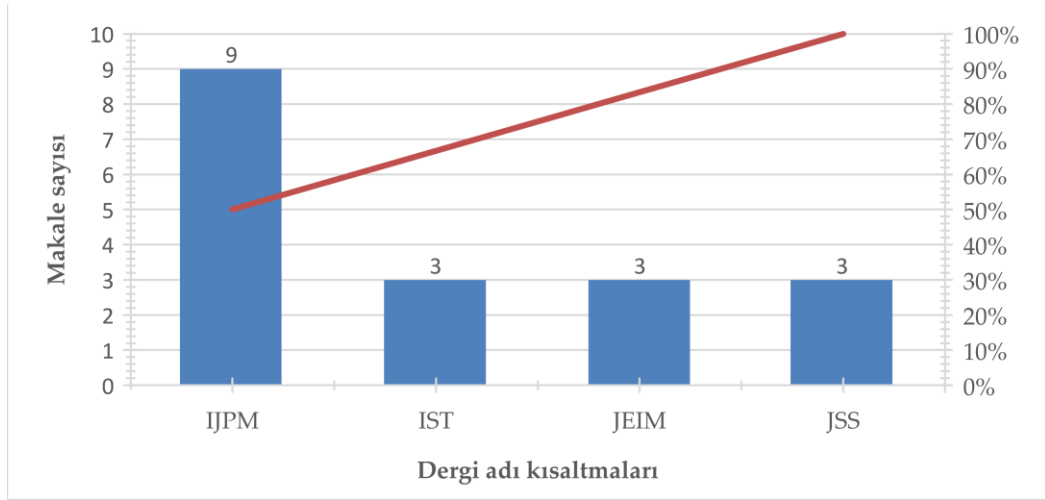
Şekil 4.1 : Makale seçme sürecine ait akış.

Çizelge 4.1 : Seçilen makalelerin dağılımı.

Kaynak	Potansiyel olarak uygun olan makaleler	İlk elemeyen sonraki makaleler	Seçilen makaleler	Oran (%)
Science Direct	1050	52	11	33.33
Emerald	1034	30	4	12.12
IEEE	69	14	5	15.15
Springer	1261	19	5	15.15
Wiley	288	20	2	6.06
Kartopu tekniği			6	18.18
Toplam	3702	133	33	100

4.5. Seçilen Makalelere Ait Yayınların İncelenmesi

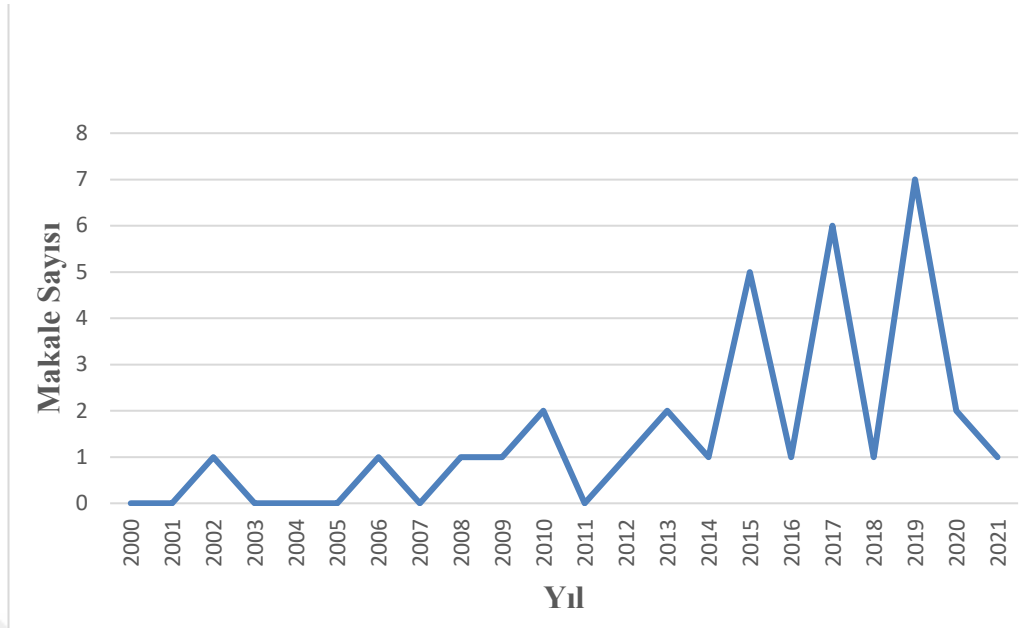
Şekil 4.2, seçilen makalelerden üç veya daha fazlasını yayınlayan dergilerin dağılımını göstermektedir. Görüldüğü gibi International Journal of Project Management (IJPM) en fazla sayıya sahiptir. IJPM dergisi, proje yönetimi alanında önde gelen ve saygın bir yayın organı olarak kabul edilmektedir. Dolayısıyla bu dergide yayınlanan makaleler, akademik topluluk tarafından geniş bir ilgi görmekte ve güvenilir bulunmaktadır. IJPM dergisinin proje yönetimi alanındaki popülaritesinden dolayı araştırmacıların IJPM dergisini tercih etmiş olduğu düşünülmektedir.



Şekil 4.2 : SLT aracılığıyla seçilen makalelerden üç veya daha fazlasını yayımlayan dergilerdeki makalelerin dağılımı.

4.5.1.Yıllara göre seçilmiş yayınların dağılımı

Seçilen 33 makalenin yıllara göre dağılımı Şekil 4.3’ de gösterilmiştir. Çevik Manifesto 2001 yılında yayınlanmış ve bu yıldan sonra çevik yazılım geliştirme ile ilgili akademik makaleler yayınlanmaya başlamıştır. Şekil 4.3’ e göre çevik yazılım geliştirme ile ilgili 2012’ den itibaren makale sayısında belirgin bir artış görülmüştür. Özellikle 2015 ve sonrasında çevik yönetimin kritik başarı faktörlerine odaklanan makalelerin sayısında önemli bir artış mevcuttur, bu da konunun popülerliğinin arttığını ve araştırmacıların ilgisini çektiğini göstermektedir. 2019’da zirve yapan yayın sayısı, çevik yönetimin iş dünyasında giderek daha fazla benimsendiğini ve organizasyonların bu yöntemi başarıyla uygulamak için ihtiyaç duydukları kritik faktörleri anlamak için artan bir talep olduğunu göstermektedir. Bununla birlikte 2020 ve 2021’de yayımlanan makale sayısında bir azalma gözlenmiştir. Fakat bu durum konunun öneminin azaldığı anlamına gelmez; bu dönemde pandemi gibi etmenlerin araştırma ve yayın süreçlerini etkilemiş olabileceği dikkate alınmalıdır. Sonuç olarak çevik yönetimin kritik başarı faktörlerini inceleyen araştırmaların artan bir trend göstermesi, bu alanda daha fazla çalışmanın gerekliliğini vurgulamaktadır ve gelecekteki araştırmaların bu konuya daha fazla odaklanması beklenmelidir.



Şekil 4.3 : Seçilen makalelerin yıllara göre dağılımı.

4.5.2. Seçilen makalelerin kalitesi

Seçilen makaleler dergi/konferans kalitesi ve atıf sayısına göre incelenmiştir. Jorge Hirsch tarafından 2005 yılında ortaya atılan h-indeksi, dergi ve konferansların kalitesini göstermektedir (Hirsch, 2005). H-indeksi SCImago Journal & Country Rank'e göre hesaplanmıştır (SJR, 2022). Çizelge 4.2'de, SLT sonucu seçilmiş olan 33 çalışmanın basıldığı yayın evi, her bir yayın evindeki makale sayısı, h-indeksi ve yayın türü gösterilmektedir. Yayınlar arasında "Computers in Human Behavior" ve "IEEE Access" dergisi en yüksek h-indekslerine sahiptir. "Computers in Human Behavior", bilgisayarlar ve insan davranışları arasındaki etkileşime odaklanarak, teknoloji ve insan faktörü arasındaki ilişkiyi anlamak için kullanılan, akademisyenler ve araştırmacılar arasında popüler olan akademik bir dergidir. "IEEE Access" dergisinin yüksek h-indeksine sahip olmasının nedenleri, derginin geniş kapsamlı konu çeşitliliğine sahip olması, IEEE tarafından desteklenmesi, açık erişim modeli ve mühendislik ile teknoloji alanlarına odaklanması olduğu düşünülmektedir. Ayrıca derginin açık erişim olması bilimsel bilgiye erişimi artırırken makalelerin geniş bir kitleye ulaşmasına olanak tanımaktadır.

Çizelge 4.3'de, SLT sonucu seçilmiş olan 33 çalışma arasından, Google Scholar'a göre en çok atıf alan ilk 10 makale gösterilmektedir (Google Scholar, 2022). Görüldüğü üzere, "Journal of Systems and Software" de yayınlanan Chow ve Cao (2008) "A Survey Study of Critical Success Factors in Agile Software Projects" makalesi en fazla

atıf alan makale olmuştur. Chow ve Cao (2008) çalışmasının yüksek atıf almasının sebebi, çevik yazılım projelerinde kritik başarı faktörlerini kapsamlı bir şekilde incelemesi ve bu alanda önemli bir boşluğu dolduran ilk akademik çalışmalardan biri olmasıdır. “Journal of Systems and Software” dergisinin yüksek h-indeksine sahip olması ve geniş bir okuyucu kitlesine sahip olmasının da yüksek atıf almasına yardımcı olduğu düşünülmektedir. Sadece Kanjanda ve Tuan (2020)’a ait olan “A Systemic Exploration of the Risk Factors in Zimbabwean Information Technology Projects” makalesi ve Perera ve Perera (2019)’a ait olan “The Impact of Client Involvement towards Agile Project Success in Sri Lankan Software Industry” çalışması beşten az atıf almıştır. Bunun nedeni, yakın zamanda yayımlanmış makaleler olduğu düşünülmektedir.

Çizelge 4.2 : H-indeksine dayalı yayınların kalitesi (Scimago Journal & Country Rank (SJR), 2022).

Yayın	Makale Sayısı	h-indeks	Türü
Computers in Human Behavior	1	251	Makale
IEEE Access	2	242	Makale
International Journal of Project Management	9	153	Makale
Journal of Systems and Software	3	113	Makale
Annals of Operations Research	1	111	Makale
Information and Software Technology	3	107	Makale
Journal of Enterprise Information Management	3	67	Makale
Software Quality Journal	2	45	Makale
Proceedings – Asia-Pacific Software Engineering Conference, APSEC	1	36	Konferans
Systemic Practice and Action Research	1	35	Makale
Journal of Software: Evolution and Process	2	32	Makale
International Journal of Systems Assurance Engineering and Management	1	28	Makale
Journal of Software Engineering and Applications	1	25	Makale
Journal of Science and Technology Policy Management	1	20	Makale
MERCon 2019 – Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference	1	6	Konferans
Proceedings – 6th Brazilian Workshop on Agile Methods, WBMA 2015	1	3	Konferans

Çizelge 4.3: En çok atıf alan 10 seçilmiş makale (Google Scholar, 2024).

Makale Adı	Yayın	Referans Sayısı
A Survey Study of Critical Success Factors in Agile Software Projects (Chow ve Cao, 2008)	Journal of Systems and Software	1662
Does Agile Work? - A Quantitative Analysis of Agile Project Success (Serrador ve Pinto, 2015)	International Journal of Project Management	1234
Framing of Project Critical Success Factors by a Systems Model (Fortune ve White, 2006)	International Journal of Project Management	1199
A Systematic Literature Review on Agile Requirements Engineering Practices and Challenges (Inayat ve diğ, 2015)	Computers in Human Behavior	714
Critical Failure Factors in Information System Projects (Yeo, 2002)	International Journal of Project Management	684
Relationships between a Project Management Methodology and Project Success in Different Project Governance Contexts (Joslin ve Müller, 2015)	International Journal of Project Management	648
Identifying Some Important Success Factors in Adopting Agile Software Development Practices (Misra ve diğ, 2009)	Journal of Systems and Software	585
The Relationship between Project Governance and Project Success (Joslin ve Müller, 2016)	International Journal of Project Management	402
Factors Associated with the Software Development Agility of Successful Projects (Sheffield ve Lemétayer, 2013)	International Journal of Project Management	341
A Project Contingency Framework Based on Uncertainty and Its Consequences (Howell ve diğ, 2010)	International Journal of Project Management	308



5. ÇEVİK YAZILIM BAŞARI MODELİ

Çevik Yazılım Başarı Modeli, çevik yazılım geliştirme metodolojisinin temel prensiplerine dayanarak proje yönetimi alanında son yıllarda yoğun ilgi gören bir araştırma konusudur. Geleneksel proje yönetimi yaklaşımları, karmaşık ve hızlı değişen iş ortamlarında yetersiz kalmaktadır. Bu yüzden bu çalışma özellikle çevik metodolojilerin etkinliği üzerine odaklanarak, projelerin başarısını etkileyen faktörleri tanımlamayı amaçlamaktadır. Çevik Yazılım Başarı Modeli, çevik kritik başarı ölçütleri ve proje başarı ölçütleri ile ilgili çalışmalara dayanarak oluşturulmuştur. Ayrıca çevik metodolojilerin kullanımının daha iyi anlaşılması için Çevik Manifesto, Çevik İlkeler ve Scrum Kılavuzu ayrıntılı olarak incelenmiştir. Çevik Manifesto, değişime yanıt verme, müşteri işbirliği, çalışan yazılım, bireylere ve etkileşimlere değer verme gibi çevik değerleri vurgulamak için kullanılmıştır. Kavramsal modele ek kavram ve öğeler eklemek için Çevik İlkeler, olaylar, eserler ve değerler incelenmiştir. Böylece çevik KBF'ler ve Çevik Yazılım Başarı Modeli'nin proje başarı ölçütleri belirlenmiştir.

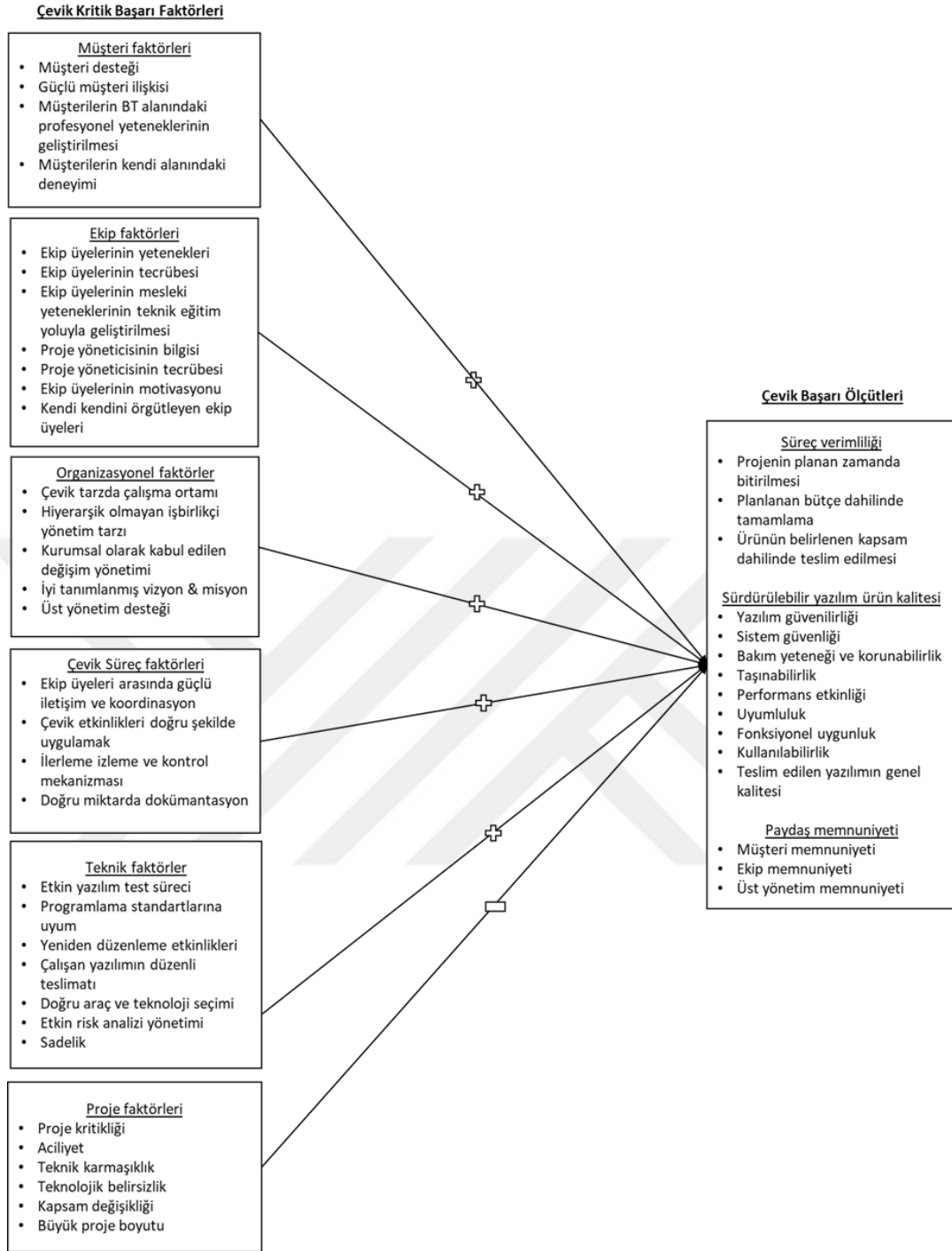
İkinci adımda, Çevik Yazılım Başarı Modeli hakkında tartışmak üzere çevik uygulayıcılarla bire bir toplantılar gerçekleştirilmiştir. Katılımcılar arasında scrum master, geliştirme ekibi üyeleri ve ürün sahibi olarak çalışmış çevik uygulayıcılar yer almıştır. Katılımcıların seçiminde amaçlı örnekleme kullanılmıştır. Amaçlı örneklemede amaca yönelik örneklem seçilmektedir. Bu bağlamda örneklemin çevik projelerde çalışmış ya da çalışmakta olan çevik uygulayıcılardan oluşması kriterine uyulmuştur. Tam ve diğ. (2020) çalışmasında önerildiği gibi toplantılar için altı çevik uygulayıcı seçilmiştir. Çevik uygulayıcıların demografik profili Çizelge 5.1'de gösterilmektedir. Bire bir toplantıların başında katılımcıların kullandıkları çevik uygulamalar hakkında genel bilgi alınmıştır. Tüm çevik uygulayıcılar etkin bir yazılım test sürecine sahip olduklarını belirtmiştir. Yeniden düzenleme faaliyetleri yaptıklarını, ürün/sprint iş listesini düzenli aralıklarla güncellediklerini, sürüm ve iterasyon grafiği grafikleri ile çevik dostu bir ilerleme izleme/kontrol mekanizmasına sahip olduklarını belirtmişlerdir. Daha sonra çevik KBF'ler ve önceki literatür

aracılığıyla belirlenen başarı ölçütleri ayrıntılı olarak tartışılmıştır. Çevik uygulayıcıların literatür taraması sonuçları hakkındaki yorumları, çevik KBF'lerin ve başarı ölçütlerinin açıklamasına ilerleyen bölümlerde dahil edilmiştir. Uygulayıcılar tarafından sağlanan öneriler yardımıyla çevik KBF'lerin ve başarı ölçütlerinin daha anlaşılabilir olması için küçük ifade değişiklikleri yapılmıştır.

Son adımda grup görüşmesinde altı çevik uygulayıcıdan oluşan grup yardımıyla çevik KBF'ler ve Çevik Yazılım Başarı Modelindeki proje başarı ölçütleri son haline getirilmiştir. Grup toplantısında çevik uygulayıcıların literatür taraması sonuçları hakkındaki yorumları alınmıştır. Çevik uygulayıcıların her bir çevik KBF ve başarı ölçütleri hakkında yaptıkları yorumlar ilerleyen bölümlerde bahsedilmiştir. Son olarak, Şekil 5.1' de gösterildiği gibi Çevik Yazılım Başarı Modeli oluşturulmuştur.

Çizelge 5.1: Çevik uygulayıcıların demografik profili.

Kaynak	Endüstri	İş Tecrübesi (Yıl)	Çevik Metodoloji Türü
Scrum Master	Bankacılık Sektörü	10 -15	Scrum, Kanban
Scrum Master	Havacılık Sektörü	10 -15	Scrum
Yazılım Geliştirici	Bankacılık Sektörü	15 yıl ve üstü	Scrum
Yazılım Geliştirici	Telekomünikasyon	10 -15	Scrum
Ürün Sahibi	Danışmanlık Sektörü	5 -10	Scrum, Kanban XP, Lean
İş Analisti	Borsa Sektörü	10 -15	Scrum



Şekil 5.1 : Çevik yazılım modeli.

5.1.Çevik Başarı Ölçütleri

Günümüzde çevik yazılım geliştirme projeleri giderek daha yaygın bir şekilde benimsenmektedir ve projelerin başarısını ölçmek önemli bir zorluk oluşturmaktadır. Proje başarısını etkileyen faktörlerin belirlenmesi ve ölçülmesi çevik yazılım geliştirme süreçlerinin etkin bir şekilde yönetilmesi için çok önemlidir. Proje Yönetim Enstitüsü (PMI), proje başarısını projenin kararlaştırılan zaman, bütçe ve kapsam dahilinde tamamlanması olarak tanımlamaktadır (PMI, 2013). Bu üç öge “demir üçgen” olarak adlandırılır. Chow ve Cao (2008), PMI tarafından tanımlanan ögelere kaliteyi, yani iyi proje çıktıları sunmayı da eklemiştir. Ahimbisibwe ve diğ. (2015) proje başarısının tanımını geliştirmiş ve teslim edilen yazılımın güvenilirliği, kullanım kolaylığı, genel kalitesi, esnekliği ve işlevselliği, ekip memnuniyeti, üst yönetim memnuniyeti ve kullanıcı memnuniyetini dahil etmiştir. Bu çalışmada çevik proje başarı ölçütleri literatür taraması, Çevik Manifesto, Çevik İlkeler, Scrum Kılavuzu, ISO/IEC 25010: 2011 kalite modeli (ISO, 2011) ve çevik uygulayıcılardan bire bir ve grup toplantısı yoluyla elde edilen yorumlara göre belirlenmiştir. Bu çalışmada çevik başarı ölçütleri üç kavramda kategorize edilmiştir: süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti. Çevik proje başarı ölçütleri ve bu ölçütlerle ilgili referanslar Çizelge 5.2’de sunulmuştur.

5.1.1. Süreç verimliliği

Çevik proje yönetiminde zaman, bütçe ve kapsamın bir araya gelmesi başarının değerlendirilmesi için esastır. Yinelemeli döngülerin zamanında gerçekleştirilmesi, kaynakların bütçe dahilinde titizlikle yönetilmesi ve üzerinde mutabık kalınan kapsam dahilinde paydaşlara karşı duyarlı olunması çevik proje başarısının temel taşlarını oluşturmaktadır. Bu çalışmada, süreç verimliliği üç öge ile açıklanmaktadır: projenin planlanan zamanda bitirilmesi, planlanan bütçe dahilinde tamamlama ve ürünün kararlaştırılan kapsam dahilinde teslim edilmesi.

Çizelge 5.2: Çevik başarı ölçütleri.

Kavram	Çevik başarı ölçütleri
Süreç Verimliliği	Projenin planlanan zamanda bitirilmesi (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Serrador ve Pinto, 2015; Joslin ve Müller, 2015, 2016; Ahimbisibwe ve diğ, 2015, 2017; Engelbrecht ve diğ, 2017; Garousi ve diğ, 2019; Ciric Lalic ve diğ, 2022)
	Planlanan bütçe dahilinde tamamlama (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Serrador ve Pinto, 2015; Joslin ve Müller, 2015, 2016; Ahimbisibwe ve diğ, 2015, 2017; Engelbrecht ve diğ, 2017; Garousi ve diğ, 2019; Ciric Lalic ve diğ, 2022)
	Ürünün belirlenen kapsam dahilinde teslim edilmesi (Stankovic ve diğ, 2013; Serrador ve Pinto, 2015; Shrivastava ve Rathod, 2015; Joslin ve Müller, 2015, 2016; Ahimbisibwe ve diğ, 2015, 2017; Engelbrecht ve diğ, 2017)
Sürdürülebilir Yazılım Ürün Kalitesi	Yazılım güvenilirliği (Ahimbisibwe ve diğ, 2015, 2017; Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Sistem güvenliği (Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Bakım yeteneği ve korunabilirlik (Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Taşımaabilirlik (Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Performans etkinliği (Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Uyumluluk (Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
	Fonksiyonel uygunluk (Ahimbisibwe ve diğ, 2015, 2017; Inayat ve diğ, 2015; Perkusich ve diğ, 2017; Garousi ve diğ, 2019)
	Kullanılabilirlik (Ahimbisibwe ve diğ, 2015, 2017; Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019)
Paydaş Memnuniyeti	Teslim edilen yazılımın genel kalitesi (Chow ve Cao, 2008; Sheffield ve Lemétayer, 2013; Stankovic ve diğ, 2013; Joslin ve Müller, 2015, 2016; Ahimbisibwe ve diğ, 2015, 2017; Engelbrecht ve diğ, 2017; Perkusich ve diğ, 2017; Garousi ve diğ, 2019)
	Müşteri memnuniyeti (Sheffield ve Lemétayer, 2013; Serrador ve Pinto, 2015; Joslin ve Müller, 2015, 2016; Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019; Ciric Lalic ve diğ, 2022)
	Ekip memnuniyeti (Sheffield ve Lemétayer, 2013; Serrador ve Pinto, 2015; Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019)
	Üst yönetim memnuniyeti (Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019)

5.1.1.1.Projenin planan zamanda bitirilmesi

Zamanında bitirme, proje takvimine bağlı kalınarak projenin zamanında tamamlanmasını ifade eder. Literatür taramasına göre belirlenen kapsam, zaman ve bütçe dahilinde yazılım teslimi en çok atıfta bulunulan proje başarı ölçütleridir (Chow ve Cao, 2008; PMI, 2013; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2015; Joslin ve Müller, 2015; Garousi ve diğ, 2019). Belirlenen süre içinde projenin tamamlanması, proje başarısının önemli bir ölçütüdür. Projenin belirlenen süre içinde tamamlanması, proje yöneticileri ve paydaşlar için güvenilirlik ve işbirliği açısından önemlidir. Bu ölçüt, projenin zaman yönetimi becerilerini, kaynakların etkin kullanımını ve ekip performansını değerlendirmede kullanılır. Ayrıca belirlenen süre içinde tamamlama, proje sürecinin verimliliğini ve müşteri memnuniyetini artırmaya yönelik stratejik bir hedef olabilir. Bu nedenle yazılım geliştirme projelerinde belirlenen süre içinde tamamlamanın önemini anlamak ve uygun stratejileri geliştirmek, araştırmacılar ve çevik uygulayıcılar için önemli bir konudur.

5.1.1.2.Planlanan bütçe dahilinde tamamlama

Her proje bir plan ve bütçe ile başlar. Ancak birçok durumda, projenin planlanan bütçe dahilinde teslim edilmesi önemli bir zorluktur. Planlanan bütçe dahilinde projenin tamamlanması, proje planlaması sürecinde belirlenen bütçe sınırları içinde kalınarak işlerin bitirilmesini ifade eder. CHAOS 2020'nin Beyond Infinity raporuna göre projelerin yalnızca %35'i zaman ve bütçe açısından tamamen başarılı olmuştur (Johnson, 2021). Bu bağlamda yazılım geliştirme projelerinde, planlanan bütçe dahilinde tamamlama, proje başarısını belirleyen önemli bir faktördür. Planlanan bütçe dahilinde tamamlama, proje yöneticileri ve paydaşlar için finansal kontrol ve kaynak yönetimi açısından önemlidir. Projenin maliyet etkinliği, harcanan kaynakların etkin kullanımını ve proje performansını değerlendirmede kullanılır. Ayrıca planlanan bütçe dahilinde tamamlama, proje sürecinin başarısını artırmaya yönelik stratejik bir hedef olabilir. Bu nedenle yazılım geliştirme projelerinde planlanan bütçe dahilinde tamamlamanın önemini anlamak ve uygun stratejileri geliştirmek önemlidir.

5.1.1.3.Ürünün belirlenen kapsam dahilinde teslim edilmesi

Yazılım geliştirme projelerinde ürünün kararlaştırılan kapsam dahilinde teslim edilmesi, projenin başarısını belirleyen önemli bir ölçüttür ve projenin önceden belirlenmiş kapsam dahilinde tamamlanması anlamına gelmektedir. Proje kapsamı, projenin hedeflerine ulaşması için önceden belirlenmiş görevlerden oluşur. Literatürde, kapsam dahilinde ürün teslimi başarı ölçütlerinden biri olarak kabul edilir (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2015; Serrador ve Pinto, 2015; Joslin ve Müller, 2016; Garousi ve diğ, 2019). Ürünün kapsam dahilinde teslim edilmesi, proje yöneticileri ve paydaşlar için ürünün işlevselliği ve kalitesi açısından kritik bir öneme sahiptir. Proje sürecinde belirlenen hedeflere ulaşılması ve müşteri memnuniyetinin sağlanmasını destekler. Ayrıca proje sürecinin etkin bir şekilde yönetilmesini ve proje başarısını artırmaya yönelik stratejik bir hedef olabilir. Bu nedenle yazılım geliştirme projelerinde ürünün kapsam dahilinde teslim edilmesinin önemi üzerine odaklanmak, araştırmacılar ve çevik uygulayıcılar tarafından dikkate alınması gereken bir proje ölçütüdür.

5.1.2. Sürdürülebilir yazılım ürün kalitesi

Yazılım mühendisliği literatüründe bir yazılım ürününün sürdürülebilirliği fonksiyonel olmayan bir gereksinim olarak ele alınmış ve yazılım ürünlerinin kalite nitelikleri sürdürülebilirlik ile ilişkilendirilmiştir (Khalifeh ve diğ, 2020). Condori-Fernandez ve Lago (2018) ve Condori-Fernandez ve diğ. (2020) çalışmalarında yazılım sürdürülebilirliğini, ISO/IEC 25010: 2011 kalite modelinde tanımlanan yazılım ürünü kalite öğeleri ile açıklamaktadır (ISO, 2011). Yazılım geliştirme süreçlerinde sürdürülebilir yazılım ürün kalitesi, önemli bir araştırma ve uygulama alanıdır. Bu kavram, yazılım ürünlerinin uzun süreli kullanımı ve bakımı için gereken kalite düzeyini ifade eder. Sürdürülebilir yazılım ürün kalitesi, yazılımın işlevselliği, performansı, güvenilirliği ve kullanılabilirliği gibi çeşitli öğeleri içerir. Bu nedenle yazılım geliştirme süreçlerinde sürdürülebilir kaliteyi sağlamak, hem yazılım endüstrisi hem de projedeki tüm paydaşlar açısından önemlidir. Sürdürülebilir yazılım ürün kalitesi, yazılım yaşam döngüsü boyunca sürekli olarak izlenmeli ve iyileştirilmelidir. Bu süreçte testler, denetimler ve geri bildirimler önemli bir rol oynar. Ayrıca sürdürülebilir yazılım ürün kalitesini sağlamak için iyi bir yazılım mimarisi ve kod kalitesi gereklidir. Bu çalışmada yazılım ürün kalitesinin sürdürülebilirliği dokuz

öge ile tanımlanmaktadır: yazılım güvenilirliği, sistem güvenliği, bakım yeteneği ve korunabilirlik, taşınabilirlik, performans etkinliği, uyumluluk, fonksiyonel uygunluk, kullanılabilirlik ve teslim edilen yazılımın genel kalitesi.

5.1.2.1.Yazılım güvenilirliği

ISO/IEC 25010: 2011 kalite modeli yazılım güvenilirliğini, “bir sistemin, ürünün veya bileşenin belirli bir süre boyunca belirli koşullar altında belirli işlevleri yerine getirme derecesi” olarak tanımlamaktadır (ISO, 2011). Bu kavram, yazılımın istenen işlevleri sağlama yeteneği ve hata toleransı ile ilgilidir. Yazılım güvenilirliği, bir yazılımın belirli bir süre boyunca istikrarlı ve beklenen şekilde çalışma yeteneğini ifade eder ve bir yazılım ürünü kusurlarının veya hatalarının sayısını tahmin etmeye yardımcı olur (Rico ve diğ, 2008). Güvenilirlik, öncelikle yazılım kullanım ömrünün iyileştirilmesi, destek ve geliştirme maliyetlerinin azaltılması gibi yazılım sürdürülebilirliğinin sosyal ve ekonomik boyutlarıyla ilgilidir (Khalifeh ve diğ, 2020). Garousi ve diğ. (2019), teslim edilen yazılımın/hizmetin güvenilirliğinin önemli bir proje başarı ölçütü olduğunu belirtmiştir. Günümüzde yazılım güvenilirliği, yazılım endüstrisinin ve kullanıcıların öncelikli endişelerinden biri haline gelmiştir. Yazılımın güvenilirliğini artırmak, hem yazılım geliştiriciler hem de son kullanıcılar için önemlidir. Ayrıca grup toplantısında tüm katılımcılar yazılım güvenilirliğinin proje başarı ölçütlerinden biri olduğu konusunda hemfikir olmuş ve önemini vurgulamışlardır.

5.1.2.2.Sistem güvenliği

Bilgi teknolojileri alanında güvenlik kavramı, hem akademik araştırmaların hem de endüstri uygulamalarının odak noktalarından birini oluşturmaktadır. Sistem güvenliği, ürünün kendisini kasıtlı veya kazara izinsiz girişlere karşı koruma yeteneğini, yani saldırıya karşı koyma yeteneğini ifade eder. Sistemin ister kötü niyetli ister kazara olsun, yetkisiz erişime karşı koyma yeteneği, yazılım ürününün başarısını tanımlar. Bilgi ve iletişim sistemlerinin, verilerin ve kullanıcıların korunması amacıyla alınan önlemleri ve bu önlemlerin etkinliğini ifade eder. Az sayıda makale, yazılım güvenliğinden bir proje başarı ölçütü olarak bahsetmiştir (Garousi ve diğ, 2019). Günümüzde internetin ve dijital teknolojilerin yaygınlaşmasıyla birlikte güvenlik endişeleri de artmıştır. Bilgi güvenliği; siber güvenlik, veri gizliliği ve yetkilendirme gibi alt alanları içeren geniş bir konu yelpazesini kapsamaktadır. Güvenlik, hem bireyler hem de kurumlar için çok önemlidir ve artan bir ivmeyle gelişen tehditlere

karşı güncel kalabilmek için sürekli olarak yenilenen bir alan olarak kabul edilmesi gerekmektedir. Bire bir toplantılarda 10 yıllık deneyime sahip bir yazılım geliştirici, “Altyapı veya hizmetlerde karşılaşılabilecek olası saldırı türlerinin farkında olmalıyız. Ayrıca sürdürülebilir bir yazılım ürünü için güvenlik kalite özelliğini yazılım geliştirme sürecinin her bir aşamasına dahil etmeliyiz. Böylelikle olası tehditlerin önüne geçmiş ve erken önlem almış oluruz” değerlendirmesinde bulunmuştur.

5.1.2.3.Bakım yeteneği ve korunabilirlik

Yazılım geliştirme süreçlerinde bakım yeteneği, yazılımın yaşam döngüsü boyunca yönetilebilirliği, değiştirilebilirliği ve korunabilirliği için önemlidir. Yazılım ürününün, geliştiricilerin ve kullanıcıların ihtiyaçlarını karşılayabilmek için uzun vadede verimli bir şekilde çalışabilir olmalıdır. Bu yüzden bakım yeteneği ve korunabilirlik “geliştiriciler tarafından bir ürünün veya sistemin ne derecede etkin ve verimli bir şekilde değiştirilebileceğinin derecesidir” şeklinde ifade edilmektedir (ISO, 2011). Üç tür bakım vardır: düzeltici, uyarlanabilir ve mükemmelleştirici. Düzeltici bakım, performans veya uygulamadaki sistem eksikliklerini gidermeyi amaçlar. Uyarlanabilir bakımla ilgili olarak sistemler, örneğin işletim sisteminin yükseltilmesi gibi değişikliklere uyarlanabilir olmalıdır. Mükemmelleştirici bakım, sistemin performans, işlem verimliliği veya bakım yapılabilirlik açısından mükemmelleştirilmesini ifade etmektedir (Chapin ve diğ, 2001). Bakım yeteneği, yazılımın başlangıçtaki tasarımından başlayarak geliştirme süreci boyunca kodlama, test etme aşamalarını içeren tüm süreçleri kapsar. Ayrıca yazılımın bakım yeteneği, gelecekteki değişikliklere uyum sağlama yeteneği, destek ve bakım maliyetlerinin düşük olması ile de ilgilidir. Sadece teknik açıdan değil aynı zamanda iş süreçleri, organizasyonel gereksinimler ve kullanıcı beklentileri gibi faktörler açısından da düşünülmelidir. Yazılım geliştirme ekibi, doğru metodolojileri ve araçları seçmeli ve yazılım yaşam döngüsü boyunca sürekli olarak iyileştirmeler yapmalıdır. Bakım yeteneği ve korunabilirlik, son kullanıcılar için güvenilirlik, performans ve kullanılabilirlik sağlayarak müşteri memnuniyetini artırır. Yazılım geliştiricileri için ise kod karmaşıklığını azaltarak geliştirme sürecini daha verimli hale getirir ve kaynakların daha etkin kullanılmasına yardımcı olur.

5.1.2.4.Taşınabilirlik

Taşınabilirlik, “bir sistem, ürün veya bileşenin bir donanımdan, yazılımdan veya diğer işletim veya kullanım ortamından başka birine ne derecede etkin ve verimli bir şekilde aktarılabilir olduğunun derecesidir” (ISO, 2011). Yazılım geliştirme süreçlerinde taşınabilirlik kavramı, yazılımın farklı platformlar ve ortamlar arasında sorunsuz bir şekilde çalışabilir olması için önemli bir faktördür. Taşınabilirlik, yazılımın bir sistemden diğerine kolayca aktarılabilmesi ve farklı donanım ve yazılım ortamlarında çalışabilmesi anlamına gelir. Bir uygulama kullanım ömrü boyunca birden fazla platforma taşınabiliyorsa kullanım ömrü uzar ve kullanıcı kitlesi genişler (Mooney, 2004; Khalifeh ve diğ, 2020). Günümüzde çeşitli araçlar, işletim sistemleri ve bulut ortamları gibi farklı platformlar arasında yazılımın taşınabilirliği önemlidir. Taşınabilir bir yazılım, kullanıcıların ihtiyaçlarına daha geniş bir yelpazede yanıt verebilir ve yazılım geliştiricilerine daha fazla esneklik ve ölçeklenebilirlik sağlar. Taşınabilirlik, yazılım ve donanım ömrünü uzatır, israfı azaltır, yazılım kullanımının uyarlanabilirliğini geliştirir ve müşterilerin sadakatini ve memnuniyetini artırır (Khalifeh ve diğ, 2020). Grup toplantısında, deneyimli yazılım geliştiricilerden biri, “Bazı programlama dilleri, C gibi, nispeten taşınabilirdir. C dilinin taşınabilirliği nedeniyle, bazı yazılım geliştiriciler kodlarını C dilinde yeniden yazmayı ve derlemeyi tercih ederler” değerlendirmesinde bulundu. Taşınabilirlik, yazılım yaşam döngüsünde süreç tasarım aşamasından itibaren dikkate alınmalı ve geliştirme süreci boyunca sürekli olarak göz önünde bulundurulmalıdır. Bu kapsamda standartlar, açık kaynaklı çözümler ve uyumlu arayüzler gibi çeşitli yöntemler ve teknolojiler kullanılmalıdır.

5.1.2.5.Performans etkinliği

Performans etkinliği, “bir bileşenin veya sistemin belirlenen işlevlerini yerine getirirken zamanı, kaynakları ve kapasiteyi kullanma derecesidir” (ISTQB, 2022). Üç alt özelliği vardır: zaman davranışı, kaynak kullanımı ve kapasite. Yazılım geliştirme süreçlerinde, performans etkinliği kavramı, yazılımın kaynakları verimli bir şekilde kullanarak hedeflenen performans düzeylerini sağlama yeteneğini ifade eder. Performans etkinliği, yazılımın işlemleri hızlı ve verimli bir şekilde gerçekleştirebilmesi, kaynakları (örneğin, bellek, işlemci, ağ bant genişliği) etkin bir şekilde kullanabilmesi ve talep edilen performans düzeylerini karşılayabilmesi anlamına gelir. Günümüzde kullanıcılar hızlı ve güvenilir bir deneyim beklediği için

performans etkinliđi, yazılımın rekabet avantajı elde etmesi için kritik bir faktördür. Ayrıca performans etkinliđi enerji tüketimini, e-atığı azaltır ve yazılım üretkenliğini artırır (Condori-Fernandez ve Lago, 2018; Khalifeh ve diğ, 2020). Ayrıca bulut bilişim gibi yeni teknolojilerin yaygınlaşmasıyla birlikte performans etkinliđi, ölçeklenebilirlik ve maliyet etkinliđi gibi bulut hizmetlerinin temel gereksinimlerinden biri haline gelmiştir. Yazılım geliştiricileri, performans etkinliğini artırmak için kodlama tekniklerini optimize etmeli, veri tabanı ve ađ işlemlerini iyileştirmeli ve ölçeklenebilir mimariler kullanmalıdır. Ayrıca performans testleri ve izleme araçları kullanarak yazılımın performansını sürekli olarak izlemek ve iyileştirmek önemlidir. Grup görüşmelerinde deneyimli bir yazılım geliştirici, “Sistem performansını belirlemek için yazılım ürününün kullanıcı girdisine ne kadar hızlı yanıt verdiğini gözlemleyebiliriz. Ayrıca sistem maksimum sınırına ulaştığında örneğin sistemdeki eş zamanlı kullanıcı sayısının maksimum sınırına ulaşması, sistemin nasıl tepki verdiğini ve sistemin fonksiyonlarını nasıl yerine getirdiğini gözlemleyebiliriz” değerlendirmesinde bulundu.

5.1.2.6.Uyumluluk

Uyumluluk, “bir ürün, sistem veya bileşenin aynı donanım veya yazılım ortamını paylaşırken diğeri ürün, sistem veya bileşenlerle bilgi alışverişinde bulunabilme ve/veya işlevlerini yerine getirebilme derecesidir” (ISO, 2011) (örneğin, yazılımın Unix, Windows ve macOS gibi farklı işletim sistemleriyle uyumluluđu). Uyumluluk, yazılım ve geliştirme maliyetlerini en aza indirirken kullanıcı iletişimini ve bilgi alışverişi esnekliğini de geliştirir (Condori-Fernandez ve Lago, 2018; Khalifeh ve diğ, 2020). Yazılım geliştirme süreçlerinde uyumluluk kavramı, yazılımın farklı platformlar, cihazlar ve ortamlarla sorunsuz bir şekilde çalışabilir olması için önemli bir gerekliliktir. Uyumluluk, yazılımın birçok farklı kullanıcı kitlesi ve çeşitli donanım/yazılım kombinasyonları tarafından kullanılabilmesini sağlar. Günümüzde çeşitli işletim sistemleri, tarayıcılar, cihazlar ve uygulama programlama arayüzleri (API’ler) gibi farklı teknolojilerin yaygınlaşmasıyla birlikte yazılımların uyumluluđu daha da önemli hale gelmiştir. Uyumluluk, yazılım geliştirme sürecinin her aşamasında dikkate alınmalı ve test edilmelidir. Yazılımın farklı platformlarda ve çevrelerde nasıl davrandığını anlamak için kapsamlı testler yapılmalı ve uyumluluk sorunları giderilmelidir. Ayrıca açık standartlar ve uyumlu arayüzlerin kullanılmasıyla yazılımın geniş bir kullanıcı kitlesiyle uyumlu olmasını sağlar. Uyumluluk, hem

yazılım geliştiricileri hem de son kullanıcılar için önemli avantajlar sunar. Yazılım geliştiricileri, daha geniş bir kitleye ulaşarak pazar paylarını artırabilir ve yazılımlarının popülerliğini artırabilir. Son kullanıcılar ise farklı cihazlarda ve platformlarda sorunsuz bir deneyim yaşayarak ürünleri daha etkili bir şekilde kullanabilirler. Grup toplantısına katılan deneyimli bir scrum master, “Sistemin uyumluluğu, ürün kalitesi hakkında geniş bilgi sağlar. Örneğin, bir mobil uygulama geliştirirken iOS, Android ve diğer mobil platformlarla uyumlu olup olmadığı ürününüzün kalitesi hakkında bilgi verir. Benzer şekilde web sitenizin Google Chrome, Firefox vb. gibi farklı tarayıcılarda sorunsuz çalışması, web sitenizin çapraz tarayıcı uyumluluğuna sahip olduğu anlamına gelir” değerlendirmesinde bulundu.

5.1.2.7.Fonksiyonel uygunluk

Yazılım ürünlerinin fonksiyonel uygunluğu, çevik yazılım proje yönetimi süreçlerinde vazgeçilmez bir unsurdur. Fonksiyonel uygunluk, “bir ürün veya sistemin belirtilen koşullar altında kullanıldığında belirtilen ve ima edilen ihtiyaçları karşılayan işlevleri sağlama derecesidir” (ISO, 2011). Fonksiyonel uygunluk, kullanıcıların ihtiyaçlarını etkili bir şekilde karşılamasına olanak sağlar. Ürünün performansını ve güvenilirliğini artırarak müşteri memnuniyetini sağlanmasına destek olur. Fonksiyonel uygunluk, uzun süreli kullanımı mümkün kılar ve bakım maliyetlerini azaltır (Khalifeh ve diğ., 2020). Ahimbisibwe ve diğ. (2015), (2017) ve Garousi ve diğ. (2019) çalışmalarında, fonksiyonel uygunluğu bir proje başarı ölçütü olarak tanımlanmıştır. Fonksiyonel uygunluk, yazılım geliştirme sürecinin her aşamasında dikkate alınmalıdır. Öncelikle proje başlangıcında fonksiyonel gereksinimler net bir şekilde belirlenmeli ve bu gereksinimler, müşteri ve paydaşlarla sıkı bir iletişim içinde sürekli olarak gözden geçirilmelidir. Ardından çevik prensiplere dayalı iteratif ve artımlı geliştirme süreçleri kullanılarak yazılımın fonksiyonel uygunluğu sürekli olarak test edilmeli ve değerlendirilmelidir. Bu süreç, hızlı geri bildirim döngüleri aracılığıyla yazılımın fonksiyonel uygunluğunu artırma ve iyileştirme fırsatları sağlayacaktır. Bire bir toplantılar sırasında uygulayıcılar, ürün teslimatını sağladıktan sonra müşteriden ürünün fonksiyonları hakkında olumlu geri dönüşler duymaktan memnun olduklarını vurgulamışlardır.

5.1.2.8.Kullanılabilirlik

Kullanılabilirlik, çevik yazılım projelerinde kritik bir öneme sahiptir. Kullanılabilirlik, “bir ürün veya sistemin belirli bir kullanım bağlamında belirli hedeflere ulaşmak için belirli kullanıcılar tarafından etkinlik, verimlilik ve memnuniyetle kullanılabilme derecesidir” (ISO, 2011). Kullanılabilirlik, kullanıcıların belirli bir bağlamda verimli ve etkili bir şekilde ürünün işlevlerini gerçekleştirebilme kabiliyetini ifade eder. Çevik yazılım projelerinde, kullanılabilirliğin sağlanması, hem müşteri memnuniyetini artırmak hem de ürünün rekabet gücünü güçlendirmek açısından çok önemlidir. Kullanıcı gereksinimleri ile uyumlu bir ürün tasarımı, sistem kullanımının kolaylaşmasını sağlayacak ve kullanıcı memnuniyeti sağlanmış olacaktır. Garousi ve diğ. (2019) çalışmasında kullanılabilirliği bir proje başarı ölçütü olarak değerlendirmiştir. Kullanılabilirliğin çevik yazılım proje yönetimi süreçlerinde etkili bir şekilde ele alınması için çeşitli stratejiler ve yöntemler benimsenmelidir. Öncelikle projenin başlangıcında kullanıcı ihtiyaçları ve beklentileri detaylı bir şekilde analiz edilmeli ve belirlenmelidir. Bu analiz, kullanıcı araştırmaları, anketler, odak grupları ve kullanıcı geri bildirimleri gibi çeşitli tekniklerle desteklenmelidir. Daha sonra geliştirme sürecinde kullanılabilirlik odaklı tasarım prensipleri ve kullanıcı arayüzü yönergeleri dikkate alınarak yazılımın kullanıcı dostu bir şekilde tasarlanması sağlanmalıdır. Kullanılabilirlik, sadece yazılımın kullanıcı arayüzüyle sınırlı değildir. Kullanılabilirlik, yazılımın performansı, erişilebilirliği, güvenliği ve diğer kalite özellikleriyle de ilişkilidir. Bu yüzden çevik yazılım projelerinde, kullanılabilirlik sağlanırken bu faktörler de göz önünde bulundurulmalıdır. Böylelikle kullanıcı odaklı bir yaklaşımla geliştirilen yazılımlar müşteri memnuniyetini artırır ve rekabet avantajı sağlar. Bir toplantı sırasında çevik uygulayıcılar, ürünün kullanılabilirliğini sağlamak için aşağıdaki soruların olumlu yanıtlanması gerektiğini vurgulamışlardır: Ürünün çalışmasını öğrenmek kolay mı? Ürünün kullanımı kolay mı? Farklı insanlar ürünü rahatlıkla kullanabilir mi?

5.1.2.9.Teslim edilen yazılımın genel kalitesi

Proje yönetiminde projenin kalitesi zaman, bütçe ve kapsam ile sınırlandırılır. Ancak sistem kalitesi, proje ve ürün yaşam döngüsünü etkileyen çok yönlü ve çok boyutlu bir yapıyı temsil etmektedir (Hughes ve diğ, 2020). Literatüre göre teslim edilen yazılımın genel kalitesi en çok atıfta bulunulan proje başarı ölçütüdür. Chow ve Cao (2008),

Sheffield ve Lemétayer (2013), Stankovic ve diğ. (2013), Joslin ve Müller (2015), Engelbrecht ve diğ. (2017), Papke-Shields ve Boyer-Wright (2017) ve Hughes ve diğ. (2020) çalışmalarında “teslim edilen yazılımın genel kalitesi” proje başarı ölçütlerinden biri olarak tanımlanmıştır. Çevik proje yönetimi süreçlerinde yazılımın genel kalitesinin sağlanması ve sürekli olarak iyileştirilmesi için etkili stratejiler ve yöntemler benimsenmelidir. Yazılımın genel kalitesinin sağlanması, çevik yazılım proje yönetimi süreçlerinin her aşamasında dikkate alınmalıdır. Öncelikle projenin başlangıcında paydaşlarla işbirliği içinde genel kalite hedefleri belirlenmelidir. Ardından yazılım geliştirme sürecinde kalite odaklı bir yaklaşım benimsenmeli ve yazılımın her aşamasına kalite kontrol süreçleri entegre edilmelidir. Bu süreçle hızlı geri bildirim döngüleriyle desteklenerek yazılımın genel kalitesinin sürekli olarak iyileştirilmesi sağlanacaktır. Yazılım kalitesinin sağlanması sadece geliştirme sürecinin bir parçası değildir. Yazılım yaşam döngüsü boyunca, kullanıcı geri bildirimleri ve performans izleme gibi süreçlerle yazılım kalitesi sürekli olarak takip edilmeli ve geliştirilmelidir. Sonuç olarak çevik yazılım proje yönetimi süreçlerinde yazılım kalitesinin sağlanması ve sürdürülebilir bir şekilde iyileştirilmesi, müşteri memnuniyetini artırmak ve proje başarısını desteklemek açısından oldukça önemlidir. Tüm paydaşlar tarafından titizlikle takip edilmesi gerekmektedir.

5.1.3. Paydaş memnuniyeti

Çevik yazılım proje yönetimi, günümüzde yazılım endüstrisinde giderek daha fazla kabul gören ve benimsenen bir yaklaşımdır. Proje paydaşlarının beklentilerini karşılama ve müşteri memnuniyetini artırma gibi önemli hedeflere odaklanarak projelerin başarılı bir şekilde yönetilmesini sağlar. Paydaş memnuniyeti, proje paydaşlarının (müşteriler, yazılım geliştirme ekibi üyeleri, üst yönetim) proje sonuçlarından memnun olma derecesini ifade eder. Çevik yazılım proje yönetimi süreçlerinde paydaş memnuniyetinin sağlanması için etkili stratejiler ve yöntemler benimsenmelidir. Paydaş memnuniyetinin sağlanması çevik yazılım proje yönetimi süreçlerinin her aşamasında dikkate alınmalıdır. Proje başlangıcında paydaşların beklenti ve gereksinimleri detaylı bir şekilde analiz edilmeli ve bu bilgiler doğrultusunda projenin planlanması yapılması ve yönetilmesi gerekmektedir. İletişim, çevik yazılım proje yönetimi sürecinde paydaş memnuniyetini artırmak için önemlidir. Sürekli iletişimin devam ettirilmesi ve düzenli toplantıların düzenlenmesi, paydaşların proje ilerlemesi ve karar alma süreçlerine katkı sağlamaktadır. İlgili çalışmada paydaş

memnuniyeti üç kavramda incelenmektedir: müşteri memnuniyeti, ekip memnuniyeti ve üst yönetim memnuniyeti.

5.1.3.1.Müşteri memnuniyeti

Çevik Manifesto’da, “Sözleşme pazarlıklarından ziyade müşteri ile işbirliğine değer vermeye kanaat getirdik.” denmektedir (Agile Manifesto, 2001a). Çevik yazılım geliştirmenin en önemli hedeflerinden biri sürekli yazılım teslimatı yoluyla müşteri memnuniyetini sağlamaktır. Eğer sistem kullanışlı ve güvenilir ise müşteri memnuniyeti uzun süreli olur (Condori-Fernandez ve Lago, 2018). Müşteri memnuniyeti, bir yazılım projesinin başarılı olup olmadığını belirleyen kritik bir ölçüttür, çünkü memnun müşteriler, uzun vadeli iş ilişkilerinin ve olumlu referansların anahtarıdır. Müşteri memnuniyeti sadece yazılımın teslimatı ile sınırlı değildir aynı zamanda müşteri iletişimi ve ilişkileri boyunca da dikkate alınmalıdır. Müşteri ile açık bir iletişim sürdürmek, beklentileri doğru bir şekilde yönetmek ve proje ilerlemesi hakkında şeffaf bir görünüm sağlamak önemlidir. Bire bir toplantılar ve Misra ve diğ. (2009), Joslin ve Müller (2015) ve Serrador ve Pinto (2015) çalışmalarına göre, müşteri memnuniyeti en önemli proje başarı ölçütlerinden biridir.

5.1.3.2.Ekip memnuniyeti

Ekip üyelerinin memnuniyeti, proje başarı ölçütlerinden biri olarak kabul edilmektedir (Sheffield ve Lemétayer, 2013; Serrador ve Pinto, 2015). Çevik Prensipte göre, “Projelerin temelinde motive olmuş bireyler yer almalıdır. Onlara ihtiyaçları olan ortam ve destek sağlanmalı, işi başaracakları konusunda güven duyulmalıdır” denmektedir (Agile Manifesto, 2001b). Çevik proje yönetimi süreçlerinde ekip memnuniyetini sağlamak, başarılı bir proje teslimi için önemlidir. Ekip memnuniyeti, bir yazılım projesinin başarılı olup olmadığını belirleyen kritik bir ölçüttür, çünkü yazılım ekibi üyelerinin memnuniyeti, aidiyet duygusunu, bir sonraki projeler için motivasyonu ve üretkenliği artırır. Bu yüzden proje başlangıcında ve proje süresince takım üyelerinin ihtiyaçlarına ve beklentilerine öncelik verilmelidir. Ayrıca takım üyeleriyle düzenli iletişim kurulmalı, geri bildirimler alınmalı ve gerektiğinde destek verilmelidir. Ekip üyelerinin memnuniyeti sadece gelecek projeler için iş performansını etkilemekle kalmaz, aynı zamanda takım atmosferini de olumlu yönde etkiler. Pozitif bir çalışma ortamı, takım üyelerinin kendilerini ifade etmelerini, fikirlerini paylaşmalarını ve yaratıcı çözümler bulmalarını teşvik eder. Bu da ekibin

mevcut ve gelecek projelerinin başarısını artırır. Bu yüzden ekip memnuniyeti proje başarı ölçütlerinin başında gelmektedir.

5.1.3.3.Üst yönetim memnuniyeti

Çevik uygulayıcılarla yapılan bire bir toplantılara ve literatüre göre üst yönetim memnuniyeti önemli çevik proje başarı ölçütlerinden biridir (Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019). Üst düzey yönetimin memnuniyeti, bir proje yönetim sürecindeki üst düzey yöneticilerin proje sonuçlarından ne kadar memnun olduklarını yansıtır ve proje başarısını belirleyen kritik bir ölçüttür. Üst yönetimin memnuniyetini sağlamak, gelecekteki projeler için onların desteğini kazanmak açısından çok önemlidir. Ayrıca üst düzey yöneticilerin projeye olan katkısı ve desteği, proje sürecinin ilerlemesi ve başarısı açısından kritiktir. Bu nedenle projenin başlangıcından itibaren üst düzey yöneticilerin beklenti ve gereksinimleri dikkate alınmalı ve sürekli olarak iletişim halinde olunmalıdır. Ayrıca proje ilerledikçe düzenli olarak raporlar sunularak üst düzey yöneticilere proje durumu hakkında şeffaf bir görünüm sağlanmalıdır. Proje sürecince üst düzey yöneticilerle etkili iletişim kurulması ve işbirliği içinde çalışılması, üst düzey yönetimin projeye olan güvenini artırır ve projenin başarısını destekler. Sonuç olarak çevik yazılım proje yönetimi süreçlerinde üst düzey yönetim memnuniyeti, proje başarısını artırır ve uzun vadeli iş ilişkilerinin güçlenmesini destekler.

5.2.Çevik Kritik Başarı Faktörleri

Çevik yazılım projelerinin başarısını etkileyen kritik faktörler, projelerin temelini oluşturan unsurlardır. Bu faktörler, geleneksel proje yönetimi metodolojilerinden farklı olarak çevik yaklaşımla yönetilen projelerin başarısını belirleyen önemli unsurlardır. Kritik faktörler, çevik yazılım projelerinin başarılı bir şekilde yönetilmesini ve sonuçlandırılmasını sağlamaktadır. Çevik metodolojiler, modern iş dünyasında giderek daha yaygın olarak tercih edilen bir yaklaşım haline gelmiştir. Bu yaklaşım, geleneksel proje yönetimi metodolojilerinden farklı olarak, esneklik, işbirliği ve sürekli gelişim prensiplerine dayanır. Çevik kritik başarı faktörleri, ilgili referanslar ve kavramlarla birlikte Çizelge 5.3’de sunulmuştur. Kavramlar: müşteri faktörleri, ekip faktörleri, organizasyonel faktörler, çevik süreç faktörleri, teknik faktörler ve proje faktörleri olarak tanımlanmıştır. Her bir kavrama ait öğeler detaylı bir şekilde ilerleyen bölümlerde anlatılmıştır.

Çizelge 5.3 : Çevik kritik başarı faktörleri.

Kavram	Kritik başarı faktörleri
Müşteri faktörleri	Proje boyunca güçlü müşteri desteği (Fortune ve White, 2006; Chow ve Cao, 2008; Misra ve diğ, 2009; Stankovic ve diğ, 2013; Silva ve Santos, 2015; Inayat ve diğ, 2015; Ahimbisibwe ve diğ, 2015, 2017; Rashid ve Khan, 2018; Garousi ve diğ, 2019; Perera ve Perera, 2019; Hughes ve diğ, 2020; Tam ve diğ, 2020)
	Güçlü müşteri ilişkisi (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Rashid ve Khan, 2018; Abrar ve diğ, 2019; Perera ve Perera, 2019)
	Müşterilerin BT alanındaki profesyonel yeteneklerinin geliştirilmesi (Ahimbisibwe ve diğ, 2015, 2017; Rashid ve Khan, 2018; Garousi ve diğ, 2019).
	Müşterilerin kendi alanındaki deneyimi (Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019)
Ekip faktörleri	Çevik ekip üyelerinin yetenekleri (Fortune ve White, 2006; Chow ve Cao, 2008; Misra ve diğ, 2009; Stankovic ve diğ, 2013; Lehtinen ve diğ, 2014; Silva ve Santos, 2015; Abrar ve diğ, 2019; Arcos-Medina ve Mauricio, 2019; Hughes ve diğ, 2020; Tam ve diğ, 2020)
	Çevik ekip üyelerinin yazılım geliştirme konusundaki tecrübesi (Fortune ve White, 2006; Chow ve Cao, 2008; Sheffield ve Lemétayer, 2013; Stankovic ve diğ, 2013; Serrador ve Pinto, 2015; Ahimbisibwe ve diğ, 2015, 2017; Arcos-Medina ve Mauricio, 2019; Garousi ve diğ, 2019; Hughes ve diğ, 2020)
	Çevik ekip üyelerinin mesleki yeteneklerinin teknik eğitim yoluyla geliştirilmesi (Chow ve Cao, 2008; Misra ve diğ, 2009; Wan ve Wang, 2010; Stankovic ve diğ, 2013; Silva ve Santos, 2015; Fatema ve Sakib, 2017; Abrar ve diğ, 2019)
	Proje yöneticisinin bilgisi (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Engelbrecht ve diğ, 2017; Arcos-Medina ve Mauricio, 2019)
	Proje yöneticisinin tecrübesi (Fortune ve White, 2006; Arcos-Medina ve Mauricio, 2019; Kanjanda ve Tuan, 2020).
	Ekip üyelerinin motivasyonu (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Lehtinen ve diğ, 2014; Silva ve Santos, 2015; Fatema ve Sakib, 2017; Arcos-Medina ve Mauricio, 2019)
	Kendi kendini örgütleyen ekip üyeleri (Chow ve Cao, 2008; Stankovic ve diğ, 2013)

Çizelge 5.3. (devam) : Çevik kritik başarı faktörleri.

Kavram	Kritik başarı faktörleri
Organizasyonel faktörler	Çevik tarzda çalışma ortamı (Chow ve Cao, 2008; Misra ve diğ, 2009; Sheffield ve Lemétayer, 2013; Stankovic ve diğ, 2013; Arcos-Medina ve Mauricio, 2019)
	Hiyerarşik olmayan işbirlikçi yönetim tarzı (Milosevic ve Patanakul, 2005; Chow ve Cao, 2008; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2015, 2017; Abrar ve diğ, 2019; Arcos-Medina ve Mauricio, 2019)
	Kurumsal olarak kabul edilen değişim yönetimi (Milosevic ve Patanakul, 2005; Fortune ve White, 2006; Wan ve Wang, 2010; Inayat ve diğ, 2015; Ahimbisibwe ve diğ, 2015, 2017; Arcos-Medina ve Mauricio, 2019; Hughes ve diğ, 2020; Rashid ve Khan, 2018; Rashid ve diğ, 2021)
	İyi tanımlanmış vizyon ve misyon (Milosevic ve Patanakul, 2005; Wan ve Wang, 2010; Ahimbisibwe ve diğ, 2015, 2017; Arcos-Medina ve Mauricio, 2019)
	Üst yönetim desteği (Milosevic ve Patanakul, 2005; Fortune ve White, 2006; Chow ve Cao, 2008; Wan ve Wang, 2010; Sheffield ve Lemétayer, 2013; Silva ve Santos, 2015; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2015, 2017; Abrar ve diğ, 2019; Arcos-Medina ve Mauricio, 2019; Hughes ve diğ, 2020)
Çevik süreç faktörleri	Ekip üyeleri arasında güçlü iletişim ve koordinasyon (Chow ve Cao, 2008; Misra ve diğ, 2009; Inayat ve diğ, 2015; Silva ve Santos, 2015; Ahimbisibwe ve diğ, 2015, 2017; Fatema ve Sakib, 2017; Rashid ve Khan, 2018; Garousi ve diğ, 2019; Rashid ve diğ, 2021)
	Çevik etkinlikleri doğru şekilde uygulama (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Inayat ve diğ, 2015; Serrador ve Pinto, 2015; Perkusich ve diğ, 2017; Abrar ve diğ, 2019; Arcos-Medina ve Mauricio, 2019; Rashid ve diğ, 2021; Cucolaş ve Russo, 2023)
	İlerleme izleme ve kontrol mekanizması (Fortune ve White, 2006; Misra ve diğ, 2009; Ahimbisibwe ve diğ, 2015, 2017; Perkusich ve diğ, 2017; Garousi ve diğ, 2019; Cucolaş ve Russo, 2023)
	Doğru miktarda dokümantasyon (Chow ve Cao, 2008; Purna Sudhakar, 2012; Stankovic ve diğ, 2013; Perkusich ve diğ, 2017; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019; Rashid ve diğ, 2021)

Çizelge 5.3. (devam) : Çevik kritik başarı faktörleri.

Kavram	Kritik başarı faktörleri
Teknik faktörler	Etkin yazılım test süreci (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Lehtinen ve diğ, 2014; Inayat ve diğ, 2015; Silva ve Santos, 2015; Arcos-Medina ve Mauricio, 2019)
	Programlama standartlarına uyum (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Silva ve Santos, 2015; Arcos-Medina ve Mauricio, 2019)
	Yeniden düzenleme faaliyetleri (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Inayat ve diğ, 2015; Perkusich ve diğ, 2017; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019)
	Çalışan yazılımın düzenli teslimatı (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Lehtinen ve diğ, 2014; Perkusich ve diğ, 2017; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019)
	Doğru araç ve teknoloji seçimi (Chow ve Cao, 2008; Wan ve Wang, 2010; Stankovic ve diğ, 2013; Hughes ve diğ, 2020)
	Etkin risk analizi yönetimi (Fortune ve White, 2006; Chow ve Cao, 2008; Stankovic ve diğ, 2013; Arcos-Medina ve Mauricio, 2019; Hughes ve diğ, 2020)
	Sadelik (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Silva ve Santos, 2015; Rashid ve Khan, 2018; Arcos-Medina ve Mauricio, 2019)
Proje faktörleri	Proje kritikliği (Chow ve Cao, 2008; Howell ve diğ, 2010; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019)
	Aciliyet (Howell ve diğ, 2010; Ahimbisibwe ve diğ, 2015; Garousi ve diğ, 2019)
	Teknik karmaşıklık (Howell ve diğ, 2010; Ahimbisibwe ve diğ, 2015, 2017; Anthony, 2019; Garousi ve diğ, 2019; Menezes ve diğ, 2019; Kanjanda ve Tuan, 2020; Ciric Lalic ve diğ, 2022)
	Teknolojik belirsizlik (Yeo, 2002; Howell ve diğ, 2010; Sheffield ve Lemétayer, 2013; Ahimbisibwe ve diğ, 2015, 2017; Anthony, 2019; Garousi ve diğ, 2019; Menezes ve diğ, 2019; Ciric Lalic ve diğ, 2022)
	Kapsam değişikliği (Ahimbisibwe ve diğ, 2015, 2017; Abrar ve diğ, 2019; Anthony, 2019; Garousi ve diğ, 2019; Menezes ve diğ, 2019)
	Büyük proje boyutu (Ahimbisibwe ve diğ, 2015, 2017; Anthony, 2019; Garousi ve diğ, 2019)

5.2.1. Müşteri faktörleri

5.2.1.1. Proje boyunca güçlü müşteri desteği

Çevik ekip üyeleri, proje boyunca güçlü müşteri desteği arzulamaktadır. Stankovic ve diğ. (2013) müşteri desteğine sahip olmanın çevik yazılım geliştirme projelerinin başarısına katkıda bulunduğunu ve müşteri bağlılığı eksikliğinin yazılım geliştirme projelerinde ilk 10 risk faktörü arasında yer aldığını belirtmiştir (Menezes ve diğ, 2019). Çevik uygulayıcılar, müşteri bağlılığının projelerin kararlaştırılan süre içinde tamamlanmasına yardımcı olduğunu vurgulamışlardır. Bu görüş, müşteri desteğinin projenin zamanında tamamlanma olasılığını artırdığını bildiren Yetton ve diğ. (2000) çalışmalarıyla da doğrulanmıştır. Proje boyunca iş süreçleri sahiplerinin desteği, müşteri odaklı bir yaklaşımın vazgeçilmez bir parçasıdır. Bu yaklaşım, müşteri memnuniyetini en üst düzeyde tutmayı amaçlar ve projenin başarılı bir şekilde tamamlanmasını sağlayan kritik unsurlar arasında yer almaktadır. Müşteri desteği, müşteri ihtiyaçlarını anlama, onlarla etkili iletişim kurma ve gereksinimlerini karşılamaya odaklanır. Çevik yazılım projelerinde müşteri desteğini sağlamanın önemi, proje sürecinde müşteri beklentilerinin ve gereksinimlerinin doğru bir şekilde anlaşılmasını sağlar. Müşteri odaklı bir yaklaşım benimseyen ekipler, müşteri ihtiyaçlarını önceliklendirir ve projenin her aşamasında müşteri geri bildirimlerini dikkate alır. Bu sayede proje süreci müşteri beklentilerine uygun olarak yönlendirilir ve müşteri memnuniyeti en üst düzeyde tutulur. Sonuç olarak çevik yazılım projelerinde iş süreçleri sahiplerinin desteğini almak, projenin başarılı bir şekilde tamamlanmasını ve müşteri memnuniyetinin en üst düzeyde tutulmasını sağlayan önemli bir başarı faktörüdür.

5.2.1.2. Güçlü müşteri ilişkisi

Çevik yazılım proje yönetimi, günümüz iş dünyasında giderek daha fazla popülerlik kazanan bir yaklaşım haline gelmiştir. Bu yaklaşım, projelerin başarısını etkileyen çeşitli kritik faktörlerle birlikte, müşteri ilişkilerinin önemini vurgulamaktadır. Güçlü müşteri ilişkisi kavramı, çevik yazılım projelerinin başarısında önemli bir faktördür. İyi bir müşteri ilişkisi, sadece müşteri memnuniyetini artırmakla kalmaz, aynı zamanda projenin başarılı bir şekilde tamamlanmasına da katkı sağlar. Ayrıca müşteri beklentilerini anlamayı kolaylaştırır ve projenin doğru bir şekilde yönlendirilmesini sağlar. Bu yaklaşım, müşteriyle sürekli iletişim ve etkileşim içinde olmayı ve proje

sürecinde onları sürekli olarak bilgilendirmeyi içerir. Çevik İlkeler, iyi müşteri ilişkilerinin değerini vurgulamakta ve “İş süreçlerinin sahipleri ve yazılımcılar proje boyunca her gün birlikte çalışmalıdırlar” demektedir (Agile Manifesto, 2001b). Akademik çalışmalar da müşteriler ve yazılım geliştirme ekibi üyeleri arasındaki ilişkinin proje başarısı için önemli olduğunu vurgulamaktadır (Chow ve Cao, 2008; Purna Sudhakar, 2012; Stankovic ve diğ., 2013; Abrar ve diğ., 2019). Sonuç olarak çevik yazılım projelerinde iyi müşteri ilişkisinin sağlanması, projenin başarısını belirleyen önemli bir faktördür. Süreç sahipleri ile güçlü bir ilişki kurmak, projenin başarılı bir şekilde tamamlanmasını sağlar ve müşterinin memnuniyetini artırarak uzun vadeli iş ilişkilerinin oluşmasına katkıda bulunur.

5.2.1.3.Müşterilerin BT alanındaki profesyonel yeteneklerinin geliştirilmesi

Çevik yazılım projelerinin başarısında kritik bir rol oynayan faktörlerden biri iş birimlerinin BT alanındaki profesyonel yeteneklerini geliştirmesidir. Müşterilerin BT alanındaki yeteneklerinin geliştirilmesi, iş birimlerinin proje sürecinde daha etkin bir rol oynamasını sağlar ve projenin başarısını artırır. Misra ve diğ. (2009) çalışmasına göre proje ekibi üyelerinin müşterilerle bilgi paylaşımına ve yaşam boyu öğrenmeye açık olmaları, çevik yazılım geliştirme uygulamalarının başarılı olma olasılığını artırmaktadır. Ayrıca müşterilerin teknik bilgisi, yazılım geliştirme sürecine olan ilgi düzeylerini ve geliştirme ekibi üyelerini destekleme isteklerini artırmaktadır. Müşterilerin BT alanındaki profesyonel yeteneklerinin geliştirilmesi, çevik yazılım projelerinde müşterilerin katılımını güçlendiren ve işbirliğini teşvik eden önemli bir unsurdur. Müşterilerin teknik bilgi ve becerilerinin artırılması, proje sürecinde karar verme sürecini hızlandırır ve gereksinimleri daha iyi ifade edilmesini sağlar. Ayrıca müşterilerin daha fazla teknik bilgiye sahip olması, geliştirme ekibiyle daha etkili bir iletişim kurmalarını sağlar ve proje sürecinde daha aktif bir rol oynamalarını teşvik eder.

Çevik yazılım projelerinde müşterinin BT alanındaki profesyonel yeteneklerinin geliştirilmesi, müşterilerinin projeye daha fazla katkı sağlamasını ve proje sürecine daha aktif bir şekilde dahil olmasını sağlar. Projenin hedeflerine ulaşılmasını kolaylaştırır ve projenin başarılı bir şekilde tamamlanmasını sağlar. Bu nedenle çevik yazılım projelerinde müşterilerin BT alanındaki profesyonel yeteneklerinin geliştirilmesi önemli bir odak noktası olmalıdır. Bu kritik başarı faktörü, literatür

tarafından da desteklenmektedir (Misra ve diğ, 2009; Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019).

5.2.1.4.Müşterilerin kendi alanındaki deneyimi

Çevik yazılım proje yönetimi, günümüzde iş dünyasında giderek daha fazla kabul gören bir yaklaşım olmaktadır. Bu yaklaşım, iş süreci sahiplerinin tecrübesinin proje yönetimi için önemli bir kritik başarı faktörü olduğunu vurgular. Müşterinin kendi alanındaki tecrübesi, çevik yazılım projelerinde geliştirme ekibi ve üst yönetim memnuniyetini artırmayı ve buna bağlı olarak proje başarısını desteklemektedir. Ayrıca projenin planlanan zamanda ve kararlaştırılan bütçe dahilinde tamamlanması için önemlidir. İlgili alanda tecrübeli müşteri ürünün gereksinimlerini tam olarak tanımlayabilir ve öngörülemeyen değişikliklere karşı koyabilir. Süreç sahiplerinin iş alanındaki tecrübesi, ürün veya hizmetin sunulma şekli, kullanıcı dostu arayüzler, kolay erişim gibi faktörlerle doğrudan ilişkilidir. Bu nedenle faaliyet gösterilen alanda tecrübeli ve bilgili müşteri ile çalışmak, yazılım ekiplerinin işini kolaylaştırmakla kalmaz ürünü satın alacak/kullanacak kullanıcıların memnuniyetini artırarak işletmenin rekabet avantajını güçlendirir.

Grup toplantısındaki deneyimli yazılım geliştiricilerden biri, “Kendi iş alanında deneyimli bir müşteri, yani iş süreçlerinin sahipleri, iş gereksinimlerini doğru bir şekilde ifade ederek ayrıntılarla çok fazla zaman kaybetmemizi önlemektedir. Deneyimli müşteriler tüm yazılım süreci boyunca doğru geri bildirimler sağlar ve projenin planlanan bütçe dahilinde tamamlanmasına yardımcı olur. Çevik yazılım projelerinde kullanıcı kabul testi süreci çok önemli ve tecrübeli müşteriler titiz kullanıcı kabul testleri sayesinde daha başarılı bir yazılım ürünü geliştirmemize yardımcı oluyor” değerlendirmesinde bulundu. Ayrıca önceki çalışmalar da müşteri deneyimi ile proje başarısı arasındaki pozitif ilişkiyi doğrulamıştır (Ahimbisibwe ve diğ, 2015, 2017; Garousi ve diğ, 2019).

Böylece müşteri faktörlerine ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 1: Müşteri faktörleri süreç verimliliğini olumlu yönde etkiler.
- Hipotez 2: Müşteri faktörleri sürdürülebilir yazılım ürün kalitesini olumlu yönde etkiler.
- Hipotez 3: Müşteri faktörleri paydaş memnuniyetini olumlu yönde etkiler.

5.2.2. Ekip faktörleri

5.2.2.1.Çevik ekip üyelerinin yetenekleri

Çevik ekip üyelerinin yetenekleri, yazılım projelerindeki başarı için önemli bir belirleyicidir ve ekip üyelerinin teknik bilgi, beceri ve deneyimlerini kapsar. Proje ekibi, projenin başarıya ulaşması için teknik, yönetsel ve mesleki beceriler açısından yeterli yeteneğe sahip kişilerden oluşmalıdır. Teknik bilgi ve becerilere sahip olan bir ekip, proje sürecinde karşılaşılan zorlukları daha etkili bir şekilde aşabilir ve hedeflere daha hızlı bir şekilde ulaşabilir. Ayrıca projenin daha verimli bir şekilde yönetilmesi sağlanır ve olası risklerin minimize edilmesine katkıda bulunur.

Çevik yazılım projelerinde ekip üyelerinin yeteneklerinin ve kabiliyetlerinin sürekli olarak geliştirilmesi ve güçlendirilmesi de önemlidir. Ekip üyelerinin teknik bilgi ve becerilerini artırmak için eğitim ve gelişim fırsatları sağlanmalıdır. Ayrıca ekip içi işbirliği ve bilgi paylaşımı teşvik edilmeli ve ekip üyelerinin motivasyonu ve bağlılığı artırılmalıdır. Ekip üyelerinin yeteneklerinin ve kabiliyetlerinin sürekli olarak geliştirilmesi, projenin başarılı bir şekilde tamamlanmasını sağlar ve proje sonuçlarının kalitesini artırır. Ekip üyelerinin yeteneklerinin proje başarısı için önemi çeşitli akademik çalışmalarla da teyit edilmiştir (Fortune ve White, 2006; Chow ve Cao, 2008; Purna Sudhakar, 2012; Lehtinen ve diğ, 2014; Abrar ve diğ, 2019; Hughes ve diğ, 2020). Bunun yanında ekip üyelerinin yeteneklerinin yetersizliği (örneğin, yetersiz mesleki beceriler) şirketler için bir risk faktörü olarak sınıflandırılmaktadır (Menezes ve diğ, 2019).

5.2.2.2.Çevik ekip üyelerinin yazılım geliştirme konusundaki tecrübesi

Çevik ekip üyelerinin yazılım geliştirme deneyimleri, projenin başarılı bir şekilde tamamlanmasında önemli bir rol oynar. Tecrübeli bir ekip, projenin gereksinimlerini daha iyi anlar ve karşılaşılan zorlukları daha etkili bir şekilde çözebilir. Ayrıca yazılım geliştirme sürecinin belirli aşamalarında daha hızlı aksiyon alabilir.

Çevik yazılım projelerinde ekip üyelerinin yazılım geliştirme deneyimlerini sürekli olarak geliştirmesi önemlidir. Deneyim kazanma, eğitim ve gelişim fırsatları sağlayarak, deneyim paylaşımını teşvik ederek ve yeni teknolojilere ve yöntemlere adapte olmalarını sağlayarak yapılabilir. Ayrıca tecrübeli ekip üyelerinin, daha az deneyime sahip olanlara rehberlik etmeleri ve onları desteklemeleri de önemlidir.

Yazılım geliştirme projelerinde, değişikliklere kolayca uyum sağlayan tecrübeli ekip üyelerine ihtiyaç duyulduğu akademik çalışmalarla da doğrulanmıştır. (Lindvall ve diğ., 2002; Turner ve Boehm, 2003; Chow ve Cao, 2008; Wan ve Wang, 2010; Purna Sudhakar, 2012; Hughes ve diğ., 2020). Hughes ve diğ. (2020) çalışmasına göre, başarılı projeler, ürün sürümü sırasında deneyim ve beceriye sahip özel personel gerektirmektedir. Bu nedenle çevik yazılım projelerinde ekip üyelerinin yazılım geliştirme tecrübelerinin önemi vurgulanmalı ve sürekli gelişim desteklenmelidir.

5.2.2.3.Çevik ekip üyelerinin mesleki yeteneklerinin teknik eğitim yoluyla geliştirilmesi

Ekip üyelerinin mesleki yeteneklerinin geliştirilmesi sürecinde teknik eğitim önemli bir rol oynar. Eğitim, ekip üyelerinin mevcut beceri setlerini genişletmelerine ve yeni teknolojilere hakim olmalarına fırsat tanımaktadır. Bu eğitimler, genellikle uzmanlar tarafından verilen interaktif kurslar, atölye çalışmaları ve uygulamalı egzersizler şeklinde düzenlenir. Ayrıca çevik ekip üyeleri için teknik eğitimlerin, günlük iş akışına entegre edilerek pratik deneyim kazandıracak şekilde planlanması da sürekli eğitim fikrinin yaygınlaşmasını sağlar.

Çevik ekip üyelerinin mesleki yeteneklerinin eğitim yoluyla geliştirilmesi, sadece mevcut projelerde değil, gelecekteki projelerde de başarı için kritik bir faktördür. Bu süreç, ekip üyelerinin motivasyonunu artırır, işbirliğini teşvik eder ve ekip içi iletişimi güçlendirir. Ayrıca teknik eğitimler sayesinde ekip üyeleri, projelerde ortaya çıkan teknik zorlukları daha hızlı bir şekilde çözebilir ve daha kaliteli ürünlerin teslimatını sağlayabilirler.

Ekip üyelerinin mesleki yeteneklerinin teknik eğitim yoluyla geliştirilmesi, yazılım projelerinin başarı olasılığını artırır (Fortune ve White, 2006; Chow ve Cao, 2008). Hızlı teknolojik değişimler nedeniyle ekip üyeleri için resmi veya gayri resmi teknik eğitim şarttır. Çevik metodolojiler, çevikliği ve sürekli öğrenmeyi vurgular. Bu nedenle hantal ve uzun süreli eğitimler yerine informal eğitimler tercih edilmektedir. Çevik metodolojilerin benimsenmesi, güçlü iletişim yoluyla ekip üyeleri arasında örtülü bilginin yayılmasını gerektirir (Misra ve diğ., 2009). Sonuç olarak ekip üyelerinin güncel kalmasını sağlar, proje kalitesini artırır ve müşteri memnuniyetini sağlayacak ürünlerin geliştirilmesine katkıda bulunur. Dolayısıyla şirketlerin çevik

yaklaşımı benimsemeleri durumunda sürekli eğitim faaliyetlerinin yaygınlaştırılmasına yönelik yatırım yapmaları önemlidir.

5.2.2.4.Proje yöneticisinin bilgisi

Stankovic ve diğ. (2013) çalışmasında proje yöneticisi çevik ekip üyelerinden biri olarak kabul edildiğinden proje yöneticisinin bilgisi ekip kabiliyeti altında sınıflandırılmıştır. Proje yöneticisinin yetkinliği ve bilgisi, yazılım geliştirme projeleri için başarı faktörleri olarak kabul edilmektedir (Fortune ve White, 2006; Chow ve Cao, 2008; Purna Sudhakar, 2012; Stankovic ve diğ, 2013; Engelbrecht ve diğ, 2017). Çevik yazılım geliştirme süreçleri, günümüzde iş dünyasında giderek daha fazla kabul gören bir yaklaşımdır. Bu yaklaşımda proje yöneticisinin bilgi ve deneyimi, projenin başarısı için kritik bir öneme sahiptir. Proje yöneticisinin bilgisi, sadece teknik yetkinliklerle sınırlı kalmaz, aynı zamanda liderlik, iletişim, problem çözme ve takım yönetimi gibi becerileri de kapsar. Bu becerilerin etkili bir şekilde kullanılması, projenin başarıyla tamamlanması için kritiktir.

Proje yöneticisi, proje sürecinin her aşamasında önemli bir rol oynamaktadır. Başlangıçta, proje gereksinimlerinin ve kapsamının belirlenmesi sürecinde, proje yöneticisinin teknik bilgisi ve endüstriyel deneyimi, doğru stratejilerin belirlenmesine ve hedeflerin netleştirilmesine yardımcı olur. Ayrıca proje yöneticisinin bilgisi, yazılım geliştirme ekibi ve iş süreçlerinin sahipleriyle etkili bir iletişim kurmasını sağlar ve motivasyonunu artırarak projenin başarısına katkıda bulunur.

Teknolojideki hızlı değişimler ve iş gereksinimlerinin evrimi, proje yöneticisinin sürekli olarak yeni bilgiler edinmesini ve yeni teknikleri öğrenmesini gerektirir. Bu nedenle proje yöneticileri düzenli eğitimlere katılmalı, trendleri takip etmeli ve mesleki gelişimlerini sürekli olarak artırmalıdır. Deneyimli yazılımcılardan biri, bire bir görüşmede proje yöneticisinin bilgi birikiminin öneminin altını çizmiştir: “Proje yöneticisinin sadece teknik bilgiye sahip olması yeterli değildir. Proje yöneticisi kurumun politika ve uygulamalarına aşina olmalıdır. Proje yöneticisi, proje süreçlerini başından sonuna kadar takip etmeli ve darboğaza yol açması muhtemel prosedürleri öngörebilecek kadar teknik ve alan bilgisine sahip olmalıdır. Proje yöneticisi sahip olduğu bilgi birikimi ile ekip üyelerini motive etmeli ve yönlendirmelidir”. Sonuç olarak alanında bilgili yönetici, çevik yazılım projelerinin başarısı için kritik bir faktördür. Proje yöneticisinin derinlemesine teknik bilgisi, liderlik becerileri ve

iletiřim yetenekleri, projenin başarılı bir şekilde tamamlanmasını sağlar ve müşteri memnuniyetini artırır. Bu nedenle şirketlerin çevik yaklaşımı benimsemeleri durumunda proje yöneticisinin bilgisinin sürekli geliştirilmesine yönelik yatırımlar yapılmalıdır.

5.2.2.5.Proje yöneticisinin tecrübesi

Yazılım geliştirme projelerinde proje yöneticisinin tecrübesi çok önemlidir. Bilgi ve deneyim eksikliği projenin başarısızlığına neden olacak risklerden biridir (Engelbrecht ve diğ, 2017). Başarılı bir proje için yöneticiler stresle başa çıkabilmeli, insanları yönetebilmeli, proje için zaman ayırabilmeli, problem çözme ve iletişim becerilerine sahip olmalıdır (Purna Sudhakar, 2012).

Proje yöneticisinin tecrübesi, sadece teknik bilgi ve becerilerle sınırlı kalmaz, aynı zamanda liderlik, iletişim ve problem çözme gibi alanlarda da derinlemesine bilgi ve yetenek gerektirir. Projenin başarılı bir şekilde tamamlanması için kritiktir ve çevik yazılım projelerinde başarı için temel bir gerekliliktir. Proje sürecinin her aşamasında önemli bir rol oynar. Başlangıçta proje yöneticisi, proje gereksinimlerini belirleme, kapsamı tanımlama ve hedefleri netleştirme sürecinde liderlik etmelidir. Deneyimli bir yönetici, proje ekibiyle etkili bir iletişim kurabilir, kaynakları etkin bir şekilde yönetebilir ve proje sürecinde ortaya çıkan zorlukları başarıyla aşabilir.

5.2.2.6.Ekip üyelerinin motivasyonu

Çevik İlkelerden biri de ekip üyelerinin motivasyonudur: “Projelerin temelinde motive olmuş bireyler yer almalıdır. Onlara ihtiyaçları olan ortam ve destek sağlanmalı, işi başaracakları konusunda güven duyulmalıdır” (Agile Manifesto, 2001b). Motive olmuş ekip üyeleri çevik yazılım projelerinin başarısını olumlu anlamda etkiler (Chow ve Cao, 2008; Stankovic ve diğ, 2013). Benzer şekilde, Fatema ve Sakib (2017) motive olmuş kişilerle çalışmanın ekip etkinliğini artırdığını bildirmiştir.

Ekip üyelerinin bireysel ve takım hedeflerine katkılarının farkında olmaları önemlidir. Hedeflerin net ve erişilebilir olması, motivasyonun sürdürülmesine yardımcı olur. Ayrıca ekip üyelerinin başarılarının tanınması ve ödüllendirilmesi, motivasyonlarını artırmada etkili bir yöntemdir. Ödüllendirme, hem maddi hem de manevi ödülleri içerebilir ve ekip üyelerini daha fazla çaba göstermeye teşvik eder.

Ekip üyelerinin motivasyonunu sürdürmek için sürekli çaba gereklidir. Scrum master ve proje yöneticisi, ekip üyelerinin ihtiyaçlarını anlamalı, onları desteklemeli ve projenin ilerleyişine katkılarını takdir etmelidir. Ekip içi iletişimin güçlendirilmesi ve işbirliğinin teşvik edilmesi de motivasyonu artırır. Ekip üyeleri, projenin amacına bağlılık duyduklarında ve kendilerini değerli hissettiklerinde, daha yüksek bir motivasyon seviyesine sahip olurlar ve bu da projenin başarılı bir şekilde tamamlanmasına yardımcı olur.

Uzman görüşmelerinde bir çevik uygulayıcı, “Yazılım sektörü doğası gereği çok streslidir. Bu nedenle verimli çalışmak ve görevleri planlanan sürede bitirmek için çalışanların motivasyonunun yüksek olması gerekir. Üst yönetim ve insan kaynakları departmanı, ekip üyeleri arasındaki iletişim ve çalışanlar arasındaki motivasyon gibi faktörler konusunda sorumluluk almalıdır. Sadece terfi vermek veya maaş düzenlemeleri yapmak yeterli değildir. Çalışanların motivasyonunu artırmanın daha iyi bir yolu bulunmalıdır” değerlendirmesinde bulundu. Sonuç olarak ekip üyelerinin motivasyonu, çevik yazılım projelerinin başarısı için önemlidir. Motive edilmiş ekip üyeleri, daha sıkı çalışır, daha yaratıcı çözümler üretir ve proje hedeflerine daha hızlı ulaşılmasına yardımcı olur. Bu nedenle şirketlerin çevik yaklaşımı benimsemeleri durumunda ekip üyelerinin motivasyonunu artırmaya yönelik çözüm odaklı davranış sergilemeleri şarttır.

5.2.2.7.Kendi kendini örgütleyen ekip üyeleri

Çevik yazılım geliştirme süreçlerinde kendiliğinden örgütlenen ekip üyeleri, proje başarısı için kritik bir unsurdur. Kendiliğinden örgütlenen ekip üyeleri, belirlenmiş roller ve yetkiler çerçevesinde hareket etmek yerine, proje hedeflerine ulaşmak için gerekli adımları kendi başlarına planlayabilir, uygulayabilir ve değerlendirebilirler. Bu esneklik ve özerklik, ekip üyelerinin işbirliğini artırır, hızlı kararlar almayı sağlar ve projenin başarıyla tamamlanmasını destekler.

Çevik İlkelere göre kendi kendini organize eden ekip üyeleri önemlidir, “En iyi mimariler, gereksinimler ve tasarımlar kendi kendini örgütleyen takımlardan ortaya çıkar” (Agile Manifesto, 2001b). Kendi kendini örgütleyen ekip üyeleri arasında itme sistemi yerine çekme sistemi vardır. Görevlerin ekip üyelerine atanmasından ziyade ekip üyeleri gereksinimleri ve tasarımları kendi kendine organize eder. Yazılım ekibi, işler için neyin en iyi olduğuna karar veren ekip üyelerinden oluşmaktadır. Geliştirme

ekibi, şelale yaklaşımında olduğu gibi komuta ve kontrol ilişkilerine ihtiyaç duymaz (Moe ve diğ, 2010). Tutarlı, kendi kendini organize eden ekip çalışması, üretkenliği artıran ve öğrenmeyi teşvik eden önemli bir kritik başarı faktörü olarak kabul edilir (Chow ve Cao, 2008; Stankovic ve diğ, 2013).

Kendiliğinden örgütlenen ekip üyelerinin sahip olduğu yeteneklerin, proje geliştirme sürecinde çeşitli avantajları vardır. Öncelikle ekip üyeleri, proje gereksinimlerine hızlı bir şekilde uyum sağlayabilir ve değişen durumlara esnek bir şekilde yanıt verebilirler. Ayrıca ekip içindeki iletişim ve işbirliği, herkesin fikirlerini paylaşmasını ve en iyi çözümleri bulmasına yardımcı olur. Bu da projenin kalitesini artırır ve müşteri memnuniyetini yükseltir.

Çevik yazılım projelerinde proje yöneticisi, ekip üyelerine güvenmeli, onları motive etmeli ve gerektiğinde rehberlik etmelidir. Ayrıca ekip içinde şeffaf bir iletişim ortamı sağlamak ve takım üyelerinin gelişimini desteklemek de önemlidir. Bu şekilde kendiliğinden örgütlenen ekip üyeleri, projenin başarısı için en iyi potansiyellerini sergileyebilirler. Sonuç olarak kendiliğinden örgütlenen ekip üyeleri, çevik yazılım projelerinin başarısı için önemlidir. Bu yaklaşım, esneklik, hız ve müşteri odaklılık gibi temel değerleri vurgular ve proje ekibinin verimliliğini artırır. Bu nedenle şirketlerin çevik yaklaşımı benimsemeleri durumunda ekip üyelerinin kendiliğinden örgütlenme yeteneklerini geliştirmesine destek olması gerekmektedir.

Ekip faktörlerine ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 4: Ekip faktörleri süreç verimliliğini olumlu yönde etkiler.
- Hipotez 5: Ekip faktörleri sürdürülebilir yazılım ürün kalitesini olumlu yönde etkiler.
- Hipotez 6: Ekip faktörleri paydaş memnuniyetini olumlu yönde etkiler.

5.2.3. Organizasyonel faktörler

5.2.3.1. Çevik tarzda çalışma ortamı

Çevik İlkeler’de “Bir yazılım takımında bilgi alışverişinin en verimli ve etkin yöntemi yüzyüze iletişimidir” denmektedir (Agile Manifesto, 2001b). Bu tarz çalışma ortamı, esneklik, işbirliği ve hız gibi çevikliğin temel prensiplerine dayanır ve proje ekibinin etkin bir şekilde işbirliği yapmasını sağlar. Çevik tarzı çalışma ortamı, geleneksel

hiyerarşik yapıları kabul etmez ve herkesin fikirlerini paylaşabileceği, yenilikçi çözümler bulabileceği bir ortam oluşturmaya çalışır.

Çevik tarzı çalışma ortamının temel özelliklerinden biri fiziksel ve dijital olarak erişilebilir bir ortam sağlamaktır. Ekip üyeleri, işbirliği yapmak ve iletişim kurmak için uygun araçlara ve teknolojilere erişim sağlayarak, projenin hızlı ve etkili bir şekilde ilerlemesini sağlarlar. Ayrıca açık ofis düzenlemeleri ve çevirim içi platformlar, ekip üyelerinin bir araya gelmesini ve fikir alışverişi yapmasını kolaylaştırır.

Çevik tarzı çalışma ortamı, ekip üyelerinin motivasyonunu artırır. Esnek çalışma saatleri, evden çalışma imkanı ve çalışanların kendi işlerini organize etme özgürlüğü, ekip üyelerinin şirkete olan aidiyet duygusunu artırır ve daha verimli çalışmalarını sağlar. Bu da proje hedeflerine ulaşma sürecini hızlandırır ve projenin başarılı bir şekilde tamamlanmasını destekler.

Çok sayıda makale, çevik yazılım geliştirme projeleri için bir başarı faktörü olarak birlikte çalışan ekip üyelerinden bahsetmektedir (Lindvall ve diğ, 2002; Chow ve Cao, 2008; Stankovic ve diğ, 2013). COVID-19 salgınıyla birlikte şirketlerin çalışma tarzı değişmiştir. Birçok şirket hibrit bir çalışma modeli benimsemektedir. Hem ofiste çalışmayı hem de uzaktan çalışma tarzını entegre etmek, çalışanların motivasyonunu ve şirkete olan bağlılığını artırmaktadır (Cucolaş ve Russo, 2023). Sonuç olarak çevik tarzı çalışma ortamı, çevik yazılım projelerinin başarısı için kritik bir faktördür. Esneklik, işbirliği ve motivasyon gibi özellikleriyle, çevik tarzı çalışma ortamı, proje başarı oranını artırır ve ekip üyelerinin performansını optimize eder. Bu yüzden şirketlerin çevik yaklaşımı benimsemeleri durumunda, çevik tarzı çalışma ortamını oluşturmak için gerekli aksiyonları almaları gerekmektedir.

5.2.3.2.Hiyerarşik olmayan işbirlikçi yönetim tarzı

Geleneksel yönetim anlayışından farklı olarak, çevik yazılım geliştirme süreçlerinde hiyerarşik olmayan işbirlikçi yönetim tarzı, proje başarısı için önemli bir faktördür. Bu yaklaşım, ekip üyelerinin yeteneklerini ve fikirlerini en iyi şekilde değerlendirerek, katılımcı bir karar alma süreci sağlar. Hiyerarşik olmayan işbirlikçi yönetim tarzı, herkesin eşit derecede katkı sağlayabileceği bir ortam oluşturur ve proje ekibinin potansiyelini en üst düzeye çıkarır.

Hiyerarşik olmayan işbirlikçi yönetim tarzı, ekip içerisinde liderlik rolünü yeniden tanımlar. Geleneksel yönetimde olduğu gibi takım lideri sadece komut veren bir figür değil, aynı zamanda ekip üyelerinin rehberi ve destekçisidir. Ekip üyelerini cesaretlendirir, motive eder ve onları güçlendirir, böylece herkesin proje başarısı için gereken sorumlulukları üstlenmesini sağlar.

Hiyerarşik olmayan işbirlikçi yönetim tarzının bir diğer önemli özelliği, şeffaflık ve iletişimidir. Scrum master, ürün sahibi ve ekip üyeleri birbirleriyle sürekli iletişim halinde olmalı ve proje sürecinin her aşamasında birbirlerine açık ve şeffaf olmalıdır. Bu sayede ekip üyeleri proje hedeflerine odaklanır ve gerektiğinde stratejik değişikliklere hızla adapte olabilirler.

Çevik yazılım geliştirme uygulayan kuruluşlar değişikliklere hızla adapte olmalıdır (Misra ve diğ., 2009). Değişimi kucaklayan bir organizasyon yaratmak için bürokratik bir yönetim tarzı yerine işbirlikçi bir yönetim tarzı kullanılmalıdır. Chow ve Cao (2008) ve Stankovic ve diğ. (2013) çalışmalarında çevik yazılım geliştirme projeleri için bir kritik başarı faktörü olarak işbirlikçi bir yönetim tarzı önermiştir.

5.2.3.3.Kurumsal olarak kabul edilen değişim yönetimi

Çevik İlke’de, “Değişen gereksinimler yazılım sürecinin son aşamalarında bile kabul edilmelidir. Çevik süreçler, değişimi müşterinin rekabet avantajı için kullanır” denmektedir (Agile Manifesto, 2001b). Çevik geliştirme süreçlerinde, kurumsal olarak kabul edilen değişim yönetimi, projenin başarılı bir şekilde tamamlanmasında kritik bir rol oynar. Bu yaklaşım, organizasyonun değişen ihtiyaçlarına hızlı ve etkili bir şekilde adapte olabilme yeteneğine odaklanır. Kurumsal olarak kabul edilen değişim yönetimi, organizasyon içindeki tüm paydaşların işbirliği yapmasını sağlar ve değişimin başarılı bir şekilde gerçekleşmesine yardımcı olur.

Kurumsal olarak kabul edilen değişim yönetimi, organizasyonel kültürün önemli bir parçası haline gelmiştir. Bu yaklaşım, değişimi engelleyen faktörleri belirleyerek, organizasyon içinde bir değişim kültürü oluşturmayı amaçlar. Bu da çalışanların değişime daha açık olmalarını sağlar ve organizasyonun rekabet avantajını artırır.

Kurumsal olarak kabul edilen değişim yönetimi, sadece organizasyon içindeki bireyler için değil, aynı zamanda paydaşlar için de önemlidir. Organizasyon; müşteriler, tedarikçiler ve diğer iş ortaklarıyla işbirliği yaparak, değişimin tüm yönlerini dikkate

almalıdır. Böylelikle kurum dışındaki değişimler de başarıyla yönetilebilir ve kurumun uzun vadeli sürdürülebilirliği sağlanır.

Sonuç olarak kurumsal kabul edilen değişim yönetimi, çevik yazılım projelerinde başarı için kritik bir faktördür. Kurumsal olarak kabul edilen değişim yönetimi stratejisi, geliştirme ekibi üyelerinin değişikliklere kolayca uyum sağlamasını sağlar. Literatürde birçok çalışmada, değişim yönetimi proje başarı faktörü olarak değerlendirilmiştir (Fortune ve White, 2006; Wan ve Wang, 2010; Rashid ve Khan, 2018; Hughes ve diğ, 2020).

5.2.3.4.İyi tanımlanmış vizyon ve misyon

Vizyon, projenin ve şirketin geleceğine rehberlik eder ve misyon şirketin amacını tanımlar. Yeo (2002), belirsiz iş ihtiyaçlarının ve net olmayan bir vizyonun en önemli beş başarısızlık faktörü arasında yer aldığını tespit etmiştir. Net vizyon ve misyon, proje ekibinin odaklanmasını sağlar ve ortak bir amaç etrafında birleşmelerini teşvik eder.

Kurum içi iyi tanımlanmış vizyon ve misyon, projenin yönetilmesinde ve iletişimde önemli bir rol oynar. Vizyon ve misyon, proje paydaşları arasında bir anlayış ve uyum sağlar ve projenin ilerleyişini yönlendiren bir kılavuz görevi görür. Bu sayede proje ekibi üyeleri kendi rollerini daha iyi anlar ve projenin başarıyla tamamlanması için gereken adımları atarlar.

İyi tanımlanmış vizyon ve misyon, projenin dış paydaşlarına da güven ve şeffaflık sağlar. Organizasyon dışındaki kişiler ve kurumlar, projenin amaçlarını ve hedeflerini anlayarak projeye destek olabilirler. Bu destek, projenin başarılı bir şekilde tamamlanmasını sağlar ve kurumun itibarını artırır. 15 yıllık deneyime sahip bir scrum master, “Bir çalışan şirketin misyonunun kendi misyonuyla uyumlu olduğunu fark ettiğinde daha iyi performans göstermeye çalışır ve şirkete bağlılığı artar. Ayrıca vizyoner bir yazılımcı, proje başlamadan önce doğru araçları ve teknolojiyi seçer. Bu da potansiyel mali kayıpları ve ekip üyelerinin uğraşması gereken karmaşıklığı azaltır” değerlendirmesinde bulundu. Bu sebeple projenin başarılı bir şekilde tamamlanması için kurumlar iyi tanımlanmış vizyon ve misyon belirlemeye ve bunları proje ekibine etkili bir şekilde iletmeye odaklanması önemlidir.

5.2.3.5.Üst yönetim desteği

Çevik yazılım geliştirme süreçlerinde üst yönetim desteği, projenin başarılı bir şekilde tamamlanması için kritik bir faktördür. Üst yönetim, organizasyonun en üstünde bulunan liderlik kademesidir ve çevik projelerdeki başarının sağlanması için gereken kaynakları ve destekleri sağlar. Bu destek, projenin stratejik hedeflerine ulaşmasını sağlar ve organizasyonun çevik kültürünü benimsemesini teşvik eder.

Üst yönetim desteği, farklı şekillerde olabilir. Finansal kaynak sağlanması, proje ekibinin ihtiyaç duyduğu yeteneklerin temini ve proje sürecinde karşılaşılan engellerin kaldırılması örnek olarak verilebilir. Ayrıca üst yönetim desteği, projenin organizasyon içindeki diğer birimlerle uyumlu bir şekilde yürütülmesini sağlar ve projenin başarılı bir şekilde tamamlanması için gereken ortamı oluşturulmasına yardımcı olur.

Üst yönetim desteği, sadece proje sürecinin başlangıcında değil, aynı zamanda proje boyunca sürekli olarak sağlanmalıdır. Bu destek, projenin hedeflerine ulaşmasını sağlamak için gerekli olan kararların alınmasını ve projenin ilerleyişinin düzenli olarak gözden geçirilmesini içerir. Ayrıca proje ekibine motivasyon sağlar ve projenin başarılı bir şekilde tamamlanmasını teşvik eder.

Akademik çalışmalarda üst yönetim desteğinin yazılım geliştirme projeleri için bir başarı faktörü olduğunu gösterilmektedir (Fortune ve White, 2006; Wan ve Wang, 2010; Sheffield ve Lemétayer, 2013; Ahimbisibwe ve diğ, 2015; Abrar ve diğ, 2019; Garousi ve diğ, 2019). Yazılım projelerinde üst yönetim desteğinin olmaması da bir risk faktörüdür (Lemon ve diğ, 2002; El Emam ve Kori, 2008; Fatema ve Sakib, 2017; Menezes ve diğ, 2019). Yazılım geliştirme ekibi üst yönetim desteğine sahip olmadığı sürece projeler başarıyla tamamlanamaz (Hughes ve diğ, 2020). Bu yüzden tüm proje boyunca üst yönetimin desteğini almak çok önemli bir başarı faktörüdür.

Organizasyonel faktörlere ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 7: Organizasyonel faktörler süreç verimliliğini olumlu yönde etkiler.
- Hipotez 8: Organizasyonel faktörler sürdürülebilir yazılım ürün kalitesini olumlu yönde etkiler.
- Hipotez 9: Organizasyonel faktörler paydaş memnuniyetini olumlu yönde etkiler.

5.2.4. Çevik süreç faktörleri

5.2.4.1.Ekip üyeleri arasında güçlü iletişim ve koordinasyon

Ekip üyeleri arasında güçlü iletişim ve koordinasyon, ekip üyeleri arasında bilgi paylaşımını ve işbirliğini artırır ve ekip çatışmalarını önlemeye yardımcı olur (Ahimbisibwe ve diğ, 2015). Bunların proje başarısı üzerindeki olumlu etkisi çeşitli çalışmalarla teyit edilmiştir (Chow ve Cao, 2008; Jun ve diğ, 2011; Stankovic ve diğ, 2013; Fatema ve Sakib, 2017; Rashid ve Khan, 2018). Ayrıca Çevik Manifesto ekip üyeleri arasındaki iletişim ve koordinasyonun önemini vurgulamakta ve “Süreçler ve araçlar yerine bireyler ve etkileşimler” denmektedir (Agile Manifesto, 2001a). Çevik süreçlerdeki scrum etkinlikleri ve eşli programlama, iletişimi ve koordinasyonu geliştirir ve çevik ekibin daha iyi bir plana ve yazılım tasarımına, daha iyi koda sahip olmasını sağlar (Taromirad ve Ramsin, 2008).

Ekip üyeleri arasında güçlü iletişim ve koordinasyonun sağlanması için stratejik yaklaşımlar gerekmektedir. Bunlardan bazıları; düzenli toplantılar, projenin hedeflerini net bir şekilde belirlemek, görevlerin açıkça tanımlanması ve ekip üyelerinin birbirleriyle düzenli olarak iletişim halinde olmalarını sağlamaktır. Ayrıca iletişim araçlarının etkin bir şekilde kullanılması da önemlidir.

Ekip üyeleri arasında güçlü iletişim ve koordinasyona proje sürecinin her aşamasında önem verilmelidir. Örneğin projenin başında sahip olunan iyi bir iletişim gereksinimlerin daha iyi anlaşılmasına ve hızlı yanıtların verilmesine fayda sağlar. Böylelikle sorunlar erken tespit edilir ve sorunların çözümüne odaklanılır. Proje sürecindeki gecikmeler ve sorunlar minimize edilir. Sonuç olarak güçlü iletişim ve koordinasyon, çevik yazılım geliştirme süreçlerinde kritik başarı faktörlerinden biridir. Proje başarısını artırmak için önem verilmesi gerekmektedir.

5.2.4.2.Çevik etkinlikleri doğru şekilde uygulamak

Çevik yazılım geliştirmedeki etkinlikler, proje ekibinin bir araya gelerek belirli amaçlar doğrultusunda işbirliği yapmasını sağlar. Doğru bir şekilde uygulandığında proje sürecinin verimliliği artar, paydaşlar arasında iletişim güçlenir ve ekibin motivasyonu artar. Çevik tarzda yazılım geliştirme sürecinde beş ana etkinlik vardır: sprint, sprint planlama, günlük toplantılar, sprint gözden geçirme ve sprint retrospektifi. Her bir sprint, süresi bir ayı geçmeyen bir proje olarak düşünülebilir (The

Scrum Guide, 2020). Sprint planlaması sırasında çevik tahmin teknikleri kullanılır. Planlama pokeri, t-shirt boyutu tahmini, nokta oylaması ve kova sistemi çevik tahmin tekniklerinden birkaçıdır (Teslyuk ve diğ, 2022). Günlük toplantılar kullanıcılar ve geliştiriciler arasında güçlü iletişimi destekler (Chow ve Cao, 2008; Stankovic ve diğ, 2013). Mann ve Maurer (2005) günlük toplantıların müşterileri projenin ilerleyişi hakkında güncel tuttuğunu ve müşterilerin yazılım geliştirme ile ilgili kafa karışıklığını önlediğini bildirmiştir. Benzer şekilde Cohn ve Ford (2003) çalışmasında günlük toplantıların önemini vurgulamıştır. Sprint gözden geçirme, her sprintin sonunda yapılan bir toplantı türüdür. Ekip üyeleri çalışma sonuçlarını kilit paydaşlara sunar ve ürün hedefine doğru ilerlemeyi tartışır. Sprint retrospektif toplantıları, ekip üyelerinin nelerin doğru/yanlış yapıldığını ve bir sonraki sprintte nelerin daha iyi yapılabileceğini anlamalarına yardımcı olur. Bu da ürün kalitesini ve projenin verimliliğini artırır (Erdoğan ve diğ, 2018).

Etkinliklerin uygun şekilde uygulanması, öncelikle bu etkinliklerin doğru bir şekilde planlanmasını ve yönetilmesini gerektirir. Örneğin sprint planlama toplantıları, günlük toplantılar ve sprint retrospektifleri gibi etkinlikler, belirli bir gündemle ve katılımcıların etkin bir şekilde yönlendirilmesiyle gerçekleştirilmelidir. Bu toplantıların etkili bir şekilde organize edilmesi, proje ekibinin odaklanmasını sağlar ve süreç boyunca doğru yönde ilerlemesini destekler. Grup toplantısındaki tüm katılımcılar, çeviklikle ilgili etkinliklerin düzgün bir şekilde uygulanmasının proje başarısını artırdığını vurgulamıştır. Sonuç olarak bu etkinlikler, proje ekibinin işbirliğini artırır, iletişimi güçlendirir. Bu yüzden scrum master, ürün sahibi ve geliştirme ekibi, çevikleşme etkinliklerine gereken önemi vererek bu başarı faktörünü sağlamlaştırmalıdır.

5.2.4.3.İlerleme izleme ve kontrol mekanizması

Çevik dostu bir ilerleme izleme/kontrol mekanizması, projenin zamanında ve planlanan bütçe dahilinde tamamlanmasını sağlar. Bu mekanizmalar, projenin ilerleyişini izlemek, riskleri yönetmek ve proje hedeflerine ulaşılmasına yardımcı olur. Çevik yaklaşımın temel prensipleri doğrultusunda ilerleme takibi ve kontrol mekanizmaları, esneklik ve adaptasyon yeteneği sağlayarak projenin başarısını artırır.

Projeyi izleme ve takip etmek için kullanılan scrum eserleri arasında ürün iş listesi, sprint iş listesi ve ürün parçası yer almaktadır. Ürün sahibi ve çevik ekip üyeleri her

bir eserin ilerlemesini bilmelidir. Her bir eserin ilerlemesi ölçülebilir. Bunlar; ürün iş listesi için ürün hedefi, sprint iş listesi için sprint hedefi ve ürün parçası için bitti tanımıdır (The Scrum Guide, 2020).

Proje ilerleme takibi ve kontrol mekanizmaları, genellikle sprint retrospektifleri, günlük toplantılar ve görsel Kanban tahtaları gibi araçlar ve teknikler aracılığıyla gerçekleştirilir. Sprint retrospektifi, proje ekibinin geçmiş dönemdeki performansını değerlendirir ve iyileştirme fırsatlarını belirler. Günlük toplantılar ile ekip üyelerinin günlük işleri ve ilerlemeleri takip edilir. Kanban tahtaları ile de proje ilerlemesini görsel olarak izlemek mümkündür ve ekibin çalışmalarını koordine edilmesine olanak sağlar.

İlerleme takibi ve kontrol mekanizmalarının etkin bir şekilde kullanılması, scrum master, ürün sahibi ve geliştirme takımına, proje sürecinin her aşamasında gerçek zamanlı bilgi sağlar. Bu bilgiler sayesinde proje boyunca performans efektif bir şekilde değerlendirilebilir ve gerektiğinde güncellemeler yapmak için kullanılabilir. Ayrıca riskleri önceden belirlemek ve önlemler almak için de ilerleme takibi ve kontrol mekanizmaları önemlidir.

Sonuç olarak çevik dostu ilerleme takibi ve kontrol mekanizmaları, proje ekibinin performansını değerlendirmek, riskleri yönetmek ve proje hedeflerine ulaşmak için gereken ayarlamaları yapmak için kullanılır. Bu nedenle proje yöneticileri, scrum master, ürün sahibi ve geliştirme ekibi, bu mekanizmaları etkin bir şekilde kullanarak proje başarısını sağlamak için önemli bir sorumluluk üstlenmektedir.

5.2.4.4.Doğru miktarda dokümantasyon

Çevik Manifesto’da, “Kapsamlı dökümantasyondan ziyade çalışan yazılıma değer vermeye kanaat getirdik” ifadesi yer almaktadır (Agile Manifesto, 2001a). Doğru miktarda dokümantasyon, projenin gereksinimlerini anlamının ve belgelemenin yanı sıra etkili bir şekilde çalışan bir yazılım ürünü geliştirmenin önemini vurgulamaktadır. Fakat Çevik Manifesto’daki bu ilke, dokümantasyonun tamamen yok sayılması gerektiği anlamına gelmez. Doğru miktarda dokümantasyon, projenin gereksinimlerini ve hedeflerini açıkça tanımlayarak, ekip üyelerinin işlerini daha etkili bir şekilde yapmalarını sağlar.

Çevik metodolojiler, geleneksel proje yönetimi metodolojilerine göre daha az dokümantasyon gerektirir. Ancak bire bir toplantılarda çevik uygulayıcılar, COVID-

19 salgını döneminde uzaktan çalışma koşullarının etkisinden dolayı örtülü bilgi aktarımının zorlaştığını, bu yüzden de salgın dönemi boyunca daha detaylı dokümantasyona ihtiyaç duyduklarını belirttiler. Değişen gereksinimlere ayak uydurmak için yalnızca yazılım kodunu kullanmak/okumanın yeterli olmadığını, analiz dokümantasyonlarına da gerekli önemin verilmesi gerektiğini belirttiler.

Dokümantasyon miktarı çok önemlidir. Fazla veya yetersiz dokümantasyonun potansiyel etkileri dikkate alınmalıdır. Gereksiz ayrıntılara dalmak, zaman kaybına ve ekibin odaklanmasını zorlaştırmaya neden olabilir. Aksine eksik veya belirsiz dokümantasyon, iletişim eksikliğine yol açabilir ve proje hedeflerinin anlaşılmasını zorlaştırabilir. Bu durumda projenin başarısı tehlikeye girebilir ve süreçte hataların artmasına neden olabilir.

Çevik yazılım geliştirme süreçlerinde doğru miktarda dokümantasyon önemlidir. Gereksiz ayrıntılardan kaçınarak, belirli bir düzeyde dokümantasyon yapmak, projenin gereksinimlerini ve hedeflerini anlamak için çok önemli bir kılavuz, yol göstericidir. Fakat fazla veya yetersiz dokümantasyonun sebep olabileceği etkiler de dikkate alınmalı ve projenin ihtiyaçlarına uygun bir yaklaşım benimsenmelidir.

Çevik süreç faktörlerine ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 10: Çevik süreç faktörleri süreç verimliliğini olumlu yönde etkiler.
- Hipotez 11: Çevik süreç faktörleri sürdürülebilir yazılım ürün kalitesini olumlu yönde etkiler.
- Hipotez 12: Çevik süreç faktörleri paydaş memnuniyetini olumlu yönde etkiler.

5.2.5. Teknik faktörler

5.2.5.1.Etkin yazılım test süreci

Yetersiz yazılım testleri yazılım projelerinin başarısız olmasına neden olabilir (Lehtinen ve diğ, 2014) ve çevik projeler için bir risk faktörüdür (Shrivastava ve Rathod, 2015). Bu nedenle ürünün piyasaya sürülmesinden ve dağıtılmasından önce etkin yazılım testlerinin yapılması çok önemlidir. Çevik test kadranları farklı test yaklaşımlarını tanımlamaktadır (Crispin ve Gregory, 2008). Çeyrek 1 kadranı, birim/bileşen testlerini kapsar. Bileşen testleri genellikle riskleri azaltmak ve

bileşendeki hataları tespit etmek ve önlemek için kodu yazan geliştirici tarafından gerçekleştirilir. Çeyrek 2, işlevsel testler, prototipler, örnekler ve simülasyonlar gibi testleri içerir. Çeyrek 3, çalışan yazılımın iş gereksinimlerini karşılayıp karşılamadığını belirlemek için yapılan testleri içerir. Kullanıcı kabul testleri, kullanılabilirlik testleri ve keşif testleri bunlara örnektir. Çeyrek 4, bir sistemin veya bileşenin işlevsel olmayan gereksinimlerini değerlendirmek için işlevsel olmayan testleri (örneğin, performans testi ve yük testi vb) kapsar (ISTQB, 2022).

Çevik yaklaşım altında yazılım test süreçlerinin etkinliği ve verimliliği büyük önem taşır. Çevik metodolojilerde yazılım testi bir aşama değil, süreç boyunca devam eden bir aktivitedir. Bu süreç, sürekli entegrasyon ve sürekli teslimat prensipleriyle uyumlu olarak işler. Her iterasyonda testlerin otomatikleştirilmesi ve tekrarlanabilirliğinin sağlanması, yazılımın hızla değişen gereksinimlere uyumunu sağlar.

Etkin test süreci çevik projelerin başarısı için kritik bir faktördür. Etkin bir test süreci, yazılımın kalitesini artırır, müşteri memnuniyetini sağlar ve projenin başarılı bir şekilde tamamlanmasına katkıda bulunur. Bu nedenle çevik yaklaşım altında yazılım testi süreçlerine gereken önem ve özenin gösterilmesi gerekmektedir.

5.2.5.2. Programlama standartlarına uyum

Programlama standartlarına uyum, yazılım projelerinde tutarlılık, kalite ve sürdürülebilirlik sağlar. Programlama standartları, kodun anlaşılabilirliğini artırır, bakımını kolaylaştırır ve ekip içinde işbirliğini teşvik eder. Ayrıca bu standartlar sayesinde yazılımın güvenilirliği artar ve hataların erken tespit edilmesi kolaylaşır. Geliştirme ekibi üyeleri standardizasyonları takip etmelidir. Literatürde standart bir mimariyi (CHAOS Report, 2015) ve iyi tanımlanmış kodlama standartlarını takip etmek (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Silva ve Santos, 2015) kritik başarı faktörüdür. Kodlama standartlarını takip etmek, geliştirme ekiplerinin daha verimli bir şekilde yazılım programı geliştirmesine yardımcı olur. Programlama standartları, geliştirme ekibi üyeleri için bir kılavuz görevi görmektedir. Kodlama standartlarına uyulduğunda yazılım ürününün bakımı nispeten kolaylaşır. Bu sayede geliştiriciler kodu daha rahat anlayabilir ve istedikleri zaman güncelleme yapabilirler.

Çevik geliştirme süreçlerinde programlama standartlarına uyum, hem bireysel geliştiricilerin hem de takımın verimliliğini artırır. Standartların belirlenmesi ve uygulanması, kodun konsantre bir şekilde yazılmasını sağlar ve zaman kaybını önler.

Ayrıca standartlara uyumlu kodlar, daha az hata içerir ve bakım maliyetlerini azaltır. Bununla birlikte bu standartların esnek olması ve gerektiğinde güncellenebilir olması önemlidir, çünkü yazılım geliştirme süreçleri sürekli olarak değişen gereksinimlere uyum sağlamalıdır.

Programlama standartlarına uyum, yazılım geliştirme endüstrisinde giderek daha fazla kabul görmektedir. Özellikle çevik yaklaşım altında standartlar takım üyeleri arasında bir kılavuz görevi görür ve yazılımın kalitesini artırarak müşteri memnuniyetini sağlar. Bu yüzden çevik projelerde programlama standartlarına uyumun sağlanması ve sürekli olarak gözden geçirilmesi gerekmektedir. Bu sayede projelerin başarılı bir şekilde tamamlanması ve sürdürülebilir bir yazılım geliştirme ortamının oluşturulması sağlanacaktır.

5.2.5.3.Yeniden düzenleme etkinlikleri

Yeniden düzenleme etkinlikleri çevik bir uygulamadır. Kod yapısının ve yazılım kalitesinin iyileştirilmesini içerir (Fowler ve Beck, 1999). Kodu yeniden düzenleme faaliyetleri, ürün işlevselliğini korurken yazılımın tasarımı, yapısı ve uygulaması gibi işlevsel olmayan nitelikleri de iyileştirmeyi amaçlar. Yeniden düzenleme faaliyetleri, optimum sonuçların elde edilmesini garanti etmeyi ve proje gereksinimlerindeki değişiklikleri karşılamayı amaçlamaktadır (Chow ve Cao, 2008; Stankovic ve diğ., 2013). Yazılımın kalitesini artırmak, bakımını kolaylaştırmak ve sürekli iyileştirme sağlamak amacıyla yapılan düzenlemeleri içerir. Kodun yapısını ve tasarımını değiştirmeden içerideki karmaşıklığı azaltarak, kodun daha anlaşılır, esnek ve sürdürülebilir hale getirir. Ayrıca yeniden düzenleme süreci, yazılımın performansını artırabilir, hataları azaltabilir ve yeniden kullanılabilirliği artırabilir.

Çevik yazılım geliştirme süreçlerinde yeniden düzenleme, yazılım yaşam döngüsünün bir parçası olarak kabul edilmelidir. Yazılımın geliştirme sürecinde sürekli olarak yeniden düzenleme yapılması, teknik borcun azaltılmasına ve gelecekteki değişikliklere hazırlıklı olunmasına yardımcı olur. Böylece yazılımın bakımı daha az maliyetli hale gelir ve müşteri taleplerine daha hızlı yanıt verilir. Yeniden düzenleme, yazılım geliştirme ekibinin teknik becerilerini artırır ve daha iyi bir kod kalitesi için sürekli çaba sarf etmesini teşvik eder.

Yeniden düzenleme etkinlikleri, yazılım geliştirme sürecinin her aşamasında uygulanabilir. Kodun daha iyi anlaşılmasını ve değiştirilmesini sağlar. Çevik ekiplerin

yeniden düzenleme uygulamalarını sadece yazılım geliştirme sürecinin başında veya sonunda değil, sürekli olarak yapmaları önemlidir. Yeniden düzenleme faaliyetlerinin çevik proje başarısı üzerindeki etkisi literatür tarafından da doğrulanmaktadır (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Rashid ve Khan, 2018).

5.2.5.4.Çalışan yazılımın düzenli teslimatı

Çevik metodolojiler, çevik projelerin başarısı için çalışan yazılımların düzenli olarak teslim edilmesine odaklanır (Chow ve Cao, 2008; Stankovic ve diğ, 2013). Teslimat stratejisinin önemi birinci ve üçüncü çevik ilkelerde belirtilmiştir. Bunlar; “En önemli önceliğimiz değerli yazılımın erken ve devamlı teslimini sağlayarak müşterileri memnun etmektir” ve “Çalışan yazılım, tercihen kısa zaman aralıkları belirlenerek birkaç haftada ya da birkaç ayda bir düzenli olarak müşteriye sunulmalıdır” (Agile Manifesto, 2001b). Çalışan yazılımın sürekli teslimatı olmadan, kod entegrasyonu ve test faaliyetleri daha zor hale gelmektedir (Leffingwell, 2008). Kısa iterasyonlar, düzenli sürüm tarihleri, sürekli entegrasyon ve uygun test faaliyetleri ile sürüm sırasında karşılabilecek olası riskler önlenebilir (Shrivastava ve Rathod, 2015).

Çalışan yazılımın düzenli teslimatı, geliştirme sürecinin düzenli aralıklarla kullanılabilir bir yazılımın müşteriye sunulmasını öngörür. Periyodik teslimatlar, yazılım geliştirme sürecinin ilerlemesini takip etmek için değerli bir ölçüttür ve müşteri geri bildirimlerini erken aşamalarda almayı sağlar. Bu da geliştirme sürecindeki eksikliklerin ve gereksinim değişikliklerinin hızlı bir şekilde ele alınmasını mümkün kılar. Bu yüzden periyodik teslimatlar, projenin başarılı bir şekilde yönetilmesi ve müşteri memnuniyetinin sağlanması için önemlidir.

Çevik yazılım geliştirme, yazılım endüstrisindeki hızlı değişime ve müşteri taleplerindeki belirsizliğe yanıt vermek için ortaya çıkmıştır. Geleneksel yazılım geliştirme süreçlerinden farklı olarak, çevik yaklaşım, esneklik ve sürekli müşteri katılımını ön planda tutar. Bu yüzden periyodik yazılım teslimatı, çevik yaklaşımın temel prensiplerinden birini oluşturur ve geliştirme ekibinin başarılı bir şekilde hızlı değişen gereksinimlere adapte olmasını sağlar.

Periyodik teslimatlar aynı zamanda yazılımın erken aşamalarında müşteri geri bildirimlerinin alınmasını sağlar. Bu geri bildirimler, yazılımın doğru yönde ilerlemesini sağlamak ve gereksinim değişikliklerini hızlı bir şekilde ele almak için değerli birer rehberdir.

5.2.5.5.Doğru araç ve teknoloji seçimi

Çevik İlkeler, geliştirme ekibinin hızlı yanıt verme kabiliyetine sahip bir yazılım geliştirmek için doğru teknoloji ve tasarıma sahip olmaya odaklanmaları gerektiğini belirtmektedir (Wan ve Wang, 2010). Doğru araç ve teknoloji seçimi, çeşitli çalışmalarda çevik başarı faktörü olarak tanımlanmıştır (Chow ve Cao, 2008; Stankovic ve diğ, 2013). Yazılım ekibinin, yinelenmeli geliştirmeyi kolaylaştıran ve yeniden düzenleme faaliyetlerini destekleyen süreçler için uygun teknolojileri ve araçları kullanması önemlidir (Chow ve Cao, 2008).

Bir yazılım projesinde kullanılacak araçlar ve teknolojilerin seçimi, projenin özellikleri, boyutu, karmaşıklığı ve hedefleri gibi birçok faktörü dikkate almalıdır. Bu seçim sürecinde, projenin gereksinimlerini karşılamak için en uygun olan araç ve teknolojilerin belirlenmesi gerekmektedir. Bu doğru seçim, projenin başarılı bir şekilde tamamlanmasına ve hedeflerine ulaşmasına yardımcı olacaktır. Kullanılan araç ve teknolojinin doğru seçilmediği takdirde proje gecikmelerle karşılaşabilir, verimlilik düşebilir ve maliyetler artabilir. Bu nedenle projenin gereksinimlerine uygun araçlar ve teknolojilerin seçilmesi, başarı için önemlidir.

Doğru araç ve teknolojilerin seçimi, projenin başından sonuna kadar sürekli olarak gözden geçirilmeli ve gerektiğinde güncellenmelidir. Teknolojik gelişmeler ve pazar koşullarındaki değişiklikler göz önünde bulundurularak, projenin ihtiyaçlarına en uygun olan teknolojilerin belirlenmesi ve kullanılması önemlidir.

5.2.5.6.Etkin risk analizi yönetimi

Risk analizi yoluyla risklerin değerlendirilmesi, bunların gerçekleşme sıklığının ve potansiyel kayıpların büyüklüğünün ölçülmesini içerir (Ahmad ve diğ, 2018). Risk yönetimi, “bir kuruluşu risk açısından yönlendirmek ve kontrol etmek için koordineli faaliyetler” (ISO 31000, 2018) ve risk analizi değerlendirmelerini içerir. Risk yönetimi, belirsizliği azaltır ve yazılım geliştirme projesinin başarı olasılığını artırır (Tavares ve diğ, 2019). Risklerin değerlendirilmesi, gerçekleşme olasılıklarının ölçülmesi ve proje üzerindeki potansiyel etkilerinin belirlenmesi önemlidir (Huang ve Han, 2008). Çevik metodolojilerde sprint kullanımı risk yönetimini destekler çünkü sprintler düzenli denetim ve adaptasyon yoluyla öngörülebilirliği artırarak ürün hedefine doğru ilerlemeyi sağlar (The Scrum Guide, 2020).

Bir yazılım projesinde risk analizi ve yönetimi süreci, projenin başından sonuna kadar sürekli olarak gözden geçirilmelidir. Proje ekibi, olası riskleri tanımlamak ve bunların potansiyel etkilerini değerlendirmek için düzenli olarak toplanmalı ve analiz etmelidir. Bu analizler, projenin karşılaşılabileceği risklerin önceliklendirilmesine ve uygun önlemlerin belirlenmesine yardımcı olur.

Risklerin etkili bir şekilde yönetilmesi, proje sürecinin başarıyla tamamlanmasını yardımcı olur. Bu yüzden tüm ekip, risk analizi ve yönetimi sürecine gereken önemi vermeli ve bu süreci sürekli olarak izlemelidir. Risklerin belirlenmesi, değerlendirilmesi ve yönetilmesi, projenin hedeflerine ulaşmasını sağlayacak kararların alınmasına destek olur. Akademik çalışmalar da proje risk analizinin çevik yazılım geliştirme projeleri için bir başarı faktörü olduğunu vurgulamaktadır (Chow ve Cao, 2008; Stankovic ve diğ, 2013).

5.2.5.7.Sadelik

Sadeleştirme, çevik kritik başarı faktörlerinden biridir ve bir yazılım projesinin başarılı bir şekilde tamamlanmasında önemli bir rol oynar. Bu faktör, projenin karmaşıklığını azaltmak ve gereksiz karmaşıklıklardan kaçınmak anlamına gelir. Çevik Manifesto'nun onuncu ilkesi, "Sadelik, yapılmasına gerek olmayan işlerin mümkün olduğunca artırılması sanatı, olmazsa olmazlardandır" şeklindedir (Agile Manifesto, 2001b). Sadelik, tasarım ve kodun mümkün olduğunca basit tutulması anlamına gelir. Sade bir tasarımda, yinelenen kodlardan kaçınılır (Beck, 1999) ve yazılım geliştirme ekibi üyeleri, anlaşılması kolay ve gereksinimleri karşılayan bir yazılım ürünü geliştirmeye çalışır (Rashid ve Khan, 2018). Sadelik örnekleri arasında kapsamlı dokümantasyon yerine çalışan yazılım (Agile Manifesto, 2001a), kullanıcı hikayelerinde Gherkin dilinin kullanılması (çevik uygulama), ürün iş listesi iyileştirmesi (çevik uygulama) ve yeniden düzenleme ve eşli programlama yoluyla kod ve tasarımın basit tutulması (çevik uygulama) yer alır.

Karmaşıklığı azaltmak için proje ekibi, gereksiz detaylardan kaçınmalı ve iş süreçlerini mümkün olduğunca basitleştirmelidir. Böylelikle proje sürecinin daha anlaşılır ve yönetilebilir olması sağlanır. Ayrıca sadeleştirme, projenin hedeflerine odaklanmayı kolaylaştırır ve gereksiz iş yükünden kaçınılmasına destek olur. Sadeleştirme süreci, proje ekibinin katılımını gerektirir ve herkesin proje sürecine katkıda bulunmasını sağlar. Sonuç olarak sadelik/basitlik, çevik bir yazılım projesinin

başarılı bir şekilde tamamlanmasında önemli bir faktördür. Sade bir tasarımın takip edilmesinin çevik projenin başarı olasılığını artıracak literatür tarafından da doğrulanmıştır (Chow ve Cao, 2008; Stankovic ve diğ, 2013; Rashid ve Khan, 2018).

Teknik faktörlere ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 13: Teknik faktörler süreç verimliliğini olumlu yönde etkiler.
- Hipotez 14: Teknik faktörler sürdürülebilir yazılım ürün kalitesini olumlu yönde etkiler.
- Hipotez 15: Teknik faktörler paydaş memnuniyetini olumlu yönde etkiler.

5.2.6. Proje faktörleri

Risk faktörü, bir yazılım geliştirme projesinin başarılı bir şekilde tamamlanmasını olumsuz etkileyebilecek bir durum olarak tanımlanmaktadır (Schmidt ve diğ, 2001). Otomasyon testleri, kısa iterasyonlar, sürekli entegrasyonun benimsenmesi, özellik güdümlü geliştirmeye öncelik verilmesi ve diğer çeşitli prosedürel yaklaşımlar yoluyla çevik metodolojiler yazılım geliştirme risklerini azaltmaya çalışmaktadır. Bununla birlikte projenin başarısız olması veya proje bütçesinin aşılma olasılığı, süregelen bir endişe kaynağı olmaya devam etmektedir (Lunesu ve diğ, 2021). Bu çalışmada risk faktörleri ve başarısızlık faktörleri, Ahimbisibwe ve diğ. (2015) çalışmasına uygun olarak proje faktörleri olarak ifade edilmektedir. Çizelge 5.3'te gösterildiği gibi, literatür ve çevik uygulayıcılarla yapılan görüşmeler aracılığıyla altı proje faktörü belirlenmiştir: proje kritikliği, aciliyet, teknik karmaşıklık, teknolojik belirsizlik, kapsam değişikliği ve büyük proje boyutu.

5.2.6.1. Proje kritikliği

Projenin kritikliği, birçok karar için önemli bir itici güçtür. Projelerin kritikliği, genellikle projenin önem derecesi ve sonuçlarının organizasyon veya paydaşlar için ne kadar hayati olduğuyla ilgilidir. Ancak bazı durumlarda aşırı kritiklik projenin başarısını tehlikeye sokabilir ve proje ekibini aşırı baskı altına alabilir. Daha kritik bir proje için uygun teknikler, araçlar ve yöntemler kullanarak riski azaltmak daha önemlidir (Koch, 2004). Çeşitli çalışmalarda proje kritikliği proje başarısını etkileyen bir faktör olarak tanımlanmıştır (Howell ve diğ, 2010; Ahimbisibwe ve diğ, 2015; Garousi ve diğ, 2019). Test odaklı geliştirme, doğru miktarda dokümantasyon, eşli programlama, yeniden düzenleme faaliyetleri, uygun teknik eğitim, kodlama

standartlarına uyma ve müşteri ile ekip üyeleri arasında iyi iletişim, proje kritikliği ile ilişkili riskleri azaltmak için önemli faaliyetlerdir.

Cockburn (2007) kritikliği, işletmedeki kusurlar nedeniyle kaybedilen para miktarına göre dört gruba ayırmaktadır. Rahatsızlık kaybı, kusurun sadece küçük bir etkisi olduğu durumdaki kaybı ifade eder. Örneğin, bir restaurant sipariş edilen pizza yerine makarna servis edebilir. Bu durumda meydana gelen kusurun işletmeye maddi anlamda küçük bir etkisi bulunmaktadır. İkincil para kaybı, bir kusurun yedek yöntemler aracılığıyla kolayca düzeltilebileceği durumlarda meydana gelen kaybı ifade eder. Örneğin, bir şirket faturalama hatası yaparsa, müşteri arayarak faturayı değiştirebilir. Temel para kaybı, şirketin yazılım programındaki bir kusur nedeniyle iflas edebileceği durum işletmeye büyük miktarda maddi bir kayba neden olur. Hayati kayıp ise, kusurların felakete sebep olabileceği durumları ifade eder. Kalp pili cihazının çalışmaması, nükleer santral merkezinde yazılımsal hataların hayati kayıplara sebep olması örnek olarak verilebilir.

Proje kritikliğinin olumsuz etkilerinden kaçınmak için geliştirme ekibi, scrum master ve ürün sahibi, kritikliği belirleyerek ve yöneterek proje sürecini daha dengeli bir şekilde ilerletmelidir. Proje hedeflerinin daha gerçekçi bir şekilde belirlenmesi, kaynakların daha dengeli bir şekilde dağıtılması, projenin kritikliğini azaltacak önlemlerden bir kaçıdır. Tüm ekip proje kritikliğinin farkında olmalı ve gerektiğinde esneklik göstererek uygun çözümler üretmelidir. Aşırı kritiklik ekip üzerinde baskıya sebep olabilir. Bu da proje başarısını olumsuz yönde etkileyebilir. Bu yüzden üst yönetimin desteği ile projenin daha dengeli bir şekilde ilerlemesini sağlayacak stratejiler belirlenerek proje geliştirmesi yapılmalıdır.

5.2.6.2.Aciliyet

Aciliyet, proje faaliyetlerinde ve karar alma sürecinde genellikle zaman kısıtı olarak karşımıza çıkmaktadır. Aciliyetin artması veya yanlış şekilde yönetilmesi proje başarısını olumsuz etkileyebilir. Uygun bir takvime sahip proje, kritik olsa bile ekip üyeleri üzerinde zaman baskısı yaratmaz. Ancak gereksiz bir şekilde yüksek aciliyet belirtilmişse veya aciliyet gereğinden fazla vurgulanmışsa, proje ekibi üzerinde gereksiz bir stres yaratabilir. Bu da verimliliği düşürebilir ve yanlış kararlar alınmasına neden olur. Ayrıca aciliyetin doğru bir şekilde belirlenmemesi durumunda projenin

hedeflerine ulaşma sürecini olumsuz etkileyebilir ve projenin başarısını tehlikeye sokabilir.

Proje yönetiminde aciliyetin önemi ve doğru yönetimi, proje planlamasından uygulamaya kadar birçok aşamada etkilidir. Özellikle aciliyetin belirlenmesi, önceliklendirilmesi ve izlenmesi, proje yöneticileri için kritik bir sorumluluktur. Doğru bir aciliyet yönetimi, projenin başarı şansını artırabilirken yanlış yönetim veya hatalı önceliklendirme, projenin aksamalarla karşılaşmasına ve sonuçta başarısız olmasına yol açabilir.

Aciliyet, birçok çalışmada proje başarısını olumsuz etkileyen bir faktör olarak bahsedilmiştir (Howell ve diğ, 2010; Ahimbisibwe ve diğ, 2015; Garousi ve diğ, 2019). Ayrıca grup toplantısında tüm uygulayıcılar, aciliyetin bir karar verirken veya bir sorunu ele alırken ekip üyeleri üzerinde baskı oluşturduğunu belirtmiştir. Çevik etkinliklerin doğru şekilde uygulanması, güçlü müşteri desteği ve ekip üyeleri arasında güçlü iletişim ve koordinasyon, aciliyetin olası risklerini azaltmaya yardımcı olabilir.

5.2.6.3. Teknik karmaşıklık

Teknik karmaşıklık, geliştirilen bileşenlerin sayısı, yeni teknoloji kullanımı ve entegrasyonun zorluğu ile ilişkilidir. Bilgi teknolojisi sistemlerinin karmaşıklığı, proje başarısını ve ürünlerin kalitesini etkiler (Garousi ve diğ, 2019; Kanjanda ve Tuan, 2020). Bir projenin başarılı bir şekilde tamamlanması, karmaşıklığının yeterince değerlendirilmemesi ve ele alınmaması durumunda etkilenebilir (Mata ve diğ, 2023). Teknik karmaşıklık bilgili ekip üyeleri gerektirir ve teknik karmaşıklık seviyesi arttıkça ekip üyelerinin proje üzerindeki kontrol derecesi azalabilir. Bu nedenle gelişen sistemlerin sergilediği artan karmaşıklık ve birbirine bağımlılık seviyeleri ile çevik metodolojiler daha kullanışlı hale gelmektedir (Dorton ve diğ, 2020).

Teknik karmaşıklığın artması veya yanlış şekilde yönetilmesi proje başarısını olumsuz etkileyebilir. Gereksiz veya aşırı teknik karmaşıklığın projeye dahil edilmesi, proje ekibinin verimliliğini düşürebilir ve projenin ilerlemesini engelleyebilir. Ayrıca teknik karmaşıklığın doğru bir şekilde yönetilmemesi durumunda proje maliyetlerini artırabilir ve zaman çizelgesini bozabilir. Teknik karmaşıklık çevik kritik başarı faktörleri arasında önemli bir yere sahiptir. Fakat yanlış yönetilmesi veya gereksiz şekilde vurgulanması, projenin başarısını olumsuz etkileyebilir. Eşli programlama, yeniden düzenleme faaliyetleri, programlama standartlarını takip etme ve çeviklikle

etkinlikleri düzgün bir şekilde uygulama gibi bazı çevik uygulamalar teknik karmaşıklıkla başa çıkmak için yararlı olabilir.

5.2.6.4.Teknolojik belirsizlik

Teknolojik belirsizlik, yazılım geliştirme süreçleri için önemli bir risk faktörüdür. Bunun nedeni genellikle ekip üyelerinin deneyim ve teknik bilgi eksikliğidir. Ahimbisibwe ve diğ. (2015) teknolojik belirsizliğin kritik başarı faktörleri ile yazılım projesi başarı ölçütleri arasındaki ilişkinin gücünü ve/veya yönünü etkilediğini belirtmişlerdir.

Teknolojik belirsizlik, genellikle yeni veya karmaşık teknolojilerin kullanılması, gereksinimlerin belirsizliği veya teknik engellerin karşılanması gibi durumlarda ortaya çıkabilir. Bu tür belirsizlikler, proje planlamasını, kaynak tahsisini ve ilerlemeyi olumsuz yönde etkileyebilir, sonuç olarak proje başarısını tehlikeye atabilir. Doğru stratejilerin kullanılması ve etkin bir proje yönetimi ile bu belirsizliklerin etkileri minimize edilebilir ve proje başarılı bir şekilde tamamlanabilir. Teknik eğitimler, daha deneyimli ekip üyeleri ve yeniden düzenleme aktiviteleri, teknolojik belirsizlik risklerini yönetmek için etkili faaliyetler arasındadır.

5.2.6.5.Kapsam değişikliği

Kapsam kayması, yazılım geliştirme projelerinin başarısızlığında önemli bir faktör olarak öne çıkmaktadır. Bir projedeki özellikle de projenin kapsamına ilişkin değişim unsurlarının etkili bir şekilde ele alınması ve denetlenmesi, proje başarısı için temel bir faktör olarak ortaya çıkmaktadır (Aizaz ve diğ, 2021). Çevik metodolojiler gereksinimlerdeki değişiklikleri kabul etse de değişiklikler proje kapsamında, zamanında ve bütçesinde sapmalara neden olabilir ve yazılım proje yönetimi (Anthony, 2019; Menezes ve diğ, 2019) ve proje başarısı (Ahimbisibwe ve diğ, 2015; Garousi ve diğ, 2019) için bir risk faktörü olarak kabul edilir.

Kapsam değişiklikleri, genellikle proje sürecinin ilerlemesi sırasında ortaya çıkan veya beklenmeyen gereksinim değişiklikleri veya önceliklerin yeniden değerlendirilmesi sonucunda ortaya çıkabilir. Bu tür değişiklikler, proje planlamasını, bütçe ve kaynak yönetimini zorlaştırabilir. Bu yüzden de proje başarısı olumsuz yönde etkilenebilir.

Proje yönetiminde, kapsam değişiklikleri ile başa çıkmak için çeşitli stratejiler kullanılabilir. Bunlar arasında, kapsamdaki sık değişikliklerden kaynaklanan

potansiyel riskleri azaltmak için yazılım geliştirme ekibi proje ilerlemesini izlemek için test güdümlü geliştirme ve yeniden düzenleme faaliyetleri uygulayabilir.

5.2.6.6.Büyük proje boyutu

Çevik bir ekip tipik olarak en fazla 10 üyeden oluşur. Büyük bir ekip, farklı becerilere ve rollerdeki birçok üyeyi içerir, bu da iletişimi ve koordinasyonu zorlaştırabilir. Araştırmalar, özellikle dağıtık çevik projeler için ekip büyüklüğünün artmasının proje başarısı için bir risk faktörü olduğunu göstermiştir (Shrivastava ve Rathod, 2015, 2017). Çevik metodolojiler ekip üyeleri arasında güçlü bir iletişim gerektirir. Büyük projelerde karar alma süreçleri daha karmaşık hale gelebilir ve herkesin katılımını sağlamak daha zor olabilir. Bu durum, proje ilerledikçe hızlı kararlar alınmasını ve esneklik sağlanmasını engelleyebilir. Ekip üyesi sayısı arttıkça iletişim ve koordinasyon sorunları ortaya çıkabilir. Ayrıca ekip üyeleri arasında örtülü bilgi aktarımı zorlaşır.

Büyük proje boyutunun olumsuz etkilerini en aza indirmek için projeyi küçük parçalara bölerek ve küçük ekipler oluşturarak işbirliğini ve iletişimi kolaylaştırmak, proje ekibinin rolleri ve sorumlulukları netleştirmek, karar alma süreçlerini hızlandırmak ve proje yönetimine katılımı artırmak faydalı olacaktır.

Proje faktörüne ait hipotezler aşağıdaki gibi verilmiştir:

- Hipotez 16: Proje faktörleri süreç verimliliğini olumsuz etkiler.
- Hipotez 17: Proje faktörleri sürdürülebilir yazılım ürün kalitesini olumsuz etkiler.
- Hipotez 18: Proje faktörleri paydaş memnuniyetini olumsuz etkiler.

6. ÇEVİK PROJE BAŞARI MODELİNİN ANALİZİ

Şekil 5.1’de gösterilen Çevik Yazılım Projesi Başarı Modelini test etmek için bir anket formu hazırlanmıştır. Modeldeki her bir kavram, literatürden referans alınarak geliştirilmiştir (ISO, 2011; Stankovic ve diğ, 2013; Ahimbisibwe ve diğ, 2017; Garousi ve diğ, 2019; Hughes ve diğ, 2020). Ankette, çevik KBF’leri ve çevik proje başarı ölçütleri için toplam 48 soru hazırlanmış ve tüm sorular Ek B’de verilmiştir. Nihai anket iki bölümden oluşmaktadır. İlk bölümde katılımcıların demografik özelliklerine ilişkin sorular yer almaktadır (örneğin, kullanılan çevik metodolojiler, sektör, çalışma modeli (yerinde, uzaktan veya hibrit), çalışılan ülke ve projedeki iş sorumluluğu (örneğin, scrum master, ürün sahibi)). İkinci bölümde, katılımcıların KBF’ler ve yazılım projesi başarı ölçütleri hakkındaki algıları, 1 kesinlikle katılmıyorum ile 7 kesinlikle katılıyorum arasında değişen 7’li Likert ölçeği ile ölçülmüştür.

Mevcut çalışmanın hedef kitlesi çevik uygulayıcıları kapsamaktadır. Bu çalışmayı başlatmak için İstanbul Teknik Üniversitesi Sosyal ve Beşeri Bilimler İnsan Araştırmaları Etik Kurulu’ndan etik onay alınmıştır (proje numarası: 262, tarih: 2022.06.16). Geniş ölçekli anketin uygulanmasından önce ankete nihai halini vermek için 20 çevik uygulayıcı ile bir pilot test gerçekleştirilmiştir. Anketin son hali oluşturulduktan sonra toplam 596 çevik uygulayıcıdan anket verisi toplanmıştır.

Demografik profillere göre katılımcıların %73.32 gibi büyük bir kısmı Türkiye’de çalışmaktadır. Bankacılık ve borsa sektöründe çalışan katılımcılar, katılımcıların neredeyse yarısını oluşturmaktadır. Ayrıca katılımcıların büyük bir kısmı (%68.12) çevik bir metodoloji olarak scrum kullanmaktadır. Demografik profillerin detayları Çizelge 6.1’de verilmiştir.

Toplanan verilerin analizinde ilk olarak modelin faktör yapısını ortaya çıkarmak için açıklayıcı faktör analizi yapılmıştır. Ardından iki aşamalı bir prosedürle yapısal eşitlik modelleme yaklaşımı uygulanmıştır. Modeli doğrulamak için doğrulayıcı faktör analizi yapılmış ve Çevik Yazılım Projesi Başarı Modeli’nde tanımlanan ilişkiler yapısal model ile test edilmiştir.

Çizelge 6.1 : Katılımcıların demografik profili.

Ülke (%)	Türkiye 73.32	İngiltere 13.76	Almanya 6.21	Diğerleri 6.71		
Endüstri (%)	Bankacılık 30.70	Borsa 18.96	Yazılım 15.44	Pazarlama 7.05	Üretim 4.87	Diğer 22.98
Ünvan (%)	Yazılımcı 37.08	İş Analisti 17.79	Scrum Master 17.62	Proje Müdürü 11.74	Ürün Sahibi 8.72	Test Uzmanı 7.05
Tecrübe (yıl) (%)	0 – 2 12.92	2 – 5 32.55	5 – 10 38.09	10 -15 14.77	15 yıl ve üstü 1.68	
Çalışma modeli (%)	Hibrit 48.83	Ofis 28.52	Uzaktan 22.65			
Çevik Metodolojiler (%)	Scrum 68.12	Kanban 29.36	XP 2.01	Lean 0.50		

6.1.Açıklayıcı Faktör Analizi

Faktör yapısını belirlemek için Principal component analizi ve Varimax rotasyonu ile bir açıklayıcı faktör analizi (AFA) gerçekleştirilmiştir. Verileri analiz etmek için IBM SPSS Statistics yazılımı sürüm 28 kullanılmıştır. İlk olarak verilerin AFA için uygun olup olmadığına karar vermek için Kaiser-Meyer-Olkin (KMO) testi yapılmıştır. KMO değeri 0.91 olarak bulunmuştur ve eşik değer 0.90'dan büyük olduğu doğrulanmıştır (Kaiser ve Rice, 1974). Ayrıca Bartlett testi sonucu anlamlıdır ($p < .001$) ve yeterli korelasyonları göstermektedir (Hair ve diğ., 2010). AFA sonuçları, özdeğerleri 1'den büyük olan toplam dokuz faktörün ortaya çıktığını ve toplam varyans açıklamasının %63.27 olduğunu göstermiştir. Faktör sayısı Çevik Yazılım Projesi Başarı Modelindeki faktör sayısı ile aynıdır. Standartlaştırılmış öge yükleri 0.60'ın altında olan üç öge bulunmuştur. Bunlar; iyi tanımlanmış vizyon ve misyon (ORG4), proje yöneticisinin bilgisi (TEAM4), proje yöneticisinin tecrübesi (TEAM5). Geleneksel proje yönetiminde, proje yöneticileri önemli yetki ve sorumluluğa sahiptir. Proje ekibi içinde merkezi karar verme noktası olarak görev vermektedirler. Ancak çevik yazılım geliştirmede dinamikler farklılaşmaktadır. Çevik metodolojilerde proje yönetiminde merkezi olmayan bir yaklaşımın varlığı kabul edillir. Kendi kendine organize olabilen ekip üyeleri bulunmaktadır. Ekip üyeleri arasında işbirliği ve değişime yanıt verebilirliğin önemi çevik ilkelerde bahsedilmektedir. Sonuç olarak, çevik ortamlarda geleneksel olarak proje yöneticileriyle ilişkilendirilen hiyerarşik etki

azalmaktadır. Bu bağlamda ‘proje yöneticisinin bilgisi’ ve ‘proje yöneticisinin deneyimini’, Çevik Yazılım Projesi Başarı Modeli’ndeki ekip kavramından çıkarılması stratejik bir karardır. Proje yöneticileri şüphesiz genel proje başarısına katkıda bulunurken etkileri geleneksel proje ortamlarına kıyasla çevik ekiplerde doğrudan veya belirgin olmayabilir. Ankete katılan çevik uygulayıcıların verilerine göre yapılan AFA çalışmasında TEAM4 ve TEAM5 listeden çıkarılmıştır. ‘İyi tanımlanmış vizyon ve misyon’, daha çok yönetsel seviyede karar verilen ve oluşturulan stratejik bir karardır. Ankete üst yönetim kadrosundan katılan katılımcı sayısı az olduğu için bu yönde bir sonuç çıktığı değerlendirilmektedir. Benzer şekilde ankete katılan çevik uygulayıcıların verilerine göre yapılan AFA çalışmasında ORG4 listeden çıkarılmıştır.

Öge yükleri 0.60’ın altında olan ORG4, TEAM4 ve TEAM5, öge listesinden çıkartılıp tekrardan AFA testi yapılmıştır. İkinci kez yapılan AFA sonuçları, özdeğerleri 1’den büyük olan toplam dokuz faktörün ortaya çıktığını ve toplam varyans açıklamasının %66.14 olduğunu göstermiştir. Faktör sayısı Çevik Yazılım Projesi Başarı Modelindeki faktör sayısı ile aynıdır. ORG4, TEAM4 ve TEAM5 çıkarıldıktan sonra yapılan AFA testinde tüm öğelerin yüklenmesi 0.60’ın üzerinde bulunmuştur. Ayrıca tüm Cronbach alfa değerleri eşik değer olan 0.70’in üzerinde olup yeterli güvenilirliğe işaret etmektedir (Fornell ve Larcker, 1981). Nihai öge listesi ve her bir öğeye karşılık gelen kavramlar Çizelge 6.2’de gösterilmektedir.

6.2.Doğrulayıcı Faktör Analizi

Doğrulayıcı faktör analiz (DFA), 45 öge ile tanımlanan 9 kavramın güvenilirliğini ve geçerliliğini incelemek için yapılmıştır. Analiz için IBM AMOS 20.0 kullanılmıştır. DFA’nın ilk analizi, Çizelge 6.3’de gösterildiği gibi model uyum değerlerinin kabul edilebilir değerlere ulaştığını ortaya koymuştur. χ^2 değerinin serbestlik derecesine bölünmesi, kök-ortalama-yaklaşım karesi (RMSEA), uyum iyiliği indeksi (GFI), düzeltilmiş uyum iyiliği indeksi (AGFI), normlaştırılmış uyum indeksi (NFI) ve karşılaştırmalı uyum indeksi (CFI), Hair ve arkadaşlarının önerilerine göre incelenmiştir (Hair ve diğ, 1998). Tüm değerler kabul edilebilir seviyededir (Joreskog ve Sorbom, 1984; Marsh ve Hocevar, 1985; Bollen, 1989; Bentler, 1990; Browne ve Cudeck, 1992; Forza ve Filippini, 1998; Gefen ve diğ, 2000), bu da modelin verilere iyi uyum gösterdiğini ortaya koymaktadır.

Ayrıca yapıların yakınsak ve ayrıçsak geçerliliği de incelenmiştir. Yakınsak geçerlilik, Çizelge 6.4 ve Çizelge 6.5’de gösterildiği gibi standartlaştırılmış faktör yükleri, bileşik güvenilirlik (CR), t-değerleri ve ortalama çıkarılan varyans (AVE) ile test edilmiştir. Tüm CR değerleri 0.70 eşiğinden büyüktür ve bu da modeldeki iç tutarlılığın yeterli olduğunu göstermektedir (Straub, 1989). 0.95 güven düzeyinde, tüm öge t-değerleri kritik değer olan 1.96’dan büyüktür (Bagozzi ve Phillips, 1991). Ayrıca AVE değerleri 0.50 eşik değerini aşmıştır (Henseler ve diğ, 2009; Hair ve diğ, 2016). Böylelikle yakınsak geçerlilik doğrulanmıştır.

Çizelge 6.2 : Standartlaştırılmış öge yükleri, açıklanan varyans, ve Cronbach’s alpha değerleri.

Kavram	Öge	Standartlaştırılmış Öge Yükleri	Açıklanan Varyans (%)	Cronbach α
Müşteri	CUST1	0.78	6.08	0.83
	CUST2	0.77		
	CUST3	0.78		
	CUST4	0.80		
Ekip	TEAM1	0.82	7.01	0.85
	TEAM2	0.77		
	TEAM3	0.77		
	TEAM6	0.73		
	TEAM7	0.74		
Teknik	TECH1	0.73	9.18	0.88
	TECH2	0.75		
	TECH3	0.73		
	TECH4	0.77		
	TECH5	0.78		
	TECH6	0.79		
	TECH7	0.76		
Çevik süreç	AGPF1	0.74	5.93	0.82
	AGPF2	0.77		
	AGPF3	0.76		
	AGPF4	0.78		
Organizasyonel	ORGN1	0.79	6.04	0.82
	ORGN2	0.76		
	ORGN3	0.81		
	ORGN5	0.80		
Proje	PRJF1	0.75	8.57	0.87
	PRJF2	0.79		
	PRJF3	0.75		
	PRJF4	0.76		
	PRJF5	0.74		
	PRJF6	0.76		

Çizelge 6.2. (devam) : Standartlaştırılmış öge yükleri, açıklanan varyans, ve Cronbach's alpha değerleri.

Kavram	Öge	Standartlaştırılmış Öge Yükleri	Açıklanan Varyans (%)	Cronbach α
Süreç verimliliği	EFFC1	0.77	4.76	0.84
	EFFC2	0.76		
	EFFC3	0.75		
Sürdürülebilir yazılım ürün kalitesi	QUAL1	0.78	13.83	0.93
	QUAL2	0.78		
	QUAL3	0.75		
	QUAL4	0.81		
	QUAL5	0.78		
	QUAL6	0.79		
	QUAL7	0.78		
	QUAL8	0.80		
	QUAL9	0.75		
Müşteri memnuniyeti	SATF1	0.73	4.73	0.83
	SATF2	0.79		
	SATF3	0.74		

Çizelge 6.3 : Doğrulayıcı faktör analizi model uygunluk istatistikleri.

Uygunluk İndeksi	Önerilen değer	Gözlemlenen değer	Model uyumu	Referanslar
χ^2 / df	≤ 3	1.84	Uyumlu	(Joreskog ve Sorbom, 1984)
RMSEA	≤ 0.08	0.04	Uyumlu	(Bollen, 1989)
GFI	≥ 0.80	0.88	Uyumlu	(Forza ve Filippini, 1998)
AGFI	≥ 0.80	0.86	Uyumlu	(Bentler, 1990; Forza ve Filippini, 1998; Gefen ve diğ., 2000)
CFI	≥ 0.90	0.94	Uyumlu	(Hair ve diğ., 1998)
NFI	≥ 0.80	0.88	Uyumlu	(Browne ve Cudeck, 1992)
χ^2 / df	≤ 3	1.84	Uyumlu	(Joreskog ve Sorbom, 1984)

Çizelge 6.4 : Yakınsak geçerlik sonuçları.

Kavram	Öge	CFA loadings	t-değeri	Kavram	Öge	CFA loadings	t-değeri	Kavram	Öge	CFA loadings	t-değeri
Müşteri	CUST1	0.76	17.07	Çevik süreç	AGPF1	0.76	16.16	Süreç verimliliği	EFFC1	0.79	19.26
	CUST2	0.78	17.55		AGPF2	0.74	16.57		EFFC2	0.79	19.32
	CUST3	0.75	17.07		AGPF3	0.72	16.28		EFFC3	0.81	19.32
	CUST4	0.67	15.24		AGPF4	0.72	16.16				
Ekip	TEAM1	0.64	15.35	Organizasyonel	ORGN1	0.75	15.81	Müşteri memnuniyeti	SATF1	0.81	17.95
	TEAM2	0.64	15.25		ORGN2	0.71	15.06		SATF2	0.74	17.95
	TEAM3	0.75	17.98		ORGN3	0.73	15.06		SATF3	0.80	17.78
	TEAM6	0.79	15.35		ORGN5	0.74	15.70				
	TEAM7	0.79	18.97								
Teknik	TECH1	0.69	15.22	Proje	PRJF1	0.74	16.99	Sürdürülebilir yazılım ürün kalitesi	QUAL1	0.77	20.33
	TECH2	0.69	15.22		PRJF2	0.74	16.99		QUAL2	0.79	20.84
	TECH3	0.67	14.79		PRJF3	0.71	16.46		QUAL3	0.77	20.27
	TECH4	0.74	16.04		PRJF4	0.72	16.52		QUAL4	0.79	20.86
	TECH5	0.75	16.27		PRJF5	0.69	15.85		QUAL5	0.78	19.42
	TECH6	0.77	16.67		PRJF6	0.74	16.99		QUAL6	0.80	21.17
	TECH7	0.71	15.50						QUAL7	0.79	21.09
									QUAL8	0.80	21.29
									QUAL9	0.74	19.42

Çizelge 6.5 : Ayrıştırıcı geçerlik sonuçları.

Kavram	AVE	CR	Yazılım ürün kalitesi	Müşteri	Ekip	Organizasyonel	Çevik süreç	Teknik	Proje	Süreç verimliliği	Müşteri memnuniyeti
Sürdürülebilir yazılım ürün kalitesi	0.61	0.93	0.78								
Müşteri faktörleri	0.55	0.83	0.38	0.74							
Ekip faktörleri	0.52	0.84	0.25	0.23	0.72						
Organizasyonel faktörler	0.54	0.82	-0.11	0.04	-0.08	0.73					
Çevik süreç faktörleri	0.54	0.82	0.48	0.19	0.22	-0.13	0.73				
Teknik faktörler	0.51	0.88	0.1	-0.03	-0.02	0.04	-0.06	0.72			
Proje faktörleri	0.52	0.87	-0.22	-0.01	-0.09	0.3	-0.11	-0.08	0.72		
Süreç verimliliği	0.63	0.84	0.46	0.39	0.41	-0.03	0.46	0.2	-0.32	0.79	
Müşteri memnuniyeti	0.61	0.83	0.51	0.5	0.43	0.01	0.45	0.03	-0.21	0.53	0.78

6.3.Yapısal Model

Yapısal modelde, her bir çevik kritik başarı faktörünün (müşteri faktörleri, ekip faktörleri, organizasyonel faktörler, çevik proje faktörleri, teknik faktörler, proje faktörleri), her bir çevik proje başarı ölçütü (süreç verimliliği, sürdürülebilir yazılım ürün kalitesi, paydaş memnuniyeti) üzerindeki etkileri analiz edilecektir. İlk olarak, araştırma modelinin uyum istatistikleri test edilmiştir. Sonuçlar, tüm uyum indekslerinin, mutlak uyum indekslerinin (RMSEA = 0.04, GFI = 0.88 ve AGFI = 0.86), artımlı uyum indekslerinin (NFI = 0.88 ve CFI = 0.94) ve Ki-Kare/serbestlik derecesinin (1.85'te) önerilen değerler içinde olduğunu ve modelin verilere uyduğunu göstermiştir (Joreskog ve Sorbom, 1984; Marsh ve Hocevar, 1985; Bollen, 1989; Bentler, 1990; Browne ve Cudeck, 1992; Forza ve Filippini, 1998; Gefen ve diğ, 2000).

Sonuçlara göre, süreç verimliliği ($R^2 = 0.51$), sürdürülebilir yazılım ürün kalitesi ($R^2 = 0.39$) ve paydaş memnuniyetinde ($R^2 = 0.51$) önemli varyans sunmuştur. Toplam 18 hipotez test edilmiş ve iki hipotez reddedilmiştir. Çizelge 6.6'da, path katsayıları ve anlamlılık düzeyleri ile birlikte sonuçlar gösterilmektedir.

Sonuçlar, müşteri faktörleri, ekip faktörleri, çevik süreç faktörleri ve proje faktörlerinin süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti açısından çevik proje başarısının önemli belirleyicileri olduğunu göstermiştir. Çevik süreç faktörü; ekip, müşteri, organizasyonel, teknik ve proje faktörlerine kıyasla 0.41 katsayısı ($p < 0.01$) ile sürdürülebilir yazılım ürün kalitesi için en güçlü belirleyicidir. Bu sonuç ekip üyeleri arasında güçlü iletişimin varlığının, çeviklikle uygulamaların, çevik dostu ilerleme izleme/kontrol mekanizmasının ve doğru miktarda dokümantasyonun sürdürülebilir yazılım ürünü kalitesini artırdığını göstermektedir. Benzer şekilde, müşteri faktörü sırasıyla 0.38, 0.30, 0.27 ($p < 0.01$) katsayısı ile paydaş memnuniyeti, sürdürülebilir yazılım ürün kalitesi ve süreç verimliliği için güçlü bir belirleyicidir. Müşteri ilişkilerinin geliştirilmesi, müşteri deneyiminin iyileştirilmesi, müşteri desteğine öncelik verilmesi ve yeterli eğitimin sağlanması projenin başarıya ulaşmasına yardımcı olabilir.

Teknik faktörlerin paydaş memnuniyeti üzerindeki etkisi 0.05 katsayısı ile önemsiz bulunmuştur ($p > 0.05$). Teknik faktörler, kod kalitesini ve sürdürülebilirliği artırmayı amaçlasa da bunların paydaş memnuniyetiyle doğrudan ilişkisini kurmak zor olabilir.

Benzer şekilde organizasyonel faktörlerin sürdürülebilir yazılım ürün kalitesi üzerindeki etkisi -0.02 katsayısı ile önemsiz bulunmuştur ($p > 0.05$). Organizasyonel faktörler destekleyici bir kültür ve yapı oluşturulmasına yardımcı olurken yazılım ürünlerinin gerçek kalitesi kapsayıcı organizasyonel faktörlerden ziyade teknik yetkinliklere ve günlük operasyonel uygulamalara bağlı olabilir.

‘Müşteri faktörleri’, ‘ekip faktörleri’, ‘organizasyonel faktörler’, ‘çevik süreç faktörleri’, ‘teknik faktörler’ ve ‘proje faktörleri’ olmak üzere toplam 6 kavram, süreç verimliliğindeki toplam varyansın %51’ini açıklamaktadır. Ayrıca ‘müşteri faktörleri’, ‘ekip faktörleri’, ‘çevik süreç faktörleri’, ‘teknik faktörler’ ve ‘proje faktörleri’ sürdürülebilir yazılım ürün kalitesinin toplam varyansının %39’unu açıklamaktadır. Son olarak, ‘müşteri faktörleri’, ‘ekip faktörleri’, ‘organizasyonel faktörler’, ‘çevik süreç faktörleri’ ve ‘proje faktörleri’ olmak üzere toplam 5 kavram, paydaş memnuniyetinin toplam varyansının %51’ini açıklamaktadır.

Çizelge 6.6 : Yapısal model tahmini.

Kavram	Süreç verimliliği	Sürdürülebilir yazılım ürün kalitesi	Paydaş memnuniyeti
R^2 (%)	0.51	0.39	0.51
Müşteri faktörleri	0.27**	0.30**	0.38**
Ekip faktörleri	0.27**	0.09*	0.27**
Organizasyonel faktörler	0.10*	-0.02	0.10*
Çevik süreç faktörleri	0.36**	0.41**	0.33**
Teknik faktörler	0.21**	0.12**	0.05
Proje faktörleri	-0.27**	-0.16**	-0.19**

*: $p < 0.05$; **: $p < 0.01$.

6.4.Çevik Yazılım Projelerinde Sektörel ve İş Rolü Farklılıklarının Etkisi

Sektörel farklılıkların ve iş rollerinin çevik yazılım geliştirme proje başarısı üzerindeki etkisinin incelenmesi önemlidir. Bu yüzden MANOVA kullanılarak, bağımsız değişkenlerin, yani sektörel farklılıkların (finans ve diğer sektörler olarak kategorize edilmiştir) ve iş rollerinin paydaş memnuniyeti, sürdürülebilir yazılım ürün kalitesi ve süreç verimliliği üzerindeki kolektif ve bireysel etkileri analiz edilmiştir. Sektörel farklılıkları incelemek için yalnızca iki grup, finans sektörü ve diğerleri seçilmiştir. Ankete katılan çevik uygulayıcılarının yarısı finansla ilgili sektörden gelmektedir. Ayrıca finans sektörü katı standartlara bağlıdır ve daha sıkı düzenleyici kurumlar

tarafından denetlenmektedir. Bu yüzden sektörel farklılıklar için finans sektörü ve diğer sektörlerden oluşan karma bir grupla sektörel farklılıklara bakılması yönünde karar verilmiştir. Benzer şekilde bu çalışmada geliştirici, iş analisti, scrum master ve ürün sahibi vb belirli iş rollerinin seçimi, çevik proje başarısı için önemli olan temel işlevleri kapsayacak şekilde stratejik olarak seçilmiştir.

Çalışmada 3 bağımlı değişken bulunmaktadır. Bunlar paydaş memnuniyeti, sürdürülebilir yazılım ürün kalitesi ve süreç verimliliğidir. 2 bağımsız değişken bulunmaktadır. Bunlar finans sektörü ve diğerleri olarak adlandırılan sektörel farklılıklar ve iş rolleridir.

Çok değişkenli test sonuçları Çizelge 6.7’de gösterilmektedir. Genel modelin Pillai’s Trace değeri 0.974 ile yüksek bir değere sahiptir ve F değeri 7308.073 ($p = .000$) ile istatistiksel olarak anlamlıdır. Bu sonuçlar, genel modelin süreç verimliliği, ürün kalitesi ve paydaş memnuniyeti üzerinde anlamlı bir etkiye sahip olduğunu göstermektedir. Wilks’ Lambda değeri 0.924, F değeri 5.239 ($p = .000$) ile iş rolleri bağımlı değişkenler üzerinde anlamlı ancak daha düşük bir etkiye sahiptir. Bu sonuç, iş rollerinin projenin çeşitli yönleri üzerinde farklı derecelerde etkiler yaratabileceğini göstermektedir. Wilks’ Lambda değeri 0.982, F değeri 3.486 ($p = .016$) ile sektörel farklılıkların bağımlı değişkenler üzerinde daha düşük bir etkiye sahip olduğunu gösterir. Bu istatistiksel sonuç, sektör bazında değişikliklerin süreç verimliliği, ürün kalitesi ve paydaş memnuniyeti üzerinde belirgin ancak sınırlı etkilere sahip olduğuna işaret eder. İş rolleri ve sektörel farklılıkların etkileşimi, Pillai’s Trace değeri 0.036 ve F değeri 2.356 ($p = .012$) ile bu iki değişkenin kombinasyonunun bağımlı değişkenler üzerinde anlamlı etkiler yarattığını, ancak etkilerin nispeten düşük olduğunu gösterir. İş rolleri ve sektörel farklılıkların etkileşimi üzerine yapılan analiz sonucunda iş rolleri ve sektörel farklılıkların özel koşullarda etkileşiminin önemli sonuçlar doğurabileceğini belirtmektedir.

Farklı kategoriler arasında değişkenlerin etkilerini karşılaştırmak için bağımsız gruplar arası etki testleri sonuçları Çizelge 6.8’de gösterilmiştir. İş rollerinin süreç verimliliği üzerinde herhangi bir anlamlı etkisi olmamakla birlikte ($F = 0.564$, $p = .639$), paydaş memnuniyeti ve sürdürülebilir yazılım ürün kalitesi üzerinde istatistiksel olarak anlamlı bir etkisi vardır. Bu, farklı iş rollerinin paydaş memnuniyeti ve sürdürülebilir yazılım ürün kalitesi üzerinde önemli bir rol oynadığını, ancak süreç verimliliği üzerinde homojen bir etkiye sahip olduğunu gösterir.

Sektörel farklılıkların süreç verimliliği üzerinde anlamlı bir etkisi olduğu görülmektedir ($F = 4.960, p = .026$). Fakat sektörel farklılıklar, sürdürülebilir yazılım ürün kalitesi ($F = .016, p = .899$) ve paydaş memnuniyeti ($F = 1.602, p = .206$) üzerinde anlamlı bir etki göstermemiştir.

İş rolleri ve sektörel farklılıkların etkileşiminin, süreç verimliliği ($F = 3.050, p = .028$) ve paydaş memnuniyeti ($F = 3.845, p = .010$) üzerinde anlamlı bir etkisi bulunmaktadır. Bu, iki değişkenin etkileşiminin, süreç verimliliği ve paydaş memnuniyeti üzerinde belirli koşullar altında önemli etkilere sahip olabileceğini gösterir. Sürdürülebilir yazılım ürün kalitesi üzerinde ise bu etkileşim anlamlı bulunmamıştır ($F = 1.520, p = .208$).

Sonuç olarak, sektörel farklılıklar, yalnızca süreç verimliliği üzerinde önemli bir etki göstermiştir. Finans sektörünün özgün talepleri ve kısıtlamaları, süreç verimliliğini belirgin bir şekilde etkilemektedir. Bu durum, finans sektöründe kullanılan belirli uygulamaların, verimli süreçler elde etmede kullanılan metodolojiler üzerinde güçlü bir etkiye sahip olduğunu göstermektedir. İş rolleri, hem paydaş memnuniyeti hem de sürdürülebilir yazılım ürün kalitesi üzerinde önemli bir etkiye sahiptir. Farklı iş fonksiyonlarındaki kişiler, bu yapıları çeşitli şekillerde algılar ve etkiler. Paydaş memnuniyeti ve sürdürülebilir yazılım kalitesindeki değişikliklerin, her role özgü farklı bakış açıları ve beceri setlerinden kaynaklanabileceği düşünülmektedir.

Çizelge 6.7 : Çok değişkenli test sonuçları.

	Etki	Değer	F	Anlamlılık
Kesişim	Pillai's Trace	0.974	7308.073	0
	Wilks' Lambda	0.026	7308.073	0
	Hotelling's Trace	37.413	7308.073	0
	Roy's Largest Root	37.413	7308.073	0
İş rolleri	Pillai's Trace	0.077	5.15	0
	Wilks' Lambda	0.924	5.239	0
	Hotelling's Trace	0.082	5.301	0
	Roy's Largest Root	0.071	13.906	0
Sektör	Pillai's Trace	0.018	3.486	0.016
	Wilks' Lambda	0.982	3.486	0.016
	Hotelling's Trace	0.018	3.486	0.016
	Roy's Largest Root	0.018	3.486	0.016
İş rolleri*Sektör	Pillai's Trace	0.036	2.356	0.012
	Wilks' Lambda	0.965	2.365	0.012
	Hotelling's Trace	0.036	2.369	0.012
	Roy's Largest Root	0.028	5.529	0.001

Çizelge 6.8 : Bağımsız gruplar arası etki testleri.

Kaynak	Bağımlı değişken	F	Anlamlılı k	Kısmi ETA Karesi
Düzeltilmiş Model	Süreç verimliliği	2.087	0.043	0.024
	Paydaş memnuniyeti	3.267	0.002	0.037
	Yazılım kalitesi	3.552	0.001	0.041
Kesişim	Süreç verimliliği	11021.9	0	0.949
	Paydaş memnuniyeti	14955.4	0	0.962
	Yazılım kalitesi	13961.7	0	0.96
		2		
İş rolleri	Süreç verimliliği	0.564	0.639	0.003
	Paydaş memnuniyeti	3.896	0.009	0.019
	Yazılım kalitesi	6.734	0	0.033
Sektör	Süreç verimliliği	4.96	0.026	0.008
	Paydaş memnuniyeti	1.602	0.206	0.003
	Yazılım kalitesi	0.016	0.899	0
İş rolleri*Sektör	Süreç verimliliği	3.05	0.028	0.015
	Paydaş memnuniyeti	3.845	0.01	0.019
	Yazılım kalitesi	1.52	0.208	0.008

7. SONUÇLARIN TARTIŞILMASI

Bu çalışmanın amacı, çevik projelerin başarısı için gerekli kritik başarı faktörlerini belirlemek ve kritik başarı faktörleri ile yazılım projesi başarı ölçütleri arasındaki ilişkiyi anlamak için bir Çevik Yazılım Projesi Başarı Modeli geliştirmektir. Araştırma modelinde ilk adım olarak, 2000-2021 yılları arasındaki son 21 yılı kapsayan sistematik bir literatür taraması yapılmıştır. Anketin veri toplama aşaması yaklaşık bir yıl sürmüş ve bu süre zarfında sürekli olarak literatür taraması yapılmaya devam edilmiş, 2022 ve 2023 yıllarına ait yayınlar da çalışmaya dahil edilmiştir. Literatür taramasından sonra başarı modelinde yer alan faktörlerin geliştirilmesi için titizlikle analiz çalışmaları yapılmıştır. Çevik kritik başarı faktörleri ve çevik başarı ölçütleri belirlendikten sonra konusunda uzman altı uygulayıcı ile bire bir toplantılar yapılmış ve bu toplantılarda her bir çevik kritik başarı faktörü ve çevik başarı ölçütü detaylıca tartışılmıştır. Uzmanlardan alınan geri bildirimler doğrultusunda model üzerinde güncellemeler yapılmıştır. Daha sonra çevik yazılım projesi başarı modelinde fikir birliğine varmak için uzmanlarla toplu bir toplantı gerçekleştirilmiştir. Nihai Çevik Yazılım Projesi Başarı Modeli elde edildikten sonra anket uygulamasına geçilmiş ve 596 çevik uygulayıcıdan elde edilen veri seti ile araştırma modelinde tanımlanan ilişkiler açıklığa kavuşturulmuştur.

Sonuçlara göre çevik süreç faktörleri ve müşteri faktörleri, ekip, organizasyonel, teknik ve proje faktörleri gibi diğer faktörleri geride bırakarak sürdürülebilir yazılım ürün kalitesi için en güçlü belirleyici olarak bulunmuştur. Bu durum, etkin iletişim ve koordinasyonun, çevik etkinliklerin uygulanmasının, çevik odaklı ilerleme izleme ve kontrol mekanizmalarının ve uygun dokümantasyonun sürdürülebilir yazılım ürün kalitesinin artırılmasındaki önemini vurgulamaktadır. Çevik proje faktörlerinin aynı zamanda süreç verimliliği ve paydaş memnuniyeti için de önemli belirleyiciler olduğu görülmüştür. Chow ve Cao (2008) araştırması da çevik yazılım projelerinin başarıya ulaşmasında uygun dokümantasyonun, ilerleme izleme ve kontrol mekanizmalarının ve günlük iletişimin önemini desteklemektedir. Çevik uygulayıcılarla yapılan görüşmelerde, etkili iletişim ve koordinasyonun ekip üyeleri arasında uyumlu işbirliğini kolaylaştırmada çok önemli bir rol oynadığı vurgulanmıştır. Günlük stand-

up toplantıları ve sprint retrospektifleri gibi çevik etkinliklerin iteratif geliştirme için temel çerçeveler olarak önemini vurgulamışlardır. Bu etkinlikler disiplinler arası katılımı ve uyarlanabilir karar alma sürecini kolaylaştırmaktadır. Ayrıca çevik uygulayıcılar sürüm ve iterasyon çizelgeleri ve Kanban panoları gibi çevik odaklı ilerleme izleme ve kontrol mekanizmalarının önemli rolünün altını çizmiştir. Bu durum, proje ekiplerinin ilerlemeyi izlemelerini, darboğazları belirlemelerini ve başarılı sonuçlar için proje dinamiklerini proaktif olarak yönetmelerini sağlar.

Müşteri desteği, iyi müşteri ilişkileri, BT alanında müşteri eğitimi ve müşteri deneyimi gibi müşteri faktörleri, çevik yazılım geliştirme projesi başarı ölçütleri üzerinde olumlu bir etkiye sahiptir. Akademik çalışmalara göre müşteri desteği de çevik yazılım geliştirme projelerinin başarısına önemli bir katkıda bulunmaktadır (Chow ve Cao, 2008; Misra ve diğ, 2009; Ahimbisibwe ve diğ, 2015). Benzer şekilde, Jun ve diğ. (2011) müşteri eğitimi ve deneyiminin proje başarısında olumlu bir rol oynadığını belirtmiştir. Toplantılar sırasında çevik uygulayıcılar, güçlü müşteri ilişkilerinin güven ve etkili iletişimi teşvik etmesi nedeniyle yazılım yaşam döngüsü boyunca güçlü müşteri ilişkisinin önemini vurgulamışlardır. Ayrıca BT çözümlerinin başarılı bir şekilde benimsenmesini ve kullanılmasını garanti altına almak için iş süreçleri sahiplerine gerekli eğitimleri sağlamanın önemli olduğunu belirtmişlerdir.

Analizler sonucunda ekip faktörlerinin de süreç verimliliği, sürdürülebilir ürün kalitesi ve paydaş memnuniyeti için önemli belirleyiciler olduğu görülmüştür. Önceki çalışmalar da ekip faktörlerinin yazılım projelerinin başarısını belirlemede önemli bir rol oynadığını göstermiştir (Lindvall ve diğ, 2002; Turner ve Boehm, 2003; Chow ve Cao, 2008; Misra ve diğ, 2009; Wan ve Wang, 2010; Ahimbisibwe ve diğ, 2015; Garousi ve diğ, 2019). Özellikle ekip üyelerinin yetenek, deneyim ve becerilerinin yazılım projelerinin başarısı üzerinde olumlu bir etkiye sahip olduğu gösterilmiştir. Bu durum, başarılı çevik proje sonuçları elde etmek için yetkin ve yetenekli bir ekibe sahip olmanın önemini vurgulamaktadır.

Çevik çalışma ortamı, işbirlikçi yönetim tarzı, kurumsal olarak kabul gören değişim yönetimi ve üst yönetim desteği gibi örgütsel faktörlerin süreç verimliliği ve paydaş memnuniyeti üzerindeki olumlu etkisi nispeten daha düşüktür. Ayrıca organizasyonel faktörlerin sürdürülebilir yazılım ürün kalitesi üzerindeki etkisi anlamlı bulunmamıştır. Bu durum, organizasyonel faktörlerin önemli olmasına rağmen, yazılım ürün kalitesinin iyileştirilmesini belirgin bir şekilde garanti etmeyebileceğini

göstermektedir. Organizasyonel özellikler çevik uygulamalar için elverişli bir atmosfer sunmakta ve destekleyici bir kültür ve yapı oluşturulmasına yardımcı olmaktadır; yazılım ürünlerinin gerçek kalitesi ise teknik yetkinlikler, günlük operasyonel uygulamalar ve geliştirme ekiplerinin uyum kabiliyeti gibi diğer faktörlere bağlıdır. Benzer şekilde Chow ve Cao (2008), çevik yazılım projesi başarısı için organizasyonel kültürün etkisini destekleyen istatistiksel kanıt bulamamıştır. Covid-19 salgını farklı çalışma modellerini hayatımıza sokmuş, son dört yılda uzaktan ve hibrit çalışma modellerine hızlı bir adaptasyon yaşanmıştır. Sonuç olarak çalışma şekillerinin son yıllarda değişmesinden dolayı organizasyonel faktörlerin çevik proje başarısı üzerindeki etkisi son yıllarda değişmiş olabilir.

Mevcut çalışmanın bir diğer bulgusu da teknik faktörlerin süreç verimliliği ve sürdürülebilir yazılım ürün kalitesi üzerinde olumlu etkileri olmasıdır. Bu bulgu Chow ve Cao'nun bulgularıyla benzerlik göstermektedir. Chow ve Cao (2008) çalışmasında yazılım test sürecinin, programlama standartlarına bağlılığın, yeniden düzenleme faaliyetlerinin, basitliğin ve çalışan yazılımın düzenli olarak teslim edilmesinin çevik yazılım projelerinin başarısına ulaşmasındaki önemine vurgu yapmaktadır. Çevik uygulayıcılarla yapılan toplantılarda, yazılım test süreçlerinin önemi, yazılım kalitesi, sürdürülebilirlik, programlama standartlarına uyum ve yeniden düzenleme faaliyetlerinin önemi vurgulanmıştır. Ancak teknik faktörlerin paydaş memnuniyeti üzerindeki etkisi anlamlı çıkmamıştır. Bu durum, sistem performansı veya kod kalitesi gibi teknik faktörler yazılım geliştirmenin önemli bileşenleri olsa da bunların paydaş memnuniyeti üzerindeki doğrudan etkisinin hemen görülmeyebileceğini göstermektedir. Müşteriler ve üst yönetim görünür ürün iyileştirmelerine öncelik vermektedir. Ayrıca en yeni teknolojilerin benimsenmesi, bu teknolojilerle geliştirilen yeni özelliklerin iş süreçleri sahiplerinin beklentileriyle örtüşmemesi halinde müşteri memnuniyetini garanti etmez.

Bir başka sonuç ise proje faktörlerinin süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti üzerinde olumsuz bir etkiye sahip olduğunu ortaya koymuştur. Grup ve birebir toplantılarda çevik uygulayıcılar, teknolojik belirsizlik, teknik karmaşıklık, aciliyet, proje kritikliği, büyük proje boyutu ve kapsam değişikliklerinin çevik yazılım proje yönetimindeki olumsuz sonuçlarına ilişkin endişelerini dile getirmişlerdir. Teknik karmaşıklık, tasarım ve yazılım geliştirmede gecikmelere ve maliyet artışlarına neden olmaktadır. Ayrıca büyük proje boyutu,

gecikmelere ve üretkenlik kaybına yol açarak iletişim ve koordinasyon sorunlarına neden olabilir.

Geliştirilen model, özellikle çevik yazılım projeleri üzerinde çalışan şirketler için farkındalık yaratmak açısından çok önemli bir çalışmadır. İlk olarak çevik uygulayıcılar, çevik yazılım projelerinin planlanması ve yürütülmesi sırasında önerilen modeli kullanmaktan fayda sağlayabilir. İkinci olarak çevik metodolojiler müşteri katılımı ve geri bildirimine dikkat çekmektedir. Kuruluşlar paydaşlar arasında güçlü iletişim kanalları kurabilir, net beklentiler belirleyebilir. Doğru stratejiler yardımıyla gereksinimler başarılı bir şekilde yönetilebilir. Üçüncü olarak Çevik Manifesto'ya göre, "kapsamlı dokümantasyon yerine çalışan yazılım" (Agile Manifesto, 2001a). verimliliği artırır. Öte yandan iyi belgelenmiş bir proje geçmişine sahip olmak, bilgi aktarımı ve bakım sırasında zor olabilir. Bu riski azaltmak için kuruluşlar doğru miktarda dokümantasyona yönelik kuruma özgü stratejiler belirleyebilir. Dördüncüsü, ekip üyelerinin ve müşterilerin profesyonel yeteneklerinin eğitim yoluyla geliştirilmesi, risklerin yönetilmesinde kritik öneme sahiptir. Ekip üyeleri, paydaşlar ve üst yönetim çevik kavramları ve uygulamaları eğitim yoluyla anlayabilir. Bu nedenle kuruluşlar, çalışanlarını eğitimlere, çalıştaylara ve konferanslara katılmaya teşvik etmelidir. Beşinci olarak Çevik Manifesto'nun altıncı ilkesi, bir geliştirme ekibi içinde etkili bilgi aktarımı için yüz yüze görüşmenin önemini vurgulamaktadır (Agile Manifesto, 2001b). Ancak COVID-19 salgını nedeniyle şirketlerin çalışma tarzı son zamanlarda değişmiştir. "Çevik tarzda çalışma ortamı" faktörü, uzaktan çalışma ortamında doğru yöntem ve araçlar kullanılarak bilginin verimli ve etkili bir şekilde aktarılabilceğini vurgulamaktadır.

8. ARAŞTIRMA KISITLARI VE GELECEK ÇALIŞMALAR

Bu çalışma, kabul edilmesi gereken birkaç kısıtı içermektedir. İlk olarak birebir ve grup toplantılarına katılan tüm çevik uygulayıcılar Türkiye'deki şirketlerde çalışmaktadır. Çevik uygulayıcıların farklı ülke profilleri mevcut çalışmaya katkı sağlayacağı düşünülmektedir. İkinci olarak, anket verilerinin sonuçları Yalın, XP gibi tüm çevik metodolojileri kapsamamaktadır. Sadece 15 anket katılımcısı projelerinde Yalın ve XP metodolojilerini kullandığını belirtmiştir. Katılımcıların çoğu Scrum ve Kanban kullandığı için raporlanan verilerde Scrum ve Kanban'a yönelik bir yanlılık bulunmaktadır. Üçüncü kısıtlama, sektör farklılıkları ile ilgilidir. Katılımcıların çoğu finans sektöründen gelmektedir ve anketi finansla ilgili yapılan çevik projelere göre yanıtlamışlardır.

Gelecek çalışmalarda, KBF'lerin çevik proje başarı ölçütleri üzerindeki etkileri hakkında daha kapsamlı bir anlayışa sahip olmak için farklı türde çevik proje yönetimi yaklaşımıyla çalışan çevik uygulayıcılardan daha fazla veri toplanabilir. Araştırma modeli, belirli çevik metodolojilere (örneğin, Scrum, Kanban) odaklanmıştır. Gelecek çalışmalar için farklı çevik metodolojilerin ve hibrit yaklaşımların etkinliğini karşılaştırarak, hangi metodolojilerin hangi bağlamlarda daha etkili olduğunu belirleyebilir. Bu tür çalışmalar, organizasyonların ihtiyaçlarına en uygun çevik yaklaşımları seçmelerine yardımcı olabilir. Benzer şekilde finans, pazarlama veya diğer sektörler gibi çevik proje yönetimi yaklaşımındaki sektörel farklılıkları görmek amacıyla daha fazla veri toplanabilir. Bu çalışmada kullanılan anket verileri, belirli bir coğrafi bölge ve kültürel bağlamda toplanmıştır. Gelecek araştırmalar, farklı coğrafi bölgelerde ve kültürel ortamlarda çevik proje yönetiminin nasıl farklılık gösterdiğini inceleyebilir. Özellikle, kültürel faktörlerin ve ulusal kültürlerin çevik uygulamalara etkisini araştırmak, küresel çapta geçerliliği olan bir çevik başarı modeli geliştirilmesine katkı sağlayabilir.

Tez çalışmasında daha çok nicel verilere dayalı analizler yapılmıştır. Gelecek çalışmalar, nitel araştırma yöntemlerini kullanarak, çevik uygulayıcıların deneyimlerini ve perspektiflerini daha derinlemesine inceleyebilir. Örneğin,

derinlemesine mülakatlar ve vaka çalışmaları, çevik proje yönetiminin pratik uygulamalarını ve karşılaşılan zorlukları daha ayrıntılı bir şekilde anlamamıza yardımcı olabilir. Vaka çalışmaları, belirli bir projedeki uygulamaları, zorlukları ve başarı hikayelerini inceleyerek çevik yönetim yaklaşımlarının pratikte nasıl işlediğine dair detaylı veriler sağlayabilir.

Gelecek çalışmalarda, kritik başarı faktörlerinin önem düzeylerini belirlemek için bulanık mantık ve çok kriterli karar verme yöntemleri kullanılabilir. Bulanık mantık ve çok kriterli karar verme yöntemleri, belirsizlikleri ve subjektif değerlendirmeleri dikkate alarak daha hassas sonuçlar elde edilmesini sağlar. Ayrıca faktörlerin hiyerarşik yapısını ve göreceli önemlerini açıkça ortaya koyar. Bu yöntem, özellikle çevik yazılım projeleri gibi karmaşık ve dinamik ortamlarda kritik başarı faktörlerini değerlendirmek için uygundur.

Sürdürülebilirlik ve yazılım ürün kalitesi ile ilgili literatürü geliştirmek için daha detaylı çalışmalar yapılabilir. Bu tür araştırmalar, sürdürülebilir yazılım geliştirme süreçlerinin nasıl optimize edilebileceğini ve sürdürülebilirlik ilkelerinin proje yönetimine nasıl entegre edilebileceğini inceleyebilir. Özellikle yazılım ürün kalitesini artırmak için çevik yöntemlerin nasıl kullanılabileceği ve bu süreçlerin uzun vadeli etkileri üzerine odaklanan çalışmalar yapılabilir. Çevik projelerde sürdürülebilirlik uygulamalarının etkilerini değerlendirmek için vaka çalışmaları ve pratik örnekler üzerinden analizler yapılabilir.

Süreç verimliliği, sürdürülebilir yazılım ürün kalitesi ve paydaş memnuniyeti için açıklanan varyanslar sırasıyla %51, %39 ve %51'dir. Bu bulgular olumlu olsa da proje başarı ölçütleri üzerindeki etkilerini ortaya çıkarmak için ek faktörler incelenebilir. Ayrıca tez çalışmasında çevik proje yönetimi üzerinde odaklanılan faktörlerin çoğu ekip içi dinamikler ve proje süreçleri ile ilgilidir. Gelecek çalışmalar, dış paydaşların (örneğin, müşteriler, tedarikçiler) rolünü ve bu paydaşlarla olan ilişkilerin proje başarısına etkisini inceleyebilir. Bu tür araştırmalar, çevik proje yönetiminde dış paydaşların katılımını iyileştirme konusunda değerli bilgiler sağlayabilir.

Mevcut çalışmaya risk ve başarısızlık faktörleri dahil edilmiştir. Ancak potansiyel zorluklarla ilgili ölçeklenebilirlik sorunları veya projenin büyümesiyle çevikliği korumadaki zorluklar gibi konuların stratejik planlama ve çevik uygulamaların sürekli rafine edilmesi yoluyla titizlikle ele alındığı daha ayrıntılı bir çalışma yapılabilir.

Tez çalışması, Covid-19 pandemisinin çevik proje yönetimi üzerindeki etkilerini de ele almaktadır. Ancak pandeminin uzun vadeli etkileri henüz tam olarak anlaşılmamıştır. Gelecek çalışmalar, pandemi sonrası dönemde çevik proje yönetiminin nasıl evrileceğini ve yeni çalışma modellerinin (örneğin, uzaktan çalışma, hibrit çalışma) çevik uygulamalara nasıl entegre edilebileceğini araştırabilir. Özellikle, uzaktan çalışma ortamında etkili bilgi aktarımı ve işbirliği stratejileri geliştirilmesi üzerine çalışmalar yapılabilir.





9. SONUÇ

Bu çalışmada, son 24 yılın sistematik bir literatür taraması, bire bir ve grup toplantıları aracılığıyla çevik uygulayıcıların girdileri ve analizlerden elde edilen Çevik Yazılım Proje Başarı Modeli, çevik yazılım geliştirme ve yönetimi için kritik başarı faktörlerini ve karmaşık ilişkileri tanımlamaktadır. Çevik kritik başarı faktörleri “Müşteri faktörleri”, “Ekip faktörleri”, “Organizasyon faktörleri”, “Çevik Süreç faktörleri”, “Teknik faktörler” ve “Proje faktörleri” olarak tanımlanmıştır. Ayrıca çevik proje başarı ölçütleri “Süreç Verimliliği”, “Sürdürülebilir Yazılım Ürün Kalitesi” ve “Paydaş Memnuniyeti” olarak tanımlanmıştır. Önceki çalışmalardan farklı olarak, bu çalışma ampirik doğrulama ve pratik uygulanabilirlik üzerinde durarak araştırmacılar ve uygulayıcılar için yol gösterici olacak bir kaynak sunmaktadır. Çalışma, teorik içgörülerle pratik bilgileri birleştirerek ve çevik proje başarı yönetimindeki önemli bir boşluğu doldurarak bu alanda bir başarıyı ortaya koymaktadır. Çalışma, çevik projelerin başarısındaki faktörlerin anlaşılmasının önemini vurgulamakta ve akademideki araştırmacılar ve çevik uygulayıcılar için çevik projelerin başarısı adına faydalı bir rehber niteliğindedir.



KAYNAKLAR

- Abrar, M. F., Khan, M. S., Ali, S., Ali, U., Majeed, M. F., Ali, A., Amin, B., & Rasheed, N.** (2019). Motivators for large-scale agile adoption from management perspective: A systematic literature review. *IEEE Access*, 7, 22660–22674. <https://doi.org/10.1109/ACCESS.2019.2896212>
- Agile Alliance.** (2023). Agile 101. Çevrimiçi erişilebilir: <https://www.agilealliance.org/agile101/> (Erişim tarihi: 22 Haziran 2023).
- Agile Manifesto.** (2001a). Manifesto for Agile Software Development. <http://agilemanifesto.org/> (Erişim tarihi: 10 Ocak 2023).
- Agile Manifesto.** (2001b). Principles behind the Agile Manifesto. <http://agilemanifesto.org/principles.html> (Erişim tarihi: 10 Ocak 2023).
- Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U.** (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, 28(1), 7–33.
- Ahimbisibwe, A., Daellenbach, U., & Cavana, R. Y.** (2017). Empirical comparison of traditional plan-based and agile methodologies: Critical success factors for outsourced software development projects from vendors' perspective. *Journal of Enterprise Information Management*, 30(3), 400–453. <https://doi.org/10.1108/JEIM-06-2015-0056>
- Ahmad, U., Ibrahim, Y., & Minai, M. S.** (2018). Malaysian public–private partnerships: Risk management in build, lease, maintain and transfer projects. *Cogent Business & Management*, 5(1), 1–25. <https://doi.org/10.1080/23311975.2018.1550147>
- Aizaz, F., Khan, S. U. R., Khan, J. A., Inayat-Ur-Rehman, & Akhunzada, A.** (2021). An empirical investigation of factors causing scope creep in agile global software development context: A conceptual model for project managers. *IEEE Access*, 9, 109166–109195. <https://doi.org/10.1109/ACCESS.2021.3100779>
- Anthony Jnr, B.** (2019). Validating the usability attributes of AHP-software risk prioritization model using partial least square-structural equation modeling. *Journal of Science and Technology Policy Management*, 10(2), 404–430. <https://doi.org/10.1108/JSTPM-06-2018-0060>
- Arcos-Medina, G., & Mauricio, D.** (2019). Aspects of software quality applied to the process of agile software development: A systematic literature review. *International Journal of System Assurance Engineering and Management*, 10(5), 867–897. <https://doi.org/10.1007/s13198-019-00840-7>

- Bagozzi, R. P., Yi, Y., & Phillips, L. W.** (1991). Assessing construct validity in organizational research. *Administrative Science Quarterly*, 36(3), 421–458. <https://doi.org/10.2307/2393203>
- Beck, K.** (1999). Embracing change with extreme programming. *Computer*, 32(10), 70–77. <https://doi.org/10.1109/2.796139>
- Bentler, P.** (1990). *Comparative fit indices in structural models*. UCLA Department of Statistics Technical Reports, pp. 238–246.
- Bollen, K. A.** (1989). *Structural equations with latent variables*. John Wiley & Sons.
- Browne, M. W., & Cudeck, R.** (1992). Alternative ways of assessing model fit. *Sociological Methods & Research*, 21(2), 230–258. <https://doi.org/10.1177/0049124192021002005>
- Chan, F. K. Y., & Thong, J. Y. L.** (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*, 46(4), 803–814. <http://10.0.3.248/j.dss.2008.11.009>
- Chapin, N., Hale, J. E., Khan, K. M., Ramil, J. F., & Tan, W. G.** (2001). Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution: Research and Practice*, 13, 3–30. <https://doi.org/10.1002/smr.220>
- Chow, T., & Cao, D. B.** (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961–971. <https://doi.org/10.1016/j.jss.2007.08.020>
- Ciric Lalic, D., Lalic, B., Delić, M., Gracanin, D., & Stefanovic, D.** (2022). How project management approach impact project success? From traditional to agile. *International Journal of Managing Projects in Business*, 15(3), 494–521. <https://doi.org/10.1108/IJMPB-04-2021-0108>
- Cockburn, A.** (2007). *Agile Software Development: The Cooperative Game*. Upper Saddle River, NJ: Addison-Wesley.
- Cohn, M., & Ford, D.** (2003). Introducing an agile process to an organization [Software Development]. *Computer*, 36(6), 74–78. <https://doi.org/10.1109/MC.2003.1204378>
- Condori-Fernandez, N., & Lago, P.** (2018). Characterizing the contribution of quality requirements to software sustainability. *Journal of Systems and Software*, 137, 289–305. <https://doi.org/10.1016/j.jss.2017.12.005>
- Condori-Fernandez, N., Lago, P., Luaces, M. R., & Places, Á. S.** (2020). An action research for improving the sustainability assessment framework instruments. *Sustainability*, 12(4), 1–25. <https://doi.org/10.3390/su12041682>
- Crispin, L., & Gregory, J.** (2008). *Agile testing: A practical guide for testers and agile teams* (1st ed., pp. 97-108). Addison-Wesley Professional.
- Cucolaş, A. A., & Russo, D.** (2023). The impact of working from home on the success of Scrum projects: A multi-method study. *Journal of Systems and Software*, 197, 111562. <https://doi.org/10.1016/j.jss.2022.111562>

- Dorton, S. L., Maryeski, L. R., Ogren, L., Dykens, I. T., & Main, A.** (2020). A wargame-augmented knowledge elicitation method for the agile development of novel systems. *Systems*, 8(3), 27. <https://doi.org/10.3390/systems8030027>
- El Emam, K., & Koru, A. G.** (2008). A replicated survey of IT software project failures. *IEEE Software*, 25(5). <https://doi.org/10.1109/MS.2008.107>
- Engelbrecht, J., Johnston, K. A., & Hooper, V.** (2017). The influence of business managers' IT competence on IT project success. *International Journal of Project Management*, 35(6), 994–1005. <https://doi.org/10.1016/j.ijproman.2017.04.016>
- Erdoğan, O., Pekkaya, M. E., & Gök, H.** (2018). More effective sprint retrospective with statistical analysis. *Journal of Software: Evolution and Process*, 30(5), 1–9. <https://doi.org/10.1002/smr.1933>
- Fatema, I., & Sakib, K.** (2017). Factors influencing productivity of agile software development teamwork: A qualitative system dynamics approach. In *Proceedings of the 2017 24th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 737–742). IEEE. <https://doi.org/10.1109/APSEC.2017.95>
- Fornell, C., & Larcker, D. F.** (1981). Structural equation models with unobservable variables and measurement error: Algebra and statistics. *Journal of Marketing Research*, 18(3), 382–388. <https://doi.org/10.2307/3150980>
- Fortune, J., & White, D.** (2006). Framing of project critical success factors by a systems model. *International Journal of Project Management*, 24(1), 53–65. <https://doi.org/10.1016/j.ijproman.2005.07.004>
- Forza, C., & Filippini, R.** (1998). TQM impact on quality conformance and customer satisfaction: A causal model. *International Journal of Production Economics*, 55(1), 1–20. [https://doi.org/10.1016/S0925-5273\(98\)00007-3](https://doi.org/10.1016/S0925-5273(98)00007-3).
- Fowler, M., & Beck, K.** (1999). *Refactoring: Improving the design of existing code*. Addison-Wesley: Reading, MA.
- Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., & Demirörs, O.** (2018). Correlation of critical success factors with success of software projects: an empirical investigation. *Software Quality Journal*. <https://doi.org/10.1007/s11219-018-9419-5>
- Gefen, D., Straub, D., & Boudreau, M.-C.** (2000). Structural equation modeling and regression: Guidelines for research practice. *Communications of the Association for Information Systems*, 4, 7. Retrieved from <http://aisel.aisnet.org/cais/vol4/iss1/7>
- Google Scholar.** (2022). Atıflar [Google Scholar citations]. Erişim tarihi 30 Eylül 2022. <https://scholar.google.com/intl/en/scholar/citations.html>
- Hair, J. F., Anderson, R. E., Tatham, R. L., & Black, W. C.** (1998). *Multivariate data analysis with readings*. Prentice-Hall.
- Hair, J. F., Hult, G. T., Ringle, C., & Sarstedt, M.** (2016). *A Primer on Partial Least Squares Structural Equation Modeling (PLS-SEM)*. Sage Publications: Thousand Oaks, CA, USA. ISBN 9781483377445.

- Hair, J.F., Black, W.C., Babin, B.J., & Anderson, R.E.** (2010). *Multivariate Data Analysis*. 7th Edition, Pearson, New York.
- Henseler, J., Ringle, C. M., & Sinkovics, R. R.** (2009). The use of partial least squares path modeling in international marketing. *Advances in International Marketing*, 20, 277–319. [https://doi.org/10.1108/S1474-7979\(2009\)0000020014](https://doi.org/10.1108/S1474-7979(2009)0000020014)
- Hirsch, J. E.** (2005). An Index to Quantify an Individual's Scientific Research Output. *Proc. Natl. Acad. Sci. U. S. A.*, 102(46), 16569–16572.
- Howell, D., Windahl, C., & Seidel, R.** (2010). A project contingency framework based on uncertainty and its consequences. *International Journal of Project Management*, 28(3), 256–264. <https://doi.org/10.1016/j.ijproman.2009.06.002>
- Huang, S.-J., & Han, W.-M.** (2008). Exploring the relationship between software project duration and risk exposure: A cluster analysis. *Information & Management*, 45(3), 175–182. <https://doi.org/10.1016/j.im.2008.02.001>
- Hughes, D. L., Rana, N. P., & Dwivedi, Y. K.** (2020). Elucidation of IS project success factors: An interpretive structural modelling approach. *Annals of Operations Research*, 285(1–2), 35–66. <https://doi.org/10.1007/s10479-019-03146-w>
- Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S.** (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51, 915–929. <https://doi.org/10.1016/j.chb.2014.10.046>
- International Organization for Standardization.** (2011). ISO/IEC 25010:2011, Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (Square)—System and Software Quality Models. ISO.
- International Software Testing Qualifications Board (ISTQB®).** (2022). *Standard glossary of terms used in software testing (Version 3.5)*. Erişim tarihi 7 Haziran 2022. <https://www.istqb.org/>
- ISO.** (2018). *ISO 31000:2018; Risk Management—Principles and Guidelines*. ISO: Geneva, Switzerland.
- Johnson, J.** (2021). CHAOS 2020: Beyond Infinity Overview. The Standish Group International.
- Joreskog, K. G., & Sorbom, D.** (1984). *Advances in factor analysis and structural equation models*. Rowman & Littlefield Publishers.
- Joslin, R., & Müller, R.** (2015). Relationships between a project management methodology and project success in different project governance contexts. *International Journal of Project Management*, 33(6), 1377–1392. <https://doi.org/10.1016/j.ijproman.2015.03.005>
- Joslin, R., & Müller, R.** (2016). The relationship between project governance and project success. *International Journal of Project Management*, 34(4), 613–626. <https://doi.org/10.1016/j.ijproman.2016.01.008>

- Jun, L., Qiuzhen, W., & Qingguo, M.** (2011). The effects of project uncertainty and risk management on IS development project performance: A vendor perspective. *International Journal of Project Management*, 29(7), 923–933. <https://doi.org/10.1016/j.ijproman.2010.11.002>
- Kaiser, H. F.** (1974). An index of factorial simplicity. *Psychometrika*, 39(1), 31–36. <https://doi.org/10.1007/BF02291575>
- Kanjanda, T., & Tuan, N.-T.** (2020). A systemic exploration of the risk factors in Zimbabwean information technology projects. *Systems Practice and Action Research*, 33(1), 77–93. <https://doi.org/10.1007/s11213-019-09515-7>
- Khalifeh, A., Farrell, P., Alrousan, M., Alwardat, S., & Faisal, M.** (2020). Incorporating sustainability into software projects: A conceptual framework. *International Journal of Managing Projects in Business*, 13(6), 1339–1361. <https://doi.org/10.1108/IJMPB-12-2019-0289>
- Kitchenham, B., & Charters, C.** (2007). *Guidelines for performing systematic literature reviews in software engineering*. Keele University and Durham University Joint Report — EBSE 2007-001.
- Klinder, J. A.-C., Prenner, N., Schneider, K., & Hohl, P.** (2019). Transformation towards agile software product line engineering in large companies: A literature review. *Journal of Software: Evolution and Process*, 31(5). <https://doi.org/10.1002/smr.2168>
- Koch, A. S.** (2004). *Agile software development: Evaluating the methods for your organization*. Artech House.
- Leffingwell, D.** (2008). *Scaling software agility: Best practices for large enterprises*. Addison-Wesley: Upper Saddle River, NJ.
- Lehtinen, T. O. A., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C.** (2014). Perceived causes of software project failures – An analysis of their relationships. *Information and Software Technology*, 56(6), 623–643. <https://doi.org/10.1016/j.infsof.2014.01.015>
- Lemon, W. F., Liebowitz, J., Burn, J., & Hackney, R.** (2002). Information systems project failure: A comparative study of two countries. *Journal of Global Information Management*, 10(2), 28–39. <https://doi.org/10.4018/jgim.2002040103>
- Lindvall, M., Basili, V., Boehm, B., Costa, P., Dangle, K., Shull, F., Tesoriero, R., Williams, L., & Zelkowitz, M.** (2002). Empirical Findings in Agile Methods - *Extreme Programming and Agile Methods - XP/Agile Universe 2002* (D. Wells & L. Williams (eds.); pp. 197–207). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45672-4_19
- Lunesu, M. I., Tonelli, R., Marchesi, L., & Marchesi, M.** (2021). Assessing the risk of software development in agile methodologies using simulation. *IEEE Access*, 9, 134240–134258. <https://doi.org/10.1109/ACCESS.2021.3115941>

- Mann, C., & Maurer, F.** (2005). A case study on the impact of Scrum on overtime and customer satisfaction. In *Proceedings of the Agile Development Conference (ADC'05), Agile Conference*, 2005 (pp. 70–79). IEEE. <https://doi.org/10.1109/ADC.2005.1>
- Marsh, H. W., & Hocevar, D.** (1985). Application of confirmatory factor analysis to the study of self-concept: First- and higher-order factor models and their invariance across groups. *Psychological Bulletin*, 97(3), 562–582. <https://doi.org/10.1037/0033-2909.97.3.562>
- Mata, M. N., Martins, J. M., & Inácio, P. L.** (2023). Impact of absorptive capacity on project success through mediating role of strategic agility: Project complexity as a moderator. *Journal of Innovation & Knowledge*, 8(1). <https://doi.org/10.1016/j.jik.2023.100327>
- Menezes, J., Gusmão, C., & Moura, H.** (2019). Risk factors in software development projects: A systematic literature review. *Software Quality Journal*, 27(3), 1149–1174. <https://doi.org/10.1007/s11219-018-9427-5>
- Milosevic, D., & Patanakul, P.** (2005). Standardized project management may increase development projects success. *International Journal of Project Management*, 23(3), 181–192. <https://doi.org/10.1016/j.ijproman.2004.11.002>
- Misra, S. C., Kumar, V., & Kumar, U.** (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869–1890. <https://doi.org/10.1016/j.jss.2009.05.052>
- Moe, N. B., Dingsøyr, T., & Dybå, T.** (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480–491. <https://doi.org/10.1016/j.infsof.2009.11.004>
- Mooney, J. D.** (2004). *Developing portable software*. In R. Reis (Ed.), *Information technology* (pp. 55-84). Springer US.
- Nasir, M. H. N., & Sahibuddin, S.** (2011). Critical Success Factors for Software Projects: A Comparative Study. *Scientific Research and Essays*, 6(10), 2174–2186. <https://doi.org/10.5897/sre10.1171>
- Ohno, T.** (2017). *Toyota Ruhu*. İstanbul: Scala Yayıncılık.
- Papke-Shields, K. E., & Boyer-Wright, K. M.** (2017). Strategic planning characteristics applied to project management. *International Journal of Project Management*, 35(2), 169–179. <https://doi.org/10.1016/j.ijproman.2016.10.015>
- Perera, C., & Perera, I.** (2019). The impact of client involvement towards agile project success in Sri Lankan software industry. In *MERCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference* (pp. 279–284). IEEE. <https://doi.org/10.1109/MERCon.2019.8818800>

- Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A.** (2017). Assisting the continuous improvement of scrum projects using metrics and Bayesian networks. *Journal of Software: Evolution and Process*, 29(6), 1–17. <https://doi.org/10.1002/smr.1835>
- Project Management Institute.** (2013). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)* (5th ed.). PMI.
- Purna Sudhakar, G.** (2012). A model of critical success factors for software projects. *Journal of Enterprise Information Management*, 25(6), 537–558. <https://doi.org/10.1108/17410391211272829>
- Rashid, N., & Khan, S. U.** (2018). Using agile methods for the development of green and sustainable software: Success factors for GSD vendors. *Journal of Software: Evolution and Process*, 30(8), 1–29. <https://doi.org/10.1002/smr.1927>
- Rashid, N., Khan, S. U., Khan, H. U., & Ilyas, M.** (2021). Green-agile maturity model: An evaluation framework for global software development vendors. *IEEE Access*, 9, 71868–71886. <https://doi.org/10.1109/ACCESS.2021.3079194>
- Reeves, J. D., Eveleigh, T., Holzer, T. H., & Sarkani, S.** (2013). Risk identification biases and their impact to space system development project performance. *Engineering Management Journal*, 25(2), 3–12. <https://doi.org/10.1080/10429247.2013.11431970>
- Rico, D. F., Sayani, H. H., & Field, R. F.** (2008). History of computers, electronic commerce and agile methods. *Advances in Computers*, 73, 1–55. [https://doi.org/10.1016/S0065-2458\(08\)00401-4](https://doi.org/10.1016/S0065-2458(08)00401-4)
- Schmidt, R., Lyytinen, K., Keil, M., & Cule, P.** (2001). Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, 17(4), 5–36. <https://doi.org/10.1080/07421222.2001.11045662>
- Scimago Journal & Country Rank (SJR).** (2022). Scimago Journal & Country Rank (SJR). Erişim tarihi 29 Eylül 2022. <https://www.scimagojr.com/aboutus.php>
- Serrador, P., & Pinto, J. K.** (2015). Does agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- Sheffield, J., & Lemétayer, J.** (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, 31(3), 459–472. <https://doi.org/10.1016/j.ijproman.2012.09.011>
- Shrivastava, S. V., & Rathod, U.** (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58, 373–387. <https://doi.org/10.1016/j.infsof.2014.07.007>
- Shrivastava, S. V., & Rathod, U.** (2017). A risk management framework for distributed agile projects. *Information and Software Technology*, 85, 1–15. <https://doi.org/10.1016/j.infsof.2016.12.005>

- Silva, K. M. B. D., & Santos, S. C. D.** (2015). Critical factors in agile software projects according to people, process and technology perspective. In *Proceedings of the 6th Brazilian Workshop on Agile Methods, WBMA 2015* (pp. 48–54). IEEE. <https://doi.org/10.1109/WBMA.2015.19>
- Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B.** (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *The Journal of Systems and Software*, 86(6), 1663–1678. <https://doi.org/10.1016/j.jss.2013.02.027>
- Straub, D. W.** (1989). Validating instruments in MIS research. *MIS Quarterly*, 13(2), 147–169. <https://doi.org/10.2307/249008>
- Tam, C., Moura, E. J. da C., Oliveira, T., & Varajão, J.** (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, 38(3), 165–176. <https://doi.org/10.1016/j.ijproman.2020.02.001>
- Taromirad, M., & Ramsin, R.** (2008). An appraisal of existing evaluation frameworks for agile methodologies. In *15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008)* (pp. 418–427). IEEE. <https://doi.org/10.1109/ECBS.2008.32>
- Tavares, B. G., da Silva, C. E. S., & de Souza, A. D.** (2019). Risk management analysis in Scrum software projects. *International Transactions in Operational Research*, 26(5), 1884–1905. <https://doi.org/10.1111/itor.12401>
- Teslyuk, V., Batyuk, A., & Voityshyn, V.** (2022). Method of software development project duration estimation for Scrum teams with differentiated specializations. *Systems*, 10(4), 123. <https://doi.org/10.3390/systems10040123>
- The Scrum Guide.** (2020). The 2020 Scrum Guide. <https://scrumguides.org/scrum-guide.html> (Erişim tarihi: 11 Aralık 2022).
- The Standish Group.** (2015). CHAOS Report 2015. Erişim tarihi 7 Temmuz 2023. https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf
- Turner, R., & Boehm, B.** (2003). Using risk to balance agile and plan-driven methods. *Computer*, 36(6), 57–66. <https://doi.org/10.1109/MC.2003.1204376>
- Wan, J., & Wang, R.** (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications*, 3, 1131–1140. <https://doi.org/10.4236/jsea.2010.312132>
- Yeo, K. T.** (2002). Critical failure factors in information system projects. *International Journal of Project Management*, 20(3), 241–246. [https://doi.org/10.1016/S0263-7863\(01\)00075-8](https://doi.org/10.1016/S0263-7863(01)00075-8)
- Yetton, P., Sharma, R., Johnston, K., & Martin, A.** (2000). A model of information systems development project performance. *Information Systems Journal*, 10(4), 263–289. <https://doi.org/10.1046/j.1365-2575.2000.00088.x>

EKLER

EK A: Seçilen makaleler

EK B: Anket soruları



- [1] Abrar, M. F., Khan, M. S., Ali, S., Ali, U., Majeed, M. F., Ali, A., Amin, B., & Rasheed, N. (2019). Motivators for large-scale agile adoption from management perspective: A systematic literature review. *IEEE Access*, 7, 22660–22674. <https://doi.org/10.1109/ACCESS.2019.2896212>
- [2] Ahimbisibwe, A., Cavana, R. Y., & Daellenbach, U. (2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies. *Journal of Enterprise Information Management*, 28(1), 7–33.
- [3] Ahimbisibwe, A., Daellenbach, U., & Cavana, R. Y. (2017). Empirical comparison of traditional plan-based and agile methodologies: Critical success factors for outsourced software development projects from vendors' perspective. *Journal of Enterprise Information Management*, 30(3), 400–453. <https://doi.org/10.1108/JEIM-06-2015-0056>
- [4] Anthony Jnr, B. (2019). Validating the usability attributes of AHP-software risk prioritization model using partial least square-structural equation modeling. *Journal of Science and Technology Policy Management*, 10(2), 404–430. <https://doi.org/10.1108/JSTPM-06-2018-0060>
- [5] Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: A systematic literature review. *International Journal of System Assurance Engineering and Management*, 10(5), 867–897. <https://doi.org/10.1007/s13198-019-00840-7>
- [6] Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961–971. <https://doi.org/10.1016/j.jss.2007.08.020>
- [7] Engelbrecht, J., Johnston, K. A., & Hooper, V. (2017). The influence of business managers' IT competence on IT project success. *International Journal of Project Management*, 35(6), 994–1005. <https://doi.org/10.1016/j.ijproman.2017.04.016>
- [8] Fatema, I., & Sakib, K. (2017). Factors influencing productivity of agile software development teamwork: A qualitative system dynamics approach. In *Proceedings of the 2017 24th Asia-Pacific Software Engineering Conference (APSEC)* (pp. 737–742). IEEE. <https://doi.org/10.1109/APSEC.2017.95>
- [9] Fortune, J., & White, D. (2006). Framing of project critical success factors by a systems model. *International Journal of Project Management*, 24(1), 53–65. <https://doi.org/10.1016/j.ijproman.2005.07.004>
- [10] Garousi, V., Tarhan, A., Pfahl, D., Coşkunçay, A., & Demirörs, O. (2018). Correlation of critical success factors with success of software projects: an empirical investigation. *Software Quality Journal*. <https://doi.org/10.1007/s11219-018-9419-5>

- [11] Howell, D., Windahl, C., & Seidel, R. (2010). A project contingency framework based on uncertainty and its consequences. *International Journal of Project Management*, 28(3), 256–264. <https://doi.org/10.1016/j.ijproman.2009.06.002>
- [12] Hughes, D. L., Rana, N. P., & Dwivedi, Y. K. (2020). Elucidation of IS project success factors: An interpretive structural modelling approach. *Annals of Operations Research*, 285(1–2), 35–66. <https://doi.org/10.1007/s10479-019-03146-w>
- [13] Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in Human Behavior*, 51, 915–929. <https://doi.org/10.1016/j.chb.2014.10.046>
- [14] Joslin, R., & Müller, R. (2015). Relationships between a project management methodology and project success in different project governance contexts. *International Journal of Project Management*, 33(6), 1377–1392. <https://doi.org/10.1016/j.ijproman.2015.03.005>
- [15] Joslin, R., & Müller, R. (2016). The relationship between project governance and project success. *International Journal of Project Management*, 34(4), 613–626. <https://doi.org/10.1016/j.ijproman.2016.01.008>
- [16] Kanjanda, T., & Tuan, N.-T. (2020). A systemic exploration of the risk factors in Zimbabwean information technology projects. *Systems Practice and Action Research*, 33(1), 77–93. <https://doi.org/10.1007/s11213-019-09515-7>
- [17] Lehtinen, T. O. A., Mäntylä, M. V., Vanhanen, J., Itkonen, J., & Lassenius, C. (2014). Perceived causes of software project failures – An analysis of their relationships. *Information and Software Technology*, 56(6), 623–643. <https://doi.org/10.1016/j.infsof.2014.01.015>
- [18] Menezes, J., Gusmão, C., & Moura, H. (2019). Risk factors in software development projects: A systematic literature review. *Software Quality Journal*, 27(3), 1149–1174. <https://doi.org/10.1007/s11219-018-9427-5>
- [19] Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869–1890. <https://doi.org/10.1016/j.jss.2009.05.052>
- [20] Perera, C., & Perera, I. (2019). The impact of client involvement towards agile project success in Sri Lankan software industry. In *MERCon 2019 - Proceedings, 5th International Multidisciplinary Moratuwa Engineering Research Conference* (pp. 279–284). IEEE. <https://doi.org/10.1109/MERCon.2019.8818800>
- [21] Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A. (2017). Assisting the continuous improvement of scrum projects using metrics and Bayesian networks. *Journal of Software: Evolution and Process*, 29(6), 1–17. <https://doi.org/10.1002/smr.1835>

- [22] **Rashid, N., & Khan, S. U.** (2018). Using agile methods for the development of green and sustainable software: Success factors for GSD vendors. *Journal of Software: Evolution and Process*, 30(8), 1–29. <https://doi.org/10.1002/smr.1927>
- [23] **Rashid, N., Khan, S. U., Khan, H. U., & Ilyas, M.** (2021). Green-agile maturity model: An evaluation framework for global software development vendors. *IEEE Access*, 9, 71868–71886. <https://doi.org/10.1109/ACCESS.2021.3079194>
- [24] **Serrador, P., & Pinto, J. K.** (2015). Does agile work? - A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. <https://doi.org/10.1016/j.ijproman.2015.01.006>
- [25] **Sheffield, J., & Lemétayer, J.** (2013). Factors associated with the software development agility of successful projects. *International Journal of Project Management*, 31(3), 459–472. <https://doi.org/10.1016/j.ijproman.2012.09.011>
- [26] **Shrivastava, S. V., & Rathod, U.** (2015). Categorization of risk factors for distributed agile projects. *Information and Software Technology*, 58, 373–387. <https://doi.org/10.1016/j.infsof.2014.07.007>
- [27] **Shrivastava, S. V., & Rathod, U.** (2017). A risk management framework for distributed agile projects. *Information and Software Technology*, 85, 1–15. <https://doi.org/10.1016/j.infsof.2016.12.005>
- [28] **Silva, K. M. B. D., & Santos, S. C. D.** (2015). Critical factors in agile software projects according to people, process and technology perspective. In *Proceedings of the 6th Brazilian Workshop on Agile Methods, WBMA 2015* (pp. 48–54). IEEE. <https://doi.org/10.1109/WBMA.2015.19>
- [29] **Stankovic, D., Nikolic, V., Djordjevic, M., & Cao, D.-B.** (2013). A survey study of critical success factors in agile software projects in former Yugoslavia IT companies. *The Journal of Systems and Software*, 86(6), 1663–1678. <https://doi.org/10.1016/j.jss.2013.02.027>
- [30] **Purna Sudhakar, G.** (2012). A model of critical success factors for software projects. *Journal of Enterprise Information Management*, 25(6), 537–558. <https://doi.org/10.1108/17410391211272829>
- [31] **Tam, C., Moura, E. J. da C., Oliveira, T., & Varajão, J.** (2020). The factors influencing the success of on-going agile software development projects. *International Journal of Project Management*, 38(3), 165–176. <https://doi.org/10.1016/j.ijproman.2020.02.001>
- [32] **Wan, J., & Wang, R.** (2010). Empirical Research on Critical Success Factors of Agile Software Process Improvement. *Journal of Software Engineering and Applications*, 3, 1131–1140. <https://doi.org/10.4236/jsea.2010.312132>
- [33] **Yeo, K. T.** (2002). Critical failure factors in information system projects. *International Journal of Project Management*, 20(3), 241–246. [https://doi.org/10.1016/S0263-7863\(01\)00075-8](https://doi.org/10.1016/S0263-7863(01)00075-8)

EK B : Anket Soruları**Çizelge B.1 : Anket soruları.**

Kavram	Kısaltma	Anket sorusu
Organizasyonel	ORGN1	Çevik proje ekibi üyeleri, çevik tarzda bir çalışma ortamında çalıştılar (Örnek: fiziksel çevik bir çalışma ortamı, uzaktan/yerinde eşli programlama için uygun çalışma ortamı, çevrimiçi görüntülü konuşma uygulamaları ve masaüstü paylaşımı gibi araçların kullanımı) (Stankovic ve diğ, 2013).
	ORGN2	Hiyerarşik bir yapı yerine, organizasyonel olarak kabul edilmiş işbirlikçi bir yönetim tarzı vardı (Stankovic ve diğ, 2013).
	ORGN3	Organizasyonel olarak kabul edilmiş bir değişim yönetimi yapısı vardı (Ahimbisibwe ve diğ, 2017).
	ORGN4	Organizasyon ve projenin vizyonu&misyonu iyi tanımlanmıştı (Ahimbisibwe ve diğ, 2017; Garousi ve diğ, 2019).
Ekip	TEAM1	Çevik proje ekibi üyeleri teknik yeteneğe sahipti (Stankovic ve diğ, 2013).
	TEAM2	Çevik proje ekibi, deneyimli ekip üyelerinden oluşuyordu (Stankovic ve diğ, 2013; Garousi ve diğ, 2019; Hughes ve diğ, 2020).
	TEAM3	Çevik proje ekibi üyeleri, çevik prosedürler ve kendi alanları hakkında yeterli teknik eğitime sahipti (Stankovic ve diğ, 2013).
	TEAM4	Proje yöneticisi, çevik prensipler ve prosedürler hakkında bilgiye sahipti (Stankovic ve diğ, 2013).
	TEAM5	Atanan proje yönetimi, projeyi etkili bir şekilde yönetme deneyimine sahipti (Hughes ve diğ, 2020).
	TEAM6	Çevik proje ekibi üyeleri, projeyi başarıya ulaştırmak için motive olmuş ve kararlıydı (Stankovic ve diğ, 2013).
	TEAM7	Çevik proje ekibi, kendi kendine organize olan ekip üyelerinden oluşuyordu, örneğin, problemleri birlikte çözebilen ve değişen koşullara uyum sağlayabilen takım üyelerine sahipti (Stankovic ve diğ, 2013).
Müşteri	CUST1	Proje, müşteri tarafından projeye boyunca güçlü bir destek aldı (Stankovic ve diğ, 2013).
	CUST2	Çevik proje ekibi üyeleri, müşteri ile iyi bir ilişkiye sahipti (Stankovic ve diğ, 2013).
	CUST3	Proje boyunca müşterilere uygun teknik eğitim, çevik prosedürler ve kendi çalışma alanları hakkında eğitim de dahil olmak üzere sağlandı (Stankovic ve diğ, 2013; Garousi ve diğ, 2019).
	CUST4	Müşterilerin kendi alanlarındaki deneyimi yüksekti (Garousi ve diğ, 2019).

Çizelge B.1 (devam) : Anket soruları.

Kavram	Kısaltma	Anket sorusu
Çevik süreç	AGPF1	Proje, güçlü iletişim ve koordinasyona sahipti, örneğin, anlık iletişim kanalları (çevik ekip üyeleri arasında, çevik ekip üyeleri ile üst yönetim arasında ve çevik ekip üyeleri ile müşteriler arasında), döngü toplantıları vb. (Stankovic ve diğ, 2013).
	AGPF2	Proje, çevikle ilgili etkinlikleri düzgün bir şekilde uyguladı (Stankovic ve diğ, 2013).
	AGPF3	Proje, çevik dostu bir ilerleme izleme/kontrol mekanizması ile takip edildi, örneğin, ürün iş listesi, sprint iş listesi, Kanban panosu, sürüm ve iterasyon grafiği vb. (Stankovic ve diğ, 2013).
	AGPF4	Proje, yazılım geliştirme için doğru miktarda dokümantasyona sahipti (Stankovic ve diğ, 2013).
Teknik	TECH1	Proje, her iterasyon için verimli bir yazılım test sürecini takip etti (Stankovic ve diğ, 2013).
	TECH2	Çevik proje ekibi, iyi tanımlanmış bir kodlama standardını takip etti (Stankovic ve diğ, 2013).
	TECH3	Çevik proje ekibi, en iyi sonuçları sağlamak ve gereksinimlerdeki değişikliklere iyi uyum sağlamak için yeniden düzenleme faaliyetlerini (refaktoring) takip etti (Stankovic ve diğ, 2013).
	TECH4	Çevik proje ekibi, kısa sürelerde sürekli olarak çalışan yazılım teslim etti (Stankovic ve diğ, 2013).
	TECH5	Çevik proje ekibi, çevik uygulamalar için uygun platformları, teknolojileri ve araçları takip etti (Stankovic ve diğ, 2013).
	TECH6	Projede, çevik metodolojinin kullanımı için önceden risk analizi yapıldı ve değerlendirildi (Stankovic ve diğ, 2013).
	TECH7	Proje, basit tasarım izledi, örneğin, kodu ve tasarımı basit tuttu, iş listesi iyileştirme çalışması yapıldı (Stankovic ve diğ, 2013).
Proje	PRJF1	Proje, yüksek düzeyde kritik öneme sahipti (kritik projelere bir örnek, önemli miktarda paranın kaybıdır, bu durumda programdaki bir kusur nedeniyle şirket iflas edebilir).
	PRJF2	Proje, yüksek düzeyde aciliyet taşıyordu (Garousi ve diğ, 2019).
	PRJF3	Proje, yüksek düzeyde teknik karmaşıklığa sahipti (Ahimbisibwe ve diğ, 2017).
	PRJF4	Proje, teknolojik belirsizliğe sahipti, örneğin, iş gereksinimlerini karşılayan yazılım geliştirme için yerleşik prosedürler/uygulamalar yoktu (Ahimbisibwe ve diğ, 2017).

Çizelge B.1 (devam) : Anket soruları.

Kavram	Kısaltma	Anket sorusu
Proje	PRJF5	Proje başında belirlenen müşteri gereksinimleri, proje boyunca çok değişti (Ahimbisibwe ve diğ, 2017).
	PRJF6	Ekipteki kişi sayısı fazlaydı (ekip 10 veya daha az üyeden oluşur) (Ahimbisibwe ve diğ, 2017).
Süreç verimliliği	EFFC1	Proje zamanında tamamlandı (Ahimbisibwe ve diğ, 2017; Stankovic ve diğ, 2013).
	EFFC2	Proje, planlanan bütçe içinde tamamlandı (Ahimbisibwe ve diğ, 2017; Stankovic ve diğ, 2013).
	EFFC3	Proje, kararlaştırılan kapsam içinde tamamlandı (Ahimbisibwe ve diğ, 2017; Stankovic ve diğ, 2013).
Sürdürülebilir yazılım ürün kalitesi	QUAL1	Proje, yazılım güvenilirliği açısından başarılıydı, örneğin, yazılım belirli bir süre boyunca hatasız çalıştı (Ahimbisibwe ve diğ, 2017, PMI, 2013).
	QUAL2	Teslim edilen yazılım, güvenlik açısından başarılı kabul edildi, örneğin, ürünün kendisini kasıtlı veya kazara müdahaleye karşı koruma yeteneği (Garousi ve diğ, 2019; ISO, 2011).
	QUAL3	Yazılım, sürdürülebilirlik açısından başarılıydı, örneğin, işletim sistemi yükseltmeleri, sistemin uygulama hatalarını düzeltmek vb. (Garousi ve diğ, 2019; ISO, 2011).
	QUAL4	Sistem, kullanıcıların beklenen fonksiyonel gereksinimlerini karşıladı (Ahimbisibwe ve diğ, 2017; Garousi ve diğ, 2019; ISO, 2011).
	QUAL5	Proje, yazılım performans verimliliği açısından başarılıydı (ISO, 2011).
	QUAL6	Proje, yazılım uyumluluğu açısından başarılıydı, örneğin, web sitenizin Google Chrome, Firefox gibi farklı tarayıcılarla uyumluluğu (ISO, 2011).
	QUAL7	Proje, yazılım taşınabilirliği açısından başarılıydı, örneğin, programı taşımak için gereken çaba, başlangıçta uygulamak için gerekenden çabadan az ise program taşınabildir (ISO, 2011).
	QUAL8	Geliştirilen uygulama, kullanımı kolaydır (Ahimbisibwe ve diğ, 2017; ISO, 2011).
	QUAL9	Proje, kalite açısından başarılıydı (ISO, 2011; Stankovic ve diğ, 2013).
Paydaş memnuniyeti	SATF1	Müşteriler, teslim edilen sistemden memnun kaldı (Ahimbisibwe ve diğ, 2017).
	SATF2	Proje ekibi, teslim edilen sistemden memnun kaldı (Ahimbisibwe ve diğ, 2017).
	SATF3	Üst düzey yönetim, teslim edilen sistemden memnun kaldı (Ahimbisibwe ve diğ, 2017).



ÖZGEÇMİŞ

Ad-Soyad : Burcu BİNBOĞA

ÖĞRENİM DURUMU:

- **Lisans** : 2012, Gazi Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği
- **Erasmus Programı** : 2010, University of Maribor, Mühendislik Fakültesi, Endüstri Mühendisliği
- **Yükseklisans** : 2015, İstanbul Teknik Üniversitesi, Endüstri Mühendisliği Ana Bilim Dalı, Endüstri Mühendisliği

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2014-2017 yılları arasında Merkezi Kayıt Kuruluşu'nda Uygulama Geliştirme Departmanında Uzman Yardımcısı olarak çalıştı.
- 2017-2021 yılları arasında Merkezi Kayıt Kuruluşu'nda Uygulama Geliştirme Departmanında Uzman olarak çalıştı.
- 2021-2024 yılları arasında Merkezi Kayıt Kuruluşu'nda Uygulama Geliştirme Departmanında Kıdemli Uzman olarak çalıştı.
- 2024 yılından beri Merkezi Kayıt Kuruluşu'nda Uygulama Geliştirme Departmanında Baş Uzman olarak çalışmaktadır.

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Binboğa, B., Gümüşsoy, Ç.A.** 2024. The Impact of Sectoral and Job Role Differences in Agile Software Projects. *International Graduate Research Symposium - IGRS'24*, May 8-10, 2024 İstanbul, Turkey.
- **Binboga, B., Gumussoy, C.A.** 2024. Factors Affecting Agile Project Success, *IEEE Access*.